

Supporting Creation of New iStar Extensions through a Process

Enyo Gonçalves^{1,3}, João Araujo², Jaelson Castro³

¹Campus Quixadá – Universidade Federal do Ceará – Brazil

²Faculdade de Ciência e Tecnologia – Universidade Nova de Lisboa – Portugal

³Centro de Informática – Universidade Federal de Pernambuco – Brazil

enyo@ufc.br, joao.araujo@fct.unl.pt, jbc@cin.ufpe.br

Abstract. *iStar is a goal-oriented requirements modelling language which has been used by industrial and academic projects of different domains. Modelling languages are commonly extended to add new constructs giving more expressiveness. iStar is often extended to incorporate new constructs. A study performed on iStar extensions identified 96 extensions and the occurrence of problems related to their quality. It was pointed out by experts in iStar extensions the need to propose a way to support the proposal of iStar extensions systematically to prevent the problem occurrence, increase the quality of extensions, and make extension creation a less challenging task. This work investigates how iStar extensions have been created and proposes a systematic way to guide the creation of quality extensions. A process to support the creation of new iStar extensions was proposed. The process was used to propose a new iStar extension and was analysed by experts. The results point to the usefulness of the process to propose new iStar extensions.*

Resumo. *iStar é uma linguagem de modelagem de requisitos que tem sido utilizada em projetos industriais e acadêmicos de diferentes domínios. Linguagens de modelagem são comumente estendidas para adicionar novos construtores e dar mais expressividade. iStar é frequentemente estendida para incorporar novos construtores. Um estudo realizado identificou 96 extensões do iStar e a ocorrência de problemas relacionados a sua qualidade. Especialistas apontaram para a necessidade de propor uma maneira de apoiar a criação de extensões de forma sistemática para evitar a ocorrência de problemas, aumentar a qualidade das extensões, e tornar a criação de extensões uma tarefa menos desafiadora. Este trabalho investiga como extensões de iStar têm sido criadas e propõe uma forma sistemática de orientar a criação de extensões com qualidade. Um processo para apoiar a criação de novas extensões de iStar foi proposto. O processo foi usado para propor uma nova extensão de iStar e foi analisado por especialistas. Os resultados apontam para a utilidade do processo para guiar a criação de novas extensões do iStar.*

Keywords. *Process, Extension, Modelling Language, Goal Modelling, iStar*

1. Introduction

Modelling languages (MLs) have been used to model several aspects of the system, including requirements. Over the last two decades, much attention has been paid to the area of goal-oriented requirements engineering (GORE) (Horkoff et al., 2019). Goal-oriented analysis is more appropriate for the earlier stages where the organisational goals are analysed to identify and justify software requirements (Mylopoulos et al., 1999). Some GORE MLs have been proposed, such as iStar. iStar (Yu, 1995) is one of the most used GORE languages, which also attracts a lot of attention from the research community. Extending an ML is to add new constructs, modify or remove the existing ones (Brambilla et al., 2012). MLs are frequently extended to adapt them to a specific application area or to improve practical aspects. The proposal of iStar extensions has increased in the last years.

This paper presents an overview of the results of a PhD thesis (Gonçalves, 2019) concerned with the analysis of existing iStar extensions and the proposition of methods, techniques and tools to support the creation of new ones. Context, research questions and objectives are presented following. Other sections of this paper present the background (Section 2) and related work (Section 3), results (Section 4) and conclusion and further research (Section 5).

1.1. Context, Research Questions and Objectives

We identified 96 extensions in a Systematic Literature Review (SLR) (Gonçalves et al., 2018b), 42.7% of them did not present (or partially presented) the meaning of constructs introduced. Only 62.5% of them presented only the concrete syntax and 77.8% of extensions with a metamodel present absence of iStar nodes and links (not conservative extensions (Brambilla et al., 2012)). 37.14% of them have no compatibility between abstract and concrete syntaxes, and 53.6% are not supported by a modelling tool. 108 conflicts related to semiotic clarity (see Section 2.3) were identified. Additionally, experienced researchers in iStar extensions mentioned in (Gonçalves et al., 2019b) they do not follow any guidance. The consequence is that some tasks sometimes are neglected/not remembered, and this has been causing the occurrence of above-mentioned problems. They also point out that a kind of guidance, such as a process, to support the proposal of iStar extensions could contribute to avoiding these problems and proposing quality extensions (Gonçalves et al., 2019b).

Thus, this research is concerned to answer the following Research Questions (RQ): **RQ1:** What are the existing iStar extensions? **RQ2:** How have the iStar extensions been proposed? **RQ3:** What can be done to improve the quality of future iStar extensions proposals? **RQ4:** How can a process be designed to support the proposal of iStar extensions more completely, consistently and without conflicts? **RQ5:** What is the effect of using this process to propose iStar extensions? **RQ6:** What is the perceived usefulness, ease of understanding and intention to use/ease of use of the proposed process?

