

## Generating user stories and acceptance criteria through extensions to the iStar framework

Laura Narváez<sup>1</sup>, Francy Muelas<sup>2</sup>, César Pardo<sup>3\*</sup>

<sup>1,2,3</sup> University of Cauca, GTI Research Group, Calle 5 # 4, 70, 19001, Popayán, Colombia

\*Corresponding author E-mail: cpardo@unicauca.edu.co

### ABSTRACT

1. Introduction: Currently, user stories and acceptance criteria are the most used primary artifact for the documentation and specification of requirements in the software development life cycle, mainly due to their concise notation and natural language. However, these artifacts are usually poorly written, generating process debts, and therefore, quality defects in the system that is developed. To minimize this problem, the capabilities of the iStar Framework—which is a framework for objective-oriented requirements modelling—have been extended, defining 9 transformation rules, a process and an application example, which will support the transformation of iStar models in models adapted to the structure of user stories and acceptance criteria according to the Gherkin format. 2. Method: For this, a systematic and detailed process was followed made up of process elements as well as roles, subprocesses, activities and input and output artifacts. Likewise, the evaluation of the proposal was carried out through a focus group made up of professionals who are experts in agile approaches, user stories and acceptance criteria. 3. Results and discussion: The results show that the participants determined that the proposed rules and the process defined for their application are clear, complete, suitable and applicable in agile projects, and that in addition; provides a valuable tool to companies to improve the identification, documentation, specification and development of functional requirements. 4. Conclusions: The proposed transformation rules, process and application example allow professionals in software organizations to have a graphic and general vision of the relationships between actors and activities within a software system, which can be translated into the generation of specifications. of functional requirements through user stories and acceptance criteria minimizing documentation defects.

**Keywords:** Agile Project Management, Agile Software Development, Software Development, iStar Framework, User Stories, Acceptance Criteria, Gherkin, GORE.

### 1. Introduction

A critical phase in the software development lifecycle is Requirements Engineering (RE), which is responsible for the identification, documentation, and development of captured requirements [1]. According to Johnson [2], the correct elicitation of requirements is one of the most critical aspects of any software project, regardless of its type. The inadequate capture of these requirements is the main cause of many problems that arise throughout the project life cycle, including rework and high costs [3], [4], ambiguities and misunderstandings [5], [6], customer dissatisfaction and low product quality [7], [8], impact on the maintainability and evolution of the software [9], [10], project delays and problems in its management [11], [12], [13], among others. This deficiency in requirements specification is a significant reason why many software projects do not achieve the expected success. A study presented in [14] revealed that 56% of defects found during the testing process are due to errors introduced in the analysis phase, because of poorly written, ambiguous, or incomplete requirements.

Consequently, according to [15] and [16], in environments with agile approaches and even in those companies that still use traditional approaches, user stories (onwards; US) are the most widely used. This is evidenced, for example, in a survey conducted by VersionOne [16], where it was found that 99% of professionals working with Scrum use US, mainly as the primary requirement artefact and unit of the functionality of the project [17], [18]. Their use is based on their concise notation and their aim to express requirements in a non-abstract way



using natural language [19]. In addition, most US focus exclusively on the end-user as the main stakeholder [20]. Although there are various formats for documenting the US, most authors recognize three basic characteristics: (i) short and explicit texts describing requirements [21], (ii) discussions between stakeholders to exchange perspectives on the US [22] and (iii) scenarios or acceptance criteria (onwards; AC) [20].

Although the use of user stories and the establishment of AC using the Gherkin template [23] offer great benefits, these artefacts are often poorly written in projects and can present process debts, which can lead to quality defects in the developed system [2], [24], e.g., defects such as ambiguity and incompleteness in the documentation are pervasive in software requirements, often due to the limited time practitioners spend writing good requirements [25]. The following example shows a poorly written user story: the user should be able to search for things on the website, its main shortcomings are: (i) lack of clarity, (ii) lack of value, and (iii) ambiguity, these aspects are described below in detail, e.g., lack of clarity, can be seen in: (i) it is not clear what kind of “things” the user can search for, products, items, people, etc.? (ii) How should the search be performed? through keywords, filters, etc.? and (iii) Where will the search results be displayed? Regarding the absence of value, there is no explanation of the benefit to the user or the problem that the search solves. Regarding ambiguity, the wording is too general and does not provide enough information for the development team to understand what to implement.

Writing user stories well will help ensure that the development team understands user needs, prioritization of features, and proper tracking of development progress. A poorly written user story creates confusion and hinders development, which is why user stories must be clear, concise, and value-focused to foster project success. Some of the fundamental drivers of this failure include: (i) insufficient understanding by analysts and developers of the needs and wants of the organisation or customer [26]; (ii) lack of clarity from the customer about what they want, which can lead to requirements that do not reflect their real needs [27]; (iii) constant changes that are not properly managed [28], [29]; and (iv) process debts, such as the periodic absence of inspection and process improvement activities [28], [30].

To address the problems associated with functional requirements specification, goal-oriented models have emerged and have proven to be useful for capturing, understanding, and communicating requirements during the early stages of software development. However, the usefulness of goal-oriented models is greatly enhanced when they can be leveraged in the later stages of the requirements analysis process, such as requirements elaboration, validation, and planning. These models can also be used throughout the entire system lifecycle, including the architecture and design of behavioural processes, coding, testing, monitoring, adaptation, and evolution [31], [32]. Requirements engineering using goal-oriented modelling is known as Goal-Oriented Requirement Engineering (GORE). This approach focuses on the intentionality and relationships between the different actors in a system.

### **1.1. Background**

From the results obtained in a systematic mapping of the literature presented in [33], it was possible to observe several efforts in the development of solutions to the problems associated with requirements specification using Goal-Oriented Requirements Engineering (GORE). Some of these efforts include: (i) the transformation of user stories to Goal Net-based models [34] and an approach for transforming user stories into Goal Net models, facilitating the transition from requirements to architectural design by capturing goals and scenarios [35]; (ii) the use of agent-oriented modelling in agile requirements engineering [36]; (iii) the increase of readability and usability in requirements modelling and/or specification through the use of models such as iStar and Tropos [31] and [37]; (iv) the specification of guidelines for transforming the iStar model into user stories [38]; (v) an approach for mapping user stories to iStar models and vice versa [39] and (vi) the development of a tool that automatically generates iStar models from user stories, improving traceability and understanding of requirements [40]. As can be seen, although there is a growing interest from the scientific community in the specification of requirements from GORE models, there is still a need to propose clear methods that show how the derived models can be transformed into detailed requirements specifications.

The analysis of recent literature reveals a growing interest in the scientific community in the specification of requirements using GORE. As can be seen, various efforts have focused on transforming user stories into models based on objectives, such as Goal Net and iStar, thus facilitating the transition from requirements to architectural design. These efforts include using agent-oriented modelling in agile requirements engineering, improving the readability and usability of requirements models, and developing automated tools that generate iStar models from user stories. Despite these advances, there is still a need to propose precise methods that show how to transform derived models into detailed requirements specifications. In this sense, this article proposes a solution that allows software industry professionals to generate agile requirements specifications more effectively. This solution is based on three elements: (i) an extension of the iStar Framework that incorporates new modeling capabilities aligned with agile requirements, (ii) transformation rules, and (iii) an application process to support the implementation of the rules in the extended iStar models, and thus; generate agile specifications in the format of user stories and acceptance criteria according to the Gherkin template. These extensions were only made in one of the elements of the iStar framework known as SR model (Strategic Rationale model), which is used to represent and analyse the internal intentions of the actors and how these intentions are connected through various elements (stereotypes) and relationships [41], [42].

The article is organised as follows: Section 2 addresses the theoretical framework, presenting in more detail some essential concepts to facilitate the reader's understanding of the document. Section 3 presents the research method, highlighting the process of defining the iStar Framework extension and transformation rules. Section 4 presents the results and discussion, where the transformation rules, a process to support their application, and a practical application example are presented. Section 5 presents the results and discussion obtained from evaluating transformation rules and their application process through a focus group. Finally, Section 6 presents the conclusions and future work.

## **2. Theoretical framework**

To contextualise this work, it is essential to describe two central and widely used theoretical elements in the field of software development: Goal-Oriented Requirement Engineering, the iStar framework, and the Gherkin language. A brief description of each is presented below.

### **2.1. Goal oriented requirement engineering - GORE**

Goal Oriented Requirements Engineering (GORE) has developed mainly in the last two decades. In GORE, goals are modelled and used to capture interactions and trade-offs between requirements, with broader applications in Software Engineering, Information Systems, Conceptual, and Enterprise Modelling. GORE focuses requirements elicitation and specification on objectives, which facilitates modelling, allows different levels of abstraction, covers functional and non-functional requirements, and provides fewer volatile objectives for analysing alternatives. GORE involves the study or application of goal-oriented models, which include the goal concept as a first-class object and can be graphical (e.g. iStar, KAOS) or textual (e.g. GBRAM). The iStar Framework was chosen to be used in this project, as the analysis of previous work shows that most proposals design models or solutions using the iStar specification, focusing on the elicitation and refinement of goals and the exploration of responsibility assignments between stakeholders for various applications.

