Requirements Engineering
https://doi.org/10.1007/s00766-018-0302-5

ORIGINAL ARTICLE

Understanding what is important in iStar extension proposals: the viewpoint of researchers

Enyo Gonçalves[1,2] · Marcos Antônio de Oliveira[2] · Ingrid Monteiro[2] · Jaelson Castro[1] · João Araújo[3]

Received: 6 July 2017 / Accepted: 30 June 2018

Abstract
iStar is a goal-based requirements modelling language, being used in both industrial and academic projects of different domains. Often the language is extended to incorporate new constructs related to a particular application domain or to adjust it to practical situations during requirements modelling. Currently, the language is undergoing standardisation, and several studies have focused on the analysis of iStar variations to identify similarities and to define a core. This does not imply or constrain the need for iStar to continue to be extended. This paper contributes to the understanding of how iStar is extended by analysing how iStar researchers perform iStar extensions. To address this question, we followed a qualitative approach based on interviews involving 20 researchers from different research groups that proposed iStar extensions. The analysis revealed a good understanding about what extending a modelling language means and pointed out differences about how extensions are proposed. We discovered categories that impact positively on iStar extensions (such as reusing existing extensions, proposing extensions in abstract and concrete syntaxes, and creating new modelling tools), and other categories that impact negatively (such as modifying representations of the original constructs, proposing extensions in an ad hoc fashion and not carefully choosing graphical representations). We also evaluated the findings of interviews through an online survey answered by 30 iStar researchers. Finally, we proposed a set of guidelines to support the proposal for better future iStar extensions.

Keywords Goal-based modelling · iStar · Extensions · Qualitative study · Survey

1 Introduction

Goal-oriented modelling is a relevant contribution to Requirements Engineering [19]. While object-oriented analysis fits well to the late stages of requirements analysis, the goal-oriented analysis is more appropriate for earlier stages, where *organisational goals* are analysed to identify and justify software requirements and position them within the organisational system [54]. Several goal-based modelling languages have been proposed, such as iStar [18], KAOS (Knowledge Acquisition in autOmated Specification) [19] and the NFR Framework [15]. This paper focuses on iStar, which represents a system and its context through the Strategic Dependency (SD) and Strategic Rationale (SR) models. Here we are interested in studying how the extensions of its models were accomplished.

Several iStar extensions have been proposed for specific application areas, such as data warehouse [26] and security [52]. The autonomic computing systems application domain was also targeted for iStar extensions related to configuration, optimisation, healing and protection of autonomic

✉ Enyo Gonçalves
enyo@ufc.br

✉ Jaelson Castro
jbc@cin.ufpe.br

Marcos Antônio de Oliveira
marcos.oliveira@ufc.br

Ingrid Monteiro
ingrid@ufc.br

João Araújo
joao.araujo@fct.unl.pt

[1] CIn, Universidade Federal de Pernambuco—UFPE, Av. Jornalista Aníbal Fernandes, s/n - Cidade Universitária (Campus Recife), Recife, PE 50740-560, Brazil

[2] Universidade Federal do Ceará - Campus Quixadá, Quixadá, CE 63900-000, Brazil

[3] NOVA LINCS, FCT, Universidade Nova de Lisboa, Lisbon, Portugal

Published online: 20 July 2018

applications [38]. Social-Technical Systems (STS) are another application area for iStar extensions, such as an extension involving *conflicts of interest* in health care presented in [14]. Legal aspects were also considered when representing regulations in iStar [25].

Owing to the proposal of a new version of iStar (named iStar 2.0) [18], there seems to be a suitable moment to discuss how the process of building iStar extensions could be systematised and to change the way that extensions are performed. Therefore, we are interested in understanding how iStar extensions have been developed, in order to improve this process.

This paper presents an exploratory study to understand further how iStar extensions are performed. We used mixed methods research [16] since it consists of a qualitative study based on interviews and a quantitative study based on a complementary survey.

Our goal is to provide a grounded descriptive theory about iStar extensions. The research questions that we intend to address are the following: *RQ1—How have the iStar extensions been developed?* and *RQ2—What could be done to propose better extensions?*

Our qualitative study[1] allowed us to capture valuable evidence about our research questions. It involved 20 participants who had a good understanding of what it meant to extend a modelling language and it also revealed different ways of making extensions in their several research groups. Furthermore, in this paper, we defined three main categories that are used to group a set of subcategories which affect positively or negatively the proposal of extensions. Based on the findings, we propose directions and nine guidelines for what could be done to improve the creation of new extensions to iStar.

Additionally, we performed a survey, with the other 30 iStar researchers, different from the ones who participated in the qualitative study, to evaluate the relevance of a set of 18 statements that synthesizes the findings of the qualitative study. By doing so, we carried out a triangulation of the findings. The results of the survey confirmed the importance of the findings of the qualitative study.

The results of this paper are useful to researchers who are interested in extending iStar because it helps to understand what is crucial to have well-formed and efficient iStar extensions and what should be avoided. Consequently, in the long term, we intend to define a process to conduct iStar extensions, which will be based on this qualitative study. The aim is to avoid inconsistencies in new extensions and conflicts with the base language and between different extensions, which would jeopardise the use of the language.

This paper is organised as follows. Section 2 describes some background on modelling languages, iStar and iStar extensions. Section 3 discusses the related work. Section 4 details the method used in this study. Section 5 describes the results of the qualitative study. The results of the survey are presented in Sect. 6. Threats to validity of this study are reported in Sect. 7. Finally, Sect. 8 draws some conclusions and points out directions for possible future works.

2 Background

This section introduces modelling language development and extensions, presents the iStar evolution and describes iStar extensions.

2.1 Modelling language development and extension

Researchers working in the area of modelling languages (ML) have focused on an abstraction challenge: What kind of modelling constructs and the underlying foundation are needed to support the development of domain or problem-level language constructs that are considered first-class modelling elements in a language [22]?

An ML is defined by its abstract syntax (metamodel and well-formedness rules) and concrete syntax. According to Kelly and Tolvanen [36], the constructs of an ML should be formalised by using a metamodel. Metamodelling means to model a modelling language itself: mapping the application area concepts to various language elements, their properties, and their connections, specified as links and the roles that elements play in the language. The concrete syntax is a set of words (textual) and symbols (graphical) that make it possible to create models or diagrams [29]. An example of a metamodel is presented in Fig. 2, where the iStar 2.0 metamodel is shown.

Along with modelling concepts, we also normally identify various domain rules, constraints, and consistency needs that a language should follow. These rules obviously need to be defined too. Having rules in the language provides many of the benefits of early error prevention; guides towards preferable design patterns; checks of completeness by informing about missing parts; minimises action of modelling work by conventions and default values; and maintains specifications consistency [36].

For an ML to be usable by software designers, it is necessary to define a set of models and its graphical and textual elements. They are used to render the model elements and use the abstract syntax as a starting point to concrete syntax

[1] According to Strauss and Corbin [59], the term "Qualitative research" means any research that produces findings not obtained through statistical procedures or other means of quantification, so the sample should be small to enable the analysis. It can refer to research about experiences, behaviours and perspectives about a theme and is used to understand a phenomenon [59].

definition [8]. An example of the usage of iStar 2.0 concrete syntax usage is presented in Fig. 3.

It is imperative, therefore, that models must follow a clearly defined structure, that is, they must conform to the associated metamodel representing the abstract syntax of the modelling language. The main reason for this is that the language aspects going to go beyond the language's abstract syntax, such as the definition of the visual notation to be used when modelling, which are heavily based on metamodels [8].

In several modelling environments and tools, metamodels are not known; they are kept hidden behind the user interface of the tools and, thus, are often not accessible for the modellers [8]. This happens because modelling tools are used to generate models by making use of only concrete syntax. It is important, however, that abstract syntax is available to users outside the tool, for example, in the tool's website.

Extending a modelling language means to add new modelling concepts [8]. According to the way new concepts are proposed, an extension can be developed using a *lightweight* or *heavyweight* strategy [48]. Lightweight mechanisms are a way of introducing extensions with little syntactic impact, by using textual markers to represent stereotypes, constraints and tagged values. The heavyweight extensions add new graphical representations and change the language's metamodel, therefore significantly affecting the modelling language.

In Fig. 4, the modelling of Error is an example of the usage of *lightweight* because it used a textual marker (*Error:*) in the label of an existing construct (Belief) to present a new concept, while failure is an instance of *heavyweight* because a new graphical representation is used to model it.

Given that the entities task, role and position are not included in Fig. 4's legend, neither in the metamodel presented in original paper [51], this extension can be considered as *non-conservative*.

We can classify an extension according to its impact on the original syntax of the language as "conservative", which keeps the original constructs, or "non-conservative", which changes or remove constructs of the original syntax. Besides, an extension can be proposed to include representation of a particular domain or application area (e.g. intelligent agents and security), or to improve practical aspects of a language (e.g. definition of module [45], cardinality [10] and information about a task, such as time, min and max duration and date of completion [51]).

We therefore consider as an extension any changes performed in abstract syntax (i.e. metamodel and well-formedness rules), concrete syntax or both. An extension that changes the abstract syntax implies introducing new metaclasses, properties or relationships in the metamodel or creating new well-formedness rules. While an extension involving concrete syntax implies creating a new graphical representation of new nodes or links, it can also involve a complementary representation, such as compartments or textual marker (as stereotypes).

2.2 iStar

iStar is a goal-based modelling language proposed in the nineties [63]. It is an ML used to model software at requirements level. It has been extended to fit several specific application areas.

In the iStar framework, stakeholders are represented as actors that depend on each other to achieve their goals, perform tasks and provide resources. Each goal is analysed from its actor point of view, resulting in a set of dependencies between pairs of actors. iStar elements are classified as Intentional Elements (Goal, Softgoal, Task and Resource), Actors (General Actor, Role, Position and Agent) and Links (Means-end, Decomposition, Contribution and Actor Links). These elements are represented in two models: Strategic Dependency (SD) and Strategic Rationale (SR). The SD model describes the links and external dependencies among organisational actors. The SR model enables an analysis of how the goals can be fulfilled through contributions from the several actors.

iStar had some variations in its default syntax (e.g. Toronto iStar [63] and Trento iStar [64]). Efforts were made towards unifying the language notation and establishing a unique core. In this sense, we can refer to a work whose purpose was to analyse the iStar (Trento and Toronto) constructs variation performed by Horkoff et al. [30] and the definition of a reference metamodel [13].

In June 2016, iStar evolved to the version 2.0 [18]. It was the result of a discussion started in 2014 in the iStar community about the standardisation of the language. The new version was endorsed by key players from the community, although thorough validation still needs to be performed. In this new version of the language some concepts were discontinued, some changes were done, and new concepts were introduced.

The new version kept the representation of general actors, roles and agents. The intentional elements, *goal*, *task* and *resource*, were not changed. Moreover, the actor link *is-a* and *contribution* link were maintained.

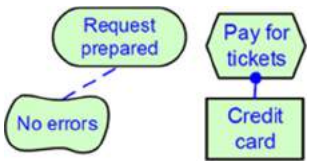


Fig. 1 Qualification and neededBy representation in iStar 2.0 [18]

Table 1 Comparison between iStar 1.0 and iStar 2.0 *Source*: https://sites.google.com/site/istarlanguage/diff

Nodes and links	iStar 1.0	iStar 2.0
Actors	General actors	General actors
	Roles, positions, agents	Roles, agents
Actor links	is-a	is-a
	is-part-of, plays, occupies, covers	Participates-in
	INS	–
Intentional elements	Goal, task, resource	Goal, task, resource
	Softgoal	Quality
Intentional elements links	Means-end, task decomposition	Refinement
	Contribution	Contribution
	–	Qualification, neededBy

The *INS* link and *Positions* were not considered in this new version. *Softgoal* was renamed as *Quality* in iStar 2.0. *Quality* is described in [18] as an attribute for which an actor desires some level of achievement. For example, in a given model, an entity could be the system under development and the *Quality* its performance. Another entity could be the business being analysed, and the *Quality* would be the yearly profit. The level of achievement may be denied precisely or kept vaguely. Qualities can guide the search for ways of achieving goals and serve as criteria for evaluating alternative ways of achieving goals.

The *is-part-of*, *plays*, *occupies*, *covers* actor links were grouped in a relationship named *participates-in,* and *means-end* and *task* decomposition were grouped in a relationship called *refinement*.

There are two kinds of refinement: *and-refinement* and *or-refinement*. The *and-refinement* is represented by a T-shaped arrow, the same representation of *task decomposition* relationship in the previous version. The *or-decomposition* is an arrow with a full head, the same as *means-end* representation in the previous iStar version.

Finally, the new *Qualification* and *neededBy* relationships were proposed. The *Qualification* relationship connects *quality* and *goals*, *tasks* and *resources*. The *neededBy* relationship connects *resources* and *tasks*, where the resource is *needed by* the related task. Figure 1 shows the graphical representations of *Qualification* (on the left-hand side) and *neededBy* (on the right-hand side).

Table 1 shows a comparison between both iStar versions.

The iStar 2.0 metamodel is presented in Fig. 2, and it shows nodes and links listed in the right column of Table 1.

A running example of iStar 2.0 is presented in [18] concerning university travel reimbursement. Students must organise their trip (e.g. to conferences) and have several goals to achieve and options related to them. To achieve their goals, students rely on other parties such as a travel agency and the university's trip management information system. In Fig. 3, we show a final view of the example to give readers an idea of some of the capabilities of iStar 2.0 [18].

2.3 iStar extensions

There are different forms in which to present an iStar[2] extension [27], but all of them introduce new concepts to iStar, for example, a set of extensions described in detail the new concepts and its representations in the iStar metamodel and concrete syntax (see works of Ali et al. [3] and Morandini et al. presented in [51]). These kinds of extensions describe how the new concepts were introduced and how to use them.

On the other hand, another set of extensions was presented together with the method to create the model, with the iStar modifications presented by illustrations with the usage of new concepts. Examples of this kind of extensions are the work of Guzman et al. [28] and the extension of Islam et al. available at [35].

In some cases, the work presents a case study or a modelling tool with a set of new concepts introduced in iStar, for example in Gans et al. [24] and Siena et al. [56]. They were selected because they are the only evidence for these extensions.

We do not consider as an extension any work that used iStar without changes in abstract syntax (changes in metamodel or validation rules) or concrete syntax (new graphical representation) because in this case the iStar is used with default syntax without any changes (extension).

In previous work [27], the iStar extensions were analysed and classified. The results point to 77.8% of extensions which extended both syntaxes are non-conservative. To check this information, it was required to analyse both the metamodel and the concrete syntax of the extensions to check this information. Therefore, only extensions that presented the abstract syntax and concrete syntax were considered in this analysis.

[2] We used the term "iStar" throughout the paper to refer to this modelling language, although the extensions presented in Sect. 2.3 extended the first version of the language, which was referred in the literature as "i*".

