

Doctoral Thesis

In partial satisfaction of the requirements for the
degree of PhD in Software

From the *i** Diversity to a Common Interoperability Framework

Carlos Cares

Advisor:
PhD. Xavier Franch

Barcelona, 2012



Departament d'Enginyeria de Serveis i Sistemes
d'Informació (ESSI),
Universitat Politècnica de Catalunya
Barcelona Tech

TABLE OF CONTENTS

List of Figures.....	3
List of Tables.....	4
List of Acronyms.....	5
CHAPTER 1. Introduction.....	6
1.1 Disciplinary Context.....	7
1.1.1 Software Engineering and Requirements Engineering.....	7
1.1.2 Goal-oriented and Agent-oriented RE Paradigms.....	9
1.1.3 The <i>i*</i> Framework.....	12
1.1.3.1 <i>i*</i> and the RE paradigms.....	12
1.1.3.2 A Brief Presentation of <i>i*</i>	13
1.2 Objectives and Research Approach.....	16
1.2.1 Definition of the Problem.....	16
1.2.2 Relevance of the Problem.....	17
1.2.3 Objective and Research Questions.....	18
1.2.4 Research Method.....	19
1.2.3 Document's Overview.....	21
CHAPTER 2. Analysis of the <i>i*</i> Language from an Interoperability Perspective.....	23
2.1 Historical Evolution of <i>i*</i>	25
2.1.1 Antecedents.....	25
2.1.2 Chronology.....	26
2.2 The <i>i*</i> Language: Summary of Variations.....	29
2.2.1 Analysis of the Three <i>i*</i> Leading Variants.....	29
2.2.2 A Quantitative Description of Recent <i>i*</i> Variations.....	32
2.2.3 On the Combination of <i>i*</i> with Other Frameworks.....	34
2.3 The Metamodels Behind <i>i*</i> Variations.....	38
2.4 Describing the <i>i*</i> Tool Interoperability Scenario.....	41
2.5 Summary of the Chapter.....	44
CHAPTER 3. Theoretical Approach to an Interoperability Framework for <i>i*</i> Language Variants.....	45
3.1 Conceptual Framework.....	46
3.1.1 Interoperability.....	46
3.1.2 The Role of Ontologies in Semantics.....	48
3.1.3 A Formalization of the Interoperability Problem.....	53
3.2 Social Theoretical Framework.....	54
3.2.1 Contributions from Sociology of Science.....	54
3.2.2 Contribution from Cybernetics.....	57
3.2.3 Contributions from Semiotics.....	58
3.3 Approaching a Theoretical Framework for <i>i*</i> Model Interoperability.....	59
CHAPTER 4. iStarML: A Proposal for <i>i*</i> Model Interoperability.....	66
4.1 Presentation.....	67
4.2 Requirements for an <i>i*</i> Model Interchange Proposal.....	67
4.3 iStarML: from Foundations to Language Features.....	70

4.4	Representing the <i>i*</i> Language Variants in iStarML	77
4.5	Translating Models Using iStarML.....	80
4.6	Characterizing Semantic Preservation.....	81
4.7	Summary.....	84
CHAPTER 5. Applying the iStarML Interoperability Proposal		86
5.1	iStarML Interoperability Proof of Concept	87
5.1.1	Analysis of the Involved Tools.....	87
5.1.1	jUCMNav	87
5.1.2	OME 3.	88
5.1.3	HiME	88
5.1.4	First Case: from OME 3 Models (Yu's <i>i*</i>) to jUCMNav Models (GRL) ...	88
5.1.5	Second Case: Translation from jUCMNav to HiME	92
5.2	Interoperability Scenarios and Technological Support	94
5.2.1	Generating and Reading iStarML by <i>i*</i> Tools	95
5.2.2	Metric Calculation	96
5.2.3	Parsing and Handling iStarML Files	96
5.2.4	Validating <i>i*</i> Models by Using iStarML	97
5.3	iStarML: A Community's Perception	98
CHAPTER 6. Conclusions		102
6.1	Answers to Research Questions and Findings.....	104
6.2	Theoretical Conclusions	108
6.3	Technological Conclusions	109
6.4	Epistemological Conclusions	110
6.5	Future Work.....	111
6.6	Final Words	112
References		113
APPENDIX A. Publications.....		121
A.1	Studying and Understanding the <i>i*</i> Framework	122
A.2	Trying <i>i*</i> on Software Engineering Contexts	123
A.3	Theoretical Foundations for an Interoperability Proposal.....	125
A.4	Proposing iStarML	125
A.5	iStarML Support and Usages.....	126
A.6	<i>i*</i> Interoperability Related Work.....	126

List of Figures

Fig. 1.1.	Requirements Engineering phases according to Sommerville [SOMM'05]	8
Fig. 1.2.	Kavakli's comparison of KAOS and <i>i*</i> life cycles.....	12
Fig. 1.3.	<i>i*</i> framework's symbols.....	15
Fig. 1.4.	<i>i*</i> tutoring system: an excerpt.....	15
Fig. 1.5.	Multi-methodological approach applied in the PhD thesis.....	21
Fig. 2.1.	Genealogy of the <i>i*</i> framework and variants.....	28
Fig. 2.2.	The <i>i*</i> reference metamodel.....	41
Fig. 3.1.	Levels of conceptual interoperability (LCIM) from [TOLK'03].....	47
Fig. 3.2.	Denotation and reference and its extension to representational languages....	50
Fig. 3.3.	Phases of Kuhn's paradigmatic view of scientific progress.....	55
Fig. 3.4.	Bourdieu's idea of scientific fields.....	56
Fig. 3.5.	Lotman's idea of semiospheres.....	59
Fig. 3.6.	The structure of the <i>i*</i> 's semiosphere and the overlapping of <i>i*</i> variants.....	62
Fig. 3.7.	Translation cases between two <i>i*</i> variants by denotation functions.....	63
Fig. 3.8.	Comparative between co-adaptation and interoperability via metamodel refactoring.....	63
Fig. 3.9.	Summary of semantic preserving relationships in Wachsmuth's framework.....	64
Fig. 3.10.	Comparing absence and presence of an <i>i*</i> supermetamodel for model translations.....	65
Fig. 4.1.	<i>i*</i> requirements representation for an interoperability proposal.....	68
Fig. 4.2.	The <i>i*</i> core concepts in the context of the <i>i*</i> metamodel.....	73
Fig. 4.3.	The iStarML's semiosphere structure.....	76
Fig. 4.4.	Example of <i>i*</i> -Tropos's diagram.....	77
Fig. 4.5.	The iStarML code for the <i>i*</i> -Tropos diagram of Fig. 4.4.....	78
Fig. 4.6.	Example of Security-aware Tropos' diagram.....	79
Fig. 4.7.	The iStarML code for the Security-aware Tropos' diagram of Fig. 4.6.....	79
Fig. 4.8.	The <i>i*</i> supermetamodel.....	82
Fig. 4.9.	Translation algorithm from the <i>i*</i> supermetamodel to an <i>i*</i> variant.....	84
Fig. 5.1.	Particularities of the HiME's metamodel related to inheritance.....	92
Fig. 5.2.	Model created using jUCMNAv (top) and the model read from HiME (bottom).....	94
Fig. 5.3.	Goal and actor metric calculation using XPath on iStarML files.....	96
Fig. 5.4.	Using the ccistarmml package to parse and modify an iStarML file.....	97
Fig. 5.5.	Xquery-istarmml generation of questions for validating <i>i*</i> models.....	98
Fig. 5.6.	Confidence intervals of the <i>i*</i> community perception about interoperability and iStarML.....	101

List of Tables

Table 2.1. Comparative of relationships in main <i>i*</i> variants	32
Table 2.2. Variations proposed by the <i>i*</i> community in the period 2006-2010 (selected venues only). Each paper increments at most in 1 each column.	33
Table 2.3. Analysis of some existing proposals using <i>i*</i> together with other formalisms	37
Table 2.6. The <i>i*</i> 's metamodels	40
Table 2.7. Detailed of purposes and interoperability issues of <i>i*</i> tools, summarized from [IWIK]	43
Table 4.1. iStarML abstract core concepts, tags and variation representations.....	75
Table 4.2. EBNF for the tag <actor>	76
Table 4.3. An approach to a conceptual matching between Waschmuth and Bunge's frameworks	82
Table 5.1. Classification of specific model translations from OME to jUCMNav.....	91
Table 6.1. Summary of research questions and their approached answers.....	104

List of Acronyms

AORE	Agent-oriented Requirements Engineering
BNF	Backus–Naur Form. The notation for context-free grammars
EBNF	Extended BNF
GORE	Goal-oriented Requirements Engineering
ITU-T	International Telecommunication Union , Telecommunication Standardization Sector
JCR	Journal Citation Report
MDD	Model Driven Development
MOF	Meta Object Facility. The OMG's specification for metamodels
NFR	Non-Functional Requirement
OCL	Object Constraint Language. The OMG's specification for UML constraints
OMG	The Object Management Group
RCP	Rich Client Platform. The programming framework
RE	Requirements Engineering
SD	Strategic Dependency Model. The <i>i*</i> diagram.
SE	Software Engineering
SR	Strategic Rationale Model. The <i>i*</i> diagram.
SVG	Scalable Vector Graphic. The W3C's recommendation
UML	Unified Modelling Language. The OMG's specification for Software Modelling
URN	User Requirements Notation
W3C	World Wide Web Consortium
XML	Extensible Markup Language. The W3C's recommendation
XSL	Extensible Stylesheet Language. The W3C's recommendation

CHAPTER 1. Introduction

1. Introduction

This PhD thesis has been formulated in the context of the Software Program at the Universitat Politècnica de Catalunya (UPC). Its domain of interest is Requirements Engineering. It tackles a fundamental problem found in one of the most widespread goal-oriented requirements engineering approaches, the i^* framework. The language proposed by this framework does not have a unified version and thus a problem of model interoperability appears. This problem has as theoretical consequence the difficulty on having a common understanding of the framework shared by the whole community. As a practical consequence, the diverse tools that have been created around the i^* framework are not able to interoperate and remain isolated; furthermore, repositories of models shared by the community are not easy to build.

This PhD thesis presents the problem in detail, proposes a theoretical foundation to characterise it and a formal framework to support model interoperability, whilst informing of unsolvable problems. From a practical point of view, the results of the thesis have been used to establish interoperability between different existing tools.

1.1 *Disciplinary Context*

1.1.1 Software Engineering and Requirements Engineering

This PhD thesis is formulated in the context of the Software Engineering (SE) discipline. SE is understood as the engineering discipline concerned with all technical and non-technical aspects of software production [SOMM'04].

The presentation of SE as an engineering discipline is relatively new and its body of knowledge has been synthesized less than ten years ago [ABRA'04]. Arguments supporting the vision of SE as an engineering discipline can be found in [WASS'96]. One of the main arguments is that engineering principles have been successfully used in order to build complex computer systems. In [SHAW'90] this position is defended as a result of reflecting on what “engineering” means.

In the context of SE, many approaches to the software development process have been formulated, In spite of their differences, virtually all of them include the Requirements Engineering (RE) phase. RE has been defined as the *“the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems”* [ZAVE'97]. It is composed of several activities that may slight vary among different proposals, but for the purposes of understanding we take as reference the proposal by Sommerville [SOMM'05] which includes elicitation, analysis, negotiation, validation, negotiation, documentation, and management. In Fig. 1.1 we reproduce the original illustration of this proposal.

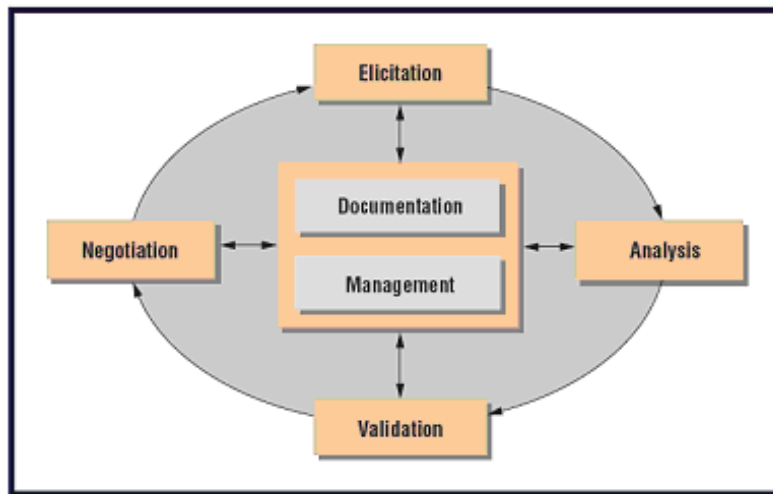


Fig. 1.1. Requirements Engineering phases according to Sommerville [SOMM'05]

Moreover, it is recognized that RE is also concerned with the relationships among these factors and their evolution over time. It is emphasised that RE is inherently broad, interdisciplinary and open-ended because it embraces from real life situations to mathematical specification languages. In the same spirit, Lamsweerde [LAMS'00] said that *“The target system is not just a piece of software, but also comprises the environment that will surround it”*. Besides, he considers many facets involved into RE, such as socio-economic, physical, technical, operational and evolutionary, among others. RE also involves non-functional concerns like safety, security, usability, flexibility, performance, robustness, interoperability, cost, maintainability and so forth. Furthermore, the RE process has many actors involved, namely, customers, commissioners, users, domain experts, software developers, system maintainers, each

one with different background, skills, knowledge, concerns, perceptions, expressions and goals [LAMS'00].

In RE social and technical concerns are usually mixed. Moreover different scholars agree about the separation of the requirements stage between early stage (organizational analysis) and late stage (requirements analysis). This separation of concerns produces a differentiation between research techniques that focus on social concerns (applicable at the early RE stage) and those that focus on technical concerns (for late RE).

In summary, we can say that RE is a (sub) discipline in SE, which has defined its relevance and activities in a wide scope involving technical and social concerns. Moreover, we can also assert that it has an evolving body of knowledge. In order to deal with this body of knowledge, RE conforms a broadly recognized research ambit that has practical and methodological open issues [CHEN'07;CHUN'09;JARK'11;LAMS'08;WIER'05;YU'09].

1.1.2 Goal-oriented and Agent-oriented RE Paradigms

In this section we aim at contextualizing the two RE paradigms where our work applies, considering first SE and then RE itself.

Every discipline evolution can be observed by dominant perspectives, which are generally called *paradigms*. However, the concept of paradigm has different interpretations depending on the point of view which. In our case, it can be taken as an engineering discipline or as a research discipline. We mainly rely this assertion on the work of Basili [BASI'86;BASI'96] and Kitchenham [KITC'02;KITC'04] among other relevant scholars who start from the assumption that SE is a research field.

One of the most influencing scholars on the scientific point of view is Kuhn [KUHN'96]. He sustains that scientific progress is achieved through paradigmatic shifts. He understands a paradigm as the total pattern of perceiving, conceptualizing, acting, validating, and valuing associated with a particular image of reality that prevails in a science or a branch of science. A paradigmatic change implies a perspective evolution with the corresponding pressure on all SE activities and artefacts.

In the first decades of SE development, its paradigms were mainly pushed by programming techniques. For example, structured programming and object-orientation first accounted with technology and later they matured to a way of perceiving, conceptualizing, acting, validating and valuing in SE (to refer the previous paradigm definition). More recently, SE paradigms mainly come from analyzing perspectives, which normally are expressed on abstract data types that soon after have evolved into developing frameworks. For instance, service-orientation [ERL'05; TSAI'05] and agent-orientation [JENN'98; WOOL'01] have been argued as paradigm in SE. Although it is not our goal, we conjecture that the latest Kuhn's description of paradigms, which opens the concept to minor paradigms, also seems to be suitable in SE. For example, aspect-orientation comes from a programming technique (AOP) [KICZ'97] and it has been sustained as a SE paradigm, however it has not evolved (maybe yet) into a socially relevant SE tendency. Additional information about software development paradigms can be found in [KAIS'05].

Besides these universally agreed SE paradigms, others exist that are well known by scholars but have a narrower scope. Due to the goal of our thesis, we target here RE paradigms. Among them, we are particularly interested in two, goal orientation and agent orientation. As we show in the next sections, they are connected and constitute research mainstreams in RE and they are relevant for us because our object of research, the i^* framework, has played (and is playing) a relevant role in both of them.

Goal-oriented requirements engineering (GORE) methodologies were introduced almost twenty years ago, recognizing a crucial role of domain understanding and stakeholders' intentions for identifying requirements of a new software system [DARD'93; LAMS'01; MYLO'99]. GORE means that both requirements elicitation and requirements specification are focused on goals. Although in [LAMS'01] it is said that [ROSS'77] has early focalized the requirements stage on goals, it is not until the nineties that goal-orientation emerges like a software engineering research mainstream [LAMS'00; MYLO'99]. Under this conception "*information systems are seen as fulfilling a certain purpose in an organisation and requirements engineering helps in the conceptualisation of these purposeful systems*" [ROLL'06].

There are different advantages of focalizing RE on goals [ROLL'06] [LAMS'01]. Just to mention some of them, we remark that: (i) goal modelling has proven to be an effective way to elicit requirements because the rationale for developing a system is found outside it, in the target organization in which system shall function; (ii) goals can be formulated at different levels of abstraction, from strategic concerns to technical concerns; (iii) goals allow covering both functional and non-functional issues, and (iv) goals are less volatile with respect to requirements which represent possible ways to meet them allowing modelling and reasoning about different alternatives to satisfy a high level goal.

The second referenced paradigm is agent-orientation. In [JENN'98] it is said that, in spite of the fact that agent orientation represents a “*melting pot*” of ideas originating in many areas (namely in distributed computing, object-oriented systems, software engineering, artificial intelligence, economics, sociology, and organisational science), the basics have become common for some related disciplines, and agent orientation “*offers a natural and powerful means of analysing, designing, and implementing a diverse range of software solutions*” [JENN'98]. Wooldridge and Cincarani [WOOL'01] hold that an agent is a system that “*enjoys*” properties like autonomy, reactivity, pro-activeness and social ability.

Agent-orientation has been a predominant paradigm and has guided a considerable number of research and development in different areas such as computer science, logic theoretical frameworks, agent-oriented languages and software environments. When this conceptual framework is used as a design metaphor, then the perspective is considering agent-orientation like a software development paradigm. Different scholars have sustained this position [CERN'05;GIOR'03;JENN'00;LIND'01;YU'02]. This design perspective has also impacted on RE yielding to what has been called Agent-oriented Requirements Engineering (AORE) [BIES'06;GOME'04;YU'97].

