

Integrating the Goal and Business Process Perspectives in Information System Analysis

Marcela Ruiz¹, Dolors Costal², Sergio España¹, Xavier Franch², Óscar Pastor¹

¹PROS Research Centre, Universitat Politècnica de València, Valencia, Spain

{lruiz, sergio.espana, opastor}@pros.upv.es

²GESSI Research Center, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

{dolors, franch}@essi.upc.edu

Abstract. There are several motivations to promote investment and scientific effort in the integration of intentional and operational perspectives: organisational reengineering, continuous improvement of business processes, alignment among complementary analysis perspectives, information traceability, etc. In this paper we propose the integration of two modelling languages that support the creation of goal and business process models: the *i** goal-oriented modelling method and Communication Analysis, a communication-oriented business process modelling method. We describe the methodological integration of the two modelling 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; and iv) to develop tool support. We provide guidelines for using the two modelling methods in a top-down analysis scenario. We also present an illustrative case that demonstrates the feasibility of the approach.

Keywords: modelling language, requirements engineering, goal modelling, business process modelling, ontological analysis, metamodel integration

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 become common practices in information system engineering. Understanding organisations and their needs for change often requires several interrelated perspectives [2-3]. The information system 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 being widely accepted that processes are 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.

adfa, p. 1, 2011.

Ruiz, M. [et al.]. Integrating the goal and business process perspectives in information system analysis. A: International Conference on Advanced Information Systems Engineering. "Advanced information systems engineering: 26th International conference, CAiSE 2014: Thessaloniki, Greece, June 16-20, 2014: proceedings". Berlín: Springer, 2014, p. 332-346.

The final authenticated version is available online at https://doi.org/10.1007/978-3-319-07881-6_23

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 the motivation for processes [6]. (ii) In the opposite direction, 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) It helps identifying cross-functional interdependencies during business change management, by supporting the identification of the goals for change and the analysis of the impact on processes [11] [8] [10].

We pursue 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. Remarkably, we consider 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. These criteria guide 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 steps from the problem investigation to the implementation of a modelling tool. We have chosen to integrate the languages proposed by i^* [3], a goal-oriented modelling method, and Communication Analysis (CA) [12], a communication-oriented business process modelling method. The reason to choose 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; it is not its notation what is important, but the underlying concepts and guidelines. Moreover, some current business process modelling suites use BPMN with a communicative approach. In addition, the authors have competence in these languages as to target the endeavour.

As a result, in this paper we present the following contributions:

- We report on the alignment between i^* and CA performed by means of ontological analyses and the investigation of overlapping concepts and semantic relations.
- We integrate the metamodels of both modelling languages, providing rationale for the design decisions, and we provide guidelines for a top-down modelling scenario.
- We describe an Eclipse-based tool that supports integrated modelling of i^* and CA.

We structure our research in terms of design science since it involves creating new artefacts and acquiring new knowledge. Our research methodology follows the engineering cycle as described by [13] (see Fig. 1). The second step corresponds to a method engineering effort; throughout the paper, we use the terminology in [14]. The resulting integrated method is exemplified by means of a running example that demonstrates the feasibility of the approach.

The paper is structured as follows. Section 2 defines the solution criteria and compares related works. Section 3 presents a running example and introduces the methods selected for integration. Section 4 presents our proposal for integrating i^* and CA. Section 5 presents guidelines for a top-down modelling scenario. Section 6 describes the modelling tool. Finally, Section 7 concludes with a discussion and future work.

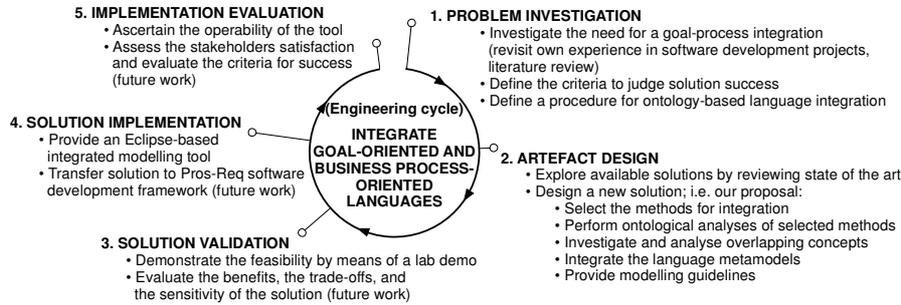


Fig. 1. Overview of the research methodology

2 Related work

In the field of business process management, there are several related works that approach goal-oriented business process reengineering from diverse angles. We analyse these approaches based on the criteria mentioned in the introduction (see Table 1): (a) ontological foundation: none, just a conceptual framework, based on ontologies but no explicit alignment, ontology-based alignment; (b) metamodel integration: none, partial, complete; (c) modelling scenarios: top-down (from goal model to business process model), bottom-up (the other way round), iterative (switching back and forth among both models), evolution (business processes evolve as driven by goal models); (d) existence or not of tool support.