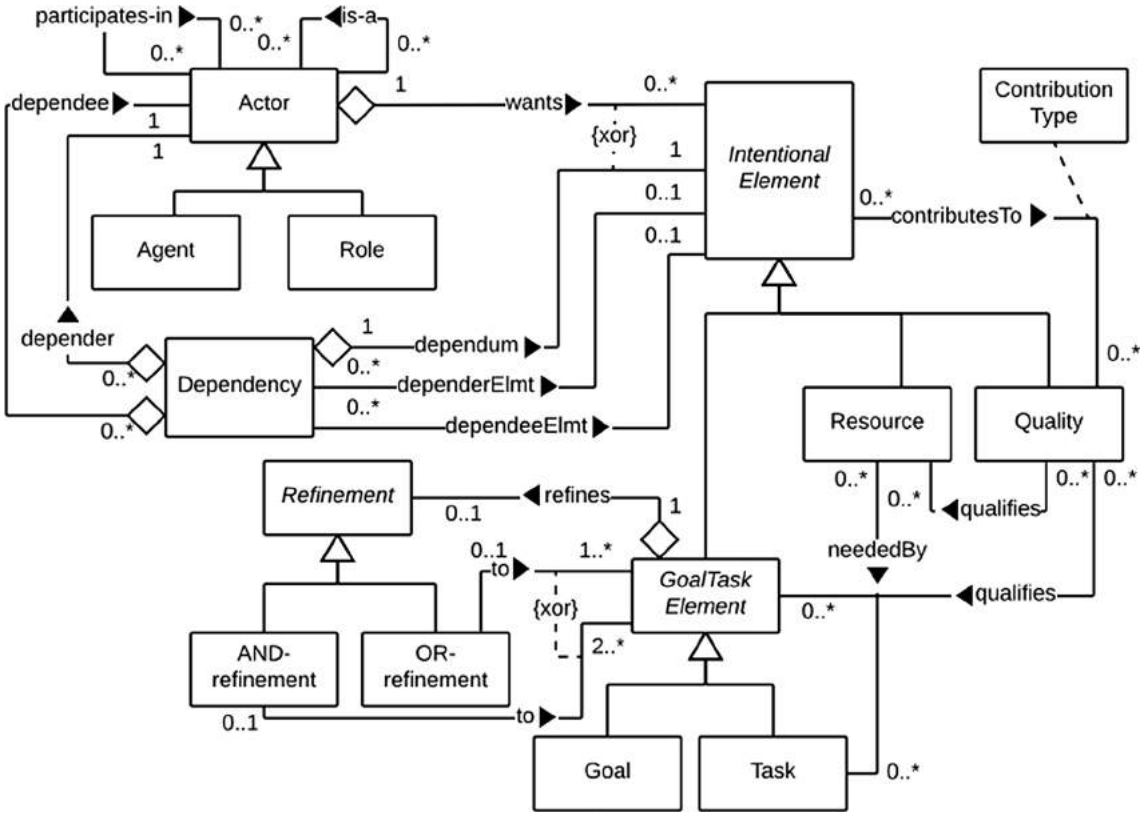


Fig. 2 Metamodel of iStar 2.0 [18]

We also classified the extensions as lightweight, heavyweight or both. The results point to seventeen extensions (17.7%) were only lightweight, thirty-seven extensions (38.5%) were only heavyweight, and forty-two extensions (43.8%) were both.

TROPOS4AS is an example of an iStar extension, it being proposed by Morandini et al. [51]. It models characteristics of adaptive systems in Tropos, proposed by Bresciani et al. [11], providing conceptual models, a graphical language and its semantics, to enable capturing requirements needed for defining and driving adaptation.

Figure 4 shows an example of a robotic room cleaner agent modelled in TROPOS4AS. We can easily identify elements that are not part of iStar default syntax such as failure, error, condition, inhibits relationship, failure relationship and goal types. We found goal types (with the values of AchieveGoal, MaintainGoal and PerformGoal) and new graphical representations for condition and failure that are not part of the original iStar.

2.3.1 Description of iStar extensions

In this subsection, we succinctly described some iStar extensions. These extensions were identified by a Systematic Literature Review detailed in Sect. 4. The purpose of this section is to introduce a variety of extensions in order to present a general overview about them, not to show a complete survey of extensions. Therefore, this overview can be useful to the reader understand the variety of domains/application areas of the extensions and the constructs related to them.

The two main iStar extensions are GRL (Goal-Oriented Requirement Language) created by Amyot et al. [5] and Tropos, proposed by Bresciani et al. [11]. GRL is an extension of iStar which models a set of satisfaction levels, new contribution types and quantitative contributions. Tropos is an agent-based methodology which extended iStar to model plans of agents. These two extensions have been a starting point for new extension proposals.

The extension proposed by Lapouchnian et al. [38] introduced the concepts related to autonomic computing of configuration, optimisation, healing and protection. Siena et al.

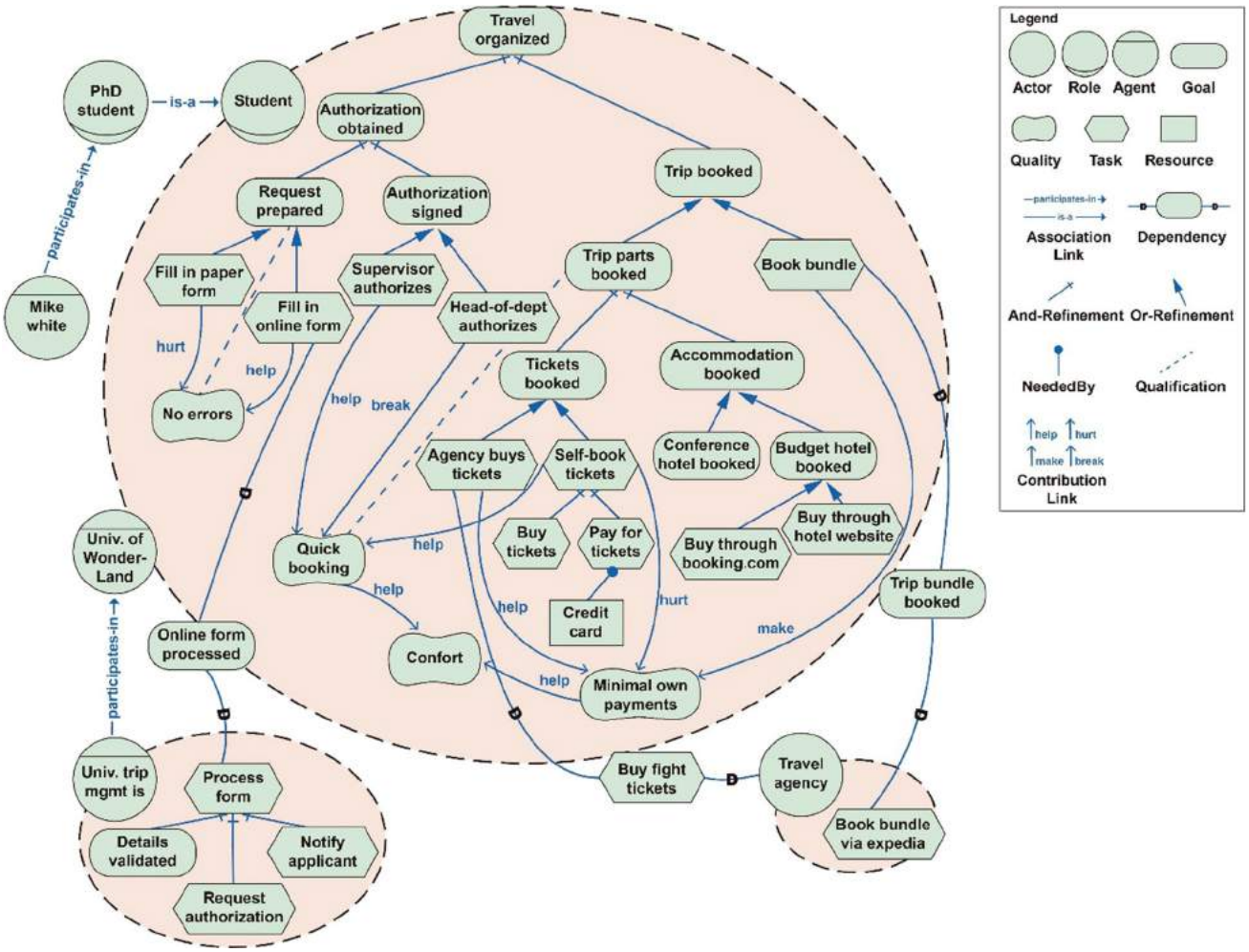


Fig. 3 A preview of the travel reimbursement scenario as captured in iStar 2.0 SR model [18]

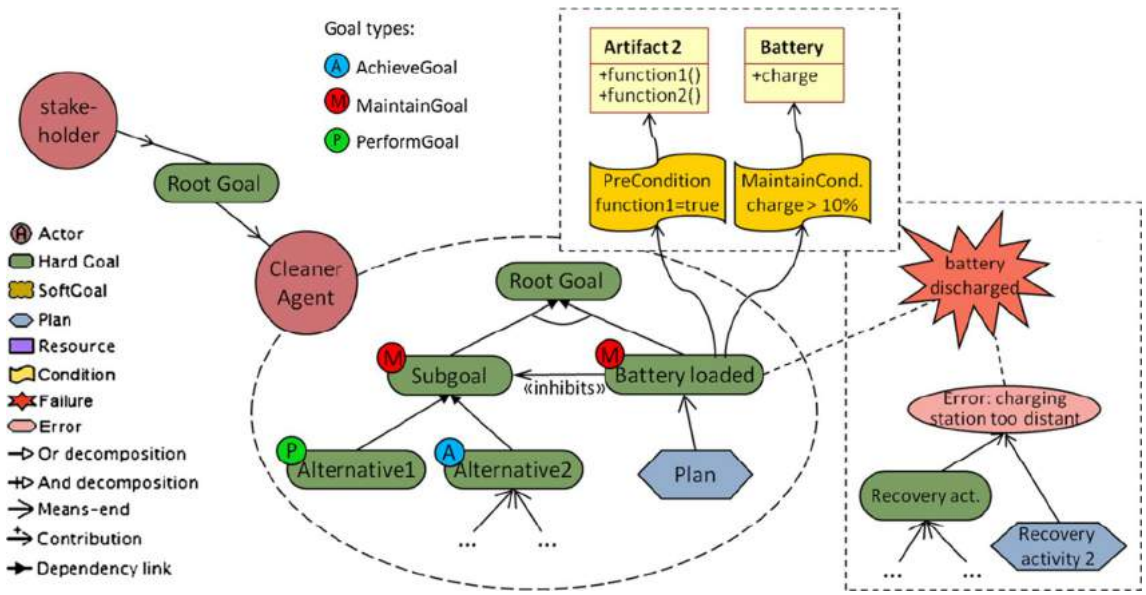


Fig. 4 Modelling of cleaner agent with TROPOS4AS [51]

[56] introduced an extension to iStar to model and analyse *norms*, then applied it to model laws and regulations applicable to European food traceability systems. It is related to the induction of new actor goals from the adoption of a *norm*, explanation of existing goals because of the imposition of *norms*, and the discovery of new roles/actors owing to the imposition of a *norm*.

Chung [14] presented an iStar extension to analyse role-based *conflicts of interest* in companies. In [45], the authors proposed an iStar extension to model data warehouse. The extension proposed by Borba and Silva [10] documents common and variable requirements in software product lines.

Liaskos and Mylopoulos [42] represented the temporal values of goal models in a lightweight and concise manner in an iStar extension, by numeric values. Another iStar extension [40] represents contextual goal models in security requirements.

Lapouchnian and Mylopoulos [39] addressed the modelling of contexts, the specification of their effects on system goals, and the analysis of goal models with contextual variability. Horkoff and Yu's [31] extension represents new graphical stereotypes and a reasoning technique[3] to introduce a backwards reasoning approach in iStar models. Preferences representation was addressed by Liaskos et al. [41] through *preference tasks* and representation of *Condition formulae*, *Condition elements* and *Effect elements* in iStar.

In [61], an iStar extension is proposed by Teruel et al. to specify requirements of collaborative systems, in which the collaboration and the awareness of other users' presence/actions are crucial.

iStar was extended by Alencar et al. [1] to deal explicitly with crosscutting concerns, as a means to address more efficiently requirements change and its impact on other requirements. Alencar et al. [2] embodied a specific notation to represent and compose aspectual iStar models, using aspect orientation to address modularity and composition of crosscutting concerns. iStar is extended to define the organisational context.

Focusing on works that extended iStar based on GRL, De Kinderen and Ma [20] introduces a goal- and value-oriented approach for purposeful language development. The work of Marosin et al. [44] shows a GRL extension to represent concepts of Enterprise Architecture as *Added value*, *Future state* and *Principle* by stereotypes associated with intentional elements. Schulz et al. [60] proposed an adaptation which is attained by the use of expansion models which cover organisational viewpoints like the skill profiles mentioned above.

[3] According to Van Lamsweerde [62], reasoning is an area studied extensively in Artificial Intelligence to generate conclusions from available knowledge. This method is used in many iStar extensions to generate a formal representation from the models.

Also, there are some iStar extensions based on Tropos. A context-aware personal agents' extension was presented by Murukannaiah and Singh [55], which introduced the concepts of context, plan and the *Context-means* relationship. The work of Asnar et al. [6] proposed a goal-modelling approach to analyse risk with iStar, in the Tropos context. The risks are analysed along with stakeholder interests, and then countermeasures are identified and introduced as part of the system's requirements. The paper of Ali et al. [4] shows an engineering approach for customising requirements models to fit their deployment environments as an essential step in a comprehensive and complete systems deployment process.

Finally, there are several iStar extensions based on Secure Tropos [52]. Software Product lines were introduced in Secure Tropos modelling in the paper of Mellado et al. [46]. The research of Islam et al. [35] represents security and privacy in Secure Tropos models. Representation of *Cloud providers* is introduced in Secure Tropos by Mouratidis et al. [53] to analyse its security and privacy characteristics. *Commitment* specifications can be used for the design and the development of applications whose interactions satisfy the security needs, so in Dalpiaz et al. [17] investigate the use of *Commitments* to model security requirements in Secure Tropos. Vulnerability modelling is addressed by Elahi et al. [21] through *Security Attacks*, *Countermeasures* and Requirements Based on *Vulnerabilities*.

Nòmos extension introduced *Law* and *Norms* reasoning elements as iStar notations. It was proposed by Ingolfo et al. across five papers [32–34, 57, 58] that introduced the notation and reasoning techniques gradually.

3 Related work

We did not find guidelines, qualitative works or surveys related to the analysis of extensions in other modelling languages. Qualitative and exploratory studies, however, have been used to investigate requirements engineering and other aspects of iStar.

Next, we present several papers related to requirements engineering studies. In [12], the authors identified 30 topics in requirements engineering elicitations based on interviews with five (5) systems engineers, and the importance of these topics was evaluated by 40 people in a quantitative survey. We followed the same steps of the authors of the paper [12], since we performed a qualitative study based on interviews and afterwards did a quantitative study using a survey to analyse the importance of the findings of the first study.