1.1.3 The *i** Framework

1.1.3.1 *i** and the RE paradigms

Several frameworks have been proposed in the context of the GORE paradigm. Two of them are clearly dominant, *KAOS* [DARD'93] and *i** [Yu'95]. Both frameworks propose their own modelling language, with a specific set of conceptual entities; a graphical notation to depict models; and a set of analysis techniques. The life cycle proposed by these frameworks was summarized by Kavakli [KAVA'02]. In her analysis, she distinguished between the activities and the models suggested by the goal-oriented frameworks. In Fig. 1.2 we reproduce both schemas. What is interesting to observe is that the concept of actor appears explicitly in the *i** life cycle which indicates an explicit association from goals to actors as part of the modelling activities. This is relevant because this assignment of responsibilities, i.e. what actor had or must have what goal, is the base of agent-oriented modelling [MAO'05]. Maybe there is an explicative factor coming from the fact that *i** was early and explicitly proposed as an agent-oriented framework [CAST'01;YU'97;YU'02] whilst *KAOS* was formulated “just” as a goal-oriented approach [LAMS'08;LETI'02].

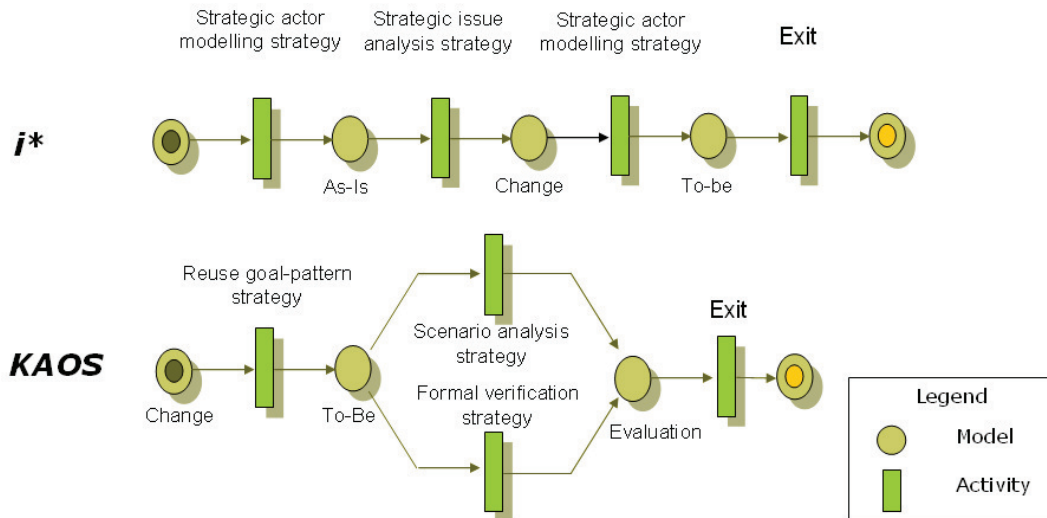


Fig. 1.2. Kavakli's comparison of *KAOS* and *i** life cycles

As a consequence, the *i** framework is considered by the RE community both as a GORE and an AORE framework. Therefore, combining goals and agents altogether

allows a goal-oriented modelling framework dealing with the foundations of agent design and its power of expression will depend on the set of language constructors.

1.1.3.2 A Brief Presentation of *i**

Developed in the first half of the 90's [YU'93;YU'94;Yu'95], the *i** framework proposes the use of two models, each one corresponding to a different abstraction level: Strategic Dependency (SD) models represent the intentional level and Strategic Rationale (SR) models represent 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 (*dependor*) depends on some other (*dependee*) in order to accomplish some intention (*dependum*).

The abstract concept of intention is materialized in specific intentional elements such as *resource*, *task*, *goal* or *softgoal*. Softgoals represent quality goals or goals that normally do not have a social agreement about how they can be accomplished. In a *dependency* it is also possible to define the importance (*strength*) for each of the involved actors using three values: *open*, *committed* and *critical*. At this strategic level, actors can be specialized into *agents*, *roles* and *positions*. A position *covers* roles. The agents represent particular instances of people, machines or software within the organization and they *occupy* positions (and, consequently, they *play* the roles covered by these positions). The actors and their specializations can be decomposed into other actors using the *is-part-of* relationship. Other domain-dependent specializations can be specified by the *is_a* relationship.

An SR model allows visualizing the intentional elements into 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 actor's boundary. The elements inside the SR model are decomposed according to three types of relationships among intentional elements:

- (i) *Means-end links*, which establish that one or more intentional elements are the means that contribute to the achievement of an end. The "end" can be a *goal*, *task*, *resource*, or *softgoal*, whereas the "means" is usually a task. There is an OR-relation

when there are many means, which indicate the different ways to obtain the end. In the original i^* definition, the possible relationships were defined as: Goal-Task, Resource-Task, Task-Task, Softgoal-Task, Softgoal-Softgoal and Goal-Goal. In Means-end links, with a softgoal as end, it is possible to specify if the contribution of the means towards the end is negative or positive.

(ii) *Task-decomposition links*, which state the decomposition of a task into different intentional elements. There is an AND-relation when a task is decomposed into more than one intentional element. It is also possible to define constraints to refine this relationship. The importance of the intentional element in the accomplishment of the task can also be marked in the same way that in dependencies of an SD model.

(iii) *Contributions links*, which represent how intentional elements contribute to the accomplishment of softgoals. These contributions can be qualitatively measured by symbols which in the case of the original i^* definition are +, ++, - and --.

Besides, SR models have additional elements of reasoning such as *routines*, *rules* and *beliefs*. A routine represents one particular course of action (one alternative) to attain the actor's goal among all alternatives. Rules and beliefs can be considered as conditions that have to be fulfilled to apply routines. These i^* elements have graphical symbols which we show in Fig. 1.3. As a small i^* sample we show in Fig. 1.4 an excerpt of an i^* Tutoring System diagram. In this figure, there appear many of the described language constructors.

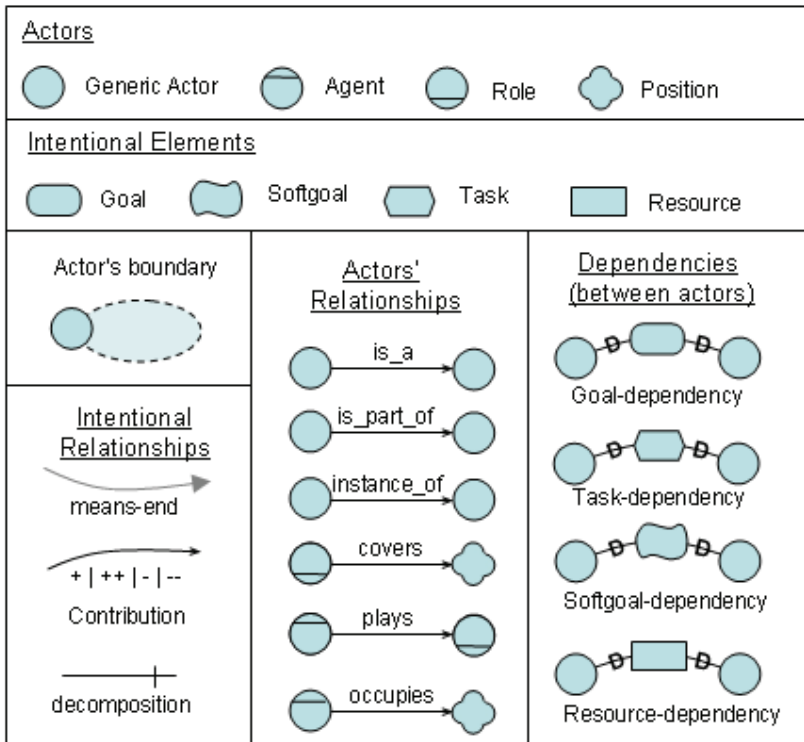


Fig. 1.3. *i** framework's symbols

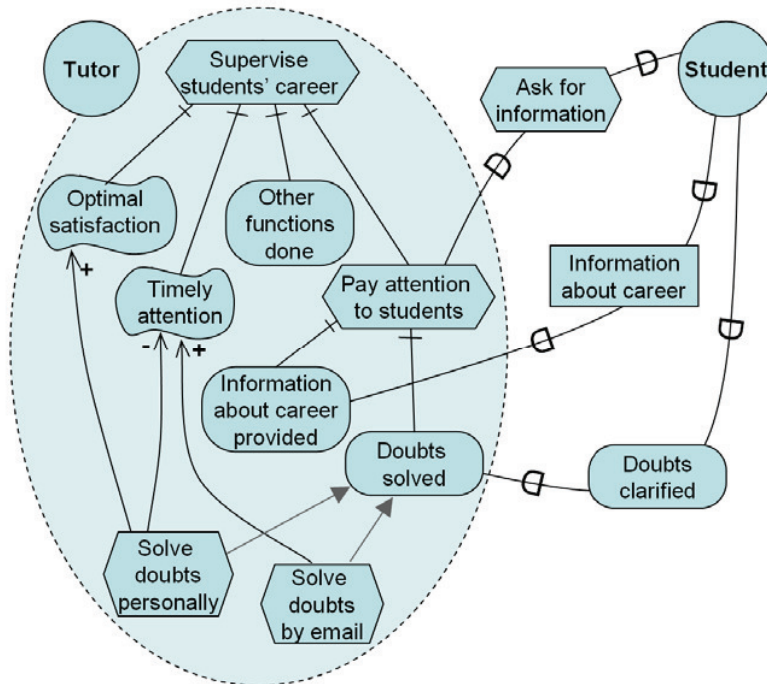


Fig. 1.4. *i** tutoring system: an excerpt

In this example appear two actors: a tutor and a student, which have three dependencies between them, a task-dependency (*Ask for information*) which means that the *Tutor* delegates (trusts) that the *Student* will ask for information in order that he/she can accomplish the task *Pay attention to students*. That is because the sense of the dependency is from the *Tutor* to the *Student*. The other two dependencies come from the *Student* to the *Tutor*. The resource-dependency (*Information about career*) means that there is a resource which could be provided by the *Tutor*, therefore the *Student* depends on the *Tutor* because of this information. Similarly, it also depends on him/her to accomplish the goal of clarifying the doubts. In the case of this diagram, the SR diagram is open and it shows the relationships among the internal elements of the *Tutor*. From this set is remarkable that some quality concerns like *Timely attention*, which is represented as a softgoal, is part of a decomposition relationship, i.e. that qualities can be modelled and, moreover, be analysed from the perspective of which concrete intentional elements (in this case tasks), can contribute to reach that quality concern.

1.2 Objectives and Research Approach

1.2.1 Definition of the Problem

Since i^* emerged in the RE community in the mid-90s, different research groups have proposed variations to the modelling language proposed in the i^* framework (for the sake of brevity, we will name it “the i^* language”). There are mainly two reasons behind this fact:

- The definition of the i^* language is loose in some parts, and some groups have opted by different solutions or proposed slight changes to the original definition. The absence of a universally agreed metamodel has accentuated this effect [FRAN'10].
- Some groups have used the i^* framework with very different purposes thus different concepts have become necessary, from intentional ones like trust, delegation and compliance, to other more related with the modelling of things, like service or aspect (see [YU'11] for an updated summary).

The adaptability of i^* to these different needs is part of its own nature, therefore these variations are not to be considered pernicious, on the contrary, flexibility may be

considered one of the framework's key success features. However, there are some obvious implications that are not so desirable:

- It damages the common understanding of the *i** framework by the whole community.
- It acts as a barrier for the adoption and diffusion of *i** by industry since practitioners are forced to choose among different *i** variants, without a clear rationale behind.
- It makes difficult to build a repository of *i** models shared and directly used by the whole community.
- It also hampers the possibility of interconnecting different *i** tools that are not compliant to the same *i** language variation.
- Finally, it makes techniques defined for one *i** variation not directly applicable into another variation.

In this context, we find necessary to study and qualify the brand of *i** variants, the existence of communalities among them, the emphasis of each proposal and remarkably the possibilities of interaction and interoperability issues among the *i** variants. Given the potential practical impact of the research, technology support for interoperability is also worth to be considered. Therefore the proposed approach should consider semantic commonalities and differences and, besides, a way of integrating current and maybe future differences.

1.2.2 Relevance of the Problem

As stated in the section 1.1.3, *i** became increasingly popular and widely used in fact not just in the RE community but in others. Some resources give evidence of this usage, such as the official *i** web site (<http://www.cs.toronto.edu/km/istar/>) and the wiki-based collaborative site (<http://istar.rwth-aachen.de/tiki-index.php>). In [YU'09] it is reviewed the roadmap of the *i** framework, its main applications and research lines. As an additional indicator, we recently reported the results of a literature review over 10 selected sources (two of them *i**-specific) that discovered 146 papers using *i** [CARE'11a].

A successful interoperability approach would support the areas where i^* is used, mainly Requirements Engineering, Business Process Modelling, Information System Design and Agent-oriented Software Engineering, among others.

1.2.3 Objective and Research Questions

We have formulated the following objective:

Objective: The i^ Interoperability Problem: The aim of this thesis is to propose a framework to understand the variations of the i^* modelling language and, considering this framework, to generate a proposal to support the interoperability and integration of these variations.*

We think that accomplishing this objective would be enough for answering the following theoretical and practical research questions. We group the questions as a set of phases:

A. Understanding and characterizing the problem

- RQ.1. What i^* variations are proposed and how can they be characterized?
- RQ.2. Which is an appropriate framework for analysing i^* differences and similarities?
- RQ.3. Why it happens? In other words, why an apparently small research community has generated so many different i^* variants? What are the implications?
- RQ.4. What are the fundamentals concepts that could guide us to an interoperability approach?

B. Approaching to interoperability for i^* variants

- RQ.5. Is it possible to formulate an interoperability framework which includes structures to support future i^* variants?

RQ.6. How to deal with semantic differences and to characterize lost of semantic if that occurs?

RQ.7. Which is the most efficient form that this semantic framework may take?

C. Evaluating the proposed interoperability framework

RQ.8. Is it possible to implement the formulated semantic framework in the current scenario of i^* -related tools?

RQ.9. Is there a perception in the i^* community about an interoperability problem? How acceptable is our approach for it?

RQ.10. Are there other applications that could bring a positive evaluation of using the proposed interoperability framework?

1.2.4 Research Method

1.2.4.1 Research Methods in SE

For tackling this research problem, we have studied different SE methodological choices. In [GLAS'02] the diversity of methods used in SE research is illustrated. The analyzed survey showed that 43.5% of research efforts use *Conceptual Analysis*, plus to a 10% of Math-based *Conceptual Analysis*. *Proof of Concepts* is used 17.2%, which generally takes the form of a software prototype development. Therefore, *Conceptual Analysis* and *Proof of Concepts* cover together more than the 70% of methodological choice in the SE research. Far away appears *Laboratory Experiments* used in a 3% of the researches, and last a list of other methodologies that do not reach 2% each one.

A different set of methods is identified in [WYNE'97]. Here the research area is specifically focused on Software Development methodologies. A set of different methods is described and its feasibility for solving research problems is analyzed. In particular, *Descriptive Research* is defined as a method to understand a phenomenon by mean of a deep study of past research and literature.

Moreover, in [BASI'93] a multi-methodological approach is promoted and a focus on empirical evaluation is supported. Special emphasis is put on quantitative research approaches following the tradition of inferential statistics. These ideas have implied an epistemological revolution on SE research because the discipline has understood that an improved research would come from other quantitative, but mainly empirical, approaches. Therefore a strong empirical movement has been developed these years [KITC'02;ZANN'06], however, the nature of data sources have been a critical point because most of the quantitative studies use students as source of statistical samples [HOFE'07].

1.2.4.2 Research Method applied in this PhD thesis

Following these recommendations and advices from both traditional research behaviour on SE and innovative epistemological positions, we formulate our research proposal as a multi-methodological approach of four stages:

- (a) *Descriptive research*, for understanding the behaviour of the i^* community, conceptual branches and emphasis of the proposals;
- (b) *Conceptual analysis*, for modelling the differences and looking for a stable conceptual kernel if it exists;
- (c) *Proof of concept*, for showing that an interoperability proposal can be formulated in an evolved semantic scenario, and
- (d) *Quantitative research*, for getting an empirical first evaluation of the specific interoperability proposal.

In Fig. 1.5 we summarize this multi-methodological approach. We have used a stair metaphor in order to represent the necessity of following the steps in the ascending order.

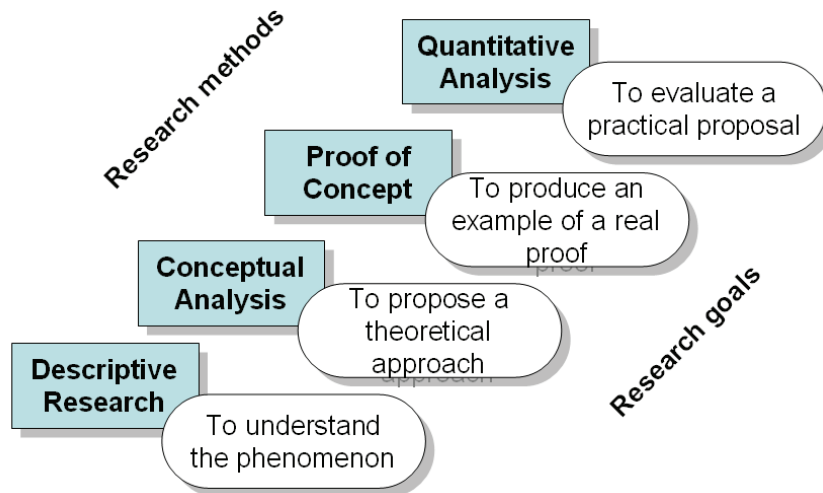


Fig. 1.5. Multi-methodological approach applied in the PhD thesis

1.2.3 Document's Overview

In order to present the results of each research stage, we firstly present in Chapter 2 a review of the genealogy of i^* variants and describe their differences. Additionally, we review the combination of i^* with other modelling frameworks analyzing the type of changes proposed over it. We end this chapter modelling relevant i^* differences under a metamodel-based approach. In Chapter 3, we review different theoretical approaches for dealing with interoperability in polysemantic scenarios, including sociology of science, cybernetics and linguistics approaches. The conceptual perspective, i.e. concepts related to semantics and modelling languages under a scientific approach, comes from philosophy of science. Finally, we arrive to a theoretical interoperability approach that is complemented with the adaptation of a metamodel approach to characterize semantic preservation of translations. In Chapter 4, we propose the iStarML format language, an XML proposal for interchanging i^* models which complains the theoretical proposal. Given the different semantic-preserving characteristics around translations, we offer a simple algorithm for a generic translation that reports the semantic preservation result. Finally, in Chapter 5 we present a summary of iStarML applications separates in three sections: proofs of concepts about models interchange, iStarML applications in the context of software process and a quantitative approach to the initial perception of the i^* community about interoperability and iStarML. The conclusions are presented in Chapter 6 where we summarize the research path followed; it includes a methodological

view and a summary of the research questions and our approach to them. We end the chapter remarking our perspective about derived open issues and our interest on some specific future work.

