Software and Systems Modeling (2020) 19:763–784
https://doi.org/10.1007/s10270-019-00770-9

REGULAR PAPER

Using empirical studies to mitigate symbol overload in iStar extensions

Enyo Gonçalves[1,2] · Camilo Almendra[1,2] · Miguel Goulão[3] · João Araújo[3] · Jaelson Castro[2]

Received: 1 October 2018 / Revised: 13 November 2019 / Accepted: 27 November 2019 / Published online: 12 December 2019

Abstract
Modelling languages are frequently extended to include new constructs to be used together with the original syntax. New constructs may be proposed by adding textual information, such as UML stereotypes, or by creating new graphical representations. Thus, these new symbols need to be expressive and proposed in a careful way to increase the extension's adoption. A method to create symbols for the original constructs of a modelling language was proposed and has been used to create the symbols when a new modelling language is designed. We argue this method can be used to recommend new symbols for the extension's constructs. However, it is necessary to make some adjustments since the new symbols will be used with the existing constructs of the modelling language original syntax. In this paper, we analyse the usage of this adapted method to propose symbols to mitigate the occurrence of overloaded symbols in the existing iStar extensions. We analysed the existing iStar extensions in an SLR and identified the occurrence of symbol overload among the existing constructs. We identified a set of fifteen overloaded symbols in existing iStar extensions. We used these concepts with symbol overload in a multi-stage experiment that involved users in the visual notation design process. The study involved 262 participants, and its results revealed that most of the new graphical representations were better than those proposed by the extensions, with regard to semantic transparency. Thus, the new representations can be used to mitigate this kind of conflict in iStar extensions. Our results suggest that next extension efforts should consider user-generated notation design techniques in order to increase the semantic transparency.

Keywords Model-based engineering · Semiotic clarity principle · Symbol overload · Experiment · Modelling language extensions · iStar

Communicated by Dr. Manuel Wimmer.

✉ Enyo Gonçalves
enyo@ufc.br

Camilo Almendra
camilo.almendra@ufc.br

Miguel Goulão
mgoul@fct.unl.pt

João Araújo
joao.araujo@fct.unl.pt

Jaelson Castro
jbc@cin.ufpe.br

[1] Universidade Federal do Ceará, Av. José de Freitas Queiroz, 5003, Cedro, CEP 6390 0-0 0 0 Quixadá, CE, Brazil

[2] LER, CIN, Universidade Federal de Pernambuco, Recife, PE, Brazil

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

1 Introduction

According to Brambilla et al. [1], model-based engineering (MBE) is a process in which software models play an important role, but they are not necessarily key artefacts of the development. A typical example that involves the use of MBE is a software development where models are created to document the system, they are a base to development, and no automatic code generation of executable code is involved. In this process, models still play an important role but are not the central artefacts of the development. Models are defined using modelling languages, which specify the constructs graphically in the concrete syntax and their conceptual relations in a metamodel.

Extending a modelling language (ML) is to add new constructs or modify the existing ones [1]. According to the way new concepts are proposed, an extension can be developed using a lightweight or heavyweight strategy [2]. The

lightweight mechanisms are a way of introducing extensions with a little syntactic impact 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 ML.

The abstraction challenge is addressed by providing a general-purpose language that has support for customising the language to a specific application area. Example customisations are profiles (e.g. UML profiles), domain-specific modelling processes and, at a fine-grained level, the use of specialised syntactic forms and constraints on specific modelling elements. The formality challenge can be handled by mapping the ML to a formal language, or annotations can be added to the ML at the meta-model level to constrain properties that should hold between language elements [3].

Modelling languages, such as UML [2], Knowledge Acquisition in Automated Specification (KAOS) [4] and iStar [5], are quite popular. In the case of UML, there is an organisation (Object Management Group) that undertakes the management of language evolution. Technical committees oversee the proposal of newer versions of the language specification, including any possible extension of the language or the establishment of extension mechanisms. The existence of a governing organisation does not deter practitioners and researchers from proposing extensions. However, the broader adoption of features often occurs after the standardisation by the governing organisation. The context of this work and its contributions are towards modelling languages that do have a governing organisation or similar group responsible for its long-term management. That is the case of iStar.

iStar [6] is a goal-based modelling language and reasoning framework which focuses on systems' intentional and social modelling. An iStar model can specify actors, associations among actors, intentional elements, social dependencies and links among intentional elements. iStar is a general-purpose language, and many extensions have been proposed to suit iStar to specific application areas, such as data warehouses [7], autonomic computing systems [8] and legal aspects [9].

Recently, the iStar research community made an effort towards unifying the language notation and establishing a core, named iStar 2.0 [5]. Such standardisation of the language is an important driver for industry acceptance and learning of this language.

Despite the proposition of this new version, the language will continue to be extended. A Systematic Literature Review (SLR) on iStar extensions [10] identified 96 iStar extensions where a great part was composed of new graphical representations. The extensions identified by this SLR were saved in a catalogue of iStar extensions to ease the identification, search and analysis of the existing iStar extensions [11]. On the other hand, the method proposed by Caire et al. [12] has been used to create symbols of a new modelling language constructs. In that work, an alternative iStar symbol set emerged through an experiment.

We adapted Caire et al.'s approach to creating new symbols for extension's constructs in iStar.

Moody's semiotic clarity principle [13] establishes the 1:1 correspondence between construct and graphical symbols. Four kinds of conflicts may occur in this context: symbol deficit, symbol redundancy, symbol overload and symbol excess. Semiotic clarity maximises precision (by eliminating symbol overload) of visual notations. There were 15 overloaded symbols identified in the SLR on iStar extensions [10]. As the iStar language has a history of more than two decades of research and practice, it is not surprising that such symbol overload arises. We analysed the adapted method to mitigate existing overloaded symbols that also can be used by proposers of new extensions to create new extension's symbols.

Extensions may be proposed based on two or more existing extensions, such as the iStar extension presented in [14] to represent dependability analysis which is based on two other extensions: an extension to model deployment [15] and other extension related to the modelling of security requirements via commitments [16]. When the reused extensions have constructs with overloaded symbols, their usage becomes potentially confusing due to the added ambiguity. Another potential situation is model composition [17] of extensions, which can merge two or more models of existing iStar extensions. The symbol overload should be avoided/corrected, and the symbols should be proposed carefully to facilitate the adoption of the iStar extensions by companies.

Motivated by the identification of the symbol overload (when a symbol denotes two or more concepts) [13] in iStar extensions [10], we investigated the usage of the adapted method to recreate 15 symbols. Note that we are not proposing a new iStar extension with these 15 concepts. Instead, we aim at adjusting these symbols to mitigate the occurrence of symbol overload. In this work, we provide new representations for current conflicting iStar extensions. We found that the original symbols proposed in the extensions all performed worse than symbols generated in our experiment. Also, not a single symbolisation technique stood out as significantly better than others. Our results indicate that the adapted method is a good way to propose new graphical representations for the extension of modelling languages. Nevertheless, pragmatic quality [18] (the actual use of the language by its users) is beyond the scope of this paper.

This paper is organised as follows. Section 2 describes the background concepts of iStar, Physics of Notations (PoN) [13] and a summary of identified overloaded symbols in iStar extensions. Section 3 presents the characterisation and results of the experiment to mitigate the symbols overload in iStar extensions. Next, Sect. 4 discusses related work.

Finally, in Sect. 5, we depict conclusions and state directions for future work.

2 Background

This section presents an overview of iStar extensions, the PoN and the problems with graphical representations of iStar extensions.

2.1 iStar extensions

There are different forms to present an iStar extension [10], but all of them introduce new concepts to iStar. Extensions can describe in detail the new concepts and their representations in the iStar metamodel and concrete syntax. For example, the work of Ali et al. [19] proposes an extended iStar framework to allow modelling of contextual goal models. The main new concept proposed is context, a new element that users can associate with other intentional elements in a model. The work provides a whole discussion towards the abstract and concrete syntax of the new constructs. Another example of a well-documented extension is the work of Morandini et al. [20]. The work provides a framework for the engineering of adaptive systems based on a previous framework, providing a thorough discussion of the conceptual model, graphical language and semantics. As an illustration, in this extended framework, a goal (from standard iStar) is endowed with a state to represent its life cycle. This kind of extension describes how the introduction of the new concepts occurred in the language and how to use them.

Other extensions are presented as a method to create models, and the iStar changes are presented through illustrations of the usage of new concepts. Examples of this kind of extensions include the proposals of Guzman et al. [21] and Islam et al. [22]. In the latter, a methodology for security and privacy requirements modelling comprises new types of dependencies for security and privacy requirements.

Some extensions are introduced as part of a case study, or a modelling tool, with a set of new concepts introduced in iStar (see, for example, Gans et al. [23] and Siena et al. [24]). The latter work aims at supporting requirements elicitation in domains articulated by norms. The extension, among other things, brings the concept Norm and the relation Normative Commitment between actors.

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

In previous work [10], we classified the iStar extensions. The results revealed that from those extensions which extended both syntaxes, 77.8% of them are non-conservative.

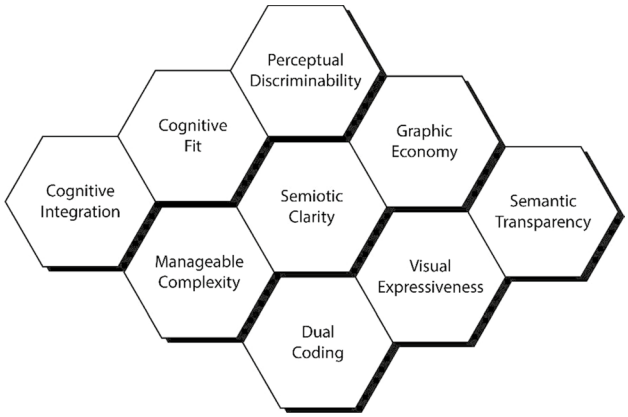


Fig. 1 Nine principles of PoN. (adapted from [13])

Determining whether an extension is conservative (which does not remove any construct of the default syntax) or non-conservative, requires an analysis of the changes introduced both in the abstract and in the concrete syntaxes.

We also classified the extensions as lightweight, heavyweight[1] or both. The results point to 17.7% of extensions that used only lightweight, 38.5% of extensions that used only heavyweight and 43.8% of extensions that used a combination of both.

2.2 The Physics of Notations

The requirements of a notational system constrain the allowable expressions in a language to maximise precision, expressiveness and parsimony, which are desirable design goals for SE notations [13]. Moody proposed a framework with nine principles to construct visual notations in SE [13]. The nine principles are cognitive integration, cognitive fit, perceptual discriminability, manageable complexity, semiotic clarity, graphic economy, dual coding, visual expressiveness and semantic transparency. Figure 1 shows these principles.