### **2.2. Framework iStar**

The iStar Framework (also known as the i\* Framework) was defined by Eric Yu [46] and distinguishes itself as a prominent methodology within the field of agent-oriented languages, offering a unique approach to requirements modelling. Unlike traditional methodologies that focus primarily on system functionalities, iStar takes a more holistic perspective, emphasizing the organisational context, social dynamics, and intentional relationships between the various actors interacting with the system. These actors may manifest themselves as individuals, groups, departments, organisations, information systems, or any entity that has goals and intentions relevant to the system in question. To facilitate the capture and analysis of this complexity inherent in socio-technical systems, iStar proposes the use of two interrelated models, each designed to represent a specific and complementary level of abstraction in the requirements engineering process:

- Strategic Dependency (SD) model (known as Strategic Dependency model): This model operates at a relatively high level of abstraction, focusing primarily on the intentional level. Its fundamental purpose is to provide a clear and concise representation of the strategic dependencies that exist between actors. The SD model visualises who depends on whom, what is expected from that dependence, and the underlying reasons. In essence, this model maps the web of dependency relationships between actors, illustrating how the objectives of one actor are intertwined with those of others. Key components of the SD model include:
  - Actors: represent the active entities within the system and its environment, characterised by their goals, intentions, and capabilities.
  - Dependencies: Describe the relationships between actors, specifying how one actor (the dependent actor) relies on another actor (the provider actor) to obtain something of value or support that it needs to achieve its objectives. These dependencies can be classified into:
    - Resource Dependencies: Where one actor depends on another for the availability of a resource, which can be either a physical good (e.g. raw materials, hardware) or an informational good (e.g. data, knowledge).
    - Task Dependencies: Where one actor depends on another for the execution of a specific task or activity.
    - Goal Dependencies: Where one actor depends on another for the achievement of a goal or desired state.
- Strategic Rationality (SR) model (known as Strategic Rationale model): This model operates at a more detailed and deeper level of abstraction, going into the rational level. Its main objective is to examine and make explicit the reasons, motivations, intentions, and internal justifications underlying the dependencies described in the SD model. The SR model seeks to provide a richer and more nuanced understanding of why actors establish certain dependencies and what they aspire to achieve through them. Key components of the SR model include:
  - Actors: like the SD model, but with a more detailed representation of their intentions, beliefs, and capabilities.
  - Goals: represent the end states that actors wish to achieve or maintain.
  - Tasks: represent the actions, activities, or processes that actors perform or carry out to achieve their objectives.
  - Resources: represent the entities, both physical and informational, that actors use, consume, produce, or transform while performing their tasks.
  - Softgoals: Represent the non-functional goals, desired qualities, or constraints that influence how goals are achieved. Softgoals often express quality considerations (e.g. safety, reliability, performance, usability) or organisational constraints (e.g. cost, time).

Together, these iStar models (SD and SR) provide a comprehensive, coherent, and detailed view of the organisational context, the intricate interactions between stakeholders, and the complex motivations that drive their actions within the developing system. This perspective is particularly valuable and crucial in the development of complex systems and socio-technical systems, where a thorough understanding of the needs, objectives, dependencies, and rationalities of the different stakeholders is critical to system design, implementation, and success.

### 2.3. Gherkin

Acceptance criteria define the conditions that a user story must meet to be considered completed and accepted by the customer or Product Owner. Gherkin provides a clear and structured way of expressing these criteria, ensuring that everyone involved has a common understanding of what is expected from the software. Gherkin uses a natural language-based syntax, which makes it easy to write and understand the acceptance criteria. The main structure is based on the following keywords:

- *Feature*: Although used to describe general functionality, it can give context to a set of acceptance scenarios.
- *Scenario*: Defines a specific scenario that describes a situation or flow of events to be tested. Each scenario represents an acceptance criterion.
- *Given*: Describes the initial state or context in which the system is before an event occurs.
- *When*: Defines the event or action that triggers the behaviour to be tested.
- *Then*: Specifies the expected result or the state the system should be in after the event occurs.
- *And, But*: Used to add more details or conditions to the <Given>, <When> and <Then> steps.

Considering the above structure, an example of an acceptance criterion is presented below:

**Feature:** User Login.

*Scenario:* Successful login with valid credentials.

*Given* the user is on the login page.

*When* the user enters a valid username and a valid password.

*And* the user clicks on the 'Login' button.

*Then* the user is redirected to the home page.

*And* a welcome message is displayed.

*Scenario:* Login failed with incorrect password.

*Given* the user is on the login page.

*When* the user enters a valid username and incorrect password.

*And* the user clicks on the 'Login' button.

*Then* an error message is displayed indicating incorrect credentials.

*And* the user remains on the login page.

Benefits of using Gherkin for Acceptance Criteria include the following: (i) clarity and precision: Gherkin helps to write clear, precise, and unambiguous acceptance criteria, the forced structure of the language helps to avoid misunderstanding; (ii) effective communication: by using a language close to natural language, Gherkin facilitates communication between members of the development team, stakeholders and the customer. Everyone can understand and contribute to the definition of the acceptance criteria; (iii) collaboration: Gherkin encourages collaboration by allowing different roles (developers, testers, business analysts, customers) to work together to define and refine acceptance criteria, and (iv) test automation: Gherkin files can be used by test automation tools (such as Cucumber, SpecFlow or Behave) to run tests and verify whether the software meets the defined acceptance criteria. This allows for quick feedback and ensures the quality of the software. Gherkin is a powerful tool for defining and documenting acceptance criteria in a clear, collaborative, and automatable way, which contributes to more effective and higher-quality software development.

### 3. Research method

Figure 1 presents the elements involved with the definition of the proposal, as can be seen, is carried out as an extension of the iStar Framework, which introduces improved capabilities in the modelling of the models obtained with this approach. The new elements introduced allow modelling scenarios to be more aligned with agile requirements, especially those based on user stories and acceptance criteria. These improvements correspond to the extension and incorporation of new capabilities to the existing stereotypes in the Framework, and facilitate the understanding and modelling of agile requirements. In addition, a set of transformation rules have been defined that; together with a proposed application process, enable software industry professionals to generate agile requirements specifications in the format of user stories and acceptance criteria according to the Gherkin template.

These extensions were made exclusively in one of the elements of the iStar framework, known as the SR model (Strategic Rationale model). This model is used to represent and analyse the internal intentions of the actors, as well as how these intentions are interconnected through various elements and relationships. The elements of the SR model include actors, goals, tasks, resources, and softgoals. For their part, actors can be agents, roles, or positions, each with specific responsibilities and capabilities. The relationships in the SR model and the

dependencies allow us to visualize how the tasks and objectives of the actors relate and contribute to each other. The SR model is an extension of the Strategic Dependency model (SD model), while the SD model focuses on strategic dependencies between actors, showing who depends on whom to achieve specific objectives, the SR model provides a deeper and more detailed insight into the reasons behind these interactions and decisions [42]. In the SR model, not only are external dependencies identified, but the internal intentions of the actors are also explored, providing a more complete understanding of the motivations and strategies underlying the system [41], [42]. This differentiation is crucial for the analysis and design of complex systems, as it allows designers to understand not only the dependencies but also the internal factors that influence the actions and decisions of the actors.

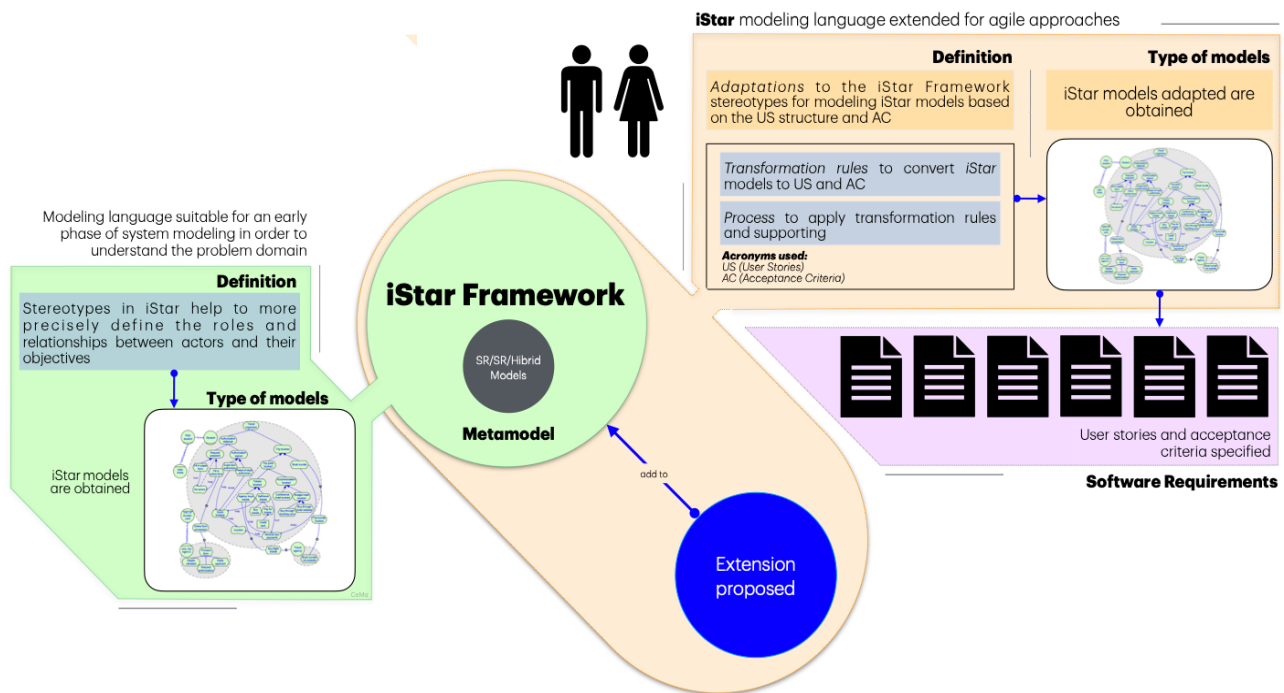


Figure 1. Summary view of elements related to the proposal

### 3.1. Process for extending iStar and defining transformation rules

Table 1 presents the process and sub-process followed to extend the notation of the iStar Framework to obtain extended models that facilitate the generation of user stories and their acceptance criteria. Likewise, Table 1 also presents the artifacts obtained during the iStar Framework notation extension process.