The general objective of this work is to investigate how to improve the quality of iStar extensions by developing a process which is useful and suitable for supporting the proposals of iStar extension in order to avoid the absence of the construct's meaning and metamodel; inconsistencies between the constructs' meaning, metamodel and concrete syntax; and the proposal of new symbols avoiding semiotic clarity problems

(see Section 2.3). The following specific objectives are required to achieve this general objective:

- Identify the existing iStar extensions, how they have been created and what can be done to improve the next extension proposals;
- Define a process to support the proposal of iStar extensions;
- Analyse the outcomes of applying the process to create a new iStar extension;
- Explore the usefulness, ease of understanding and intention to use/ease of use of the proposed process.

2. Background

This section presents iStar, development and extension of ML and quality in ML.

2.1. iStar and iStar Extension

iStar is a goal-based modelling language proposed in the nineties (Yu, 1995). It is an ML used to model software at the requirements level. It has been used in different domains such as enterprise engineering, security, privacy and trust.

In the iStar, stakeholders are represented as actors that depend on one 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 provides a description of the links and external dependencies among organisational actors. The SR model enables an analysis of how the goals can be fulfilled through contributions from several actors.

In June 2016, iStar had evolved to version 2.0 (Dalpiaz et al., 2016). Some changes were performed in this new version of the language. Some concepts were discontinued, and new concepts were introduced. The new version maintained the representation of general actors, roles and agents, but position is not part of iStar 2.0. The is-a link was maintained; is-part-of, plays, occupies and covers were unified in participates-in link, and INS is not used in iStar 2.0. Intentional elements goal, task and resource were not changed and softgoal was named quality. Means-end and task decomposition links were unified in refinement, while contribution links were maintained and two new links were proposed in iStar 2.0: qualification and neededBy.

2.2. Modelling Languages Development and Extension

A modelling language is defined by its abstract syntax (metamodel and well-formedness rules) and concrete syntax. Along with modelling concepts, we also identify various domain rules, constraints, and consistency needs which a language should follow. For a modelling language to be usable by software designers, it is necessary to define a set of models and their graphical and textual elements (Brambilla et al., 2012).

Extending a modelling language is to add new constructs or modify the old ones (Brambilla et al., 2012). According to the way new concepts are proposed, an extension can be developed using a light-weight (textual markers) or heavy-weight (new symbols)

strategy (Brambilla et al., 2012). We can classify an extension according to its impact on the original syntax of the language as "conservative", which keeps the original syntax without changes, or "non-conservative", which apply changes or reductions.

2.3. Quality of Modelling Languages

Lindland, Sindre and Solvberg (1994) define analysis of models based on three quality levels: Semantic Quality, Syntactic Quality and Pragmatic Quality. Semantic quality is concerned with the meaning of the models, syntactic quality is related to the graphical and textual representation of the models, and the pragmatic quality is related to the creation of the models.

In this context, Moody (2009) establishes a framework with nine principles to construct visual notations in SE. The nine principles are cognitive integration, cognitive fit, perceptual discriminability, manageable complexity, semiotic clarity, graphic economy, dual coding, visual expressiveness and semantic transparency.

Semiotic Clarity Principle establishes that it should be a 1:1 correspondence between semantic constructs and graphical symbols. When there is not a 1:1 correspondence, one or more of the following anomalies can occur (Figure 1): **Symbol deficit** (a construct without any symbol), **Symbol redundancy** (a construct with multiple symbols), **Symbol overload** (a symbol used by multiple constructs) and **Symbol excess** (a symbol without any construct).

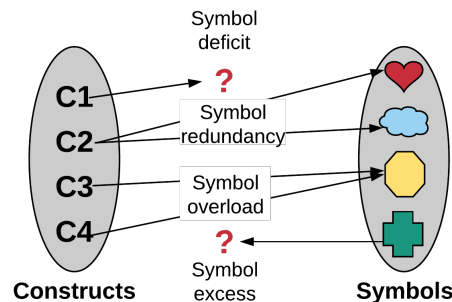


Figure 1. Principle of semiotic clarity (based on (Moody, 2009)).

Complementary, Caire et al. (2013) presents a multi-step study to create visual notation of modelling languages based on Moody.

3. Related Work

We did not find any process to support the proposal of iStar extensions. Thus, we presented here works involving processes and iStar and about guidelines in UML.

The work of Franch et al. (2011) defines an approach to join iStar with other MLs. Their paper presents results of a search for the references that join iStar with other MLs. It also presents guidelines to be followed considering the theoretical, technical, methodological and community aspects. However, it is not related to iStar extensions when the concepts to be introduced to iStar are not from an existing modelling language. It has not considered important concepts such as PoN (Moody, 2009) and the compilation of opinions of iStar experts. It also does not present a catalogue.

The process proposed in this work uses a set of guidelines identified as a result of a qualitative study with the experienced extenders of iStar. In this way, the ITU-T

Z.119 (2007) establishes guidelines for UML profile design. These guidelines define a set of recommendations about how to create a profile document, how to specify semantics and notation. The ITU-T Z100 (2018) standard mentions the definition of textual description in an extension of modelling languages. However, we did not find any work about guidelines about the creation of iStar extensions.

