iStar4RationalAgents: An iStar extension to model Multi-Agent Systems with Rational Agents

Enyo Gonçalves (Contact author)
Universidade Federal do Ceará, Av. José de Freitas Queiroz, 5003 – Cedro CEP 63900-000, Quixadá, CE – Brazil, +55 88 3412 0919, +55 88 3412 3348
enyo@ufc.br

João Araújo
NOVA LINCS, FCT
Universidade Nova de Lisboa, Lisboa– Portugal
joao.araujo@fct.unl.pt

Jaelson Castro
Universidade Federal de Pernambuco, Recife, PE – Brazil
jbc@cin.ufpe.br

Abstract: Requirements engineering is a relevant discipline in software development. The development of systems of specific fields requires techniques of requirements specification suitable for their needs. Multi-agent systems (MAS) involve a wide variety of agents that interact with each other to achieve their goals. Usually, the agents in a MAS can be reactive or proactive, this choice defines the rationale of its elements. Rational Agents is the term used to mention a set of four kinds of reactive and proactive agents. Conceptual models which represent the rational agents' intentionality can be used to design and analyse MAS in a systematic and structured manner. Goal modelling is a way to specify requirements which can provide a more detailed and systematic analysis of MAS decisions. However, the modelling of MAS with rational agents different is a non-trivial task, due to the specificity of their domain concepts, also at the requirements level. This paper presents an approach to model MAS with rational agents in requirements level using iStar. This is part of a Model-Driven Development approach which has been proposed to support the development of MAS with rational agents involving requirements, architecture, code and test. We extended iStar to support the modelling of main concepts of this domain in a systematic way based on a process to conduct iStar extensions. We modelled a MAS to Distance education to validate and illustrate the usage of our extension and evaluate the results using a survey with experienced researchers/developers in MAS.

Keywords: Goal-Modelling; iStar; MDD approach; Multi-agent system; Rational agents;

1. INTRODUCTION

Autonomous software based on artificial intelligence (AI) have been widely applied to solve a vast set of problems in companies. In this context, agents are complex entities with behavioral properties, such as (i) autonomy (i.e., they are able to execute without interacting with humans), and (ii) interaction (i.e., they are able to interact by sending and receiving messages and not by explicit task invocation [37]. Multi-agent system (MAS) is the subarea of AI that investigates the behavior of a set of autonomous agents, aiming to resolve a problem beyond the capacity of a single agent [22].

A simple agent can act based on reactive or proactive behavior and can be classified according to its internal architecture that determines distinct agency properties, attrib-utes and mental components [33]. Russell and Norvig [33] define four types of agents according to their internal structure: Simple Reflex Agent, Model-Based Reflex Agent, Goal-based Agent and Utility-Based Agent. The type of agent is selected according to the environment characteristics and to the subproblem that the agent will resolve. A MAS may encompass multiple types of agents [37].

A Model-driven approach has been proposed to model MAS with rational agents. The development of MAS with rational agents is supported in the modelling of the architectural level by the MAS-ML 2.0 extension [10], the code generation is supported by the approach proposed by [27] and the test is supported by the proposal of [36].

Zave [43] cites that Requirements Engineering (RE) is concerned with real-world goals for functions and constraints on software. RE is an important part of the software development process, from elicitation and specification to maintenance of requirements [24]. Evidence shows that requirements errors, such as misunderstandings and omissions, are more expensive to repair in later phases, such as project, development and test [2]. According to Nuseibeh; Easterbrook [31], modelling is the construction of abstract descriptions that are amenable to interpretation, and it is a fundamental activity in the RE area to reduce misunderstandings and omissions. Therefore, it is necessary a proposal to model requirements of MAS with rational agents.

The modelling of requirements for a multi-agent system can be preferably an extension of a known and trusted modelling language, such as iStar. We have chosen iStar since it is one of the most used Goal-Oriented Requirements Engineering (GORE) languages and it supports the modelling of part of the concepts of MAS with rational agents such as goal, belief, agent, tasks and supports the modeling of organisational concepts as well. Additionally, this language has a process to conduct iStar extensions [17][18] which make easy the proposal of new iStar extensions.

This paper presents an approach to model MAS with rational agents in requirements level using iStar named iStar for rational agents. This extension was proposed following the PRISE, a process to conduct iStar extensions. We represented the constructs as a set of stereotypes and four constructs were represented as new symbols proposed by an experiment similar to the presented by [5]. This approach is supported by iStar4rationalagents tool. We present conceptual models to illustrate the modelling of MAS with rational agents based on a real case example of a course of programming in distance education offered in MOODLE. This proposal was evaluated by experienced researchers/developers of MAS.

This paper is structured as follows. Section 2 presents the main concepts of iStar and MAS. Section 3 presents related work. Our proposal to model MAS with rational agents is described in Section 4. In Section 5 the modelling of the MAS to support the distance education course of programming people with disabilities is presented. The evaluation of MAS is presented in Section 6. Finally, conclusions and future work are discussed in Section 7.

2. BACKGROUND

This section presents concepts related to iStar and Multi-Agent Systems with rational agents.

2.1. ISTAR

iStar is a goal-based modelling language proposed in the nineties [40]. It is an ML used to model software at require- ments level. It has been extended to fit several specific application areas.

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

iStar had some variations in its default syntax (e.g. Toronto iStar [40] and Trento iStar [41]). Efforts were made towards unifying the language notation and establishing a unique core. In this sense, we can refer to a work whose purpose was to analyse the iStar (Trento and Toronto) constructs variation performed by Horkoff et al. [20] and the definition of a reference metamodel [6].

In June 2016, iStar evolved to the version 2.0 [8]. It was the result of a discussion started in 2014 in the iStar com- munity about the standardisation of the language. The new version was endorsed by key players from the community, although thorough validation still needs to be performed. In this new version of the language some concepts were discontinued, some changes were done, and new concepts were introduced.

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

The INS link and Positions were not considered in this new version. Softgoal was renamed as Quality in iStar 2.0. Quality is described in [8] as an attribute for which an actor desires some level of achievement. For example, in a given model, an entity could be the system under development and the Quality its performance. Another entity could be the business being analysed, and the Quality would be the yearly profit. The level of achievement may be denied precisely or kept vaguely. Qualities can guide the search for ways of achieving goals and serve as criteria for evaluating alternative ways of achieving goals.

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

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

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

Table 1 shows a comparison between both iStar versions.

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

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

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

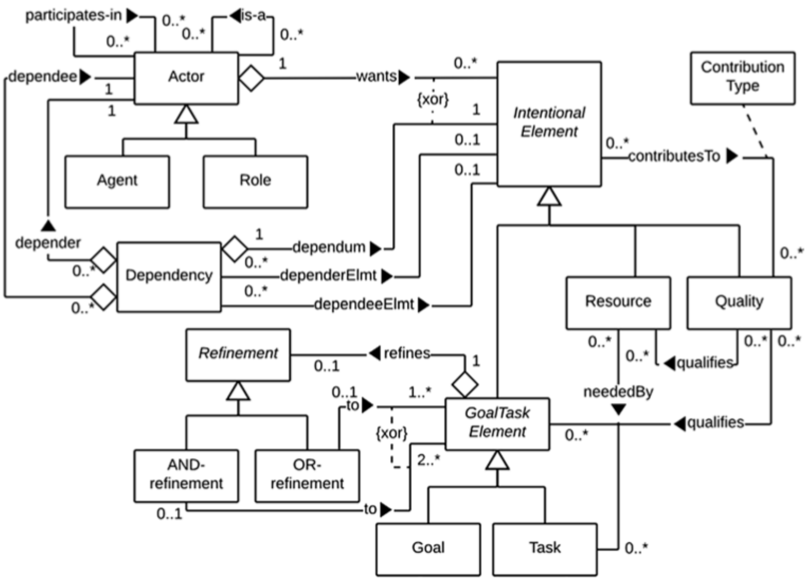


Fig. 1. Metamodel of the iStar 2.0 [8].

A running example of iStar 2.0 is presented in Dalpiaz et al. (2016) concerning University travel reimbursement. Students must organise their travel (e.g., to conferences) and have several goals to achieve, and options related to them. To achieve their goals, students rely on other parties such as a Travel Agency and the university's trip management information system. In Fig. 2 we show a final view of the example to give readers an idea of the capabilities of iStar 2.0 [8].

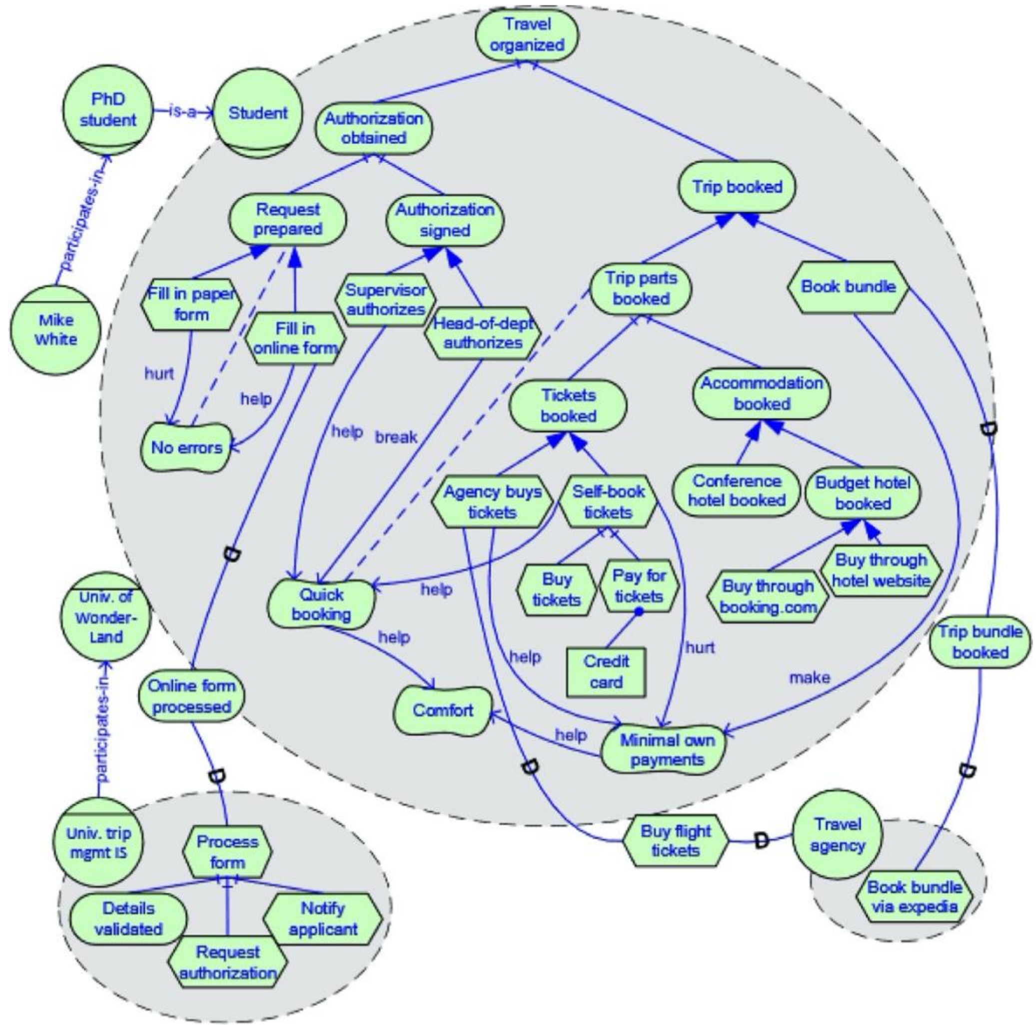


Fig. 2. A preview of the travel reimbursement scenario as captured in iStar 2.0 SR model [8].

2.2. Multi-Agent Systems with Rational Agents

Nowadays, agent technology has been widely applied to solve a vast set of problems such as those in healthcare [21], in the Internet of the Things [32] and in robotics [1].

Russell and Norvig [33] define an agent as an entity that can perceive its environment through sensors and act in the environment through actuators. Unlike objects, agents are more complex entities with behavioural properties, such as (i) autonomy (i.e., they are able to execute without interacting with humans), and (ii) interaction (i.e., they are able to interact by sending and receiving messages and not by explicit task invocation) [37].

