The final publication is available at

http://dx.doi.org/10.1016/j.is.2015.03.007

Additional Information

GoBIS: an integrated framework to analyse the Goal and Business process perspectives in Information Systems

Marcela Ruiz[*1], Dolors Costal[+], Sergio España[*], Xavier Franch[+], Óscar Pastor[*]

PROS Research Centre, Universitat Politècnica de València
Camino de Vera s/n, Edificio 1F, 46022 Valencia, Spain
+*GESSI Research Group, Universitat Politècnica de Catalunya (UPC)*
Calle Jordi Girona 1-3, Campus Nord, Edifici Omega, 08034 Barcelona, Spain

Abstract

Context: Organisational reengineering, continuous process improvement, alignment among complementary analysis perspectives, and information traceability are some current motivations to promote investment and scientific effort for integrating goal and business process perspectives. Providing support to integrate information systems analysis becomes a challenge in this complex setting. **Objective**: The GoBIS framework integrates two goal and business process modelling approaches: *i** (a goal-oriented modelling method) and Communication Analysis (a communication-oriented business process modelling method). **Method**: In this paper, we describe the methodological integration of both methods with the aim of fulfilling several criteria: i) to rely on appropriate theories; ii) to provide abstract and concrete syntaxes; iii) to provide scenarios of application; iv) to develop tool support; v) to provide demonstrable benefits to potential adopters. **Results**: We provide guidelines for using the two modelling methods in a top-down analysis scenario. The guidelines are validated by means of a comparative experiment and a focus-group session with students **Conclusions**: From a practitioner viewpoint (modeller and/or analyst), the guidelines facilitate the traceability between goal and business process models, the experimental results highlight the benefits of GoBIS in performance and usability perceptions, and demonstrate an improvement on the completeness of the latter having an impact on efficiency. From a researcher perspective, the validation has produced useful feedback for future research.

Keywords
Modelling language, requirements engineering, goal modelling, business process modelling, ontological analysis, metamodel integration, performance and usability analysis, *i**, iStar, Communication Analysis.

1 Introduction

Organisations are aware of the importance of evolving to keep pace with changes in the market, technology, environment, law, etc. [1]. As a result, continuous improvement and reengineering have

---

[1] Corresponding author. Tel.: +3496 387 70 00 Ext. 83534.
E-mail addresses: lruiz@pros.upv.es (M. Ruiz), dolors@essi.upc.edu (D. Costal),
sergio.espana@pros.upv.es (S. España), franch@essi.upc.edu (X. Franch), opastor@pros.upv.es (O. Pastor).

become common practices in information system (IS) engineering. Understanding organisations and their needs for change often requires several interrelated perspectives [2, 3]. The IS engineering community has contributed a number of modelling languages that are typically oriented towards a specific perspective, requiring approaches to their integration [4].

In this paper, we focus on extending a business process perspective with intentional aspects of organisations. Business process modelling languages provide primitives to specify work practice (i.e., activities, temporal constraints, and resources). Despite the fact that processes are widely accepted as a means to achieve organisational goals [5], process models give little attention to the strategic dimension [6]. The analysis, prioritization, and selection of organisational strategies are the scope of intentional modelling languages, which focus on the business roles, their goals, and their relationships.

Business processes and goals are intrinsically interdependent [7] and several works provide detailed arguments in favour of combining both perspectives: (i) An integrated approach allows understanding of the motivation for processes [6]; (ii) in a non-integrated approach, goals may be used to guide process design [8]; (iii) traceability is enhanced, which is necessary for enterprise management [9] and facilitates the sustainability of organisations [10]; (iv) integration also helps in identifying cross-functional interdependencies during business change management by supporting the identification of the goals that motivate change and the analysis of their impact on processes [8, 10, 11].

The GoBIS framework (Goal and Business Process Perspectives for Information System Analysis) pursues this aim by integrating a goal-oriented and a business process-oriented modelling language. There are several criteria that one would expect from modelling language integration. For the framework definition, we highlight the following: (i) The languages to combine need to be formally described; (ii) the integration itself should be well founded in theory; (iii) it should clarify the scenarios where the integrated approach can be applied and provide some scenario-dependent guidelines; (iv) it should provide tool support; (v) empirical studies should demonstrate some benefits to potential adopters. These criteria guided our research. A comparative review (see Section 2) reveals that proposals with similar aims do not fulfil one or several of the above-mentioned criteria, revealing that the challenge remains open.

This paper presents our design science endeavour [12] from the problem investigation to the solution design, the implementation of a modelling tool, and the solution evaluation. We have chosen to integrate the languages proposed by *i** [3] (a goal-oriented modelling method), and Communication Analysis (CA) [13] (a communication-oriented business process modelling method). The reason for choosing *i** is its expressiveness to specify dependencies, with which we intend to trace strategic motivations and processes. In the case of CA, we aim to get the most out of the communicational techniques in order to analyse business processes; its notation is not what is important, but rather the underlying concepts and modularity guidelines. Moreover, some current business process modelling suites use BPMN with a communicative approach. In addition, the authors are competent in these languages and are able to confront the challenge.

The contributions of the paper are the following:

- We present the *iStar2ca* guidelines v2.0 which are intended for a top-down modelling scenario and are an evolution of the *iStar2ca* guidelines v1.0 reported in [14] (see the evolution time-line in Figure 1).

The guidelines design is a method engineering effort; throughout the paper, we use the terminology introduced in [15].

- We report a comparative experiment and a focus-group session, which were carried out with students and whose feedback has been taken into account to produce the *iStar2ca* guidelines V2.0. The results demonstrate that the subjects' effectiveness (specifically, business process model completeness) is greatly improved by the use of the iStar2ca guidelines, without compromising efficiency. Also the perceptions of the usability of the guidelines are positive.
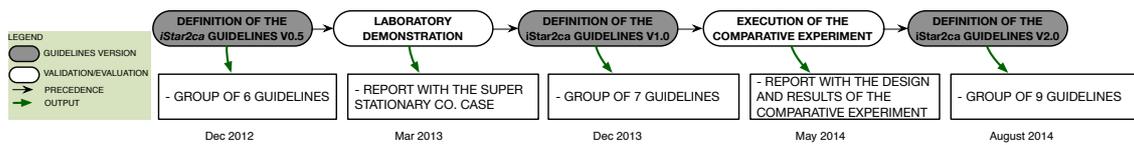

Figure 1.   Evolution time-line for the *iStar2ca* guidelines

The paper is structured as follows. Section 2 compares related works. In the Section 3, we introduce and exemplify the methods selected for integration. Section 4 describes the research methodology and Section 5 details the process and proposal for integrating *i** and CA. Section 6 presents guidelines for a top-down modelling scenario. Section 7 describes the modelling tool and technical support. Section 8 presents how the top-down scenario guidelines were evaluated through a lab-demo, a comparative experiment, and a focus group session with students. Finally, Sections 9 and 10 conclude with a discussion and future lines of work.

2   Related work

The combination of different methods and models to obtain a profound and comprehensive understanding of the system to be produced has received a lot of attention in the IS literature. On the one hand, some general frameworks have been proposed to reconcile different perspectives; for instance, Salay et al., [16] propose the concept of macromodel for the development, comprehension, consistency management, and evolution of a collection of related models. On the other hand, different approaches focus on some of these perspectives (typically two) in a specific domain of interest and provide a customized solution for their particular case. This is the approach followed in the GoBIS framework presented in this paper.

There are several related works that focus on the integration of goal and business process perspectives in the domain of business process management and maintenance. These approach goal-oriented business process reengineering from diverse angles. We analyse these approaches based on the criteria mentioned in the introduction (see Section 1), which may take different values:

(a) Ontological foundation: none; conceptual framework (concepts and their relationships are defined); based on ontologies (but without an explicit mapping); ontological mapping.
(b) Metamodel integration: yes, no, N/A (not available).
(c) Modelling scenarios: top-down (from goal model to business process model); bottom-up; iterative (switching back and forth among between the two goal and business process models); evolution (business processes evolve according to the goal models); N/A (no information provided to judge).
(d) Tool support: yes, no (depending on whether a tool has been implemented or not).

(e) Performed validations/evaluations: lab-demo (laboratory demonstrations), case studies, controlled experiments, none.

Some works focus on modelling the *as-is* system (the reverse engineering process in reengineering frameworks). For instance, Andersson et al., [17] propose a goal elicitation method to deepen the understanding of current processes. The authors conclude that a suitable semantics and representation to relate goal and business process models is needed, which is a motivation for our proposal. Guizzardi et al., [6] discuss the alignment of goal and process modelling methods (using Tropos and ARIS, respectively) and propose a three-stage method to model the as-is system.

Other works focus on supporting the evolution of the business process model, i.e., from the *as-is* to the *to—be* systems. Cardoso et al., [18] propose a goal-based pattern definition language for business process evolution, where processes are trajectories in a space of all possible states and goals are final states. Soffer et al., [19] present a formal approach for analyse the dependency of softgoals on processes; as a practical result, they enable the evolution rationale to be modelled.