The principle of semiotic clarity, based on Goodman's theory of symbols [26], establishes that there should be a 1:1 correspondence between semantic constructs and graphical symbols. This correspondence is necessary to satisfy the requirements of a notational system, as defined in Goodman's theory of symbols. When there is not a 1:1 correspondence, one or more of the following anomalies can occur (Fig. 2):

[1] 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 [25].

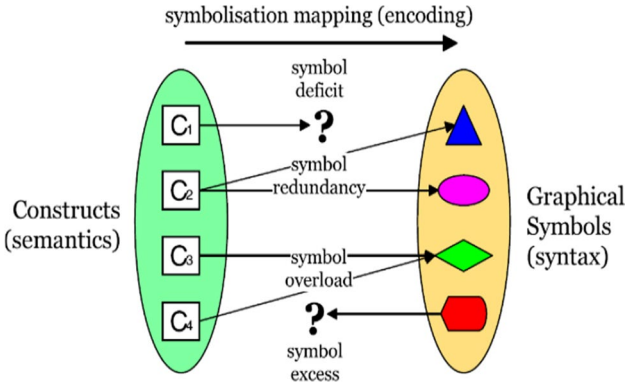


Fig. 2 Principle of semiotic clarity: there should be a 1:1 correspondence between semantic constructs and graphical symbols. (adapted from [13])

- Symbol deficit: when a semantic construct is not represented by any symbol.
- Symbol redundancy: when a semantic construct is represented by multiple symbols.
- Symbol overload: when the same symbol is used to represent multiple constructs.
- Symbol excess: when a symbol does not represent any semantic construct.

Semiotic clarity maximises expressiveness (by eliminating symbol deficit), precision (by eliminating symbol overload) and parsimony (by eliminating symbol redundancy and excess) of visual notations.

The cognitive integration principle analyses the existence of mechanisms to integrate different diagrams of a modelling language to maintain the perceptual integration between them. The cognitive fit principle recommends considering many levels of users' skills in the representation. The principle of perceptual discriminability establishes that different symbols should be distinguishable from each other. Manageable complexity refers to the ability of a visual notation to represent information without overloading the human mind, so it is related to the amount of proposed representation and its organisation in modules and hierarchy. Graphic economy establishes that the number of different graphical symbols should be cognitively manageable. Dual coding that using text and graphics together to convey information is more effective than using either on their own. The visual expressiveness principle defines the degree of variables considered to define the notation, like shape, texture, brightness, size and colour. Finally, the principle of semantic transparency consists of using visual representations whose appearance suggests their meaning.

2.3 Symbol overload

In this section, we present symbol overload conflicts to be targeted by our experiment, which occur when a symbol is used to represent two or more different concepts. These constructs were identified in an SLR of iStar extensions [10]. The categories (kind of conflicts) used in the analysis of this SLR were defined according to the clarity semiotic of the Moody's work [13], which are presented in Sect. 2.2.

We presented the names and meanings of the recreated constructs. The list of symbols and references is presented in Table 1.

Norm (T1) is a construct proposed as a means for communicating standards of behaviour, which acts as an abstraction for any deontic prescriptions (such as laws and regulations). Duty (T2) is a kind of norm that represents an obligation to be performed. A predicate (T3) is part of a statement (composed of subject, predicate and object) applied to intentional elements. Actions (T4) are tasks performed by agents whose norms address. Norm, duty, predicates and actions are represented in these referenced papers by a triangle.

Security and vulnerability restrictions (O1) can be defined as constraints of security/vulnerability applied to intentional elements. Plan (O2) is a sequence of actions/tasks to be performed to reach a goal. Security and vulnerability restrictions and plan are represented as an octagon in these papers.

A double-headed arrow is used to represent a new link indicating that an actor has a security/privacy property (D1), and it is also used as a satisfaction link (D2), representing a way to satisfy a security/vulnerability restriction.

A parallelogram is used to represent service (P1) in the context of the service-oriented architecture (SOA) where companies offer software services. Fact (P2) is defined as verifiable on monitorable data, a fact truth value requires monitoring some characteristics and history of a set of relevant environment elements.

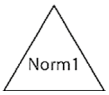
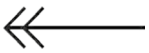
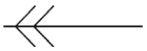
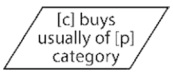
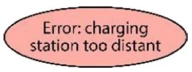
The marker ✓ (M1) has been used to represent the status of an intentional element [27]. It was also used to represent capability (M2). An ellipse is used to represent error (E1) and vulnerability (E2). A rectangle is used to represent fact (R1) and cause and effect (R2).

3 Mitigating symbol overload in iStar extensions

This section presents the design, execution, results and discussion of a multi-stage experiment to mitigate symbol overload conflict found in existing iStar extensions.

Symbol overload, found in the literature (see Sect. 2.3) during an analysis of existing iStar extensions, involved 15 constructs: Action, Capability, Cause, Duty, Effect, Error, Fact, Norm, Plan, Predicate, Satisfaction Relationship, Security/

Table 1 Graphical representations of constructs with symbol overload

Form	ID	Concept	Symbol	References
Triangle	T1	Norm		[24]
	T2	Duty		[41]
	T3	Predicate		[42]
	T4	Action		[43]
Octagon	O1	Security and Vulnerability Restriction		[44] [45]
	O2	Plan		[46]
Double-headed arrow	D1	Security/Privacy Relationship		[45]
	D2	Satisfaction Relationship		[44]
Parallelogram	P1	Service		[47]
	P2	Fact		[48]
Marker ✓	M1	Capability		[49]
	M2	Status of an intentional element		[27]
Ellipse	E1	Error		[20]
	E2	Vulnerability		[45]
Rectangle	R1	Fact		[50]
	R2	Cause		[51]
	R3	Effect		[51]

Privacy Relationship, Security and Vulnerability Restrictions, Service, Vulnerability. Symbol overload can be mitigated by designing and evaluating new, alternative graphical representations. We do not create a new iStar extension with these concepts but adjusting their symbols to mitigate the occurrence of symbol overload in existing iStar extensions.

We adapted the experimental design reported by the Caire et al. method [12], in which core iStar constructs' concrete syntax was evaluated against new possible alternatives. In our study, we started with proposed extensions in the literature and compared them with new symbols elaborated using expert and user-based notation design techniques. We aim

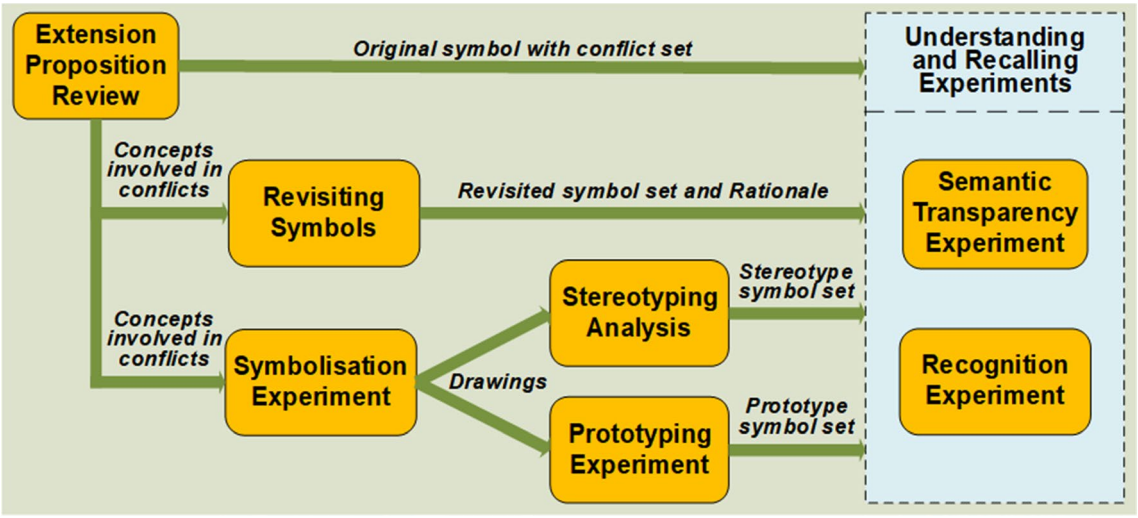


Fig. 3 Research design adapted from [12]

to evaluate those extensions in terms of understanding and recalling, focusing on the semantic transparency principle of PoN. By doing so, we expect to propose preferred notations for conflicting concepts, as well as to suggest better methodological procedures for future language extensions works.

3.1 Study design

Caire et al. [12] proposed an approach to creating symbols to the standard syntax of a modelling language. We believe that there are some additional observations to be considered during the proposal of extensions of existing modelling languages. Thus, we adapted this method to propose new graphical representations to the proposal of constructs of an iStar extension. We included the following changes:

(1) We trained the participants in iStar and the domain of the 15 concepts during all steps of the experiment;
(2) We introduced restrictions to the creation of symbols during the revisiting symbols and symbolisation experiment. We introduced two recommendations:

   (i) Create abstract symbols easy to be drawn in a paper without a tool. It was a recommendation of the experts in iStar extensions interviewed in the context of a qualitative study to understand how the iStar extensions have been proposed and what could be done to improve the next proposals [20];
   (ii) Avoid using the representation of symbols of the standard iStar and the symbols of the existing iStar extensions not to produce new symbol overloads (we presented a list of these symbols to be avoided).

Four groups of alternative symbols were used during the semantic transparency and recognition experiments: (i) the Original group represents the symbols proposed by the papers that describe the extensions, part of them has different representations in different extensions; and (ii) three representations developed using different techniques: PON + R, Stereotype, Prototype. Then, for each concept, there will be at least four representations to be compared. Figure 3 illustrates the steps of this part of the study.

The steps are summarised below and detailed in Sect. 3.2 to 3.8.

Extension Propositions Review This step was conducted in previous research [10] and found constructs with symbol overload and its related concepts. These symbols composed the Original symbol set.

Revisiting symbols For each concept with symbol overload, specialists created alternative representations based on PON principles and the restrictions presented at the beginning of this subsection (i.e. create abstract symbols and avoid using the existing representations). These symbols composed the PON + R (Physics of Notations + Restrictions) symbol set. For each new representation, a design rationale was registered.

Symbolisation experiment (Study 1) This was a preliminary experiment with naïve participants to draw symbols for concepts with symbol overload. These drawings were used in the further steps of the study.

Stereotyping analysis (Study 2) We identified the most common symbols produced in the previous experiment, for each concept with symbol overload. The most frequent symbols composed the Stereotype symbol set.

Prototyping experiment (Study 3) Naïve participants ranked the "best" representations for each concept with

symbol overload. The best-ranked symbols composed the Prototype symbol set.

Semantic transparency experiment (Study 4) We evaluated the ability of naïve participants to infer the meanings of the symbols from all sets.

Recognition experiment (Study 5) We evaluated the ability of naïve participants to learn and remember symbols from all sets.

We highlight that each participant participated in only one step above.