As stated in [39], intelligent agents possess the following characteristics:

- Autonomy: agents operate without the direct intervention of humans or others and have some kind of control over their actions and internal state;
- Sociability: agents interact with other agents (and possibly humans) via some type of agent communication language;
- Reactivity: agents perceive their environment and respond in a timely fashion to changes that occur in it;
- Proactivity: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

Multi-Agent System (MAS) is the subarea of artificial intelligence that investigates the behaviour of a set of autonomous agents, aiming to resolve a problem that is beyond the capacity of a single agent [22].

The agent-oriented development paradigm requires adequate techniques to explore its benefits and features in order to support the construction and maintenance of this type of software [42]. As it is the case with any new software engineering paradigm, the successful and widespread deployment of MASs requires modelling languages that explore the use of agent-related abstractions and promote the traceability from the design models to code. To reduce the risk when adopting a new technology, it is convenient to present it as an incremental extension of known and trusted methods, and to provide explicit engineering tools that support industry-accepted methods of technology deployment [7].

A simple agent can act based on reflex or proactive behaviour. Reflex agents are more suitable to sub-problems that require quick responses. When acting together, reflex agents can achieve good results, as in the case of ant colonies [9]. On the other hand, goal-based agents are more suitable for complex problems which involve the use of complex algorithms and require more time to solve.

Agents can be classified according to their internal architecture that determines distinct agency properties, attributes and mental components [33]. Russell and Norvig [33] define four types of agents according to their internal architectures: simple reflex agent, model-based reflex agent, goal-based agent and utility-based agent. The internal architecture of an agent is selected according to the subproblem that the agent will resolve. MAS may encompass multiple types of agents with different internal architectures [38].

The agent internal architectures can be categorised based on proactive and reactive foundations. In this context, four types of internal agent architectures were defined by Russell and Norvig [33]. These architectures are detailed in the next sections.

2.2.1. Simple Reflex Agent

A Simple reflex (or reactive) agent [33] is considered the simplest internal architecture. Condition-action rules are used to select the actions based on the current perception. These rules follow the form “if condition then action” and determine the action to be executed if the perception occurs. This architecture assumes that at any time the agent receives information from the environment through sensors. These perceptions consist of the representation of state aspects that are used by the agent for making decisions. A subsystem is responsible for the decision-making, that is, responsible for processing the perception sequence and selecting a sequence of actions from the set of possible actions for the agent. The agent performs the selected action upon an environment through actuators.

2.2.2. Model-based Reflex Agent

The structure of this type of agent is similar to the simple reflex agent presented before since it deals with the information by using condition-action rules. In order to handle partially observable environments and to reach a more rational performance, the agent is also able to store its current state in an internal model.

According to Weiss [38], model-based reflex agents select actions by using the information in their internal states. A function called next function is introduced to map the perceptions and the current internal state into a new internal state used to select the next action. Such a state describes aspects of the world (called model) that cannot be seen in the current moment but were perceived earlier or have come out by inferences [33]

2.2.3. Goal-based Agent

Sometimes the knowledge about the current state of the environment is not enough to determine the next action and additional information about desirable situations is required. Goal-based agents are Model-Based Agents that set a specific goal and select the actions that lead to that goal. This allows the agent to choose a goal state among multiple possibilities.

Planning activity is devoted to finding the sequence of actions that are able to achieve the agent’s goals [33]. Goal-based agent with planning involves the next function component and also includes the following elements:

- Formulate goal function, which receives the state and returns the formulated goal;

- Formulate problem function, which receives the state and the goal and returns the problem;
- Planning, which receives the problem and uses search and/or logic approaches to find a sequence of actions to achieve a goal; and
- Action, which is represented with its preconditions and postconditions.

2.2.4. UTILITY-BASED AGENT

Considering the existence of multiple goal states, it is possible to define a measure of how desirable a particular state is. In this case, aiming to optimize the agent performance, the utility function is responsible for mapping a possible state (or group of states) to that measure, according to the current goals [33]. Thus, the utility function is incorporated into the architecture. In addition, the utility-based agent preserves the same elements as those of a Goal-based agent: next function, formulate goal function, formulate problem function, planning and action.

3. RELATED WORK

We presented here an iStar extension which is part of an MDD approach to develop MAS with rational agents. In [10], an extension to MAS-ML (Multi-Agent Systems Modeling Language [35]) to model MAS with rational agents in the architectural level is presented. The extension is supported by a tool (MAS-ML tool [11]) and illustrated by the modeling of a MAS to virtual auctions. An extension was proposed to Java Agent Development framework (JADE)[1] to support the development of rational agents and other MAS entities such as organisation, environment and agent roles [27]. In addition, the code generation from MAS-ML models created by MAS-ML tool to the JADE extended version. Finally, Silveira et al. [36] proposed an agent-based approach to select test cases and test the performance of rational agent. Interactions between agent and environment are realized in order to evaluate the agent performance for each test case. These works cover a great part of the software development life cycle, but the requirements level is not covered by them. Thus, our paper is proposed to fill this need in this approach.

On another hand, a process has been proposed to conduct the proposal of iStar extensions. PRISE (PRocess to conduct iStar Extensions) [17][18] is a well-defined process based on a Systematic Literature Review (SLR) of iStar extensions [12] and interviews and survey with experts [16]. PRISE is supported by tools such as a catalogue of iStar extensions [13] and a tool to support the creation of the artifacts of the PRISE [14].

Finally, we can highlight that intelligent system is one of the application groups which has more iStar extensions identified (19 extensions) [12]. We did not find anyone iStar extension to model rational agents.

However, we found a set of extensions which defines constructs that we selected to reuse. In Towards Augmenting Requirements Models with Preferences [26], the authors extend iStar to include the specification of optional user requirements and user preferences, aggregated together into weighted formulae to be optimized. They team this with an automated reasoning tool, adapted from state-of-the-art research in artificial intelligence planning with preferences, in order to synthesize solutions that both comply with the goals and optimize stakeholder preferences and optional requirements. The effect link is used in this extension [26] to represent Condition Elements (CE) and Effect Elements (EE) as a consequence of tasks. We consider the effect link suitable to connect perceptions and actions of the reflex agents of our approach.

TROPOS [4] is an agent-based methodology which extended iStar to model plans of agents. Plans are represented in TROPOS by tasks in blue colour. Xipho [30] is an extension of TROPOS to represent for engineering context-aware personal agents. Xipho represents plans as octagons. We consider the construct plan proposed by TROPOS and Xipho suitable to represent plans of agents of our approach. Additionally, we selected the graphical representation of plan proposed by [19] to this construct.

1 Java Agent Development. http://jade.tilab.com/

An iStar extension for generating law-compliant requirements by using a taxonomy of legal concepts and a set of primitives to describe stakeholders and their strategic goals is presente in [34]. Actions are represented by generating a set of requirements for a new system which comply with a given law. We consider the construct plan proposed by this extension.

4. ISTAR4RATIONALAGENTS

iStar4RationalAgents is an extension proposed with PRISE process. The final artefact generated by the process is available here[2]. The extension is described following based on this final version of the extension specification.

We did a literature review (not systematic) to identify the agents' architecture and select a subset of them, main references were [22], [33], [35], [37], [38] and [39]. We contacted three experts in agents when we were studying this application area.

We identified four kinds of rational agents and their roles, environment and organisation beyond their internal element to be introduced by the extension, being 19 nodes and one link. Additionally, three existing constructs of the iStar default syntax were used to connect new constructs of the extension (see Section 4.2.2). Table 2 presents the list of concepts and descriptions.

Table 2. List of concepts to be introduced with concept meanings.

Name of the Concept	Concept Meaning
	Nodes
Simple reflex agent	It is considered the simplest internal architecture. Condition-action rules are used to select the actions based on the current perception. These rules follow the form "if condition then action" and determine the action to be executed if the perception occurs. This architecture assumes that at any time the agent receives information from the environment through sensors. [33]
Model-based reflex agent	The structure of this type of agent is similar to the simple reflex agent since it deals with the information by using condition-action rules. To reach a more rational performance, the agent is also able to store its current state in an internal model select actions by using the information in their internal states. [33] and [38]
Goal-based agent	Sometimes, the knowledge about the current state of the environment is not enough to determine the next action and additional information about desirable situations is required. Goal-based agents have explicit goals and select the actions that lead to their goals. This allows the agent to choose a goal state among multiple possibilities. [33]
Utility-based agent	Considering the existence of multiple goal states, it is possible to define a measure of how desirable a particular state is. In this case, aiming to optimize the agent performance, the utility function is responsible for mapping a possible state (or group of states) to that measure, according to the current goals. [33]
Simple reflex agent role	A role related to a simple reflex agent [10] and [35]
Model-based reflex agent role	A role related to a model-based reflex agent [10] and [35]
Goal-based agent role	A role related to a goal-based agent [10] and [35]
Utility-based agent role	A role related to a utility-based agent [10] and [35]
Goal	A targeted state of the world. When goals are associated with roles, they can be of two types: right (optional) or duty (obligation) [33] and [35]
Belief	it is an internal model to storage of the history of the current states of the environment [33] and [35]
Perception	It consists of the representation of state aspects that are used by the agent for making decisions [33]
Planning	It is the activity devoted to finding the sequence of actions that can achieve the agent's goals [33]
Plan	Plan is a sequence of pre-established actions which make the agent to reach a goal [35]
Next function	The next function is introduced to map perceptions and status current internal state to a new internal state, which will be used to select the next action [33]

[2] https://www.cin.ufpe.br/~ler/iStar4rationalagents/

Formulate goal function	Receive status and return goal formulated, i.e., a subset of states of the world that meet the goal [33]
Formulate problem function	It is a function that represents the problem to be solved in a formal way [33]
Utility function	Returns the utility degree according to the current goals of the agent [33]
Action	An action that is performed by an agent. When represented in roles there is extra information if the action is mandatory (duty) or optional (right) [33] and [35]
Duty	Duty is a kind of action which the agent should perform (Mandatory) when playing a role [35]
Right	Right is a kind of action which the agent can perform or not (Optional) when playing a role [35]
Organisation	It is an element that groups agents [35]
Environment	It is an element that is the habitat for agents and organisations [35]
	Link
Cause-effect	It links the actions related to perceptions and represents the condition-action rules [33]

4.1. DEFINING METAMODEL AND VALIDATION RULES OF THE EXTENSION

We included the concepts presented in Table 2 in the iStar metamodel. We considered the relationship of extension's constructs and iStar constructs. We related the concepts to be introduced with the iStar standard concepts. Thus, we concluded that we could specialise eleven iStar constructs. Simple reflex agent, model-based reflex agent, goal-based agent and utility-based agent specialise istar agent. Next function, formulate goal function, formulate problem function, utility function and action specialise istar task. Organisation and environment specialise actor. Agent goals are represented by iStar goals without changes, agent beliefs are represented by beliefs in the iStar initial version without changes. The list of the relations between extensions and iStar constructs is given in Table 3. The concepts which specialise the iStar constructs inherit the same characteristics of the base, so the specific types of agents have a boundary, for example.

Table 3. List of the relations between extensions and iStar constructs.

Name of the Concept	Relation with iStar constructs
	Nodes
Simple reflex agent	*Specialise iStar agent*
Model-based reflex agent	*Specialise iStar agent*
Goal-based agent	*It specialises iStar agent*
Utility-based agent	*It specialises iStar agent*
Simple reflex agent role	*It specialises iStar agent role*
Model-based reflex agent role	*It specialises iStar agent role*
Goal-based agent role	*It specialises iStar agent role*
Utility-based agent role	*It specialises iStar agent role*
Goal	*It is represented by iStar goal*
Belief	*It is represented by iStar belief (from the first version of iStar). It can be related to belief by neededby relationship*
Perception	*It can be used as a source of a refinement link.*
Planning	*It specialises iStar task. It can be decomposed in actions and utility-function. It relates to goal by refinement.*
Plan	*It specialises iStar task. It can be refined in actions.*
Next function	*It specialises iStar task. It can be related to belief by neededby*
Formulate goal function	*Specialise iStar task*
Formulate problem function	*Specialise iStar task*
Utility function	*Specialise iStar task*
Action	*Specialise iStar task*
Duty	*It is a property related to Task*
Right	*It is a property related to Task*
Organisation	*Specialise iStar actor*
Environment	*Specialise iStar actor*
	Link
Cause-Effect	*Connect perceptions to actions*

Participates-in is a link of iStar 2.0 which is used to represent that an agent is part of a company (represented by another agent) and when an agent plays a role. In this way, we used this link in three more situations: i) connect an organisation and the roles that can be played in its context (representing ownership); ii) represent that an agent, organisation or role inhabit an environment (representing inhabit); and iii) link an agent and the roles that can be played by him in the context of an organisation (representing play).