CHAPTER 2. Analysis of the i^* Language from an Interoperability Perspective

2. Analysis of the *i** Language from an Interoperability Perspective

The proposal of this thesis has been a long-term work that has been carried out by different studies. Most of the time, these studies were focalised on one or two of our research questions and were made in different moments. They allowed not only advancing in the understanding of *i** but also refining and redefining our research questions and research work according to the findings, as it is normal in qualitative research. In this chapter, we present the descriptive and interpretative initial work which allowed confirming the existence of *i** variants, knowing the influences among them and having a dimension of their differences.

We reviewed the historical evolution of *i** along time through a literature review conducted in 2007 and lately updated in 2008. This study identified the most relevant variants at that moment and confirmed a previous hypothesis about the three main variants of *i**. Recently, we conducted a second literature review in which we identified the most usual variations in the language proposed by the community. We present first these two studies with the aim of showing the diversity of *i** variants.

In order to show specific technical differences, we get back again to a study we performed earlier in 2005, focused on the three most widespread *i** variants. In this study, we especially focused on differences concerning intentional element links. Next, we introduce metamodels as the way to describe the variants and present an unpublished collection of existing proposals on *i** metamodels. This allows illustrating the type of *i** differences at the metalevel using a reference metamodel that we proposed that same year (2005) where, by using refactoring operations, it is possible to obtain the metamodels of the three main *i** variants. The type and amount of these refactoring is a way to understand the differences among the variants.

Finally, with the goal of showing how these differences have implied a scenario without model interoperability, we summarize the information available of the current offers of *i** tools from the *i** wiki web site [IWIK] and describe their interoperability levels.

Using these descriptive studies and analyses we answer some of our initial research questions about the nature of differences, their origin, perspectives for analysis and state of the interoperability in the *i** community.

2.1 Historical Evolution of *i**

In this section we summarize the more relevant points of the evolution of *i** that is summarized in Fig. 2.1.

2.1.1 Antecedents

In the field of goal- and agent-oriented modelling languages, we may cite KAOS [DARD'91;DARD'93] as a fundamental proposal that includes several concepts that also appear in *i**: system goal, goal reduction (which is later called *goal decomposition* in *i**) and the notion of linking a goal to agents, which have the responsibility to accomplish the goals. Besides, in 1992, we find the basis of the Non-Functional Requirements (NFR) Framework [MYLO'92], also known as the NFR Framework proposal. It introduces the concept of non-functional requirement as a system goal that should be satisfied. Also, in this proposal appear the concepts of *dependency* link between goals as well as a type of link called *justification-for-selection*, which later became the *contribution-to-softgoal* link in *i**.

Then, in [YU'93], it is formulated a simple but relevant change from the modelling point of view. Conversely to KAOS, where agents are associated to goals, in [YU'93] goals and tasks are linked to agents, conforming dependencies among system agents, thus the point of view is agent-oriented, in the sense of the individuals, and social-oriented (or context-oriented) in the sense of the dependencies between agents. In addition, the agents are extended to roles and positions, altogether becoming actors. Afterwards, Yu [YU'93] identifies many agent conceptual contributions from the Artificial Intelligence discipline.

In [YU'94], almost all the elements of i^* are referenced, in fact it is said that “*we propose a model that aims to capture the motivations, intents, and rationales that underlie a software process*”. Therefore, we may find dependencies under the form depender-dependum-dependee, intentional elements like resources, task, goals and softgoals, and the contribution to softgoal and actor’s subtypes were formulated. Furthermore, an actor metamodel in Telos [MYLO'90] is provided.

As the culmination of this initial stage, the i^* framework is proposed in [Yu'95]. The rest of the elements were added to the language, remarkably by two models: the Strategy Dependency (SD) and Strategic Rationale (SR) models. In that document were also proposed goal and softgoal decompositions, using means-end relationships, and task decompositions, using the homonymous relationship. Moreover, a comprehensive collection of examples about business process reengineering in the context of software systems was provided.

2.1.2 Chronology

To develop an historical perspective of i^* , in 2007 we reviewed the main computer science libraries from 1995 to 2007. Specifically, we focused the search on: the ACM digital library, IEEE Xplore, editorial’s web sites from Springer (www.springerlink.com) and Elsevier (www.sciencedirect.com), the integrated bibliography indexers DBLP (<http://www.informatik.uni-trier.de/~ley/db/>) and BibFinder (<http://kilimanjaro.eas.asu.edu>¹). These sources were ranked using the electronic resources Google Scholar and Citeseer for paper citations, and the Journal Citation Report (JCR) for journals. Moreover, we specially looked for the proceedings of conferences and workshops on agent orientation (AMAS, AOIS, AOSE, CEEMAS), requirements engineering and knowledge engineering (RE, CAiSE, REFSQ, WER, ER, SEKE) which not always are part of the above mentioned bibliographic databases. Finally, the i^* web site (<http://www.cs.toronto.edu/km/istar/>) and *Tropos Project* at Trento (<http://trinity.dit.unitn.it/~tropos/>) were also included in the search. Meetings, workshops and conference participations, and personal communications were instrumental to locate documents not available through literature review.

¹ Although the Bibfinder is not currently operational, it was a main reference site at that moment.

Because of this analysis, we found some sequences of publications constituting relevant i^* variants and others which can be considered as new and specialized proposals which have not become a variant. On this set of variants, we detected two particular situations that we tracked: 1) silences and ambiguities: according to Meyer's definition [MEYE'85] it means that the semantics (or even the syntax) of some language construct has not been detailed and it must be inferred from the examples, which is not always possible; 2) contradictions, either because two different documents state contradictory usages in reference to a particular language construct, or because an example does not follow the stated semantics.

The main conclusion is that there has been, and is being currently under progress, a wide and diverse scientific production around i^* topics. This fact becomes apparent by looking at Fig. 2.1 where we have considered the milestones of the three main lines of evolution (what we call *leading variants*): the i^* language proposed as part of the i^* framework; GRL, which has become part of the Z.151 standard [ITUT'08]; and the language proposed as part of the Tropos method [BRES'04;CAST'01]. We show in the figure the main milestones that have been produced for each leading variant.

For the i^* language, we mention Yu's thesis document [Yu'95], which lately was summarized on the i^* wiki [IWIK] (with some minor changes). For GRL, we mention the first document [GRL'01], lately consolidated on the standard itself [ITUT'08]. For Tropos, the first widespread proposal is [CAST'01].

The main variants appear in both *Requirements Engineering* and *Agent-oriented Software Engineering* research tracks. Finally, we updated this study in 2009 adding to new i^* variants proposals, one related to norms modelling [SIEN'08] and another regarding service modelling. We have analysed them by looking for the evolution of the used concepts and language constructs in the modelling languages; the arrows in Fig. 2.1 illustrate the influences on each other. Transparent ovals stand for proposals that are in the mainstream of the i^* evolution, whilst grey ovals identify other proposals that have influenced this evolution. The thick lines in transparent ovals identify the three milestones that characterise each of the considered lines of evolution. Most of the identified variants have been formulated and/or detailed beyond the mentioned

bibliographic reference, however for the purposes of this chronology we have kept the focus on the first proposal that formulates the variation as an autonomous one. Moreover, the localization on the graph corresponds to this reference.

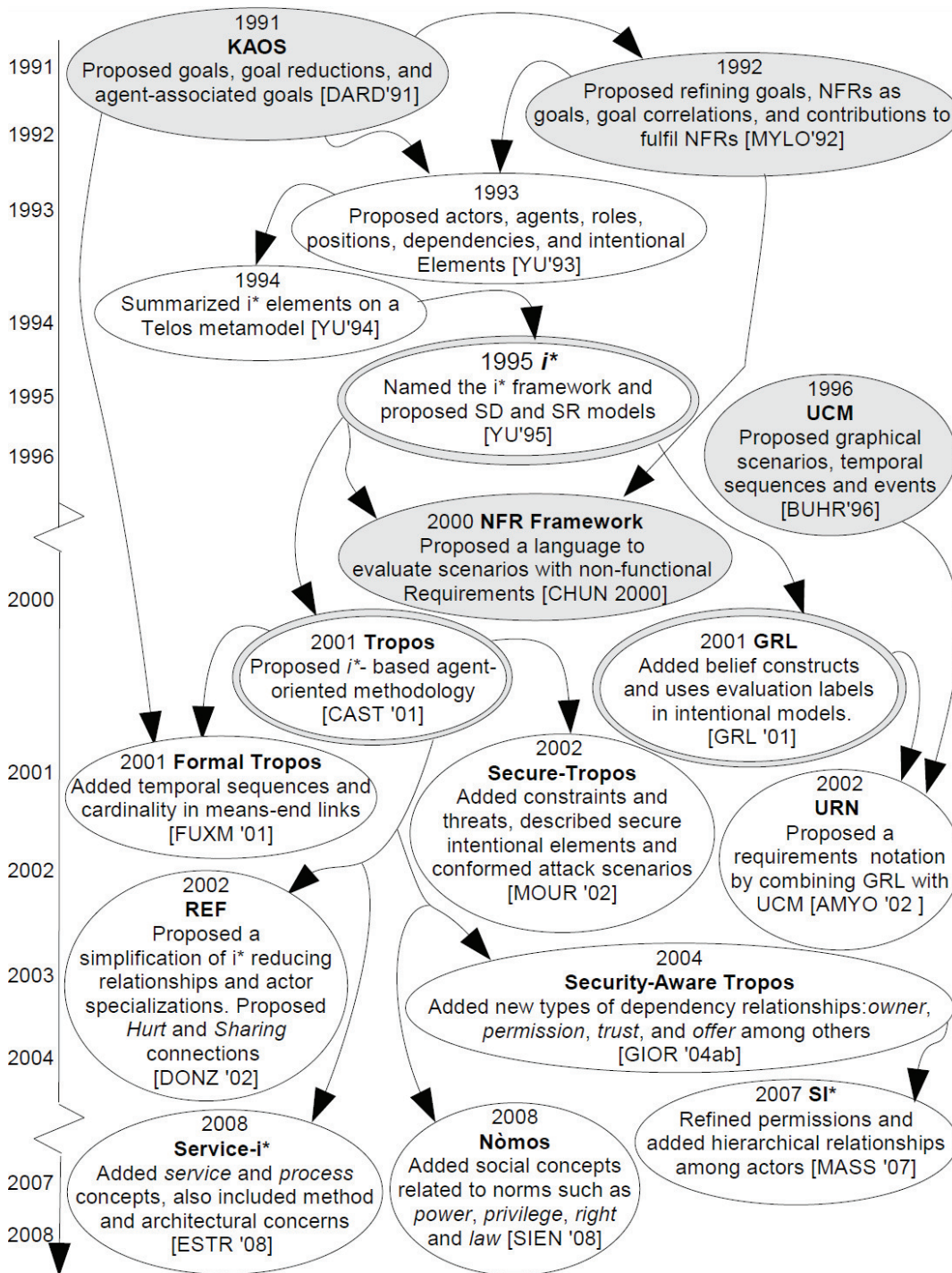


Fig. 2.1. Genealogy of the *i** framework and variants

2.2 The *i** Language: Summary of Variations

2.2.1 Analysis of the Three *i** Leading Variants

In the previous section we have identified what we consider the three existing *i** leading variants. In Chapter 1 we have already described the main, seminal one, the language defined in the *i** framework itself as defined by E. Yu. We focus now on GRL and Tropos' variants mainly highlighting their differences with respect to that seminal *i** [AYAL'05]. It is important to remark that, given the purposes of the thesis, we are focusing on the language proposed in these variants, not paying attention to methods or techniques.

The Goal-oriented Requirement Language (GRL) [GRL'01;ITUT'08] is a language used in goal-oriented modelling and reasoning with non-functional requirements. It has been strongly influenced by the NFR framework [MYLO'92]. Its main aim is to specify non-functional requirements. GRL is part of URN (User Requirements Notation) [AMYO'02] that has been accepted as standard of ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) [ITUT'08].

GRL uses the following conceptual categories (as *i** does): intentional elements, intentional links, actors and their links, and dependencies among actors. The main differences with respect to *i** are:

- Specializations of actors (role, position and agent) are not defined.
- Actor links are not part of the language.
- It offers constructs for enabling relationships with external elements.
- It has additional elements of argumentation and/or contextualization as beliefs, correlations and contribution types.
- It uses new evaluation icons for specifying satisfaction states (satisfied, weakly satisfied, denied, weakly denied).

Tropos [BRES'04;CAST'01;GIUN'03;SANN'02] is an agent-oriented software methodology which has been considered one of the most relevant proposals in the field, in fact it is frequently considered in comparative analyses among agent-oriented software methodologies [ALON'04;CERN'05;DAM'03;SUDE'05]. The application of the Tropos methodology starts at an early stage of requirements analysis and ends with the software implementation. Its predominant position in RE is precisely rooted on its use of the *i** framework. In the requirements analysis stage, the actors are used to model stakeholders of the domain and the system to be constructed. Thus, dependencies represent dependencies between stakeholders and dependencies between them and the new software system. In the design stage, the actors represent the components of the system architecture and the agents that should be implemented. The dependencies represent the data and control interchange between components and agents, and define the abilities or responsibilities of each one that must be implemented.

The differences between the language proposed in Tropos and *i** are related to the syntax of some concepts. For example, Tropos does not distinguish between SD and SR models. However, it proposes different views for each development stage, e.g. Tropos models explicitly distinguish between aspects related to the domain modelling and to the software system.

Regarding the language, we may distinguish among two main streamlines: Tropos as proposed by Castro, Kolp and Mylopoulos [CAST'01] (we name it Tropos'01), which considers Yu's *i** as the underlying language; and Tropos as proposed by Sannicola, Perini and Giunchiglia [SANN'02] (we name it Tropos'02), which provides explicitly a user guide defining the language, which evolved and has a more mature description in [SUSI'05]. In the first case, the use of *i** does not adhere strictly to Yu's specification. For instance, this variant allows grouping actors inside the boundary of other actors and also allows decomposing softgoals while in Yu's specification, only contributions are allowed for softgoals. In spite of the fact that some of these differences do not appear explicitly defined, they are observable in the examples offered by the authors.

If we consider the three leading variants altogether, maybe the most remarkable fact is that the three main types of links, namely means-end, decomposition and contribution, have been defined and redefined in the main *i** streamlines. More precisely, they differ

in: the lexicon used; the intentional elements allowed as source and destination of the relationships; the combination of the elements that take part in the contribution; and the expressive power of the types of contributions. Although these changes seem not to have a great impact in the language, in fact they may provoke some semantics misunderstandings and misuses as shown in [WEBS'05].

In Table 2.1 we show these dissimilarities. According to the explanation above, we have split Tropos into Tropos'01 and Tropos'02. For each of the resulting variants compared, we show how the language construct is named, the valid combinations of intentional elements according to their type, and the way of combining the elements that take part in the construct (AND, OR or none). For the valid combinations, we use the symbol '[' to separate the two types of elements that may be connected by the link. In the case of means-end links, the left-hand side of the arrow represents the *means* and the right-hand side represents the *end*. In decomposition links, the left-hand side represents the basic elements of the decomposition and the right-hand side the compound element. In the contribution links, the left-hand side is the contributors' part and the right-hand side is for the contributed element (*contributee*).

Table 2.1. Comparative of relationships in main i^* variants

		Yu's i^*	GRL	Tropos'01	Tropos'02
Means-end	Name	<i>means-end</i>	<i>means-end</i>	<i>means-end</i>	<i>means-end</i>
	Connected elements	$G \rightarrow G$ $T \rightarrow G S R T$	$T \rightarrow G T R$	$G T \rightarrow T$ $T \rightarrow G$	$R T \rightarrow G S T$
	Operation	OR	OR	OR	AND
Decomposition	Name	<i>task-decomposition</i>	<i>decomposition</i>	<i>decomposition</i>	<i>AND/OR decomposition</i>
	Connected elements	$G S R T \rightarrow T$	$G S R T \rightarrow T G$	$G S T \rightarrow G S T$	$G S \rightarrow G S$
	Operation	AND	AND	AND	AND OR
Contribution	Name	<i>means-end</i>	<i>correlation. contribution</i>	<i>contribution</i>	<i>contribution</i>
	Connected elements	$S T \rightarrow S$	<i>correlation:</i> $S T \rightarrow S$ <i>contribution:</i> $S T L \rightarrow S B L$	$S T \rightarrow S$	$G S R T \rightarrow G S$
	Operation	Does not exist	<i>contribution:</i> AND, OR <i>correlation:</i> Does not exist	Does not exist	Does not exist
	Attributes	+, -, sup, sub	Make, Break, Help, Hurt, Some+, Some-, Equal, Unknown	++, +, -, --	++, +, -, --
		G Goal S Softgoal T Task	R Resource B Belief L Link (Decomposition, Contribution, Means-End or Correlation)		

2.2.2 A Quantitative Description of Recent i^* Variations

With the same goal (uncovering i^* diversity) in this section we use a different approach, more quantitative-oriented. We have carried out [CARE'11a] a review in the following conferences and journals for the period 2006-2010: CAiSE, REJ, DKE, IS Journal, RE, ER, RiGiM, WER, i^* workshop, and also including the recent book on i^* [YU'11]. Our goal has not been completing a systematic review but to get a representative sample of the community proposals in this period as a way to know what the major trends concerning language variability are. In total, we have found 146 papers about i^* in these sources (without including papers talking about goal-modelling in general, since we are interested in language-specific issues). From them, we have discarded 83, which are not really relevant to our goals (i.e., papers not directly related with the constructs offered by the language). For the remaining 63 papers, Table 2.2 shows how many of them propose addition, removal or modification of concepts classified into six different types. It must be taken into account that a single paper may propose more than one construct variation and that similar changes are proposed or assumed in different papers. In

addition, it is necessary to remark that most papers just focus on some specific part of the language; in that case we assume that the other part remains unchanged.