4. Results

This section presents a general description of the main results of this research.

4.1. An SLR and a Catalogue of iStar Extensions

The detailed version of this SLR is available in (Gonçalves et al., 2018b). We followed the guidelines by Kitchenham and Brereton (2013). Thus, we defined the following string search: (“i*” OR “framework i” OR iStar OR i-star OR eye-star OR “Goal-oriented Requirement Language (GRL)” OR Tropos) AND (requirements OR modelling OR modelling) AND (goal modeling OR goal modelling OR goal-oriented) AND extension OR extends OR extended OR extensibility OR patterns OR profile). We performed a search in ACM, EI Compendex, IEEE Xplore, Science Direct, Scopus, ISI Web of Science and Springer databases considering papers published until 2016 (the year that this SLR was performed). The selection was performed in pair, and we followed inclusion and exclusion criteria. We also performed snowballing and asked the specialists to suggest papers. We used eight quality assessment criteria. Finally, 96 iStar extensions were selected, their distribution per year is presented in Figure 2.

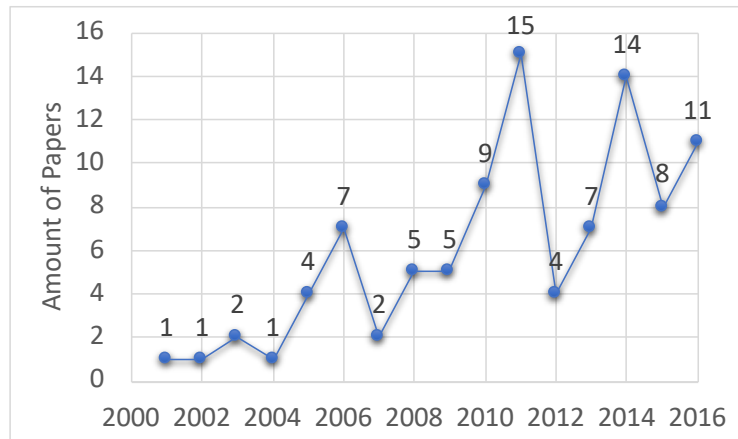


Figure 2. Distribution of selected papers per year.

Data were extracted to answer eight research questions (SLR_RQ). Following we presented the answer to them.

SLR_RQ1. What are the application areas of iStar extensions? We the iStar extensions identified in nine categories: Social and Intelligent Systems were classified in first place with 19 works each one (19.8%); Tied in second place with 18 papers (18.7% each one) identified, we have Security; Contextual and Enterprise each have 17 papers and are classified in third place; General Development has 15 papers published (15.6%); Software Product Lines has five papers (5.2%), this group is in fifth place, and Aspects, in the sixth, has three papers (3.1%); The remaining papers were grouped as other Non-Functional Requirements – it has 11 papers (11.45%).

SLR_RQ2. Do the papers present the meaning of the constructs? We used three possible values: Meaning Presented (MP), Meaning Partially Presented (MPP) and Meaning Not Presented (MNP). The results point to MP: 55/96 papers (57.3%), MPP: 35/96 papers (36.5%) and 6/96 papers (6.2%) MNP.

SLR_RQ3. Which syntax level do the extensions address (abstract, concrete or both)? Most of the papers (60/96; 62.5%) extended only the concrete syntax of iStar without representing the new constructs in a metamodel. Some of the papers (35/96; 36.5%) extended both syntaxes (Concrete and Abstract), and only one paper (1%) extended only the abstract syntax.

SLR_RQ4. How was the abstract syntax extension proposed? We analysed the presence of iStar default constructs in the extension metamodel. We classified the extension metamodel as complete or not complete. We considered papers which presented abstract syntax (36 papers). Thus, we found 8 papers complete and 28 papers not complete. We also analysed papers related to the presence of well-formedness rules. Thus, we identified 12 papers which define well-formedness rules and 84 papers which not define them.

SLR_RQ5. Is there compatibility between the metamodel and the concrete syntax of the extensions? We analysed papers that extend both syntaxes (35 papers). These papers were classified as Compatible or Not Compatible. 62.86% (22/35) of the papers were considered compatibility and 37.14% (13/35) of the papers do not have compatibility between both syntaxes.

SLR_RQ6. How was the concrete syntax proposed in the extensions? We analysed the papers that extended iStar in concrete syntax or in both syntaxes (95 papers, the paper that extends only the abstract syntax was not considered). We found 307 constructs, which 54% (167 constructs) are nodes and 46% (140 constructs) are links. Furthermore, we identified 31.6% (30/95) of extensions that proposed only new nodes, 13.7% (13/95) of extensions that proposed only new links and 54.7% (52/95) of extensions involve the proposed new nodes and links.

SLR_RQ7. Are there conflicts with the constructs in the concrete syntax of the extensions? The analysis of conflicts is related to the occurrence of semiotic clarity problems: Symbol deficit, Symbol redundancy, Symbol overload and Symbol excess (see Section 2.3). We found the occurrence of 108 problems.