Some works are related to iStar. The variations of the use of iStar 1.0, such as the syntax of University of Toronto and others, were investigated by Horkoff et al. [30]. They surveyed 15 student assignments and 15 academic works

containing examples of iStar models. A qualitative analysis has been performed in order to understand the motivations behind the syntax variations. Yet in the educational context, two other research works [7, 9] describe the experience of iStar teaching, at University of Toronto [7], Open University and City University of London [9]. Both papers present descriptions of positive and negative findings in iStar teaching based on authors' experience.

Differently from the papers discussed in this section, our work intends to explore iStar extensions. We aim, therefore, to understand how iStar extensions are performed and what is important from the point of view of specialists in iStar extensions.

4 Research methodology

[Research Problem] Often iStar is extended to model a specific application area and/or practical situations. Nevertheless, because of the absence of a method to guide iStar extensions, the existing iStar extensions have been proposed in an ad hoc[4] fashion, which can result in problems of incompleteness, inconsistency and conflicts.

The analysis of incompleteness, inconsistency and conflicts was performed in a previous work (see results of SLR [27] for more details). Several problems related to incompleteness, inconsistency and conflicts were identified in [27]. With respect to incompleteness, there was a lack of definition of the meaning of the constructors introduced in the extensions. Moreover, some extensions were not defined in the abstract syntax (metamodel and constraints). Regarding inconsistency, several inconsistencies between the abstract syntax and concrete syntax were identified. As far as conflicts are concerned, five different types of conflicts were detected: one concept with two or more representations in concrete syntax; two or more concepts with only the same representation in concrete syntax; new constructs in conflict with the iStar default syntax; wrong representation of iStar default syntax construct; and last but not least, representation of constructs that are not part of the extension.

[Objective] The purpose of this work is to contribute to an understanding of how iStar extensions are made and to identify what can be done to help to define future extensions. It will contribute to the definition of a systematic approach to guide other iStar extensions.

[4] The term ad hoc is used throughout the paper, so we presented the meaning of this phrase according to Cambridge dictionary (https://dictionary.cambridge.org/dictionary/english/ad-hoc and https://dictionary.cambridge.org/dictionary/learner-english/ad-hoc) and Merriam-Webster dictionary (https://www.merriam-webster.com/dictionary/ad%20hoc): not regular or planned, only for a particular purpose or case without consideration of wider application.

[Study Design] We choose to use qualitative and quantitative methods together to gain a more complete understanding of our research questions (see research questions in Sect. 1). The qualitative study was performed to analyse the point of view of extenders, and the quantitative study was used to realise a triangulation of the findings and to try to confirm its importance. Thus, we used mixed methods [16] since this research is composed of a qualitative study and a quantitative study.

[Population] The universe of this research (population) consists of authors of iStar extensions, so to identify authors of iStar extensions, we performed a search in ACM, EI Compendex, IEEE Xplore, Science direct, Scopus, ISI Web of Science and Springer databases using the following search string: *("i*" OR "framework i" OR iStar OR i-star OR eye-star OR "Goal-oriented Requirement Language (GRL)" OR Tropos) AND requirements AND (goal modeling OR goal modelling OR goal-oriented) AND [(extension OR extends OR extended OR extensibility) OR (patterns OR profile)].*

We also considered all papers of the iStar repository at citeulike (http://www.citeulike.org/group/14571). A manual search was also performed in seven editions of the International iStar workshop (2008, 2010, 2011, 2013, 2014, 2015 and 2016) and in the book on Social Modelling for Requirements Engineering [65]. Note that the proceedings of the first two editions of International iStar Workshop (2001 and 2005) were not considered because they are not available. Furthermore, we did a backward and forward snowballing in the selected papers. Finally, we contacted the authors of certain papers to try to find papers not identified in the search.

The search and selection of iStar extension papers were held in the context of a previous Systematic Literature Review (SLR) [27]. As a result, we found 96 iStar extensions from 2001 to 2016 and identified 153 different authors. Thus, our universe (Population) consists of 153 authors of papers from 75 different universities describing iStar extensions.

Section 4.1 presents the methodology of our qualitative study and Sect. 4.2 shows the methodology of our survey.

4.1 Methodology of qualitative study

We chose a basic qualitative research approach conducted with a selection of the 153 researchers who have proposed iStar extensions.

[Sample of Participants] We used a non-probabilistic sample whose purpose was to choose richer cases for study [47]. The following criteria were used to define the sample of participants: (i) the number of extensions proposed, and (ii) the authorship order. We aimed at authors with few extensions (less or equal to 3), those who were first authors and authors with many extensions (more than 3) irrespective of the authorship order.

Table 2 Summarised version of the script interview

Part 1. Profile—pre-survey
What is your current occupation? How many years of experience do you have using iStar?
Part 2. Experience on iStar and Extensions
1. What is extending a modelling language?
2. How would you describe the process followed in the creation of your extensions?
3. How were new extensions' concepts chosen?
4. Considering abstract and concrete syntaxes, how these syntaxes were specified in your extensions?
5. What were the difficulties found when defining the abstract and concrete syntaxes for your iStar extensions?
6. What are the advantages of providing a modelling tool that supports the extension?
7. Please cite one iStar extension that you consider well done, and other that you consider not so good and why.
Part 3. Inconsistency Analysis
8. Given two scenarios: Scenario 1: Two extensions represent the same concept in two different graphical forms. Scenario 2: Two extensions represent two different concepts using the same graphical form. Comment on the problems described in those scenarios.
Part 4. Finalisation
9. Which actions could be done to ease the process of extending iStar?
10. Is there something about the extensions that we did not mention in the interview and you would like to talk about?

Table 3 Profile of participants

Participant identification																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Current occupation (P = Professor, S = PhD Student, PD = Postdoc, R = Requirements engineer in a company)																			
P	P	P	P	S	S	P	P, R	P	P	P	P	P	PD	P	P	P	P	P	P
Number of iStar extensions																			
2	4	2	1	1	1	1	1	7	5	1	8	4	2	5	1	4	2	6	9
Experience with iStar (in years)																			
20	15	2	20	5	3	14	4	11	13	11	11	16	5	17	2	12	20	14	23

We invited 33 authors of iStar extensions to be interviewed. Eight of these did not answer our contact or answered informing that they were not available. Consequently, 25 authors agreed to participate in our research.

We interviewed them until saturating the findings, that is, when a sequence of interviews was made and new findings no longer appeared, given that the findings of previous interviews have been repeating themselves. The saturation was achieved with 20 interviews, so we had 20 participants whose profile descriptions are available in Table 3.

[Collection Preparation] Clarification and consent terms were sent to participants before each interview session. Semi-structured interviews were conducted using an interview script with open questions. A short version is presented in Table 2, and the complete version is available in "Appendix A". The interview script was validated by the fourth and the fifth authors of this paper. It was tested previously by eight Ph.D. students in Computer Science and through a pilot interview with two experts in iStar extensions who helped to improve it. The data of the pilot interviews were not considered in the analysis.

The part 1 of the script is related to the profile of the participant. Then, we introduced a conceptual question to start the questionnaire (Question 1). The questions 2, 3, 4 and 5 are related to the RQ1 (How have the iStar extensions been developed?). Finally, the questions 6, 7, 8, 9 and 10 are related to the RQ2 (What could be done to propose better extensions?). The questions were based on the results of the SLR [27], where, for example, we needed clarifications about how the iStar extensions had been proposed (Question 2) and needed to get the opinion of the researchers about some problems that had been occurring (Question 8).

[Data Collection] Interviews were conducted in Portuguese (7), Spanish (3) and English (10), according to the participants' native languages. The interviews were conducted via Skype during September and October 2016. Twenty participants were interviewed: nine authors with many extensions (more than 3) and 11 authors with few extensions (less or equal to 3). Researchers were interviewed from 10 different countries and research groups of 19 different universities (from a total of 75 different universities identified), being nine in Europe, five in North America and six in South America. Seventeen participants were professors (one of them also mentioned being an engineer in a company), two participants were Ph.D. students and one postdoc. The profiles of the participants

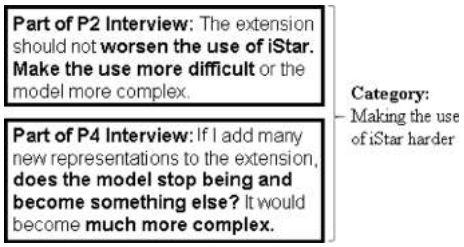


Fig. 5 Construction of a category using opened coding

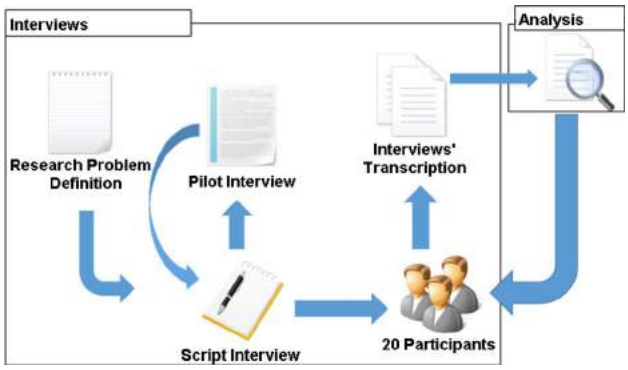


Fig. 6 Overview of interview methodology

are shown in Table 3, which presents the data about current occupation, number of extensions and experience with iStar (in years).

The mean of participants' experience is 11.72 years, with values ranging from 2 to 23 years.

Each interview was conducted by one author of this paper, while a second author took annotations. All interviews were recorded (with permission of each participant). They took on average 38 min and altogether resulted in 12 h and 40 min of audio time.

[Data Analysis and Synthesis] Qualitative data were then analysed using procedures from the Grounded Theory methodology defined by Strauss and Corbin in [59]. Grounded Theory aims at building a new local theory from collected data rather than from predefined concepts. The statements presented in [59] were used to categorise and synthesise data, to build an evidence-based theory on how iStar is extended. The audios of the interviews were transcribed, and the MAX QDA 12 (a tool used to perform qualitative analysis) was used to support the analysis.

We labelled portions of text using text codes (initially opened coding, then closed coding). Coding consists of giving a label to important portions of the interview transcriptions. The opened coding is used at the beginning of analysis to identify relevant portions of the interview transcriptions and create the codes. The closed coding is used to identify relevant portions of the interview transcriptions based on codes identified in opened coding. We started with opened coding, where several codes were identified. We then performed a closed coding step where interview transcripts were re-evaluated to try to identify codes not found in the first analysis.

These codes were then related to each other giving rise to the categories that were named according to the method of constant comparison [59]. According to [47], data are grouped in a similar dimension. The dimension is tentatively given a name; it then becomes a category.

Initially, we did an intra-participant analysis (i.e. analysis in the transcription of each participant) to create labelled portions of text using codes and an inter-participant analysis (i.e. analysis between the transcriptions of the participants) to relate these codes giving rise to the categories that were named following a constant comparison method [59]. Figure 5 illustrates the category creation process.

This step included the process of understanding how the categories identified in the coding related to each other and classifying categories under common themes, thus creating hierarchical classifications.

The related categories were grouped in core categories. A core category is the main conceptual element through which all other categories and properties are connected. In this subsection, we present the identification of categories in transcriptions.

The relationships between categories were then mapped, leading to propositions that support the main topic. Finally, we performed retrospective interviews to clarify the information identified in the data analysis.

[Credibility, Consistency and Transfer of Results] The information collected was analysed by the authors of this paper during meetings and the identified inconsistencies discussed. When necessary, further explanations were required from the participants. Figure 6 shows an overview of the methodology followed in our qualitative research.

Additionally, we applied a survey to evaluate the findings of the qualitative study. More details about the methodology of the survey are given in next section.

4.2 Methodology of the survey

We followed the principles of the Survey Research proposed in [37]. This survey is cross-sectional. The application takes place through a self-administered questionnaire via the internet, since the participants were in different countries.

The main goal of this survey was to evaluate the relevance of the findings of the previous interviews. Our research question is this: *Are the statements about iStar extensions extracted from the interviews important for the iStar researchers that extended the language?*

[Sample of Participants] The sample is composed of the 133 researchers in iStar extensions who did not participate in interviews.

[Collection Preparation] The survey is composed of quantitative questions to take personal data (experience with iStar, country and gender) and the evaluation of eighteen statements derived from the identified results in the qualitative study. Clarification and consent terms were sent to participants with the invitation to participate in this research.

The options of the questions were defined in a Likert scale with these values: Totally Irrelevant (1), Unimportant (2), Neutral (3), Important (4) and Very Important (5). We validated the survey testing it with eight Ph.D. students in Computer Science. We also applied a pilot with five researchers with experience in iStar extensions. We made the changes suggested by participants of the test and the pilot. Their suggestions were not used in the data analysis.

[Data Collect] We applied the survey to respondents between 18th November and 2nd December 2016. We invited 133 authors of iStar extensions who did not participate in the previous interviews.

We had responses from 30 of them (22.5%). The average of respondents' experience with iStar is 8.6 years, with experience ranging from one year to 23 years. The participants were from Spain (5), Mexico (5), Italy (3), Belgium (2), Brazil (4), Chile (3), Germany (1), USA (1), Colombia (1), Portugal (1), North Korea (2), Morocco (1) and Luxembourg (1). We received responses from twenty-two (22) men and eight (8) women.

[Data Analysis] We calculated the mean, mode and standard deviation of the responses. We also performed the hypotheses tests to analyse whether each statement could be considered statistically important. Our general hypothesis is that the statements are important to the iStar extension researchers. This general hypothesis should be applied to each statement (S1–S18) producing eighteen null hypotheses, one for each statement. The description of the statements is presented in Sect. 6.1.

Finally, we extracted a set of guidelines from the statements which can help in the proposal of future extensions.

5 Results and discussion of the qualitative study

In our study, we explored categories that may influence the development of iStar extensions. This section reports the results and discussions, being organised to present the conceptual understanding of the participants about what is necessary to extend a modelling language. Then, it describes how iStar extensions have been made, and depicts the main categories identified in the analysis together with related categories to each other. It also explains how those main categories are related and how categories are affected (positive and/or negative). Finally, we present the participants' views on how to improve the iStar extension process. As said before, this analysis was performed based on Grounded Theory [59].

In this section, we presented parts of the transcriptions of the interviewees in italic, between double quotation marks and with larger indentation.

It was not possible to comment on a comparison between our results and results of similar studies for other modelling languages because, despite searching, we did not find similar studies.

5.1 The viewpoint of the participants on extending a modelling language

Initially, we addressed interviewee's understanding of what it means to extend a modelling language. This is covered in the interview script by question 1.

Recall that to extend a modelling language means to introduce new constructs or modify old ones (see Sect. 2.1 for a complete description). We started the interview asking them about this concept.

In general, the participants had no difficulties to describe what it is to extend a modelling language. Their answers were succinct and similar. In the following, we show some excerpts to illustrate their opinion:

> It is adding new elements in the language, elements that do not exist yet. This addition can be through an extension of existing concepts or add a purely new concept. (P2).

> It is a way to adapt or improve the model to be able to represent aspects that previously could not be represented. (P5).

> It proposed new elements, new figures, new graphics that helps to add new meanings to be represented with the meanings that already exist. (P8).