Table 1. Overview of the process to support the extension of the iStar notation to US and AC

Name artifact	Description
Sub-process 01 - iStar notation study	In this sub-process, an analysis of the notation was carried out to show possible solutions to the proposal put forward through an iStar model obtained from a requirement that was taken as an example, with the participation of the researchers and an expert user of iStar.
- (Input) Specification of the iStar Framework	Theoretical documentation on iStar notation
- (Output) Model iStar designed	Model iStar built during the performance of the sub-process activities.
Sub-process 02 - iStar notation extension	In this sub-process the iStar notation was extended to support the representation of US and AC. For the extension of the iStar Framework notation, the following recommendations of the iStar expert user and research community were taken into account: (i) not to add more stereotypes, in this sense, only the SR model was extended by means of labels and additional information that could enrich the

Name artifact	Description
	notation of the Framework, that is why new labels were incorporated to the SR model, (ii) not to make changes or extensions to the meta-model, the extension or inclusion of labels made did not affect the increase or decrease of the existing stereotypes in the Framework, i.e.; the changes were made in the specificity provided by the existing stereotypes.
- (Input) Model iStar designed	Model designed.
- (Output) Instantiated example	Model iStar based on a requirement taken as an example, this example is obtained after making the extension to the notation iStar.
Sub-process 03 - generation of the transformation rules from iStar notation to US and CA	In this sub-process, the activities corresponding to the definition of the transformation rules that allow using the iStar extension and generating the requirements specification according to the format of the Gherkin template are carried out. The activities are carried out with the participation of the researchers and an expert user of iStar.
- (Input) Instantiated example	Instantiated example
- (Output) Transformation rules	Transformation rules that allow the extension of the iStar notation to be used and support the derivation of the US and CA specification.

### 3.2. iStar modification considerations

For the extension of the iStar Framework notation, the following recommendations from users and the iStar community were taken into account: (i) not to add more stereotypes to those already defined by the notation; the proposal was limited to extending the Strategic Rationality (SR) model through labels and additional information to enrich the notation, (ii) not to make any changes or extensions to the metamodel; the extension or inclusion of labels did not affect the number of existing stereotypes in the notation, i.e.: the changes were made within the specificity provided by the existing stereotypes, and (iii) the paths should be modelled from left to right and their reading should be done in the same way.

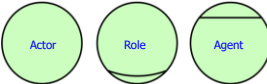

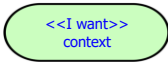
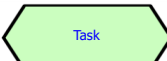
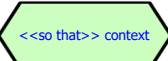

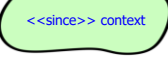
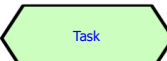
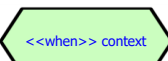
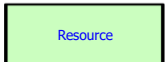
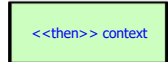
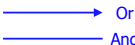
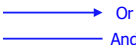
## 4. Results and discussion



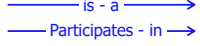
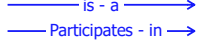
The following sections present the iStar extensions and transformation rules obtained following the application of the process in the previous section. Likewise, a detailed process that supports the application of the proposal is presented, accompanied by a practical example. This will allow readers and professionals to clearly and precisely understand how to operate and apply the proposed methodology.

### 4.1. Proposed transformation rules

Table 2 presents the 9 defined rules, which can be used both to transform iStar models according to the user story structure and acceptance criteria making use of the Gherkin template. As a precondition, full use must be made of these structures, i.e.: “*I like*”, “*I want*”, “*so that*” for a US and “*since*”, “*when*”, and “*then*” for AC. On the other hand, before starting to model, it is important to take into account that some suggestions and indications of the iStar Framework are maintained, one of them is the paths must be modelled from left to right and their reading must be done in the same way. In the following, Table 2 presents the 9 defined transformation rules that extend the capability of the iStar Framework, which can be used both to transform iStar models into US and CA specifications, as well as to transform US and CA into iStar models. Likewise, Table 2 presents an identifier that allows to quickly consult the rules, the elements that make up a US and a CA with their respective description, the iStar stereotype that represents the element of the US or CA with its respective description, and, finally, the extended iStar stereotype is shown, which allows to graphically visualise how to adapt the stereotypes suggested by the Framework to the proposed structure.

Table 2. Suggested rules for extending the iStar Framework notation

#	Element US/AC	Description US/AC element	iStar Stereotype	iStar stereotype description	iStar Extended Stereotype
1	I as a	Represents the type of user for whom a need is being met.		It can be of type Actor, Role or Agent, and represents assets or autonomous entities that aim to achieve their objectives through the exercise of their know-how, in collaboration with other actors.	The same iStar stereotypes
2	I want	Describes the goal or objective to be achieved.		It is a state of affairs that the actor wants to achieve and has well-defined achievement criteria.	
3	So that	Expresses the reason or value that the person obtains from the implementation of the feature or functionality.		It is an attribute for which an actor desires some level of achievement.	
4	Since	Describes the condition.		It is an attribute for which the actor desires some level of achievement. The label “since” is applied only to disaggregate the “so that” of the US being represented.	
5	When/ And	Describes an event or action. The use of “and” is optional when you want to complement the event.		It represents actions that an actor wants to be executed, usually for the purpose of achieving a goal. The “when” label is applied only to disaggregate the “since” of the acceptance criterion being represented.	
6	Then	Describes the result or sequence. The use of “and” is optional when you want to complement the result or consequence.		It is a physical or informational entity that the actor requires to perform a task. The “then” tag is applied only to disaggregate the “when” of the acceptance criterion being represented.	
7	Not available	Not available		The refinement relations “or” and/or “and” allow to relate the elements: “so that” and	

#	Element US/AC	Description US/AC element	iStar Stereotype	iStar stereotype description	iStar Extended Stereotype
8	Not available	Not available		<p>“since”; “since” and “when”; “when” and “and”.</p> <p>The elements: “and” and “then” or “when” and “then” must be linked by a NeededBy relationship.</p>	
9	Not available	Not available		<p>The actors must be related in the same way as the iStar Framework rules in the SD model [43].</p>	

#### 4.2. Process for applying the transformation rules

Table 3 presents the suggested process, and the activities and work products needed to generate, through an input requirement, the specification of the US and its CA according to the Gherkin template. This process makes use of the transformation rules presented in Table 2. In the process, mention is made of SDSituations (Strategic Dependency Situations), which are components used to model and analyze strategic dependencies between actors within a system. These situations help to identify and represent how different actors depend on each other to achieve their objectives [41]. It is important to mention that the iStar Framework community suggests that SDSituation models are used to identify the happy path of the requirements associated with a project. The SDSituations together with the iStar notation are only used to model the happy paths and not the alternate paths, to speed up and not make additional effort in the design of diagrams that may not be necessary to model.

Table 3. Process for the transformation of iStar models to US and AC

Name artifact	Description
Activity 01 - Revise US and CA drafting guidelines and recommendations	In this activity, the document containing the guidelines and recommendations for writing US and CAs was reviewed. This activity has no output devices.
- (Input) <i>Tips and best practices for writing US and CA</i>	Document with the compilation of tips and best practices for writing US and CA.
Activity 02 - Generate the SDSituation model	In this activity the strategic dependency diagram (SDSituation) for the requirement was made.
- (Input) <i>Requirement</i>	Input for the generation of the SDSituation models, containing the requirement described in natural language.
- (Output) <i>SDSituation of the requirement</i>	Document with the SDSituation models generated from the requirement showing the ordering of the time needed between each scenario until the requirement is completed.
Activity 03 - Generate the iStar requisition model	This activity presents the SR model of the requirement with the different actors involved.
- (Input) <i>Requirement</i>	
- (Input) <i>SDSituation of the requirement</i>	Artifacts of input and output of Activity 02.
- (Output) <i>Model iStar of the requirement</i>	Document with the SR diagram from the requirement and the SDSituation model.
Activity 04 - Transformation of the model to US and CA specification.	In this activity, the US and CA specification is generated using the transformation rules.

Name artifact	Description
- (Input) SDSituation of the requirement	Artifacts of input and output of Activity 02 and Activity 03, respectively.
- (Input) Model iStar of the requirement	The transformation rules allow to use the iStar extension and to support the requirements notation with the Gherkin template format, see the rules in Table 2.
- (Input) Transformation rules	
- (Output) Specification of the US and its CAs	Document with the US and CAs drafted using the Gherkin template format.
Activity 05 - Evaluation of the quality of the resulting US and CA	In this activity the evaluation of the quality of the US and CAs defined in Activity 04 is carried out, the resources to support their evaluation are: INVEST (for US) and SMART (for CA), among others.
- (Input) Specification of the US and its Cas	Artifacts of output of Activity 04.
- (Input) INVEST, SMART, other methods.	Document specifying the quality assessment criteria according to the chosen method.
- (Output) Outcome of the evaluation of US and CAs	Document with the results obtained after applying the methods to the US and CA generated in Activity 04.