SLR_RQ8. Is there any process defined to support the creation of iStar extensions? We did not find any process to support the creation of iStar extensions during this SLR.

Additionally, we proposed CATIE (Gonçalves et al., 2018c), a catalogue to group the iStar extensions and their constructs. This catalogue is available at <http://istarextensions.cin.ufpe.br/catalogue/>.

4.2. A Mixed-method Study with Experts

This section presents a mixed-method study (Creswell, 2014) based on interviews (Merriam, 2009) and survey (Kitchenham and Pfleeger, 2002) with experts in iStar extensions. A detailed version of this study is available in (Gonçalves et al., 2019b).

We made the pilot to the interviews and survey with two experts in iStar extensions. The interviews were performed via Skype during September and October 2016, and the survey was applied via google forms between November and December 2016. We interviewed 20 experts and other 30 answered the survey.

Main questions of the script interview are presented following: 1) What is extending a modelling language? 2) How would you describe the steps followed in the creation of your extensions? 3) What were the difficulties when defining the abstract and concrete syntaxes for your iStar extensions? 4) Cite one iStar extension that you consider well done, and one that you consider not so good and why; 5) Which actions could be done to ease the creation of iStar extensions?

Main interviews findings were used as questions of a survey. Participants of the survey chose the level of importance they consider each statement in Likert scale of five levels (Very important, important, Don't know, Unimportant and Totally irrelevant).

Interviews results

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. The participants mentioned a set of tasks generally performed when creating an iStar extension. We joined these tasks following: perform a literature review, describe the meaning of the constructs, search existing extensions, propose abstract syntax, propose the concrete syntax and illustrate the usage of the extension.

We analysed the interviews and created codes, which were grouped and produced core categories and their factors. We identified three main categories during the analysis of the interviews: Preservation of iStar original syntax, Motivation to extend iStar represents and Acceptance of the iStar extension. We presented their factors below.

Preservation of iStar original syntax means that the iStar's original constructs are maintained in the extension, that is, the iStar extension is conservative. It is related to some positive (+) or negative (-) categories such as the 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.

Motivation to extend iStar 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 (+).

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 (+).

Finally, we analysed how to improve the next iStar extensions. Main findings are listed following: 1) Create a repository of iStar extensions; 2) Include only new essential constructs; 3) Create a kind of guidance about new constructs; 4) Propose extension mechanism to iStar; 5) Try to use textual properties instead of new symbols.

Survey results

We summarised the main findings of the interviews in 18 statements. Each statement was analysed by the participants of the survey. 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 and iStar experts and use of iStar to model systems 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 iStar original constructs; 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 method to guide iStar extensions; S18: Defining extension mechanisms to iStar.

Only three statements (S10, S12 and S15) had the value of neutral in their mean and mode. These statements have a high number of Neutral. However, differently of S12 and S15, S10 got a great number of very important (8) and important (6). Thus, we did not consider the statements S12 and S15.

As the final result, we grouped the statements in nine guidelines. G1: Preserve the iStar original syntax, i.e., propose conservative extensions; 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.

4.3. PRISE: A Process to support iStar extensions

Processes are important because they give consistency and structure to a set of activities. These characteristics are useful when we know how to do something well through existing projects, and we want to ensure that others do the same in the execution of the next projects. Processes are also important because they allow us to

gather existing experiences and pass them on from the savvy to the novice (Pfleeger, 2003).

PRISE (Gonçalves, 2019) (Gonçalves et al., 2020) is the main result of this research. Its main goal is to conduct the creation of new quality iStar extensions systematically, enforcing the completeness, consistency and absence of conflicts. It was proposed based on a set of studies: SLR of iStar extensions (Gonçalves et al., 2018b) and Catalogue of iStar extensions (Gonçalves et al., 2018c) (Section 4.1), Mixed-method study with iStar extenders and Guidelines of iStar extensions (Gonçalves et al., 2019b) (Section 4.2).

PRISE is a defined process, i.e., it documents what is done (product/artefacts), when (steps/tasks/activities/sub-processes), by whom (roles), things that are used (inputs) and things produced (results/artefacts). PRISE is modelled using Business Process Model and Notation (BPMN). An interactive version of this process is available in <http://www.cin.ufpe.br/~ler/prise>, and it is possible to view and iterate in all processes and templates for the artefacts. PRISE tool (Gonçalves et al., 2019c) was proposed to support the creation based on this process.

PRISE has three roles. Extender: It represents who extends iStar, e.g. a PhD/master student and his/her advisor; Expert in iStar extensions: It represents the specialists in iStar extensions. We presented in (Gonçalves et al., 2018b) a ranking of authors which published more iStar extensions that represent these experts (not exclusive); Expert in domain/application area: It represents the specialists of the domain/application area targeted by the extension, e.g. an expert in multiagent systems.