Despite this understanding, two researchers reported having not developed extensions, even when papers of their authorship had included representations that were not present in the original version of iStar. When asked, they confirmed that they had proposed a new construct with new representation, but this did not represent an extension.

Next, we show a dialogue that illustrates this scenario. P14 cited Tropos [1], a methodology for the development of multi-agent systems that use iStar as the basis for their representations. Below, we have used the labels C1, C2, …, Cn as identifiers to the mentioned concepts to maintain the anonymity of participants.

(P14): With this paper, no I did not propose an extension to Tropos. I worked only on methodological issues. [Interviewer]: Have you added a graphical representation of the C1 concept that did not exist in Tropos/iStar.? (P14): Yes, I added a new representation of this concept.

We believe that the reason for them reporting that they did not perform an extension was that the participants intended to say that they only had partially carried out the extension.

5.2 How iStar extensions have been proposed

We analysed how the extensions were developed by the participants through the questions Q2–Q5. We did not identify a standard way to extend iStar. The steps followed by the participants had many differences, and each researcher described a distinct set of activities. Maybe this reflects the comment of P7 mentioning that "*this is one of the serious problems of iStar in general, things are being extended in an* ad hoc *way*". We show two passages that illustrate the differences among the ways of proposing iStar extensions:

> First iStar was compared with KAOS, then we detected, through an experiment, the possible weakness of iStar. We found that anyone had already proposed some extensions accordingly. And we proved that there was no extension. So, we continue creating the new syntax, always trying to keep the original philosophy of the language. We create new symbols. (P8).

> Firstly, a literature review was done because there are goal-oriented models. So, a study was done to identify what they were. Then, to delve into each of these models, for example, iStar and Tropos, which was what we worked with, I tried to understand more broadly the elements of the model with, the formal definition of each of these elements and how they are used. So, I identified a gap and proposed an extension to include the representation of behaviour. (P6).

Next, we present an overview of interesting aspects of the activities mentioned to extend iStar, collected from the interviews. We begin by addressing the need to extend it, based on the type of extension that is being proposed, whether for an application area or not. These are examples of concepts proposed for an application area: Crosscut [1], Norms [32] and Failure [51].

> To extend, we analyse the concepts of iStar, and realised that to represent the concepts we wanted, the existing ones were not enough. (P1).

When the extension is proposed to improve some practical aspect of the language, the need emerges when a limitation is detected during the usage of iStar. These are examples of concepts proposed to improve practical aspects of the language: Module [45], Cardinality [10] and timing information about a task such as min and max duration, date of completion [51].

> In fact, it was something coming from the real world to the model. We had difficulty to represent the concept C2. So, we created the model and identified that it was a problem of the model itself. (P5).

> Basically, we selected from both our own experience and our needs. So first we updated the most transversal constructs, such as C3 concept extensions that are not dependent on a domain. (P19).

The first step of extending iStar as introduced by P12 is this:

> I guess the process typically starts with some problem, some kind of weakness in the baseline of the language so there is a problem in the modelling domain. There is something you cannot represent with existing modelling languages. (P12).

Therefore, there is no single starting point in the case of extensions to a specific application area. Another possibility is the need to start from the group's research interests in which the participant belongs, as illustrated from the excerpt of P10's interview:

> I am looking at what I need to do my reasoning, and then I go and add them. If I cannot represent it, I go and add them in my diagrammatic notation. (P10).

Seven participants mentioned a literature review, systematic in three cases. For them, that is an important step to check if there is any extension already proposed for that domain/application area and to identify the domain/application area concepts which are included in iStar. In some cases, iStar is used to model a system of the domain that is intended to extend, to make sure of the needs and that iStar is suitable for modelling that domain.

P8 and P12 cited that the modelling language should be chosen after the definition of the need for the extension. P12 mentions that:

> So, I don't take for granted that I have to extend one specific language, but I choose which one to extend if none of them satisfies my needs. (P12).

Four participants (P1, P6, P9 and P16) acknowledged the need to provide a complete description of the meaning of constructs to be included by the extension. An example is presented below:

> Several problems were identified later because even though we were trying to preserve the semantics, there were new elements that, from the point of view of the

> original language, were not seen. So, it was really necessary to describe them better. (P1).

Comments such as the one from P1 reiterate that the concepts' presentation is something that should be given more attention in future extensions.

Typically, when the concepts to be introduced by the extension are identified, they are represented in the abstract and concrete syntaxes of iStar (see Sect. 2.1). In this context, two participants said that sometimes they apply the extensions to the concrete syntax and later represent them in the abstract syntax.

Eight participants reported that they included these concepts only at the level of concrete syntax. When directly asked about the abstract syntax definition, they mentioned that it is very important to an extension.

> In some extensions, I have relied on the abstract and concrete syntaxes, but I believe I base myself more on the concrete one because that is the one that I can better understand. (P13).

> The extension was simple, so I worked only on the concrete syntax, but we should work with both depending on the goals of the research. (P17).

> Well, I would not bother with a metamodel. I would think the basic schema of iStar is quite sufficient, no real need to render it in yet another modelling formalism like a class diagram. So, what is done is the concrete syntax for additions in other words. (P18).

> Having a metamodel would help clarify and sort out any potential ambiguities. (P20).

> There was the fact that the metamodels by themselves are not sufficient, so we needed to have constraints, so the standard processes them as national language constraints, but a lot of them had been formalised in OCL in the tool. (P15).

P12 commented about the importance of maintaining the high level of modelling in iStar extensions:

> So sometimes you have a concept that doesn't fit at all with the modelling language you take as a baseline. So, if you have, for example, a high-level modelling language like iStar it doesn't make sense to consider very low levels of detail such as…use for encryption if you talk about security. (P12).

In this sense, during the interviews, we identified three profiles of participants: (i) participants who began their research with extensions performing only extensions in the concrete syntax and some years later they changed to extend both; (ii) participants who began their research with extensions only doing them in concrete syntax and even after maturing this type of research continued performing some extensions without considering abstract syntax; and (iii) participants who performed their extensions always considering both syntaxes.

Participants P1 and P7 have also mentioned that sometimes researchers are more concerned about showing how the extensions should be used (through statements) so that the details of the changes made in the modelling language are somewhat neglected in research.

The need for a more careful choice of graphical representations of the constructs was highlighted by participants P1, P2, P4, P7, P8, P9, P10, P12, P15, P17, P19 and P20. Next, we present two reports.

> We are not user-driven in our selection of symbols. So, things like the physics of notation and other frameworks essentially try to raise awareness about this problem. (P15).

> It is true that [we need] to define the concrete syntax, the more satisfactory as possible. We need concepts that modelling groups do not have and we need search how the graphical representation should be. Our group never made it, we know it's important to do, but we never did. (P19).

Thus, the quotes presented above are evidence of problems related to the principle of semiotic clarity presented in [49] and emphasised the importance to better address this problem. Maintaining consistency between the representations of the metamodel and the concrete syntax is considered important by participants. Nonetheless, all participants said that they do not perform this check or do it informally.

> We had a concern, but it was not a formal concern, such as taking a test if everything in the concrete syntax is in the abstract syntax. (P2).

> There was a concern, yes. And it's important, otherwise you generate confusion for the modeller, because if you create something in the abstract syntax that it will not be found in the concrete syntax and vice versa, then it is unclear. (P7).

5.3 Identifying and relating categories in iStar extensions

After the analysis of the participants' views on extensions, their opinions about what is the meaning of a modelling language extension as well as what the description of their iStar extensions is, the next step is to identify and relate the categories that affect the proposal of iStar extensions. In this section, we present the identification of categories, showing part of the transcriptions that motivated them, as well as how they are related (Sect. 5.3.1).

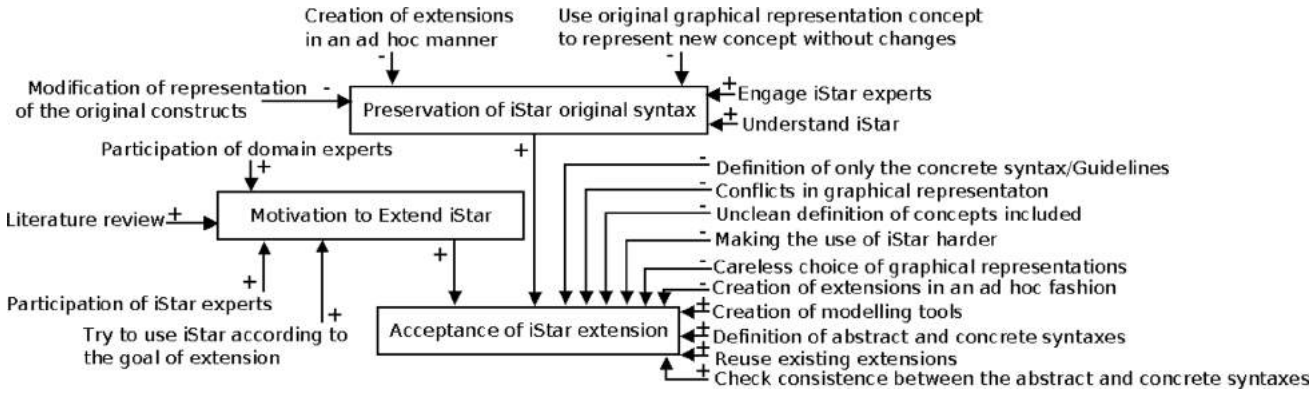


Fig. 7 Categories and their relationships

The related categories were grouped in three core categories (*Preservation of iStar original syntax*, *Motivation to extend iStar*, and *Understanding/acceptance of the iStar extension*). Below, we present the three core categories and their related categories (see Fig. 7). We also include a sample of transcriptions that originated the related categories.

Preservation of iStar original syntax is a core category in iStar extensions. It means that the iStar original constructs of iStar are maintained in the extension, that is, the iStar extension is conservative. It is related to some positive (+) or negative (−) categories such as creation of extensions in an ad hoc fashion (−); modification of the representation of the original constructs (−); use of original graphical representation concept to represent new concept without changes (+); understanding iStar (+) and engaging iStar experts (+). Some excerpts presented below are related to the preservation of the iStar original syntax.

> This is one of the serious problems of iStar in general; it is because things are being extended in an ad hoc way. (P7)—Creation of extensions in an ad hoc fashion.

> With the core of the language, we agree that by extending iStar we should not contradict what is in the core. So, I do not consider a good extension one that changes the behaviour of the standard language elements. (P19)—Modification of the representation of the original constructs.

According to P12, this kind of modification may be unclear among the community. P10, however, seems to follow the opposite view:

> [I am] not worried to be compliant with the language… I would take the freedom to violate the conventions in order to achieve my goals for research. (P10).

This discrepancy deserves to be highlighted, and it is an important characteristic to be considered.

P17 comments about the importance of a tool to verify if the extension is compliant with iStar. According to P17, it is good for some researchers, but to others who want to have more freedom and do not care about that compliance, it may not be that useful.

> In our works, we tried to use iStar to model systems of the application area without modifying iStar. In some cases, we needed to extend it. (P19)—Use of original concept graphical representation to represent new concept without changes.

> Today, what I recommend is to think whether what you want to represent cannot be represented with the language the way it is currently. (P2)—Understand iStar.

Five participants mentioned the need to involve the iStar experts. An excerpt related to this category is presented in next category.

Motivation to Extend iStar It represents actions which can help to identify the need to extend iStar. Some related categories that may contribute to this core category are the following: participation of domain experts (+); participation of iStar experts (+); literature review (+); and try to use iStar according to the goal of the extension (+).

The following excerpt motivated the creation of this central category:

> So, I keep saying even to collaborators, as we are using the mindset of goals, not a specific language to extend, so if the concept is not novel by itself, then the contribution to knowledge is extremely small if your aim is to extend iStar… I think [we must] ask the question whether the extension is needed in the first place. (P9).

In the following, we present several quotations related to these factors.

> The need came mainly from the background of my advisor who has been working for a long time in this

extension area. (P6)—Participation of domain experts and iStar experts.

It was performed a non-systematic literature review. (P7)—the Literature review.

It was from the modelling of a scenario with iStar, we modelled a scenario and identify the need to extend. (P5)—try to use iStar according to the goal of extension.

Acceptance of the iStar extension represents the acceptance of the extension by the iStar community. It is a core category related to the following subcategories: creation of extensions in an ad hoc fashion (−); unclear definition of the concepts included (−); definition of only the concrete syntax/statements of use (−); definition of abstract and concrete syntaxes (+); checking consistency between the abstract and concrete syntaxes (+); careless choice of graphical representations (−); making the use of iStar harder (−); conflicts in the graphical representation (−); creation of modelling tools (+); reuse of existing extensions (+).

The following excerpt motivated the creation of this core category:

Any extension that resulted in a paper that never was followed is obviously a bad extension because it never led to the evolution of the language itself or acceptance by the community. (P15).

The fact that extensions are made in an ad hoc fashion has been treated in the first core category. However, this factor also affects this category too. Below, we present excerpts related to other factors of this main category.

An extension that I do not consider very good is an extension that is not based on a metamodel. (P19)—Definition of only the concrete syntax/statements of use.

We treated this, but it was not a formal concern, kind of taking a test if everything in the concrete syntax was in the abstract syntax… It was a concern only when seeing both… (P2)—Check consistency between the abstract and concrete syntaxes.

Concerning the concrete syntax, the biggest difficulty is when you need to add new symbols to represent the concepts… determine which the symbols are. Usually, we have done this based on labels, but this is an aspect that should be improved in the future. (P19)—Careless choice of graphical representations.

If you have a modelling tool people can use your tool in your extension. If you don't have a tool, probably people would forget about your extension or not use it. Second, you can ensure syntactic well-formedness. Syntactic correctness can be already ensured by the tool. Further, you can analyse your models automatically so can add automatic reasoning on the models. And fourth, as I was saying earlier you can actually decide to hide some of the extensions like the attributes and use the properties to show those so that I can observe a less cluttered kind of image. (P12)—Creation of modelling tools.

Well, it was based on another extension to modelling concept C4. (P5)—Reuse existing extensions.

You cannot worsen the use of iStar. Making the use more difficult or the model more complex. (P2)—Making the use of iStar harder.

A good extension must not conflict with other existing extensions. (P4)—Conflicts in the graphical representation.

Regarding conflicts in the graphical representation, P14 mentioned that he made changes in the representation of one of the concepts. It can be identified in the dialogue below:

[Interviewer] Did you introduce a different graphical representation to represent concept C5? (P14) Yes, in this case, we introduced a new notation for a concept already existing, but we did not introduce a new concept. We just changed the way of representing (represent differently with another figure).

5.3.1 Relating categories and its impacts in iStar extensions

The model in Fig. 7 shows the concepts presented in Sect. 5.3, showing them as categories and how they impact on iStar extensions, either positively or negatively.

The non-conservative extensions are related to the impact on the syntax/semantics of the original constructs of the language. Therefore, kind of extension happens when an extension removes part of the original iStar constructs in its specification.