Some works focus on modelling the as-is system (reverse engineering). For instance, [15] proposes 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. [7] discusses the alignment of goal and process modelling methods (using Tropos and ARIS, respectively) and proposes a three-stage method to model the as-is system.

Other works focus on supporting the business process model evolution. [16] proposes 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. [17] presents a formal approach to analysing the dependency of softgoals on processes; as a practical result, they enable modelling the evolution rationale.

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

Refs.	Ontological Foundation	Metamodel Integration	Modelling Scenario	Tool Support
[16]	Conceptual framework	N/A	Evolution	No
[15]	None	No	Top-down	No
[6]	Based on ontologies	No	N/A	No
[7]	Conceptual framework	No	Top-down	No
[11]	Conceptual framework	Complete	Iterative	No
[9]	None	No	Top-down	No
[8]	Conceptual framework	No	Top-down	No
[10]	Conceptual framework	No	N/A	No
[17]	Based on ontologies	No	Evolution	No
Ours	Ontologically aligned	Complete	Currently top-down Potentially may support the rest	Yes

Other works focus on modelling the to-be system (forward engineering). [8] presents an informal, seminal approach in which goals provide a basis for process definition. [11] defines a method that takes as input an as-is business process model and produces a to-be goal model and a to-be business process model.

Some of the above-mentioned works elaborate a conceptual framework to clarify definitions [7-8, 11, 16], and [17] 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 regards to modelling language integration, [11] relies on EKD metamodels (both perspectives are integrated *a priori*) but, noticeably, none of the works report a proper, rigorous metamodel integration ([7] mentions it as future work, though).

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 [6, 9-10], what can be used as input for our guidelines definition. Similarly, the pattern-based approach in [16] could be adapted to the context of i^* and CA.

3 Running example

In the rest of the paper we will use as running example the SuperStationery Co. case, a company that provides office material to its clients. The company acts as an intermediary: catalogued products are bought from suppliers and sold to clients. In this paper we focus on the intentional and operational aspects of sales management (acronym SALE). A relevant excerpt of the i^* model for the SuperStationery Co. case is shown in Fig. 2. We assume in the paper that the reader is familiar with i^* .

In this paper we will provide guidelines to support the transformation of such an i^* model into a CA model (i.e. a top-down scenario). CA is a requirements engineering method that analyses the communicative interactions between the information system (IS) and its environment [12]. 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 description of new meaningful information that is conveyed to the information system in the event [18]. The CED (see Fig. 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. It is intended to account 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 norms and guidelines referred as unity criteria (which act as modularity guidelines) [19]. In addition, CED consist of actors that trigger the CE and provide the input information (primary), actors who need to be informed of the occurrence of an event (receiver) and relationships to specify communicative interactions (ingoing/outgoing) and precedence relationships among CE.