Table 1. Summary of the review of the state of the art

Refs	Ontological Foundation	Metamodel Integration	Modelling Scenario	Tool Support	Validation/ Evaluation
[18]	Conceptual framework	N/A	Evolution	No	Case study
[17]	None	No	Top-down	No	Lab-demo
[5]	Based on ontologies	No	N/A	No	Lab-demo
[6]	Based on ontologies	No	Top-down	No	Case study
[11]	Conceptual framework	Yes	Iterative	No	Lab-demo
[9]	None	No	Top-down	No	Lab-demo
[8]	Conceptual framework	No	Top-down	No	Lab-demo
[10]	Conceptual framework	No	N/A	No	Lab-demo
[19]	Based on ontologies	No	Evolution	No	Lab-demo
[20]	None	No	Top-down	No	Lab-demo
Ours	Ontological mapping	Yes	Top-down (potentially may support the rest)	Yes	Lab-demo Comparative experiment,

Other works focus on modelling the *to-be* system (forward engineering). Kueng et al., [8] present an informal, seminal approach in which goals provide a basis for process definition. Kavakli et al., [11] define a method that takes an *as-is* business process model as input and produces a *to-be* goal model and a *to-be* business process model. Leonardi et al., [20] propose a set of heuristics in order to specify business process models (based on the CA method) from goal-oriented specifications. This work provides a strategy to link processes and organisational objectives in *to-be* scenarios.

Some of the above-mentioned works elaborate a conceptual framework to clarify definitions [6, 8, 11, 18], and [19] even builds upon an existing ontology. However, none of them performs an ontological analysis to guide the integration of the modelling methods, which is our selected approach.

With regard to modelling language integration, Kavakli et al., [11] rely on EKD metamodels (goal and business process perspectives are integrated *a priori*), but, noticeably, none of the above-mentioned works report a proper, rigorous metamodel integration (Guizzardi et al., [6] does mention it as future work).

We have taken the previous works as a reference and attempted to cover the gaps in terms of ontology-based analysis, metamodel integration, and tool support (see last row in Table 1). Some works analyse semantic relations between goals and business processes [5], [3, 9, 10], which can be used as input for the definition of the guidelines. Similarly, the pattern-based approach in [18] could be adapted to the context of *i** and CA.

It is worth mentioning approaches that enrich goal models in order to include information related to business process. Lapouchnian et al., [21] propose adding textual annotations to goal models in order to add control flow details for subgoals (e.g., data dependencies, precedence constraints). Ghose et al., [22] enrich goal models with precedence relationships, which facilitates the derivation of business processes. Similarly, Kazhamiakin et al., [23] define a set of formal annotations in goal models in order to add constraints in goal models for future operationalization of goals. All these approaches extend the expressiveness of goal modelling languages. However, we intend to keep goals and processes as separate and complementary perspectives, which is a strategy that is in line with common practices in the area of enterprise modelling architecture [24]. Instead of targeting the generation of complete and fully-detailed process models, GoBIS facilitates creating initial process models that are later extended by business analysts, who perform additional requirements elicitation and re-modelling (which are out of the scope of this paper).

3 Background and running example

This section presents an overview of the languages being integrated, *i** and CA. Throughout the paper, we use a running example based on the case of SuperStationery Co., which is an intermediary company that buys and sells office material. The focus of the example is on the intentional and operational aspects of sales management.

3.1 The *i** framework in a nutshell

The *i** framework is a goal- and agent-oriented requirements engineering method [3]. It proposes two models: a Strategic Dependency (SD) model, which represents the intentional level; and the Strategic Rationale (SR) model, which represents the rational level. An SD model consists of a set of nodes that represent actors and a set of dependencies that represent the relationships among them, expressing that an actor (depender) depends on other actor (dependee) in order to obtain an objective (dependum). The dependum is an intentional element that can be a resource, task, goal, or softgoal. An SR model allows the intentional elements to be visualised inside the boundary of an actor in order to refine the SD model with reasoning capabilities. The dependencies of the SD model are linked to intentional elements inside the boundary of the actor. The elements inside the SR model are decomposed according to three types of links: task-decomposition links, means-end links, and contribution to softgoal links.

Actors can be specialised into agents, roles, and positions. Agents are actors with a concrete physical manifestation. Roles are abstract characterizations of the behaviour of a social actor within a context. A position covers roles.

Figure 2 shows an excerpt of an *i** model for the SuperStationery Co. case. Due to space limitations, the SR diagram is shown only for some of its actor roles: `Client`, `Supplier`, `Truck driver`, and `Insurance dept clerk`.
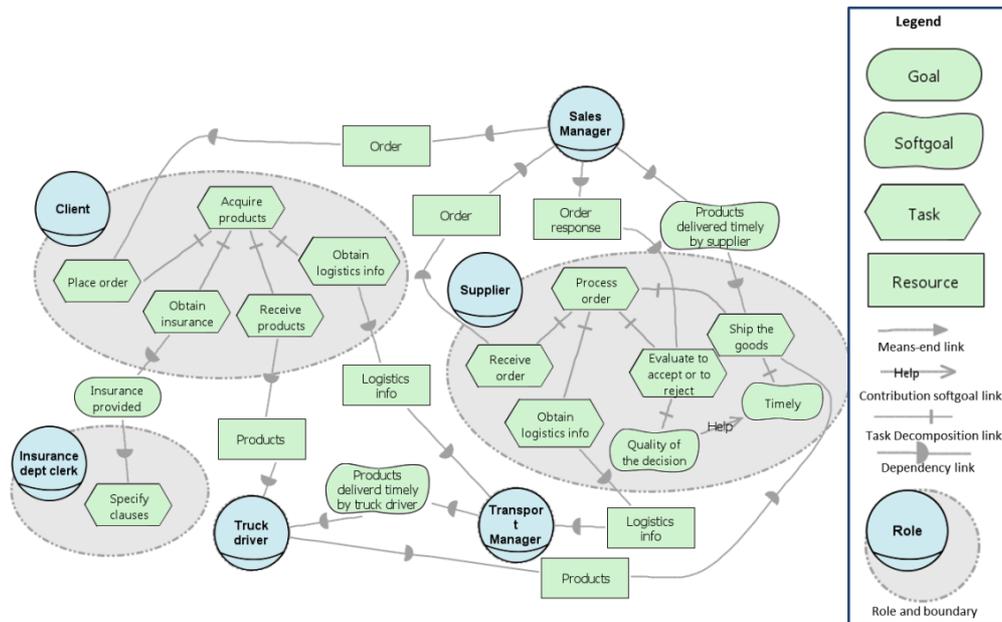

Figure 2.   Excerpt of an *i** model for the SuperStationery Co. case

3.2   The Communication Analysis method in a nutshell

CA is a requirements engineering method that analyses the communicative interactions between the IS and its environment [13]. CA provides business process modelling and gets the most out of communicational techniques in order to analyse business processes. In this way, it improves conventional proposals for business process modelling. Therefore, the method focuses on external IS functions: information acquisition and distribution. CA offers requirements structure and several modelling techniques: 1) the Communicative Event Diagram (CED) describes business processes from a communicational perspective; 2) the Event Specification Template allows the structuring of the requirements; and 3) Message Structure specifies the description of new meaningful information that is conveyed to the information system in the event [25].

Figure 3 shows an excerpt of a CA model for the SuperStationery Co. case. It includes the CED for the management of a sale. The CED (see Figure 3) consists of communicative events (CE). A CE is an organisational action that is triggered as a result of a given change in the world (e.g., `A client places an order`) and accounts for that change by gathering information about it. A CE is structured as a sequence of actions that are related to information (acquisition, storage, processing, retrieval, and/or distribution), which are carried out in a complete and uninterrupted way. CE are identified by the guidelines (referred as unity criteria) which act as modularity guidelines [26]. CE can be specialised by

means of event variants, which are alternative transitions that define the composition of a CE (e.g., in Figure 3, `Supplier evaluates the order` is specialised into `Order is rejected` or `Order is accepted`). In addition, CED specifies primary roles that trigger the CE and provide the input information, the receiver roles that need to be informed of the occurrence of an event, the interface roles that are in charge of editing and entering input information, and the relationships to specify ingoing and outgoing communicative interactions, and precedence relationships among CE. In the example, in the CE `A client places an order`, the `Client` acts as primary role, the `Manager` as receiver role, and the `Salesman` as interface role.
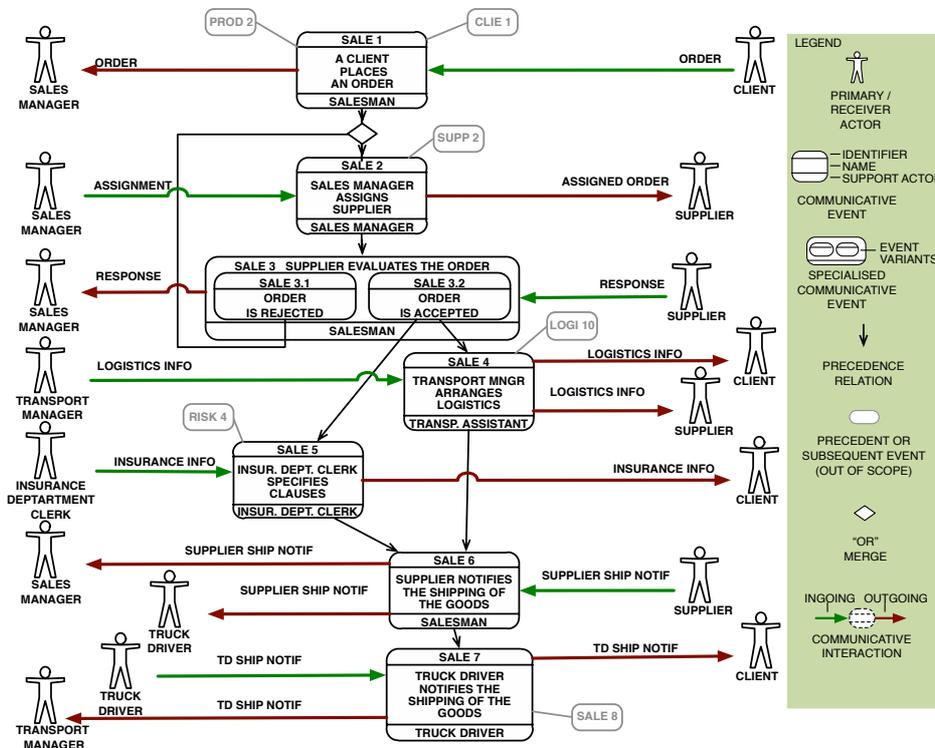
Figure 3. Excerpt of a CA model for the SuperStationery Co. case

4 Research Methodology

We structure our research in terms of design science since it involves creating new artefacts and acquiring new knowledge (see Figure 4). Our research methodology follows the cycles described by Wieringa [12]. We performed two engineering cycles (EC1 and EC2) that mainly create an integrated metamodel, the *iStar2ca* guidelines V0.5 and V1.0 (see Figure 1). For the validation of the guidelines we performed a research cycle (RC1) that consists of a comparative experiment and a focus group session that led to the creation of the *iStar2ca* guidelines V2.0.