Table 2.2. Variations proposed by the *i** community in the period 2006-2010 (selected venues only). Each paper increments at most in 1 each column.

	Actors	Actor links	Dependencies	Intentional Elements (IE)	IE links	Diagrams
New	4	24	10	21	21	19
Removed	8	5	2	1	0	0
Changed	3	1	1	36	43	0

An analysis of this table follows:

- On actors. The most usual variation is getting rid of the distinction on types of actors, like remarkably GRL does. Some special type (e.g., “team”) may appear.
- On actor links. Most of the variants include *is_part_of* and *is_a* but some get rid of one (e.g., GRL just keeps *is_part_of*) or even both. Of course, having just a generic type of actor means not having the links bound to specific types like *plays*. Finally, some proposals use new actor links, like in Nòmos: *A embodies B* means the domain actor A has to be considered as the legal subject B in a law.
- On intentional elements. Although all virtually all variants keep the four standard types (*goal*, *softgoal*, *task* and *resource*), we may find a lot of proposals of new intentional elements. To name a few, GRL adds *beliefs*, Nòmos adds *norms*, and even *aspects* appear as dependums. There are not many modification proposals, e.g., resources may be classified as physical or informational with consequences for class diagram generation in an MDD (Model Driven Development) process.
- On intentional element links. Most of the variants keep the general idea of the three link types (*means-end*, *task decompositions* and *softgoal contributions*), some of them merge two of them, e.g., GRL defines a link *decomposition* that merges *means-end* and *task-decomposition*. Then we have lots of variations about types of decompositions (e.g., Tropos allows both *AND* and *OR* means-end links), contribution values (labels such as +, – vs. *make*, *help*, etc.), correctness conditions (e.g., whether a resource may be a mean for a goal), etc. Finally, some modifications usually occur in the form of labels, e.g., quantitative labels for contributions in GRL, multiplicity in some Tropos-based variants, etc. A special type of modification is

- relaxing some conditions, e.g., allowing links among intentional elements that belong to different actors, or contributions to goals.
- On dependencies. About modifications, we may find the addition of attributes which qualify the type of dependency, e.g., Secure-Tropos adds *trust* and *ownership* qualifiers. Then, we have new types of relationships that may be interpreted as dependencies, like Nòmos' legal relations. In addition, a quite usual variation is to get rid of dependencies' strength, probably due to the difficulty of interpreting the concept in a reasoning framework. The type of depender and dependee also presents constraints sometimes, e.g., GRL forces them to be intentional elements, actors are not allowed in this context.
 - On diagrams. The distinction among SD and SR diagrams is not always kept, some proposals just have a single model in which the actors may be gradually refined. One type of diagram that was depicted in Yu's thesis, but not recognised as such, was actor diagram, and some authors have promoted this third type of diagram. In addition, several proposals of types of diagrams exist, from the generic concept of module to specific proposals like interaction channel.

2.2.3 On the Combination of *i** with Other Frameworks

To finish this analysis, we present a fourth study, related to the combination of *i** with other modelling frameworks. From the previous research, we can conclude that variants emerge from the attempt of adapting *i** to deal with specific contexts of domain modelling, such as software production (Tropos), security and trust (SI*), etc. Using the same bibliographic data as in the previous section, we performed a different analysis for this issue [FRAN'11a]. From the same set of 146 papers on *i**, we discarded here 93 that do not tackle the combination of *i** with other modelling framework. For the remaining 53, we made a detailed analysis and when required, we consulted additional documents (e.g., PhD thesis where more details are provided, older papers, or papers in other venues). We have identified the scenarios below concerning how *i** and the modelling frameworks are aligned (see Table 2.3):

- Model coupling (8 proposals). Adds the benefits of goal-oriented models into some other, usually well-established notation (class diagrams [ALEN'09], Z

[KRIS'09], use case diagrams [SANT'02], data base schemas [JIAN'06], etc.) in a non-invasive way. As a result, both types of models coexist but they are not really merged.

- Model reinterpretation (18 proposals). In this scenario, i^* elements are fundamentally the same, but they are interpreted in a particular setting. We may mention the use of i^* for representing software architectures [GRAU'07] or software process models [ESFA'10]. This scenario facilitates the understandability of a model-based solution when i^* was already a choice of model for requirements engineering.
- Model merging (19 proposals). Brings some ontology into i^* . Representative examples are: the REA ontology and e3 together [GAIL'08], normative compliance frameworks [SIEN'09], security aspects [MOUR'07], data warehouse schemas [GIOR'05], etc. In this case, the i^* ontology is enriched with new elements that impact into several aspects ranging from purely syntactical (e.g., how to represent the new concepts) to semantic (e.g., evaluation procedures).
- Paradigm merging (8 proposals). A particular case of the scenario above, in which the GORE paradigm behind i^* is blended together with some other. Examples are the reconciliation of i^* with service-based modelling [ESTR'10;LIU'06], aspect-oriented modelling [ALEN'08] and agent-oriented modelling [GIOR'11].

We have analysed the proposals from different perspectives:

- Theoretical. How i^* is extended with new concepts from the modelling framework M in order to tailor it to a specific context or domain. E.g., there is a lack of consensus whether to represent these concepts by adding new constructs or by refining those that already exist in i^* .
- Technical. How the resulting framework may provide to the engineer capabilities that facilitate its usage and integration with the other modelling framework during the whole development and maintenance process.
- Methodological. How an i^* -based framework may deliver an effective engineering method and how it can provide means to ensure quality in a rigorous way.

- Community-related. How the proliferation of i^* -based frameworks may be articulated into a global view, instead of having isolated proposals whose lack of collaboration hampers their usability.

Note that there is a relation between the type of scenario and the type of model integration. E.g., in model reinterpretation, the syntactic integration does not apply since i^* is used directly, whilst in model coupling, semantic integration is not necessary because the two models are kept separated. Table 2.3 crosses the previously identified scenarios for i^* integration with the described challenges as example for a specific set of i^* -related proposals.

As a final summary, we can state: 1) there is not a shared vision about which elements should be included in a proposal when combining i^* with other modelling framework, and 2) the level of development of most frameworks still provides room for making them broader (in scope) and deeper (with more rigorous definition of existing elements). From these observations, we have developed our own vision of how to define such combinations, not included here since it is not part of the answer to any thesis' research questions, see [FRAN'11a] for more details.

Table 2.3. Analysis of some existing proposals using *i** together with other formalisms

Scenario	Approach	Domain	Modelling framework	Theoretical Challenges	Technical Challenges	Methodological Challenges	Community Challenges
Model coupling	Alencar et al. [ALEN'09]	MDD under OO paradigm	UML class diagrams	Implicit from transformation guidelines	Metrics provided in a separate paper	Transformation guidelines given	Not addressed
	Krishna et al. [KRIS'09]	Requirements Specification	Z specifications	Implicit from transformation guidelines	Not addressed	Transformation guidelines given	Not addressed
	Santander et al. [SANT'02]	Requirements Specification	UML use case diagrams	Implicit from transformation guidelines	Not addressed	Transformation guidelines given	Not addressed
	Jiang et al. [JIAN'06]	Data Modelling	Data Schemas	Implicit from transformation guidelines	Data quality concerns (in another paper)	Transformation guidelines given (in another paper)	Not addressed
Model reinterpretation	Grau et al. [GRAU'07]	Architectural Design	General concepts of software architecture	Semantic integration (informally)	Some structural metrics (in OCL)	Design guidelines are provided	Not addressed
	Esfahani et al. [ESFA'10]	SPI Under Agile paradigm	Agile Process Conceptual Framework	Layout of ontological alignment (by a table)	Need of process indicators suggested	A method is suggested by an example	Not addressed
Model merging	Gailly et al. [GAIL'08]	Business Modelling	REA Business Ontology + e3	Syntactic integration + transformation guidelines	Not addressed	Not addressed	Not addressed
	Siena et al. (Nómos) [SIEN'09]	Norms and Law Modelling	Hohfeld's juridical taxonomy	Transformational integration (defined in PhD thesis)	Not addressed	Design guidelines given (in a separate document)	Not addressed
	Mouratidis et al. (Secure-Tropos) [MOUR'07]	Security under AO paradigm	Jansen-Karygiannis' agent attack framework	Syntactic integration + explanations	Not addressed	It is embedded as part of Tropos Methodology	Not addressed
	Giorgini et al. [GIOR'05]	Data Warehouse modelling	DFM (Dimensional Fact Model)	Syntactic integration	Not addressed	An example shows a procedure	Not addressed
Paradigm merging	Estrada et al. [ESTR'10]	Service Modelling	Service-Oriented conceptual framework	Semantic integration (in another document)	Not addressed	Design guidelines (in another document)	Not addressed
	Liu et al. [LIU'06]	Service Modelling	Service-Oriented conceptual framework	Accurate semantic integration	Not addressed	Method provided (in another doc.)	Not addressed
	Alencar et al. [ALEN'08]	Aspect Orientation	Aspect-Oriented conceptual framework	Semantic integration implicit from design guidelines	Not addressed	A method and design guidelines are provided	Not addressed
	Giorgini et al. (Tropos) [GIOR'11]	Agent Orientation	Agent-Oriented conceptual framework	Transformational integration (integrated metamodel)	Traceability by connecting different proposed stages	Stages and design guidelines are detailed	Not addressed

2.3 The Metamodels behind *i** Variations

Metamodels have been the traditional tool in SE to express valid models. In fact, metamodels are *statements about what can be expressed in valid models of a certain modelling language* [SEID'02]. The fact of having several *i** variants is reflected in having multiple metamodels, either explicit or implicit.

The language used to specify a metamodel is called *metalanguage*. It is relevant to note that metamodels represent only what can be expressed in valid models but not what that expressions mean, i.e. a metamodel specifies the syntax of a modelling language but not what the valid constructions on that language implies to the modelled world. Therefore semantics is out of the scope of metamodelling although remarkably this is an issue that has not been always clear for the complete SE community [HARE'04].

In the case of the *i** language, its syntax has been specified with different means, by different groups and at different moments. The first published *i** metamodel appeared in the seminal proposal [YU'95] which used Telos [MYLO'90] as metalanguage. In spite of the clarity of this specification, there are some missing concepts, for example, contribution links do not appear in the metamodel. They are explained on text as a type of *means-end* link but in the metamodel there is no *purpose* to reach an instance of a *SoftgoalClass* nor *how* to contribute to it. A UML metamodel for this seminal *i** leading variant is also shown in [AYAL'05].

Tropos metamodel from [SUSI'05] uses UML as its metalanguage. In it, for example, the *Contribution* class has a mandatory value. Other evident difference is that *Softgoal* and *Goal* have been generalized to an abstract goal concept and, in order to differentiate them, the *i** concept of goal is called *Hardgoal*, thus *Goal* (abstract) is specialized to *Softgoal* and *Hardgoal*. This specialization does not appear on Yu's metamodel. This new *Goal* class appears on the associations that define intentional relationships (means-end, contribution and decomposition), therefore there are not differences between goals and softgoals in intentional relationships. Another difference is that this metamodel replaces the *Task* class by the *Plan* class.

GRL [GRL'01] also has its own definition. In this original definition BNF grammar was used as metalanguage. It shows how *Intentional Elements* are specified and the particular case of *belief*, added as a specialization of intentional element. The case of *belief* is remarkable because it conceptually appears on the Yu's thesis but without a graphical representation. A UML version of the GRL metamodel [ROY'07] shows some difference from the original BNF specification. For example, a *belief* is not considered an *intentional element* but a different type of GRL's node. Most recently, in [AMYO'09] also appears a UML representation of a GRL metamodel. Here, *Belief* is added to the intentional types. The original concept of correlation proposed in the first GRL metamodel does not appear in [ROY'07], however it appears in [AMYO'09] as a Boolean value in the contribution types.

Some efforts have appeared also to propose unifying *i** metamodels. The first attempt, trying to consolidate the three *i** leading variants, appeared at [AYAL'05]. In the same direction, it was presented the metamodel at [LUCÉ'08] that assumes the existence of *i** variants and presents a UML metamodel which can be adjusted to two variants (Yu's *i** and Tropos) by using OCL constraints. A relevant difference between these two metamodels is that in [AYAL'05] it is used a generic core *i** common concepts, whilst in [LUCÉ'08] the concepts are added over the same metamodel. Thus, while in [AYAL'05] the class *Intentional Element* appears and specific variants are obtained by adding classes, in [LUCÉ'08] all known classes appear, e.g. this metamodel has the class *Plan* (from Tropos) and the class *Task* (from *i**) as part of the set of classes. In Table 2.6 we have summarized the collected *i** metamodels and some of their relevant differences. It is necessary to remark that not all the variants have their corresponding metamodels.

As a summary, metamodels seem a proper tool to reason about *i** variants and their differences. Based on all the work presented in this chapter, we have improved the initial work presented in [AYAL'05] arriving to a Reference Metamodel for *i** which pretends to include only stable concepts in it and that may be considered the seed of this PhD thesis. In our view, starting from this reference metamodel, particular *i** variants may be obtained, which at their turn can give us a general idea of the differences (and distance) among the *i** variants. We have detailed the foundations of this work in

[CARE'10][CARE'11c][CARE'11e]². In Figure 2.2 we show a revised version of the proposed *i** Reference Model as a way of illustrate our contribution on this topic. We want to make explicit that our metamodel specification considers explicit constructs from the *i** proposal that do not necessarily have a corresponding graphic symbol or graphic differentiation in the modelling language.

Table 2.6. The *i**'s metamodels

Metamodel	Meta-language	Involved <i>i*</i> variants	Features
[Yu'95]	Telos	Seminal <i>i*</i>	Telos allows distinguishing between abstraction levels. Basic concepts are defined, basic rules for intentional link are missing however.
[GRL'01]	BNF	Seminal GRL	Adds <i>Belief</i> as intentional element and <i>correlation</i> as intentional links. It allows generating expressions without semantic.
[SANN'02]	UML	Tropos	<i>Softgoal</i> and <i>Hardgoal</i> are specializations of <i>Goal</i> . <i>Means-end</i> is not a construct in the metamodel. In its place, <i>Means-end Analysis</i> (as class) is part of it. <i>Belief</i> is a Class in the Metamodel. The concept of <i>Plan</i> replaces to <i>Task</i>
[SUST'05]	UML	Tropos	Similar to the previous one, however <i>Belief</i> is omitted from the metamodel and <i>Actor's</i> relationships appear explicit in the metamodel. Neither <i>is_part_of</i> nor <i>is_a</i> relationships are included in Tropos's metamodels.
[AYAL'05]	UML	<i>i*</i> , GRL, and Tropos	It offers a reference metamodel including common concepts from previous <i>i*</i> variants and it get each specific <i>i*</i> variant by refactoring operations.
[ROY'07]	UML	GRL	Graphical concepts are added to the metamodel (such as <i>GRLNode</i> and <i>Connection</i>). <i>Belief</i> exists but it is not considered an Intentional Element. <i>Means-end</i> does not appear as a valid type of intentional link. The concept of <i>Boundary</i> is omitted in this presentation
[LUCE'08]	UML	<i>i*</i> and Tropos	It includes some abstract concepts in a similar way than [AYAL'05] such as <i>Internal Element</i> , <i>Intentional Element</i> , and <i>Node</i> . Core <i>i*</i> concepts are all included like classes, e.g. the actor's relationships plays, covers and occupies are all UML classes. Concepts coming from different <i>i*</i> variants are represented by different colours on classes.

² It is worth to mention that, although this reference is from 2011, the scientific work is dated as of end of 2008, due to a long book editing process.

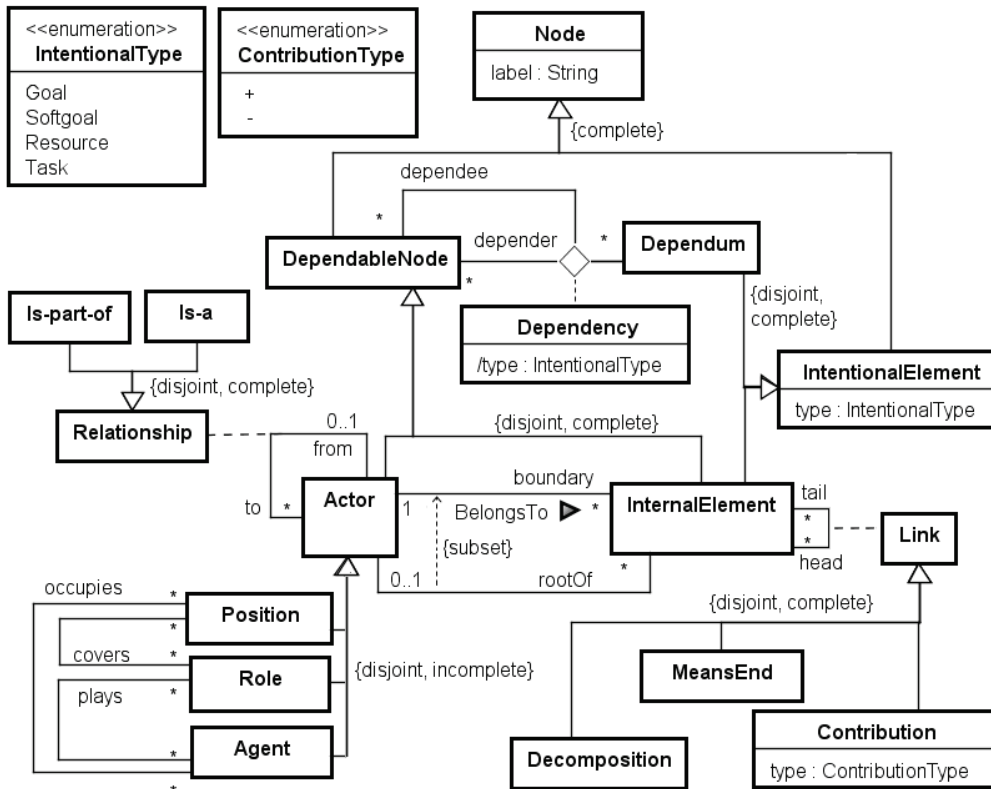


Fig. 2.2. The *i** reference metamodel

2.4 Describing the *i** Tool Interoperability Scenario

As a side effect of the growing interest around *i**, several research groups have developed software tools for a diversity of purposes. In Table 2.7 we have extracted an evaluation of tools from [IWIK]. Due to its relevance we have additionally added REDEPEND [LOCK'08] which does not appear in the mentioned comparative. In addition, two tools included presented in the Tool Fair of the fifth *i** Workshop (*iStar*'11) have been added: *I*Prefer* [LI'11] and *iStar Tool* [MALT'11]. Therefore, we present fifteen tools, whilst in the rows we have selected two types of evaluation items. The first two groups refer to the objectives of the tool and the specific *i** variants supported. The other two groups of evaluation items are related to technical interoperability, dealing mainly with the type of formats that the tools can write.