The result is presented in Fig. 3. New metaclasses are represented in grey and dotted relationships connect metaclasses to the list of stereotypes.

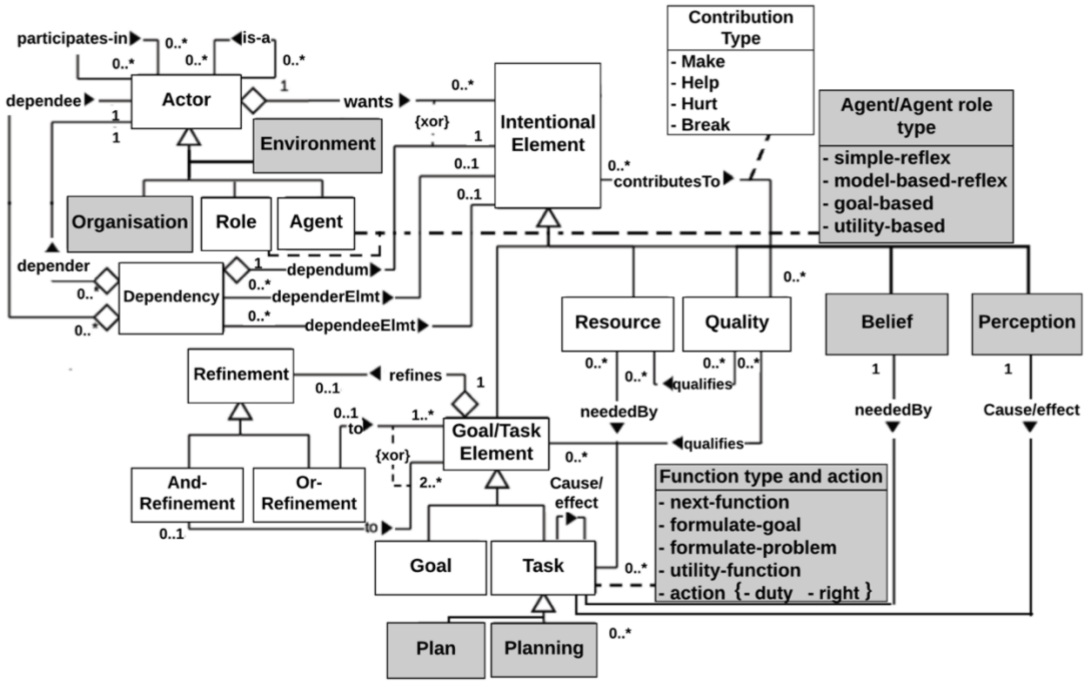


Fig. 3. Metamodel of the iStar extension to model MAS with rational agents.

We represented each kind of rational agent and their roles by a stereotype associated with Agent metaclass. Thus, we created four new stereotypes: simple-reflex, model-based-reflex, goal-based and utility-based – they are represented by the Agent/AgentRole Type. These stereotypes were applied to agent role, too. We did not create a stereotype to represent the agents of the MAS-ML work because this type of agent will be represented by an agent without any additional stereotype.

Organisation and environment are represented by new metaclasses which specialise Actor metaclass. Planning and plan are represented by new metaclasses which specialise task. Perception is a new metaclass that specialises intentional element. Therefore, belief, a metaclass that was removed in iStar 2.0, was added as an intentional element again. Finally, we represented the stereotypes of next-function, formulate-problem, formulate-goal and utility-function and action by the function types and action, duty and right related to task metaclass. The relationship neededby was extended to link beliefs and next-function tasks. The cause/effect was included to connect perception and action (represented by task metaclass) and to connect next-function and action.

We also created a set of validation rules to analyse the well-formedness of the four types of agents. We represented these rules using Object Constraint Language (OCL). Table 4 shows the validation rules applied to Agent metaclass.

Table 4. Validation rules of the agents of the new iStar extension.

Rule	OCL rule
Rule 01 – if it is a Simple reflex agent, it should have perception and task and should not have goal, belief, plan or	`self.ownedAgentSemantics->value() = 'simple-reflex'` **implies** `self.ownedperception -> isEmpty() = false` **and** `self.ownedtask -> isEmpty() = false` **and** `self.ownedgoal -> isEmpty() = true` **and** `self.ownedbelief -> isEmpty() = true` **and** `self.ownedplanning -> isEmpty() = true` **and**

planning	`self.ownedplan -> isEmpty() = true`
Rule 02 – if it is a Model-based reflex agent, it should have tasks, perceptions and belief and should not have goal, plan or planning	`self.ownedAgentSemantics->value()='model-based-reflex'` **implies** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` `self.ownedgoal -> isEmpty() = true` **and** `self.ownedplanning -> isEmpty() = true` **and** `self.ownedplan -> isEmpty() = true` **and** `self.ownedperception -> isEmpty() = true`
Rule 03 – if it is a Goal-based agent, it should have goal, planning, perception, task and belief	`self.ownedAgentSemantics->value() = 'goal-based'` **implies** `self.ownedgoal -> isEmpty() = false` **and** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` **and** `self.ownedplanning -> isEmpty() = false` **and** `self.ownedperception -> isEmpty() = false` **and** `self.ownedplan -> isEmpty() = true`
Rule 04 – if it is a Utility-based agent, it should have goal, planning, perception, task and belief	`self.ownedAgentSemantics->value() = 'utility-based'` **implies** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` **and** `self.ownedgoal -> isEmpty() = false` **and** `self.ownedplanning -> isEmpty() = false` **and** `self.ownedperception -> isEmpty() = false` **and** `self.ownedplan -> isEmpty() = true`
Rule 05 – if it is a MAS-ML agent, it should have goal, plan, task and belief	`self.ownedAgentSemantics->isEmpty() = true` **implies** `self.ownedgoal -> isEmpty() = false` **and** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` **and** `self.ownedplan -> isEmpty() = false` **and** `self.ownedplanning -> isEmpty() = true` **and** `self.ownedperception -> isEmpty() = true`

These rules stablish the internal elements allowed for each type of agent role. Rule 01 defines a simple reflex agent has having perception and task, but should not have goal, belief, planning or plan. Rule 02 defines a model-based reflex agent has having belief, perception and task, but should not have goal, planning or plan. Rule 03 and 04 defines a goal-based agent and a utility agent as having belief, goal, planning, perception and task, but should not have plan. Finally, Rule 05 stablishes MAS-ML agent has having goal, plan, task and belief and should not have planning or perception.

We also created a set of validation rules to analyse the well-formedness of the four types of agent roles. We represented these rules using Object Constraint Language (OCL). Table 5 shows the validation rules applied to agent role metaclass.

Table 5. Validation rules of the agent roles of the new iStar extension.

Rule	OCL rule
Rule 01 – if it is a simple reflex agent role, it should have task and should not have goal, belief, perception, plan or planning	`self.ownedAgentSemantics->value() = 'simple-reflex'` **implies** `self.ownedtask -> isEmpty() = false` **and** `self.ownedgoal -> isEmpty() = true` **and** `self.ownedbelief -> isEmpty() = true` **and** `self.ownedplanning -> isEmpty() = true` **and** `self.ownedplan -> isEmpty() = true` **and** `self.ownedperception -> isEmpty() = true`
Rule 02 – if it is a model-based reflex agent role, it should have task or belief and should not have goal, perception or planning	`self.ownedAgentRoleSemantics->value() = 'model-based-reflex'` **implies** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` `self.ownedgoal -> isEmpty() = true` **and** `self.ownedplanning -> isEmpty() = true` **and** `self.ownedplan -> isEmpty() = true` **and** `self.ownedperception -> isEmpty() = true`
Rule 03 – if it is a goal-based agent role, it should have goal, task and belief and should not have planning, plan or perception	`self.ownedAgentSemantics->value() = 'goal-based'` **implies** `self.ownedgoal -> isEmpty() = false` **and** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` **and** `self.ownedplanning -> isEmpty() = true` **and** `self.ownedperception -> isEmpty() = true` **and** `self.ownedplan -> isEmpty() = true`

Rule 04 – if it is a utility-based agent role, it should have goal, task and belief and should not have planning, plan or perception	`self.ownedAgentSemantics->value() = 'utility-based'` **implies** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` **and** `self.ownedgoal -> isEmpty() = false` **and** `self.ownedplanning -> isEmpty() = true` **and** `self.ownedperception -> isEmpty() = true` **and** `self.ownedplan -> isEmpty() = true`
Rule 05 – if it is a mas-ml agent role, it should have goal, task and belief and should not have planning, plan or perception	`self.ownedAgentSemantics->isEmpty() = true` **implies** `self.ownedgoal -> isEmpty() = false` **and** `self.ownedtask -> isEmpty() = false` **and** `self.ownedbelief -> isEmpty() = false` **and** `self.ownedplanning -> isEmpty() = true` **and** `self.ownedplan -> isEmpty() = true` **and** `self.ownedperception -> isEmpty() = true`

These rules establish the internal elements allowed for each type of agent role. Rule 01 defines a simple-reflex agent role as having task, but should not have goal, belief, perception, planning or plan. Rule 02 defines a model-based reflex agent as having belief and task, but should not have goal, perception, planning or plan. Rule 03 and 04 defines a goal-based agent and a utility-based agent as having belief, goal and task, but should not have planning, perception or plan. Finally, Rule 05 establishes that MAS-ML agent role has goal, belief and task and should not have plan, planning or perception.

The restrictions of internal elements of environment and organisation were made in the metamodel and it was not necessary to propose validation rules to these entities.

4.2. DEFINING CONCRETE SYNTAX OF EXTENSION

This section presents the list of the constructs of this extension, how the new graphical representations were chosen, and how to model a system using the extension.

4.2.1. LIST OF THE GRAPHICAL REPRESENTATION OF THE CONSTRUCTS

The representations of extension's constructs are presented in the next table. We used the iStar extensions catalogue to identify the representation of reused constructs and reused the representations of action and cause-effect. We also used the iStar original constructs to represent some domain concepts.

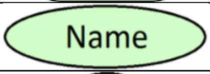
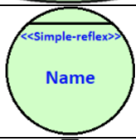
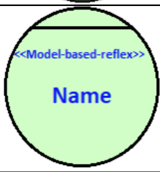
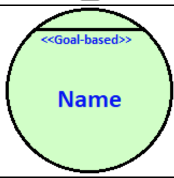
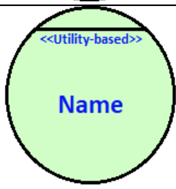
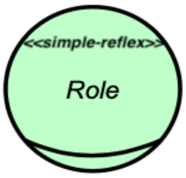
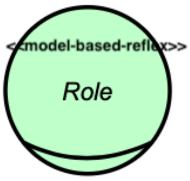
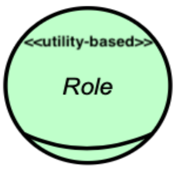
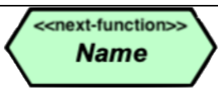
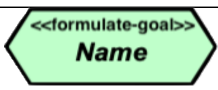
We classified the representations of the extension concepts into three groups:

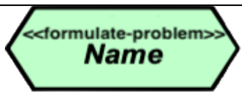
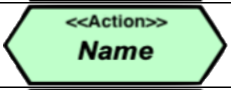
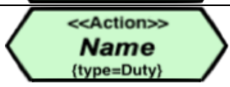
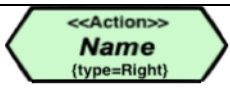
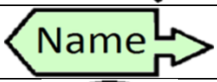
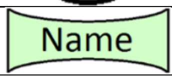
1. **Constructs represented by iStar constructs as proposed:** two domain concepts are represented by the iStar constructs. They are used by the extension but have no impact on the graphical representations;
2. **Constructs represented by iStar constructs added with textual markers:** seventeen domain concepts are represented by textual representations added to iStar constructs. These constructs have a similar meaning of the iStar constructs and specialise them by textual markers. The impact of this representations is not high since we are using the original representations, not creating new ones.
3. **Constructs represented by iStar new graphical representation:** five domain concepts are not represented by textual markers of the iStar concepts since they are conceptually different. We found the plan in previous extensions and reused it. Thus, we proposed new graphical representations for the other four constructs.

