

# piStar-ext: Supporting the Creation of iStar Extensions with the piStar Tool

Enyo Gonçalves<sup>1</sup>, Guilherme Rodrigues<sup>1</sup>, Paulo Miranda<sup>1</sup>,  
João Pimentel<sup>4</sup>, João Araujo<sup>3</sup>, Jaelson Castro<sup>2</sup>

<sup>1</sup> Universidade Federal do Ceará, Brazil

<sup>2</sup> Universidade Federal de Pernambuco, Brazil

<sup>3</sup> Universidade Nova de Lisboa, Portugal

<sup>4</sup> Universidade Federal Rural de Pernambuco, Brazil

enyo@ufc.br, guilhermerodriguesbr@outlook.com,  
paulomirandamss12@gmail.com, joao.hcpimentel@ufrpe.br,  
joao.araujo@fct.unl.pt, jbc@cin.ufpe.br

**Abstract.** Many extensions have been proposed since the initial iStar version, in the 90's. We have been analysing these extensions and proposing ways to support new ones. In previous work, we have proposed extension mechanisms to iStar 2.0. Since modelling tools play a central role during the modelling of systems, we believe that a modelling tool that supports the creation of extensions through a graphical user interface could make this task easier. In this paper, we report on the results of supporting iStar extensions with piStar-ext, an iStar 2.0 web modelling tool. Our tool allows the proposal of new constructs using textual representation as well as new symbols. We illustrated our proposal by representing an existing iStar extension with the new features included in piStar-ext.

**Keywords:** iStar, Extension Mechanisms, Goal Modelling Tool.

## 1 Introduction

iStar is a goal-based modelling language proposed in 90's and has been widely adopted by the requirements engineering academic community [15]. In 2016 it was updated to iStar 2.0, through a collaborative effort [1]. Since its creation, iStar has been extended to represent specific constructs for specific domains. Extensions can make a modelling language more expressive and suitable to a specific domain/application area, improving the legibility of the models. Extensions also can provide a more accurate model transformation or code generation. Several previous works were conducted to identify [2] and catalogue [5] the existing iStar extensions, understand the opinion of experienced extenders [3], propose extension mechanism [4] and define a process to support the creation of new extensions [8][7], among others. We highlight the proposal of extension mechanisms which involves iStar stereotypes, iStar tagged-values as visual lightweight mechanisms and two new elements named iStar groupers and iStar OCL constraints, which are hidden as properties. These mechanisms can facilitate the proposal of new iStar extensions in a systematic way. However, the effective use of these mechanisms depends on having these mechanisms available in a modelling tool. The iStar

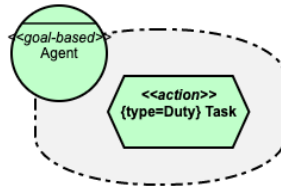
Wiki<sup>1</sup> lists 29 different iStar modelling tools. Only a small subset of these tools supports iStar 2.0, but none has a Graphical User Interface (GUI) that allows the creation of the extensions mechanisms.

In this paper, we present the piStar-ext, a web-based tool that supports the creation of iStar 2.0 diagrams that require the extensions mechanisms proposed in [4]. It is an evolution of the piStar modelling tool [13]. Section 2 describes the background and related work, Section 3 presents the results and Section 4 concludes the paper and presents further research.

## 2 Background and Related Work

### 2.1 Extension Mechanisms in iStar

The proposal of extension mechanisms presented in [4] involves iStar stereotypes (represented between “<<” and “>>”) and iStar tagged-values (represented between brackets) visual light-weight mechanisms as a specialisation of iStar extension mechanisms. In UML there are 33 default stereotypes such as <<Utility>>, applied to a Class to inform that it has no instances, but rather denotes a named collection of attributes and operations. So, we used the default values (Id, Reference to, Status and Logic) proposed by [4] to tagged values. Node identifier could be referred to Id. Reference to external representation and reference to another part of the diagram can be represented by the value Reference to. Node status can be labelled as Status. Finally, Label with syntax can be labelled as Logic. Thus, a goal can be labelled, for example as <<Business>> {Id = G1} Accommodation booked. Fig. 1 presents an agent with the stereotype <<goal-based>> and a task with the stereotype <<action>> and the tagged value type.



**Fig. 1.** Illustration of Stereotype and Tagged Value in iStar Diagrams.

Additionally, it would have two new elements named iStar groupers and iStar OCL constraints, which could be hidden as properties in a modelling tool to represent validation rules. The iStar groupers are useful to group constructs and make it easy to apply stereotypes or tagged values for a set of the constructs in a group. Thus, we could propose a group named Rational Grouper, which includes Actor and Role to apply the stereotype <<goal-based>> to this group.