It is a crucial observation that different tools are not competing but conform a wide application scenario and, moreover, *i** tools constitutes a mirror which reflects the focus of each *i** variant. For example, the tool TAOM4E [BERT'06], implementing Tropos, can be classified as a CASE tool, focused on (agent-oriented) software design including even code generation; OpenOME, implementing Yu's *i**, is for representing and analysing software requirements including conflict analysis and alternative ways of reaching organizational goals; jUCMNav, implementing GRL/URN, which aims at integrating telecommunications requirements with real time systems specifications (anyhow, the high level of GRL specifications allows modelling other socio-technical contexts). When there is not a specific variant underlying a research group, the result is a general-purpose tool, e.g. REDEPEND-REACT is a plug-in for a popular drawing tool. In spite of this interesting scenario, we have experienced that orchestration of functionalities and even a simple reuse of models are seriously limited and current advances on *i** models interoperability among different tools is exclusively reached by using the interchange format proposed in this thesis.

Although most of the tools declare to import and export XML files, they correspond to different XML schemas which means that an XML file written for some tool A cannot be read by the tool B. In general, existing *i**-based tools and development frameworks have not been capable to interoperate, which prevents taking advantage of existing functionalities. But the main reason is not interchange format, i.e. XML grammar, but the fact that each tool is adhering to a different *i** variant and no means are provided to bridge the gap with others.

Table 2.7. Detailed of purposes and interoperability issues of *i** tools

Support to:	OpenOME	OME	REDEPEND REACT	REDEPEND	TAOM4E	GR-Tool	T-Tool	ST-Tool	J-PRiM	jUCMNav	Snet Tool	DesCARTES	VISIO	I* Prefer	IStar Tool
<i>i*</i> modelling	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes			Yes	Yes	Yes	Yes
Model analysis	Yes		Yes	Yes					Yes			Yes		Yes	
Tropos					Yes							Yes			
Tropos goal reasoning						Yes						Yes			
KAOS and FT							Yes								
Security Analysis								Yes							
(URN) = UCM+GRL										Yes					
Networks scenarios											Yes			Yes	
PRiM methodology			Yes						Yes						
NFR, A UML												Yes		Yes	
<i>i*</i> framework															
Yu95	Yes		Yes	Yes					Yes			Yes		Yes	Yes
Tropos					Yes	Yes	Yes	Yes				Yes			
Secure Tropos								Yes							
Iterative Tropos								Yes				Yes			
GRL										Yes					
Customizable													Yes		
Does the tool allow importing files?															
Text	Yes								No						
XML	Yes		Yes	Yes					No	Yes		Yes	Yes	Yes	Yes
XMI					Yes				No	Yes					
Does the tool allow exporting files?															
XMI					Yes				No	Yes					
Text	Yes	Yes							No						
XML	Yes		Yes	Yes			Yes	Yes	No	Yes		Yes	Yes	Yes	Yes
EXCEL			Yes	Yes					No						
Images	Yes	Yes	Yes	Yes	Yes			Yes	No	Yes		Yes	Yes	Yes	Yes

2.5 Summary of the Chapter

In this chapter we have reviewed the state of interoperability in the *i** community. The main focus of course is on model interoperability, so that we have focused on language definition, but we have also paid attention to tool interoperability, for the practical implications behind. Our descriptive study has considered an historical review of the language, the analysis of its variants and metamodels behind, and a panoramic on tools. In general, the scenario is not optimistic. We have shown that: 1) there are variants with different genesis and aims; 2) new variants emerge continuously; 3) research challenges about integration are not normally addressed. Moreover, we have shown that there are different metamodels even related to the same variant. Last, the different aim of the tools is a motivation to interoperate, however, the problem is that they are not based on the same set of concepts.

Contemporary RE and especially GORE supports the idea of *understanding why*, in the sense of social concerns which guide a given solution. To *understand why* means to have a theoretical model about what is happening in the *i** community: Why are there different variants? What are the bases of these phenomena? Will it stop? What can be expected from an interoperability approach? Is interoperability meaningful in this prolific scenario of variants? Will an interoperability approach survive this scenario? Hence, what should be the theoretical background to confront this problem? Of course, we can tackle the interoperability problem from a technical point of view, but *understanding why* mandates us to look beyond technical concerns. Therefore, our theoretical background will include social concerns about sociology of science, human communication, technical communication as an extension of human communication, and interoperability. This is the aim of the next chapter.

CHAPTER 3. Theoretical Approach to an Interoperability Framework for *i Language Variants**

3. Theoretical Approach to an Interoperability Framework for *i** Language Variants

In this chapter, we briefly review the theoretical frameworks related to our research. The goal of the thesis is to propose an interoperability framework for those *i** language variants used by the *i** community. Therefore, we will cover first technical concerns referring to model and tool interoperability and their related fundamentals concepts, such as syntax, semantics, and the role of ontologies in it. Nevertheless, beyond technical concerns, we will review social theories related to human communication and, particularly, scientific communication which allow us to have (at least) a reasonable explanation about why the *i** language variants have evolved this way and what it may be considered feasible to propose an interoperability framework. Finally, we will finish this chapter showing the theoretical approach to reach the thesis's goal.

3.1 Conceptual Framework

3.1.1 Interoperability

Interoperability is widely understood as “*a property referring to the ability of diverse systems and organizations to work together (inter-operate)*” [WIKP'11]. In a more accurate way as a property of “*computer systems or software to exchange and make use of information*” [OXFO'11]. This definition is interesting because it goes beyond data interchange; it implies pragmatic issues: to make use or to work together implies human behaviour. In our case, *i** model interchange is a means not just for sharing models and perspectives of analyses. Given the diversity of tools and their purposes, for sure enabling interoperability means to use algorithms and available transformations among tools. Therefore, it would be a proper way of *work together*.

Since the focus of *i** is socio-technical modelling, we have selected TolK's interoperability framework [TOLK'03] which adds concepts that are closely related to dealing with technical problems in a social context, e.g. usage, ontology, and process, among others. Fig. 3.1 shows the five levels of this model. After a first inspection we

can conclude that the current interoperability scenario in the i^* community is at level 0. To be at level 1 would mean, under a model interoperability point of view, that we have models represented under the same language (at data level it means to have a common interchange protocol). However considering the previously described semantic differences among i^* variants, even in the case of having an interchange protocol, it does not seem feasible to reach neither level 4, nor level 3, nor even level 2, because the failing requirement of a common ontology acts as a barrier to reach advanced levels of interoperability.

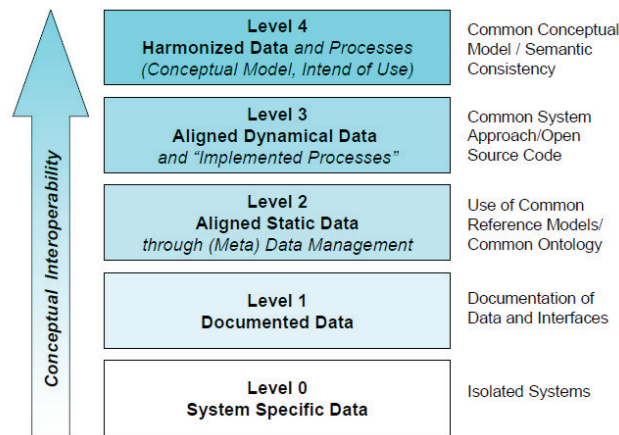


Fig. 3.1. Levels of conceptual interoperability (LCIM) from [TOLK'03]

In order to illustrate the interoperability limitations we review four specific types of asymmetries, which constitute a barrier to reach level 2 in TolK's classification:

Structural asymmetries. It happens when different structures describe the same concept. For example, some i^* metamodels describe intentional elements as enumeration types whilst in other there is a different class to represent each of them.

Heterogeneous asymmetries. It happens when different description methodologies are used, for example we have Telos, BNF, MOF and UML metamodels to describe i^* variants.

Descriptive asymmetries. It happens when we have homonyms, synonyms or different attributes for the same concept, e.g. a dependency with different type of values, e.g. trust, or permission on one variant and only delegation or general dependency in others.

Semantic asymmetries. The concepts of the different schemata do not match exactly due to overlapping or subsetting. For example the concept of softgoal on REF i^* 's variation [DONZ'02] [DONZ'04] excludes the representation of constraints and proposes an extra symbol to differentiate them. Thus, a REF's constraint includes functional and non-functional requirements that are expressed as concrete requisites. Conversely, non-functional requirements are represented by softgoals and there are no differences among generic and concrete non-functional requirements. Therefore, there is the same (softgoal) symbol, but it involves a different concept on REF than on i^* .

Although the first two kinds of asymmetries could be approached by proposing equivalences between existing metamodels, the other two are fundamental and may prevent interoperability, requiring further analysis. Therefore, we include a review of the related theories that may support this analysis, namely theoretical frameworks for ontologies and semantics, which would help us to justify an interoperability approach.

3.1.2 The Role of Ontologies in Semantics

Among the diverse approaches to semantics available in the literature, we have selected the framework formulated by the contemporary philosopher Mario Bunge [BUNG'08]. The reasons are:

- (a) the aim of Bunge's proposal is to constitute a semantics of science, i.e., what researchers are referring to, which seems proper to us because the different i^* metamodels are representations of a world that the i^* researchers are attempting to refer;
- (b) Bunge's focus are symbolic languages like the ones used in mathematics, sciences and philosophy, therefore it also seems to be applicable for i^* analysis;
- (c) he includes the concept of semantics and ontology as part of the basic definitions as also Tolk's interoperability levels do;

- (d) Bunge clearly separates the semantic references from the truth values of expressions. In other words, in this thesis it is out of scope to know if an i^* goal of a specific i^* actor is really a goal of this actor. The interest is just to enable interoperability therefore truth values are not a matter of question as well as in Bunge's framework;
- (e) Bunge offers a tuple-based algebraic formalization quite similar to traditional formal proposals from computer sciences, e.g. [ENGE'74;HOPC'69;McNA'82];
- (f) Bunge includes as part of his proposal the existence of modelling (representational) languages and discusses about semantics applied to this kind of representations;
- (g) the conceptualization of representations concurs with the engineering perspective of [SMIT'93] about seeing engineering design as a way of producing representations on modelling languages.

The definition of Language given by Bunge is: $\mathcal{L}_{\mathcal{K}} = \langle \Sigma, \square, \circ, \xi, \tau, \Omega, \Delta \rangle$. Although it is possible to go deeper into the detailed properties of each element, they will be explained just by using common computer science concepts: Σ is the alphabet; \square represents the words separator (white space); \circ represents the concatenation operator. As usual, words on Σ are included in Σ^* and composed words (phrases) are in Σ^{**} . ξ is a collection of applications which produce the complete phrases or well formed formulas (wff) of $\mathcal{L}_{\mathcal{K}}$. τ from Σ^{**} to Σ^{**} , is called the transformation device which transforms wffs into others, i.e. like translators of equivalence phrases. Bunge called this part of the language the *syntactic part*.

The *semantic part* is given by the elements Ω and Δ . Ω represents the set of objects and Δ represents the decoding function, which is a many-to-one function from Σ^{**} to $\mathcal{P}(\Omega)$, the power set over Ω . Δ takes a sentence from $\mathcal{L}_{\mathcal{K}}$ and gives to it an object or a set of objects. These objects, in words of Bunge, can be anything, e.g. individuals, sets, concrete or abstract relationships, possible or impossible facts, linguistic elements, etc. The role of Δ is to inject a meaning to an expression.

Bunge adds that a complete language specification should include its pragmatic component. Therefore the pragmatics of $\mathcal{L}_{\mathcal{K}}$ can be represented as an application $\prod_{\mathcal{K}}$ from the set Σ^* of expressions (Bunge says with or without sense) to a set of

behavioural elements of the members of \mathcal{K} , who are the users of the language $\mathcal{L}_{\mathcal{K}}$. That is a relevant concept because the Language is defined for a given community, which also corresponds with the case of i^* . We will not include this part on the model in order to avoid too much complexity; however, for practical issues, we will interpret that applying transformation devices (τ) are part of the behaviours of \mathcal{K} .

Bunge proposes a formalization for representations, i.e. expressions in a scientific language, in the same sense of engineering models from [SMIT'93], i.e. expressions in modelling languages. The reference appears as a relationship \mathcal{R} which is valid between constructs (C) and objects of any kind (Ω). Basically, a construct is a human construction in the sense of concept which has concrete manifestations on “reality”, i.e. on Ω . Therefore \mathcal{R} is a set of pairs, by convention the graph³ $\mathcal{E}(\mathcal{R}) \subseteq C \times \Omega$ with $C \subset \Omega$. Therefore, he introduces a variation to the classic semiotic proposition from Pierce [ECO'00], where a symbol is defined in terms of a sign (as mark), an object and an *interpretant* (explanation in the same language level). In the case of Bunge’s proposition a construct replaces the interpretant, thus the function \mathcal{D} maps language’s signs to constructs. Therefore a symbol is a meaningful sign (left part of Fig. 3.2). In order to explain the relationship between Δ , \mathcal{D} and \mathcal{R} the formalization includes the relation $\underline{\Delta}$, which is called the *factual representation*, i.e. $\mathcal{E}(\underline{\Delta}) \subseteq C \times F$, where F are the facts that belongs to Ω . Therefore the above illustration is extended by considering representations. The result is reproduced in the right part of Fig. 3.2.

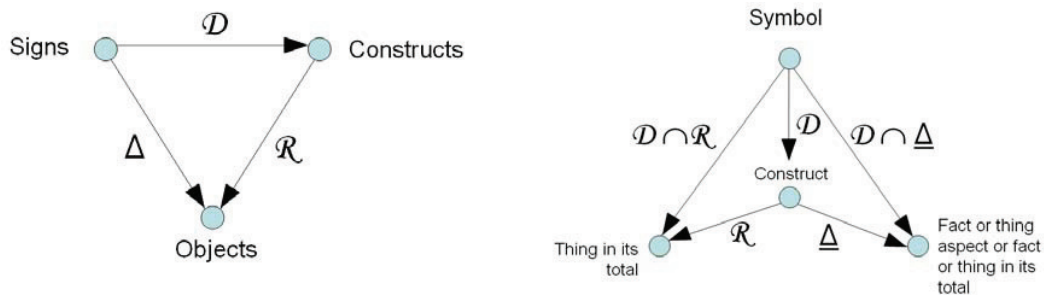


Fig. 3.2. Denotation and reference and its extension to representational languages

³ It is necessary to clarify the operator \mathcal{E} in the expression $\mathcal{E}(\mathcal{R})$. \mathcal{E} comes from the concept of *extension* and Bunge interchange *extension* and *graph* many times. Lately he discusses about different interpretations of “sense” of a concept. One of the senses is the *extension* sense, which identifies the individuals which exist in a particular set. In a general sense the relationship is the edge (e.g. means) and the constructs and objects are nodes.

In this case, the symbol means something (expressed at its construct) which represents a (probably) partial view of existing objects. For example, in a specific i^* variant, it would be a particular goal of an actor in a given organizational unit. At the same time the complete model can represent a partial view of a reality, e.g. an i^* strategic dependency model.

Therefore it is possible to see that Bunge's conceptual framework of representational languages is applicable to i^* models because the co-domains of the functions Δ and $\underline{\Delta}$ are the power set coming from the interpreter's ontology, therefore it is necessary to know what kind of objects exist in the world and it is exactly what an ontology does. For example, in the "real world", we can have some stakeholder statement about efficiency that, depending on the engineer's ontology (coming from the corresponding i^* variant), would be modelled as a softgoal, or as some constraints if the variant have the corresponding language construct. Therefore, the difference is not in the "real world", the difference is on how the requirement engineer distinguishes the elements of the world, i.e. the ontology that he/she applies.

In general, the term "ontology" has two main interpretations: as a discipline and secondly as a conceptual artefact. Bunge treats the concept mainly with the second meaning; he exemplifies the concept in the ontology of the electrodynamics theory, which assumes that the world is conformed by bodies and electromagnetic fields. We do not go deep in the Computer Science meaning; we want just to remark that a very similar concept is that of representational ontology in [STUD'98]. Representational ontologies are very interesting because they accomplish the philosophical principle that an ontology is previous to experience. It has been put in the metaphor *the eyes which we see the world*. Therefore, it is mainly a cognitive tool to match objects from "reality" to concepts.

Consequently, Bunge's framework offers a different perspective on i^* variants distinguishing additional elements involved on the sharing of models and operations (i.e. on interoperability issues), such as symbols, language constructs, denotations, mapping functions to reality, and interpreters' ontology. A relevant concept here is the

concept of language $\mathcal{L}_{\mathcal{K}}$ of a given community \mathcal{K} . Specifically, under this framework, models can be shared (subset of phrases on Σ^{**} , e.g. SD diagrams) and also operations, e.g. metric algorithms existing in τ . If for all phrases $\underline{x}_{\mathcal{A}}$ in $\Sigma_{\mathcal{A}}^{**}$ it exists $\underline{\Delta}_{\mathcal{B}}(\underline{x}_{\mathcal{A}})$ (i.e. expressions $x_{\mathcal{A}}$ of some community \mathcal{A} have a meaning for the community \mathcal{B}) then we can reach the harmonized data level of Tolk (level 3). However it is easy to see that if $\Sigma_{\mathcal{A}} \cap \Sigma_{\mathcal{B}} \neq \phi$, there will be common phrases on \mathcal{A} and \mathcal{B} , i.e. $\exists x / x \in \Sigma_{\mathcal{A}}^{**} \wedge x \in \Sigma_{\mathcal{B}}^{**}$. To use an operation $\tau_{\mathcal{A}}$ in \mathcal{B} means that it exists $\underline{\Delta}_{\mathcal{B}}(\tau_{\mathcal{A}}(x_{\mathcal{B}}))$ i.e. the community \mathcal{B} has a meaning to the transformation $\tau_{\mathcal{A}}$ applied to some of their phrases ($x_{\mathcal{B}}$), thus $\tau_{\mathcal{A}}$ can be a shared resource between communities \mathcal{A} and \mathcal{B} . Now, can we sustain that $\underline{\Delta}_{\mathcal{B}}(\tau_{\mathcal{A}}(x_{\mathcal{B}}))$ requires that $\tau_{\mathcal{A}}(x_{\mathcal{B}}) \in \Sigma_{\mathcal{A}}^{**}$? We have a negative answer to this, for example a goal analysis proposed for some specific i^* variant \mathcal{A} can produce a result for a model m coming from some other i^* variant \mathcal{B} even if $m \notin \Sigma_{\mathcal{A}}^{**}$. Thus, we can affirm that, in Bunge's framework, it is possible to share operations (Tolk's interoperability at level 4) and, at the same time, not having harmonized data (Tolk's interoperability at level 3). It is more than using the transformation on the part of m that is "understood" in \mathcal{A} . The fact that the transformations $\underline{\Delta}_{\mathcal{B}}$ and $\underline{\Delta}_{\mathcal{A}}$ are established on different Ω opens the interoperability space to a different meaning-making process in \mathcal{A} . In communication theory this phenomena is called appropriation [MANS'04;MART'06]. Therefore, we sustain that it is already represented in Bunge's framework constituting a valid perspective for analysing i^* communities. It is a relevant assert for our thesis because it means that a high level of interoperability can be reached even if lower interoperability Tolk's levels are not accomplished.