### 4.3. Example of application

An example of the application of the proposed transformation rules is presented below, which is related to the organization of university trips (see Table 4), and the steps for the transformation of the natural language requirement to US and AC starting from the SR model of the example (see Figure 1 taken from [42]).

Table 4. Description of the requirement in natural language based on the example presented in [42]

Identifier	Title	Responsible
Requirement 001	Organisation of university trips	Jhon Smith
Description (natural language)		
Students must organise their travel (e.g. to conferences), for this they have several objectives to achieve, and options related to them. To achieve their objectives, students rely on other parties, such as a travel agency and the university's travel management information system.		

To better exemplify the extensions and rules presented in Table 2, the iStar Framework community recommends using SDSituation diagrams as a tool to move from tacit knowledge to explicit knowledge and to have a better clarity of the happy paths to follow in a requirement. To model SDSituation diagrams there is no software tool available to model, so the Draw.io modelling tool was used to develop this project. Figures 2 and 3 present the SDSituation models of organising a university trip and their main features such as: obtaining authorisation and booking the already authorised trip. On the one hand, the situations in Figure 2 are called: *Obtain authorisation*, *Request paper authorisation*, *Request authorization online*, *Notify the applicant*, *Obtain signed authorisation*, *Authorised by supervisor* and *Authorised by head of department*, each situation shows which step the PhD student must follow to obtain the travel authorisation. In Figure 3 the scenarios are named: *Book authorised travel*, *Book by peers*, *Book travel package*, *Book the package through Expedia travel agency*, *Book tickets*, *Buy through the agency*, *Self-book tickets*, and *Book accommodation*, each scenario shows which step the PhD student must follow to book the already authorised travel. In addition, Figures 2 and 3 show the sequencing of the scenarios. Based on the SDSituation diagrams obtained (see Figure 2 and Figure 3), Figure 4 presents the SR model proposed by [43] for the requirement in Table 4.

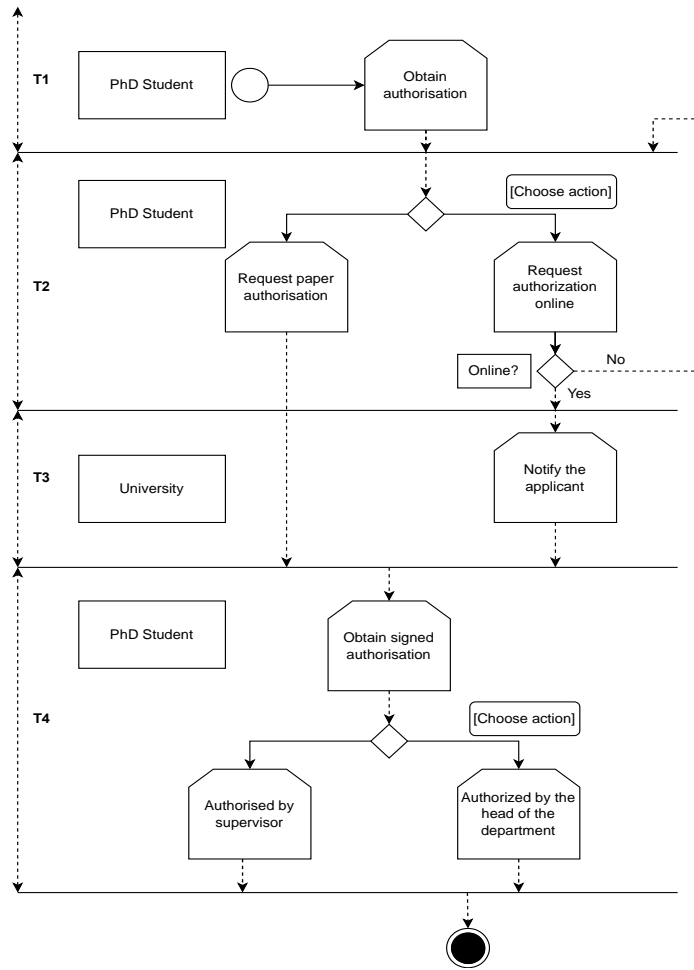


Figure 2. SDSituation to obtain travel authorization

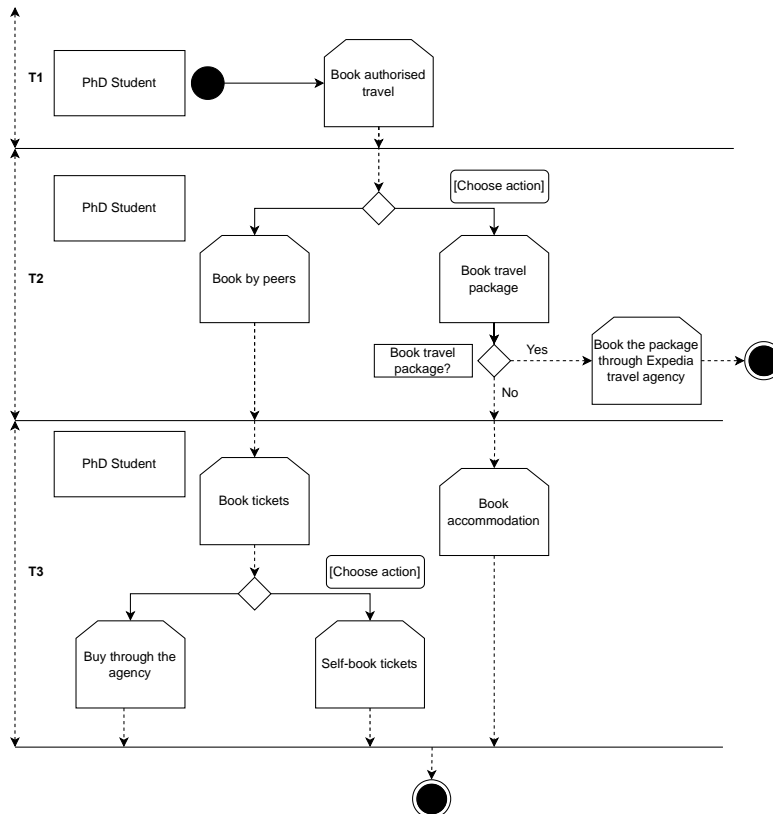


Figure 3. SDSituation to book authorised travel

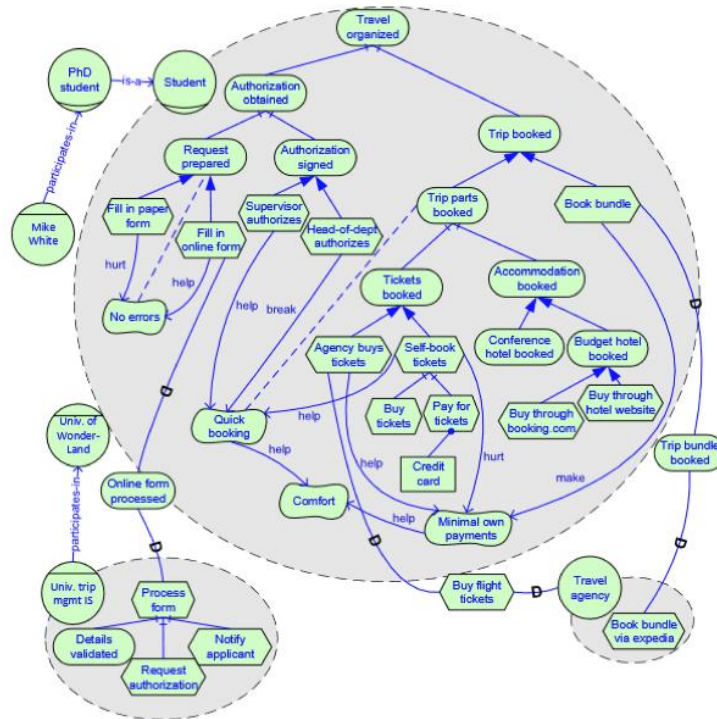


Figure 4. Model iStar for the requirement (see Table 4): Organisation of university travel, taken from [43]

As can be seen, Figure 5 and Figure 6 present the iStar (extended) models obtained after applying the transformation rules to the iStar model presented in Figure 4, these figures present a transformation that facilitates the analysis of the requirement modelled in iStar (see Figure 4), especially because it extends its specificity and detail by using a structure that provides more information and context close to the user story structures and their acceptance criteria. In this sense, Figure 5 corresponds to the extended iStar model for the user story we have named US01, which corresponds to the completion of the travel request by the PhD student. Figure 6 corresponds to the extended iStar model for the user story we have denoted US02, which corresponds to the booking of the trip by the PhD student. As can be seen in Figure 5 and Figure 6, to extend the original iStar model (see Figure 4), the rules proposed in Table 2 have been applied.