Based on analysis of the participants' interviews, we identified that owing to the iStar extensions being performed in an ad hoc manner, there is a greater possibility that the extension is non-conservative, that is, does not preserve the iStar standard constructs. Furthermore, we identified in transcriptions of interviews of P7, P10, P12 and P13 that the fact that the extensions are carried out in an ad hoc fashion has a negative effect on the preservation of the original syntax, that is, in the definition of conservative extensions. For example, the participant P10 said, *I chose an* ad hoc *way… So, I was not worried to be compliant with the language… I would take the freedom do violate the language conventions in other to achieve my goals for research.*

iStar has some scalability issues, which have been focused on in various studies, such as the Systematic Literature Review presented in [43]. Our work discovered that researchers believe that the representation of many new constructs can impact scalability, making the use of iStar more difficult and consequently mischaracterising the original representation.

Consequently, to understand the language and to try to use iStar according to the extension goal before performing the extension may help preserve the original definition of iStar. Also, the participation of iStar experts in proposing a new extension helps the preservation of the original definition of iStar. Extensions must be proposed only when necessary, that is, for a specific application area or to address a deficiency of the language itself. Thus, a literature review (systematic or not) with the participation of domain experts is fundamental to understanding the targeted application area of the extension and to defining whether it is necessary. At the same time, the participation of iStar specialists in the extensions' work and using original iStar to model many projects can be useful for identifying the need for extensions.

Extension acceptance is related to its understandability by the users. Therefore, unclearly describing the concepts included may hinder the understanding of designers who want to use the language. Similarly, only defining the concrete syntax and usage statements can bring a deficiency in understanding because of the lack of the abstract syntax representation, which defines the new constructs and their relationships with the new concepts. Consequently, the definition of both, abstract and concrete syntaxes, can impact positively on the understanding and acceptance of iStar extensions.

In general, the graphical representations of the new constructs are made arbitrarily, and a careful choice is seen as key by six participants. Two participants cited Daniel Moody's work [50] pointing out a possible way to treat these representations.

The reuse of existing extensions as the baseline for new extensions is stated by five participants as a way of reusing the constructs already proposed and it avoids redundancies such as a concept represented in two different ways. Nevertheless, it is still a little-used approach.

Before introducing new concepts in iStar, it is necessary to evaluate the need for proposing this extension, since a useful and well-performed extension is likely to be well accepted by iStar community. It is also important to highlight the need for preserving the original definition of iStar. If the proposed extension degrades the original definition of iStar this will question the need for this extension and will compromise its acceptance. And consequently, for the extension to be well accepted, it is important that it must justify how useful for a domain or iStar community it is, and preserve the original iStar syntax.

5.4 Using results to improve iStar extensions

In this section, we present the findings related to improving the way of extending iStar. It summarises the responses to the Question 9 of the script interview (*Which actions could be done to ease the process of extending iStar?)* These recommendations have not been tested in practice, so they only define strategies derived from the results of this research and provide inputs for the next steps towards treating the extensibility of iStar.

Scalability is a recurring concern in the iStar community. The limitation on scaling iStar is identified as one of its largest barriers to industrial adoption [23]. If iStar could provide mechanisms to allow a high-quality modelling of broad and complex cases, there would be a greater adoption of iStar. Therefore, the iStar framework requires solutions and means to address its scalability [23]. P12 said, however, that there should have been a balance in scalability to define extensions:

> Deciding what are the essential elements to include without becoming too trivial but at the same time making sure that the extension really captures the domain that you want to introduce. (P12).

P12 also commented about an alternative that proposes new graphical representation in iStar. Part of the extensions are not visible in the notation itself but could be visible with a modelling tool; therefore, if you have an editor, then you can use the properties of your iStar models as an alternative to propose new different graphical syntax.

The opinion of six participants is that to add another graphical representation may further accentuate the scalability problem in iStar. Also, the participants pointed out that adding many new constructs mischaracterises the original iStar, thus failing to meet the purpose of the language. On the other hand, some extensions are proposed to improve the pragmatic aspects of the language, and some of them have been proposed to improve scalability.

Therefore, new graphical representations should be proposed to represent new first-order constructs, which should be at the same abstraction level of *goal*, *quality*, *resource*, *task*, *actor* and iStar relationships. Extensions to iStar should avoid making iStar harder to use. Reuse of existing representations including the default syntax or other extensions should be encouraged. P12 emphasised that the extensions should always have very clear semantics.

To facilitate the reuse of existing extensions, six participants suggested creating a repository containing the extensions already known. Another suggestion made by P7 and P17 was the definition of a process, and a set of guidelines to systematise the way extensions are proposed. They also drew attention to relevant issues in the former extensions that have been neglected such as applying the extension to

the abstract and concrete syntaxes and checking consistency between them.

Three participants (P2, P4 and P7) commented that it is necessary to better define the relationship between new concepts and iStar existing concepts, to describe how these concepts are related. P20 commented about having to know and to provide some guidance on the naming of the elements proposed in his extension.

Non-conservative extensions, which remove iStar constructs or change the relationship between existing constructs, are discouraged by five participants.

Another result from the interviews was the need for a more systematic and rigorous way to propose graphical representations of extensions. A way indicated by the participants was followed in [49], which is based on experiments to define graphical representations of new constructs. Besides, P20 cited that it would be interesting to explain the choice of each new graphical representation introduced. This participant also said that it should be simple enough for people to hand draw and do it on the whiteboard and scrap it off and make changes to it very quickly. P12 also commented about it and mentioned that some aspects of extensions are not so evident when the modelling is performed by using paper. Furthermore, P20 suggested an analysis to reduce the risk of conflicts in visual notation.

The need to provide extensions more clearly, with a description of new concepts, is also determining. One way to validate the understanding of the description of the extensions is performing experiments where participants analyse descriptions and then discuss their understanding of such descriptions.

Furthermore, it is interesting that the iStar extensions need to be mature before their publications. This maturity is required due to the possibility of changes in preliminary results, generating inconsistencies in its definition.

Five participants commented about an important topic of current research, the standardisation. P7, P12, P15, P19 and P20 suggested that iStar should have well-defined extension mechanisms.

P20 stated that if one has many application areas, which are specialised, then he believes that one should think about extension mechanisms when designing the core. This participant mentioned one possible way as something such as stereotypes in UML (Unified Modelling Language). P19 also commented about the importance of incorporating extension mechanisms into the core similarly to UML profiles.

P7 said that iStar could have a richer textual representation. Therefore, one way to improve the textual representation of language terms is adding lightweight extension mechanisms, which are those represented via textual notation. That would facilitate the proposal of lightweight extensions to iStar and would allow the creation of tools that have these mechanisms present. P15 states that it would

Fig. 8 Main suggestions to help the iStar extensions proposal

be interesting to propose extension mechanisms aligned to a possible standardisation by an organisation such as the International Telecommunications Union ITU. The final contribution is related to a deeper reflection from P17:

> Is it good or bad to make it easier to extend iStar? It can be both. If you extend, you can add expressiveness to the language, but you can also add confusion with many extensions overlapping. (P17).

This passage points to the dilemma of extending or not iStar. There is a trend that iStar will continue to be extended. We believe that this is not an easy question to answer, but it is necessary to raise the level of systematisation, guided by well-defined extension mechanisms and processes.

Finally, we summarise the main suggestions to help the iStar extensions proposal of this section in Fig. 8.

5.5 Relating the qualitative study with SLR about iStar extensions

As it was mentioned in Sect. 4 (Research Methodology), a SLR was performed to identify the iStar extensions and analyse them. The results of the SLR can be found in [27]. In this subsection, we relate the results of the SLR with the results of the qualitative research presented in the previous subsections of Sect. 5.

Twelve participants mentioned that they proposed extensions considering only the concrete syntax. This information is consistent with the data found in the SLR. During the execution of the SLR, it was identified that part of the extensions was defined in the context of the presentation of a method or statement. Participants P1 and P7 mentioned that iStar researchers are more concerned with describing how to use their own proposed extensions. In the SLR, we also identified the occurrence of problems in the extensions due to the original constructs (shown in Sect. 4). This fact was also mentioned in the interviews as being a problem.

The participants also suggested the creation of a repository to facilitate the reuse of existing extensions. We created an online repository for this purpose. The extensions

found by the SLR are available there. This repository can be accessed in http://istarextensions.cin.ufpe.br/catalogue/.

The conflicts in graphical representation were mentioned in the last paragraph of Sect. 5.3. In the SLR of iStar extensions, seven instances of extensions were identified that changed the original graphical representations of iStar. The dialogue presented in the last paragraph of Sect. 5.3 provided further evidence that this kind of problem has occurred.

6 Results and discussion of the survey

After the analysis of the qualitative study, we submitted an online survey, with the findings of the interviews, to analyse if the extenders agree with the findings. The methodology followed in the survey was presented in Sect. 4.2.

In this section, we describe the identification and description of the statements (Sect. 6.1) and results of the survey (Sect. 6.2). Finally, we present a set of guidelines in Sect. 6.3.

We could not compare the data of our results and data of similar studies for other modelling languages despite searching and not finding similar studies.

6.1 Identification and description of statements

We represented the findings of the interviews as statements, which were identified from the categories described in Sect. 5.3 and from the suggestions to improve iStar extensions given by the interviewees (Sect. 5.4).

We wrote the categories as statements such as the category *Reuse existing extensions* which was represented as the statement *S15—Reusing other existing extensions to improve the understanding and acceptance of new extensions*. We needed to change the categories with negative impact to represent it positively; for example, the category *Careless choice of graphical representations* (−) was represented as *S13—Performing a careful choice of graphical representations*.

Another set of the statements was inspired by the results presented in Sect. 5.4 related to improving the iStar extensions. An example, regarding *S9—Relating concepts introduced by the extensions with the iStar concepts*, was identified in the eighth paragraph of Sect. 5.4.

Therefore, the statements S1, S2, S3, S4, S5, S6, S7, S8, S10, S11, S13, S15 and S16 were identified from categories of Fig. 7. The statements S9, S12, S14, S17 and S18 are identified from Sect. 5.4.

The statements measured in the survey are listed below:

- S1—Preservation of iStar original syntax;
- S2—Dealing with the negative impact of extensions that are carried out in an ad hoc fashion;
- S3—Literature review, participation of domain experts and iStar experts and use of iStar to model systems of application area before extending it;
- S4—Understanding and acceptance of iStar extensions;
- S5—Dealing with the negative impact of proposing extensions with an unclear definition of the concepts;
- S6—Dealing with the negative impact of defining only concrete syntax;
- S7—Proposing concrete and abstract syntaxes;
- S8—Checking consistency between abstract and concrete syntaxes;
- S9—Relating concepts introduced by the extensions with the iStar concepts;
- S10—Proposing extensions with a smallest possible number of modifications and new representations;
- S11—Proposal of simple graphical representations, able to be drawn on the paper without a tool;
- S12—Proposing new graphical representation only to represent constructs in same abstraction level of intentional elements, actors and iStar relationships;
- S13—Performing a careful choice of graphical representations;
- S14—Dealing with the negative impact of conflicts and redundancies in the graphical representation;
- S15—Reusing other existing extensions to improve the understanding and acceptance of new extensions;
- S16—An iStar extension should not complicate the usage of iStar;
- S17—Proposing a process or a methodology to guide the iStar extensions;
- S18—Defining extension mechanisms to iStar.

Next, we describe each statement in detail as follows. The survey only had the statements, and the longer descriptions were added only in the manuscript for clarifying them. The *Preservation of iStar original syntax* (S1) suggested that all nodes and links of the default iStar syntax should be maintained in the extension. In other words, the non-conservative extensions, which remove iStar default constructs, are discouraged. Therefore, eliminating the representation of quality, for example, is not considered a good practice.

Once there is no process to conduct iStar extensions, they have been proposed in an ad hoc fashion. This fact has a negative impact on the proposal and many problems of inconsistencies, incompleteness and conflicts have been occurring (see results of SLR [27]). Thus, while a process is not proposed, the extenders of iStar should consider *Dealing with the negative impact of extensions that are carried out in an* ad hoc *fashion* (S2) in order to avoid inconsistency.

A set of actions were pointed out as a way of helping the definition of an iStar extension. *Literature review, participation of domain experts* and iStar *experts and use of iStar*

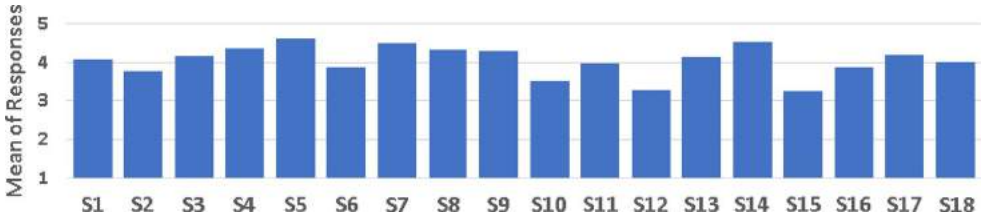


Fig. 9 Results of mean of evaluation of interviews' findings

to model systems of application area (S3) were pointed to better understand the target of the extension.

The statement *Understanding and acceptance of iStar extensions* (S4) is related to the acceptance of the iStar extension. Thereby, an unclear iStar extension would not be well accepted by the Requirements Engineering community.

Dealing with the negative impact of proposing extensions with an unclear definition of the concepts (S5) is concerned with the clear presentation of all concepts introduced by the iStar extension. Concepts which are not described cannot be understood by those who intend to use the extension.

The metamodel and well-defined rules are important to a modelling language and their extensions, once they represent what is allowed and what is forbidden in its usage. Thus, the statements S6 (*Dealing with the negative impact of defining only concrete syntax*) and S7 (*Proposing concrete and abstract syntaxes*) are concerned with this issue.

It is important to consider the *Checking consistency between abstract and concrete syntaxes* (S8), once the syntaxes can be inconsistent. The inconsistency can be described by the representation of a new construct in only one of them, for example including a new graphical representation in concrete syntax, without adding a new metaclass to represent it in the extension metamodel.

The new concepts introduced by the extension can be related to the iStar concepts, specialising the iStar constructs or be connected by iStar relationships or other relationships. Therefore, the statement **S9** (*Relating concepts introduced by the extensions with the iStar concepts*) is related to this task.

The extension should minimise the number of modifications and new representations in iStar as much as possible (S10—*Proposing extensions with a smallest possible number of modifications and new representations*).

The iStar constructs are represented in concrete syntax by simple symbols, such as circle, rectangle and line with an arrow. Thereby, the new constructs should consider a *Proposal of simple graphical representations, able to be drawn on the paper without a tool* (S11).

iStar has a set of high-level constructs, such as goal, resource, task, actor and qualification link. Thus, the extension should also consider including concepts of high abstraction level. Constructs to represent functions or classes of programming are not suitable. Sometimes, a new symbol is added to denote concepts which can be represented by a specialisation of an existing iStar construct. For example, a security goal can be represented by a stereotype added to the iStar goal. (S12—*Proposing new graphical representation only to represent constructs in same abstraction level of intentional elements, actors and iStar relationships*).

The experts participating in the qualitative study suggested that the choice of graphical representations could be proposed in a careful way (S13). The participants consider the work of Moody et al. [49] a good way to do this.

Conflicts and redundancies (see results of SLR [27]) have a negative impact on the iStar extension proposal. Thus, while a process is not proposed, the extenders of iStar should consider *Dealing with the negative impact of conflicts and redundancies in the graphical representation* (S14).