3.1.3 A Formalization of the Interoperability Problem

Using Bunge's Framework the interoperability problem may be stated as the translation of a model $m_{\mathcal{A}}$ into a model $m_{\mathcal{B}}$. However a model m is in the context of a community \mathcal{K} which has been produced with a modelling language $\mathcal{L}_{\mathcal{K}}$ (e.g. an i^* variant). Thus, expressions on $\mathcal{L}_{\mathcal{K}}$ (models) represent situations (or "things") constructed using the objects ($\Omega_{\mathcal{K}}$) that \mathcal{K} believes that exist in "reality", i.e. the \mathcal{K} 's ontology. Therefore the translation between models from the community \mathcal{A} to the community \mathcal{B} can be represented by a mapping function $\varphi_{\mathcal{A}\mathcal{B}}: \langle \mathcal{L}_{\mathcal{A}}, \Omega_{\mathcal{A}} \rangle \rightarrow \langle \mathcal{L}_{\mathcal{B}}, \Omega_{\mathcal{B}} \rangle$, $\varphi_{\mathcal{A}\mathcal{B}}(m_{\mathcal{A}}) = m_{\mathcal{B}}$. However, it is necessary that the meaning remains unchanged, i.e. $\underline{\Delta}_{\mathcal{A}}(m_{\mathcal{A}}) = \underline{\Delta}_{\mathcal{B}}(m_{\mathcal{B}})$, which implies that $\Omega_{\mathcal{A}} \cap \Omega_{\mathcal{B}} \neq \emptyset$ given that the corresponding co-domains of the decoding functions $\underline{\Delta}_{\mathcal{A}}$ and $\underline{\Delta}_{\mathcal{B}}$ are $\Omega_{\mathcal{A}}$ and $\Omega_{\mathcal{B}}$. Note that if there is, at least, a family of models in \mathcal{A} which can be shared with \mathcal{B} then \mathcal{A} and \mathcal{B} can interchange them, therefore $\Omega_{\mathcal{A}} \cap \Omega_{\mathcal{B}} \neq \emptyset$, but it does not necessary implies $\Omega_{\mathcal{A}} = \Omega_{\mathcal{B}}$. Therefore, under this framework we can sustain that model interchange is partially possible even when $\Omega_{\mathcal{A}} \neq \Omega_{\mathcal{B}}$, which would be interpreted as a continuous extension to the discrete steps presented in Tolk's interoperability framework. Here for example, having the same ontology (i.e. $\Omega_{\mathcal{A}} = \Omega_{\mathcal{B}}$) is a requisite to e.g. having the same *Intend of Use*. We sustain that common parts between $\Omega_{\mathcal{A}}$ and $\Omega_{\mathcal{B}}$ enable highest interoperability levels, even if some mappings cannot be computed.

The problem arises with those constructs in $m_{\mathcal{A}}$ that are not representable under $\mathcal{L}_{\mathcal{B}}$ because the interpretation (i.e. the decoding function) uses objects from $\Omega_{\mathcal{A}}$ which do not have corresponding objects on $\Omega_{\mathcal{B}}$. It means that communities \mathcal{A} and \mathcal{B} share only a portion of syntax and semantic elements of the languages $\mathcal{L}_{\mathcal{A}}$ and $\mathcal{L}_{\mathcal{B}}$. In the rest of this chapter, we will review some theories which show how other theories present the communication or interoperation under polysemantic scenarios. We will gather ideas from these theories to formulate our interoperability approach.

3.2 Social Theoretical Framework

In this section, we briefly review some system and social-based theories regarding the social context of our interoperability problem. Either as a research approach or as an RE approach, it is necessary to understand why, or at least to have an approach to understand why. Due to the context of our interoperability problem, i.e. a communication problem inside a scientific community, we looked for some answers in sociology of science, but also, in general theory of systems, particularly in cybernetics and linguistics, specifically semiotics. We present here these theoretical contributions that offer us reasonable explanations to the existence of i^* variants. Besides, these theories provide us some ideas and guidelines to approach an interoperability solution.

3.2.1 Contributions from Sociology of Science

Sociology of Science embraces those studies that explain the foundations of science (i.e., Philosophy of Science) from a sociological perspective. We found again Thomas Kuhn as one of the most relevant contemporary philosophers. Our interest here in Kuhn's work has to be with his modelling of science evolution through discontinuous jumps [KUHN'96]. Kuhn called these jumps "scientific revolutions", which are led by a reduced set of pioneers who change the basic conceptualizations and build a new way of doing science. This new way of perceiving and researching is called "scientific paradigm". Each paradigm has an initial and underlying ontology, which is "known" (in the beginning) only by the pioneers, who try to communicate it and teach the new way of doing research using this ontology. The impossibility of explaining the new ontology based on the previous ontology is called by Kuhn "the incommensurability problem". In spite of that, along time, the ontology is spread, and used, not always as it was proposed but according to the particular interpretations of the followers. This phase is called "the revolutionary stage". However, at some moment, the conceptualization converges to a stable and shared ontology (the key concepts of the paradigms) and epistemology (what the community accept as valid methods for knowledge production). When it happens, the following phase, the normal-science stage, starts. The cycle continues when theoretical anomalies appear (unsatisfactory explanations) and a new pre-revolutionary stage germinates. Although the original Kuhn's idea was to explain

big scientific movements, lately he recognized that there are a lot of small and even micro-revolutions which present the same behaviour. Fig. 3.3 illustrates the stages.

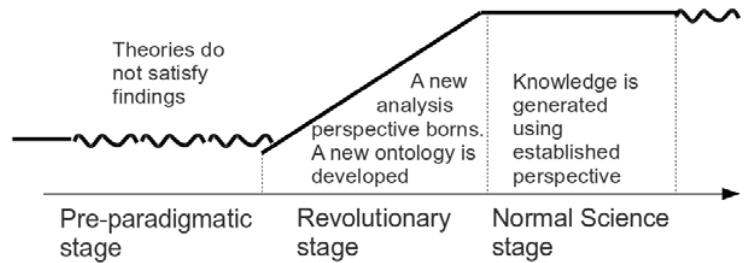


Fig. 3.3. Phases of Kuhn's paradigmatic view of scientific progress

Bourdieu's theory of scientific fields [BOUR'75] is the second approach from sociology of science that we have reviewed. In this theory, the key concept is that of symbolic capital. Scientific behaviour is associated to fields that have, in their centre, the highest concentration of symbolic capital; normally the leaders/pioneers of the fields occupy these positions. They dominate the concepts and try to spread their ideas. Scientifics try to maximize their symbolic capital in two ways: moving to the centre of the field, which means to exactly follow the pioneers' ideas and collaborate with them; or generating new scientific fields by the intersection with other fields that allow them occupying the centre of a new scientific field. Either in the zones of lower symbolic capital or in the new fields, the concepts are not used as in the centre of the reference scientific field. In Fig. 3.4, the idea of scientific fields is illustrated.

It is relevant for this work that Bourdieu's theory coincides with the idea that there is an elusive ontology at the starting point of a scientific field. This ontology is better known and understood by who have proposed it. At the beginning, some doubts came out concerning a relationship between Kuhn's and Bourdieu's theories. On the one hand, Kuhn talked about scientific revolutions as big and relevant scientific movements; on the other hand, Bourdieu's fields appear like common and almost a daily scientific behaviour. However, at the prologue of the last edition of Kuhn's traditional book [KUHN'96], he puts in relative terms how big a scientific revolution could be and he even talks about micro revolutions which keep the already described Bourdieu's behaviour.

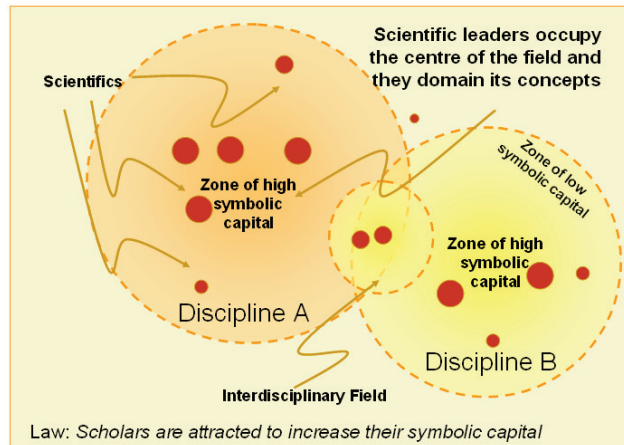


Fig. 3.4. Bourdieu's idea of scientific fields

In both theories, it is recognized the fact that research activity without a fully shared ontology, as it happens in the i^* community, is feasible. In addition, from these two theories, we cannot expect the paradigms (goal-orientation leaded by the i^* framework) be extended over time or the i^* field be kept the same way in the centre and in the very far periphery. A point of difference between these two approaches is the position about community agreement on a shared ontology. From Kuhn, it is predicted that having a shared ontology stops the revolutionary stage, whilst from Bourdieu, it is predicted that we will always have uncontrolled interpretations and uses. However, this apparent contraction is not real if we suppose that having static scientific fields would be part of Kuhn's "normal science".

3.2.2 Contribution from Cybernetics

Cybernetics has been conceptualized as the General Theory of Control [BEER'67]. We can pay attention to both classical and new cybernetic conceptual frameworks. A classic contributor has been Ross Ashby, who proposed a definition of intelligence from the control perspective [ASHB'57]. Thus, intelligence is understood as a repertoire of behaviours; therefore having more intelligence or variability means having a broader repertoire of behaviours. In this theory, it is said that humans are able to create intelligence amplifiers in order to enlarge their control capabilities, which means to increase variability. One of Ashby's examples is the difference between the ships being loaded quickly and easily by movements of a control handle, or slowly and laboriously by hand [ASHB'57]. Other intelligence amplifiers can be a dictionary, spectacles, a calculator and of course, a modelling tool, because they improve the repertoire of behaviours.

In addition, contemporary cybernetics takes concepts as *autopoiesis* [MATU'98] to explain that biological-based systems continuously regenerate the processes that produce them (*autopoiesis*). This should be understood from both an internal point of view (transformations) and as an external one (interactions). It is said that biological systems are operationally closed systems that are self-produced and self-referred. It means that their actions are the effect of their interpretations (meaning-making process). It also implies that operational distinctions (based on its ontology) that use a system in order to guide its interactions and transformations are not observable since they are internal to the system. Therefore, a biologically-based communication process emerges without an explicit (non-external and non-shared) ontology. As an extension of that, and due to the fact that intelligence amplifiers (e.g., modelling tools) are part of the interactions of the system with its environment, then interoperability will take place if the meaning-making process produces some interpretation (e.g., for an arriving model) which improve variability.

3.2.3 Contributions from Semiotics

Semiotics is about the interpretation of signs, syntax, semantics and pragmatics. Its main focus is the study of models that explain human communication from both individual and collective perspectives. What is an interesting point for us are collective perspectives because they may model communication in scientific communities. A relevant semiotic concept, which changes depending on the community, is that of language expressions and their meanings. Semiotics considers that some language expressions can reach stable meanings in the natural dynamic of meaning systems (at least to a wide set of people) which implies that these expressions can be used as interpretants, i.e. sentences that explain other meanings. This is why an explanation might be completely understood by some people meanwhile some others will have a different conception about it or partially get what it means.

Yuri Lotman [LOTM'03] introduced the concept of semiosphere to explain communication inside medieval human cultures. Firstly, he expressed that mono-semantic systems do not exist in isolation. These related systems are part of a continuous sphere of meaning called semiosphere. Lotman explains that the boundary is the area of accelerated semiotic process (interpretations). This theoretical approach affirms that in peripheral areas, where structures are “slippery”, less organized and more flexible, the dynamic process meets with less opposition and, consequently, develops more quickly. Then, one may say that the new semiosphere grows leaving in the centre the dominant semiotic system constituted by a wide set of stable concepts. This theory seems highly applicable to the i^* community as part of a more general software engineering community and also for a particular i^* variant community. From this perspective, it can be easily understood the sentence from Lotman [LOTM'03] saying that the creation of meta-structural self-descriptors (grammars) appears to be a factor which dramatically increases the rigidity of the semiosphere’s structure and slows down its development. For example, developing a collective standard will reduce the space of models’ interpretations.

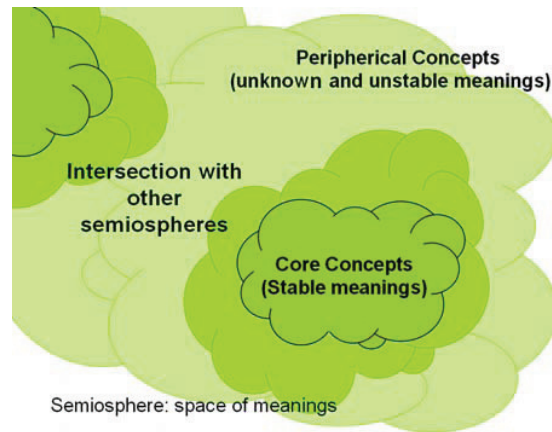


Fig. 3.5. Lotman's idea of semiospheres

3.3 Approaching a Theoretical Framework for i* Model Interoperability

On the light of the reviewed theories we want to address some initial assumptions or beliefs that were conceptualized as problems. To establish contra-argumentations, we present these theoretical problems confronting them with contrary facts derived from reviewed theories.

Initial problem:

According to Tolk, high levels of interoperability require a common and shared ontology.

Contra factual theoretical contributions:

From cybernetics, and specifically from Maturana's theory, an ontology is always internal. It is the artefact for the closed process of meaning-making. Therefore, communication and, of course interoperability, cannot depend of making explicit something that always has been internal.

From semiotics, and specifically from Lotman's theory, polysemantic contexts exist, and in these contexts, the meaning-making process is done in terms of more stable concepts.

From sociology of science, specifically from Kuhn's theory, in a paradigmatic movement, the underlying ontology evolves. In spite of that, there is communication and stability comes.

Implications for a solution approach:

If there are more stable concepts then we can use them as references in order to facilitate interpretation, e.g. in an i^* language variation we can have the new concept of *support* and it does not mean anything, but if we express that it is an actor link, then the scope of possible meanings is highly reduced. Therefore, if we generate a structure which includes a relationship to stable i^* concepts, then the interpretation will be facilitated.

Initial problem:

Derived from Tolk's framework, it will not possible to reach an interoperability approach while semantic variability around i^* persists.

Contra factual theoretical contributions:

From semiotics, semantic variability has a specific behaviour, like semiospheres. A particular case of semiosphere seems to be that of scientific fields (from sociology of science), where stable concepts lay in the centre, which means that there is not semantic stability on the periphery, in spite of that some kind of communication can be reached.

Implications for a solution approach:

The solution approach ideally should keep the shape of a scientific field or semiosphere, a relevant feature will be differentiating core concepts from intermediate and peripheral ones.

Initial problem:

There are always different research group proposing new or extending existing i^* language constructs, therefore it is impossible to propose an interoperability approach that includes all of them.

Contra factual theoretical contributions:

From sociology of sciences (Kuhn), new proposals around i^* are part of the same revolutionary scientific stage, which means that there is a unique underlying ontology which evolves into a stable one.

From semiotics, there are common concepts and new concepts based on the stable ones. Besides, the stage will stop at some moment (as any other scientific revolutionary stage does) and a common semantic reference will be expanded and adopted.

Implications for a solution approach:

The solution approach should include stable i^* references in order to facilitate interpretation. Therefore it should consider possible different meanings (interpretations) and different levels of detail should be provided to allow that.

Therefore, our basic proposal is to design an interoperability language which has the form of the i^* semiosphere, it means a kernel with the stable concepts, intermediate levels with flexible concepts and, also, has the possibility to represent “lazy” (term coming from Lotman) i^* concepts or even external concepts. In this map, the i^* variants should be included because if they are i^* variants then a relevant part of their stable concepts are located in the same place of the i^* semiosphere. In Fig. 3.6 we illustrate this idea.

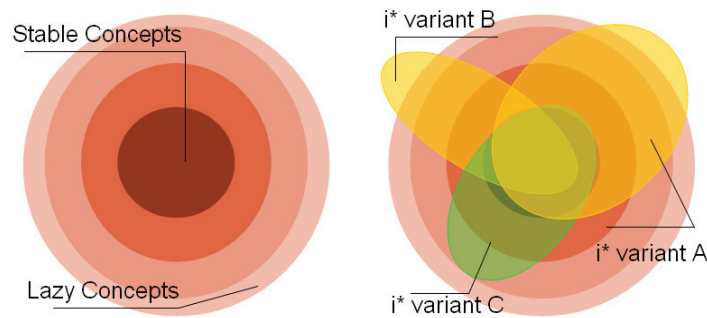


Fig. 3.6. The structure of the i^* 's semiosphere (left) and the overlapping of i^* variants (right)

Therefore, if we formulate an interchange approach which fits this shape, we could state that any model belonging to any i^* language variant will be somewhere in the semiosphere, therefore it would be representable in the interchange proposal. However there are some semantic problems that we will describe in terms of the Bunge's framework. When we have two variants, let's say A and B, we have also two decoding functions, two spaces of “real objects” (ontologies) and therefore two spaces of interpretations. However, coming from the same semiospheres, the situation is illustrated in Fig. 3.7. It means that there is a part that does not need any translation ($\underline{\Delta}_A$), a part that can be interpreted because there is an explanation in terms of known structures ($\underline{\Delta}'_A$), and a part that is not possible to translate because there is not a corresponding element in $\mathcal{P}(\Omega_A)$ ($\underline{\Delta}_B$). Therefore, we need some theoretical approach which allows, ideally in a formal way, categorizing and qualifying this situation, which is unavoidable if the translation occurs in a polysemantic context.