---

<sup>1</sup> <http://istarwiki.org/>

## 2.2 piStar tool

The piStar tool [13] is a web tool to create iStar models available at <https://cin.ufpe.br/~ler/pistar/>. It was developed by using web technologies HTML, CSS and Javascript. The code is structured in two main packages: language, which defines the metamodel, constraints and shapes of the iStar constructs; and app, which presents the main functionalities of the piStar tool. Fig. 2 shows an overview of the piStar tool user interface. The top of Fig. 2 shows the modelling palette, inspired by the design of the OME tool. The elements and links of the palette can be added to the drawing area. The visual design of the diagram is based on the original iStar 2.0 notation [1]. Actors can be collapsed and presented at will. Thus, one can create different views (Strategic Dependencies and Strategic Relationships) of the same model.

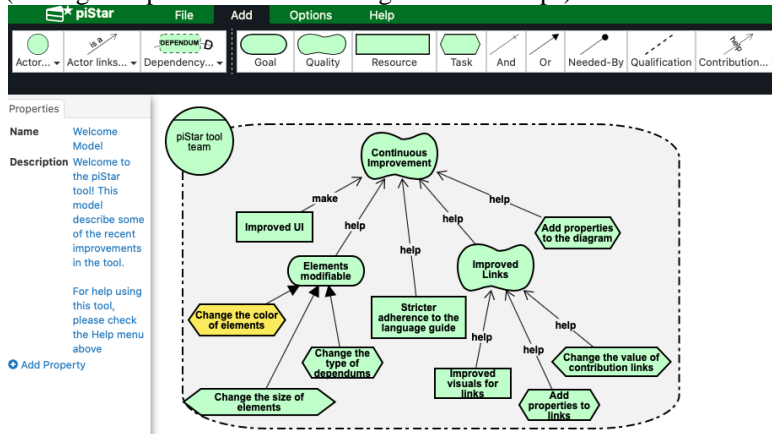


Fig. 2. An Overview of the piStar tool.

## 2.3 Related work

A vast number of modelling tools have been proposed for iStar. We mention here two tools that support iStar 2.0. iStar 2.0 editor [12] is a desktop-based iStar modelling tool created using Sirius<sup>2</sup>, an eclipse framework for the model-driven development of modelling tools. The authors presented a metamodel of the tool and pointed out some details about its implementation. The Leaf 2.0 tool<sup>3</sup> is similar to piStar in the sense that it is a web-based iStar 2.0 modelling tool. It constitutes the basis of three more specific tools: CreativeLeaf [11] and GrowingLeaf/BloomingLeaf [10].

Web tools such as piStar can make it easier to model when compared with desktop tools such as iStar 2.0 editor since they do not require any software installation. Unlike piStar, Leaf does not support the creation of dependency links. iStar 2.0 editor and Leaf do not support the creation of new iStar extensions by using a GUI.

<sup>2</sup> <https://www.eclipse.org/sirius/>

<sup>3</sup> <http://www.cs.toronto.edu/~amgrubb/leaf>

### 3 piStar-ext

piStar-ext<sup>4</sup> is a specific tool based on piStar [13] with support for the creation of extensions. We added a set of new features to the piStar tool to support the creation of iStar extensions based on our proposal of extensions mechanisms presented in [4]. Fig. 3 presents an overview of the new features which support the creation of iStar extensions.

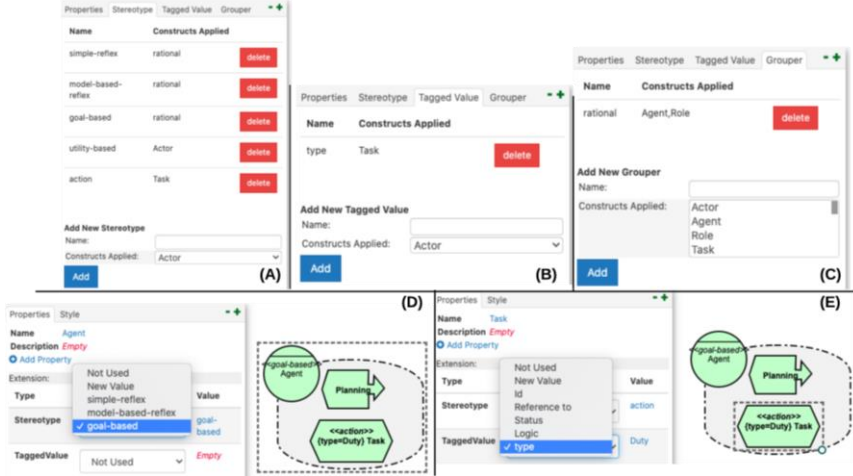


Fig. 3. New Features of piStar-ext tool.