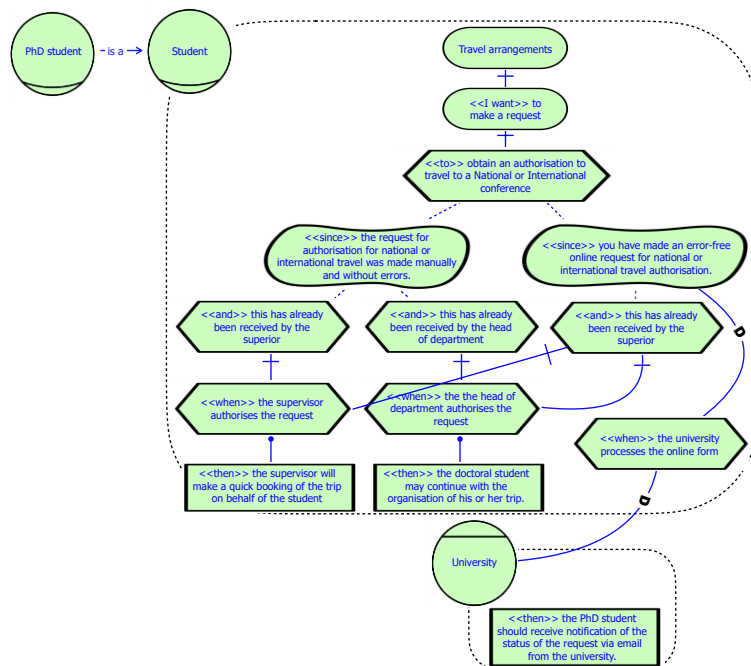


Figure 5. Model iStar obtained for the execution of a request

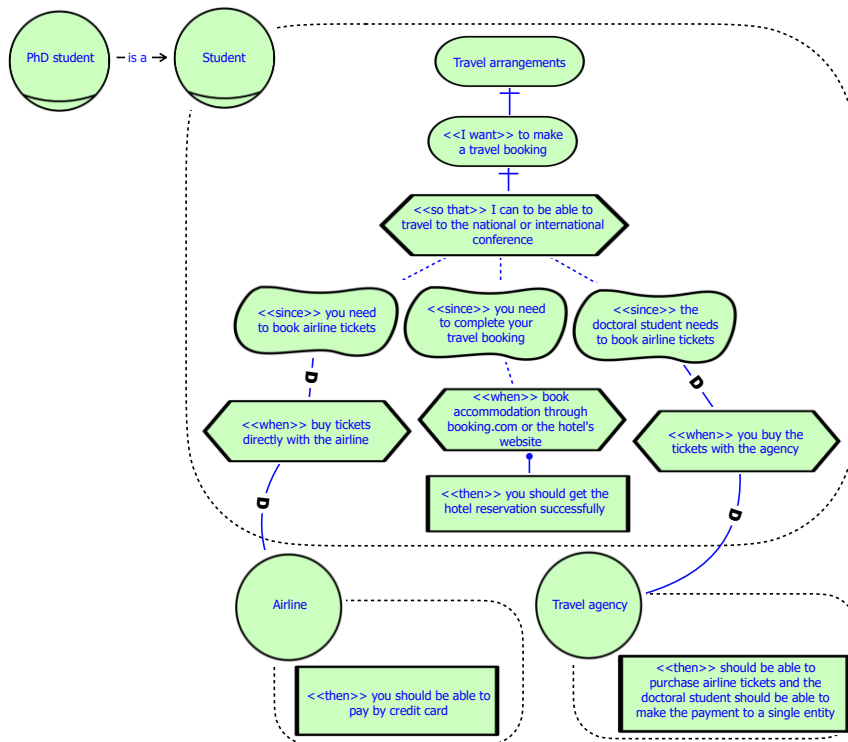


Figure 6. Model iStar obtained for booking the trip

From the models obtained in Figure 5 and Figure 6, it is easier to obtain the specification of the user stories US01 and US02, this specification is presented in Table 5, which shows in detail where this specification is obtained from the extended iStar models, and the transformation rule applied to transform these models into user stories specification. As can be seen, Table 5 applies rules 1, 2, 3, and 9 (from Table 2) for the transformation of the iStar US model (from Figure 5 and Figure 6), and the user stories obtained are the following: (i) US01: I as a PhD student I want to make a request so that I can obtain travel authorisation for a national or international conference and (ii) US02: I as a PhD student I want to make a travel booking so that I can be able to travel to the national or international conference. Rule 9 corresponds to the relationships between the stereotypes.

Table 5. Application of the proposed transformation rules for user stories: US01 and US02

#	US of model iStar	Stereotypes specifying US	Transformation model iStar to US	Rule
US01			<i>I as a PhD student</i>	1, 9
			<i>I want to make a request</i>	2
			<i>so that I can obtain travel authorisation for a national or international conference</i>	3
US02			<i>I as a PhD student</i>	1, 9

#	US of model iStar	Stereotypes specifying US	Transformation model iStar to US	Rule
			<i>I want to make a travel booking</i>	2
			<i>so that I can be able to travel to the national or international conference</i>	3

Table 6 exemplifies the application of the proposed transformation rules (see Table 2), in which it is possible to obtain more specification and context in terms of the acceptance criteria of the US01. The rules that are mainly applied are: 4, 5, 6, 7, and 8 of Table 2. Likewise, it is possible to infer from Figure 5 that the acceptance criteria that could be obtained are five, these correspond to the 5 paths that can be followed through the model before finding no more detail or stereotypes than those that can be reached and followed. Due to space limitations, Table 6 only presents the route for three of the paths that can be travelled, and that provides more specification in terms of acceptance criteria for US01 (see Table 5). Finally, Table 7 presents the specifications obtained for user stories US01 and US02 along with their acceptance criteria in Gherkin format applying the proposed transformation rules.

Table 6. Application of the proposed transformation rules for ACs of user story US01

#	US of model iStar	Stereotypes specifying US	Transformation model iStar to US	Rule
US001			since the request for authorisation for national or international travel was made manually and without errors	1, 9
			and this has already been received by the superior	2
			when the supervisor authorises the request	3
	then the supervisor will make a quick booking of the trip on behalf of the student			

Table 7. Process for transformation of iStar models to US and AC

User story	Acceptance criteria
US01: <i>I as a PhD student I want to make a request so that I can obtain travel authorisation for a national or international conference</i>	<i>Since</i> the request for authorisation for national or international travel was made manually and without errors <i>and</i> this has already been received by the superior <i>when</i> the supervisor authorises the request <i>then</i> the supervisor will make a quick booking of the trip on behalf of the student.
	<i>Since</i> the request for authorisation for national or international travel was made manually and without errors <i>and</i> this has already been received by the head of department <i>when</i> the head of department authorises the request <i>then</i> the doctoral student may continue with the organisation of his or her trip.
	<i>Since</i> you have made an error-free online request for national or international travel authorization, <i>and</i> this has already been received by the superior <i>when</i> the supervisor authorises the request <i>then</i> the supervisor will make a quick booking of the trip on behalf of the student.
	<i>Since</i> you have made an error-free online request for national or international travel authorization, <i>and</i> this has already been received by the superior <i>when</i> the head of department authorises the request <i>then</i> the doctoral student may continue with the organisation of his or her trip.
US02: <i>I as a PhD student I want to make a travel booking so that I can be able to travel to the national or international conference</i>	<i>Since</i> you have made an error-free online request for national or international travel authorization <i>when</i> the university processes the online form <i>then</i> the PhD student should receive notification of the status of the request via email from the university.
	<i>Since</i> you need to book airline tickets <i>when</i> buy tickets directly with the airline <i>then</i> you should be able to pay by credit card.
	<i>Since</i> you need to complete your travel booking <i>when</i> book accommodation through booking.com or the hotel's website <i>then</i> you should get the hotel reservation successfully.
	<i>Since</i> the doctoral student needs to book airline tickets <i>when</i> you buy the tickets with the agency <i>then</i> should be able to purchase airline tickets and the doctoral student should be able to make the payment to a single entity.

## 5. Evaluation

The proposed transformation rules that extend the capabilities of the iStar model and allow obtaining user story specifications and acceptance criteria from extended iStar models, were evaluated through a focus group as a qualitative research technique. To carry out the focus group, the steps proposed by Kontio *et al.* [44], [45] were carried out, they are: (i) defining the research problem, (ii) selecting participants, (iii) conducting the focus group session, (iv) data analysis and reporting, (v) improvement actions, (vi) research construction and (vii) challenges, limitations, and bias. This method allows conducting focus groups in software engineering to gather qualitative information that can improve software development processes or products, and as can be inferred, it focuses on obtaining deep and varied perceptions of the participants on a specific topic.

### 5.1. Defining the research problem

The main objective of this evaluation was to obtain personal perceptions through group interaction. In this sense, the objective was to collect suggestions, opportunities for improvement, and even recognition of approval and validity of the proposed solution. The focus group was able to obtain the opinions and perceptions of professionals with experience in agile software development. In addition, the proposal was evaluated concerning five variables: (i) completeness, (ii) suitability, (iii) applicability in agile approaches, (iv) ease of understanding

and understanding, and (v) general aspects. Furthermore, these elements allowed the application of a focused approach and meticulously structuring the execution of the debate protocol, ensuring a clear and coherent guide for the discussion. Likewise, it allowed the relevant documents to be shared with the participants to be exhaustively defined and shared, ensuring that everyone had the necessary information to contribute in an informed and meaningful manner. Additionally, robust and systematic methods for capturing and recording information were established, allowing for accurate and complete data collection. Finally, a detailed and rigorous analysis of the information obtained during the debate was carried out, using advanced coding and categorization techniques to identify patterns, emerging themes, and diverse perspectives. This comprehensive approach not only facilitated obtaining high-quality qualitative data but also provided a solid foundation for the interpretation and application of the findings in the context of software development. Feedback from participants included opinions on the adequacy of the defined stereotypes, the adequacy of the proposed rules, the clarity and ease of understanding of the process, the applicability to agile approaches, and the need to add, modify or remove process elements, stereotypes or rules.