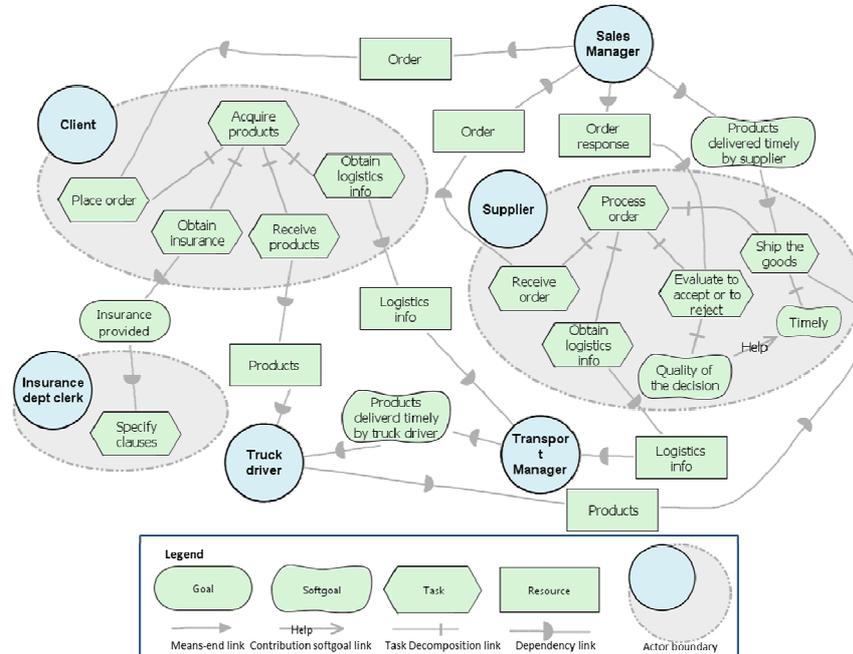


Fig. 2. Excerpt of an i^* model for the SuperStationery Co. case

4 Modelling language integration

In order to combine the process and intentional perspectives, we undertake a method engineering effort [14], 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 is put in integrating the product and the process models of the methods. Taking the integration map proposed in [14] as reference method, Fig. 4 presents how we have operationalised each of its intentions, and points to the corresponding section.

4.1 Ontological alignment between i^* and CA

Integrating the product models of two methods requires identifying pairs of concepts that have the same semantics, so as to later merge them. 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 [14]. 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 concepts of a method are mapped to the concepts of reference ontology. This is commonly used to assess to which extent the method covers the concepts of the ontology, and vice versa. Among other different possible options (e.g., BWW, Chisholm's, DOLCE, UFO, etc.), we have chosen FRISCO as reference on-

tology because we have already satisfactorily used it in previous analysis related to one of the two methods, CA [20].