Furthermore, we extended the neededby relationship to connect next-function and beliefs establishing that beliefs are needed by the next-function. When an action is represented inside the agent roles, it can be defined as a right (a task that can be executed) using the property type='right' or a duty (a task that should be executed) using the property type='duty'. The results are presented in Table 6.

Table 6. List of the concrete syntax representation of the new iStar extension.

Name of the Concept	Representation	Explanation about the graphical representation	Reused?
Constructs represented by iStar constructs as proposed			

	Nodes		
Goal		The same construct used in the iStar 2.0	Yes
Belief		From the initial version of iStar	Yes
Simple reflex agent		The types of agents were proposed in this extension as an agent stereotype. This type of agent must have perceptions and actions into his boundary	No
Model-based reflex agent		The types of agents were proposed in this extension as an agent stereotype. This type of agent must have perceptions, next function, beliefs and actions into his boundary	No
Goal-based agent		The types of agents were proposed in this extension as an agent stereotype. This type of agent must have goals, perceptions, next function, belief, formulate goal function, formulate problem-function and actions into his boundary	No
Utility-based agent		The types of agents were proposed in this extension as an agent stereotype. This type of agent must have goals, perceptions, next function, belief, formulate goal function, formulate problem-function, utility function and actions into his boundary	No
Simple reflex agent role		The type of agent role related to simple-reflex agent	No
Model-based reflex agent role		The type of agent role related to model-based reflex agent	No
Goal-based agent role		The type of agent role related to goal-based agent	No
Utility-based agent role		The type of agent role related to utility-based agent	No
Next function		Next-function is represented as a stereotype related to a task	No
Formulate goal function		Formulate goal function is represented as a stereotype related to a task	No

Formulate problem function		Formulate problem function is represented as a stereotype related to a task	No
Utility function		Utility function is represented as a stereotype related to a task	No
Action		The action is represented as a stereotype related to a task	Yes
Duty		It is used when an action is represented in agent roles to give extra information that it is mandatory by the property {type=duty}	No
Right		It is used when an action is represented in agent roles to give extra information that it is optional by the property {type=Right}	No
Cause-effect		It uses the contribution link and adds a textual marker eff. Used to link perceptions/next-function to actions in reactive agents	Yes
Constructs represented by iStar new graphical representation			
Perception		It is based on a magnifying glass, which is used to perceive things of the environment	No
Planning		It is like a task, but there is an arrow on the right edge	No
Organisation		Organisations are actors that join agents. Thus, this representation is a specialisation of the actor with an analogy for a group of agents at the top	No
Environment		It is like a window from which it is possible to see the environment	No
Plan		It is like a task, but there is no arrow on the left edge	Yes

The details about how these graphical representations were chosen are presented in the next section.

4.2.1.1. Choosing New Graphical Representations of this Extension

The new symbols proposed by this extension for organisation, environment, perception and planning were created using an experiment based on the work of Caire et al. [5]. We performed this five-step experiment with 152 participants. We created graphical representations for these concepts to be used in studies 4 and 5 of this experiment. Seventy participants made drawings for these four concepts (Step1). Also, we analysed the drawings, grouped them and identified the stereotype of the population (the most significate drawing from the most frequent group) (Study 2). Then, we submitted questionnaires to identify the prototype (Study 3) to 29 participants, the semantic transparency (Study 4) to 48 participants and the recognition (Study 5) to 45 participants. The graphical representations of the subsets are presented in Fig. 4.

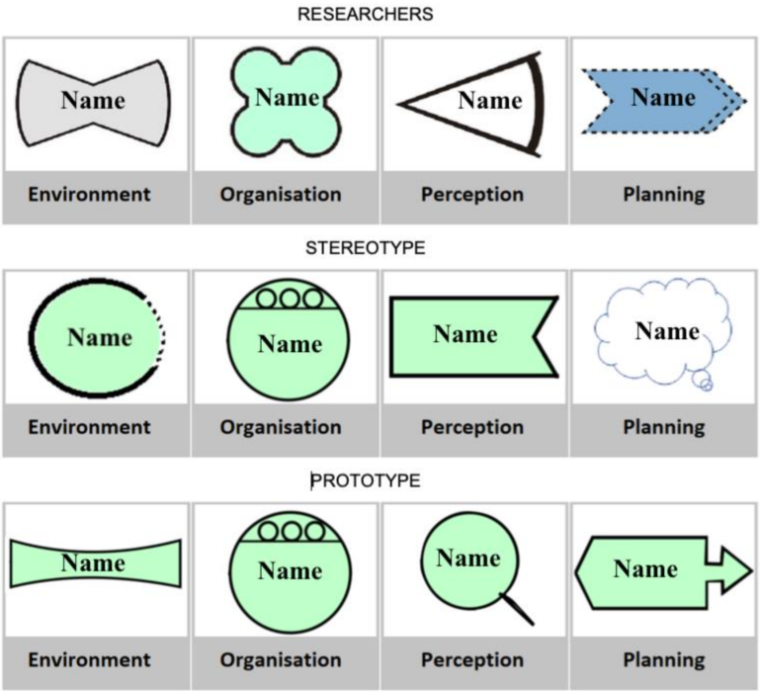


Fig. 4. Graphical representation of the constructs by the subsets.

We presented the results of the Hit rate of study 4 (Semantic transparency experiment) in the Table 7. Bolded values indicate the best hit rate of a concept among all factors.

Table 7. Hit rate (in %) means per concept per factor in semantic transparency experiment.

Concepts	**Factors**		
	Researchers	Stereotype	Prototype
Environment	26.67	22.22	**46.67**
Organisation	33.33	**61.11**	60.00
Perception	53.33	27.78	**60.00**
Planning	**53.33**	33.33	46.67
Mean	41.66	36.11	53.33
Group size	15	18	15

We presented the results of the Hit rate of study 5 (Recognition experiment) in Table 8. Bolded values indicate the best hit rate of a concept among all factors.

Table 8. Hit rate (in %) means per concept per factor in recognition experiment.

Concepts	**Factors**		
	Researchers	Stereotype	Prototype
Environment	81.25	**94.12**	83.33
Organisation	87.5	88.24	**91.67**
Perception	87.5	82.35	**91.67**
Planning	68.75	82.35	**100**
Mean	81.25	86.76	91.66
Group size	16	17	12

Finally, we selected symbols based on results of studies 4 and 5. These symbols are used in this extension to represent the new concepts. Fig. 5 shows the selected symbols.

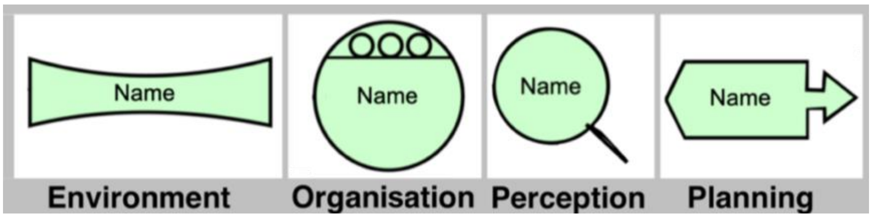


Fig. 5. Final representations.

4.2.2. DESCRIBING HOW TO USE THE EXTENSION

The SD model should be created to represent the agents, roles, organisations and environments involved in the MAS and representing the relationship between them. An agent can play a role, inhabit an environment and be part of an organisation. They are represented by the iStar link participates-in as presented in the table above. The dependencies between agents, roles, organisations and environments can be expressed too. The main objective of this modelling is representing the MAS to be developed in the early phase when the MAS is starting to be proposed and the decisions about the agents' intentional elements like goal, task, planning, perception and internal details of the agents' intentional elements are not clear yet. Fig. 6 shows an agent playing a role (i), an agent being part of an organisation (ii), an Agent inhabiting an environment (iii), an organisation inhabiting an environment (iv), an agent playing a role in an organisation which inhabits an environment (v) and a dependency between an agent and an environment. We used only simple-reflex agents and roles but these links can be used with all types of agent.

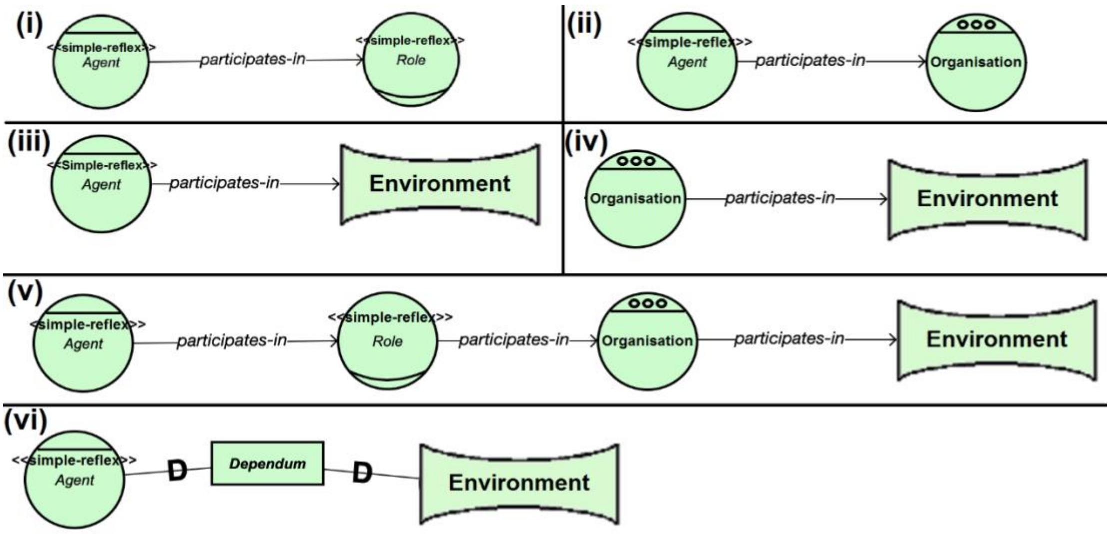


Fig. 6. Generic representations of new constructs in SD model.

Moreover, the SR model should be created to represent the internal details of the agents, roles, organisations and environments involved in the MAS, and representing the relationship between them. The SD model is the starting point to the creation of this model.

In our approach, we consider the modelling of the agents' intentional elements as a refinement of their internal elements, modelled in a similar approach to that used by [29]) and [25].

The boundaries of the agents should be detailed regarding their intentional elements according to the validation rules presented in the Table 4. The type of agent defined by MAS-ML is represented by an agent without any stereotype and its boundary is composed by goals, beliefs, plans and actions (i). In the reflex agents (simple reflex (ii) and model-based simple reflex (iii)), the perceptions and actions (and next-function in case of model-based reflex agent) should be related by the refinement link with the action as the source. Goals of the goal-based (iv) agents are decomposed on perceptions (which are decomposed on next-function) and planning (which are decomposed in actions). The same applies to utility-agents (v), but these ones have a utility-function related to the planning by an and-refinement. The beliefs are represented by a neededby link connecting beliefs and next-function. Fig. 7 presents the boundary components for each type of agent on SR diagram.