13. IMPLEMENTATION VALIDATION
- Ascertain the operability of the tool
- Asses the stakeholders
satisfaction and evaluate the criteria
of success (future work)

1. PROBLEM INVESTIGATION (Section 2)
- Investigate the need for a goal-process integration (revisit own experience in software projects, literature review)
- Define the criteria to judge solution success
- Define a procedure for ontology-based language integration

2. ARTEFACT DESIGN (Section 5)
-Explore available solutions by reviewing the SOTA
- Design a new solution
- Select the methods for integration
- Perform ontological analysis of selected methods
- Investigate and analyse overlapping constructs
- Integrate the language metamodels
- Provide modelling guidelines: conception of the modelling *iStar2ca* guidelines V0.5

12. SOLUTION IMPLEMENTATION (Section 7)
- Provide an Eclipse-based integrated modelling tool
- Transfer the solution to Pros-Req software development framework (Future work)

SOLUTION DESIGN (REPRISE OF STEP 5)
Provide a refined version of the modelling guidelines accounting the experimental results,: the *iStar2ca* guidelines V2.0

EC1 INTEGRATE GOAL-ORIENTED AND BUSINESS PROCESS-ORIENTED LANGUAGES

3. SOLUTION VALIDATION (Section 6)
- Demonstrate the feasibility by means fo a lab-demo: : Report with the SuperStationery Co.
- Evaluate the benefits and the sensitiviy of the solution (future work)

11. RESULTS EVALUATION
- Analysis of collected data: practitioners' performance and usability perceptions
- Application of statistical test
Corroboration of hypothesis

EC2 CREATE A REFINED VERSION OF THE *iStar2ca* GUIDELINES

4. PROBLEM INVESTIGATION (Section 2)
- Explore existent solutions and related works

10. RESEARCH EXECUTION

9. DESIGN VALIDATION
- Evaluate the experimental design
- Review the instruments for collecting data
- Analyse threats on the experiment validity

RC1 COMPARATIVE EXPERIMENT WITH PRACTITIONERS (Section 8)

5. ARTEFACT DESIGN (Section 6)
-Provide a refined version of the modelling guidelines: the *istar2ca* guidelines V1.0

6. SOLUTION VALIDATION (Section 8)
- Evaluate the benefits and practitioners' performance and perceptions

8. RESEARCH DESIGN
- Define the experimental context
- Design the experiment

7. RESEARCH PROBLEM INVESTIGATION
-Establish the research questions of the comparative experiment, variables and hypothesis.

Figure 4.   Overview of the research methodology

5   Modelling language integration

In order to combine the process and intentional perspectives, we undertake a method engineering effort [15], where *i** and CA are considered method chunks. Note that the analysis of project situations is out of the scope of this paper. Instead, the focus placed on integrating the product and the process models of the methods. Taking the integration map proposed in [15] as reference method, Figure 5 presents how we have operationalised each of its intentions and points to the corresponding section.

5.1   Ontological mapping between *i** and CA

Integrating the product models of two methods requires identifying pairs of constructs that have the same semantics so as to merge them later. When the two product models have different terminology, as in the case of *i** and CA, Ralyté and Rolland suggest adapting the product models by means of name unification and transformation [15]. We have opted for ontological analysis, which is an equivalent strategy that offers strong theoretical foundations to method analysis and comparison. In an ontological analysis, the constructs of a method are mapped to the constructs of reference ontology. This is commonly used to assess to what extent the method covers the constructs of the ontology, and vice versa. Among other different possible options (e.g., BWW, Chisholm's, DOLCE, UFO, etc.), we have chosen FRISCO [27] as the reference ontology. FRISCO is intended to provide a conceptual framework and suitable terminology for the most fundamental concepts in the information system field, including the notions of information, communication, organisation, and information system. Therefore, it naturally fits the analysis of methods such as *i** and CA, which offer different perspectives for information system

engineering. In fact, we have experienced the appropriateness of this ontological analysis in a previous analysis related to one of the two methods, CA [28].
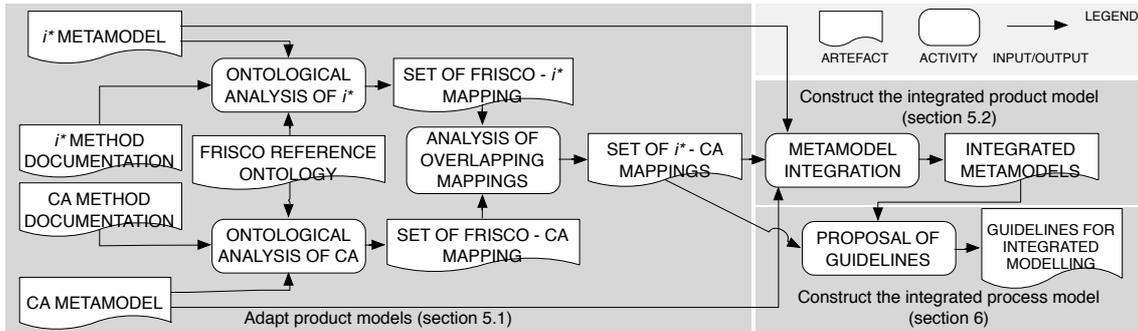

Figure 5.   Flow of modelling language integration
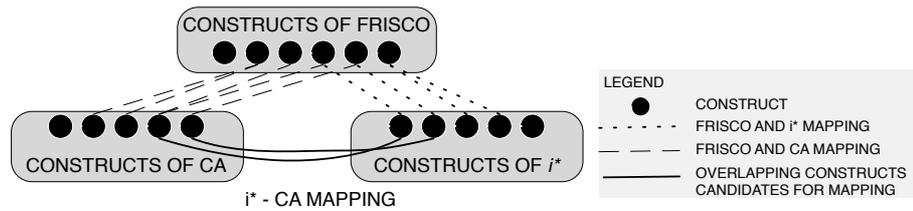

Figure 6.   Ontological analysis as a means to map constructs and guide method integration

The procedure is illustrated in Figure 6. We have performed a separate ontological analysis of each method, establishing a complete mapping first between, the constructs of *i** and the constructs of the FRISCO ontology and then between the constructs of CA and those of the FRISCO ontology[2]. These complete mappings are reported in [29]. Since we are not interested in the criteria commonly applied in ontological analyses (e.g., construct excess, laconicism), we instead identify which pairs of constructs from each method are mapped onto the same ontological FRISCO construct (this is an alternative way of verifying construct similarity [15]). These constructs are considered overlapping, and, therefore, they are candidates for the *i**-CA mapping. Finally, we analyse the overlap and decide whether to map the concepts unconditionally (i.e., in all cases) or under certain conditions (see [14] for details).

Table 2 summarises the overlapping constructs found in this analysis. Each row describes a pair of constructs that overlap. FRISCO mappings consist of FRISCO constructs (underlined) that are qualified when necessary. Table 2 also indicates the ontological mappings between *i** and CA that were decided in view of the FRISCO mappings (additional information of the methods was necessary). They are unconditional except in two cases. The CA method provides a set of unity criteria to identify and encapsulate communicative events that help to define them at an appropriate level of modularity (see [26] for details). Therefore, an *istar.task* is aligned to a *ca.communicative event* only if it satisfies the unity

---

[2]The *i**– FRISCO mapping was performed by the second author and revised by the fourth author. The CA – FRISCO mapping was performed by the third author and revised by the first author. Then all authors met to discuss the mappings,identify the candidate overlapping concepts and decide the alignment.

9

criteria. An *istar.resource* is aligned to a *ca.message structure* only if it is informational (e.g., a delivery note). Hence, *istar.physical resource* is not aligned to *ca.message structure* (e.g., a pallet of boxes is not a message structure).