The main artefacts of the PRISE are listed below:

- Extension specification: It is an artefact which is created by the sub-processes. It joins the results of all sub-processes. We used the status (Analysed, Concepts described, Developed or Validated/evaluated) between brackets to identify the execution level;
- Catalogue of iStar extensions: This is the catalogue of iStar extensions resulting in the SLR of Section 4.1;
- List of concepts to be reused: the constructs which can be reused are listed and the reference to the extension which define them;
- Guidelines of iStar extensions: The guidelines used by PRISE were presented in Section 4.2;
- Concepts meaning, abstract syntax and concrete syntax: These three artefacts describe the representation of the list of constructs in three different levels;
- Checklist for verification of problems: It is used as quality assurance regarding the occurrence of problems of incompleteness, inconsistencies and conflicts.

Main flow of the PRISE is presented in Figure 3. The flow starts with the intention to extend iStar. We presented an overview describing each sub-process, their main tasks and artefacts on the next paragraphs.

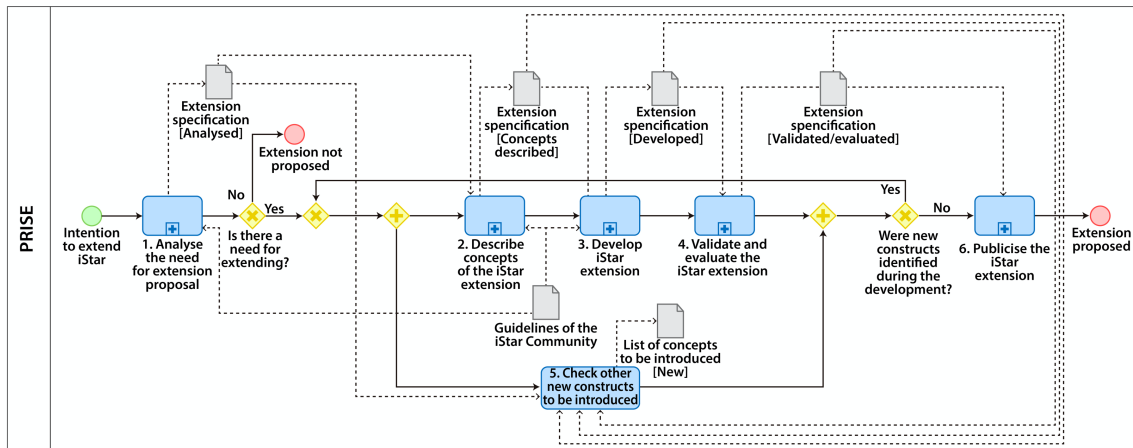


Figure 3. Main flow of PRISE.

Sub-process 1 - Analyse the need for extension. This sub-process analyses the need to propose the extension and creates the list of the concepts to be introduced (without a description of them yet). The sequence of tasks involves the study of the application area by a literature review, the identification of concepts to be introduced, mitigate issues with experts in iStar and in the application area, try to model an example of the application area with the iStar default syntax and use the catalogue of iStar extensions to identify whether there is an existing extension similar to the new one. Finally, it generates the Extension specification [Analysed].

Then, a decision is made based on the analysis of the need for an extension proposal. If there is a need for the extension, the process execution continues with the proposal. Otherwise, the process ends without proposing an iStar extension. When the process execution continues, three sub-processes (2, 3 and 4) are executed in sequence and task 5 is executed in parallel to these three sub-processes.

Sub-process 2- Describe concepts of the iStar extension. This sub-process describes the meaning of the concepts identified in sub-process 1. It is important to try to identify constructs to be reused and to analyse whether it is possible to relate them with the iStar constructs. It creates the Extension specification [Concepts described].

Sub-process 3- Develop iStar extension. This is the main part of PRISE. The Extender should consider the set of guidelines provided by iStar experts for the elaboration of extensions before starting this sub-process. This sub-process introduces new constructs to iStar. The concepts meaning, abstract syntax and concrete syntax are created in this sub-process. The proposal of iStar extension mechanisms (Gonçalves et al., 2018a) is related to concrete syntax definition. The creation of new symbols is performed based on a careful way, such as the Caire et al. (2013) method. The checklist for verification of problems is used to analyse the extension developed. The creation of a modelling tool to support the extension is focused on this sub-process.

Sub-process 4- Validate and evaluate the iStar extension. This sub-process illustrates the usage of the extension, validates the extension with experts, refines and evaluates the iStar extension developed in sub-process 3. Corrections and improvements are applied to the extension when necessary. The checklist for verification of problems is used one more time here. This sub-process generates the Extension specification [Validated/evaluated].

Task 5- Check other new constructs to be introduced. PRISE is an iterative process. Thus, new constructs (different from the set identified in sub-process 1) can be identified during sub-processes 2, 3 and 4 by this task. New constructs are listed by this task to be considered in the next iteration. This task generates the List of concepts to be introduced [New]. When new constructs are identified in task 5, the execution of PRISE returns to task 2. Otherwise, the execution continues to the sub-process 6.