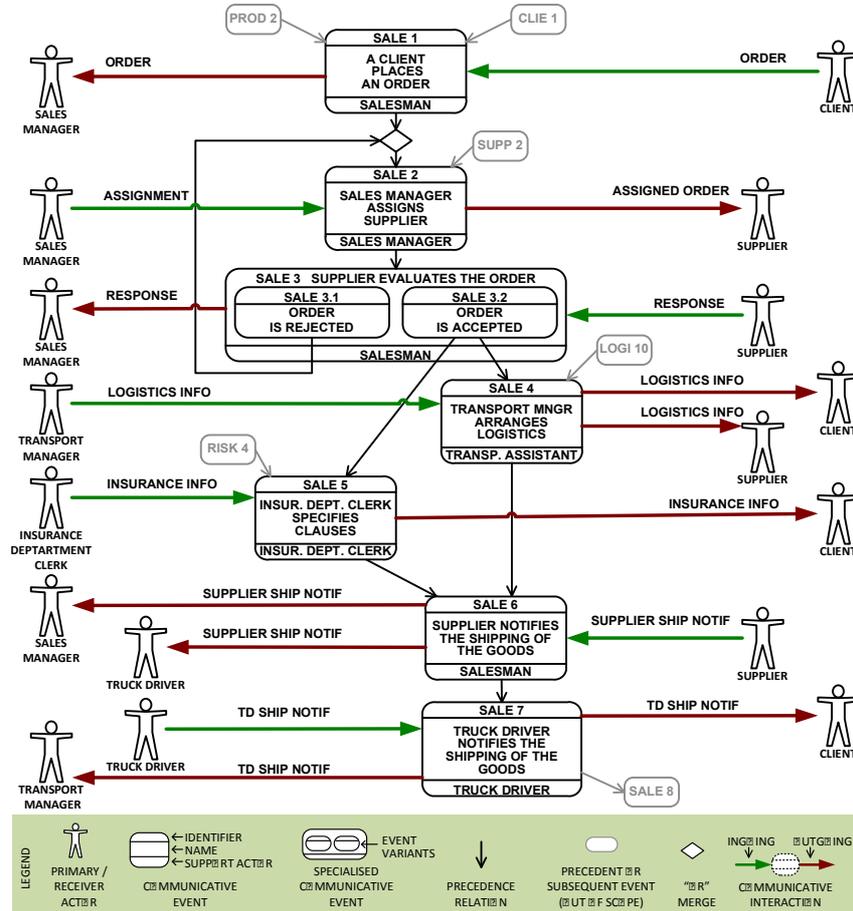


Fig. 3. Excerpt of a CA model for the SuperStationery Co. case

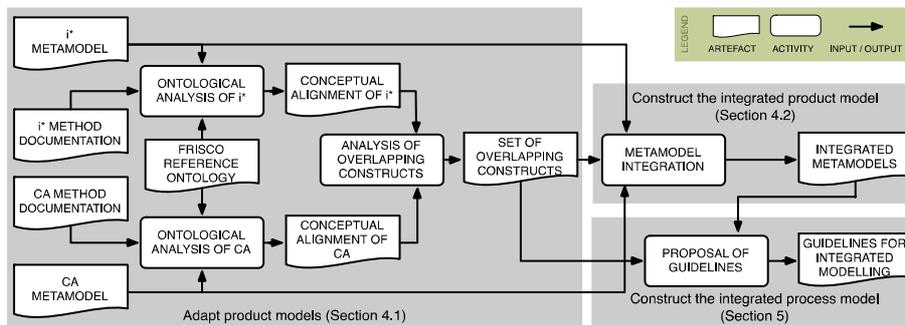


Fig. 4. Flow of modelling language integration

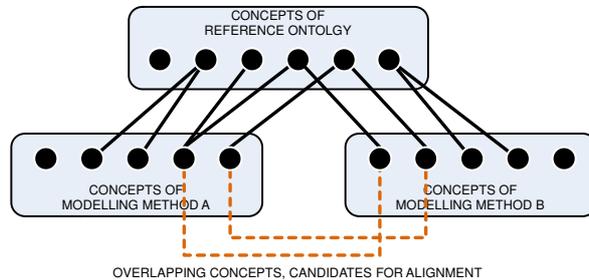


Fig. 5. Ontological analysis as a means to align concepts and guide method integration

The procedure is illustrated in Fig. 5. We have performed a separate ontological analysis of each method, establishing a complete mapping between: first, the concepts of *i** and the concepts of the FRISCO ontology and, second, the concepts of CA and those of the FRISCO ontology¹. This complete mappings are reported in [21]. Then, we are not interested in the criteria commonly applied in ontological analyses (e.g. construct excess, laconicism). Instead, we identify which pairs of concepts from each method are mapped onto the same ontological concept (this is an alternative way of verifying concept similarity [14]). These concepts are considered overlapping and, therefore, they are candidates for the integration. Finally, we analyse the overlap and decide whether to align the concepts unconditionally (i.e. in all cases) or under certain conditions.

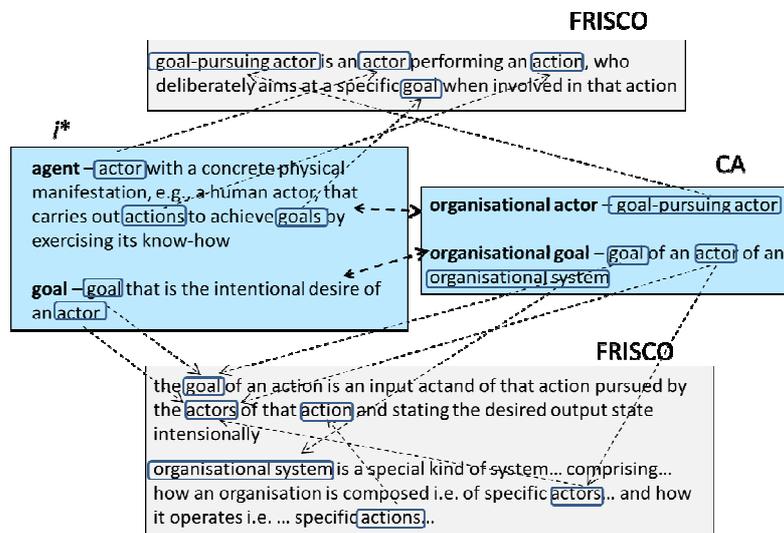


Fig. 6. Specific mappings for two pairs of overlapping concepts

¹ 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.

For the sake of illustration, Fig. 6 shows two examples of the type of reasoning process that has been applied during the ontological alignment. For instance, the method concepts *istar.agent* and *ca.organisational actor* map onto related FRISCO concepts (*frisco.goal-pursuing actor* is a specialisation of *frisco.actor*). Additionally, *istar.agent* is qualified as having physical manifestation and know-how; we interpret this additional qualification as compatible and also applicable to *ca.organisational actor*. Therefore, we consider the two method concepts as overlapping. Similarly, the method concepts *istar.goal* and *ca.organisational goal* map onto the same FRISCO concept (i.e. *frisco.goal*). Therefore, *istar.goal* and *ca.organisational goal* are overlapping concepts.