Table 2. Mappings of candidate overlapping concepts

i− FRISCO mapping		CA − FRISCO mapping		*i*− CA mapping
*i**	**FRISCO**	**CA**	**FRISCO**	
agent	actor with a concrete physical manifestation, for instance, a human actor that carries out actions to achieve goals by exercising its know-how	organisational actor	goal-pursuing actor	equivalent
role	type of actors such that it characterizes the behaviour of agents	organisational role	type of goal-pursuing actors	equivalent
goal	goal that is an intentional desire of an actor	organisational goal	goal of an actor of an organisational system	equivalent
task	action that involves one actor in its pre-state and in its post-state	communicative event (CE)	composite transition	CE -> task task -> CE (1)
resource	input actand of an action (if it is physical) or data that is the input actand of an action (if it is informational) such that an actor desires its provision and there are no open issues about how it will be achieved	message structure (MS)	type of messages; it is an input actand of a composite transition (i.e., the "communicative event")	MS -> resource resource -> MS (2)

Mapping conditions: (1) task satisfies unity criteria (2) resource is informational

We have opted for ontological analysis to guide our cognitive process during the integration of *i** and CA concepts. It has provided us with a systematic way to map concepts and has prevented possible biases due to our subjective preconceptions (e.g., *istar.task* and *ca.communicative event* may look equivalent intuitively, but the systematic analysis has shown that they only map under specific conditions). More specifically, the lessons learned from the application of ontological analysis are the following: (1) the systematic ontological analysis has helped us to focus only on concepts with objective reasons for being candidate overlapping concepts (i.e., only concepts that map into the same or related FRISCO concepts are candidates for mapping); (2) the FRISCO mappings have provided the rationale to decide whether two concepts are totally equivalent or they map under specific conditions; and (3) in the latter case of concepts mapping under specific conditions, the FRISCO mappings have contributed to identifying those conditions.

5.2 Metamodel integration: the GoBIS metamodel

To integrate *i** and CA metamodels, we analysed the *i**-CA mapping of constructs presented in Section 5.1. For each pair of mapping constructs, we need to decide whether we keep both corresponding metaclasses (one for each modelling method) or just one metaclass. We provide some heuristics to make this decision and the implications of each choice (see Figure 7.a for the starting point).

In some cases, the two constructs are totally equivalent in the sense that their mapping is clear-cut (the constructs in the first three rows in Table 2 fall into this category). In these cases, the simplest solution is to keep only one metaclass and it needs to be decided which of the two involved metaclasses is removed.

Then, the relationships in which the removed metaclass, participated must be connected to the metaclass that is kept (see Figure 7.b). In other cases, the mapping of two constructs is qualified with a condition specifying under what circumstances both constructs can be mapped (the constructs in the last two rows of Table 2). Thus, we propose to keep both metaclasses and create a relationship between them to provide traceability in cases where the specific constructs are mapped (see Figure 7.c). The application of these heuristics to our case is summarised in Table 3.

(a) Example of two metamodels to integrate | (b) Case in which the constructs are totally equivalent (A≡C) and we decided to keep A | (c) Case in which the mapping is conditional (A≅C)
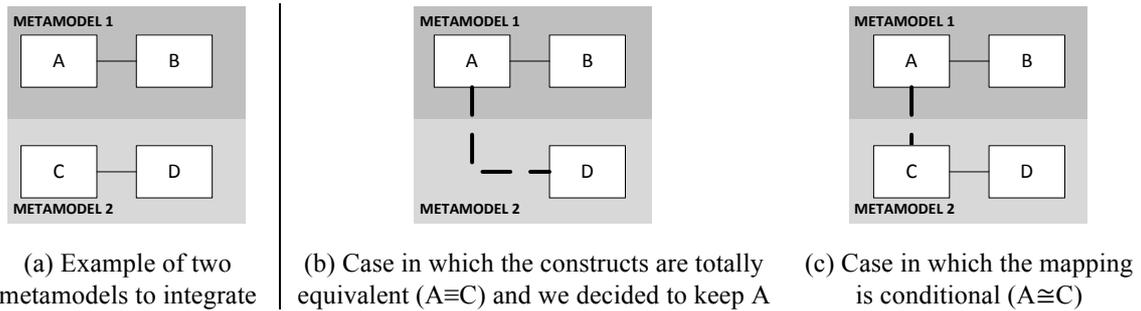
Figure 7.   Deciding which metaclasses to keep

Table 3. Metamodel integration

i* metaclass	CA metaclass	Metaclasses kept	Rationale
AGENT	ORGANISATIONAL_ACTOR	AGENT	- Equivalent constructs
ROLE	ORGANISATIONAL_ROLE	ROLE	- i* provides more detailed definitions
GOAL	GOAL	GOAL (from i*)	
TASK	COMMUNICATIVE_EVENT	Both	- Equivalent under specific conditions
RESOURCE	MESSAGE_STRUCTURE	Both	

To create the integrated GoBIS metamodel, we have started from: (1) the i* metamodel presented in [30] (adapted to be compliant to the *i** wiki version taken as reference in this work), and (2) the CA metamodel [28]. Both source metamodels are described in detail in [29] along with a discussion on why we selected them among other options. Following Table 3, all metaclasses from the source *i** metamodel have been kept in the GoBIS metamodel. Conversely, metaclasses ORGANISATIONAL_ROLE, ORGANISATIONAL_ACTOR and GOAL that are part of the CA metamodel have been removed. A detailed description of the integrated metamodel is presented in [29]

6   Top-down scenario guidelines: *iStar2ca guidelines 2.0*

We present the *iStar2ca* guidelines version 2.0 for a top-down scenario, an evolution of the *iStar2ca* guidelines V1.0 motivated by the empirical results described in the Section 8. The guidelines facilitate obtaining a CA model having as input a given *i** model and establishing the mappings identified at a metamodel level. They indicate how to derive *ca.communicative events* and *ca.message structures* since these CA elements only map into *i** elements under specific conditions. As seen in the previous section, *ca.organisational actors*, *ca.organisational roles*, and *ca.goals* are always mapped from *istar.agents*, *istar.roles*, and *istar.goals*.

Due to the strategic focus of *i** models, some informational *istar.resources* or some *istar.tasks* that should map into *ca.message structures* or *ca.communicative events* may not be explicitly represented if they do not add strategically relevant knowledge. The *iStar2ca* guidelines not only provide advice on how to obtain CA elements from explicit *i** elements but also on how to derive CA elements from *i** elements that are not explicit but whose existence can nevertheless be deduced from the model. For example, the existence of an implicit informational *istar.resource* `Insurance info` can be deduced from the istar.goal `Insurance provided` (see Figure 2).

CA focuses on communicational interactions among roles. Therefore, most guidelines involve *i** dependencies among roles because satisfying the dependency will generally require some type of interaction. Each type of dependum has an associated guideline except resource dependums since informational and physical resources require different treatment.

Guideline 1 deals with the case of dependums that are informational resources that (according to our metamodel alignment) map into *ca.message structures*.

> ***Guideline 1.*** The dependum of a dependency D among two roles maps into a message structure M if this dependum is an informational resource. In that case, D induces a communicative event C such that: (1) C's primary role is D's dependee role; (2) C's receiver role is D's depender role; (3) C's ingoing and outgoing interactions specify M; (4) if any of the SR elements of D's dependee and depender roles are tasks, they map into C.

In our SuperStationery Co. case (see Section 3), the resource `Order` of the dependency from `Sales Manager` to `Client` maps into the CA message structure `Order` (see ingoing interaction in Table 4). The dependency for the `Order` from the `Sales Manager` to the `Client` indicates that the communicative event `A client places an order` (which allows the `Client` to communicate the order to the `Sales Manager`) is needed.

Table 4. Guideline 1 applied to dependency for `Order` from `Sales Manager` to `Client`
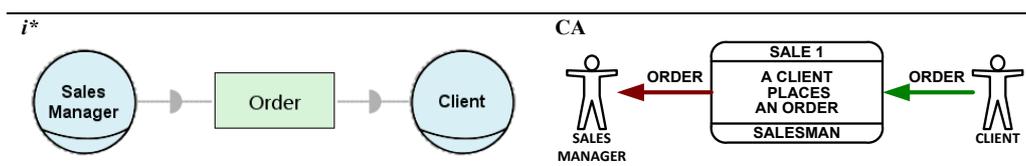
The rationale is that the *dependency* indicates that the *depender* expects to receive information from the *dependee*, and, therefore, a *communicative event* is needed to allow the *dependee* to communicate that information to the *depender*.

The *ca.interface actor*, however, cannot be determined from the *i** model. It may or may not coincide with the *primary actor*. It may even be an actor that does not appear in the *i** model at all because it is not strategically relevant. In the previous example, the *ca.interface actor* `Salesman` does not appear in the *i** model.

Two *istar.tasks* (i.e., one for the *dependee* and another for the *depender*) may map into a single *communicative event*. The reason is that *i** provides a separate SR diagram for each actor and then the behaviour of a single *communicative event* with two involved actors appears distributed in two *istar.tasks*

that are visualized inside the boundary of the two actors. In our example, the task `Place order` in the `Client` SR maps into this new communicative event and a task of the `Sales Manager` SR (not shown in Figure 2 for reasons of space) also maps into it.

A dependency may be connected to SR elements that are not tasks, thus indicating that the task of communicating the resource information is implicit in the *i** model.

Guideline 1 is also applied to the dependencies for `Order` (from the `Supplier` to the `Sales Manager`), `Order response` (from the `Sales Manager` to the `Supplier`), and `Logistics Info` (from `Supplier` and `Client` to the `Transport Manager`) in order to map to the communicative events `SALE 2`, `SALE 3`, and `SALE 4`, respectively (see Figure 2 and Figure 3).

The following group of guidelines deals with the rest of the dependency types (i.e., where dependums are goals, tasks, softgoals, or physical resources). Although these dependum types do not map directly into *ca.message structures*, they may indicate the existence of informational resources that are not explicit in the *i** model. We define an abstract guideline that yields to four actual guidelines (from Guideline 2 to 5) depending on the type of dependum.