An adequate approach to handle this situation has been formulated by Guido Wachsmuth [WACH'07]. Wachsmuth defines different semantic-preserving categories and matches them with specific refactoring operations on metamodels. The way of

handling semantic-preserving features respond to changes on corresponding metamodels, which we can match to ontologies. In Fig. 3.8, we establish the analogy between the concepts from Wachsmuth’s proposal (left) and our interpretation to the model translation problem (right).

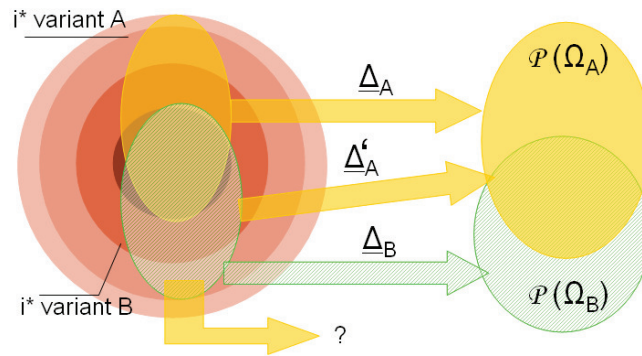


Fig. 3.7. Translation cases between two i^* variants by denotation functions

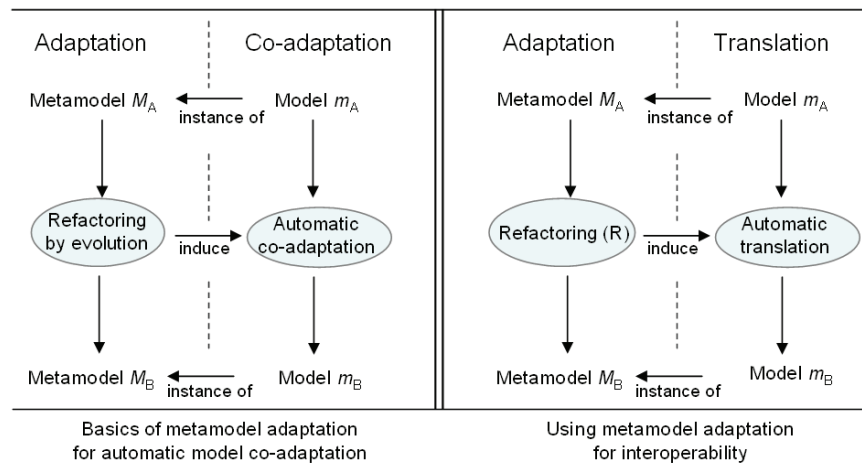


Fig. 3.8. Comparative between co-adaptation and interoperability via metamodel refactoring.

To characterize refactoring operations Wachsmuth proposes some basic concepts:

- \mathcal{M}_M represents all the metamodels conforming to a specific metamodel formalism M , denoted by $\mathcal{M}_M := \{\mu \models M\}$.
- $C_M(\mu)$ represents the concepts defined by a particular metamodel μ . In our case, typical concepts would be *actor*, *intentional element*, etc.

- $I(\mu)$ represents the set of all metamodel instances conforming to a metamodel μ , denoted by $I(\mu) := \{ i \models \mu \}$. In our case, we focus on those μ which are a metamodel of some i^* variation and then for each μ , $I(\mu)$ are i^* models built as instances of μ .
- $I_C(\mu)$ represents the set of instances $I(\mu)$ of restricted the specific set of concepts C , i.e., $I_C(\mu) \subseteq I(\mu)$. For instance, we may refer to the set of concepts C which are part of SD models, and then $I_C(\mu)$ would represent SD models built according to the metamodel μ .

Using these concepts, five types of generic relationships between metamodels are defined (see 1st and 2nd columns in Fig. 3.9) which yield to five degrees of semantic preservation. The transformation from one metamodel to another implies a relationship R between the source and target metamodels, thus, the type of semantic preservation of R (if any) will depend of which of these generic relationships is subset (see 3rd column in Fig. 3.9). Besides, the different types of semantic preservation imply different types of instance preservation (see 4th column in Fig. 3.9). Therefore, we have a way of characterizing semantic preservation which also can inform us if the type of modification that suffer the translated model (instance-preservation).

Relation	Definition	Semantics-preserving (s-p)	Instance-preserving (i-p)
Equivalence	$\mu_1 \equiv \mu_2$ iff $I(\mu_1) = I(\mu_2)$	$R \subseteq \equiv$ is strictly s-p	is strictly i-p
Variant modulo φ	$\mu_1 \equiv_{\varphi} \mu_2$ iff $\Phi: I(\mu_1) \rightarrow I(\mu_2)$ is a bijective function	$R \subseteq \equiv_{\varphi}$ is s-p modulo variation	is i-p modulo variation
Submetamodel / Supermetamodel	$\mu_1 \subseteq \mu_2$ iff $C(\mu_1) \subseteq C(\mu_2)$ $I(\mu_1) = I_{C(\mu_1)}(\mu_2)$	$R \subseteq \subseteq$ is introducing s-p $R \subseteq \supseteq^{-1}$ is eliminating s-p	is strictly i-p is partially i-p modulo variation
Enrichment	$\mu_1 \supseteq \mu_2$ iff $C(\mu_1) = C(\mu_2)$ $I(\mu_1) \subseteq I(\mu_2)$	$R \subseteq \supseteq$ is increasing s-p $R \subseteq \subseteq^{-1}$ is decreasing s-p	is strictly i-p is partially i-p modulo variation
Extension	$\mu_1 \subseteq_{\varphi} \mu_2$ iff $C(\mu_1) = C(\mu_2)$ $\Phi: I(\mu_1) \rightarrow I(\mu_2)$ is an injective function	$R \subseteq \subseteq_{\varphi}$ is increasing modulo variation $R \subseteq \supseteq_{\varphi}^{-1}$ is decreasing modulo variation	is i-p modulo variation is partially i-p modulo variation

Fig. 3.9. Summary of semantic preserving relationships in Wachsmuth's framework.

Now, if we see the definition of supermetamodel (row 3 in Fig. 3.9), and we assume that a model, named the i^* supermetamodel, exists, therefore any existing metamodel of an i^* variation is a submetamodel of the i^* supermetamodel. Therefore, at least theoretically, if we model the i^* semiosphere, we will find an i^* supermetamodel. Therefore, following Wachsmuth, if we could model refactoring operations from the i^*

supermetamodel to the particular variants, then we would have a feasible translation from each variant to another. This hypothetical scenario would exhibit three advantages: (i) supporting at some extent interoperability between models belonging to different metamodels; (ii) given a number k of i^* variants, providing a framework that offers translation from one variant to each other with linear complexity in terms of transformation functions (k functions) instead of quadratic (k^2-k pair-wise functions); (iii) the type of semantic preservation would be characterized with a clear specification of preservation (strict, modulo variation, increasing or decreasing). In Fig. 3.10 we illustrate this assumption, in the left-hand side it appears the case of a quadratic amount of translations, in the centre appears the case of having a supermetamodel and a linear amount of translations. In the right-hand side is illustrated the fact that having an i^* supermetamodel (dark circle) means that any i^* variant can be represented in such metamodel, therefore, the transformation can be characterized using Wachsmuth's semantic preservation concepts.

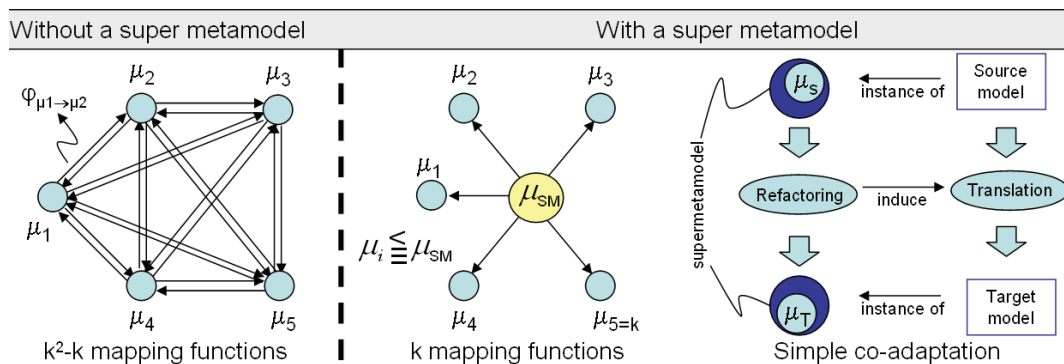


Fig. 3.10. Comparing absence and presence of an i^* supermetamodel for model translations

Summarizing, we have reached a theoretical position and theoretical approach to an interoperability framework for i^* variants interoperability. We can say that this approach has a semiosphere shape, allows handling a polysemantic context, assumes the existence of an i^* supermetamodel and, additionally, includes a re-interpretation of a formal approach to characterize semantic preservation for i^* model translations problem. Maybe, the most relevant point is that, at least theoretically speaking, the interoperability approach has linear complexity instead of quadratic complexity. In the next chapter we add a set of specific requirements to a technical solution approach which has the objective of implementing this theoretical approach.

CHAPTER 4. iStarML: A Proposal for *i** Model Interoperability

4. iStarML: A Proposal for i^* Model interoperability

4.1 *Presentation*

In this chapter, the XML-based iStarML model interchange proposal is presented. It is rooted on the theoretical background introduced in the previous chapter by proposing a set of requirements that require features to facilitate its use, usage and usefulness. Considering these requirements, we proposed the basic structure of iStarML; then we review its compliance with the theoretical background, and next we show the basic language constructs of this interchange proposal. Last, two representational samples are shown and, using these samples, we tackle the topic of semantic loss proposing an algorithm to categorize semantic preservation.

4.2 *Requirements for an i^* model interchange proposal*

At this point, the existing problem of interoperability among i^* language variants has been formulated. Following the theoretical analysis, a set of requirements to the interoperability approach is proposed. We represent this set of requirements using i^* itself (see Fig. 4.1). The root of analysis is to reach a community-wide interchange language for i^* models interoperability. We have elaborated a top-down analysis using our theoretical approach. The explanation of the non-functional requirements and their relationship to the reviewed theories follows.

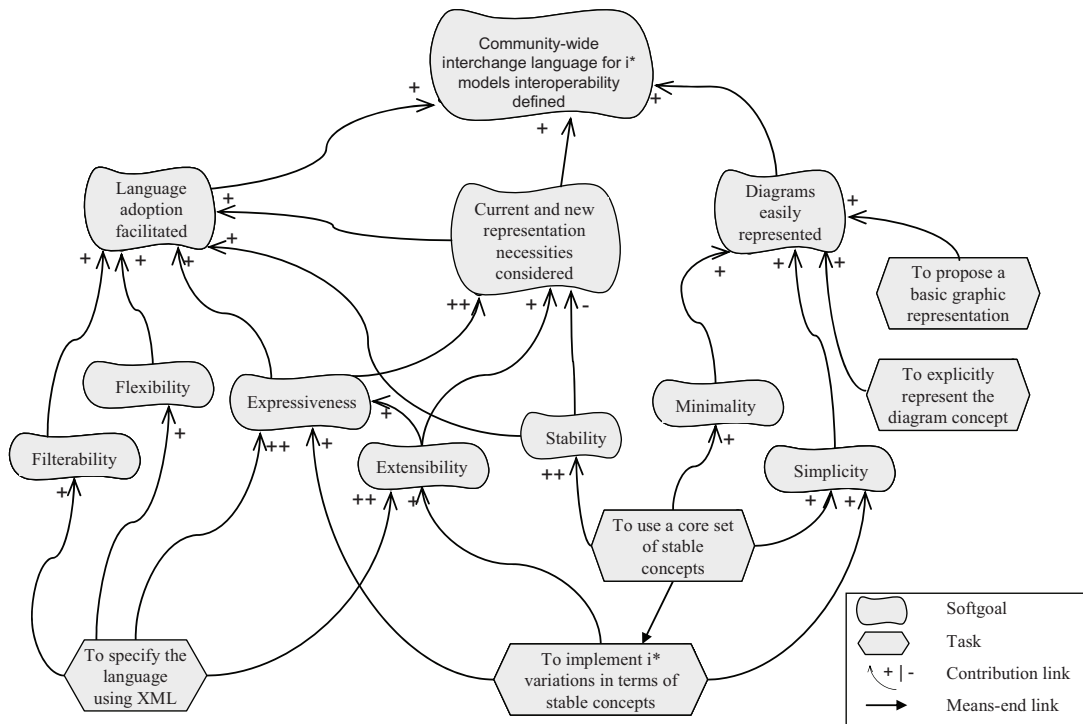


Fig. 4.1. *i** requirements representation for an interoperability proposal

Expressiveness. The model interchange language shall allow the representation, at least, of the most known variants of the *i** language and, also, the design of other language variants.

Theoretically speaking, the interoperability proposal would basically allow replicating the semiosphere structure, hence, specific groups would find their own known language constructs as part of the model interchange language. It is a major challenge because this requirement implies to find the way of representing core concepts using language structures which appear to be stable ones and represent *i** variations with different language structures which should be flexible at the same time.

Stability. The main model interchange language elements shall represent mature and stable *i** constructs. As a result, the language shall represent the maturity of *i** established along its temporal use.

These stable concepts were already studied [CARE'11c] and they are used to formulate the model interchange proposal. Besides, if these stable concepts are represented under main structures of the proposal then it would contribute to implement a proposal which

imitates a semiosphere structure. From the assumption of an underlying i^* supermetamodel these stable language constructs would be the main concepts of a super set of concepts.

Extensibility. The model interchange language structure shall allow extending the i^* language with new constructs, and/or considering new aspects of existing constructs.

We are assuming that i^* modelling has achieved the behaviour of a paradigm. Although this implies that the language may become stable in the next years, there is no indication that the “revolutionary” stage could easily reach its final state. Thus, it seems to be reasonable to wait for new proposals extending or specialising i^* language constructs which should be possible to represent under the proposed model interchange language.

From the semiosphere point of view, it would be desirable to reach extra semiotic spaces, i.e. to reach symbols and meanings out of the semiosphere. From a cybernetic point of view, this requirement would allow the model interchange language to be adapted to other ontologies improving the structural coupling with interacting systems. From Wachsmuth’s formal perspective, *extensibility* would be a way of implementing the increasing module-variation preservation.

Filterability. The interchange elements shall be easily separable among different criteria in order to perform adequate analysis. It means that new elements (due to extensibility) will be described as part of the known language constructs in order to allow their filterability.

The theoretical process of meaning-making is basically a process of matching perceived language expressions to the internal ontology. In this process, it is possible that some language elements cannot get sense. They would be easily filtered without loss of the main meanings. Thus, *filterability* allows representing multi-ontology i^* variants’ space. From Wachsmuth’s formal perspective, *filterability* would be a way of implementing (mainly detecting) the decreasing module-variation preservation.

Simplicity. The model interchange language’s structure shall be easily readable by humans. It implies that language elements correspond as much as possible to the most

common names of the selected i^* constructs. Moreover, the design process involves human co-operation, this means using the models to get information either for understanding models and/or for transforming them. Therefore, simplicity means to reach legible language structures to parse, query, understand, and transform model representations.

This feature is desirable due to the pretension of representing the i^* semiosphere. As a semiotic problem, the most used symbols should be easily recognizable by members of the community.

Flexibility. The model interchange language shall allow representing incomplete i^* -related information, e.g. incomplete diagrams, and shall allow tools processing i^* diagrams even though they include constructs which are not directly treatable by a specific tool.

Practical reasons related to the human design problem are considered. It is said that the design process is progressive and it involves human collaboration (co-elaboration). It would imply sharing models at all design stages; hence representing incomplete models *flexibility* is a desirable feature.

Minimality. The model interchange language's elements shall constitute a minimal set of language constructs for representing the required knowledge on i^* .

Just as a way of avoiding redundancy and implementing simplicity, a traditional condition of minimality has been added to the set of the interchange language constructs. However, once given the open structure this minimality should be understood as there are no multiple representations for core concepts.

4.3 *iStarML: from foundations to language features*

To support the requirements above, a technological approach using an XML-based interchange file format is proposed. Nowadays, XML is the de-facto interchange format

in Internet which has reach to a fifth edition since 1998 [BRAY'08]. Moreover, from its irruption, XML has been used in many different disciplines [DONG'00]. The XML language is based on tags that could be nested and mixed with text data. Also, the tags admit attributes for keeping track of properties. Moreover, for defining specific languages using this structure, it is possible to use different Schema Languages [DONG'00]. Besides, there are many software tools and complementary languages (e.g. XPath [CLAR'99]) which help to create, parse and process any XML-based language.

Following the usual naming pattern in the XML community, the interchange proposal has been named *i** Markup Language or simply iStarML. The main distinguishing feature in iStarML is its aim to cover a set of *i** language variants, furthermore these variants are not known in advance and may be still non-existing. Because any XML language is assuming an underlying language metamodel, iStarML has been formulated under the assumption of having a common supermetamodel for *i** language variants which will be its underlying language metamodel. Moreover, its different level structure (e.g., tag elements, sub tags, attributes) can be understood as degrees of importance, hence, this structure could be used to represent the *i** semiosphere.

All these considerations are assuming that human behaviour theories would be feasible to remain in the model interchange language structure which means to fulfil the established requirements. Therefore, the way how the requirements are fulfilled is presented below.

The fact that iStarML is an XML-based language contributes to the goal of flexibility (XML allows specifying optional structures), filterability (the use of some known XML query languages, such as XPath [CLAR'99], allows selecting particular elements in an *i** diagram), extensibility (by the redefinition or use of extensible XML data types) and expressiveness (XML optional attributes also allow representing the current and future variations of the language).

To use a core set of stable *i** language concepts contributes to stability (iStarML focuses in the most mature concepts, i.e. those concepts which have been used in different *i** related proposals with the same meaning), minimality (a core set means that there is neither redundancy of concepts nor redundancy of language constructs) and

simplicity (having a reduced set of clear and differentiable concepts contributes to an easy understanding of the language).

To implement *i** language variations in terms of stable concepts fixes a relevant implementation strategy that makes it possible both to keep the focus on a set of mature and abstract concepts and, at the same time, to include *i** language variations as options of this core set. Thus, it contributes to extensibility (a broad door is kept open in order to represent language variations) and expressiveness (under the same schema, it is possible to represent current language variations). As a side effect, filterability becomes possible because both variations and new elements can be filtered since the supporting language structure is known.