## **5.2. Selecting participants**

In this activity, the profiles of the participants and the corresponding selection criteria were carefully defined. The established criteria included: possessing advanced knowledge of agile approaches and having more than two years of practical experience in the industry, specifically in the use and application of user stories and acceptance criteria in real projects. Verification of these requirements was carried out by reviewing proven and certified experience. During the recruitment process, 10 potential candidates were initially identified, however, 2 of them were discarded for not meeting the strictly established criteria, resulting in the selection of 8 participants who fully satisfied the required profile. Once the participants were selected, they were sent a formal invitation to participate in the focus group, which was positively accepted by all of them. Subsequently, the date and time of the debate session were coordinated, ensuring a three-week margin for adequate planning and preparation. With the discussion session scheduled, proposal documents were sent to participants for prior review, ensuring that everyone had access to the information necessary for a productive and focused discussion. There were no challenges regarding low acceptance and participation by the participants.

## **5.3. Conducting the focus group session**

This activity was meticulously coordinated by a member of the research group who acted as moderator, accompanied by another member who played the role of rapporteur. The structure and sequence of the session were elaborately planned and communicated to participants in advance, ensuring that everyone was informed about the development of the activity. During the session, the rapporteur was responsible for recording in detail each observation and comment made by the participants, using structured note-taking techniques to ensure complete and accurate capture of the information. This systematic recording allowed for rigorous monitoring of the discussions and facilitated subsequent analysis. To strengthen the focus group discussion and gain a deeper understanding of the participant's engagement with the application example and the clarity of the proposed rules, several methodological strategies were implemented. First, a significant segment of the session was devoted to a detailed exploration of the application example. Instead of a simple presentation, an interactive approach was proposed where participants were invited to analyse the example in depth, breaking it down into its constituent elements and discussing its applicability in various agile scenarios.

Specific questions designed to provoke reflection and critical analysis were asked, such as: ‘How would the extended iStar models adapt to projects with different levels of complexity?’, ‘What practical challenges do you anticipate in applying these rules in your daily experience?’, and ‘To what extent does the example facilitate the understanding of the transformation from models to user stories and acceptance criteria? In addition, practical exercises and simulations were introduced where participants applied the concepts to hypothetical cases, allowing them to actively assess their understanding and ability to apply the proposal. Secondly, special emphasis was placed on the thorough clarification of the proposed rules. The moderator took a proactive role in explaining each rule in detail, using concrete examples and analogies to facilitate understanding. An open

dialogue was promoted where participants were able to express their doubts, concerns, and suggestions for improvement.

Hypothetical scenarios illustrating the application of the rules in a variety of contexts were presented, and participants were invited to identify possible ambiguities, inconsistencies, or areas where the rules could be confusing. They were encouraged to propose modifications or additions that would contribute to increasing the accuracy and clarity of the rules. To foster rich discussion and meaningful exchange of views, the moderator employed group dynamics techniques designed to stimulate interaction and constructive debate. Spaces were created where participants could share their experiences, perspectives, and points of view, even when these were divergent. An atmosphere of trust and mutual respect was established, where honesty and openness in communication were valued.

After the session, participants were asked to complete a questionnaire specifically designed to address the questions presented in Table 8. This questionnaire not only allowed for the collection of additional data and clarification of points discussed during the session but also provided an opportunity for Participants will reflect and expand their answers, thus contributing to a deeper and more exhaustive understanding of the topics discussed.

The proposed questionnaire consisted of 17 questions, the first 13 (questions 1-13) related to the defined variables, and the last 4 (questions 14-17) open (see Table 8). The first 13 questions were evaluated using a 5-point Likert scale as follows: (1) Very bad, very dissatisfied, (2) Bad, not very satisfied, (3) Good, sufficient, adequate, somewhat satisfied, (4) Fairly good, adequate, satisfied and (5) Very good, very adequate, very satisfied. Besides, it is important to highlight that each of the defined questions was evaluated according to the following aspects: clarity, simplicity, neutrality, scope, ambiguity, and coherence between the evaluated variables, among others.

Table 8. Research questions applied in the focus group

Variable	#	Questions	Scale				
			1	2	3	4	5
Completeness	Q1	Do you consider that the rules defined for the transformation of iStar models to US and CA specifications are sufficient?	0	0	0	6	2
	Q2	Do you think the stereotypes defined in the iStar notation extension are sufficient?	0	0	2	6	0
	Q3	Do you consider that the process elements defined (activities, roles and artifacts) in the iStar stereotyping process to US and CA specification are sufficient?	0	0	0	6	2
Suitability	Q4	Do you consider that the proposed rules cover the minimum elements necessary to consider for the transformation of iStar models to US and CA specification?	0	0	2	4	2
	Q5	Do you consider that the process defined for the transformation of iStar models to US and CA specifications meets the proposed objective?	0	0	2	5	1
	Q6	Do you consider that the proposed stereotyping meets the minimum necessary elements of Gherkin's template?	0	0	1	2	5
	Q7	Do you consider that the transformation of iStar models to US and CA specification meets the minimum necessary elements of the Gherkin template?	0	0	1	3	4
Applicability in agile approaches	Q8	Do you consider that the transformation from iStar models to US and CA specifications is suitable for an agile development context?	0	0	2	4	2
	Q9	Based on your experience, do you consider that the elements defined (activities and artifacts) in the process for the transformation of iStar	0	0	2	3	3

Variable	#	Questions	Scale				
			1	2	3	4	5
		models to US and CA specification are appropriate and can be successfully applied in an agile software development project?					
Ease of understanding and understanding	Q10	Do you consider that the process defined for the transformation of iStar models to US and CA specifications and the elements that describe it (activities and artifacts) are easy to understand?	0	1	2	3	2
	Q11	Do you find the BPMN diagrams generated through this proposal useful to facilitate understanding of the proposed process?	0	0	2	2	4
	Q12	Do you consider that the rules defined for the transformation of iStar models to US and CA specifications are clear and easy to understand?	0	1	2	3	2
	Q13	Do you consider that the stereotypes defined in the iStar notation are clear and easy to understand?	0	0	3	4	1
General aspects	Q14	Do you consider that elements should be added, modified or eliminated in the process defined for the transformation of iStar models to US and CA specifications?	Open question				
	Q15	Do you think elements should be added, modified or removed for iStar stereotyping?	Open question				
	Q16	Do you consider that rules defined for the transformation of iStar models to US and CA specifications should be added, modified or deleted?	Open question				
	Q17	Do you have any additional comments about the process, stereotypes or proposed rules?	Open question				

#### 5.4. Data analysis and reporting

After carrying out the focus group, the analysis of the contributions and responses of the participants was carried out during the proposal debate session through the questionnaires completed at the end of the session. Finally, special attention was paid to the documentation and thorough analysis of the results. The rapporteur played a crucial role in thoroughly recording all observations, comments, and suggestions from participants. Structured note-taking techniques were used to ensure complete and accurate capture of information. Following the focus group session, a rigorous analysis of the data collected was carried out, using coding and categorisation methods to identify patterns, emerging themes, and areas of consensus or discrepancy. The results of this analysis were captured in a detailed report that accurately reflected the views and perceptions of the participants, both on the application example and on the clarity and usefulness of the proposed rules. This activity was carried out following the strategies defined in the information capture and registration methods phase. Table 8 presents the count of the participants' responses for each of the questions according to the established scale. Likewise, Figure 4 shows the consolidated distribution of the results obtained from questions 1-13. As can be seen, in general, the participants had a favourable opinion about the completeness, suitability, applicability in agile approaches, ease of understanding, and understanding of the extension of the iStar model to support obtaining the specification of user stories and acceptance criteria. However, the responses to questions Q10 and Q12 were not favourable, indicating areas where the proposal could benefit from further refinement. These questions highlighted specific weaknesses or gaps in the current approach, suggesting the need for targeted improvement actions. Consequently, it was determined that a series of enhancement measures should be implemented to address the identified issues. These measures will be aimed at strengthening the overall proposal, ensuring it meets the desired standards and effectively addresses the concerns raised by the participants. By incorporating this feedback, the proposal can be significantly improved, leading to a more robust and effective solution. On the other hand, open questions Q14, Q15, Q16, and Q17 allowed participants to propose adjustments and improvements to the proposal, some of which are addressed in the next section.