We added three new tabs which appear when the modeller select a blank area in the canvas (Fig. 3-A, Fig. 3-B and Fig. 3-C). They are related to the creation of a set of stereotypes, tagged values and groupers.

Fig. 3-A presents the creation of iStar textual stereotypes tab, which is used to specialise existing iStar nodes and links. An iStar stereotype can be applied to an iStar grouper, to an iStar node or an iStar link. In Fig. 3-A the stereotypes simple-reflex, model-based reflex, goal-based and utility based, which represent the kind of rational agents presented by [14], are applied to rational grouper. Complementary, the stereotype action is applied to Task to represent agents' actions[14].

Fig. 3-B presents the creation of iStar tagged values which are used to add visual information to the constructs. An iStar tagged value can be applied to an iStar grouper, to an iStar node or an iStar link. Fig. 3-B presents the tagged value *type* applied to the Task to represent the kind of Task. The possible values to *type* are duty, which represents that a mandatory Task, and right, which represents that an optional Task.

Note that a blank diagram with only the definition of stereotypes, tagged-values and groupers can be created and shared as a basis to create models of an iStar extension.

Fig. 3-C presents the creation of iStar groupers which are used to create a subset of iStar constructs to facilitate the application of stereotypes and tagged values. An iStar grouper can join a subset of iStar constructs. Fig. 3-C depicts *rational* grouper (related

<sup>4</sup> The tool is available to use at [www.cin.ufpe.br/~ler/pistar-ext](http://www.cin.ufpe.br/~ler/pistar-ext). The Code is available at <https://github.com/guiRodrigues/pistar-supporting-creation-istar-extensions>

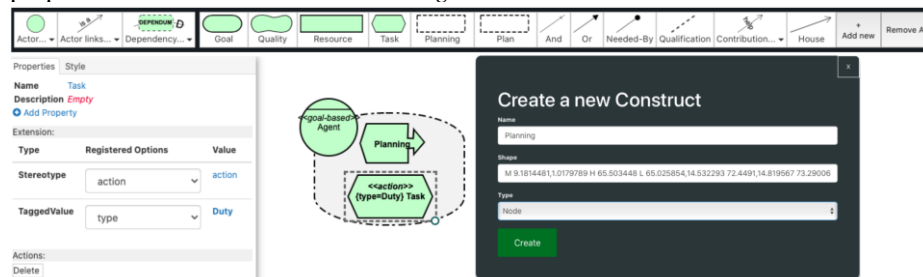
to rational agents [6]) applied to Actor and Role. This grouper is used by the stereotype shown in Fig. 3-A. Groupers can be applied to tagged-values as well.

Fig. 3-D presents the usage of the goal-based stereotype applied to an Agent. Fig. 3-E presents the usage of the tagged value type to a Task. Note that a set of the four default values of tagged value (Id, Reference to, Status and Logic) [4] are presented as options. Stereotypes created in the Stereotypes tab (Fig. 3-A) also appear as an option in the dropdown list to be applied to an element (Fig. 3-D). Note that goal-based appears as a possible stereotype in Fig. 3-D. Tagged values created in the tab tagged values (Fig. 3-B) also appear as an option in the dropdown list to be applied to an element (Fig. 3-E). Note that *type* appears as a possible tagged value in Fig. 3-E. Additionally, the modeller can create a new stereotype or tagged value during the modelling of the elements of the diagram.

In summary, to add new textual elements follow these steps: 1. Click in the blank area of the canvas, 2. Create the Groupers, Stereotypes and Tagged values, 3. Add new nodes and links to canvas, 4. Apply the pre-defined Stereotypes and Tagged values. It is also possible to create new stereotypes and tagged values when creating new models.