Many iStar extensions such as Secure Tropos have been well accepted by the iStar community. Therefore, when new iStar extensions reuse existing extensions, this can improve its understanding and acceptance (S15).

An iStar extension, which introduces many constructs to iStar, can weaken the iStar language, ending up looking like a new language. In these cases, it can be chaotic to use the iStar extension. In other words, *An iStar extension should not complicate the usage of iStar* (S16).

As mentioned before, in general, the iStar extensions are proposed in an ad hoc way, and because of this some problems have been occurring in iStar extensions. Therefore, the statement S17 (*Proposing a process or a methodology to guide the iStar extensions*) is related to the proposal of a process or methodology to conduct new extensions.

The participants of the interviews suggested that it will be interesting to define extension mechanisms to iStar, such as the extension mechanisms present in UML (S18).

6.2 Results of the survey

The results reiterated the importance of interviews' findings, once all statements obtained the mean of responses between 3 (Neutral) and 5 (Very Important). We highlight that nine out of the eighteen statements obtained the mean of responses between 4 (Important) and 5 (Very Important).

In Fig. 9, we present the mean of the survey responses for each statement. The descriptive statistics results (mean, median, mode and standard deviation) of the evaluation survey are shown in Table 4.

Table 4 Descriptive statistics of the evaluation survey

Statement	Mean	Median	Mode	Standard deviation
S1—Preservation of iStar original syntax	4,066666	4	5	1,01483252
S2—Dealing with the negative impact of extensions that are carried out in an ad hoc fashion	3,766666	4	4	0,97143098
S3—Literature review, participation of domain experts and iStar experts and use of iStar to model systems of application area before extending it	4,166666	4	5	1,01991660
S4—Understanding and acceptance of iStar extensions	4,366666	4	4	0,6149479
S5—Dealing with the negative impact of proposing extensions with an unclear definition of the concepts	4,633333	5	5	0,80871687
S6—Dealing with the negative impact of defining only concrete syntax	3,866666	4	4	0,81930724
S7—Proposing concrete and abstract syntaxes	4,5	5	5	0,73108327
S8—Checking consistency between abstract and concrete syntaxes	4,333333	4,5	5	0,88408664
S9—Relating concepts introduced by the extensions with the iStar concepts	4,3	4	5	0,79437678
S10—Proposing extensions with the smallest possible number of modifications and new representations	3,5	3	3	1,13714706
S11—Proposal of simple graphical representations, able to be drawn on the paper without a tool	3,966666	4	4	0,99942512
S12—Proposing new graphical representation only to represent constructs in same abstraction level of intentional elements, actors and iStar relationships	3,3	3	3	1,2077337
S13—Performing a careful choice of graphical representations	4,133333	4,5	5	1,10588810
S14—Dealing with the negative impact of conflicts and redundancies in the graphical representation	4,533333	5	5	0,62881022
S15—Reusing other existing extensions to improve the understanding and acceptance of new extensions	3,266666	3	3	1,01483252
S16—An iStar extension should not complicate the usage of iStar	3,866666	4	4	0,86036613
S17—Proposing a process or a methodology to guide the iStar extensions	4,2	4,5	5	0,99654575
S18—Defining extension mechanisms to iStar	4	4	5	1,01709525

Fig. 10 Detailed results of evaluation of interviews' findings

It is possible to identify in Fig. 9 and Table 4 that the mean of responses to all statements is above 3 (Neutral). Thus, it is an evidence of the importance of the statements for the community. Although the mean of S12 and S15 be above 3, they are closer to 3.

The mode gives the information about the most frequent answers. Table 4 shows the mode of all statements, and it is possible to identify that only three statements (S10—*Proposing extensions with a smallest possible number of modifications and new representations,* S12—*Proposing new graphical representation only to represent constructs in same abstraction level of intentional elements, actors and iStar relationships* and S15—*Reusing other existing extensions to improve the understanding and acceptance of new extensions*) have the value three in its mode. The other statements have the mode 4 or 5. These data show one more evidences about the importance of the statements for the experts.

In Fig. 10 we detail the distribution of the responses for each statement. We can relate both (Fig. 10 and Table 4) to understand the results better.

It is possible to identify in Fig. 10 that the number of responses *Very important* or *Important* is greater than the responses of *Totally Irrelevant* or *Unimportant* for all statements. The mode presented, in Table 4, shows that 5 (*Very important*) and 4 (*Important*) are the most frequent

responses to almost all statements. The standard deviation can also be used to understand this fact once these values are low (approximately 1).

It is important to highlight that the statements S7, S8, S13, S14 and S17 have a great amount of *Very important* responses (≥ 15 responses).

We can identify that S10, S12 and S15 (especially S15) have a great number of *Neutral*. Hence, they have the lowest value of the mean. All responses to this survey can be found in Table 5 of "Appendix B".

We performed the hypotheses tests using these data to analyse the results of the survey. The results are presented in Table 6 of "Appendix C".

6.3 Defining guidelines for future extensions

We believe that the statements of the findings can be useful for future iStar extensions, so we defined a set of guidelines from the statements extracted from the interviews and validated by the survey. The guidelines G1, G3, G4, G6 and G7 were defined, respectively, according to the statements S1, S3, S5, S8 and S9. The guidelines G2, G5, G8 and G9 were proposed by the join of pairs of statements.

The guidelines are presented below.

- G1—Preserve the language (iStar) original syntax;
- G2—Carry out consistent, complete and without-conflicts extensions and follow a process/method to do them;
- G3—Perform a literature review, include the participation of domain experts and iStar experts and model systems of application area before extending;
- G4—Describe a clear definition of the extension concepts;
- G5—Propose concrete and abstract syntax of the extension;
- G6—Check consistency between abstract and concrete syntaxes;
- G7—Relate concepts introduced by the extensions with the iStar concepts;
- G8—Define extensions with the smallest possible number of modifications and new representations in order not to complicate the use of the modelling language (iStar);
- G9—Propose careful and simple graphical representations, able to be drawn on paper without a tool.

We joined the four pairs of the statements because they are related one each other. The statements S2 (*Dealing with the negative impact of extensions that are carried out in an* ad hoc *fashion*) and S14 (*Dealing with the negative impact of conflicts and redundancies in the graphical representation*) in the guideline G2 (Carry out consistent, complete and without-conflicts extensions and follow a process/method to do them). The statements S6 (*Dealing with the negative impact of defining only concrete syntax*) and S7 (*Proposing concrete and abstract syntaxes*) were joined in the guideline G5 (*Propose concrete and abstract syntax of the extension*). The statements S10 (*Proposing extensions with a smallest possible number of modifications and new representations*) and S16 (*An iStar extension should not complicate the usage of iStar*) were joined in the guideline G8 (*Define extensions with a smallest possible number of modifications and new representations in order not to complicate the use of the modelling language (iStar)*). The statements S11 (*Proposal of simple graphical representations, able to be drawn on the paper without a tool*) and S13 (*Performing a careful choice of graphical representations*) were joined in the guideline G9 (*Propose careful and simple graphical representations, able to be drawn on paper without a tool*).

We did not consider the statements S12 and S15 in the guidelines definition, once their means were not significantly larger than three. The importance of them (S12 and S15) was not confirmed in the hypotheses tests (See "Appendix C"). The statement S4 (*Understanding and acceptance of iStar extensions*) represents a consequence of proposing good iStar extensions. Therefore, it is not a guideline and was not included in the list of guidelines.

Furthermore, the statements S17 (*Proposing a process or a methodology to guide the iStar extensions*) and S18 (*Defining extension mechanisms to iStar*) are considered future works, not a guideline.

7 Threats to validity

According to Kitchenham and Pfleeger [37], there are four aspects that we need to consider: Criterion Validity, Construct Validity, Face Validity and Content Validity.

The Criterion Validity is a measure of how well an instrument compares with another predecessor instrument. Construct Validity is the observation of how an instrument "behaves" when in use. It can be convergent or divergent. Face Validity is a superficial analysis of items by naive people, to test their understanding of it. Finally, the Content Validity is an assessment of how appropriate the instrument looks to part of participants.

In this section, we presented the threats to validity of both studies. Section 7.1 presents the threats to validity of the qualitative study, and Sect. 7.2 presents the threats to validity of the survey.

7.1 Threats to validity of the qualitative study

Criterion validity In the paper [12], which is similar to ours since it used mixed methods, the authors analysed documents of previous projects. They used semi-structured

interviews in the qualitative stage and the survey with a Likert-type scale of five levels in the quantitative part.

The documental analysis of previous iStar extensions was presented in the paper of SLR [27]. The qualitative study of the paper [12] was made to confirm the findings of the documental analysis and identify aspects not identified in the documental analysis. Our script interview, therefore, has a broader scope once we have more general questions to confirm and does not bias the responses of the participants.

Construct validity We recorded the audio during the interviews to make feasible their transcription and the analysis, so we asked for permission at the beginning of the interview. This information could inhibit the responses of the participants. We mitigated this threat informing the participants that the audio files and transcriptions would be maintained private and they would be anonymous. We also presented a confidentiality and privacy term.

Part of the question 7 of the qualitative study asked about iStar extensions which the participants consider not well done and reasons for their opinion. In almost all cases, the participants did not mention a specific paper with a bad extension but revealed what they considered a bad iStar extension in general. It was sufficient for us.

Face validity We tested the script interview with eight PhD students in computer science of our research group. They did not propose any iStar extension. We received several comments about the script interview; then, we corrected the script interview accordingly.

We can consider this previous evaluation a limitation because the number of participants is not so large (eight participants) in this previous step. However, we mitigated this threat asking them to evaluate again after the corrections of their comments.

Again, the participants of this test did not propose any iStar extension; however, they knew many iStar extensions and had already used some of them, thus mitigating this threat.

Content validity We performed a pilot involving two experts in iStar extensions to validate the script interview in the qualitative study. We can consider this previous evaluation a limitation because of the small number of participants.

We tried to mitigate this threat to Content Validity considering the feedback of the two experts in iStar extensions who participated in the pilot. We also mitigated this threat to Face Validity by the validation of the script interview by the fourth and the fifth authors of this paper.

During the interviews, the participants suggested three adjustments in the script interview. We tried to mitigate these threats to Content Validity with the corrections in script interview.

Conclusion validity There is a threat to conclusions in the qualitative study if the data collection technique is not robust enough. The sample of participants in the qualitative study was made considering the different universities from ten different countries. This threat, therefore, is primarily due to the remote location of a great part of the participants and the mix of languages used in conducting the research. These factors may have led to some misunderstandings that might have been relevant.

We believe, however, that these effects are at least partially mitigated by the fact that each interview was conducted by one author of this paper while a second author took notes. We also recorded the audio of interviews for a detailed transcription and analysis later. The misunderstandings and inconsistencies identified during the analysis of the qualitative study were discussed. When necessary, further explanations were sought from the participants to mitigate this threat to conclusion validity.

Furthermore, we did not do an independent double coding, that is, when two researchers do the coding independently of each other and afterwards compare whether they obtain the same results. This fact could compromise the results and conclusions. Nevertheless, we did a pair coding where the two researchers who conducted the interviews analysed the transcriptions and created the codes together. We believe that the approach that we followed can mitigate this threat.

7.2 Threats to validity of the survey

We now comment on the threats to the validity of the survey.

Criterion validity We did not find a previous survey for this purpose so that we could compare it with ours. It can be considered a threat to Criterion Validity.

The paper [12] is related to ours because it consists of a survey used to analyse the importance of a set of statements. The authors used the Likert-type scale in five levels.

Maybe a higher range of values could contribute to obtaining better results concerning the two inconclusive statements; however, we opted for the default scale with five values widely used in other surveys.

Construct validity When we invited the participants to answer this survey, we explained that the statements to be evaluated were identified in a previous study with researchers of iStar extensions. This information could have caused apprehension if they considered unimportant what other researchers considered important.

Despite this, we received 42 answers Totally Irrelevant (1) or Unimportant (2). Furthermore, we could not confirm statistically the importance of two statements with the hypotheses tests (see Table 6 in "Appendix C"), so this threat was not detected in the overall results.

Face validity The Face Validity of the survey is like the Face Validity of the interviews. We tested the survey with eight PhD students in computer science of our research group. They did not propose any iStar extension. We

received several comments about the survey; then, we corrected it accordingly.

We can consider this previous evaluation a limitation because of the small number of participants (8 participants). We mitigated this threat, however, by asking them to evaluate again after the corrections of their comments.

Despite the re-evaluation, the participants of this test did not propose any iStar extension, although they knew many iStar extensions and had already used some of them, thus mitigating this threat.

Content validity We performed the pilot involving five researchers. It was done to test the understanding of participants about the survey. We analysed the feedback sent by the participants of the pilot and applied the suggested improvements in the survey.

During the application of the survey, we received some comments from seven participants. We tried to mitigate these threats to Content Validity with the participation of two experts in iStar extensions in the development of this research, which validated the survey before the submission to the participants.

Conclusion validity In this survey, we did not have a large number of participants. We could not then make statistical inferences or to reveal a true pattern in the data. This threat for the conclusions validity can be mitigated inviting the researchers who will propose the next iStar extensions to answer this survey.

8 Conclusions

In this paper, we presented the main results of a basic qualitative research and a survey on how iStar has been extended. We described how the extensions have been proposed and the complex interaction between the positive and negative categories that influenced this process. Beyond analysing the current situation, we have compiled a set of actions (recommendations) suggested by participants to improve the way of proposing iStar extensions. Additionally, a set of statements/guidelines were identified. These contributions emerged from our analysis using grounded theory and may be useful for proposing future extensions.

The results indicate that extensions are carried out in an ad hoc fashion and that there are several different ways to propose extensions. Therefore, some critical activities seem to not be considered when proposing several new extensions (such as the definition of abstract syntax and consistency check between abstract and concrete syntax). The researchers believe that the extensions should be proposed in a precise manner and only when it is really needed. Hence, it is important to carefully analyse when new constructs should be proposed as first-order artefacts, which in turn demands a new graphical representation. This calls for an opinion from researchers of iStar and experts from the application area.

The reuse of existing extensions was mentioned as a practice to be considered. The definition of the extension mechanisms is also referred to as fundamental.

We also related the results of a previous SLR and the qualitative study. We were able to confirm some results found in the SLR with the participants of the qualitative study. For example, many of them mentioned that they only extend the concrete syntax; this information is consistent with the data found in the SLR.

On the other hand, we identified some unexpected information. We were surprised by two participants who stated that they did not propose any iStar extension, even when confronted with the new graphical representations introduced by them. Similarly, we were puzzled by the way that the changes to abstract and concrete syntax are made. We found out that some extensions were first proposed to the abstract syntax and later to the concrete syntax. Other authors, however, mentioned the opposite, that is, they first extended the concrete syntax and then the abstract syntax.

A novel information for us is related to what it means to be a good iStar extension. We highlight two issues about it: (i) good extension results in a work that is used by the community and (ii) good extension must not conflict with existing extensions.

We were startled by the need to better describe how the new concepts are related to original iStar concepts. A comment that iStar could have a richer textual representation was also a surprise for us.