In Sects. 3.2 to 3.7, we describe the application of the methodology and the results for each step. In Sect. 3.8, we analyse data from the last two experiments and recommend symbols to mitigate the symbol overload. Finally, in Sect. 3.9, we analyse the notation design techniques and provide some recommendations for future language extension proposers.

This study was conducted at Universidade Federal do Ceará – Campus Quixadá, in Brazil northeast region. We involved two professors, a designer and undergraduate students from various undergraduate programs (Computer Science, Software Engineering, Information Systems, Digital Design and Computer Engineering). The study occurred between September and December of 2016. Clarification and consent terms were prepared and sent to the participants for each step of the study.

The students were from second to the third year; all of them had courses on system analysis and modelling. Besides, all students and the designer were short trained (about 1 h long) in the iStar modelling language and the domains and application areas related to the concepts under investigation. This training was important to set a basic understanding of iStar fundamental constructs and purpose (goal modelling), which is something one needs to be aware of when proposing or evaluating extensions for the language.

According to Tichy [28], situations where using students as surrogates for professionals are acceptable include the following:

(a) Student subjects have been trained sufficiently well to perform the tasks asked of them. They must not be overwhelmed by the complexity of or unfamiliarity with the tasks or domain. One can only study behaviour that will occur;
(b) Student subjects are used to establish a trend. Say a study compares two methods to see which one is better. If one method has a clear relative advantage over the other with student subjects, then one can make the argument that there will be a difference in the same direction (although, perhaps, of different magnitude) for professionals, provided the professionals similarly use the methods;
(c) Student subjects are used to eliminate alternate hypotheses. Suppose an experiment with student subjects that shows no clear difference between two alternative methods. Unless there is evidence of radically different approaches by professionals, then it is highly unlikely that a noticeable effect will magically appear among professionals. It will also be nearly impossible to find professionals to participate in a follow-up experiment in this case and, therefore, allow negative results with student subjects to be published and thereby help the community discard wrong assumptions and move on;
(d) Studies with students are a prerequisite for getting professionals to participate. It is hard to overemphasise this point: experiments must be tested and debugged with students before running them with professionals. The experimental design and the trends found may be worth publishing, even if the follow-up experiment is the "real thing".

Granada et al. [29] performed an experiment to choose new symbols to a WebML visual notation. The authors used students and professionals (experts). Their results show similar performance for both kinds of participants.

We believe that students are the main potential group of users of the iStar extensions. Thus, we selected participants with this profile.

3.2 Extension propositions review

Two or more different concepts have been represented by the same graphical symbol in 15 cases in our study. A detailed description of these overloaded symbols is presented in Sect. 2.3. In summary, we have the following representations presented in Fig. 4. We highlight that Fact, Plan, Service and Vulnerability concepts have more than one representation. Note that there are concepts with two or more representations presented in Fig. 4. This conflict is due to the proposal of different symbols in the iStar extensions. These constructs have both overload and redundancy.

3.3 Revisiting symbols

This part of the experiment consists of the proposal of new graphical representations following the principles of the Physics of Notations (PoN). Starting from the concepts and its domain, we proposed a revised symbol set based on principles of the PoN. Explicit design rationale was provided for each symbol. We refer to this symbol set as PON + R (Physics of Notations + Restrictions) symbol set for the remainder of the paper.

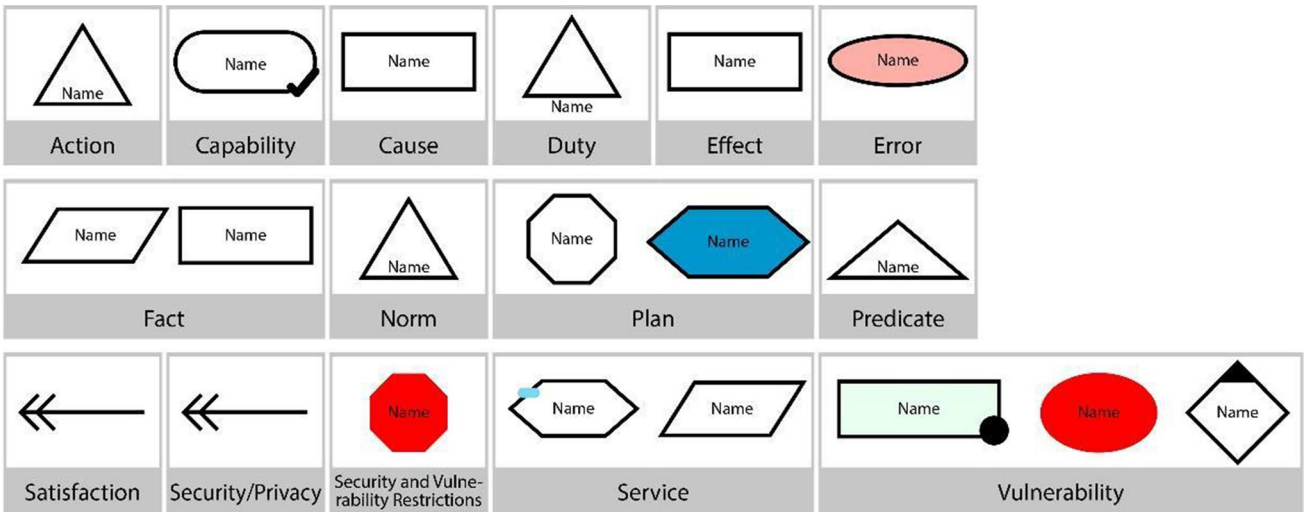
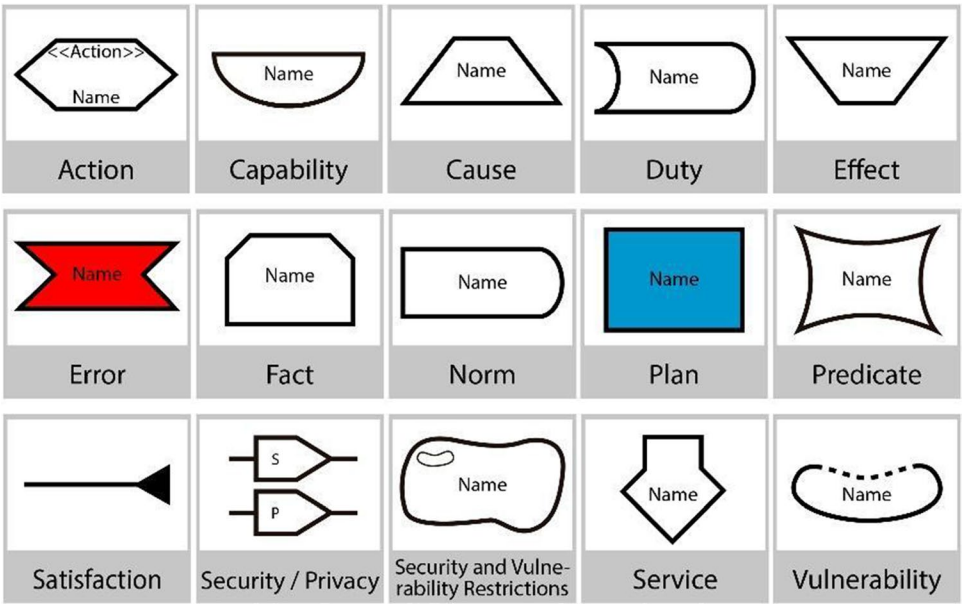


Fig. 4 Original symbol set

Fig. 5 PON + R symbol set



Tree participants conducted this proposal: two professors (the first two authors) and a designer from Universidade Federal do Ceará – Campus Quixadá. The final set was chosen through a consensus among the participants. First, the participants proposed a symbol set individually and provided a written rationale for each choice. Second, in a meeting, these symbol sets and rationale were then shown to each other. Then, they would vote in their final choice (not necessarily her drawing), until a symbol received two or three votes. Additionally, we followed a good practice identified in a previous qualitative study [20], which states that graphical representations should be simple to be hand-drawn and preferably shaped as abstract figures for maintaining the consistency with the iStar default representation. Figure 5 shows this result.

3.4 Symbolisation experiment

In this experiment, we asked non-experienced participants to generate symbols for the 15 concepts with symbol overload. We followed the same steps used by Caire et al. [12] which used the sign production technique, developed by Howell and Fuchs [30], to recreate the iStar default symbols. This experiment involves asking members of the target audience to generate symbols to represent concepts.

3.4.1 Participants

There were 98 participants (30 females and 68 males), all undergraduate students in Digital Design (23), Software Engineering (30) and Information Systems (45) courses of Universidade Federal do Ceará – Campus Quixadá. They had no previous knowledge on iStar extensions until the training provided. We chose students from informatics because they present a similar cognitive profile: they had courses related to application areas such as security and artificial intelligence. Digital Design students had training focused on drawing, art and communication as well as user interface design, Software Engineering students had training focused on software production and a more in-depth view about requirements engineering, and Information Systems students have a broader knowledge on the application areas involved in the conflicting concepts, as well as software production.

3.4.2 Materials

We used drawing pads, pencils and erasers as materials for drawing the symbols. The drawing pad comprises the first page with a demographic questionnaire, and other 15 pages for sketching symbols, each presenting a concept, its description and a space for drawing.

3.4.3 Procedure

We started with a short training in iStar language, and a discussion on the concepts to be graphically represented. We also gave the participants a list of the symbols introduced by the iStar extensions, and we instructed them not to use these symbols.

To avoid the generation of new symbols that conflict with core iStar constructs and already proposed extensions, we provided a handout with these two sets of notations and instructed them not to use similar constructs. This constraint avoids multiple conflicts among the symbol set. We recommended participants to prefer simple drawings, the same recommendation used in the previous revisiting symbols step.

We started with a short training in iStar language, and a discussion on the 15 concepts from those participants would create symbols. The training aimed at presenting the purpose of the iStar modelling language and its core elements. The discussion of concepts intended to declare the meaning and to show the domain involved briefly.

We instructed the participants to not reuse symbols from the iStar core symbols and the Original set (Fig. 5). We provided handouts with these two sets of notations. Such restrictions did not hinder the creativity of the participants, as there is a vast space of graphical forms participants could use. Indeed, this recommendation helped to avoid the generation of new overloaded symbols. This multi-stage experimentation aimed to generate alternatives for previously proposed symbols. There is no point in creating symbols with the same symbols of the core constructs of the modelling language. Also, there is no need to generate symbols similar to the ones in the Original set, as this set is going to be further included in the understanding and recalling experiments (Fig. 4). We also recommended participants to prefer simple drawings, the same recommendation used in the previous revisiting symbols step. We then asked each participant to draw a symbol to represent each of the 15 concepts.

3.4.4 Results

The 98 participants produced a total of 1417 drawings (response rate of 96.4%—1417 out of 1470), which we consider a very good response rate. The concepts with the absence of responses were Security and Vulnerability restrictions (23.5%—23 absences out of 98), Security and Privacy relationships (17.8%—17/98), Predicate (12.2%—4/98), Satisfaction (3%—3/98), Vulnerability (3%—3/98), Action (2%—2/98), Service (1%—1/98).