Sub-process 6- Publicise the iStar extension. This sub-process details how to make the iStar extension accessible. The new extension should be added to the catalogue of iStar extensions, endorsed by an expert in iStar extension and published in journal or conferences. Finally, the proposal of the iStar extension is finished.

4.3.1. A New Extension Proposed Using PRISE

This section presents a new iStar extension created with PRISE to model MAS with rational agents. A detailed version of this study is available in (Gonçalves et al., 2019a).

Multi-agent system (MAS) is the area of Artificial Intelligence that investigates the behaviour of a set of autonomous agents. Russell and Norvig (2003) define four types of rational agents: Simple Reflex Agent, Model-Based Reflex Agent, Goal-based Agent and Utility-Based Agent. We identified and described the meaning of the constructs. We also represented the extension in the iStar metamodel and created validation rules. The extension metamodel and validation rules are available¹. The representations of the extension concepts can be classified into three groups:

1. Constructs represented by iStar constructs as proposed: five domain concepts are represented by the iStar constructs. They are used to represent an agent being part of an organization, an agent inhabiting an environment, an organization inhabiting an environment, an agent playing a role in an organization which inhabits an environment and a dependency between an agent and an environment. Furthermore, we extended neededby to connect next-function and beliefs establishing that beliefs are needed by the next-function;
2. Constructs represented by iStar constructs added with textual markers: These constructs have a similar meaning of the iStar constructs and specialize them by textual markers. They are four stereotypes (<<simple-reflex>>, <<model-based-reflex>>, <<goal-based>> and <<utility-based>>) applied to agents or agent roles, <<action>> and the specific functions (<<next-function>>, <<formulate-problem>>, <<formulate-goal>> and <<utility-function>>) applied to task. When an action is represented inside the agent roles, it can be defined as a right (an action that can be executed) using the property {type='right'} or a duty (an action that should be executed) using the property {type='duty'};
3. Constructs represented by new symbols: we found the Plan in an existing iStar extension and reused it. Four concepts (Environment, Organization, Perception and Planning) are represented by new symbols. Thus, the new symbols proposed to these four concepts were created using an experiment based on the work of Caire et al. (Caire et al., 2013). We performed a five-step experiment with 152 participants. All steps of this experiment are available².

¹ www.cin.ufpe.br/~ler/iStar4rationalagents/metamodel&rules

² www.cin.ufpe.br/~ler/iStar4rationalagents/experiment-representations

Figure 4 presents the final graphical representations of this experiment. We also modelled a MAS to support distance education courses³, created a modelling tool⁴ and applied a survey with researchers of MAS from Brazil⁵.

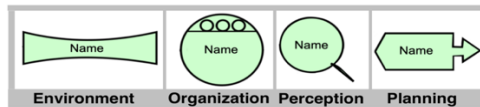


Figure 4. New Symbols of iStar4RationalAgents.

4.3.2. Evaluation of the PRISE

This section presents the main results of a mixed-method study (Creswell, 2014) based on interviews (Merriam, 2009) and survey (Kitchenham and Pfleeger, 2002) with experts to identify their point of view about PRISE (Gonçalves et al., 2019b). The population of this study is the 153 authors of iStar extensions identified by the SLR presented in Section 4.1. We invited 5 authors to the interviews (3 participated) and 148 authors to participate in the survey (20 participated). We made a pilot with two experts in iStar extensions. The interviews (via Skype) and survey (via google forms) were performed during June and August 2018. The participants watched a video about the PRISE available in https://youtu.be/_LF4u-MOsD0.

[Interviews]: Main questions of the script interview are presented following: 1) What is lacking to create iStar extensions more systematically? 2) Is it useful a process for supporting iStar extensions? Why? 3) What is your opinion about PRISE? What are its strongest points? What are its weaknesses? 4) What is the effort to understand PRISE? 5) Is it suitable to create iStar extensions? Why? To whom? 6) Would PRISE be useful to create your next extensions? 7) Do you have comments about PRISE?

Participants P1 and P3 cited some parts of PRISE as necessary for creating iStar extensions more systematically, such as the guidelines, and to verify problems of incompleteness, inconsistencies and conflicts. P2 mentioned a process that systematises the way in which they are proposed.

The three participants of this study were unanimously affirmed that, excluding PRISE, they do not know any process proposed to support iStar extensions. P1 and P2 mentioned that it is useful to propose a process or another kind of guidance, such as a method or guidelines. Participant P3 was more emphatic about the definition of a process: “I think a process defined can be more systematic and help to pay attention to all the different aspects that would not be necessarily have considered without it.” (P3).

As strong points, P1 cites that the tasks of PRISE are suitable and contribute to the extenders do not forget any step of the proposal, such as defining the metamodel or checking inconsistencies. P2 mentioned that principles and fundamentals behind it are very solid. P3 said that the description is clear and detailed and there is no such process defined. The size (number of tasks) was mentioned as a weakness by all participants.