We evaluated the findings of the qualitative analysis using a survey (quantitative study) with 30 iStar researchers. The survey presented 18 statements that represent the main findings of the interviews and analyse the opinions of the respondents about the importance of these findings. The results of the survey reiterate the importance of the findings of the qualitative analysis.

Based on the categories and statements identified, we proposed a set of guidelines to assist in the extension of iStar. It is likely that some of them may be useful to those interested in extending other modelling RE languages. For example, the following guidelines are more general: G2 (Carry out consistent, complete and without-conflicts extensions and follow a process/method to do them); G3 (Perform a literature review, participation of domain experts and iStar experts and model systems of application area before extending); G4 (Describe a clear definition of the extension concepts); G5 (Propose concrete and abstract syntax of the extension); G6 (Check consistency between abstract and concrete syntaxes); and G8 (Define extensions with the smallest possible number of modifications and new representations in order not to complicate the use of the modelling language).

We believe, therefore, that these guidelines could be applicable to other modelling languages in the RE area, such as KAOS and NFR; however, it is necessary to perform further studies with researchers to confirm or deny them.

We searched for similar papers for other languages and did not find works which investigate the extension of other modelling languages. It is not clear if iStar is unique or common in its extension issues. As future work, therefore, we intend to replicate the steps of our study with other modelling languages, such as UML, KAOS, NFR. In doing so, we will be able to make a more concrete comparison on the findings of the evolution of the extensions in each of them.

As ongoing work, we are currently working on a process to guide the extensibility of iStar considering the suggestions made by the researchers interviewed. It is based on the reuse of existing extensions and recommendations identified during the interviews. The process will guide the proposal of iStar extensions, defining the related concepts, abstract and concrete syntax, and keeping the traceability. Finally, it is important to illustrate the use of the process and, as well, the use of extension mechanisms of a new iStar extension in an application domain to be chosen.

Acknowledgements The authors thank all participants of this study. We also thank CNPQ/Brazil (Conselho Nacional de Desenvolvimento Científico e Tecnológico) for the financial support to the execution of this work, Universidade Federal do Ceará (UFC), LER-Universidade Federal de Pernambuco (LER/UFPE) and NOVA LINCS Research Laboratory (Ref. UID/CEC/04516/2013).

Appendix A: Script interview (complete version)

This is the complete interview script used to conduct the interviews. It is composed of 11 questions structured in 4 parts.

Part 1. Profile: pre-survey

- What is your current occupation (Professor/Researcher/Developer)?
- How many years of experience do you have using iStar?
- We identified the following iStar extensions proposed by you. (Show the list of iStar extensions of author identified). Are there any extensions to IStar done by you that we have not mentioned?

Part 2. Experience on iStar and extensions

1. Based on your experience, what is extending a modelling language?
2. How would you describe the process followed in the creation of your extension(s)? In other words, what were the tasks/activities performed since the moment of identification of the necessity of extending, up to the moment when the extension was done?
3. Contextualization: With your extension(s), new concepts were introduced in iStar through new forms of representation/modification of existing representations. How were these new concepts selected/chosen?
   - The identification was made based on the bibliography/references in the field? Systematic Literature Review? Others' studies?
4. Contextualization: Generally, a modelling language creation/extension involves the proposal of its abstract syntax and concrete syntax.

The abstract syntax is a way to represent the concepts involved in the modelling language in a structured way. This is done through a metamodel and well-formedness rules that are used to verify the correctness of the models to be created. The figure below shows an iStar metamodel (Fig. 11).

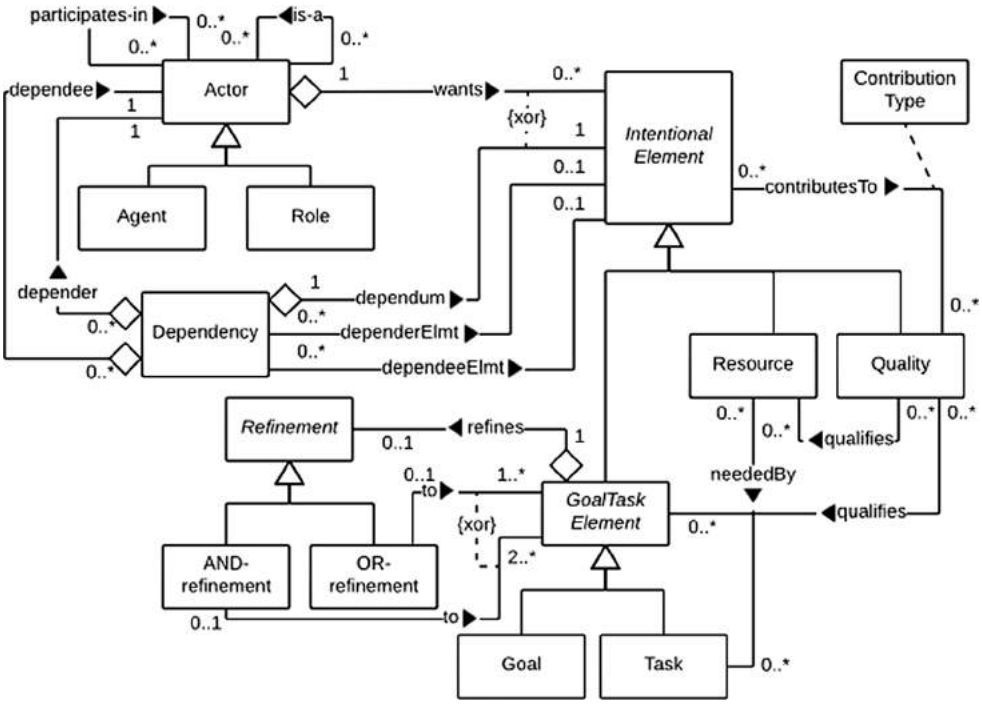
The concrete syntax is a graphical representation of a modelling language. Below is an example of a model that uses the concrete syntax of iStar (Fig. 12).

Considering the concepts presented above, how were these syntaxes specified in your extensions (abstract/concrete/both)?

- In case the abstract syntax has not been considered: Have you considered the representation of the extension in the abstract syntax? Why?
- In case the abstract syntax had been considered: How do you evaluate the importance of using the abstract syntax in your extension?
- There was some concern in maintaining consistency between the concrete and abstract syntaxes? In the case the response is yes, How? If the interviewed has difficulty: Through traceability between metaclasses of the metamodel and related graphical representation, for example.
- Do you think that it is important to maintain the consistence between them?
- Do you think that it is important to maintain the consistence between the extension and iStar syntaxes? In other words, is it important to represent the abstract and concrete syntaxes completely in the way we have defined?

5. What were the difficulties when defining the abstract and concrete syntaxes for your iStar extension(s)?
   - Have you reused some graphical representations of an existing extension? Why/why not?
   - How was chosen the graphical representation for the new constructs?

Fig. 11 iStar metamodel



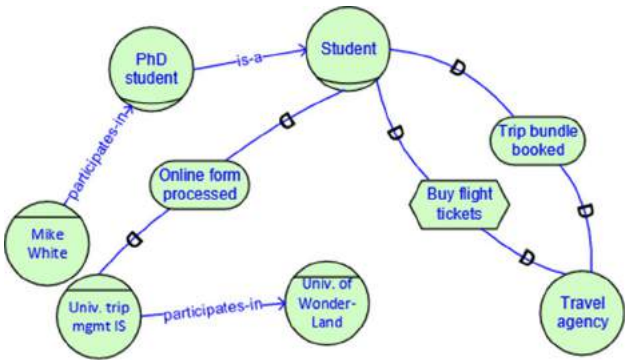


Fig. 12 Illustration of usage of concrete syntax of iStar

- Do you consider important a carefuly chosen the graphical representation?

6. What are the advantages of providing a modelling tool that supports the extension?
   - What can be done to help researchers to implement its extensions in tools?
7. Cite one iStar extension that you consider that was well done and why. Cite an example of an extension that you consider not so good and tell us why.

Part 3. Inconsistency analysis

8. Given the following two hypothetical scenarios related to iStar extensions to model multi-agent systems:

Hypothetical Scenario 1: Suppose there are two extensions that represent the same concept in two different graphical forms. For example:

- The Extension A add a diamond to represent *Commitment*;
- The Extension B uses a pentagon to represent *Commitment*.

Hypothetical Scenario 2: Suppose that there are two extensions that represent two different concepts using the same graphical form. For example:

The Extension A adds a triangle to represent *Norm*;

The Extension B uses a triangle to represent *Predicate* (Fig. 13).

Comment on the problems described in those scenarios in the following situations:

A user that receives an iStar diagram with norms and predicate.

A researcher that wants to reuse the notation of commitment in new extensions.

Part 4. Finalisation

9. Which actions could be done to ease the process of extending iStar?
10. Is there something about the extensions that we did not mention in the interview and you would like to talk about?
11. Do you have some question about the interview?

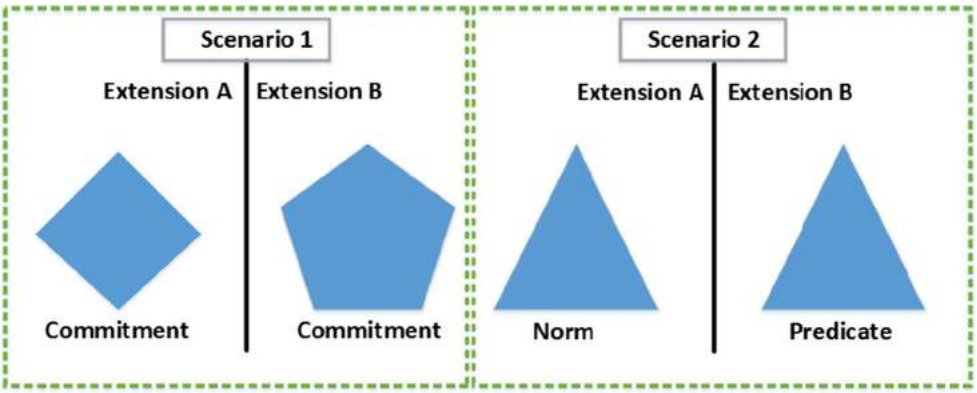


Fig. 13 Problems in a hypothetical situation of iStar extensions

Appendix B: Responses to survey

The responses to survey of Sect. 6 are presented in Table 5.

Appendix C: Evaluation survey data

We are interested in investigating if it is possible to consider the statements important to the iStar extensions

Table 5 Responses to evaluation survey

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18
R1	4	5	5	4	5	4	5	5	5	3	4	4	5	5	4	5	5	5
R2	5	3	5	5	2	3	5	5	5	4	4	3	3	5	3	3	3	5
R3	5	5	4	4	5	5	5	4	4	3	3	3	3	4	2	3	5	4
R4	5	5	1	5	5	4	4	4	4	2	4	4	5	5	5	4	5	5
R5	5	4	5	3	5	3	4	4	5	5	5	3	5	5	2	4	5	1
R6	5	5	5	4	5	5	5	5	5	4	1	1	2	4	3	3	4	5
R7	4	4	4	4	4	5	5	4	5	3	3	3	5	5	5	5	5	5
R8	5	4	5	4	5	3	5	5	4	3	5	4	5	5	5	5	5	4
R9	5	3	3	5	5	4	5	5	5	2	4	3	5	5	2	4	5	3
R10	3	4	4	4	5	4	5	4	4	3	4	3	3	5	3	5	2	3
R11	5	4	4	4	5	5	5	5	5	4	4	5	4	5	4	4	5	5
R12	3	5	4	5	5	2	2	5	4	3	4	2	4	5	3	5	2	3
R13	4	4	5	4	5	3	4	5	5	4	5	5	5	5	5	2	5	5
R14	5	4	4	5	5	3	5	5	4	2	3	2	4	5	4	3	4	5
R15	5	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4	5	4
R16	4	4	5	4	5	4	4	5	5	5	5	3	5	5	3	4	4	3
R17	2	4	2	4	5	4	4	3	3	5	2	2	4	4	3	4	4	3
R18	2	4	4	4	5	4	3	3	3	4	4	2	4	4	2	5	4	4
R19	4	3	4	4	4	4	4	4	4	5	5	4	4	5	3	4	5	4
R20	4	5	5	3	5	3	5	4	4	5	5	5	5	5	3	4	4	4
R21	3	3	5	5	4	4	4	4	5	5	4	3	4	3	3	3	4	4
R22	5	2	4	5	5	4	5	4	5	2	5	5	5	4	2	4	4	4
R23	5	4	2	4	5	5	5	5	5	3	4	4	3	4	3	4	2	3
R23	5	2	4	5	5	5	5	5	5	2	3	3	5	5	3	4	5	5
R25	4	4	4	5	4	3	4	4	4	3	4	4	4	5	4	4	3	3
R26	3	3	5	5	4	4	4	1	3	3	5	5	5	3	1	3	5	3
R27	2	1	4	5	5	3	5	4	4	2	5	1	5	4	3	3	3	3
R28	4	4	4	5	5	4	4	4	4	5	3	4	2	4	4	5	5	5
R29	3	4	5	4	2	4	5	5	4	2	3	2	1	4	3	2	4	5
R30	4	3	5	4	5	3	5	5	2	4	4	2	5	4	4	4	5	5

Table 6 Results of hypotheses tests

Statement	Hypotheses	v	*P* value
S1—Preservation of iStar original syntax	**H_0**	**305.5**	**3.871e−05**
	H_1	305.5	1
S2—Dealing with the negative impact of extensions that are carried out in an ad hoc fashion	**H_0**	**261**	**0.0004703**
	H_1	261	0.9996
S3—Literature review, participation of domain experts and use of iStar to model systems of application area before extending it	**H_0**	**396.5**	**3.452e−05**
	H_1	396.5	1
S4—Understanding and acceptance of iStar extensions	**H_0**	**406**	**9.882e−07**
	H_1	406	1
S5—Dealing with the negative impact of proposing extensions with an unclear definition of the concepts	**H_0**	**457**	**5.205e−07**
	H_1	457	1
S6—Dealing with the negative impact of defining only concrete syntax	**H_0**	**223.5**	**5.022e−05**
	H_1	223.5	1
S7—Proposing concrete and abstract syntaxes	**H_0**	**429**	**1.124e−06**
	H_1	429	1
S8—Checking consistency between abstract and concrete syntaxes	**H_0**	**385.5**	**9.321e−06**
	H_1	385.5	1
S9—Relating concepts introduced by the extensions with the iStar concepts	**H_0**	**371**	**3.472e−06**
	H_1	371	1
S10—Proposing extensions with the smallest possible number of modifications and new representations	**H_0**	**182**	**0.008756**
	H_1	182	0.9921
S11—Proposal of simple graphical representations, able to be drawn on the paper without a tool	**H_0**	**274**	**0.000137**
	H_1	274	0.9999
S12—Proposing new graphical representation only to represent constructs in same abstraction level of intentional elements, actors and iStar relationships	H_0	154	0.08568
	H_1	154	0.9198
S13—Performing a careful choice of graphical representations	**H_0**	**321.5**	**6.456e−05**
	H_1	321.5	0.9999
S14—Dealing with the negative impact of conflicts and redundancies in the graphical representation	**H_0**	**406**	**8.313e−07**
	H_1	406	1
S15—Reusing other existing extensions to improve the understanding and acceptance of new extensions	H_0	105.5	0.07913
	H_1	105.5	0.9279
S16—An iStar extension should not complicate the usage of iStar	**H_0**	**259**	**6.364e−05**
	H_1	259	0.9999
S17—Proposing a process or a methodology to guide the iStar extensions	**H_0**	**358.5**	**1.394e−05**
	H_1	358.5	1
S18—Defining extension mechanisms to iStar	**H_0**	**216**	**0.0001593**
	H_1	216	0.9999

Bold values indicate confirmed hypotheses

researchers, so we considered the following hypotheses for each statement:

- H_0: The statement is important to iStar researchers.
- H_1: The statement is not important to iStar researchers.