3.5 Stereotyping analysis

In this step, we analysed the sets produced in the previous step and identified the most common drawings produced for each concept. Such classification defines the population stereotype or mode drawing. The rationale for doing this is that the representation most commonly produced should also be the most frequently recognised as representing that concept by members of the target audience [30, 31].

3.5.1 Participants

The first two authors conducted this analysis. The stereotyping procedure comprises objective tasks related to classification and ranking of drawings.

3.5.2 Procedure

We used the same method followed in [12] to identify stereotypes, i.e. the judges' ranking method [31]. The first and second authors individually classified the drawings in groups based on their visual and conceptual similarity. We reviewed the grouping of the drawings. We selected the most representative drawings, i.e. the ones that are more similar to the rest of the drawings of the group. Finally, for each concept, we ranked the groups with the highest number of drawings (the stereotypical group), resulting in 15 stereotypical drawings.

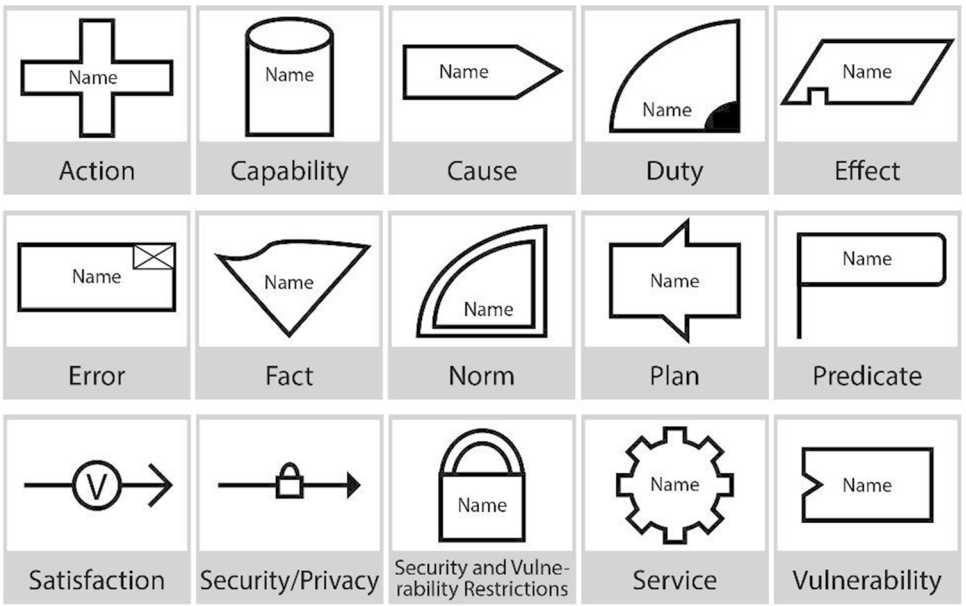


Fig. 6 Stereotype symbol set

3.5.3 Results

We classified the drawings, and the main result of this step is a set of 15 stereotypical drawings, one per target concept (Fig. 6).

3.6 Prototyping experiment

In this experiment, non-experienced participants analysed the drawings created in the symbolisation experiment and chose which best represents each related concept. The population prototype produced the most frequently chosen drawings [31].

3.6.1 Participants

There were 34 non-experienced participants in this experiment, all students in the courses of Digital Design, Software Engineering, Information Systems and Computer Sciences. We used different participants from those in Study 1 but drawn from the same underlying population. The usage of different participants prevented the bias that would result in having the authors of the drawings selecting which would be best suited for each construct.

3.6.2 Procedure

We gave iStar training to the participants and provided them with the link of the electronic form so that they could choose the most suitable representation for each construct. We used LimeSurvey (www.limesurvey.org) as a tool. Firstly, the participants entered their demographic data. Secondly, they navigated through 15 screens, one for each extension concept. The name and definition of the concept were displayed at the top of the screen with the candidate drawings below: radio buttons were provided to select the best representation. Participants devised a total of 1417 drawings. We selected a representative drawing from each category identified in the Stereotyping experiment rather than using all drawings from the symbolisation experiment. Participants were asked to identify the drawing that most effectively conveyed each concept, irrespectively of their artistic quality. Both the order of the screens (concepts) and the position of the drawings on each screen were randomised to counteract sequence effects. No time limit was set, but subjects took on average 10 min and 53 s to complete the task.

3.6.3 Results

The result of this experiment was a set of 15 prototypical drawings, one per extension concept (Fig. 7).

The drawings to Capability, Error, Predicate, Security/Privacy Relationship and Security and Vulnerability Restrictions (5 concepts) were the same in prototype and stereotype choices.

3.7 Semantic transparency experiment

In this study, we evaluated the capability of participants to infer the meanings of given symbols. We followed a blind interpretation study (also named comprehension test [32, 33] or recognition test [30]). The comprehensibility of the

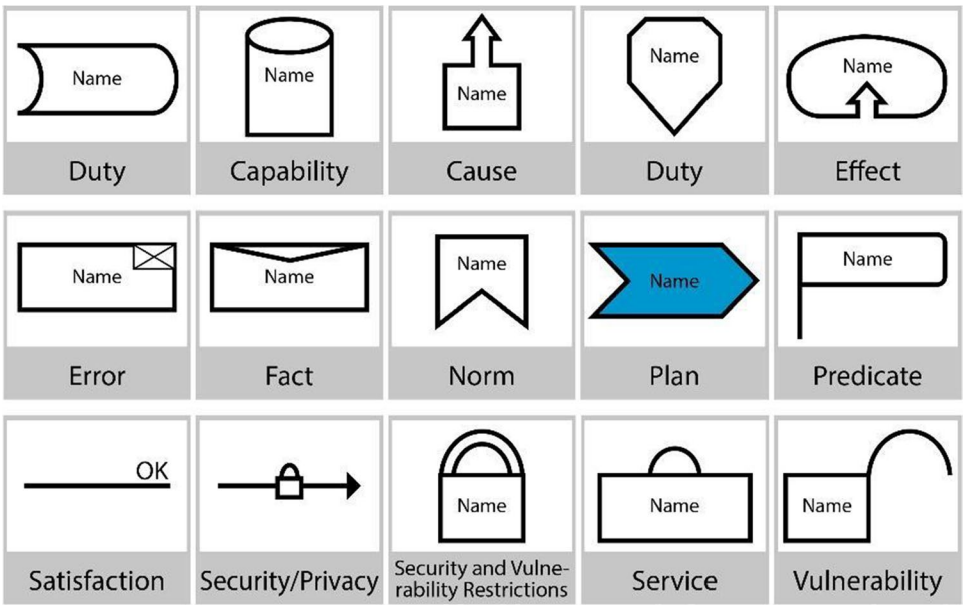


Fig. 7 Prototype symbol set

symbol is typically measured by the percentage of correct responses (hit rate).

3.7.1 Participants

In previous research, semantic transparency has almost always been evaluated by experts, who are poorly qualified to do this: the definition of semantic transparency refers to "novice readers" [13], which they most certainly are not. For this reason, we recruited naïve participants for this experiment. There were 64 participants, undergraduate students in Digital Design, Computer Science, Software Engineering and Information Systems at Universidade Federal do Ceará – Campus Quixadá. We used different participants than those previous studies but drawn from the same underlying population.

3.7.2 Experimental design

Each of the four symbol sets (Original, PON + R, Stereotype, Prototype) represents a notation design technique. We aimed to evaluate which representation for each concept had better comprehension rates, and additionally, we compared the techniques with each other. Participants were randomly distributed in experimental groups, as indicated in Table 2.

3.7.3 Materials

An electronic, web-based questionnaire was used to collect participants responses. For each experimental group, a version of the questionnaire was produced, sharing the same structure. The first page was used to ask the screening question and collect demographic data. In the remaining pages, each of the 15 symbols was displayed at the top of the page (the stimulus) and the list of 15 concepts and definitions displayed as alternatives (the candidate responses). An exception is the group of Original representations which has an additional five symbols. These extra symbols were needed because the literature review found three concepts (Fact, Plan, Service) with two extensions and one concept (Vulnerability) with three extensions.

We used LimeSurvey as a survey tool. Participants should indicate which concept most likely corresponded to the symbol. Both the order of appearance of symbols and the order of listing of alternatives were randomised to counteract sequence effects.

3.7.4 Procedure

Participants were randomly assigned to experimental groups and provided with a link to the online form. They were instructed to work alone and not discuss their responses with other participants. No time limit was set, but participants took on average 10 min and 50 s to complete the task. We only considered fully answered questionnaires. Thus, we discarded five answers.

Table 2 Factors and sample sizes for semantic transparency experiment

Factors	Original	PON + R	Stereotype	Prototype
Sample size	$n = 16$	$n = 16$	$n = 17$	$n = 15$

Table 3 Hit rate (in %) means per concept per factor in semantic transparency experiment

Concepts	Factors			
	Original	PON + R	Stereotype	Prototype
Action	0.0	**68.8**	0.0	6.7
Capability	6.3	25.0	**35.3**	26.7
Cause	12.5	0.0	5.9	**13.3**
Duty	0.0	6.3	**11.8**	0.0
Effect	**12.5**	0.0	11.8	6.7
Error	0.0	**37.5**	29.4	26.7
Fact	**12.5**/6.3	6.3	5.9	0.0
Norm	18.8	12.5	0.0	**26.7**
Plan	5.3/**25.0**	0.0	0.0	6.7
Predicate	0.0	0.0	5.9	**6.7**
Satisfaction Relationship	6.3	18.8	11.8	**26.7**
Security/Privacy Relationship	12.5	25.0	**64.7**	53.3
Security and Vulnerability Restrictions	18.8	6.3	**35.3**	33.3
Service	25.0/6.3	0.0	**47.1**	6.7
Vulnerability	6.3/12.5/0.0	0.0	0.0	**53.3**
Mean	9.4	13.8	17.6	19.6
Standard deviation	7.8	18.6	19.3	16.9
Group size	16	16	17	15

Bolded values indicate the best hit rate of a concept among all factors

3.7.5 Results

The traditional way of measuring comprehensibility of graphical symbols (ISO, 2007) and (ISO, 2003) is by measuring hit rates (percentage of correct responses). The results of this analysis are presented in Table 3.

The Hit Rate Grand Mean value was 14.6%, very low compared to ISO threshold of 67% for comprehensibility. Only one symbol out of 65 reached the ISO threshold. Most of the symbols (49 out of 65) had low hit hate (< 20%), and the means are below 20% for each technique.

Comparing to Caire's study of iStar original symbols, we observed lower hit rates. It is important to note that the extensions' concepts were narrow, specific constructs, in contrast with iStar original basic, general constructs. We think that the proposal of abstract figures is important to maintain consistency with the iStar default symbols, but this recommendation impacts on the semantic transparency of the symbols.