Abstract guideline. The dependum of a dependency D between two roles induces a message structure M the dependee is required to give information to the depender about the <u>intentional satisfaction</u> of this dependum. In that case, D induces a communicative event C such that: (1) C's primary role is D's dependee role; (2) C's receiver role is the D's depender role; (3) C's ingoing and outgoing interactions specify M; (4) if any SR elements of D's dependee and depender roles are tasks, those tasks map into C.

We call this dependency an *informationable dependency*. The actual guidelines refine the notion of *intentional satisfaction* according to the type of the dependum.

Guideline 2. When the dependum is a goal, the notion of <u>intentional satisfaction</u> of the abstract guideline refines into attainment of this goal.

The informational resource mapping into the new *message structure* is implicit in the *i** model. In our example, the goal dependency for `Insurance provided` from the `Client` to the `Insurance dept clerk` is an informationable dependency because the client needs to receive the clauses of the insurance. The message structure `Insurance info` represents this information. The communicative event `Insur. Dept. clerk specifies clauses` is obtained with communicative roles `Insurance Dept Clerk` and `Client`. The tasks `Specify clauses` (of the `Insurance Dept Clerk`) and `Obtain Insurance` (of the `Client`) map into this event.

Guideline 3. When the dependum is a task, the notion of <u>intentional satisfaction</u> of the abstract guideline refines into accomplishment of this task.

Guideline 4. When the dependum is a softgoal, the notion of <u>intentional satisfaction</u> of the abstract guideline refines into level of satisfaction of this softgoal.

In our example, the softgoal dependency for `Products delivered timely by supplier` from the `Sales Manager` to the `Supplier` is an informationable dependency because the sales manager needs to be informed about the time when the products are shipped in order to supervise its timeliness. The message structure `Supplier ship notif` represents this information. The communicative event

`Supplier notifies the shipping of the goods` is obtained with the communicative roles `Supplier` and `Sales Manager`. There is another informationable softgoal dependency for `Products delivered timely by truck driver` from the `Transport Manager` to the `Truck Driver`. The event `Truck Driver notifies the shipping of the goods` is obtained with communicative roles `Truck Driver` and `Transport Manager`.

> ***Guideline 5.*** When the dependum is a physical resource, the notion of <u>intentional satisfaction</u> of the abstract guideline refines into the provision of this physical resource.

The dependency for the physical resource `Products` from `Truck Driver` to `Supplier` leads to the creation of the communicative event `Supplier notifies the shipping of the goods`, which is merged with the notification that the `Supplier` gives to the `Sales Manager` (`SALE 6`). Thus, we add the receiver role `Truck Driver` to `SALE 6`. Similarly, the dependency for `Products` from `Client` to `Truck Driver` leads to the addition of the receiver actor `Client` to `SALE 7`. In the general case the decision of merging events is not derivable from the *i** model only and must be decided in an ad-hoc way by analysing whether the events can be performed simultaneously. Specific cases in which event merging can be deduced from the *i** model are captured by Guideline 8.

> ***Guideline 6.*** An actor about which relevant information must be registered in the IS indicates that a communicative event and its corresponding message structure must be specified in order to register the actor information.

This guideline is only applied if the information about the actor is necessary. In our example, a message structure is required for `Client` in order to keep a registry of clients. Some of the information to be kept is: `VAT number, Client name, Telephone, Registration date, Client Addresses`. The communicative event `Clie 1` is also specified; it is not visible in Figure 3 because it is part of another process (i.e., `Client management`; acronym `CLIE`).

In general, an *i** model does not provide information to deduce the ordering of the communicative events obtained from it or whether several communicative events can be merged into a single one and be performed simultaneously. However, there two are specific cases in which we can obtain the ordering of events as described in Guideline 7 and the merging of events as described in Guideline 8.

In some cases where there are two dependencies with the same dependum in an *i** model, a precedence between the two mapped communicative events is implicitly induced. Guideline 7 represents this case.

> ***Guideline 7.*** Two dependencies (D1 and D2), mapping into two communicative events (C1 and C2), indicate that C1 precedes C2 in the communicative event diagram if:
> (1)   D1 and D2 have the same dependum
> (2)   the depender of D1 is the dependee of D2

In our example, there are two dependencies with the same dependum `Order` such that the `Sales Manager` is the depender in one and the dependee in the other. This indicates that the communicative event `A client places an order` where the `Sales Manager` receives the order must precede the event `Sales Manager assigns supplier` where the `Sales Manager` provides the order to the `Supplier`. Similarly, from the two dependencies with the dependum `Products`, it follows that the event

`Supplier notifies the shipping of the goods` must precede the event `Truck driver notifies the shipping of the goods`.

If an *i** model has two dependencies with the same dependum and dependee role, this indicates that the two mapped communicative events should be merged. Guideline 8 corresponds to this case.

Guideline 8. Two dependencies, (D1 and D2), mapping into two communicative events (C1 and C2), indicate that C1 and C2 can be merged into a single communicative event C in the communicative event diagram if:

(1)   D1 and D2 have the same dependum

(2)   D1 and D2 have the same dependee

In our example, the two dependencies for the `Logistic Info` from `Supplier` and `Client` to the `Transport Manager` map to the communicative event `SALE 4` according to Guideline 1. Note that that the resulting communicative event is a merge of the communicative events induced by each dependency. The communicative event has the common dependee role (`Transport Manager`) as primary actor and its ingoing and outgoing interactions specify the message structure corresponding to the common dependum. In this case, the resulting communicative event has two receiver roles, which correspond to the depender roles (`Supplier` and `Client`).

For traceability purposes, Guideline 9 provides some tips for naming the elements of the CED obtained by the guidelines. This guideline applies in the case where the dependee's SR related element is a task. Because the names of the CED elements are derived from the *i** elements, we suggest following the good practices for naming *i** elements reported in [3].

Guideline 9. Assume a dependency D that induces a communicative event C such that C's primary and receiver actors are P and R respectively; and assume C's message structure is M. The recommended names for C, P, R, and M are determined as follows:

- C is based on the name of D's dependee + the name of the SR related element of the dependee+ the name of D's dependum (optional)
- P is based on the name of D's dependee role
- R is based on the name of D's depender role
- M is based on the name of D's dependum if D's dependum is an informational resource; otherwise M is based on the name of the intentional satisfaction that corresponds to D's dependum.

In our example, the communicative event `SALE 1` is derived from the dependency for `Order` from `Sales Manager` to `Client`. Then, the names of the dependee (`Client`), the Client's SR related element (`Place order`), and `Order` are used to give a name to `SALE 1` (`A client places an order`). The name of the dependee is used as the name of the primary role (`Client`), and the name of the depender is used as the name of the receiver role (`Sales Manager`). The message structure for the ingoing and outgoing relationships takes the name of the dependum `Order` since this dependum is an informational resource.

The CA elements obtained by applying the proposed guidelines are part of those that conform Communicative Event Diagrams (CEDs). On the other hand, the internal structure of *ca.message* structures is not obtained by applying the guidelines because *i** models do not provide the details about the resources. It is necessary to explore organisational documents to obtain it.

7   Tool support

Technological support for the GoBIS framework is necessary to carry out future case studies and facilitate industrial adoption. Although existing tools allow creating separate *i** and CA models, we aim to support the combined modelling (See Figure 8 with the combined modelling for the SuperStationery Co; the CA model in the right and the *i** model in the left). The CE SALE 1 is aligned with the task PLACE ORDER.
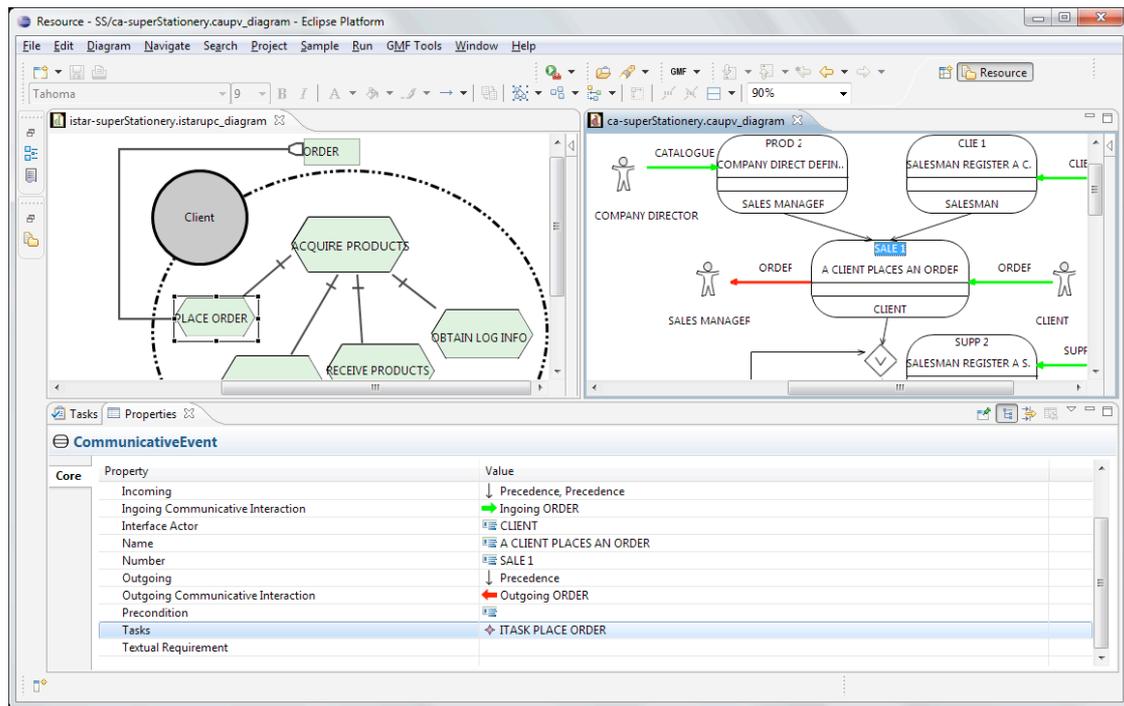

Figure 8.   Screenshot of the prototype: combined modelling of the *i** and CA for the SuperStationery Co.