Table 2 summarises the overlapping concepts found in this analysis. Each row describes a pair of concepts that overlap. FRISCO mappings consist of FRISCO concepts (underlined) that are qualified when necessary. Table 2 also indicates the ontological alignments decided in view of the 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 adequate level of modularity (see [19] for details). Therefore, an *istar.task* maps to a *ca.communicative event* only if it satisfies the unity criteria. An *istar.resource* maps to a *ca.message structure* only if it is informational (e.g. a delivery note). Hence, *istar.physical resource* does not map to *ca.message structure* (e.g. a pallet of boxes is not a message structure).

Table 2. Mappings of candidate overlapping concepts

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

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

4.2 Metamodel integration

To integrate i^* and CA metamodels, we analysed the alignment of concepts presented in Section 4.1. For each pair of aligned concepts we need to decide whether we keep both corresponding metaclasses (one of each modelling method) or just one metaclass. We provide some heuristics to make such decision and the implications of each choice (see Fig. 7.a for the starting point).

In some cases, the two concepts are totally equivalent, in the sense that their alignment is clear-cut (concepts in the first three rows in Table 2 fall into this category). In such 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 need to be connected to the metaclass that is kept (see Fig. 7.b). In other cases, the alignment of two concepts is qualified with a condition specifying under which circumstances both concepts can be considered aligned (concepts in the last two rows of Table 2). Then, we propose to keep both metaclasses and create a relationship between them (see Fig. 7.c) to provide traceability in cases where the specific concepts are aligned. The application of these heuristics to our case is summarised in Table 3.

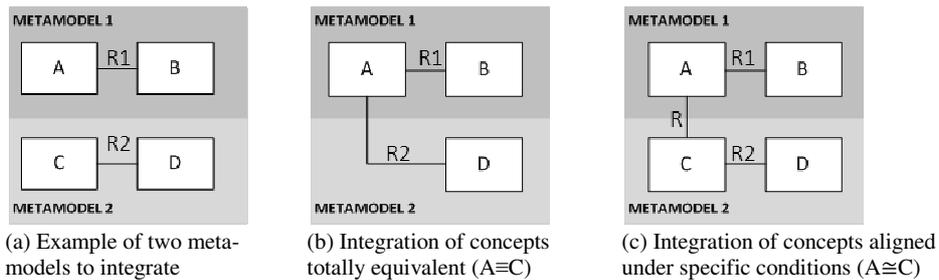


Fig. 7. Deciding which metaclasses to keep

To create the integrated metamodel, we have departed from the i^* metamodel which is basically compliant to the i^* wiki version taken as reference in this work [22] and the CA metamodel [20]. Full description of the integrated metamodel is presented in [21].

Table 3. Metamodel integration

i^* meta-class	CA metaclass	Metaclasses kept	Rationale
AGENT	ORGANISATIONAL ACTOR	AGENT	- Equivalent concepts - i^* provides more detailed definition
ROLE	ORGANISATIONAL ROLE	ROLE	
GOAL	GOAL	GOAL (from i^*)	- Equivalent under specific conditions
TASK	COMMUNICATIVE EVENT	Both	
RESOURCE	MESSAGE STRUCTURE	Both	

5 Top-down scenario guidelines and illustration

We present our guidelines for a top-down scenario whose main purpose is to guide the mapping from *i** elements into CA elements. Other possible scenarios (e.g., bottom-up, iterative, evolutionary, etc.) are not considered in the paper, but it is worth to remark that the proposed FRISCO-based ontological mapping make possible the formulation of similar guidelines for these cases.

The following guidelines help to implement the mappings that we have identified at a metamodel level when departing from a specific *i** model to obtain a CA model. 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 proposed guidelines provide advice not only 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 which 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 Fig. 2).

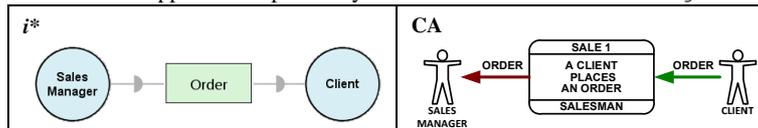
CA focuses on communicational interactions. Therefore, most guidelines involve *i** dependencies, because satisfying a dependency will require some type of interaction in general. Each type of dependum has an associated guideline except resource dependums in which informational and physical resources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

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

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

In our SuperStationery Co. case, 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 that allows the Client 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 an information from the *dependee* and, therefore, a *communicative event* is needed to allow the *dependee* communicate that information to the *depender*.