Literature originated symbols had the worst performance, only for three concepts standard symbols had a better hit rate than alternatives. Also, similarly to Caire's study, the Stereotype group had the best performance.

We measured the semantic transparency coefficient of the symbols. This coefficient was proposed in [12] and is a scale from − 1 to + 1: it can be negative for symbols whose appearance implies an incorrect meaning (semantically perverse), and it can be close to zero for symbols which are semantically opaque and positive for semantically transparent symbols. A symbol's semantic transparency coefficient is calculated using the following formula [12]:

$$\frac{\text{Highest frequency} - \text{expected frequency}}{\text{Total responses} - \text{expected frequency}}$$

Expected Frequency is the number of responses expected by chance (= n/s, where n is the number of participants in the group and s is the number of symbols). Highest Frequency is the number of responses of the most voted concept, and it can be positive or negative. If the most voted is the target concept, it is given a positive signal, else if it is a distractor concept is given a negative signal. Total Responses is the number of participants. The semantic transparency coefficients for all symbols are shown in Table 4. No single factor stood out with all positive values or higher values from all others. All factors had negative means; for two of them (Stereotype and Prototype) the mean was close to zero (semantically opaque). In a per concept analysis, from the ten concepts that had at least one symbol semantically transparent (positive coefficient), only two were from the literature (Original factor).

As mentioned previously, the extensions' concepts that we investigated are all narrow, specific constructs. The use of abstract figures (simple shapes) may hinder the semantic transparency, what may be an explanation for the absence of factors with semantic transparency coefficient significantly above zero.

Table 4 Semantic transparency coefficient results

Concepts	Factors			
	Original	PON + R	Stereotype	Prototype
Action	− 0.27	**0.67**	− 0.39	− 0.21
Capability	− 0.54	**0.20**	**0.31**	**0.21**
Cause	− 0.34	− 0.34	− 0.26	− 0.29
Duty	− 0.41	− 0.34	− 0.20	− 0.43
Effect	− 0.27	− 0.47	− 0.32	− 0.29
Error	− 0.27	**0.33**	**0.24**	**0.21**
Fact	− 0.34/− 0.34	− 0.34	− 0.20	− 0.50
Norm	− 0.27	− 0.34	− 0.26	**0.21**
Plan	− 0.34/**0.20**	− 0.27	− 0.26	− 0.36
Predicate	− 0.34	− 0.34	− 0.32	− 0.29
Satisfaction Relationship	− 0.27	− 0.27	− 0.32	**0.21**
Security/Privacy Relationship	− 0.34	− 0.34	**0.62**	**0.50**
Security and Vulnerability Restrictions	− 0.61	− 0.41	**0.31**	− 0.43
Service	**0.20**/− 0.41	− 0.27	**0.43**	− 0.43
Vulnerability	− 0.27/− 0.41/− 0.27	− 0.27	− 0.32	**0.50**
Mean	− 0.30	− 0.29	− 0.06	− 0.09
Standard deviation	0.19	0.26	0.34	0.36
≠ 0 (one-sample *t* test)	**Perverse** ($p = 0.000$)	**Perverse** ($p = 0.041$)	**Opaque** ($p = 0.485$)	**Opaque** ($p = 0.331$)

Bolded values indicate the positive, and underscored values indicate the highest

3.8 Recognition experiment

This experiment evaluates participants' ability to learn and remember symbols from the different symbol sets. Participants were given one of the symbol sets to learn and then had to recall their meanings: this represents a recognition task. This experiment also allows us to evaluate the effect of semantic transparency on cognitive effectiveness, as recognition performance provides a measure of cognitive effectiveness.

3.8.1 Participants

There were 66 participants in this experiment, undergraduate students of Digital Design, Computer Science, Software Engineering and Information Systems at Universidade Federal do Ceará – Campus Quixadá. The participants of this step did not participate in previous steps of this work.

3.8.2 Experimental design

Five groups of symbols were used, and the groups of symbols were the same as in Study 4 with one additional group: PON + R with an explanation (PON + R Explained). The PON + R Explained group comprises the same drawings from PON + R group generated in the revisiting symbols step, plus an explanation for the design of each symbol.

3.8.3 Materials

We prepared the training material and the testing material. The training material contained the name of the concept, a description and its graphical representation. This material was used before the start of the test. The testing material was a questionnaire with the graphical representations, and the participants should choose the name of the concept related to the graphical representation shown. We used LimeSurvey to apply this part of the experiment.

3.8.4 Procedure

Participants were instructed to study the training materials until they understood all symbols and their meanings (learning phase). They then proceeded to the testing phase, where symbols were presented one per page, and participants had to identify the corresponding concept. Participants were not allowed to take notes during the learning phase or back to the training materials during the testing phase. No time limit was set, but subjects took on average 6 min and 35 s to complete the task.

Table 5 Hit rate (in %) means per concept per factor in recognition experiment

Concepts	Factors				
	Original	PON + R	Stereotype	Prototype	PON + R Explained
Action	23.1	**100.0**	76.9	50.0	91.7
Capability	0.0	58.3	69.2	62.5	**66.7**
Cause	7.7	**75.0**	53.8	43.8	41.7
Duty	0.0	58.3	**61.5**	37.5	58.3
Effect	0.0	66.7	38.5	**68.8**	58.3
Error	76.9	**100.0**	92.3	81.3	83.3
Fact	7.7/15.4	58.3	46.2	31.3	33.3
Norm	7.7	66.7	**84.6**	56.3	50.0
Plan	30.8/0.0	66.7	61.5	**75.0**	58.3
Predicate	15.4	66.7	53.8	**62.5**	50.0
Satisfaction Relationship	7.7	**91.7**	53.8	81.3	83.3
Security/Privacy Relationship	38.5	83.3	69.2	**87.5**	83.3
Security and Vulnerability Restrictions	61.5	66.7	**92.3**	81.3	75.0
Service	0.0/0.0	58.3	61.5	43.8	**66.7**
Vulnerability	69.2/61.5/76.9	**91.7**	61.5	81.3	66.7
Mean	25.0	73.9	65.1	62.9	64.4
Standard deviation	28.4	15.4	15.9	18.4	16.8
Group size	13	12	13	16	12

Bolded values indicate the best hit rate of a concept among all factors

3.8.5 Results

We also used hit rates (percentage of correct responses) to measure this step. The results of this analysis are presented in Table 5.

The Hit Rate Grand Mean was 56.2% with four groups above 60%. Only 12 (out of 65) symbols performed below 20%, all of them of the Original set. Comparing to Caire's study, we also observed higher hit rates than the comprehensibility study.

It is possible to identify that the percentage of correct responses is significantly larger than the originally proposed constructors for the concepts analysed. We highlighted the graphical representation with the highest score for each concept. The text explaining the meaning of the graphical representation could improve the hit rate in four cases. The hypothesis test about the results of this experiment is presented in Sect. 3.9.

3.9 Comparing notation design techniques

The symbol sets used in the previous experiments were generated by specialists (Original and PON + R) or by users (Stereotype and Prototype). Although the literature review is not a design method per se, we think it aligns accurately with the other three techniques in the sense that it can be used as an approach to identify and select graphical representations for concepts. Caire et al. [12] reported that symbol sets developed through symbolisation experiments (e.g. PON + R, Stereotype and Prototype) performed better than those originally proposed in the literature, in the semantic transparency and recognition studies. They also provided a partial order for hit rate performance in semantic transparency study (*Stereotype* > *Prototype* = PON + R > *Original*). Caire et al. did not provide a total or partial order based on recognition study data, although they indicated that the Original notation performed worse than all others concerning hit rates.

For our study, we assumed an exploratory point of view and established equality hypotheses. Through the analysis, rejection of proportion equality hypothesis triggered additional inequality hypothesis formulation in order to confirm if differences were significant. Initial main hypotheses were established for each study:

$$H_{1\,(\text{Semantic Transparency})}: \text{Original} = \text{Revisited} = \text{Stereotype} = \text{Prototype}$$

$$H_{2\,(\text{Recognition})}: \text{Original} = \text{Revisited} = \text{Stereotype} = \text{Prototype} = \text{Revisited Explained} \quad ()()()$$

We performed a test for normality of data from Study 4 and Study 5 (symbol's hit rates in Tables 3 and 5) using Shapiro–Wilk's test. For both studies, we reject the assumption of the normal distribution of hit rate means. The results for Study 4 and 5 are summarised in the box plot in Fig. 8.

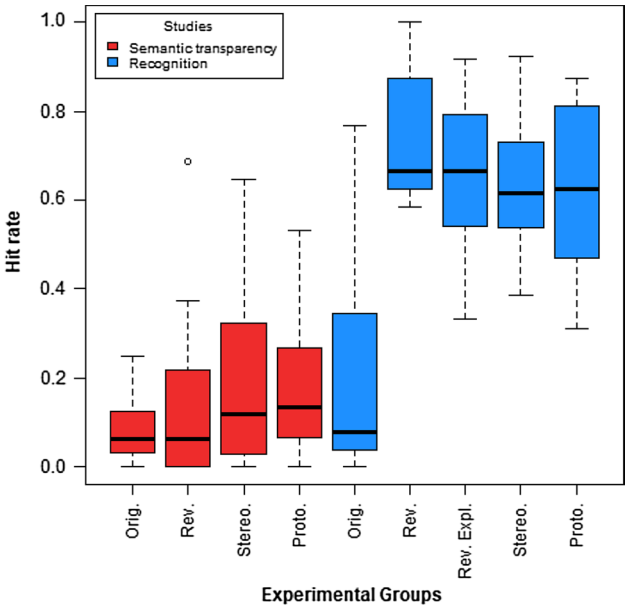


Fig. 8 Box plot of the responses in semantic transparency and recognition experiments

By analysing this combined box plot, we noticed that the performance of all notation techniques in Study 4 (semantic transparency) is very low, with a hit rate under 20% for the majority of symbols. However, these numbers are similar to those achieved by standard symbols of iStar in Caire's study. In Caire's study, experts and users were told no restrictions upon the shape or overall guidelines for symbol creation. In our experiment, we recommended the use of simple drawings, as this is a general recommendation from iStar experts [10].

In Study 5 (Recognition), the Original set had a very low performance of hit rate compared to the other experimental groups, although the upper whisker ranges from 30% up to 80%. The other four groups in Study 5 had similar performances in terms of both median and distribution. From our results, we do not recommend a single technique as the best. We recommend that extension proposers perform (if possible) all these techniques to generate candidate symbols. In Sect. 3.10, we discuss how to choose a symbol for a concept based on the results of the multi-stage experiment.

Some hypotheses testing analysis is discussed as follows. We used nonparametric Pearson's Chi-square test for differences in the proportion of hit rate among groups. In Study 4, the semantic transparency' hit rate (H1) did not differ by notation design approach, χ^2 ($df = 3$, $N = 960$) = 4.79, $p = 0.188$.