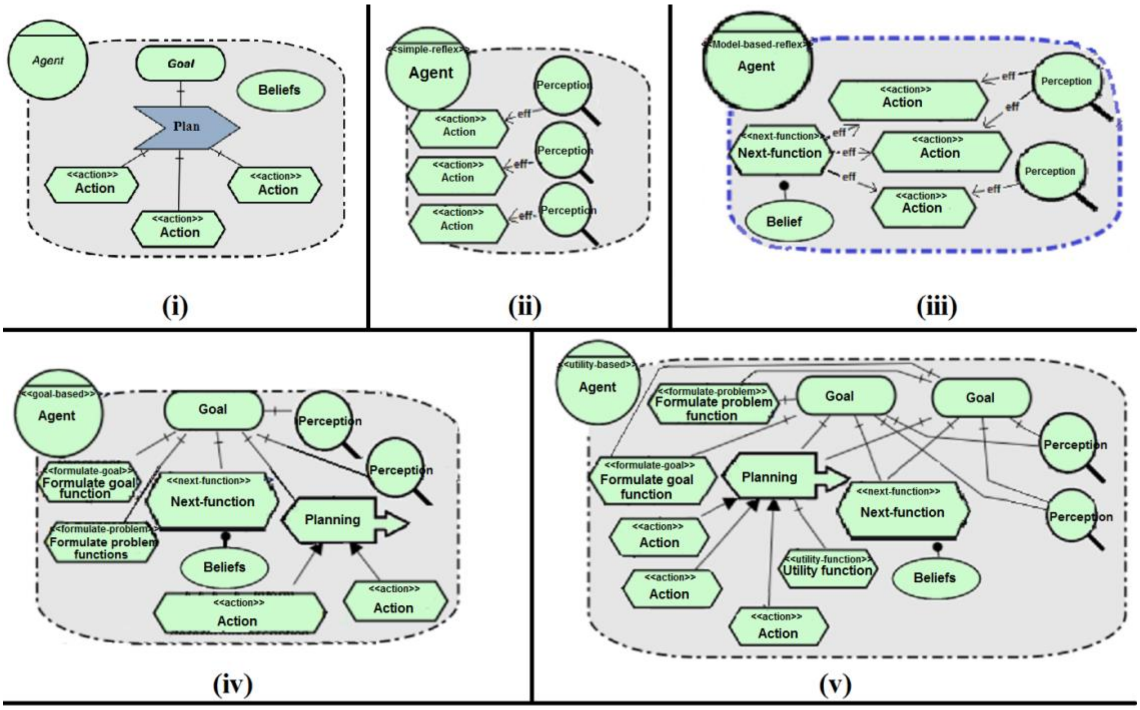


Fig. 7. Generic representations of agents in SR model.

The boundaries of roles related to goal-based and utility agents and type of agent defined by MAS-ML should represent goals, beliefs and actions related to the role. The boundary of roles of model-based-reflex agents should contain beliefs and actions related to the role and the roles of simple-reflex agents should contain actions related to the role. The actions in roles related to agents should have the information about which of them are mandatory (duty) and which are optional (right). Fig. 8 presents the boundary components for each type of agent role on SR diagram.

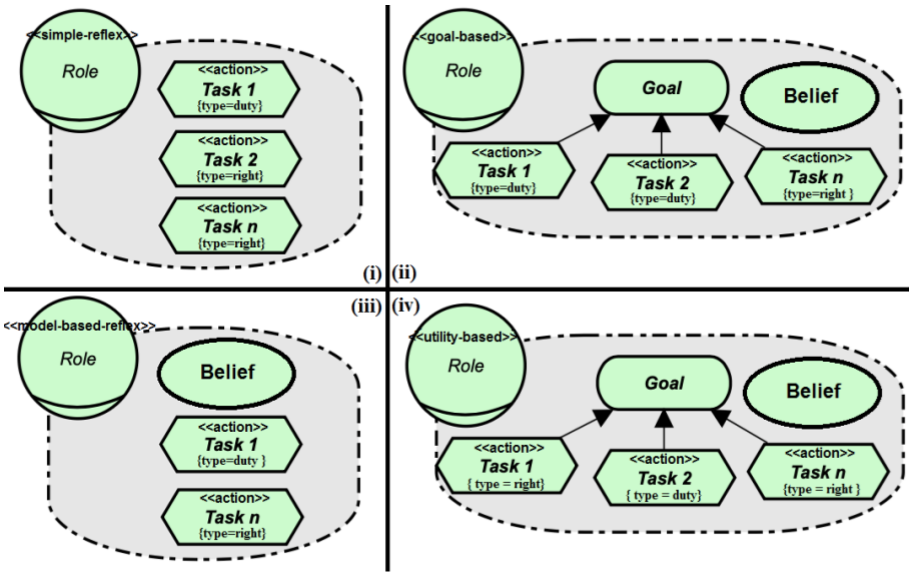


Fig. 8. Generic representations of agent roles in SR model.

Additionally, the environment and organisation can be composed of the original iStar nodes and links such as goal, quality, resource and task nodes and refinement, qualification, neededby and contribution.

Note 1: Organisations are represented in MAS-ML with the all internal elements of the agents that are part of the organisation. However, iStar has scalability problems, and the creation of a node with all elements can be a problem. Therefore, the organisation should represent additional intentional elements not present on the agents. Fig. 9 shows the generic representation of environment and organisation.

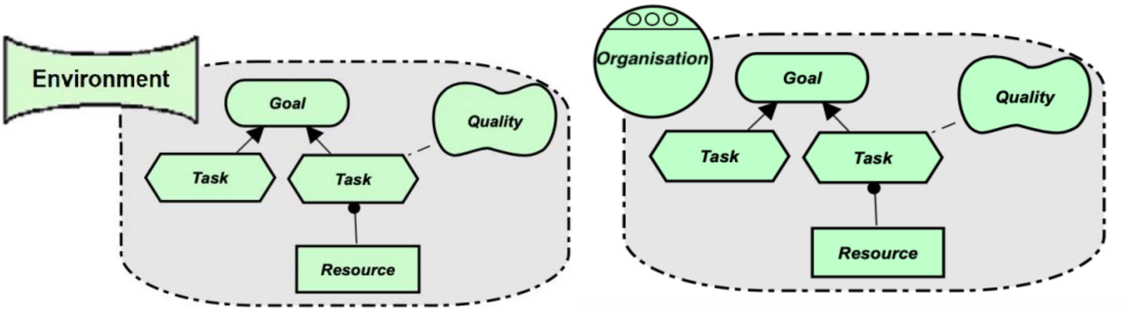


Fig. 9. Generic representations of environment and organisation in SR model.

Note 2: As mentioned in this section, we are proposing this extension in the context of a Model-driven approach which will consider MAS-ML as architectural models. MAS-ML presents more nodes to represent object and object roles, and these will be represented in iStar by Actors and roles without the need to include new stereotypes. The iStar original nodes and links are simply used in MDD approach to represent objects and roles in works such as [28].

Section 5 presents an example of the usage of this iStar extension to model a MAS to Moodle.

4.2.3. PISTAR4RATIONALAGENTS

We applied the extension in the piStar modelling tool. The extended tool was tested by the modelling of the illustration presented in Section 5. We did not found corrections to be made. The tool is available at www.cin.ufpe.br/~ler/piStar4rationalagents. Fig. 10 shows an overview of the piStar4rationalagents.

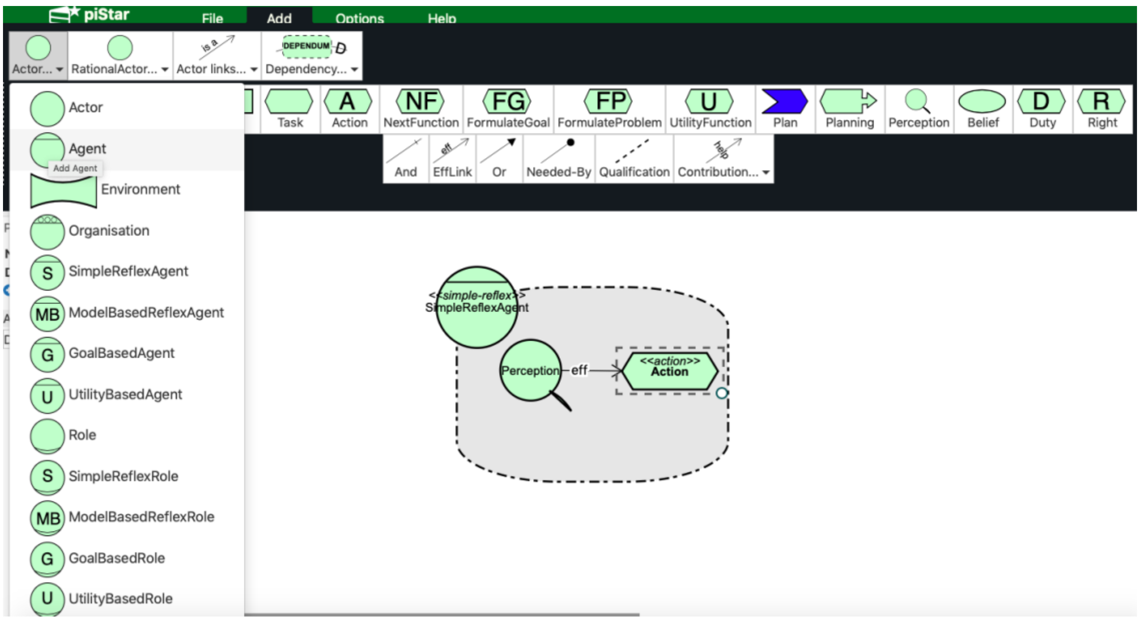


Fig. 10. Overview of piStar4rationalagents.

5. ILLUSTRATION

MOODLE (Modular Object-Oriented Dynamic Learning Environment) is widely used in distance education courses. MOODLE is free software available for download at https://www.moodle.org. It can be used without costs and has enabled the development of courses in distance education around the world. This software is available in a standard distribution which can be customised. This environment is used in courses of the Universidade Aberta do Brasil (UAB), a university of the Brazilian government that offers courses in distance education. Thus, we modelled a MAS to courses of UAB offered in partnership with the Universidade Estadual do Ceará (UAB/UECE).

Five agents were considered in the modelling:

Helper Agent: This agent is a simple reflex type and has a list of several insights into the difficulties the user has, and before this, it chooses the appropriate action. This agent realises at what point the user is and at the same time independently offers tips on how to make the best use of a particular functionality, specifically, the action for a particular job;

Companion Learning Agent: This type of agent should be able to choose independently among a predetermined range of affective interaction strategies such as messages of support. It presents encouraging messages (positive reinforcement) when the user, through the manifested interactions, gives evidence that is straightforward to follow discussion and/or the proposed tasks and/or content, and even when the student poses scores and iteration higher than the average in your class or workgroup. Because of the need to keep class notes for comparison and send messages quickly, this agent is characterised as a model-based reflex agent;

Pedagogical Agent: This agent should be able to accompany the student in the different disciplines that participate to contribute to the user through tips, suggestions and messages related to the topic ongoing and not only affective nature of messages (support). It is a goal-based agent because it needs to create a study strategy, suggesting disciplines for the student based on the disciplines that the student is doing. This agent is modelled as a goal-based agent;

Group Agent: This agent is a utility-based agent. It should be able to autonomously assist users, students and educators, the composition of working groups taking into account affinity themes or learning profiles. For this, it must consider certain criteria established by a trainer of one or more classes, or by the user interested in integrating the working groups;

Searcher Agent: This agent finds extra material (pages, projects, and other digital objects) related to the courses in the MOODLE and sent to the students. This agent is modelled as a MAS-ML default agent.

We modelled the MAS to MOODLE with our extension proposed here. Initially, the requirements engineer should model the SD-extended diagram with the representation of the type of agents with the specific stereotype. Their roles should be represented and also the organisation and environment. Finally, the participates-in link should be used to represent the roles played by the agents and the environment that they inhabit. The participates-in is also used to represent the ownership of the agents by their organisations. The SD diagram for the MOODLE MAS represents MOODLE as the environment where the agents can perceive and act. We modelled one organisation to coordinate the agents' behaviours. Finally, the diagram represents four agents and two roles performed by two of them. Fig. 11 shows the SD diagram for the MOODLE MAS.

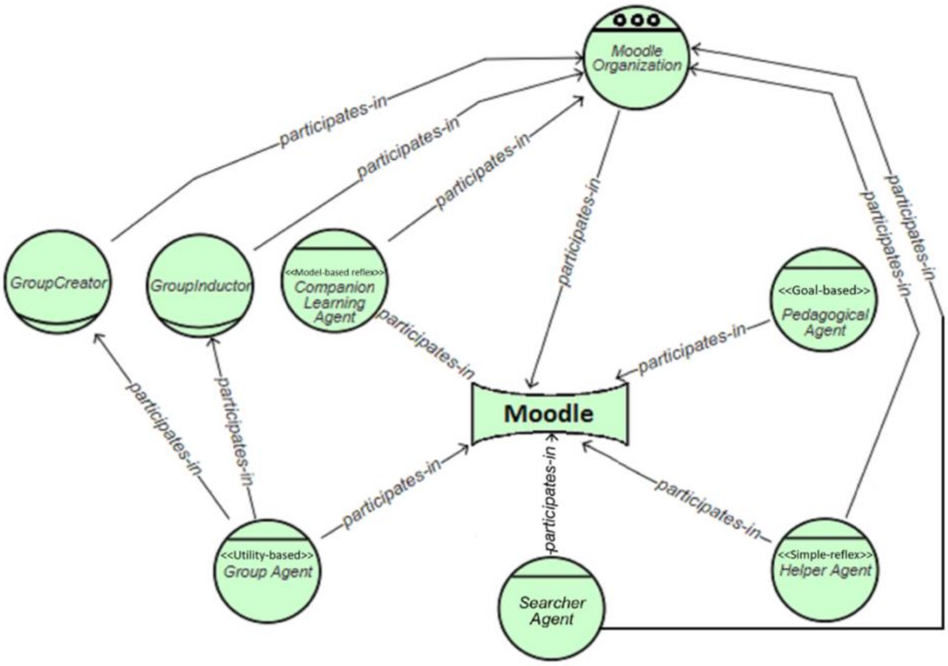


Fig. 11. Modelling of iStar SD model to MOODLE using the new iStar extension.