The *ca.interface actor*, however, cannot be determined from the *i** model. It may coincide with the *primary actor* or not. 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* visualized into 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 Fig. 2 for space reasons) also maps into it.

A dependency may be connected to SR elements which are not tasks 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) to map to the communicative events SALE 2, SALE 3 and SALE 4 respectively (see Fig. 2 and Fig. 3).

The following group of guidelines deals with the rest of 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 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 induces a message structure M if it is required that the dependee gives information to the depender about the intentional satisfaction of such dependum. In that case, D induces a communicative event C such that: (1) C's primary actor is D's dependee actor, (2) C's receiver actor is the D's depender actor, (3) C's ingoing and outgoing interactions specify M, (4) if any SR elements of D's dependee and depender actors 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 intentional satisfaction 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 stands for this information. The communicative event Insur. Dept. clerk specifies clauses is obtained with communicative roles In-

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 intentional satisfaction of the abstract guideline refines into accomplishment of this task.

Guideline 4. When the dependum is a softgoal, the notion of intentional satisfaction 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 to supervise its timeliness. The message structure Supplier ship notif stands for this information. The communicative event Supplier notifies the shipping of the goods is obtained with 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 intentional satisfaction 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 actor 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.

Finally, we provide two additional guidelines. Guideline 6 describes the derivation of information to be registered in the information system from an *istar.actor* and guideline 7 deals with ordering events in the CA model.

Guideline 6. An actor about which relevant information has to be registered in the information system 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 Fig. 3 because it is part of another process, i.e. client registry.

In general, an *i** model does not provide information to deduce the ordering of the communicative events obtained from it, however if two chained dependencies with the same dependum appear in it then a precedence between the two mapped communicative events is implicitly induced. Guideline 7 stands for 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, since there are two chained dependencies with the dependum `Products`, the event `Supplier notifies the shipping of the goods` must precede the event `Truck driver notifies the shipping of the goods`.

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.

6 Realising goal and process integration: Prototype

A technological support for the integration of *i** and CA is necessary to carry out validations and future case studies. Although existing tools allow creating separate *i** and CA models, we intend to support the combined modelling.

We chose Eclipse (<http://www.eclipse.org>) as a 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) [23] approach in order to develop a tool for both methods. This way, the handbooks 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 Platform-Independent Model layer of MDA. According to these PIMm, we have specified the Platform-Specific Metamodels (PSMm) that are compliant with Eclipse. These PSMm corresponds to 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 compliant with GMF [24]. We adapted it based on the result of the metamodel integration (Section 4).

With respect to *i**, there are several metamodels available. We analysed the PIMm presented in [22] and we opted to maintain the most of its concepts, although it required some adaptations to account for the metamodel integration and make it GMF-compliant. To design the PSM metamodel for *i**, we have analysed three tool-oriented metamodels: the `OpenOme` metamodel [25], the metamodel presented by Giachetti [26], and the unified metamodel for *i** [27]. For further information and technical details about the prototype see [21].

7 Conclusions and further work

Given the existence of complementary perspectives in information system analysis, this work faces the challenge of integrating a goal-based and a business process-based modelling language. When attempting such task we had several criteria in mind: i) the languages should be described in a rigorous manner; ii) the integration should be theoretically underpinned; iii) the usage scenarios of the integrated language should be taken into consideration; and iv) tool support should be provided. We performed a review of related works, taking into account such criteria, to find out that there was indeed space for improvement. As a result, in this paper we undertake the integration of *i** and Communication Analysis (CA). We have selected these languages for their expressiveness and their associated elicitation and specification techniques.

Following a design-science methodology, we created three new artefacts: 1) the ontological alignment between *i** and CA, 2) the integrated metamodel, and 3) the guidelines for integrated modelling. For this purpose, we reported the ontological analysis of both languages. This analysis provided a sound theoretical foundation for integrating both modelling languages. Beyond supporting a 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 depending on whether their associated concepts were totally equivalent or their alignment assumed a specific condition. We then integrated their metamodels, providing rationale for the design decisions. Moreover, we identified usage scenarios. In this paper we provided guidelines for top-down scenarios.