We chose Eclipse (http://www.eclipse.org) as the technological platform. We used Eclipse Modelling Framework (http://www.eclipse.org/modeling/emf) and Graphical Modelling Framework (GMF, http://www.eclipse.org/modeling/gmp) to implement the metamodels and modelling tools for each method. We have followed a Model-Driven Architecture (MDA) [31] approach to develop a tool for both methods. This way, method and language specifications of *i** and CA correspond to the Computation-Independent Model layer of MDA. The abstract syntax of both methods are represented by means of Platform-Independent Metamodels (PIMm), which correspond to the Platform-Independent Model layer of MDA. According to these PIMm, we have specified the Platform-Specific Metamodels (PSMm) that are compliant with Eclipse. This PSMm correspond to the Platform-Specific Model layer of MDA. Finally, we defined the concrete syntax of both languages (graphical and textual appearance). The implemented tools correspond to the Code Model layer of MDA.

Previous works present a PSMm for CA models that are compliant with GMF [32]. We adapted it based on the result of the metamodel integration in Section 5. With respect to *i**, there are several metamodels available. We analysed the PIMm presented in [30] and we opted to maintain most of its concepts,

although it required some adaptations to account for the metamodel integration presented in Section 5 and to make it GMF-compliant. To design the PSM metamodel for *i**, we analysed three tool-oriented metamodels: the OpenOme metamodel [33], the metamodel presented by Giachetti [34], and the unified metamodel for *i** [35]. As a result, a combined modelling prototype is obtained with support for traceability link specification between CA and *i** modelling elements. See [29] for further information, screenshots, and technical details about the prototype.

The developed prototype helps the application of the *iStar2ca* guidelines (see Section 6) in the sense that after the manual application of the *iStar2ca* guidelines, the prototype makes traceability link specification possible among the modelling elements of *i** and CA. From a practitioner perspective, the prototype facilitates the analysis of explicit traceability links among modelling elements. As a future work, we plan to implement transformation rules and a wizard with automated application of the guidelines.

8   Validation by means of a controlled experiment

We have performed a comparative experiment to assesses the performance and perceptions of students applying the *iStar2ca* guidelines V1.0 [14]. This version includes Guidelines 1 to 7 (see Section 6). This experiment has been designed according to Wholin et al. [36], and it is reported according to Jedlitschka & Pfahl [37] and Juristo & Moreno [38].

8.1   Experimental design

The experiment goal, according to the Goal/Question/Metric template [39] is to **analyse** *the resulting CA* *models* **for the purpose** of *evaluation* **with respect to** their *performance and perception* **from the point of view of** *the researchers and practitioners* **in the context of** a last-year master *course at the* Universitat Politècnica de València (UPV). The research questions are formulated in Table 1.

Table 1. Research questions (RQ)

RQ	When the subjects are creating CA models from *i** models applying the iStar2ca guidelines…	To answer this question we compare … when they apply the iStar2ca guidelines to when they apply their own criteria
RQ1	… is their performance affected?	… the completeness and validity…
RQ2	… is their efficiency affected?	… their efficiency…
RQ3	… are their perceptions impacted?	… the perceived usefulness, perceived ease of use, and their intention to use…

The experiment was conducted in 2014 (from March to May) within the course of Information Systems Engineering[3] at the (UPV) in Spain. The subjects were 19 students with a background in software engineering and information system analysis methods and techniques. Prior to the experiment, the subjects filled out a demographic questionnaire[4] revealing that some of them had experience in industry (21,4%), most of them previously knew the CA method (71,4%), and few of them knew the *i** language

---

[3] http://www.upv.es/titulaciones/MUISMFSI/indexi.html
[4] The material for the demographic questionnaire and a replication package for this controlled experiment can be found at http://hci.dsic.upv.es/istar2ca_exp/

(7,1%). None of them had been in contact with the *iStar2ca* guidelines before. Based on this information, we designed an intensive training session in the *i** and CA methods in order to balance the knowledge of the subjects. The course contents and planning were updated in order to incorporate the experimental set-up, still maintaining the original course objectives. The subjects executed the experimental task as part of the course assignments and received grades for it, contributing to their motivation.

Variables. We consider one independent variable (a.k.a. factor [38]):

- **CA derivation strategy**. The strategy to obtain CA models from *i** models in top-down scenarios. This variable can have two values (a.k.a. treatments [38]):
  - o **Users apply their own criteria**, serving the purpose of a control group.
  - o **Users apply the *iStar2ca* guidelines V1.0** as defined in [14].

We consider the following dependent variables (a.k.a. response variables [38]), which are expected to be influenced to some extent by the independent variable. We have adapted the Method Evaluation Model (MEM) [40] to structure the variables of this experiment. Effectiveness has been decomposed into CA model completeness and CA model validity according to the model quality framework by [41].

- **CA model completeness.** The degree to which all the elements that should appear in the CA model (because they represent relevant phenomena of the domain) are actually contained in the model. To facilitate this calculation, the researchers take into account a reference model containing the minimum indispensable elements.
- **CA model validity**. The degree to which all the elements contained in the CA model should actually appear in the model in the right way (e.g., an element representing a phenomenon that does not occur in the domain is invalid). Acting as reviewers, the researchers identified candidate invalid elements based on a reference model and then discussed them until they agreed on the verdict.
- **Subject efficiency.** The degree of success during the application of a derivation strategy of CA models according to the time consumed (the CA model completeness divided by the time consumed).
- **Perceived usefulness** (PU). The degree to which the subject considers that a CA derivation strategy is effective in achieving its intended objectives. This and the next two variables are measured by means of the Method Evaluation Model (MEM) questionnaire [40], which uses a 5-point Likert scale format to obtain subject perceptions.
- **Perceived ease of use** (PEOU). The degree to which a subject considers that using a CA derivation strategy is free of effort.
- **Intention to use** (ITU). The willingness of the subject to use the CA derivation strategy in the future.

Hypotheses. In Table 2 we define null hypotheses (represented by a 0 in the subscript) that correspond to the absence of an impact of the independent variables on the dependent variables. Alternative hypotheses (represented by a 1 in the subscript; e.g., $H1_1$ is the alternative hypothesis to $H1_0$) suppose the existence of such an impact. Alternative hypotheses correspond to our expectations: the *iStar2ca* guidelines will have a positive impact on the dependent variables. For the sake of brevity, alternative hypotheses are omitted (but they can be found in [42]).

Table 2. Specification of hypotheses

Null hypothesis id.	Statement: The CA derivation strategy from i^* models does not influence…
$H1_0$	… the completeness of the resulting CA models
$H2_0$	… the validity of the resulting CA models according to incorrect elements
$H3_0$	… the efficiency of the subjects
$H4_0$	… the perceived usefulness
$H5_0$	… the perceived ease of use
$H6_0$	… the perceived intention to use

8.2 Procedure

Properly speaking, we have performed a quasi-experiment because the subjects were not sampled randomly across the population [43]; however, this is typical in software engineering experiments. As part of the experimental task, we provided to the subjects with two different input i^* cases (A and B) to increase the external validity of the experiment. We created two equivalent versions (A1, A2, B1, and B2) for each case, changing the application domain but maintaining the overall structure of the underlying problem so that the resulting models are isomorphic. Exhaustive analysis of the input i^* models is out of scope of this paper; nevertheless we acknowledged as part of the experiment set up for further analysis and research. The design, according to [36] is a "blocked subject-object study" (a.k.a. "paired design blocked by experimental objects" [38]) and it is shown in Table 3.

Table 3. Experimental design

|		Input i^* model (i.e. object)	
	CA model derivation strategy	Group 1	Group 2
Session 1	Users apply their own criteria (C)	Case A1	Case B1
Session 2	Users apply the iStar2ca guidelines (G)	Case B2	Case A2
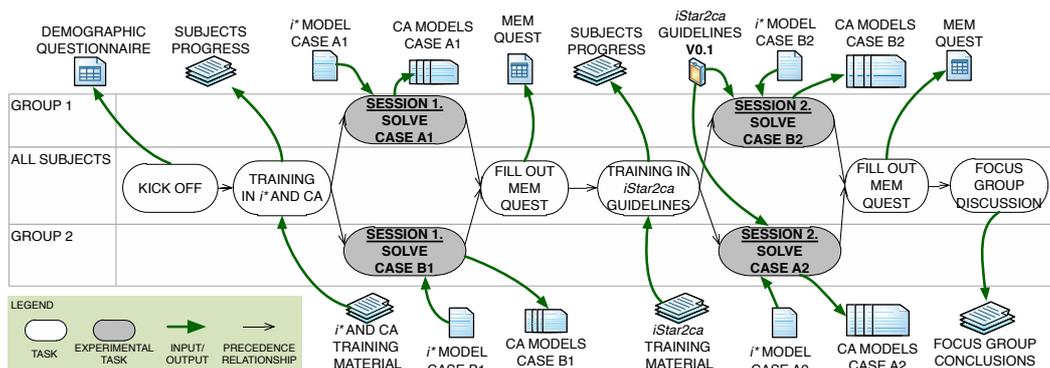
Figure 9.   Experimental procedure

Figure 9 presents an overview of the experimental procedure. During the training sessions, the subjects solved some exercises and they received feedback on their performance. To randomly allocate subjects into two groups, we used http://www.random.org/. We obtained two groups of 10 and 9 subjects, respectively; the statistical techniques we use are robust to this imbalance. Finally, we gathered additional qualitative information in the focus group session.