The participants consider PRISE easy to understand and that the number of steps may imply it is difficult to follow. They agree that PRISE is suitable to support the

³ www.cin.ufpe.br/~ler/iStar4rationalagents/sdmoodle and www.cin.ufpe.br/~ler/iStar4rationalagents/srmoodle

⁴ <https://www.cin.ufpe.br/~ler/piStar4rationalagents/>

⁵ <https://www.cin.ufpe.br/~ler/iStar4rationalagents/evaluationsurvey>

proposal of iStar extensions by the iStar community. P2 commented that he did not see any reasons why to say that it is not suitable, and it can definitely improve the way that people can avoid common mistakes. Regarding to whom it is suitable, they said it is suitable to both experienced and novices, but mainly for novices.

We received different kinds of responses regarding whether PRISE is useful to create their next extensions. P1 commented that is not focused on proposing new iStar extensions at the moment, P2 said he/she could consider using PRISE when proposing his/her next extensions, and P3 will try to use it.

Regarding improvements, the participant P2 suggested maintaining the tools developed in more robust repositories such as GitHub (github.com/) because sometimes the researcher makes it available in a link in places that are not permanently accessible. We introduced this suggestion in the task 3.5.4. Make the tool available in a link.

[*Survey*]: We submitted 20 statements about PRISE, and each one was analysed by the participants of the survey. Table 1 shows of mean and mode for questions 1, 2, 3, 4, 5, 6, 8, 10, 13, 15, 16 and 17. These questions used Likert scale of answers (Strongly Agree, Agree, Don't Know, Disagree and Strongly Disagree). All medians and modes are Strongly Agree or Agree. Answers Strongly Agree and Agree received an average of 16.91 responses for each question, representing 84.58% of the responses.

Table 1. Median and Mode of the Questions.

Questions	Median	Mode
1. There is a lack of process to guide the creation of new iStar extensions	Agree	Strongly Agree
2. The proposal of a process for supporting iStar extensions is necessary	Strongly Agree	Strongly Agree
3. It is relevant following the guidelines of the iStar community	Strongly Agree	Strongly Agree
4. It is important to try to reuse the constructs of existing iStar extensions	Strongly Agree	Strongly Agree
5. A catalogue can help to identify the constructs to be reused	Strongly Agree	Strongly Agree
6. It is important to verify the completeness	Strongly Agree	Strongly Agree
8. It is important to verify the consistency between constructs meaning, abstract and concrete syntaxes	Strongly Agree	Strongly Agree
10. It is important to verify the occurrence of conflicts	Strongly Agree	Strongly Agree
13. The use of PRISE is feasible	Agree	Agree
15. PRISE is important to create future iStar extensions	Agree	Agree
16. The iStar community can benefit from the usage of PRISE	Agree	Agree
17. PRISE can be useful to create my next extensions	Agree	Agree

Questions 7, 9 and 11 are about measuring the importance level to avoid inconsistencies, incompleteness and conflicts. We used a scale from 0 to 5 in these questions. Question 7 (contribution of the PRISE to avoid incompleteness); we received only responses between 3 and 5 with median 4. In Question 8 (contribution of the PRISE to avoid inconsistencies), we received one response 0, one response 2, and the other 18 responses between 3 and 5 with a median of 4. Finally, in Question 11 (contribution of PRISE to avoid conflicts), we received one response, 2, and the other 18 responses between 3 and 5 with a median of 4.

Questions 12, 14 and 18 are about the difficulty level to understand, the suitability of PRISE to propose future extensions and the intention to use PRISE in the next iStar extensions. We identified in these questions that 90% of the participants considered the difficulty level to understand PRISE medium or easy, 70% considered PRISE Very Suitable or Suitable and the all participants pointed out to use PRISE.

Questions 19 and 20 are about to whom PRISE can be useful and if the participants recommend PRISE to other researchers. In Question 19, two (2) participants pointed out that PRISE is useful for experts, one (1) participant pointed out as useful to extenders without experience, and eleven (11) pointed to it as useful for both (experts and without experience). In Question 20, 70% per cent of the participants answered yes and 30% answered maybe. We had not any No response.

5. Conclusions and further research

Extensions have been occurring in modelling languages. In this work, we analysed existing iStar extensions and proposed a way to create new quality iStar extensions.

We answer the RQs (presented in Section 1) following. (RQ1) The existing iStar extensions were analysed by the SLR of Section 4.1. (RQ2) We analysed how extensions were created in the SLR (Section 4.1). (RQ3) Experienced extenders pointed out a set of actions that contribute to the proposal of iStar extensions (Section 4.2) such as engagement of experts, reuse of existing extensions, literature review and use iStar to model a system before the extension proposal. We also presented a set of guidelines. (RQ4) We proposed PRISE (Section 4.3), a process to support iStar extensions completely, consistently and without conflicts. We modelled this process in a detailed way, presenting the description of each task and artefact to be used to give guidance of the proposal of iStar extensions. (RQ5) PRISE was used to propose a new iStar extension to model rational agents (Section 4.3.1). PRISE can be considered valid to support new iStar extensions avoiding problems. The effect of using the process to create new extensions is the high quality of the extensions proposed. (RQ6) PRISE was evaluated and considered useful (Section 4.3.2).