In Study 5, the recognition' hit rate (H2) did differ by notation design approach, χ^2 ($df = 4$, $N = 990$) = 76.57, $p = 0.000$. Thus, for Study 5, we performed further testing with pairs of approaches that are described in Table 6.

Table 6 All pairs of approaches for two-sided for testing differences in hit rate (recognition experiment—H2)

Null hypothesis	p value	χ^2	N
Original = PON + R	0.000	60.16	375
Original = Stereotype	0.000	38.17	390
Original = Prototype	0.000	36.49	435
Original = PON + R Explained	0.000	35.05	375
PON + R = Stereotype	0.084	2.98	375
PON + R = Prototype	0.023	5.17	420
PON + R = PON + R Explained	0.068	3.33	360
Stereotype = Prototype	0.706	0.14	435
Stereotype = PON + R Explained	0.976	0.00	375
Prototype = PON + R Explained	0.826	0.05	420

1 degree of freedom for two-sample test

The pair-by-pair comparison pointed out some differences. Table 7 shows all pairs for which differences were confirmed. All other pairs did not differ significantly. Additionally, we calculated the effect size of the pairs using Cohen's h statistic.

Original notations had the worst performance, which builds upon findings from [12]. However, we did not observe better hit rates for user-generated notations than expert generated, and the PON + R approach had a significantly better hit rate compared to Prototype.

Interestingly, if we test the recognition's hit rate of PON + R, Stereotype, Prototype and PON + R Explained (omitting the Original group), they did not differ, χ^2 ($df = 3$, $N = 795$) = 6.27, $p = 0.099$.

The Cohen's h test results pointed out that there is a small effect for the $PON + R > Prototype$ pair. Thus, we can consider them without a relevant difference. For the other pairs, we identified the effect size large ($PON + R > Original$) and medium ($Stereotype > Original$, $PON + R\ Explained > Original$ and $Prototype > Original$). Therefore, we can consider that there are relevant differences.

3.10 Choosing the symbols to represent the concepts

In this work, we conducted a multi-stage experiment to a set of 15 concepts that need new symbols, because of overloading symbols in the proposed extensions. The entire experiment could have a single concept as the sole experimental object. However, it would be costly and lengthy to run 15 multi-stage experiment executions. The 15 concepts pervade many domains, so there is no expectation of reaching a set of best symbols to be adopted by iStar community readily. Therefore, the objective of the following analysis is to answer which symbol would be the better choice for each concept, based on the empirical data gathered in the

Table 7 Pairs of approaches with significant differences in hit rate (recognition experiment)

Alternative hypothesis	*p* value	χ^2	*N*	Cohen's h	Effect size
PON + R > original	0.000	60.16	375	0.84	Large
PON + R > prototype	0.012	5.17	420	0.24	Small
Stereotype > original	0.000	38.17	390	0.65	Medium
PON + R Explained > original	0.000	35.05	375	0.63	Medium
Prototype > original	0.000	36.49	435	0.60	Medium

1 degree of freedom for two-sample test

Table 8 Hit rate of combined results

Concept	Notation	Hit rate*
Action	PON + R	0.821
Capacity	Stereotype	0.500
Cause	PON + R	0.321
Effect	Prototype	0.387
Error	PON + R	0.643
Fact	PON + R	0.286
Norm	Prototype	0.419
Duty	Stereotype	0.333
Plan	Prototype	0.419
Predicate	Prototype	0.355
Security and Privacy Relationship	Prototype	0.710
Security and Vulnerability Restriction	Stereotype	0.600
Satisfaction	Prototype	0.548
Service	Stereotype	0.533
Vulnerability	Prototype	0.677

*Combined hit rate from both studies. PON + R Explained not included

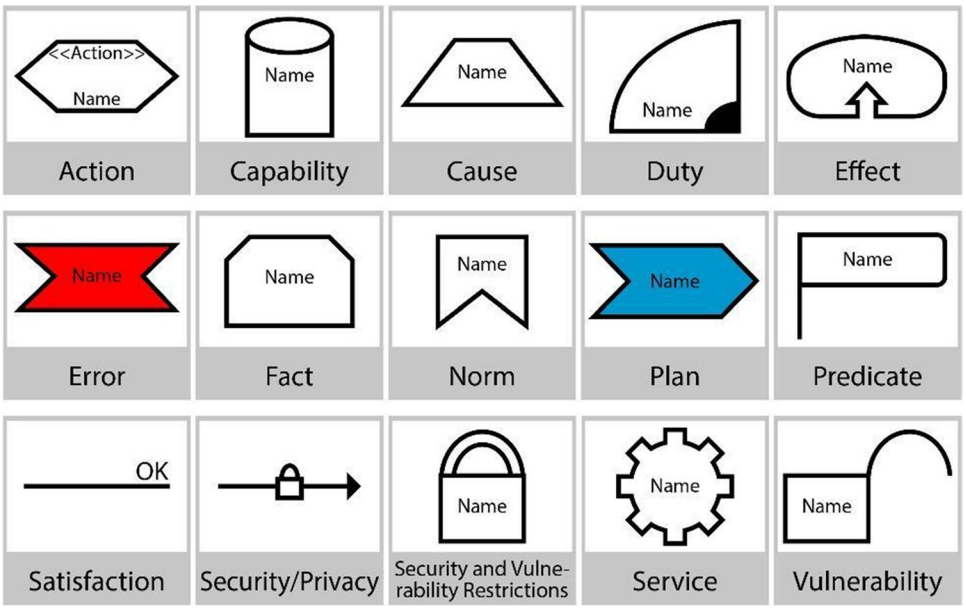
experiments. The execution of these experiments could confirm a choice made by extension proposer or provide a better one. Further analysis is needed to move on to the standardisation of such concepts as elements of the language (we discuss this topic further in Sect. 3.11).

From Tables 3 and 5, we identified concept notations that had best mean in both experiments: Action (PON + R), Capacity (Stereotype), Error (PON + R), Duty (Stereotype), Security and Vulnerability restriction (Stereotype) and Service (Stereotype). Other concepts had different best notations in the experiments, and we combined the hit rate into a single measure (sum of hits from Study 4 and 5 divided by the sum of sample sizes) to indicate best overall notations (see Table 8). The combined hit rate variable was tested for normality with Shapiro–Wilk's, and we could not reject the normality hypothesis ($W = 0.97$, $p = 0.115$).

The set of graphical representation selected to represent the constructs is presented in Fig. 9.

Following the bad performance of originals notations in general, not a single Original notation had a better hit rate performance than expert and user-generated ones. These

Fig. 9 Final symbol set



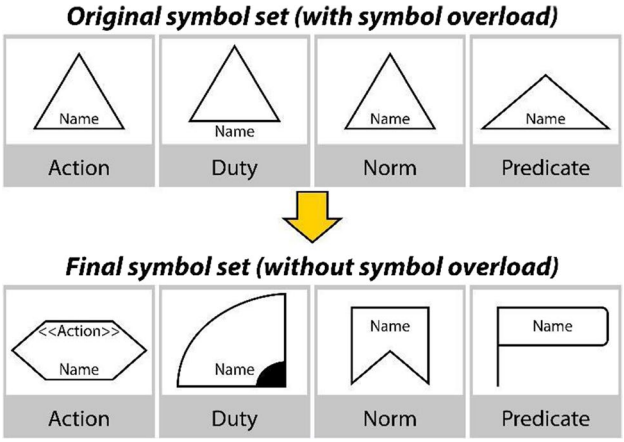


Fig. 10 Evolution in the graphical representation with symbol overload

graphical representations will be included in the catalogue of iStar extensions [11] as alternative representations that mitigate the conflicts found in the literature [10].

The data and scripts used in the analyses of Sect. 3 are available at http://www.cin.ufpe.br/~ejtg/mitigating-conflicts-in-istar-extensions/.

This final symbol set mitigates the symbol overload problem in iStar extensions. We present this scenario in Fig. 10, in which four different concepts share the same graphical representation, i.e. a triangle. At the bottom of the figure, the final result of symbols provides an individual representation for each one of these concepts, resolving the existing symbol overload.

3.11 Discussion

We analyse the results of this multi-stage experiment by two perspectives: symbol selection for resolving conflicts and comparison of symbol design techniques.

We conducted the experiments for a set of 15 concepts, handling each of them individually during the procedure. The best symbols represent what is expected to bring more semantic transparency and recognition rate to iStar models. Indeed, this set of symbols represents a good choice to mitigate overloading conflicts in the collection of iStar extensions. At least, the best-ranked symbols for each concept are a robust initial set for further empirical investigation. Further empirical studies could analyse the behaviour of the symbols in the presence of other language symbols.

The results from the comparison of notation design techniques indicate that there are recurring problems with semantic transparency of iStar extensions proposed by the scientific community. In our study, the Original symbol set performed worse than all other symbol sets from PON + R, PON + R Explained, Stereotype and Prototype. Caire et al. found similar results regarding the core elements of iStar. Authors of extension proposals would better pay more attention to this aspect of the extension. The adoption of more systematic methods to choose and recommend symbols would be valuable for the evolution of the ML.

Our results show that the techniques PON + R, Stereotype and Prototype performed equally within the sample subjects, with no statistical difference. From our results and Caire's results, we cannot indicate the use of a single technique as the best. Our findings may indicate that experts and users-generated techniques for symbolisation that are based on any kind of consensus or ranking will develop outputs of similar performance. The participation of user and experts may explain the reason for those techniques to outperform the ad hoc techniques used by extension proposers.

We realise that the complete method is time- and effort-consuming since it proposes the design of some variations and their assessment. Nonetheless, the experimentation of user- and expert-defined symbols using a sample of the modelling language user population will provide useful empirical data to support final notation decisions. In a simplified execution of this method, modellers may adopt a single-notation design technique. In the context of a broader audience, not restricted to domain experts, the symbols defined will probably have a better semantic transparency and recognition rate than ad hoc chosen symbols.

We believe the adaptations of this method can be used during the proposal of new iStar extensions to propose the new symbols carefully. A catalogue of extensions can help reuse and avoid symbol redundancy and overload [11]. Also, the adapted method is better used as part of a systematic process for extensions creation [34, 35]. Also, it is possible to apply the adapted method to propose new symbols of extensions of other modelling languages. However, it is necessary to perform more tests involving other modelling languages to evaluate and compare the results.

3.12 Threats to validity

According to Juristo and Moreno [36], there are four aspects that we need to consider in threat analysis: Conclusion Validity, Internal Validity, External Validity and Construct Validity. So, we presented these threats to validity of our experiment.