We have also developed an Eclipse-based tool to support the integrated modelling. We have taken into account the MDA layers and distinguished a platform-independent metamodel and a platform-specific metamodel, which involves technological restrictions. This has facilitated clarifying the design rationale.

As a next step, we plan to provide guidelines for naming the CA elements and *i** elements in order facilitate the traceability among the elements of the integrated models. In addition, we plan to confront top-down scenario so as to validate it (trade-off and sensitivity), transfer the solution to some of our industrial partners, and assess stakeholders' satisfaction. Also, we plan to provide guidelines to bottom-up, iterative and evolution scenarios.

Acknowledgements

This work has been supported by the Spanish MICINN PROS-Req (TIN2010-19130-C02-01, TIN2010-19130-C02-02); the Generalitat Valenciana ORCA (PROMETEO/2009/015); the FPI-UPV pre-doctoral grant; the European Commission FP7 Project CaaS (611351); and the ERDF structural funds.

References

1. Hammer, M. and J. Champy, Reengineering the corporation: a manifesto for business revolution. 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, Department of Computer Science, University of Toronto, 1995
4. Zikra, I., J. Stirna, and J. Zdravkovic, Bringing enterprise modeling closer to model-driven development, *PoEM 2011*, Springer LNBI. p. 268-282.
5. 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. *IJBPM*, 2011. **5**(2): p. 144-158.
6. Yu, E. and J. Mylopoulos, From E-R to "A-R" - Modelling strategic actor relationships for business process reengineering, *ER 1994*, Springer-Verlag: Manchester. p. 548-565.
7. Guizzardi, R.S.S., G. Guizzardi, J.P.A. Almeida, and E. Cardoso, Bridging the Gap between Goals, Agents and Business Processes, *i** 2010: Hammamet, Tunisia.
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, *PKAW 2006*, Springer: Berlin. p. 25-39.
10. Morrison, E.D., et al, Strategic alignment of business processes, *WESOA 2011*.
11. Kavakli, V. and P. Loucopoulos, Goal-driven business process analysis application in electricity deregulation. *Information Systems*, 1999. **24**(3): p. 187-207.
12. España, S., A. González, and Ó. Pastor, Communication Analysis: a requirements engineering method for information systems. *CAiSE 2009*.
13. Wieringa, R., Design science as nested problem solving, *ACM DESRIST 2009*.
14. Ralyté, J. and C. Rolland, An assembly process model for method engineering. *CAiSE 2001*. Interlaken, Switzerland.
15. Andersson, B., et al, Towards a formal definition of goal-oriented business process patterns. *Business Process Management Journal*, 2005. **11**(6): p. 650-662.
16. Cardoso, E., et al, A Method for Eliciting Goals for Business Process Models based on Non-Functional Requirements Catalogues. *IJISMD 2*(2), 2011.
17. 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.
18. González, A., M. Ruiz, S. España, and Ó. Pastor, Message Structures a modelling technique for information systems analysis and design, *WER 2011*.
19. Gonzalez, A., S. España, and O. Pastor, Unity criteria for business process modelling: A theoretical argumentation for a software engineering recurrent problem., *RCIS 2009*.
20. España, S., Methodological integration of Communication Analysis into a model-driven software development framework, PhD thesis, UPV 2011.
21. Costal, D., et al, Integration of *i** and Communication Analysis.2013, *GESSI - UPC & PROS - UPV* http://hci.dsic.upv.es/ca/CAiSE2014_TR.pdf
22. Lopez, L., X. Franch, and J. Marco, Making Explicit Some Implicit *i** Language Decisions, *ER 2011*, Brussels, Belgium. p. 62-77.
23. OMG, MDA Guide, in *How is MDA used?* 2003, OMG. p. 1-62.
24. Ruiz, M., A model-driven framework to integrate Communication Analysis and OO-Method, Master thesis, Universitat Politècnica de València, 2011
25. OpenOME, an open-source requirements engineering tool. Available: <https://se.cs.toronto.edu/trac/ome/>.
26. Giachetti, G., Supporting Automatic Interoperability in Model Driven Development Processes, PhD thesis, Universitat Politècnica de València, 2011.
27. Santos, E., C. Silva, F. Alencar, M.J. Silva, and J. Castro, Towards a unified metamodel for *i**, *RCIS 2008*.