It is common for iStar extensions to create constructs using new symbols (as in the extension proposed to model rational agents [6] and the recreation of a subset of symbols of existing extensions [9]). Thus, we included a way in piStar to create constructs based on new symbols using the GUI via the new button (Add new) that was added to the palette. When the modeller clicks on this button, a pop-up window opens with the definition of the new construct. The modeller should inform data about the name of the construct, shape (parameters of the svg image that makes the symbol which can be created with a tool such as Inkscape in [inkscape.org](http://inkscape.org) or [svgicons in http://svgicons.sparkkk.fr/](http://svgicons.sparkkk.fr/)), type (node or link), source and target (mandatory when proposing a link). Then the new construct is included in the palette and can be used in the diagrams. Fig. 4 shows the button to add new construct on the right side of the palette, the pop-up window with the specifications of the new construct, new constructs named *Planning* and *Plan* added to the palette and the usage of the new construct *Planning* in canvas. *Planning* is a construct used to represent the creation of a sequence of tasks by an agent to achieve their goals. These constructs were proposed in [6].

Fig. 4 illustrates the usage of piStar-ext to create iStar models of an extension representing rational agents [6]. New constructs were proposed to model the kinds of agent and roles and a set of specific agent functions using stereotypes. New symbols were proposed for constructs such as *Planning* and *Plan*.



**Fig. 4.** Adding Constructs with New Symbol in piStar-ext tool.

## 4 Conclusions and Further research

iStar has been extended since its initial proposal in the 90's. In previous work, we considered some iStar extensibility mechanisms [4]. In this paper, we discuss how these extension mechanisms were included in the piStar tool, an iStar 2.0 modelling tool. We also presented an example of the usage of this new feature included in piStar. As future work, we intend to include OCL Rules (part of the iStar extension mechanism not implemented in this version of piStar) in the piStar tool. We also intend to perform an experiment to analyse time and effort to add the constructs of an iStar extension by comparison of the usage of two different versions of this tool (piStar and piStar-ext).

**Acknowledgements.** The authors thank CNPq, CAPES, FACEPE and NOVA LINC'S UID/CEC/04516/2019 for the financial support to the execution of this work.

## References

1. Dalpiaz, F., Franch, X., Horkoff, J. iStar 2.0 Language Guide. arXiv:1605.07767, 2016, Available in <http://arxiv.org/pdf/1605.07767v1.pdf>
2. Gonçalves, E., Castro, J., Araújo, J., Heineck, T. A Systematic Literature Review of iStar Extensions, *Journal of Systems and Software*, 2017.
3. Gonçalves, E., Monteiro, I., de Oliveira, M. A., Castro, J., Araújo, J. Understanding What is Important in iStar Extensions Proposals: The Viewpoint of Researchers, *Requirements Engineering Journal*, 2017.
4. Gonçalves, E., Araújo, J., Castro, J. Towards Extension Mechanisms in iStar 2.0. 11<sup>th</sup> International i\* Workshop co-located with the 30<sup>th</sup> International Conference on Advanced Information Systems Engineering, 2018.
5. Gonçalves, E., Heineck, T., Araújo, J., Castro, J.: CATIE: a catalogue of iStar extensions. *Cadernos do IME-Série Informática* 41,23–37, 2018.
6. Gonçalves, E., Araújo, J., Castro, J. iStar4RationalAgents: Modeling Requirements of Multi-Agent Systems with Rational Agents, 38<sup>th</sup> International Conference on Conceptual Modeling, 2019.
7. Gonçalves, E., Araújo, J., Castro, J. PRISE: A process to Support iStar Extensions, *Journal of Systems and Software*, 2020.
8. Gonçalves, E., Araújo, J., Castro, J. A Process to Support the Creation of iStar Extensions. 35<sup>th</sup> ACM/SIGAPP Symposium on Applied Computing, p. 1363–1370, 2020.
9. Gonçalves, E., Almendra, C., Goulão, M., Araújo, J., Castro, J. Using empirical studies to mitigate symbol overload in iStar extensions. *Software and Systems Modeling*, 2020.
10. Grubb, A., Song, G., Chechik, M. GrowingLeaf: Supporting requirements evolution over time, 9<sup>th</sup> International iStar workshop, 2016.
11. Horkoff, J., Maiden, N. Creative Leaf: A Creative iStar Modeling Tool. 9<sup>th</sup> International iStar workshop, 2016.
12. Liebenberg, M., Roßmaier, K., Lakemeyer, G. An iStar 2.0 Editor Based on the Eclipse Modelling Framework. 10<sup>th</sup> International iStar workshop iStar. 2017.
13. Pimentel, J., Castro, J. piStar tool—a pluggable online tool for goal modeling, IEEE 26<sup>th</sup> International Requirements Engineering Conference, 2018.
14. Russell, S., Norvig, P. Artificial Intelligence: A Modern Approach, Prentice Hall (2003).
15. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. Social modeling for requirements engineering. Mit Press, 2011.