Conclusion validity For recognition and semantic transparency experiments, we recruited undergraduate students from computer science-related programs at the same university campus. Participants were homogenous regarding lack of previous knowledge with goal modelling and basic skills in system modelling and development. We randomly distributed subjects across experimental groups, aiming at balancing the sample sizes. As the collected hit rate metrics were tested for normality and rejected, we

performed nonparametric statistical tests for comparing notation design techniques. We also used this approach in the comparison of representations for individual concepts

Each subject had to answer a 15- or 20-long (for those assigned to Original's treatment) questionnaire. To avoid boredom effect [36] on end-of-survey questions, we configured the survey tool to randomise the sequence of appearance of the questions. Each question had 15 concepts available as response alternatives. Also, to avoid any bias, we configured the survey tool to randomise the alternatives listing order as well.

Internal validity We provided basic training on iStar modelling to all participants, to present the language purpose and core constructs. As it was a short, introductory lecture, we cannot rely only on subjects' recent memory to avoid them, suggesting conflicting representations in symbolisation experiments. Thus, during the experiments, we made available a leaflet with iStar' core representations. Also, there was a risk that the participants of the symbolisation experiment would propose graphical representations in conflict with other representations proposed in iStar extensions. We addressed this risk with a second leaflet with the graphical representations of the constructs proposed in iStar extensions. We asked them to avoid drawings that shared too much resemblance with those in the leaflets. This restriction may constrain the freedom of creativity of participants in different ways, but also may be an inspiration to draw similar symbols.

An important guideline pointed out during training and restated during symbolisation experiments was a general orientation towards the use of simple drawings and abstract shapes (as recommended by the experts in iStar extensions during interviews [37]). We did not discard any drawings not following this recommendation. Although it may hinder participants' full creative engagement, we believe it generated more suitable drawings for effective adoption in tools and official guidelines.

External validity We chose undergraduate students as participants, as they had no previous knowledge of the goal language notation while had some knowledge about the application areas related to the constructs' proposal. Such sample population can be considered a reasonable proxy for the non-experienced user profile in the context of modelling language usage.

However, it can be argued that these undergraduate students may lack technical background on the concepts introduced by the extensions, as these were created typically by domain experts. This potential conceptual barrier might help to explain why they struggled both with the original and the proposed notations. As such, further research is required to assess the extent to which more experienced users from those domains would benefit from the introduction of the proposed notations.

Construct validity We chose to assess the suitability of symbols for a given concept by analysing them in two perspectives: semantic transparency and recognition. Semantic transparency provides a way to evaluate to what extent the meaning of a construct can be inferred from its visual concrete syntax [13]. ISO adopts meaning inferring by symbol as basic measure for comprehensibility [32]. Recognition provides an evaluation of how easy is to learn and remember the meaning of a symbol. Caire et al. argue that such setting is close to what users do in daily activities using modelling languages. However, the short exposure to the symbols and meanings may affect the learning efficacy. Future experiments could control for time of exposure to evaluate whether longer learning phases affect hit rates. Also, the other principles of PoN could be included in the measurement objectives, to investigate the effects of techniques in each principle. Therefore, further research is needed to validate the adequacy of these techniques for the assessment of modelling languages.

We applied training, and then participants answered questionnaires, which could have led them to feel as if they were being evaluated. Such a feeling may pose an evaluation apprehension effect, which confounds with the outcome of the experiment. During the experiment, participants were informed that all drawings and questionnaire responses would be handled anonymously, and none of the tasks they performed would influence their academic evaluations.

Our experiment evaluated symbols of extensions separately, without putting them in the context of a complete model. This isolated evaluation of the notation could be suboptimal, and the presentation of the symbols within a model would contribute to reaching more suitable results. However, it is worth noting that in doing so, participants may be biased to draw symbols similar to those already presented in the models. Moreover, it may influence the creativity of the participants. Since an experiment has not created the symbols of iStar extensions, there is no evidence about their acceptance and expressiveness.

4 Related work

The related work involves the proposal of iStar extension mechanisms [38], the Caire et al. [12] which is the basis for the proposal of our work and other works. Finally, we highlighted the absence of work on extensions in modelling languages.

We considered the proposal of an extension mechanism to iStar as a related work because it represents a complementary way to propose new constructs in iStar. The proposal of iStar extension mechanisms to iStar was presented by Gonçalves et al. [38]. This paper presents an analysis of the lightweight constructs' representation of existing

iStar extensions and results of a survey with experts in iStar extensions to select a subset to be considered in the proposal of an extension mechanism. Finally, it was presented a proposal of extension mechanisms based on this analysis and in a benchmark of extension mechanisms of other modelling languages. The proposal involves the creation of iStar stereotypes and iStar tagged values as visual lightweight mechanisms. The tagged values have a set of default values related to the representations most frequently used in previous iStar extensions.

Additionally, two new elements named iStar groupers and iStar OCL constraints could be hidden as properties in a modelling tool. The iStar groupers are useful to group metaclasses and make easy define constraints for a set of metaclasses in a group. We think a great part of the new constructs can be represented as textual stereotypes or by using other textual markers. However, when a new construct could not specialise an existing iStar, the extender should propose a new graphical symbol. Thus, paper [38] and our paper can be used in a complementary way.

We highlighted the method proposed by Caire et al. [12] since it was used as the basis of our experimental design. These authors conducted a set of experiments to improve the graphical representation of modelling languages, and the authors used iStar to illustrate the usage of their principles.

Many experiments have been performed involving goal modelling and iStar using the Caire et al. [12] proposal. They were included because they represent evidence of the adoption of their method by the scientific community. Santos et al. [39] proposed new symbols to KAOS [4] using an experimental design based on Caire et al. [12]. Santos et al. [39] performed the symbolisation experiment, stereotyping analysis, prototyping experiment and semantic transparency experiment. Finally, Santos et al. [39] concluded that the semantic transparency of the prototyping symbol set was significantly higher than the standard one. Henriques et al. [40] followed the same steps of Santos et al. [39] to recreate the symbols of the Low-Code Process Modelling Language. Both works Santos et al. [39] and Henriques et al. [40] are similar to our work once they are based on Caire et al. [12] design, but they were not used in the context of extensions of modelling languages. The use of eye-tracking devices has been used in recent research involving analysis of requirements models. The work of Santos et al. [39] uses eye-tracking devices to analyse the ease of understanding and inspection of iStar models comparing the graphical representations of default iStar and the iStar symbols proposed by Caire et al. [12]. However, Santos et al. [39] did not find significant differences during the analysis of the participants' data.

None of the studies presented above describes an evaluation of representations used in iStar extensions, to propose a ranking and be used as a parameter of choice in future extensions. We did not find any work which analyses the use of empirical studies in the proposal of constructs of extensions in other modelling languages.

5 Conclusions

In this paper, we presented the results of a multi-stage experiment whose objective was to mitigate the existing symbol overload in iStar extensions, identified in previous work [10].

Thus, we adapted a method proposed by Caire et al. [12] to mitigate symbol overload in extensions of a modelling language. This method was initially proposed to create symbols of the original syntax of the modelling languages. We analysed the use of the adapted method to mitigate the existing overload notation conflicts in iStar extensions identified in an SLR [10]. Symbol overload means that two or more extensions are using the same symbol for different concepts. Thus, we derived alternative symbols of the existing iStar extensions and evaluated the best representations concerning semantic transparency.

We proposed new representations for them and asked the participants of Study 1 to draw their representations. We included the training of the participants in the modelling language to be extended and included restrictions to the creation of new constructs. Then, we identified the most frequently drawn symbol for each construct in Study 2 to generate the Stereotype symbol set. We also analysed the most frequently chosen symbol by our participants in Study 3 (Prototyping experiment), so we generated the Prototype symbol set. Finally, we performed Studies 4 and 5 to analyse the semantic transparency and the recognition of the Original, PON + R, Prototype and Stereotype symbol sets. The result was used to select new representation which can be useful in the usage of existing extensions with conflict.

As a whole, none of the full symbol sets was significantly better than the alternatives. In the end, we selected four (4) symbols from the PON + R set, four (4) from the Stereotype set and seven (7) from the Prototype set. All the symbols in the final symbol set had significantly better general results than the symbols of the Original set. Despite this improvement, most symbols had weak results in the semantic transparency experiment (Study 5). These results may be useful when iStar extensions are proposed, which reuse two or more extensions in notation conflict. We believe that the adapted Caire et al.'s experimental design can be used to propose symbols of extensions in other modelling languages. The use of the method helps mitigating symbol overload conflicts with existing extensions.

We intend to use this adapted method to propose symbols of a new iStar extension and analyse its use to propose new symbols of extensions of other modelling languages.

Practitioners and experts working on the design of modelling languages and extensions should refrain from the use of ad hoc symbol selection and prefer user- or expert-generated techniques. The use of a single technique will probably result in better semantic transparency of the symbols. If multiple techniques are used, the adapted method proposed in this work provides an approach to integrate the results and support the decision process.

We presented in this paper an isolated evaluation of the symbols. As future work, we intend to analyse the symbols in the context of a complete model. Thus, we can compare the results of our paper with the results of this future work.

The results of this paper are part of research where we analysed the existing iStar extensions and proposed how to improve them. Thus, in two previous works, we analysed how the iStar extensions were proposed: an SLR [10] and a qualitative study [37].

As future work, we are currently working on a process to guide the proposal of next iStar extensions. This process is based on the reuse of existing extensions identified in the study [10], including the representations of this paper, and recommendations identified during the interviews of paper [37]. The process will consider the definition of the related concepts, abstract and concrete syntax maintaining the traceability.

Acknowledgements The authors thank all participants of this study. We also thank CNPQ/Brazil (Conselho Nacional de Desenvolvimento Científico e Tecnológico) and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) 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/04516/2019).