We chose the *Wilcoxon* test to test the hypotheses. The results of hypotheses tests are presented in Table 6. We tested H_0, that is, if the statement is important (greater than three). Following that, we tested H_1, that is, if the statement is not important (less than three).

When the *p* value is lower than 0.05 it means that hypotheses tested is true at a confidence level of 95%. The results of the hypotheses tests confirmed that S1–S17 are important (H_0) with 95% of confidence.

According to the results of S12 (*Proposing new graphical representation only to represent constructs in the same abstraction level of intentional elements, actors and iStar relationships*) and S15 (*Reusing other existing extensions*

Table 7 Script of RStudio to evaluate data of survey

Type of test	Script
Adherence test (Shapiro–Wilk)	cc < −c[a]
	shapiro.test(cc)
Hypotheses test (Wilcoxon)	cc < −c[a]
	wilcox.test(cc, mu = 3, conf.int = TRUE, alternative = "two.sided")
	wilcox.test(cc, mu = 3, conf.int = TRUE, alternative = "less")
	wilcox.test(cc, mu = 3, conf.int = TRUE, alternative = "greater")

[a]Here the values of responses for each statement should be inserted between comma

to improve the understanding and acceptance of new extensions), it is not possible conclude with 95% of confidence that they are important (H_0) or not important (H_1).

The results of the hypotheses tests if S12 was not conclusive for the H_0 and H_1. In extensions related to practical aspects, sometimes a different abstraction level is necessary for the constructs, such as modules, information about time and cardinality. These representations are useful to iStar, but they can be considered in a different abstraction level of intentional elements, actors and iStar relationships.

The results of the hypotheses tests of S15 were also not conclusive for the H_0 and H_1. We can understand the divergence of responses to this statement by the example given in the following. Considers two existing iStar extension E1 and E2, where E1 was well defined and E2 was not well defined. On the one hand, E1 is clear, complete, without inconsistencies and conflicts. On the other hand, E2 is unclear, incomplete and with inconsistencies and conflicts. Therefore, when E1 is reused, it can improve the acceptance and understanding of new extensions. When E2 is reused, however, probably it will not contribute to improve the acceptance and understanding of new extensions.

The scripts used to perform these tests using RStudio[5] are presented in Table 7.

References

1. Alencar F, Moreira A, Araújo J, Castro J, Silva C, Mylopoulos J (2006) Towards an approach to integrate i* with aspects. In: 8th International bi-conference workshop on agent oriented information system in 18th international conference on advanced information systems engineering
2. Alencar F, Castro J, Lucena M, Santos E, Silva C, Araújo J, Moreira A (2010) Towards modular i* models. In: ACM symposium on applied computing, pp 292–297
3. Ali R, Dalpiaz F, Giorgini P (2008) Location-based software modelling and analysis: Tropos-based approach. In: International conference on conceptual modelling, Lecture Notes in Computer Science, volume 5231. pp 169–182
4. Ali R, Dalpiaz F, Giorgini P (2014) Requirements-driven deployment. In: Software and systems modelling. Springer, Berlin, pp 433–456
5. Amyot D, Ghanavati S, Horkoff J, Mussbacher G, Peyton L, Yu E (2010) Evaluating goal models within the goal-oriented requirement language. Int J Intell Syst 25(8):841–877
6. Asnar Y, Giorgini P, Mylopoulos J (2011) Goal-driven risk assessment in requirements engineering. Requir Eng J 16(2):101–116
7. Babar Z, Nalchigar S, Lessard L, Horkoff J, Yu E (2015) Instructional experiences with modeling and analysis using the i* framework. In: iStar teaching workshop in 27th international conference on advanced information systems engineering, pp 31–36
8. Brambilla M, Cabot J, Wimmer M (2012) Model-driven software engineering in practice. In: Morgan and Claypool publishers series synthesis lectures on software engineering
9. Bennaceur A, Lockerbie J, Horkoff J (2015) On the Learnability of i*: experiences from a new teacher. In: iStar teaching workshop in 27th international conference on advanced information systems engineering, pp 43–48
10. Borba C, Silva C (2009) A comparison of goal-oriented approaches to model software product lines variability. In: Workshop on requirements, intentions and goals in conceptual modeling in 28th international conference on conceptual modeling, advances in conceptual modeling: challenging perspectives, Lecture Notes in Computer Science, volume 5833. Springer, Berlin, pp 244–253
11. Bresciani P, Perini A, Giorgini P, Giunchiglia F, Mylopoulos J (2004) Tropos: an agent-oriented software development methodology. Auton Agents Multi Agent Syst 8(3):203–236
12. Burnay C, Jureta I, Faulkner S (2014) An exploratory study of topic importance in requirements elicitation interviews. In: 26th international conference on advanced information systems engineering, lecture notes in computer science, volume 8484. Springer, Berlin, pp 180–195
13. Cares C, Franch X (2011) A metamodeling approach for i* model translations. In: 23th international conference on advanced information systems engineering. Lecture notes in computer science, volume 6741. Springer, Berlin, pp 337–351
14. Chung V (2006) Considering role-based conflicts of interest in analysing and designing e-health systems with goal-oriented methodologies. In: International conference on privacy, security and trust, paper 78
15. Chung L, Nixon B, Yu E, Mylopoulos J (2000) Non-functional requirements in software engineering. In: International series on software engineering, vol 5. Springer, US
16. Creswell J (2014) A concise introduction to mixed methods research. Sage Publications, Thousand Oaks
17. Dalpiaz F, Paja E, Giorgini P (2011) Security requirements engineering via commitments. In: 1st workshop on socio-technical aspects in security and trust, pp 1–8

[5] RStudio is a tool to perform statistical analysis based on commands in R. It is available to download at www.rstudio.com.

18. Dalpiaz F, Franch X, Horkoff J (2016) iStar 2.0 language guide. arXiv:1605.07767. Available in https://sites.google.com/site/istarlanguage/. Accessed 20 July 2017
19. Dardenne A, van Lamsweerde A, Fickas S (1993) Goal-directed requirements acquisition. Sci Comput Program 20(3):3–50
20. De Kinderen S, Ma Q (2015) Requirements engineering for the design of conceptual modelling languages. Appl Ontol 10(1):7–24
21. Elahi G, Yu E, Zannone N (2010) A vulnerability-centric requirements engineering framework: analysing security attacks, countermeasures, and requirements based on vulnerabilities. Requir Eng 15(1):41–62
22. France R, Rumpe B (2007) Model-driven development of complex software: a research roadmap. In: Conference on future of software engineering. IEEE Computer Society, pp 37–54
23. Franch X (2012) The i* framework: the way ahead. In: 6th International conference on research challenges in information science, pp 1–3
24. Gans G, Lakemeyer G, Jarke M, Vits T (2006) SNet: a modelling and simulation environment for agent networks based on i* and ConGolog. In: 14th international conference on advanced information systems engineering. Springer, Berlin, pp 328–343
25. Ghanavati S, Amyot D, Rifaut A (2014) Legal goal-oriented requirement language for modelling regulations. In: 6th International workshop on modelling in software engineering in 36th international conference on software engineering, pp 1–6
26. Giorgini P, Rizzi S, Garzetti M (2005) Goal-oriented requirement analysis for data warehouse design. In: 8th ACM international workshop on data warehousing and OLAP, pp 47–56
27. Gonçalves E, Heineck T, Castro J, Araújo J (2018) A systematic literature review of iStar extensions. J Syst Softw 137:1–33
28. Guzman A, Martinez A, Agudelo F, Estrada H, Perez J, Ortiz J (2016) A methodology for modeling Ambient Intelligence applications using i* framework. In: International iStar workshop in IEEE international requirements engineering conference, pp 61–66
29. He X, Ma Z, Shao W, Li G (2007) A metamodel for the notation of graphical modeling languages. In: 31th international computer software and applications conference, vol 1. IEEE Computer Society, pp 219–224
30. Horkoff J, Elahi G, Abdulhadi S, Yu E (2008) Reflective analysis of the syntax and semantics of the i* framework. In: 27th International conference on conceptual modeling, lecture notes in computer science, volume 5232. Springer, Berlin, pp 249–260
31. Horkoff J, Yu E (2010) Finding solutions in goal models: an interactive backward reasoning approach. In: 29th International conference on conceptual modeling, lecture notes in computer science, volume 6412. Springer, Berlin, pp 59–75
32. Ingolfo S, Siena A, Mylopoulos J, Susi A, Perini A (2013) Arguing regulatory compliance of software requirements. Data Knowl Eng 87:279–296
33. Ingolfo S, Jureta I., Siena A., Perini A., Susi A. (2014) Nomos 3: legal compliance of roles and requirements. In: 33th international conference on conceptual modeling, Lecture Notes in Computer Science, volume 8824. Springer, Berlin, pp 275–288
34. Ingolfo S, Siena A, Mylopoulos J (2014) Goals and compliance in Nòmos 3. In: 7th international i* workshop in 26th international conference on advanced information systems engineering
35. Islam S, Mouratidis H, Kalloniatis C, Hudic A, Zechner L (2012) Model based process to support security and privacy requirements engineering. Int J Secure Softw Eng 3(3):1–22
36. Kelly S, Tolvanen J (2008) Domain-specific modelling: enabling full code generation. Wiley, Hoboken
37. Kitchenham B, Pfleeger S (2002) Principles of survey research. Softw Eng Notes 26(6):16–27
38. Lapouchnian A, Yu Y, Liaskos S, Mylopoulos J (2006) Requirements-driven design of autonomic application software. In: 16th conference of the center for advanced studies on collaborative research, pp 80–94
39. Lapouchnian A, Mylopoulos J (2009) Modelling domain variability in requirements engineering with contexts. In: 28th international conference on conceptual modeling, Lecture Notes in Computer Science, volume 5829, Springer, Berlin, pp 115–130
40. Li T, Horkoff J, Mylopoulos J (2014) Integrating security patterns with security requirements analysis using contextual goal models. In: IFIP working conference on the practice of enterprise modelling, Lecture Notes in Business Information Processing, volume 197, pp 208–223
41. Liaskos S, McIlraith S, Mylopoulos J (2009) Towards augmenting requirements models with preferences. In: 24th IEEE/ACM international conference on automated software engineering, pp 565–569
42. Liaskos S, Mylopoulos J (2010) On temporally annotating goal models. In: 4th international i* workshop in 22th international conference on advanced information systems engineering, pp 62–66
43. Lima P, Vilela J, Gonçalves E, Pimentel J, Holanda A, Castro J, Alencar F, Lencastre M (2016) An extended systematic mapping study about the scalability of i* models. CLEI Electron J 19(3):1–6
44. Marosin D, Ghanavati S, Van Der Linden D (2014) A principle-based goal-oriented requirements language (GRL) for enterprise architecture. In: 7th international i* workshop in 26th international conference on advanced information systems engineering
45. Mate A, Trujillo J, Franch X (2014) Adding semantic modules to improve goal-oriented analysis of data warehouses using I-star. J Syst Softw 88:102–111
46. Mellado D, Mouratidis H, Fernandez-Medina E (2014) Secure Tropos framework for software product lines requirements engineering. Comput Stand Interfaces 36(4):711–722
47. Merriam S (2009) Qualitative research: a guide to design and implementation. Jossey-Bass, San Francisco
48. Miles R, Hamilton K (2006) Learning UML 2.0. O'Reilly, Newton
49. Moody D (2009) The physics of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Trans Softw Eng 35(6):756–779
50. Moody D, Heymans P, Matulevičius R (2010) Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. Requir Eng J 15(2):131–175
51. Morandini M, Penserini L, Perini A, Marchetto A (2015) Engineering requirements for adaptive systems. Requir Eng J 22(1):77–103
52. Mouratidis H, Giorgini P (2007) Secure tropos: a security-oriented extension of the tropos methodology. Int J Softw Eng Knowl Eng 17(2):285–309
53. Mouratidis H, Islam S, Kalloniatis C, Gritzalis S (2013) A framework to support selection of cloud providers based on security and privacy requirements. J Syst Softw 86(9):2276–2293
54. Mylopoulos J, Chung L, Yu E (1999) From object-oriented to goal-oriented requirements analysis. Commun ACM 42(1):31–37
55. Murukannaiah P, Singh M (2014) Xipho: extending tropos to engineer context-aware personal agents. In: 13th international conference on autonomous agents and multi-agent systems, pp 309–316
56. Siena A, Maiden N, Lockerbie J, Karlsen K, Perini A, Susi A (2008) Exploring the effectiveness of normative i* modelling: results from a case study on food chain traceability. In: 20th international conference on advanced information systems engineering, Lecture Notes on Computer Science, volume 5074. Springer, pp 182–196
57. Siena A, Mylopoulos J, Perini A, Susi A (2009) Designing law-compliant software requirements. In: International conference on

conceptual modeling, Lecture Notes in Computer Science, volume 5829. Springer, pp 472–486
58. Siena A, Jureta I, Ingolfo S, Susi A, Perini A, Mylopoulos J (2012) Capturing variability of law with nomos 2. In: 31st international conference on conceptual modelling, Lecture Notes on Computer Science, volume 7532. Springer, pp 383–396
59. Strauss A, Corbin J (2007) Basics of qualitative research: 2nd edn. In: Techniques and procedures for developing grounded theory, 3rd edn. Sage Publications, Inc
60. Schulz F, Meissner J, Rossak W (2013) Tracing the interdependencies between architecture and organization in goal-oriented extensible models. In: 3rd Eastern European regional conference on the engineering of computer based systems, pp 25–32
61. Teruel M, Navarro E, López-Jaquero V, Montero F, González, P (2011) CSRML: a goal-oriented approach to model requirements for collaborative systems. In: 33rd international conference on conceptual modeling, Lecture Notes on Computer Science, volume 6998, pp 33–46
62. Van Lamsweerde A (2008) Systematic requirements engineering: from systems goals to UML models to software specifications. Wiley, Hoboken
63. Yu E (1995) Modelling strategic relationships for process reengineering. Ph.D. Thesis on Computer Science, University of Toronto
64. Yu E. (1997) Towards modelling and reasoning support for early phase requirements engineering. In: 3rd IEEE international symposium on requirements engineering, pp 226–235
65. Yu E, Giorgini P, Maiden N, Mylopoulos J (eds) (2011) Social modelling for requirements engineering. MIT Press, Cambridge