The SR diagram shows the internal elements of the agents. The set of related elements depends on the type of agent. The possible combinations are defined by the rules presented in Section 4.1. The perceptions of the agents are related to the environment by a dependency relationship. Fig. 12 shows the SR diagram to MOODLE MAS.

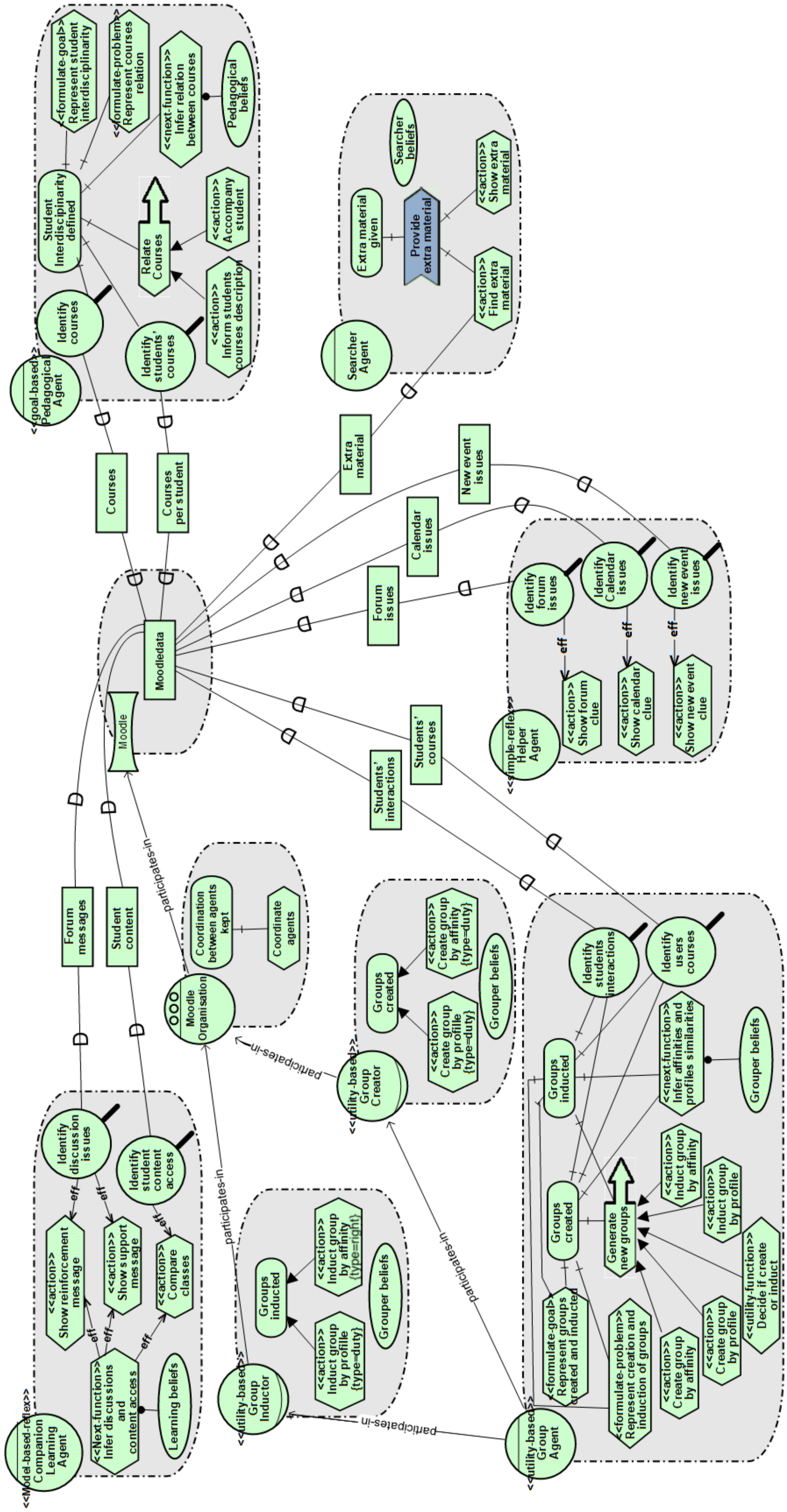


Fig. 12. Modelling of iStar SR model to MOODLE using the new iStar extension.

Additionally, the MASs presented in these models were designed at architectural level, coded, tested and implanted in two courses in distance education. The first is for the Universidade Aberta do Brasil, a federal university that offers courses in distance education, in association with Universidade Estadual do Ceará (http://www.uece.br/sate/) and the second one is the distance education in software development to people with disabilities (http://projetolead.com.br). It shows that this iStar extension is valid to be included during the development of MASs.

It was not necessary to perform tasks 4.2, 4.3 and 4.4. Task 4.5 was performed, and it is presented in the next subsection.

6. EVALUATION

[Objective] The purpose of this evaluation is to identify the point of view of the researchers in MAS extensions about this extension. It will contribute to improving the extension.

[Study Design] We used a study based on a survey with multiple-choice and open questions to analyse the opinions of the researchers in MAS about this extension. We followed the steps suggested by Kitchenham and Pfleeger [23].

[Population] The universe of this research (population) consists of researchers of MAS from Brazil. We considered the authors of papers of Brazilian events as WESAAC, AUTOSOFT, BRACIS and CBSOFT.

[Sample of Participants] We selected authors of the last five editions of these events. Thus, we contacted 164 researchers from 37 different universities/companies.

We received a total of 22 responses from 13 universities and three companies. Their profile is summarised as follows: Regarding the degree, four (4) are PhDs, nine (9) are PhD students, five (5) masters, three (3) master students and one (1) bachelor.

Regarding current work/research position, eleven (11) are professors, one (1) is a postdoc student, three (3) are system analysts/software architects, six (6) are PhD students, one (1) is a developer and a master's student, and one (1) is a master's student. Regarding the expertise level in MAS, nine (9) stated having advanced knowledge about MAS, nine (9) had an intermediate knowledge and four (4) were emerging. Fig. 13 synthetizes the level of experience in MAS of the participants.

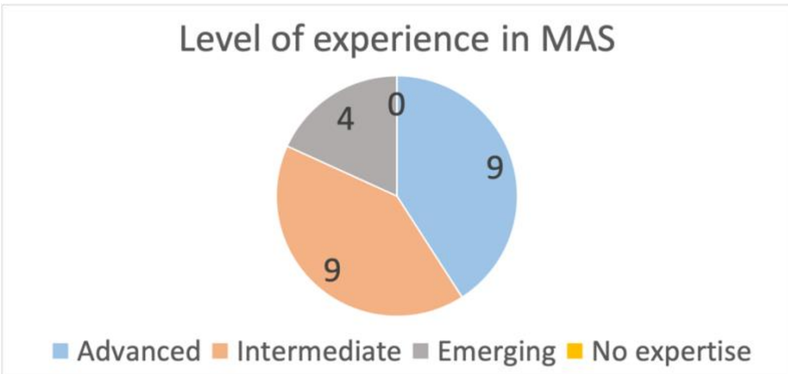


Fig. 13. Level of experience in MAS of the iStar4rational agents' evaluation participants.

Regarding the kind of expertise with MAS (the participants could choose more than one option), five (5) created textual requirements, six (6) modelled requirements (using iStar, Kaos, TROPOS or others), twelve (12) modelled the architecture (using MAS-ML, AML, ANOTE or others), twelve (12) studied an existing model, nineteen (19) coded MAS and eleven (11) tested MAS. Fig. 14 presents the kind of expertise of the participants.

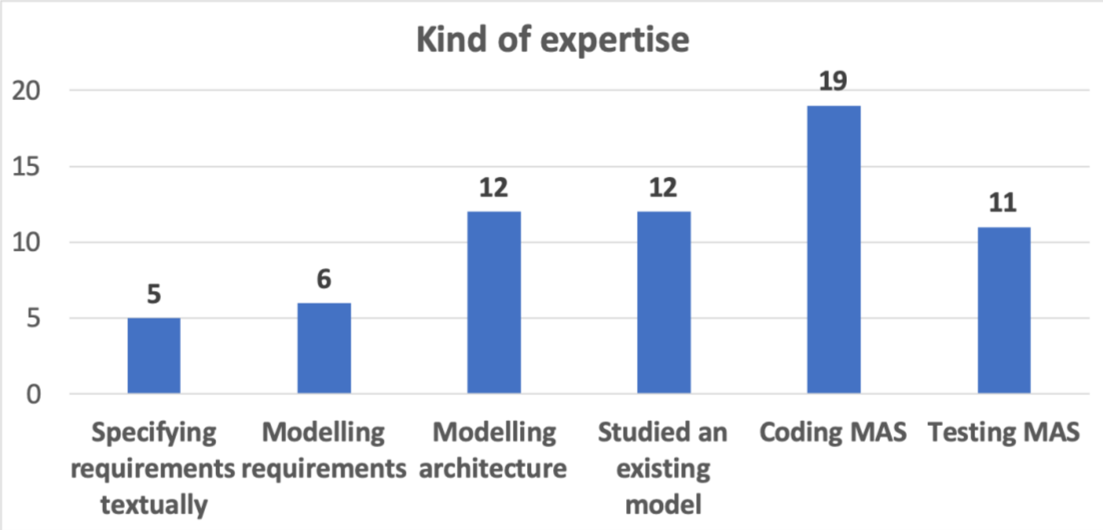


Fig. 14. Expertise of the participants of the iStar4rational agents' evaluation.

[Collection Preparation] A survey was submitted with a set of multiple-choice questions and open questions. The complete structure of this survey is available at https://www.cin.ufpe.br/~ler/iStar4rationalagents/evaluationsurvey.

Initially, we analysed the profile, background and demographics. The initial form required the following fields:

- Email, Degree (PhD, PhD student, MSc, MSc student, BSc, Current Work/Research Position (Students fill in with "student");
- Current work/research institution;
- How experienced are you with Multi-Agent Systems? (Advanced, Intermediate, Emerging, No expertise);
- Which kind of experience do you have with Multi-Agent Systems? (Creating textual requirements specification, Modelling requirements (using iStar, Kaos, TROPOS or others), Modelling the architecture (using MAS-ML, AML, ANOTE or others), Studying an existing model, Coding, Testing, Other).

The first section of the survey presents a conceptualisation about rational agents and iStar, shows the modelling of a MAS to TAC-SCM using iStar without the extension and asks some questions regarding the identification of entities in the diagrams and about the difficulty level to identify these entities.

The following questions are related to this first part. Question 1.1 asks the participant to select the correct concept related to each entity/node. We selected one construct per kind of construct in the model, presented their names and asked them to select the correct kind of concept that it represents. We used a multiple-choice grid to relate the values of the lines (Buyer agent, Production agent, Delivery agent, Manager Agent, Seller Agent, Buyer Agent, TAC-SCM, TAC-SCM Main) with columns (simple reflex agent, model-based reflex agent, goal-based agent, utility-based agent, default agent, organisation, environment, agent role).

Question 1.2 asks to select the difficulty level to identify the entities/nodes (Scale from 1-Very easy to 10-Very hard).

Question 1.3. is related to selecting the correct concept related to each Internal element (Use the scroll bar to see all options). We used a multiple-choice grid to relate the values of the lines (Offer Product (in Seller Agent), Identify Supplier Quotation (in Buyer Agent), Infer Business Similarities (in Manager Agent), Stock of parts and pcs (in Production Agent), Delivery Products (in Delivery Agent), Represent Demand (in Production Agent), Represent Maximization (in Manager Agent), Estimate Selling Price (in Manager Agent), Produce Computers (in Production Agent), Offer Bid (in Buyer), Pay (in Buyer)) with columns (action, belief, perception, plan, planning, next-function, formulate goal function, formulate problem function, utility function, duty, right).