References

1. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice. Morgan & Claypool Publishers Series Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, San Rafael (2012)
2. Miles, R., Hamilton, K.: Learning UML 2.0. O'Reilly, Newton (2006)
3. Mussbacher, G., Amyot, D., Breu, R., Bruel, J., Cheng, B., Collet, P., Combemale, B., France, R., Heldal, R., Hill, J., Kienzle, J., Schöttle, M., Steimann, F., Stikkolorum, D., Whittle, J.: The relevance of model-driven engineering thirty years from now. In: Model-Driven Engineering Languages and Systems, pp. 183–200. Springer International Publishing (2014)
4. Dardene, A., Van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program. **20**, 3–50 (1993)
5. Dalpiaz, F., Franch, X., Horkoff, J.: iStar 2.0 language guide. arXiv:1605.07767. May 2016. http://arxiv.org/pdf/1605.07767v1.pdf
6. Yu, E.: Towards modelling and reasoning support for early phase requirements engineering. In: Proceedings of the 3rd IEEE International Conference on Requirements Engineering (1997)
7. Giorgini, P., Rizzi, S., Garzetti, M.: Goal-oriented requirement analysis for data warehouse design. DOLAP (2005)
8. Lapouchnian, A., Yu, E., Liaskos, S., Mylopoulos, J.: Requirements-driven design of autonomic application software. In: Conference of the Center for Advanced Studies on Collaborative Research (2006)
9. Ghanavati, S., Amyot, D., Rifaut, A.: Legal Goal-Oriented Requirement Language (Legal GRL) for modelling regulations. In: 6th International Workshop on Modelling in Software Engineering, MiSE (2014)
10. Gonçalves, E., Castro, J., Araujo, J., Heineck, T.: A systematic literature review of iStar extensions. J. Syst. Softw. **137**, 1–33 (2018)
11. Gonçalves, E., Heineck, T., Araújo, J., Castro, J.: CATIE: a catalogue of iStar extensions. Cadernos do IME-Série Informática **41**, 23–37 (2018)
12. Caire, P., Genon, N., Heymans, P., Moody, D.: Visual notation design 2.0: towards user comprehensible requirements engineering notations. In: 21st IEEE International Requirements Engineering Conference (RE) (2013)
13. Moody, D.: The, "Physics" of notations: towards a scientific basis for constructing visual notations in software engineering. IEEE Trans. Softw. Eng. **35**(5), 756–779 (2009)
14. Mendonça, D.F., Rodrigues, G.N., Ali, R., Alves, V., Baresi, L.: GODA: a goal-oriented requirements engineering framework for runtime dependability analysis. Inf. Softw. Technol. J. **80**, 245–264 (2016)
15. Ali, R., Dalpiaz, F., Giorgini, P.: Requirements-driven deployment. J. Softw. Syst. Model. **13**(1), 433–456 (2014)
16. Dalpiaz, F., Paja, E., Giorgini, P.: Security requirements engineering via commitments. In: 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST) (2011)
17. France, R., Rumpe, B.: Model-driven development of complex software: a research roadmap. In: Future of Software Engineering at ICSE' 07, pp. 37–54, Minneapolis (2007)
18. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. IEEE Softw. **11**(2), 42–49 (1994)
19. Ali, R., Dalpiaz, F., Giorgini, P.: Location based software modelling and analysis: tropos-based approach. In: International Conference on Conceptual Modelling, Lecture Notes in Computer Science, vol. 5231, pp. 169–182 (2008)
20. Morandini, M., Penserini, A., Perini, A., Marchetto, A.: Engineering requirements for adaptive systems. Requir. Eng. J. **22**(1), 77–103 (2015)
21. Guzman, A., Martinez, A., Agudelo, F., Estrada, H., Perez, J., Ortiz, J.: A methodology for modelling ambient intelligence applications using i* framework. In: International iStar Workshop in IEEE International Requirements Engineering Conference, pp. 61–66 (2016)
22. Islam, S., Mouratidis, H., Kalloniatis, C., Hudic, A., Zechner, L.: Model based process to support security and privacy requirements engineering. Int. J. Secure Softw. Eng. **3**(3), 1–22 (2012)
23. Gans, G., Lakemeyer, G., Jarke, M., Vits, T.: SNET: a modelling and simulation environment for agent networks based on i* and Congolog. In: International Conference on Advanced Information Systems Engineering (2006)
24. Siena, A., Maiden, N., Lockerbie, J., Karlsen, K., Perini, A., Susi, A.: Exploring the effectiveness of normative i* modelling: results from a case study on food chain traceability. In: International Conference on Advanced Information Systems Engineering (2008)
25. Liu, L., Yu, E., Mylopoulos, J.: Security and privacy requirements analysis within a social setting. In: IEEE International Conference on Requirements Engineering (2003)
26. Goodman, N.: Languages of Art: An Approach to a Theory of Symbols. Bobbs-Merrill Co., Indianapolis (1968)
27. Horkoff, J., Yu, E.: Finding solutions in goal models: an interactive backward reasoning approach. In: International Conference on Conceptual Modelling (2010)

28. Tichy, W.F.: Hints for reviewing empirical work in software engineering. Empir. Softw. Eng. **5**(4), 309–312 (2000)
29. Granada, D., Vara, J.M., Brambilla, M., Bollati, V., Marcos, E.: Analysing the cognitive effectiveness of the webml visual notation. Softw. Syst. Model. **16**(1), 195–227 (2017)
30. Howell, W.C., Fuchs, A.H.: Population stereotypy in code design. Org. Behav. Hum. Perform. **3**(3), 310–339 (1968)
31. Jones, S.: Stereotypy in pictograms of abstract concepts. Ergonomics **26**(6), 605–611 (1983)
32. Foster, J.J.: Graphical symbols: test methods for judged comprehensibility and for comprehension. ISO Bull., 11–13 (2001)
33. Zwaga, H.J., Boersema, T.: Evaluation of a set of graphic symbols. Appl. Ergon. **14**(1), 43–54 (1983)
34. Gonçalves, E.: PRISE: a process to support iStar extensions. Ph.D. thesis in Computer Science, Universidade Federal de Pernambuco (2019)
35. Gonçalves, E., Araujo, J., Castro, J.: PRISE: a process to support iStar extensions. J. Syst. Softw. (2020) (submitted, for a copy contact: enyo@ufc.br)
36. Juristo, N., Moreno, A.M.: Basics of Software Engineering Experimentation. Springer, Berlin (2001)
37. Gonçalves, E., De Oliveira, M., Monteiro, I., Castro, J., Araujo, J.: Understanding what is important in iStar extension proposals: the viewpoint of researchers. Requir. Eng. J. **24**, 55–84 (2018)
38. Gonçalves, E., Araujo, J., Castro, J.: Towards extension mechanisms in iStar 2.0. In: 11th International i* Workshop co-located with the 30th International Conference on Advanced Information Systems Engineering (2018)
39. Santos, M., Gralha, C., Goulão, M., Araújo, J.: Increasing the semantic transparency of the KAOS goal model concrete syntax. In: 37th International Conference on Conceptual Modelling (2018)
40. Henriques, H., Lourenço, H., Amaral, V., Goulão, V.: Improving the developer experience with a low-code process modelling language. In: 21st International Conference on Model Driven Engineering Languages and Systems (2018)
41. Siena, A., Jureta, I., Ingolfo, S., Susi, A., Perini, A., Mylopoulos, J.: Capturing variability of law with Nomos 2. In: International Conference on Conceptual Modelling (2012)
42. Schulz, F., Meissner, J., Rossak, W.: 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 (2013)
43. Siena, A., Mylopoulos, J., Perini, A., Susi, A.: Designing law-compliant software requirements. In: International Conference on Conceptual Modelling (2009)
44. Mellado, D., Mouratidis, H., Fernandez-Medina, E.: Secure Tropos framework for software product lines requirements engineering. Comput. Stand. Interfaces **36**, 711–722 (2014)
45. Mouratidis, H., Islam, S., Kalloniatis, C., Gritzalis, S.: A framework to support selection of cloud providers based on security and privacy requirements. J. Syst. Softw. **26**, 2276–2293 (2013)
46. Murukannaiah, P., Sigh, M.: Xipho: extending Tropos to engineer context-aware personal agents and multi-agent systems (2014)
47. Estrada, H., Martinez, A., Santillán, L.C., Pérez, J.: A new service-based approach for enterprise modelling. Computacion y Sistemas **17**, 625–639 (2013)
48. Ali, R., Dalpiaz, F., Giorgini, P.A.: Goal modelling framework for self-contextualizable software. In: Enterprise, Business Process and Information Systems Modelling Workshop on International Conference on Advanced Information Systems Engineering (2013)
49. Chopra, A., Dalpiaz, F., Giorgini, P., Mylopoulos, J.: Modelling and reasoning about service-oriented applications via goals and commitments. In: International Conference on Advanced Information Systems Engineering (2010)
50. Giorgini, P., Rizzi, S., Garzetti, M.: GRAnD: a goal-oriented approach to requirement analysis in data warehouses. Dec. Support Syst. J. **45**(1), 4–21 (2008)
51. Liaskos, S., Mylopoulos, J.: On temporally annotating goal models. In: International i* Workshop (2010)

Enyo Gonçalves received his Ph.D. in Computer Science at Universidade Federal de Pernambuco, Brazil, in 2019. He completed his Master in Computer Science at the Universidade Estadual do Ceará, Brazil, in 2009. He received a B.S. degree in Computer Science from Universidade Estadual do Vale do Acaraú, Brazil, in 2007. Mr. Gonçalves is currently a Professor at Universidade Federal do Ceará, Brazil. Enyo's research is focused primarily on the software engineering methods and their application to a wide variety of problems in model-based engineering (MBE) applied to requirements engineering (RE) and multi-agent systems (MAS). His current research projects include analysis of extensions in modelling languages.

Camilo Almendra is currently a Ph.D. candidate in Centro de Informática at Universidade Federal de Pernambuco, Brazil. His research interests include requirements engineering, safety–critical systems and safety certification. Currently, he is a professor at Universidade Federal do Ceará, Brazil.

Miguel Goulão received his Ph.D. degree from Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa (FCT/UNL), in 2008. He is an Assistant Professor of the Informatics Department of FCT/UNL and an associate member of the NOVA LINCS research centre. He has 20 years of experience with experimental software engineering (ESE) and its applications to several software engineering areas. Miguel has been applying ESE to evaluate claims on object-oriented design, software process improvement, software evolution and reengineering,

component-based software engineering, and, more recently, in requirements engineering and software languages engineering. Miguel is currently focusing his research on how complexity affects the usability of software development languages, namely requirements engineering and domain-specific languages. Miguel has published over 60 papers in peer-reviewed international journals, conferences and workshops and served as PC member and journal reviewer in top-ranked conferences and journals. He was a co-author of the paper receiving the best paper award in CAiSE 2014 and of the paper receiving the János Szentes Award for the best paper on Software Metrics presented at the 6th European Conference on Software Quality, in 1999.

João Araújo is a Professor at the Department of Informatics at the Universidade Nova de Lisboa, Portugal, and a full member of the Portuguese Research Center NOVA LINCS. He holds an M.Sc. from Universidade Federal de Pernambuco and a Ph.D. from Lancaster University, UK, both in the area of software engineering. His principal research interests are requirements engineering (RE), advanced modularity and model-driven engineering (MDE), where he has published several papers on these topics in journals, international conferences and workshops. Within these subjects he has also been involved in several projects, such as AMPLE (funded by the European Union), Aspects for Space Domain (funded by ESA), etc. He has served in the organisation of several conferences such as RE, MoDELS, ICSE, ECOOP, AOSD. He was the co-general chair of the IEEE Requirements Engineering Conference in Lisbon, 2017. He was awarded with the most influential paper of the AOSD 2013 and best paper award of CAiSE 2014, among others. He has launched the series of workshops on model-driven RE (MoDRE) that has been held in RE conference.

Jaelson Castro is a Professor at Universidade Federal de Pernambuco, Brazil, where he leads the Requirements Engineering Laboratory. He holds a Ph.D. in Computing from Imperial College, London, UK. His research interests include requirements engineering, model-driven safety–critical system development and robotics. He serves on the editorial board of the Requirements Engineering Journal and the Journal of Software Engineering Research and Development and acted as Editor-in-Chief of the Journal of the Brazilian Computer Society—JBCS.