According to this design, we take two measurements for each subject, one before and one after the introduction of the *iStar2ca* guidelines. As a consequence, we apply the Paired Sample T-Test to statistically analyse the collected data. For the data that does not fulfil the T-Test assumptions, we apply the Wilcoxon signed-ranked non-parametric test. A summary of the results is presented below (further details can be found in **[42]**).

CA model completeness with respect to a reference model

The results of descriptive statistics show an average completeness of 58% when the subjects apply their own criteria versus 79% when they apply the *iStar2ca* guidelines. A Paired Sample T-Test was applied to verify the null hypotheses $H1_0$. A **significant** difference ($p<0,05$) between the applications of the subjects' criteria against the *iStar2ca* guidelines is demonstrated. Thus, the null hypothesis $H1_0$ is rejected and the alternative hypothesis $H1_1$ is corroborated, demonstrating that the application of *iStar2ca* guidelines as CA derivation strategy yields greater completeness in resulting CA models. This analysis contributes to answering RQ1.

CA model validity with respect to a reference model

The results of descriptive statistics show an average of 4% of validity errors per model when the subjects applied their subjective criteria versus 3,4% when they applied the *iStar2ca* guidelines. By applying the Paired Sample T-Test, we observe there is **no significant difference** ($p=0,582$) between the application of the *iStar2ca* guidelines and the subjects' criteria. Therefore, $H2_1$ is not corroborated, and we conclude that the application of the derivation strategy does not influence the validity of the resulting CA models. In any case, the descriptive statistics indicate favourable results for the *iStar2ca* guidelines. This analysis contributes to answering RQ1.

Subject efficiency

To measure the subjects' efficiency, we recorded the time (in minutes) that each subject spent during the experimental task. The results of the descriptive statistics show that applying the *iStar2ca* guidelines took longer than applying subjective criteria. A possible explanation for these results is related to the forms that the subjects had to fill out in to indicate the reasoning steps while applying the guidelines. During the focus group session, the subjects expressed that they spent a lot of time filling out these forms and, conversely, the application of the guidelines was agile and easy for them. Nevertheless, there are no significant results to corroborate the alternative hypothesis according to the results of the Paired Sample T-Test ($p>0,05$). As a result, $H3_1$ was not corroborated, and we conclude that the CA derivation strategy does not influence the subjects' efficiency. This analysis contributes to answering RQ2.

Subject perceptions

Subject perceptions were collected by means of the MEM questionnaire [40], containing 5-point Likert scale questions (the questionnaires can be found in [44]). Each variable has several questions, whose answers are aggregated as an average value. For **perceived usefulness**, the averages obtained from the descriptive statistics indicate that the PU of the *iStar2ca* guidelines (a value of 3,8) is higher than the PU of their own subjective criteria (3,5). For **perceived ease of use**, we found that the subjects found the iStar2ca guidelines easier to apply (3,7) than their own criteria (3,5). For **intention to use**, the results

were positive for the application of the *iStar2ca* guidelines (3,6 versus 3,3). However, the results of the Paired Sample T-Test indicate that such differences are not significant (p>0,05). Thus, <u>H4$_1$, H5$_1$ and H6$_1$ are not corroborated, and we conclude that there is no influence of the subjects' perceptions on the CA derivation guidelines</u>. This analysis answers RQ3.

Focus group session

To obtain additional qualitative information about subject perceptions, we conducted a focus group. The number of participants was 17 because two students missed this class for reasons unrelated to the experiment. The focus group lasted two hours. In short, the participants were asked to fill post-its with positive or negative aspects of the guidelines and for suggesting improvements and discussion about the content of each post-it was encouraged. For the sake of brevity, we do not present here all the results obtained from this session and we just briefly describe two of them that have had a direct influence on the *istar2ca* guideline revision. First, a good number of participants suggested to develop additional guidelines for deciding the merging or ordering of events and for deciding event variants. It is worth to mention that after analysing these cases we identified an additional guideline (guideline 8 incorporated to version 2.0) for merging events and we concluded that it was not possible to identify guidelines for more cases of event merging, event ordering or for deciding event variants because they would require input knowledge not deducible from *i** diagrams. Second, regarding the guideline description we got contradictory suggestions: some participants advocated for more details while others proposed to lessen them. As a consequence, we evaluate positively to maintain the abstract guideline in the description because it can serve as a short version of guidelines 2 to 5 and we consider an interesting line of further work to develop different versions of the guideline description with different levels of detail.

8.3   Analysis of the threats to the validity of the results

Internal validity. The subjects matured their competence in *i** and CA after their first experimental task, what can have had a positive impact in performance. We considered other experimental designs that minimised this threat (e.g. completely randomized design, a staggered training) but discarded them due to the risk of resentful demoralisation (students not applying the guidelines may feel that they are treated unfairly) and constraints of the course. To minimise the threats of maturation and testing (i.e. the subjects learning what we are intending to measure can have an impact on their attitude), we did not provide feedback about the evaluation of the models resulting from the first task until the second task was over.

External validity. It is common practice in software engineering experiments to use students as surrogates for real practitioners, since involving real practitioners is costly. However, we acknowledge that this limits the generalisation of the experiment results. The input *i** model may also have an impact on the performance, but using too many distinct ones can have a confounding effect. Thus, we used two distinct cases, what seems to be a reasonable number, given the number of subjects.

Construct validity. We minimised the threat of inadequate preoperational explication of constructs by means of using the MEM questionnaire (which is proved valid for measuring perceptions) and a widely accepted framework for conceptual model quality evaluation. On other matters, the fact that we compare the guidelines with subjective criteria may seem an inadequate comparison of treatments. However, we are indeed comparing a new approach with the current practice of CA practitioners.

9 Discussion

In this section, we present a discussion that is based on the results of the experimental task and the focus group session. Also, we discuss additional observations that are related to the complete framework presented in the paper that are worth discussing.

Comparative experiment

The results of the comparative experiment and the focus group session have demonstrated how the *iStar2ca* guidelines V1.0 improve the task to obtain business process models from goal models. We acknowledge that there are threats to the validity of the results. The main ones are related to the use of students instead of real practitioners (which limits the generalisation of the results) and the maturation that occurred in the subjects when they performed a similar task for the second time. The results of the experiment and the focus group triggered the following important changes which were applied to the *iStar2ca* guidelines V1.0: the creation of Guideline 8 that was motivated by the results of the focus group session (with the purpose of identifying situations where the merging of events is needed) and Guideline 9; which was motivated regarding the model evaluation that we performed over the resulting CA models after the guidelines application (with the purpose of naming the obtained CA elements from *i** elements in order to make the traceability between models explicit). Empirical results for creating the Guidelines 8 and 9 led to the creation of the *iStar2ca* guidelines V2.0 (see Guidelines 8 and 9 in the section 6).

Scenarios

The integration of the goal and business process perspectives can be used in different modelling scenarios: top-down, bottom-up, iterative and evolution scenarios (see Section 5). It is worth to remark that the proposed FRISCO-based ontological mapping makes possible the formulation of modelling guidelines for the mentioned scenarios. We have contributed to the top-down scenario with the formulation of guidelines that help to apply it. Guidelines for the other possible scenarios are considered as our future lines of work. We have chosen to provide guidelines for top-down scenarios as an artefact, which together with the bottom-up and evolution guidelines (to develop in the future) can conform a base for reengineering IS projects.

Integrated usage of *i** and CA models

Something that should not be surprising is that the top-down scenario guidelines cannot map all *i** elements into CA elements and, conversely, not all the elements of a CA model can be obtained from an *i** model. The reason for this is intrinsic to the methods since *i** and CA offer different perspectives of the IS with different focuses. *i** offers a goal-oriented perspective while CA gives a business-communication perspective. These two different perspectives of a single IS support a separation of concerns which is very useful in facilitating the IS analysis. The GoBIS framework facilitates the consistent use of both views. In other words, (1) it offers an integrated metamodel for both views, and (2) the CA elements which are derivable from *i** (and thus should be compliant to *i** elements) can be derived by following the *iStar2CA* guidelines. The use of the guidelines ensures the consistency of the complementary views. The rest of the knowledge needed to complete the business-communication perspective of the IS (e.g., the ordering between some of the events, the internal structure of message structures) that cannot possibly be obtained from an *i** model should be elicited as complementary requirements of the IS. A future work related to the integrated usage of *i** and CA is mentioned in Section 10.