In question 1.4, the participant should select the difficulty level to identify the agent's internal elements (Scale from 1-Easy to 10-Hard). Question 1.5 asks about the participant's opinion about the representation of MAS with rational agents using the original iStar syntax. The following options are presented: It is expressive enough to represent this kind of systems and an extension is not necessary, it is not expressive enough to represent this kind of systems and an extension is necessary or I don't know.

The second part of the survey presents the extension (According to the previous sections of this chapter), shows the modelling of an MAS to MOODLE using the iStar extension and asks some questions regarding identifying entities in the diagrams and about the difficulty level to identify these entities (questions similar to the first part). We highlight that the systems modelled in these two parts have similar complexity and number of nodes and links. Finally, the third and last part asks the opinion of the participants about the strongest and weakest points, about the need to propose the extension, the existence of problems in the extension, the usefulness, the opinion about the representation of the introduced concepts, difficulty to understand, the feasibility and intention in use, and finally, if it is useful for the iStar community and if it will be recommended to other researchers.

The survey was validated by two experts in iStar extensions (one of them has experience with MAS). We tested the survey with a computer science professor from Universidade Federal do Ceará with experience in MAS. The pilot data were not used during the analysis. [Data Collection] The survey was written in English and submitted by Google Forms between December 2018 and February 2019. [Data Analysis] We represented the results of the survey's questions by graphics and tables which present an analysis in descriptive statistics. Data of this survey is available at www.cin.ufpe.br/~ler/iStar4rationalagents/data.

6.1. RESULTS AND DISCUSSION OF THE SURVEY

In this section, we describe the results of the survey. We could not compare the data of our results and data of similar studies because this is the first study to evaluate this extension. We start our analysis comparing the perception of the participants about the modelling of the MAS using iStar and the iStar extension.

The results of the identification of MAS entities point to the following percentage of correct responses **without** the extension (first part): simple reflex agent – 27.3% (6/22), model-based reflex agent – 50% (11/22), goal-based agent – 59% (13/22), utility-based agent – 68.1% (15/22), MAS-ML default agent – 9% (2/22), agent role – 63.6% (14/22), environment – 63.6% (14/22) and organisation – 36.4% (8/22).

The results of the identification of MAS entities point to the following percentage of correct responses **with** the iStar extension (third part): simple reflex agent – 77.3% (17/22), model-based reflex agent – 72.7% (16/22), goal-based agent – 90% (20/22), utility-based agent – 72.7% (16/22), MAS-ML default agent – 72.7% (16/22), agent role – 68.2% % (15/22), environment – 90.9% (20/22), organisation – 95.4% (21/22).

The comparison between the two representations of MAS entities is summarised in Fig. 15. It is possible identify that all representations with the extension had a higher number of correct responses than the representations without the extension.

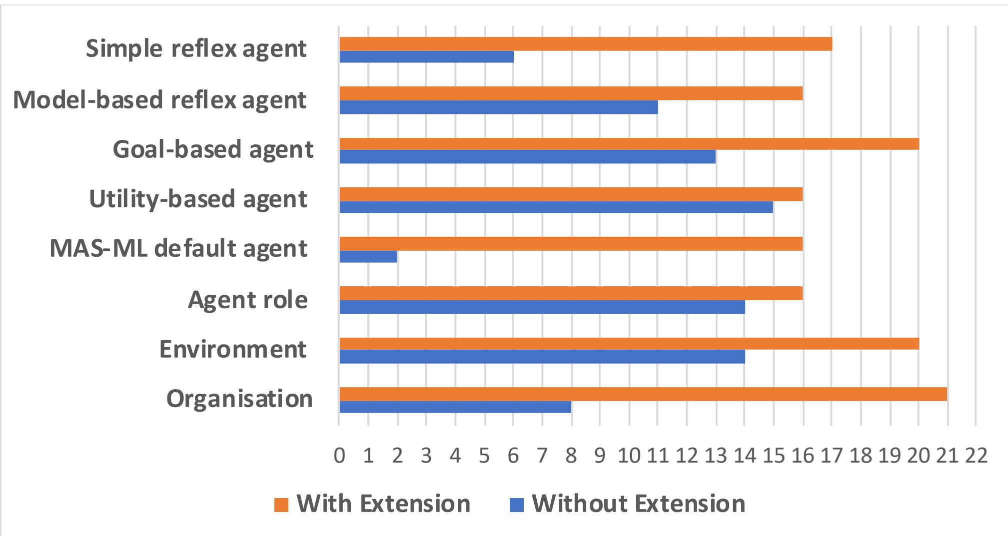


Fig. 15. Comparison between the two representations of MAS entities (with and without the extension).

We also analysed the internal elements of the agents. The results of the identification of MAS internal elements point to the following percentage of correct responses **without** the extension (first part): action – 54.5% (12/22), belief – 27.3% (6/22), perception - 31.8% (7/22), next function – 27.3% (6/22), formulate goal function – 18.2% (4/22), formulate problem function – 0 % (0/22), utility function – 9.1% (2/22), plan – 22.7% (5/22), planning – 18.2% (4/22), duty – 9.1% (2/22), right – 9.1% (2/22).

The results of the identification of MAS internal elements point to the following percentage of correct responses **with** the extension (third part): action – 90.9% (20/22), belief – 72.3% (17/22), perception – 72.7% (16/22), next function – 63.6% (14/22), formulate goal function – 63.6% (14/22), formulate problem function – 59.1% (13/22), utility function – 59.1% (13/22), plan – 68.2% (15/22), planning – 59.1% (13/22), duty – 40.9% (9/22), right – 27.3% (6/22).

The comparison between the two representations of MAS internal elements is summarised in Fig. 16. It is possible identify that all representations with the extension had a greater number of correct responses than the representations without the extension.

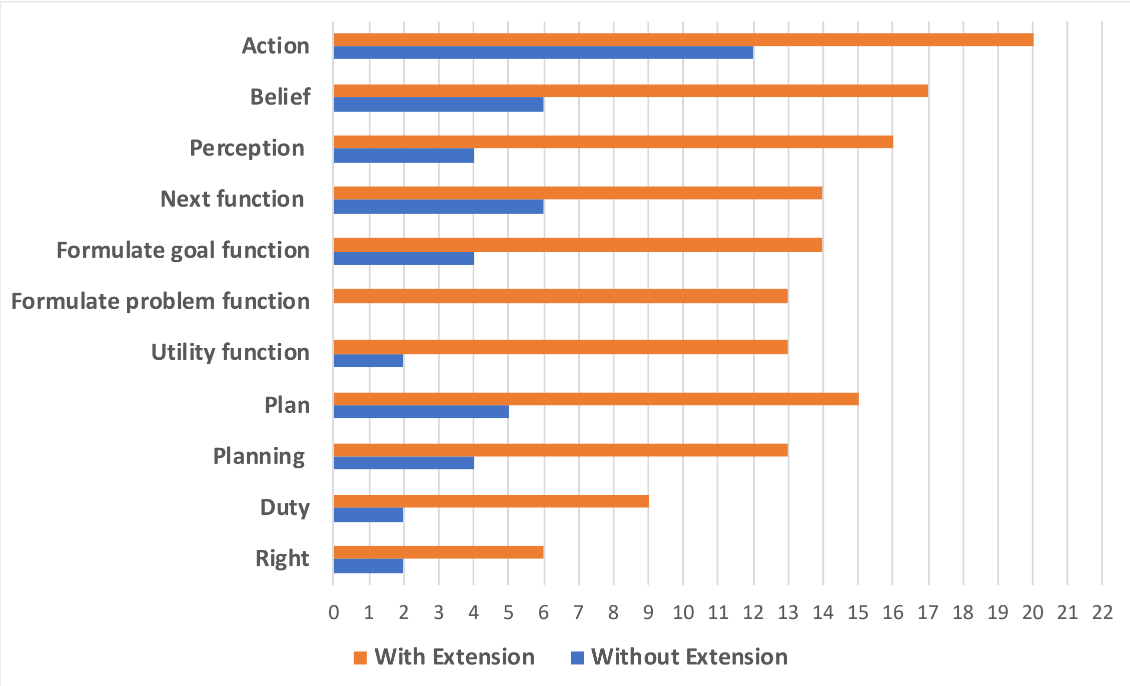


Fig. 16. Comparison between the two representations of MAS internal elements (with and without the extension).

The extension improved the perception of the MAS constructs introduced by the extension. Table 9 shows the general results about the correct identification of the entities to iStar without extension and with extension.

Table 9. Comparison of correct identification.

Criteria	iStar without extension	iStar extension
1. Correct identification of MAS entities (agents, agent role, environment and organisation)	47.12%	80.1%
2. Correct identification of MAS internal nodes and links (plan, planning, action, belief, next-function…)	20.7%	62%

We also analysed the difficult level perceived by the participants to identify the MAS entities and internal nodes and links. The extension reduced the difficulty level to identify the MAS entities and internal nodes and links by about 50%. Table 10 shows the results of the difficulty level without extension and with extension.

Table 10. Comparative of the difficulty level.

Criteria	iStar without extension	iStar extension
1. Difficulty level to identify MAS entities (mean – scale 0–10)	6.3	2.9
2. Difficulty level to identify MAS internal nodes and links (mean – scale 0–10)	7.5	3.8

The last question of the first part of the survey concerned the opinion about the representation of the rational agents with the iStar default syntax (without the extension). Of the participants, 72.8% (16/22) responded that It is not expressive enough to represent this kind of systems and an extension is necessary and 13.6% (3/22) consider It is expressive enough to represent this kind of systems and an extension is not necessary and 13.6% (3/22) answered don't know.

The second part consists of the presentation of the extension. Consequently, there are no responses to this part.

The third part is composed of three initial open questions to reveal the strong points and weaknesses, improvements and identification of problems. They recognised that the extension facilitates the identification of MAS entities and their internal elements; this is a common strong point mentioned by 13/22 participants. The weakness mentioned by two participants is that with the extension, new concepts are represented in iStar and there is the need to learn and represent these elements in the models. We believe that it is not something related to this extension, but it is a general consequence of all extensions. One participant questioned us about the representation of the protocol of communication between different MAS. We understand that it is part of the representation of a MAS, but this representation is considered only at the architectural level in our approach. The participants did not identify any problem in the extension.

The extension received a good evaluation of the experts who participated in the survey. Table 11 shows the results for questions 3.4 – 3.6, 3.8, 3.10 and 3.11. Median, mode and number of responses received for each option (Strongly Agree, Agree, Don't Know, Disagree and Strongly Disagree) were presented for each question.

Table 11. Results of the validation survey of the questions 3.4 to 3.6, 3.8, 3.10 and 3.11.

Questions	Median	Mode	Strongly Agree	Agree	Don't Know	Disagree	Strongly Disagree
3.4. There is a lack of an iStar extension to model rational agents.	Agree	Agree	5	9	7	0	1
3.5. The proposal of an iStar extension to model rational agents is useful.	Agree	Agree	10	10	2	0	0
3.6. The representations of the new constructs are suitable.	Agree	Agree	7	15	0	0	0
3.8. The use of this extension is feasible for modelling Multi-Agent Systems with rational agents.	Agree	Agree	8	13	1	0	0
3.10. The Multi-Agent Systems community can benefit from the usage of this extension in the modelling of their next systems.	Agree	Agree	7	13	2	0	0
3.11. This iStar extension can be useful to model my next MAS.	Agree	Agree	5	13	4	0	0

The participants answered about how suitable the proposed extension is to be used to model the next MAS, with the results that five (5) participants consider Very Suitable, sixteen (16) consider Suitable and one (1) Don't know. Also, the difficulty level to understand this extension (question 3.7) was mentioned by the participants: two (2) very easy, twelve (12) easy, six (6) medium, two (2) hard and none answering very hard. Finally, 86.4% (20/22) of the participants mentioned the intention to recommend this iStar extension to other researchers (question 3.12), while 13.6% (3/22) of the participants mentioned maybe recommending it.

6.2. THREATS TO VALIDITY

According to Kitchenham and Pfleeger [23], there are four aspects that we need to consider: Criterion Validity, Construct Validity, Face Validity and Content Validity.