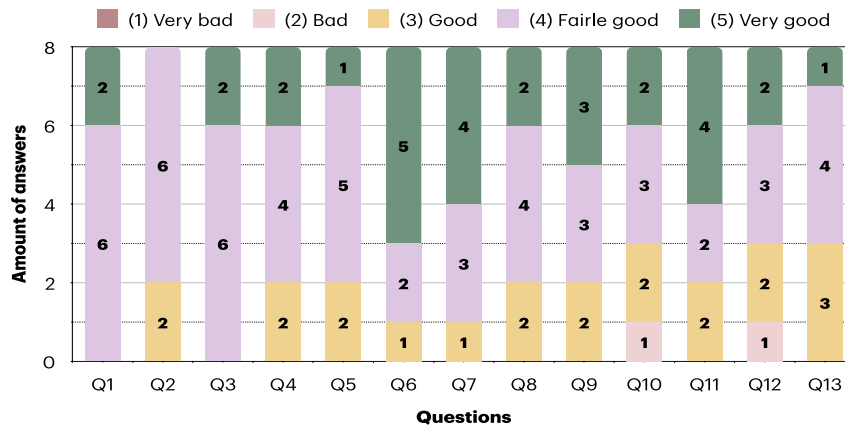


Figure 4. Consolidation of questions 1-13 answered by the focus group

### 5.5. Analysing the discrepancies in the responses

Through the analysis of the results, some discrepancies could be perceived among the focus group participants in several aspects, mainly about the sufficiency and clarity of the evaluated proposal. The areas where these discrepancies are evident are detailed below:

- Question Q2. Adequacy of the stereotypes defined in the extension of the iStar notation: While 6 participants considered the stereotypes to be ‘Fairly good, adequate, satisfied’, 2 participants found them to be only ‘Good, sufficient, adequate, somewhat satisfied’. This suggests that some participants perceived that the stereotypes might need improvement or addition.
- Question Q4. Adequacy of the proposed rules for the transformation of iStar models to HU and CA specification: In this question, there was a split in the responses as follows: 4 participants chose ‘Fairly good, adequate, satisfied’, 2 participants chose ‘Good, sufficient, adequate, somewhat satisfied’ and 2 participants chose ‘Very good, very adequate, very satisfied’. This indicates that, although the majority were satisfied, there were divergent opinions on the adequacy of the rules.
- Questions Q10 and Q12. Clarity and ease of understanding of the defined process: In these questions, the greatest discrepancy is observed. Some participants indicated that they were ‘Bad, not very satisfied’ with the clarity of the process and rules, while others were ‘Very good, very adequate, very satisfied’. This reveals that the proposal was not equally clear to all participants.
- Question Q9. Applicability in agile approaches: There were split responses with 3 participants answering, ‘Very good, very adequate, very satisfied’, 3 participants answering, ‘Fairly good, adequate, satisfied’, and 2 participants answering ‘Good, sufficient, adequate, somewhat satisfied’. This indicates that, although the majority agreed that it could be successfully applied in agile projects, some were not entirely convinced.
- Open questions Q14, Q15, Q16, and Q17: Responses to the open questions also show discrepancies in terms of the need to add, modify, or remove process elements, stereotypes, or rules. Some participants suggested adding more descriptions or examples, while others felt that the elements were adequate as they were.

Based on the results obtained, it can be observed that, although most participants had a favourable opinion of the proposal, there were notable discrepancies in terms of the perceived clarity, sufficiency, and applicability of the defined elements. This suggests that, while the proposal was generally well received, there were still areas that required further improvement or clarification to achieve greater uniformity in participants' perceptions; these identified opportunities for improvement are addressed in the following section.

### 5.6. Improvement actions

The results, comments, and opinions of the participants were carefully analysed and considered to implement improvement actions in the proposal. This review process allowed us to identify key areas of improvement and refine the proposal significantly. As a result of this detailed analysis, a new version of the proposal was

developed, which incorporates the valuable contributions and suggestions of the participants. This revised version, presented in this work, reflects a more robust and effective approach, aligned with the needs and expectations identified during the evaluation process. Some of the improvements made were: (i) the description of some activities was improved, (ii) application examples that allow readers and interested parties to better understand the application of the proposal, (iii) some typos and typographical errors were corrected and (iv) a section on benefits and limitations of the proposal was added. Regarding the limitations of the focus group, some of them and how they were addressed are described below: (i) Not all experts had the same level of knowledge about the iStar notation, therefore, two weeks before carrying out the focus group, reading material was sent to each participant explaining in detail the proposed process, (ii) a total of 10 people were invited to carry out the focus group to mitigate the risk of not meeting the minimum participation of 4 people, and (iii) to minimize the risk of not having an active partition of the participants, the moderator motivated the continued participation of the participants.

### **5.7. Research construction**

To ensure that the research construct in this study was valid and aligned with our research objectives, three key techniques were employed. First, the established content and format for the focus group session were defined, maintained, and rigorously respected, ensuring a coherent and uniform structure that facilitated the collection of comparable and relevant data. Second, instrumentation errors were reduced by audio recording the discussion session, allowing for an accurate and detailed review of all interventions and participant comments. Finally, to minimize potential bias in the interpretation of the results, a person external to the research was involved, who reviewed all interpretations made during the analysis. This external review provided an unbiased and critical perspective, contributing to the objectivity and accuracy of the study's conclusions.

### **5.8. Challenges, limitations, and bias**

Below are some limitations and solutions that emerged during the focus group: (i) although there was a predefined format and agenda, it was initially challenging for the moderator to control the discussion style of the less active participants. This situation was quickly corrected by more experienced researchers once detected; (ii) some embarrassing situations caused by incorrect answers from the participants were mitigated thanks to the active intervention of the moderator, who facilitated the discussion constructively; and (iii) to mitigate the risk of limited knowledge and understanding of participants, individuals with similar experience were selected, advance reading material was provided, and complex topics were broken down into more “manageable chunks”.

Furthermore, in focus groups, there may be a selection bias that prevents the diversity of relevant opinions or perspectives from being adequately represented. To diminish this aspect, the inclusion of a variety of perspectives and experiences relevant to the topic at hand was ensured. Also, based on experience, some participants may have a disproportionate influence on the discussion, limiting the participation of other members. To ensure equal participation, rules were established that fostered an environment where all participants felt comfortable sharing their opinions, avoiding the domination of the discussion by a few individuals. Likewise, to avoid the influence of the moderator on the direction of the discussion, an impartial moderator was used who neutrally guided the debate, avoiding influencing the responses of the participants. Finally, the difficulty in generalizing the results was considered because focus groups usually consist of small and specific samples, which can make it difficult to extrapolate the findings to a broader population. To address this limitation, the focus group was complemented with another data collection technique, such as the inclusion of a perception survey applied individually, which allowed for obtaining a more complete and generalizable vision of the research object.

## **6. Conclusions and future work**

This article presents nine rules extending the capabilities of the iStar framework notation, which can be used to design iStar models and facilitate the obtaining and specifying of user stories and their acceptance criteria. These

rules significantly improve the understanding of software requirements in agile projects. This is achieved by adding new capabilities to the iStar model stereotypes based on the elements of the Gherkin template, which are described in detail in this article. In addition, a comprehensive example is included based on one of the most recognized and used models in the iStar Framework research community, allowing interested parties to understand and apply the proposal in their projects and research more effectively.

The practical implementation of the proposed transformation rules provides a clear roadmap for software engineering professionals in industry and academia seeking to improve the specification of requirements in agile projects. By extending the iStar framework with new modelling capabilities aligned with agile requirements and providing a detailed process for applying these rules, the proposal facilitates the generation of requirements specifications using user stories and acceptance criteria. This standardization not only improves the clarity and completeness of the requirements but also promotes greater understanding among the project team members, which facilitates adoption in agile teams. Feedback from the focus group further underscores the proposal's applicability in agile contexts, highlighting its potential to optimize software development processes and improve the quality of the final product.

The proposed extension to the iStar framework allows organizations to obtain a graphical overview of the relationships between actors and activities within a software system in development, with clarity for those companies that use agile approaches and software requirements specifications through user stories and acceptance criteria. This can support improving the quality of the requirements defined for a system and, therefore, the quality of its processes and products.

Beyond the immediate improvements in requirements specification, the proposal has the potential to positively impact the software development lifecycle in the long term. By providing a systematic methodology for the transformation of iStar models to user stories and acceptance criteria in Gherkin, it facilitates greater clarity and accuracy in the communication of requirements from the early stages of development. This, in turn, can reduce ambiguity and misunderstandings that often lead to rework, defects, and increased costs. In addition, the test automation facilitated by Gherkin enables more efficient and effective software validation, which contributes to improving the quality of the final product and speeding up delivery. Ultimately, the adoption of this approach could result in a more efficient, predictable, and lower-cost software development process, with tangible benefits in terms of software quality, customer satisfaction, and project success.

Conducting a focus group to evaluate the proposal made it possible to bring together experts in Software Engineering, who, based on their experience and knowledge, identified opportunities for improvement. The use of this qualitative study technique provided valuable feedback, resulting from the discussion of different points of view among the participants. Although some improvement actions were highlighted, in general terms, the focus group participants considered that the rules were suitable, complete, clear, and easy to understand.

In future work, it is expected to develop a software tool that automates the transformation of iStar models to user story specifications and acceptance criteria based on the defined rules. In addition, a method for evaluating the quality of the generated documentation is planned, based on evaluation criteria defined following the ISO/IEC/IEEE 29148 standard.

### **Acknowledgements**

Professor César Pardo is grateful for the contribution of the Universidad del Cauca, where he currently works as a full professor.

### **References**

- [1] R. Pressman and B. Maxin, *Software Engineering: A Practitioner's Approach*, 1st ed., vol. 1. McGraw Hill, 2014.