As further research, we intend to keep the catalogue of iStar extensions updated. Thus, we intend to perform an SLR in 2021, five years after the initial SLR. We also intend to support the reasoning approaches in iStar extension. We believe this effort can be useful because various iStar extensions propose reasoning approaches. Finally, the generalisation of the results to other modelling languages could be analysed. It is necessary to perform some adjustment to specificities of each other's modelling language such as a specific catalogue, verification of their specific nodes and links and a list of the experts. We are just performing this effort with the KAOS and UML.

Acknowledgments. We thank CNPQ, CAPES, FACEPE and NOVA LINCS UID/CEC/04516/2019 for the financial support to the execution of this work.

References

- Brambilla, M.; Cabot, J.; Wimmer, M. (2012) *Model-Driven Software Engineering in Practice*. Morgan & Claypool Publishers.
- Caire, P.; Genon, N.; Heymans, P.; Moody, D. (2013) *Visual Notation Design 2.0: Towards User Comprehensible Requirements Engineering Notations*. 21st IEEE International Requirements Engineering Conference (RE), pp 115-124.
- Creswell J (2014) *A concise introduction to mixed methods research*. Sage Publications.
- Dalpiazz, F.; Franch, X.; Horkoff, J. (2016) *iStar 2.0 Language Guide*. arXiv:1605.07767, May 2016, Available in <http://arxiv.org/pdf/1605.07767v1.pdf>

- Franch, X.; Mate, A.; Trujillo, J. C.; Cares, C. (2011) On the Joint Use of i* With Other Modelling Frameworks: A Vision Paper, 19th Requirements Engineering Conference.
- Gonçalves, E.; Araujo, J.; Castro, J. (2018a) Towards Extension Mechanisms in iStar 2.0. 11st International i* Workshop.
- Gonçalves, E.; Castro, J.; Araujo, J.; Heineck, T. (2018b) A Systematic Literature Review of iStar Extensions. *The Journal of Systems and Software*, v. 137.
- Gonçalves, E.; Heineck, T.; Araujo, J.; Castro, J. (2018c) CATIE: A Catalogue of iStar Extensions. *Cadernos do IME. Série Informática*, v. 48, pp. 23-37.
- Gonçalves, E.; Araujo, J.; Castro, J. (2019a) iStar4RationalAgents: Modeling Requirements of Multi-Agent Systems with Rational Agents. 38th Conf. Conceptual Modeling.
- Gonçalves, E.; De Oliveira, M.; Monteiro, I.; Castro, J.; Araujo, J. (2019b) Understanding What is Important in iStar Extension Proposals: The Viewpoint of Researchers, *Requirements Engineering*, v. 24, n. 1, pp. 55-84.
- Gonçalves, E.; Heineck, T.; De Oliveira, L.; Araujo, J.; Castro, J. (2019c) PRISE Tool: A Tool to Support the Proposal of iStar Extensions Based on PRISE, 22nd Workshop on Requirements Engineering.
- Gonçalves, E. (2019) PRISE: A Process to Support iStar Extensions, PhD thesis in computer Science, Universidade Federal de Pernambuco, Brazil.
- Gonçalves, E.; Araujo, J.; Castro, J. (2020) PRISE: A Process to Support iStar Extensions, *The Journal of Systems and Software*, v. 168.
- Horkoff, J.; Aydemir, F.; Cardoso, E.; Li, T.; Maté, A.; Paja, E.; Salnitri, M.; Piras, L.; Mylopoulos, J.; Giorgini, P. (2019) Goal-Oriented Requirements Engineering: An Extended Systematic Mapping Study. *Requirements Engineering*, v. 24, n. 2.
- ITU-T 100 (2018) Specification and Description Language Implementer's Guide.
- ITU-T 119 (2007) Guidelines for UML Profile Design.
- Kitchenham, B., Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information Software Technology* 55, 2049–2075.
- Kitchenham B., Pfleeger S. (2002) Principles of survey research. *Software Eng. Notes*.
- Lindland, O. I., Sindre, G., Solvberg, A. (1994) Understanding Quality in Conceptual Modelling. *IEEE Software*, v. 11, n. 2, pp. 42-49.
- Merriam S. (2009) *Qualitative research: guide to design and implementation*. Jossey.
- Moody, D. (2009) The “Physics” of Notations. *IEEE Trans. on Soft. Engineering*, v. 35.
- Mylopoulos, J., Chung, L., Yu, E. (1999) From Object-oriented to Goal-oriented Requirements Analysis. *Communications of the ACM*, v. 42, pp. 31-37.
- Pfleeger, S. (2003) *Software Engineering: Theory and Practice*. 2th ed., Pearson.
- Russell, S., Norvig, P. (2003) *Artificial Intelligence: A Modern Approach*, Prentice Hall
- Yu, E. (1995) *Modelling Strategic Relationships for Process Reengineering*. PhD. Thesis in Computer Science, University of Toronto, Toronto.