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

Criterion validity. We did not find a previous qualitative study for this purpose. Thus, we could not compare this evaluation with earlier studies.

Construct validity. We created the survey with different kinds of questions: Likert Scale questions (3.4 – 3.11), Measure effort or contribution questions (1.2 and 2.2), Yes/no/maybe question (3.12), Open questions (3.1, 3.2 and 3.3), Specific options (1.5) and multiple-choice relation questions to analyse the understanding of the models (1.1, 1.3, 2.1, 2.3). Thus, it could make the execution confusing. We mitigated this threat to construct validity by presenting an explanation of the kind of questions at the beginning of the survey to clarify the structure of the survey to the participants. Furthermore, there were invited participants whose first language is English. Consequently, we created the questions of the survey in English. It could cause misunderstandings for other participants whose first language is Portuguese. However, we did not receive any comments about these threats to construct validity during the pilot or the execution.

Face validity. We tested the survey with a computer science professor of the Universidade Federal do Ceará with experience in MAS. We can consider this previous evaluation a limitation because of the small number of participants (1 participant). We mitigated this threat, however, by asking him to evaluate again after the corrections the participant's comments.

Content validity. We performed the pilot involving one researcher. It was done to test the understanding of the participant in the survey. We analysed the feedback sent by the participant of the pilot and applied the suggested improvements to the survey.

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

We consider the profile of the participants suitable for this evaluation since a great number of the participants (18/22) mentioned have advanced/intermediate expertise in MAS and none answered they did not know MAS. However, the answers about the kind of knowledge in MAS pointing to the creation of textual requirements / model requirements were mentioned by eleven (11) participants and twelve (12) mentioned that they had studied models. Thus, there was not a high number of participants with expertise in modelling (about half); this result could be viewed as a threat to content validity. We tried to mitigate this threat presenting iStar in part of the survey.

Conclusion validity. In this survey, we did not have a large number of participants. We could not then make statistical inferences or reveal a true pattern in the data. Furthermore, we restricted the responses to the Brazilian community. These threats to the conclusion validity can be mitigated by invitation of researchers from other countries as a next step.

We did not identify new constructs to be introduced (Task 5-Main process). Therefore, the last sub-process was performed (sub-process 6 – Publicise the iStar extension).

7. CONCLUSION AND FURTHER RESEARCH

In this paper, we presented an iStar extension to model MAS with rational agents in requirements level, its validation by modeling of a MAS for distance education and an evaluation by survey with experienced participants.

We followed PRISE, a process to conduct iStar extensions, during our proposal. We represented the constructs as a set of stereotypes and four constructs were repre-sented as new symbols proposed by an experiment similar to the presented by Caire et al. [5]. This approach is supported by iStar4rationalagents tool.

We illustrated our proposal by modelling of a MAS with rational agents a course in distance education offered in MOODLE. We modeled a MAS with five different kind of agents, agent roles, organization and environment.

Finally, our proposal was evaluated by experienced researchers/developers of MAS using a survey. We identified that the extension can easier the identification of the kinds of agents and their elements and can make the interpretation of the diagrams better than using iStar without extension. The participants agreed that the proposal of an extension is useful to fill a lack of techniques to represent the MAS with rational agents, the representations of the constructs were considered good and the extension could be useful to model their next MAS.

This extension could be used to model other MAS. We modeled a MAS to TAC-SCM during the application of the survey. Thus, we intend to publish these results as a next step to illustrate another example of usage of this extension. We are working in the proposal of rules to transform iStar4RationalAgents models to MAS-ML 2.0 models.

ACKNOWLEDGMENTS

The authors thank all participants of this study. We also thank CNPQ/Brazil (Conselho Nacional de Desenvolvimento Científico e Tecnológico), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), FACEPE (Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco) and NOVA LINCS UID/CEC/04516/2019 for the financial support to the execution of this work.

REFERENCES

[1] Anjomshoae, S.; Najjar, A.; Calvaresi, D.; Främling, K. Explainable Agents and Robots: Results from a Systematic Literature Review. 18th International Conference on Autonomous Agents and MultiAgent Systems, pp 1078-1088, 2019.

[2] Boehm, B. W. Software Risk Management: Principles and Practices. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research, v 8, n 1, pp 387, 2007.

[3] Brambilla M, Cabot J, Wimmer M, Model-driven software engineering in practice. In: Morgan and Claypool publishers series synthesis lectures on software engineering, 2012.

[4] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J. Tropos: An Agent-Oriented Software Development Methodology, Autonomous Agents and Multi-Agent Systems Journal, v 8, n 3, pp 203-236, 2004.

[5] Caire, P.; Genon, N.; Heymans, P.; Moody, D. Visual Notation Design 2.0: Towards User Comprehensible Requirements Engineering Notations. 21st IEEE International Requirements Engineering Conference (RE), pp 115-124, 2013.

[6] Cares C, Franch X, A metamodeling approach for i* model translations. In: 23th international conference on advanced infor- mation systems engineering. Lecture notes in computer science, volume 6741. Springer, Berlin, pp 337–351, 2011.

[7] Castro, J.; Alencar, F.; Silva, C. Agent-oriented Software Engineering. In: Breitman, K., Anido, R. (Eds.), Updates in Computers. PUC-Rio, Rio de Janeiro, pp. 245– 282, 2006.

[8] Dalpiaz F, Franch X, Horkoff J (2016) iStar 2.0 language guide. arXiv:1605.07767. Available in https://sites.google.com/site/istar language/. Accessed 20 July 2017

[9] Dorigo, M.; Stützle, T. Ant Colony Optimization. The MIT Press, Cambridge, Massachusetts, 2004.

[10] Gonçalves, E.; Cortés, M. I.; Campos, G.; Lopes, Y.; Freire, E.; Silva, V. T.; Oliveira, K.; Oliveira, M. A. MAS-ML 2.0: Supporting the Modelling of Multi-Agent Systems with Different Agent Architectures. The Journal of Systems and Software, v 108, pp. 77-109, 2015.

[11] Gonçalves, E., Oliveira, K, Cortés, M. I., Feijó, A. R., Oliveira, F. R., Silva, V. T.: MAS-ML TOOL - A Modeling Environment for Multi-agent Systems. 13th International Conference on Enterprise Information Systems, p. 192, 2011.

[12] Gonçalves, E., Castro, J., Araujo, J., Heineck, T. A Systematic Literature Review of iStar extensions. Journal of Systems and Software, v. 137, p. 1-33, 2018.

[13] Gonçalves, E.; Heineck, T.; Araujo, J.; Castro, J. CATIE: A Catalogue of iStar Extensions. Cadernos do IME. Série Informática, v. 48, pp. 23-37, 2018.

[14] Gonçalves, E., De Oliveira, L., Heineck, T., Araujo, J., Castro, J. PRISE Tool: A tool to support the proposal of iStar extensions based on PRISE, 22nd Workshop on Requirements Engineering, 2019.

[15] Gonçalves, E.; Araujo, J.; Castro, J. iStar4RationalAgents: Modeling Requirements of Multi-Agent Systems with Rational Agents. 38th International Conference on Conceptual Modelling, 2019.

[16] Gonçalves, E., Monteiro, I., de Oliveira, M. A., Castro, J., Araujo, J. Understanding What is Important in iStar Extensions Proposals: The Viewpoint of Researchers, Requirements Engineering Journal, 24, 55–84, 2019.

[17] Gonçalves, E.; Araujo, J.; Castro, J. A Process to Support the Creation of iStar Extensions. 35th ACM/SIGAPP Symposium on Applied Computing, Requirements Engineering Track, 2020.

[18] Gonçalves, E.; Araujo, J.; Castro, J. PRISE: A Process to Support iStar Extensions. Journal of Systems and Software (To appear), 2020.

[19] Gonçalves, E., Almendra, C., Goulão, M., Araújo, M., Castro, J. Using Empirical Studies to Mitigate Symbol Overload in iStar Extensions, Software and Systems Modelling, 2019.

[20] Horkoff J, Elahi G, Abdulhadi S, Yu E, Reflective analysis of the syntax and semantics of the i* framework. In: 27th International conference on conceptual modeling, lecture notes in computer science, volume 5232. Springer, Berlin, pp 249–260, 2008.

[21] Isern, D.; Moreno, A. A Systematic Literature Review of Agents Applied in Healthcare. Journal of Medical Systems, v. 40, n. 2, 2016.

[22] Jennings, N.R. Coordination Techniques for Distributed Artificial Intelligence. Foundations of Distributed Artificial Intelligence. Wiley, pp. 187–210, 1996.

[23] Kitchenham, B.; Pfleeger, S. Principles of Survey Research, Software Engineering Notes, v. 27, n. 5, pp. 1- 20, 2002.

[24] Kotonya, G.; Sommerville, I. Requirements Engineering: Processes and Techniques. John Wiley & Sons. ISBN 0-471-97208-8, 1998.

[25] Lapouchnian, A.; Lespérance, Y. Modelling Mental States in Agent-Oriented Requirements Engineering. 18th International Conference on Advanced Information Systems Engineering, Lecture Notes in Computer Science, pp. 480-494, 2006.

[26] Liaskos, S., Mcilraith, S., Mylopoulos, J. Towards Augmenting Requirements Models with Preferences. 24th IEEE/ACM International Conference on Automated Software Engineering, pp. 565-569, 2009.

[27] Lopes, Y. S.; Cortés, M. I.; Gonçalves, E.; Oliveira, R. JAMDER: JADE to Multi-Agent Systems Development Resource. Advances in Distributed Computing and Artificial Intelligence Journal, v. 7, n. 3, pp. 63-98, 2018.

[28] Melo, J.; Brito, A.; Sa Filho, C.; Junior, J.; Alencar, F. Formalização de Regras de Mapeamento de i* para Diagrama de Classe em UML. 29th Simpósio Brasileiro de Engenharia de Software, 2015.

[29] Mouratidis, H.; Giorgini, P. Secure Tropos: A Security-Oriented Extension of The Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering, v. 17, n. 2, pp. 285-309, 2007.

[30] Murukannaiah, P.; Singh, M. Xipho: Extending Tropos to Engineer Context- aware Personal Agents, 13rd International Conference on Autonomous Agents and Multi-agent Systems, pp. 309-316, 2014.

[31] Nuseibeh, B.; Easterbrook, S. Requirements Engineering: a Roadmap. Proceedings of the 22nd Conference on the Future of Software Engineering, pp. 35–46, 2000.

[32] Pico-Valencia, P.; Holgado-Terriza, J. A. Agentification of the Internet of Things: A Systematic Literature Review. International Journal of Distributed Sensor Networks, v.14, n. 10, pp. 1-20, 2018.

[33] Russell, S.; Norvig, P. Artificial Intelligence: A Modern Approach, 2nd ed. Prentice Hall, Upper Saddle River, NJ ISBN 0-13-790395-2, 2003.

[34] Siena, A., Mylopoulos, J., Perini, A., Susi, A., 2009. Designing law-compliant software requirements. In: International Conference on Conceptual Modelling.

[35] Silva, V. T. From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language, PhD thesis in Informatics. Rio de Janeiro: PUC, 2004.

[36] Silveira, F. R. V.; Campos, G. A. L.; Cortes, M. I. A Problem-Solving Agent to Teste Rational Agents: A Case Study with Reactive Agents. 16th International Conference on Enterprise Information Systems, Lisboa, 2014.

[37] Wagner, G. The Agent-Object-Relationship Meta-Model: Towards A Unified View of State and Behaviour. Information Systems, v. 28, n. 5, pp. 475–504, 2003.

[38] Weiss, G. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Massachusetts, 1999.

[39] Wooldridge, M. An Introduction to Multiagent Systems. John Wiley & Sons, 2009.

[40] Yu E, Modelling strategic relationships for process reen- gineering. Ph.D. Thesis on Computer Science, University of Toronto, 1995.

[41] Yu E., Towards modelling and reasoning support for early phase requirements engineering. In: 3rd IEEE international sym- posium on requirements engineering, pp 226–235, 1997.

[42] Zambonelli, F.; Jennings, N.; Wooldridge, M. Organizational Abstractions for the Analysis and Design of Multi-Agent Systems. In: International Workshop on Agent-Oriented Software Engineering, LNCS 1957. Springer, Berlin, pp. 235–251, 2000.

[43] Zave, P. Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys, v. 29, n. 4, pp. 315-321, 1997