- 
- [2] J. Johnson, "Chaos The Dollar Drain of IT Project Failures," *Application Development Trends*, vol. 2, no. 1, pp. 41–47, 1995.
- [3] B. W. Boehm and V. R. Basili, "Software defect reduction top 10 list," *IEEE Computer*, vol. 34, no. 1, pp. 135–137, 2001, doi: 10.1109/2.962984.
- [4] B. W. Boehm and P. N. Papaccio, "Understanding and controlling software costs," *IEEE Transactions on Software Engineering*, vol. 14, no. 10, pp. 1462–1477, 1988, doi: 10.1109/32.6191.
- [5] D. M. Berry, E. Kamsties, and M. M. Krieger, "From contract drafting to software specification: Linguistic sources of ambiguity," 2003. [Online]. Available: <https://www.eecs.uottawa.ca/~adjointe/2003/BKK03.pdf>
- [6] G. Kotonya and I. Sommerville, *Requirements Engineering: Processes and Techniques*, 1st ed., vol. 1. Wiley, 1998.
- [7] A. M. Davis, *Software requirements: objects, functions, and states*, 1st ed., vol. 1. Prentice Hall, 1993.
- [8] K. E. Wiegers and J. Beatty, *Software Requirements*, 3rd ed., vol. 1. Microsoft Press, 2013.
- [9] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, 1st ed., vol. 1. Wiley, 1997.
- [10] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, vol. 20, no. 5, pp. 449–480, 2010, doi: 10.1111/j.1365-2575.2007.00259.x.
- [11] Standish Group, "The CHAOS Report," 1995. [Online]. Available: [https://www.standishgroup.com/sample\\_research\\_files/chaos\\_report\\_1995.pdf](https://www.standishgroup.com/sample_research_files/chaos_report_1995.pdf)
- [12] D. M. Fernández *et al.*, "Naming the pain in requirements engineering," *Empir Softw Eng*, vol. 22, no. 5, pp. 2298–2338, 2017, doi: 10.1007/s10664-016-9451-7.
- [13] D. Zowghi and C. Coulin, "Requirements elicitation: A survey of techniques, approaches, and tools," in *Engineering and Managing Software Requirements*, Springer, 2005, pp. 19–46. doi: 10.1007/3-540-28244-0\_2.
- [14] G. E. Mogyorodi, "Requirements Based Testing," *The Journal of Defense Software Engineering*, vol. 1, no. 1, pp. 12–15, 2003.
- [15] M. Cohn, *User Stories Applied: For Agile Software Development*, 1st ed. Addison-Wesley, 2004.
- [16] Y. Wautelet, S. Heng, M. Kolp, and I. Mirbel, "Unifying and extending user story models," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8484 LNCS, pp. 211–225, 2014, doi: 10.1007/978-3-319-07881-6\_15/COVER.
- [17] G. Lucassen, F. Dalpiaz, J. M. van der Werf, and S. Brinkkemper, "The use and effectiveness of user stories in practice," *Requir Eng*, vol. 21, no. 3, pp. 205–234, 2016, doi: 10.1007/s00766-016-0250-0.
- [18] K. S. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, 1st ed., vol. 1. Addison-Wesley, 2012.
- [19] D. Leffingwell, *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, 1st ed., vol. 1. Addison-Wesley, 2010.
- [20] B. Ramesh, L. Cao, and R. Baskerville, "Agile requirements engineering practices and challenges: an empirical study," *Information Systems Journal*, vol. 20, no. 5, pp. 449–480, 2010, doi: 10.1111/j.1365-2575.2007.00259.x.
- [21] A. Cockburn, *Writing Effective Use Cases*, 1st ed., vol. 1. Addison-Wesley, 2001.
- [22] O. Liskin, S. Schneegaß, and K. Schneider, "Understanding the role of user stories in agile projects," in *Proceedings of the 22nd IEEE International Requirements Engineering Conference*, 2014, pp. 441–446. doi: 10.1109/RE.2014.6912313.
- [23] M. Morales and J. Garzás, *Gherkin: Una guía de supervivencia ágil*, 1st ed., vol. 1. España: 233 Grados de TI, 2016. Accessed: Aug. 27, 2023. [Online]. Available: <https://bit.ly/3xjhrSy>
-

- 
- [24] E. Bjarnason, K. Wnuk, and B. Regnell, “Requirements are slipping through the gaps—A case study on causes & effects of communication gaps in large-scale software development,” in *2011 IEEE 19th International Requirements Engineering Conference (RE)*, 2011, pp. 37–46. doi: 10.1109/RE.2011.6051636.
- [25] D. Zowghi and V. Gervasi, “On the interplay between consistency, completeness, and correctness in requirements evolution,” *Information and Software Technology*, vol. 45, no. 14, pp. 993–1009, 2003.
- [26] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, 1st ed. Wiley, 1997.
- [27] A. M. Davis, *Software Requirements: Analysis and Specification*, 1st ed., vol. 1. Prentice Hall, 1995.
- [28] R. L. Glass, *Software Runaways: Lessons Learned from Massive Software Project Failures*, 1st ed., vol. 1. Prentice Hall, 1998.
- [29] D. Ramos, R. Noriega, J. Rubén Laínez, and A. Durango, *Curso de Ingeniería de Software: 2ª Edición*, 2nd ed., vol. 1. IT Campus Academy, 2017.
- [30] A. Martini, T. Besker, and J. Bosch, “Process Debt: a First Exploration,” in *Asia-Pacific Software Engineering Conference, APSEC*, Singapore, 2020, pp. 316–325.
- [31] J. Horkoff *et al.*, “Goal-oriented requirements engineering: an extended systematic mapping study,” *Requir Eng*, vol. 24, no. 2, pp. 133–160, Jun. 2019, doi: 10.1007/S00766-017-0280-Z/TABLES/8.
- [32] A. van Lamsweerde, “Goal-Oriented Requirements Engineering: A Guided Tour,” in *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, 2001, pp. 249–262. doi: 10.1109/ISRE.2001.948567.
- [33] F.-I. Muelas-Muelas, L.-C. Narváez-Jiménez, and C.-J. Pardo-Calvache, “Agile Functional Requirements Modeling from Goal-Oriented Requirements Engineering: A Systematic Mapping Study,” *Revista Facultad de Ingeniería*, vol. 31, no. 62, p. e14839, Dec. 2022, doi: 10.19053/01211129.V31.N62.2022.14839.
- [34] J. Lin, H. Yu, Z. Shen, and C. Miao, “Using goal net to model user stories in agile software development,” *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2014, doi: 10.1109/SNPD.2014.6888731.
- [35] L. Liu and E. Yu, “From Requirements to Architectural Design – Using Goals and Scenarios,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 14, no. 06, pp. 741–760, 2004, doi: 10.1142/S0218194004002061.
- [36] T. Tenso and K. Taveter, “Requirements engineering with agent-oriented models,” *ENASE 2013 - Proceedings of the 8th International Conference on Evaluation of Novel Approaches to Software Engineering*, pp. 254–259, 2013, doi: 10.5220/0004569302540259.
- [37] P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini, “Tropos: An Agent-Oriented Software Development Methodology,” *Autonomous Agents and Multi-Agent Systems*, vol. 8, no. 3, pp. 203–236, 2004, doi: 10.1023/B:AGNT.0000018806.20901.ef.
- [38] C. S. Filho, A. Sousa, J. Melo, M. Lucena, and F. Alencar, “Specifying Guidelines to Transform i\* Model into User Stories: an Overview,” in *International i\* Workshop*, Ottawa, Canada, 2015, pp. 109–114. Accessed: Aug. 27, 2023. [Online]. Available: <http://ceur-ws.org/Vol-1402/paper21.pdf>
- [39] A. Jaqueira, M. Lucena, F. Alencar, C. Agra, and E. Aranha, “Integrating user stories and i\* models: A systematic approach,” in *ICEIS 2015 - 17th International Conference on Enterprise Information Systems, Proceedings*, SciTePress, 2015, pp. 428–438. doi: 10.5220/0005400904280438.
- [40] G. Mussbacher, D. Amyot, and M. Weiss, “Visualizing Early Aspects with Use Case Maps and i\*,” in *Transactions on Aspect-Oriented Software Development VII*, Springer, 2010, pp. 105–143. doi: 10.1007/978-3-642-15171-6\_5.
- [41] A. de Pádua Albuquerque Oliveira and L. M. Cysneiros, “Defining Strategic Dependency Situations in Requirements Elicitation,” in *Proceedings of the 8th Workshop in Requirements Engineering (WER)*, Rio de Janeiro, Brazil, 2006, pp. 12–23.
-

- [42] F. Dalpiaz, X. Franch, and J. Horkoff, “iStar 2.0 Language Guide,” May 2016, Accessed: Aug. 27, 2023. [Online]. Available: <https://arxiv.org/abs/1605.07767v3>
- [43] E. Gonçalves, J. Castro, J. Araújo, and T. Heineck, “A Systematic Literature Review of iStar extensions,” *Journal of Systems and Software*, vol. 137, pp. 1–33, 2018, doi: <https://doi.org/10.1016/j.jss.2017.11.023>.
- [44] J. Kontio, J. Bragge, and L. Lehtola, “The focus group method as an empirical tool in software engineering,” in *Guide to Advanced Empirical Software Engineering*, Springer London, 2008, pp. 93–116. doi: 10.1007/978-1-84800-044-5\_4/COVER.
- [45] J. Kontio, L. Lehtola, and J. Bragge, “Using the focus group method in software engineering: obtaining practitioner and user experiences,” in *International Symposium on Empirical Software Engineering*, Redondo Beach, USA: IEEE Press, 2004, pp. 271–280. doi: 10.1109/ISESE.2004.35.
- [46] E. S.-K. Yu, “Modelling strategic relationships for process reengineering”. PhD thesis, University of Toronto Computer Center Toronto, Ont. M5S 1A1, Canada, 1996. [Online]. Available: <https://dl.acm.org/doi/10.5555/269793>.