10 Conclusions and future lines of work

Given the existence of complementary perspectives in information system analysis, this work confronts the challenge of integrating a goal-based and a business process-based modelling language. A review of related works reveals that there is room for improvement. As a result, in this paper, we have reported a design science endeavour that undertakes the integration of *i** and Communication Analysis (CA). We have selected these languages for their expressiveness and their associated elicitation and specification techniques. To provide a sound theoretical foundation for the integration, we have first performed the ontological analysis of both languages and analysed their overlapping concepts (which are semantically equivalent). Beyond supporting conceptual reasoning, the analysis also facilitated making concrete decisions regarding metamodel integration; e.g., we selected one metaclass from a single language (removing the other) or we kept both metaclasses depending on whether their associated concepts were totally equivalent or their alignment assumed a specific condition. In this paper, among the several scenarios in which the integration is potentially valuable, we provide guidelines for top-down scenarios. We have also developed an Eclipse-based tool to support the integrated modelling. Moreover, we have conducted a comparative experiment with master students to evaluate the usage of the top-down modelling scenario guidelines. The results indicate that applying the proposed guidelines yields a significant improvement over applying subjective, common-sense criteria, and that, even though it requires systematic reasoning, it does not have a significantly impact on efficiency. We have also improved the guidelines based on the experiment feedback.

As future work, we plan to replicate the comparative experiment and, if necessary, further improve the top-down scenario guidelines. We will design an additional set of guidelines to help designers elicit missing requirements from stakeholders in order complete the CA model. We also plan to carry out a technical action research effort to apply our approach under practical conditions of practice and foster industrial transference. With regard to the modelling tool, we plan to develop a wizard that helps apply the guidelines in a semi-automatic way. Finally, we would like to complement the GoBIS framework with guidelines for the rest of the envisaged scenarios (i.e., bottom-up, iterative, and evolution) and to create a complete method specification indicating the participant roles (e.g. requirements engineer, business analysts, representative user) and their collaboration flow.

Acknowledgements

References

1. Hammer, M. and J. Champy, *Reengineering the corporation: a manifesto for business revolution (revised)*. 2003, New York: HarperCollins.
2. Henderson, J.C. and N. Venkatraman, *Strategic alignment: leveraging information technology for transforming organizations.* IBM Syst. J., 1999. **38**(2-3): p. 472-484.
3. Yu, E., *Modelling Strategic Relationships for Process Reengineering*, University of Toronto, 1995.

4. Zikra, I., J. Stirna, and J. Zdravkovic, *Bringing enterprise modeling closer to model-driven development*, in *The Practice of Enterprise Modeling*, 2011, Springer LNBIP. p. 268-282.

5. Yu, E. and J. Mylopoulos, *From E-R to "A-R" - Modelling strategic actor relationships for business process reengineering*, *13th International Conference on the Entity-Relationship Approach*. 1994, Springer-Verlag: Manchester. p. 548-565.

6. Guizzardi, R.S.S., et al., *Bridging the Gap between Goals, Agents and Business Processes*, in *Fourth International i* Workshop*. 2010: Hammamet, Tunisia.

7. Cardoso, E.C.S., R.S.S. Guizzardi, and J.P.A. Almeida, *Aligning goal analysis and business process modelling: a case study in health care.* International Journal of Business Process Integration and Management, 2011. **5**(2): p. 144-158.

8. Kueng, P. and P. Kawalek, *Goal-based business process models: creation and evaluation.* Business Process Management Journal, 1997. **3**(1): p. 17-38.

9. Koliadis, G. and A. Ghose, *Relating business process models to goal-oriented requirements models in KAOS*, in *Advances in Knowledge Acquisition and Management*, 2006, Springer: Berlin. p. 25-39.

10. Morrison , E.D., et al., *Strategic alignment of business processes*, in *7th International Workshop on Engineering Service-Oriented Applications*. 2011: Paphos, Cyprus.

11. Kavakli, V. and P. Loucopoulos, *Goal-driven business process analysis application in electricity deregulation.* Information Systems, 1999. **24**(3): p. 187-207.

12. Wieringa, R., *Design science as nested problem solving*, in *4th International Conference on Design Science Research in Information Systems and Technology*. 2009, Philadelphia, Pennsylvania. p. 1-12.

13. España, S., A. González, and Ó. Pastor, *Communication Analysis: a requirements engineering method for information systems*, *(CAiSE'09)*, Springer LNCS 5565: Amsterdam, The Netherlands. p. 530-545.

14. Ruiz, M., et al., *Integrating the Goal and Business Process Perspectives in Information System Analysis*, *(CAiSE 2014)*. 2014, Springer: Thessaloniki, Greece. p. 332-346.

15. Ralyté, J. and C. Rolland, *An assembly process model for method engineering*, *(CAiSE 2001)*, A.G.a.M.C.N. K. R. Dittrich, Editor. 2001: Interlaken, Switzerland.

16. Salay, R., J. Mylopoulos, and S. Easterbrook, *Using Macromodels to Manage Collections of Related Models*, *(CAiSE)*. 2009, LNCS 5565, Springer: Amsterdam, The Netherlands.

17. Andersson, B., et al., *Towards a formal definition of goal-oriented business process patterns.* Business Process Management Journal, 2005. **11**(6): p. 650-662.

18. Cardoso, E., et al., *A Method for Eliciting Goals for Business Process Models based on Non-Functional Requirements Catalogues.* (IJISMD), 2011. **2**(2).

19. Soffer, P. and Y. Wand, *On the notion of soft-goals in business process modeling.* Business Process Management Journal, 2005. **11**(6): p. 663-679.

20. Leonardi, M.C. and R. Giandini, *Una estrategia de integración de Modelos de Objetivos con Análisis Comunicacional* in *16th Workshop on Requirements Engineering*. 2013: Montevideo, Uruguay.

21. Lapouchnian, A., Y. Yu, and J. Mylopoulos, *Requirements-driven design and configuration management of business processes*, *BPM'07*, Springer, Editor. 2007. p. 246-261.

22. Ghose, A., et al., *Goal-Driven Business Process Derivation*, *ICSOC 2011*. 2011, Springer: Paphos, Cyprus. p. 467-476.

23. Kazhamiakin, R., M. Pistore, and M. Roveri, *A framework for integrating business processes and business requirements*, *EDOC'04*. 2004, IEEE. p. 9-20.

24. Ulrich, F., *Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges.* Software and Systems Modeling, 2012. **13**(3): p. 941-962.

25. González, A., et al., *Message Structures a modelling technique for information systems analysis and design*, in *XIV Workshop on Requirements Engineering (WER'11)*, Rio de Janeiro - Brasil. p. 407-418.

26. Gonzalez, A., S. España, and O. Pastor, *Unity criteria for business process modelling: A theoretical argumentation for a software engineering recurrent problem*, *(RCIS 2009*, Fes, Morocco. p. 173-182.

27. Falkenberg, E.D., W. Hesse, et al., *A framework of information system concepts*. IFIP WG 8.1 TG, 1998.

28. España, S., *Methodological integration of Communication Analysis into a model-driven software development framework*, PhD thesis in Universitat Politècnica de València, Spain, 2011.

29. Costal, D., et al., *Integration of i* and Communication Analysis*. 2013, GESSI - (UPC) & PROS - (UPV). Available from: http://hci.dsic.upv.es/ca/CAiSE2014_TR.pdf

30. Lopez, L., X. Franch, and J. Marco, *Making Explicit Some Implicit i* Language Decisions*, in *30th International Conference on Conceptual Modeling (ER 2011)*. 2011: Brussels, Belgium. p. 62-77.

31. OMG, *MDA Guide*, in *How is MDA used?* 2003, OMG. p. 1-62.

32. Ruiz, M., *A model-driven framework to integrate Communication Analysis and OO-Method*, MSc in Universitat Politècnica de València, Spain, 2011.

33. OpenOme. *OpenOME, an open-source requirements engineering tool*. Available from: https://se.cs.toronto.edu/trac/ome/.

34. Giachetti, G., *Supporting Automatic Interoperability in Model Driven Development Processes*, PhD thesis in Universitat Politècnica de València, Spain, 2011.

35. Santos, E., et al., *Towards a unified metamodel for i**, in *Second International Conference on Research Challenges in Information Science (RCIS 2008)*. 2008.

36. Wohlin, C., et al., *Experimentation in Software Engineering*. 2004: Springer.

37. Jedlitschka, A. and D. Pfahl, *Reporting guidelines for controlled experiments in software engineering*, in *4th International Symposium on Empirical Software Engineering*. 2005, IEEE: Queensland, Australia

38. Juristo, N. and A. Moreno, *Basics of Software Engineering Experimentation*. 2001: Springer.

39. Basili, V.R., G. Caldiera, and H.D. Rombach, *Goal Question Metric Paradigm*, in *Encyclopedia of Software Engineering* 1994.

40. Moody, D.L., *The Method Evaluation Model: A Theoretical Model for Validating Information Systems Design Methods*, *ECIS 2003*. 2003: Naples, Italy.

41. Lindland, O.I., G. Sindre, and A. Solvberg, *Understanding quality in conceptual modeling.* IEEE Software, 1994. **11**(2): p. 42-49.

42. Ruiz, M., et al., *Appendix: Gobis - An integrated framework to analyse the Goal and Business process perspectives in Information Systems*, 2014. http://hci.dsic.upv.es/istar2ca_exp/Appendix_GoBIS.pdf

43. Robson, C., *Real World Research: A Resource for Social Scientists and Practitioners-Researchers*. 2 ed. 2002: Blackwell, Oxford/Madden.

44. Ruiz, M., et al. *iStar2ca guidelines - Instruments and material for the comparative experiment ISE 2014*. 2014; Available from: http://hci.dsic.upv.es/istar2ca_exp/.