Finally, two additional constructs for the language have been considered. On the one hand, due to the highly graphical nature of the *i** language, a construct for describing the graphical appearance of an *i** model component has been proposed, so it additionally contributes to expressiveness. On the other hand, a construct for delimitating diagrams has been included. This diagram construct contributes to expressiveness because different diagrams can be included in the same file. Moreover, having only one diagram is also considered to be a contribution to simplicity.

At the end of this requirements analysis, two open matters remain. Firstly, determining the set of the *i** language core concepts and secondly, to design the precise form that the iStarML specification takes.

In order to obtain the set of *i** core concepts existing on different variants, we studied 6 of the first *i** variants checking the usage, definition or redefinition of modelling constructs. The details of this study are in [CARE'11c] and have been represented in the *i** reference metamodel in Fig. 2.2. In order to get a more extensible and abstract view of core *i** concepts, those that are always present in each *i** variant, we have made some changes to that metamodel: all specialization constraints were changed from complete to incomplete allowing an easier addition of new subclasses; non-universal integrity constraints (e.g., restrictions on types of intentional elements) were removed; and Links were abstracted from InternalElements to IntentionalElements, allowing the definition

of links between dependums (and dependencies). The resulting metamodel is presented in Fig. 4.2.

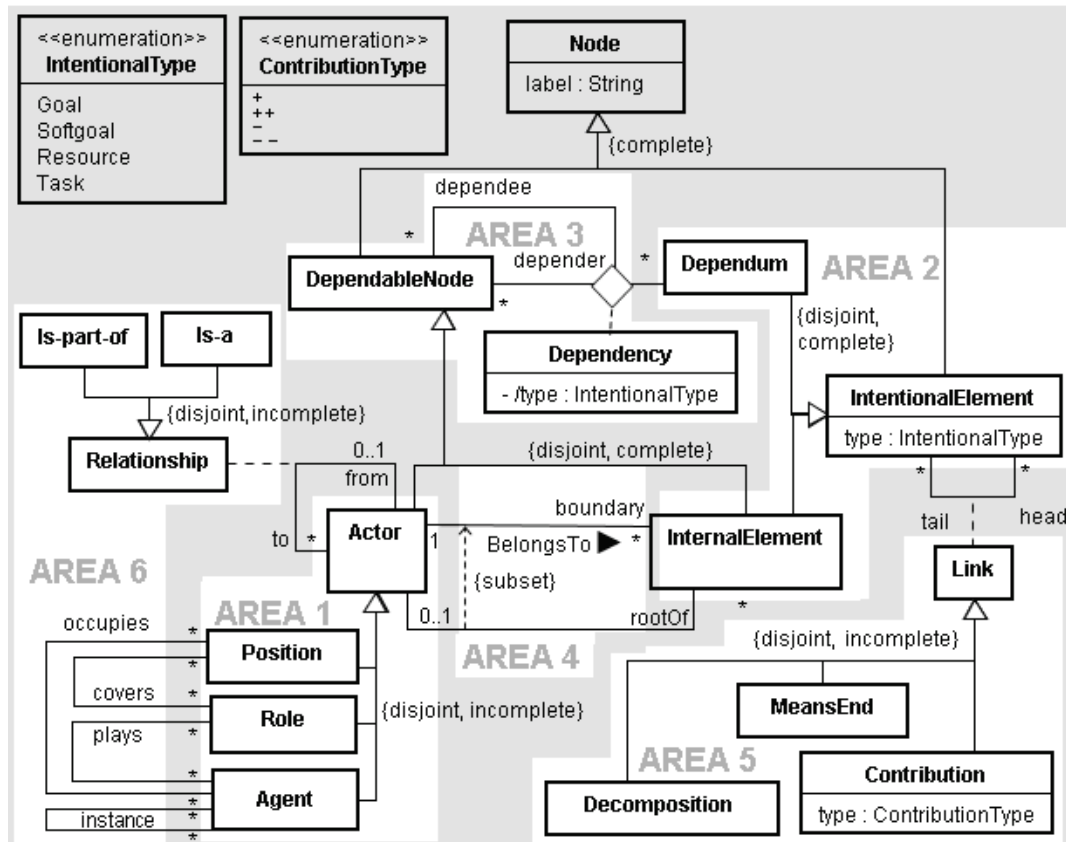


Fig. 4.2. The *i** core concepts in the context of the *i** metamodel

In this metamodel six different parts are distinguished which are highlighted in the figure and that yield to six types of core concepts:

- (a) actor (Area 1), for representing organizational units, humans or software agents;
- (b) intentional element (Area 2), for representing the set of elements which give rationality to the actor's actions, e.g. goals and tasks;
- (c) dependency (Area 3), for representing actors' dependencies in order to accomplish their own goals;
- (d) boundary (Area 4), for representing the scope of actors;

- (e) intentional element link (Area 5), for representing the relationships among intentional elements such as means-end or decomposition relationships; and
- (f) actor association link (Area 6), for representing the relationships among actors such as *is_part_of* and *is_a*, among others.

Each area has been considered as a category of core concepts that drives the structure of iStarML. Table 4.1 summarizes this result:

- The first column shows the core concept name and identifies the corresponding labelled area in the metamodel. For instance, in the first row we are proposing the mature concept of *actor*, which means that the concept is common and indeed it should belong to the set of *i** supermetamodel concepts.
- The second column describes the core concept, i.e. the common meaning extracted from the analysis of stable concepts of main *i** variants [CARE'11c].
- The third column identifies the name of the element that represents the core concept in the metamodel. In the first row, the tag `<actor>` corresponds to a main XML element.
- The fourth column describes the conceptual variations of the core concept. For instance, the tag `actor` has been specified with the attribute `actorType`. Therefore, if the tag `actor` is used with some `actorType` value (e.g., “role”), then it is possible to say that it is the case of a “generic” *i** language variant. However, if the `actorType` value is not used, the case will be about the representation of some *i** variant which does not specialize its actors (e.g., GRL). If additional actor types are required (beyond the known *i** language variants), then the additional values are also valid because any string is an admitted value for `actorType`.

In addition to these main elements, two explicit constructs initially considered for representing *i** diagrams and graphic expressions have been included. This action was attained by defining the corresponding tags `<diagram>` and `<graphic>`. In the first case, iStarML design allows many *i** diagrams being represented in the same file. In the second case, the `<graphic>` tag is a nested structure which specifies the graphic features that allow a graphic display of the *i** elements. Given that XML forces a tree representation, we have added attributes that allow referring to already represented elements, namely *iref* for intentional elements and *aref* for actors, thus the normal

graphs of i^* can be also represented. In order to support complex graphic expressions, SVG substructures have been considered. SVG is an XML-based graphic language gaining popularity [TOLK'03]. In order to express the syntactical choices, the traditional extended BNF (EBNF) meta language [SETH'96] has been used, see Table 4.2 for the case of the `<actor>` tag. The complete EBNF specification for iStarML appears in [CARE'07].

Table 4.1. iStarML abstract core concepts, tags and variation representations

Abstract core concept	Core Representation	Tag	Variation Representations
Actor (Area 1)	An <i>actor</i> represents an entity which may be an organization, a unit of an organization, a single human or an autonomous piece of software.	<code><actor></code>	By using the <i>type</i> attribute, traditional actors' specializations (role, position or agent) or new actors' types can be specified.
Intentional element (Area 2)	An <i>intentional element</i> is an entity which allows relating different actors conforming a social network or, also, to express the internal rationality elements of an actor.	<code><ielement></code>	By using the <i>type</i> attribute, traditional (goal, softgoal, resource and task) or other intentional elements can be configured. The attribute <i>state</i> can be used to specify an open set of intentional satisfiability values.
Dependency (Area 3)	A <i>dependency</i> is a relationship which represents the explicit dependency of an actor (depender) respect to the other actor (dependee).	<code><dependency></code> <code><dependee></code> <code><depender></code>	By using the <i>value</i> attribute on tags <i>dependee</i> and <i>depender</i> an open set of dependency features can be configured.
Boundary (Area 4)	A <i>boundary</i> represents a group of intentional elements. The common type of boundary is the actor's boundary which represents the vision of an omnipresent objective observer with respect to the actor's scope.	<code><boundary></code>	By using the <i>type</i> attribute, other explicit viewpoints (different from an omnipresent observer) can be added. No i^* variation has this feature but we think that including subjectivity is a natural extension to intentional models. This attribute could handle some extension like that.
Intentional element link (Area 5)	An <i>intentional element link</i> represents an n -ary relationship among intentional elements (either in the actor's boundary or outside).	<code><ielementLink></code>	By using the <i>type</i> and <i>value</i> attributes, traditional (decomposition, means-end and contribution) and new relationships can be represented. For example an or decomposition can be represented setting <i>type</i> to "decomposition" and <i>value</i> to "or"
Actor association link (Area 6)	An <i>actor relationship</i> is a relationship between two actors.	<code><actorLink></code>	By using the <i>type</i> attribute, traditional (is_a, is_part_of, plays, occupies and covers) and new and less used actors' relationships can be represented

Therefore, under a technical perspective, it is possible to affirm that iStarML achieves the goal of representing different *i** language variants by representing: (a) core *i** concepts using main tags; (b) known concept variations using defined values for defined attributes; (c) particular concepts variations using additional values for defined attributes, and (d) particular or projected concept variations using new values for new attributes.

Table 4.2. EBNF for the tag <actor>

<i>actorTag</i> ::=	<actor <i>basicAtts</i> [<i>typeAtt</i>] { <i>extraAtt</i> } > [<i>graphic-node</i>] { <i>actorLinkTag</i> } [<i>boundaryTag</i>] </actor> <actor <i>basicAtts</i> [<i>typeAtt</i>] { <i>extraAtt</i> } /> <actor aref="string" /> <actor aref="string"> [<i>graphic-node</i>] </actor>
<i>typeAtt</i> ::=	type="actorType"
<i>actorType</i> ::=	<i>basicActorType</i> string
<i>basicActorType</i> ::=	agent role position

Recalling the semiotic perspective of the semiosphere, where less stable concepts can be described in terms of the more stable ones, we are proposing a way to implement this structure using XML. In Fig. 4.3 this layered structure of iStarML is illustrated.

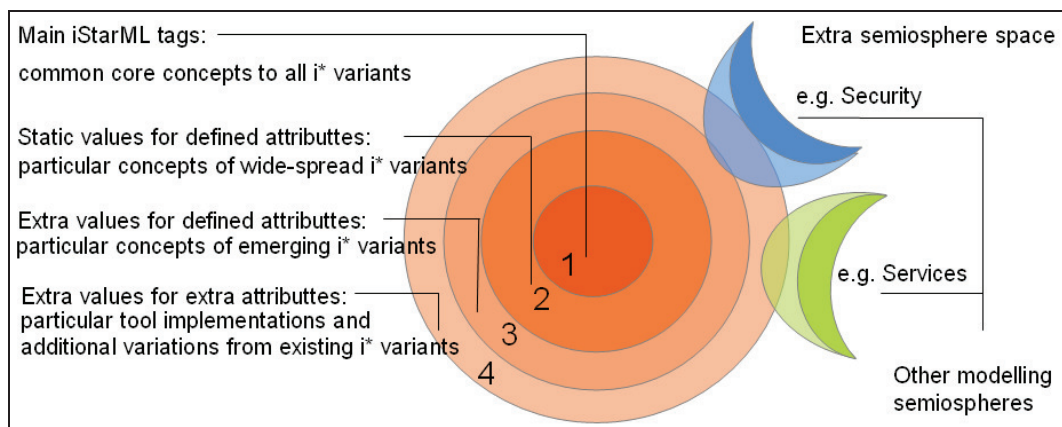


Fig. 4.3. The iStarML's semiosphere structure

4.4 Representing the *i** language variants in iStarML

In order to illustrate how the iStarML structure deals with *i** language variant representations, two examples are shown. Besides, in both of them, we use these representations to reflect on their interoperability (although a more complete analysis will be carried out in Chapter 5).

The first example, in Fig. 4.4, illustrates a basic diagram from the *i**-Tropos variant. This diagram shows that the intention of the actor A to accomplish the goal G depends on the actor B. The actor B has a scope of action (boundary) that includes a set of tasks. There are two decompositions: an and-decomposition (U1, U2 and U3) and an or-decomposition (V12 and V13). In Tropos, the line crossing the arrows indicates this difference. In fact, one of the differences between the seminal *i** and *i**-Tropos is to distinguish between simple decompositions (*i**) and decompositions of type “and” or “or” (*i**-Tropos). Under iStarML, this difference is represented by using the attribute *value* from the tag *iElementLink* (see Row 5 of Table 4.1), whose specific values are precisely “and” and “or”. This variation of decomposition’s values is established beyond core *i** concepts. In terms of the presented layer structure, the variation is established in the second layer. In Fig. 4.5 the corresponding iStarML code is shown.

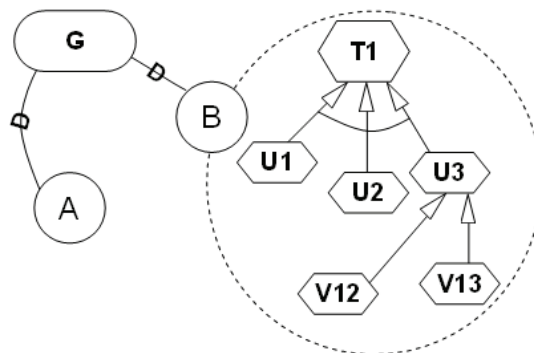


Fig. 4.4. Example of *i**-Tropos’s diagram.

```

<istarml version="1.0">
  <diagram name="Tropos's decomposition example">
    <actor id="100" name="A">
    </actor>
    <actor id="200" name="B">
      <boundary>
        <ielement name="T1" type="task">
          <ielementLink type="decomposition" value="and">
            <ielement name="U1" type="task"/>
            <ielement name="U2" type="task"/>
            <ielement name="U3" type="task">
              <ielementLink type="decomposition" value="or">
                <ielement name="V12" type="task"/>
                <ielement name="V13" type="task"/>
              </ielementLink>
            </ielement>
          </ielementLink>
        </ielement>
      </boundary>
    </actor>
    <ielement type="goal" name="G">
      <dependency>
        <depender aref="100"/>
        <dependee aref="200"/>
      </dependency>
    </ielement>
  </diagram>
</istarml>

```

Fig. 4.5. The iStarML code for the *i**-Tropos diagram of Fig. 4.4.

The interoperability-related question that immediately arises is: how can this *i**-Tropos model be converted into a Yu's *i** model (compliant with this thesis proposal)? Following Wachsmuth's formulation, it is possible to say that there is a partial instance preservation module-variation due to an injective function taking models from *i**-Tropos into Yu's *i** models. If this function just reproduces the structure, the only conflicting case is the distinction between "and" and "or" decompositions, represented in iStarML by the attribute value ("and" and "or"). Here there are many possibilities that the Tropos community can adopt in order to interpret this structure (e.g. converting or-decompositions into means-ends). However, we are assuming the worst semantic case, i.e., that there is not context information in order to adopt some bijective function. Therefore, the default interpretation comes from the fact of omitting the *value* attribute from the *iElementLink* structure, which is unknown in the target community. Obviously, it has semantic implications, and the semantic-preservation characterization corresponds to a decreasing-module-variation semantic preservation according to Wachsmuth, which should award to the users of the translated model that some loss of semantic occurred in the process.

In the second example (Fig. 4.6) a diagram from the Tropos' variant called Security-Aware Tropos is shown. In this variant, different types of dependencies are allowed. iStarML represents this extra characterization of dependencies using particular values in the *depender* and *dependee* tags, therefore, a particular extension to *i**, in this variant, is supported by core iStarML's tags. In Fig. 4.7, the corresponding iStarML code is shown. Note that the iStarML representation allows representing the permission on a separate dependency than the delegation one.

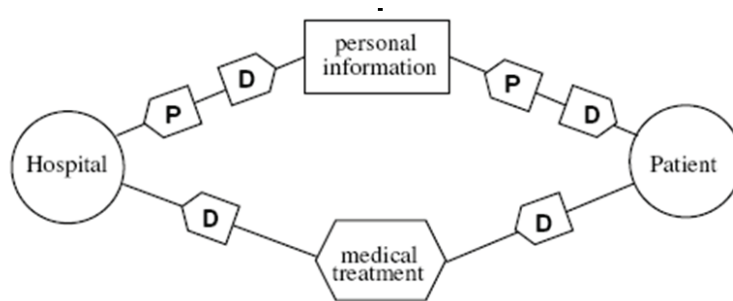


Fig. 4.6. Example of Security-aware Tropos' diagram

```

<?xml version="1.0"?>
<istarmml version="1.0">
  <diagram name="Example 8.4">
    <actor id="X1" name="Hospital"/>
    <actor id="A2" name="Patient"/>
    <ielement type="resource" name="Personal information">
      <dependency>
        <depender aref="X1" value="delegation"/>
        <depender aref="A2" value="permission"/>
        <dependee aref="A2" value="delegation"/>
        <dependee aref="X1" value="permission"/>
      </dependency>
    </ielement>
    <ielement type="task" name="medical treatment">
      <dependency>
        <depender aref="A2" value="delegation"/>
        <dependee aref="X1" value="delegation"/>
      </dependency>
    </ielement>
  </diagram>
</istarmml>

```

Fig. 4.7. The iStarML code for the Security-aware Tropos' diagram of Fig. 4.6.

4.5 Translating models using iStarML

Let's recall the problem as it was presented in Section 3.1.3, i.e. as a mapping function $\varphi_{\mathcal{A}\mathcal{B}}: \langle \mathcal{L}_{\mathcal{A}}, \Omega_{\mathcal{A}} \rangle \rightarrow \langle \mathcal{L}_{\mathcal{B}}, \Omega_{\mathcal{B}} \rangle$, $\varphi(m_{\mathcal{A}}) = m_{\mathcal{B}}$ for translating models from community \mathcal{A} to community \mathcal{B} . The problem appears when some constructs in $m_{\mathcal{A}}$ are not representable under $\mathcal{L}_{\mathcal{B}}$ due to the difference between the ontologies $\Omega_{\mathcal{A}}$ and $\Omega_{\mathcal{B}}$. Let \tilde{I} the set of all iStarML expressions and the function $istarml: \mathcal{L}_{\mathcal{K}} \rightarrow \tilde{I}$. It is clear that $istarml$ does not have an inverse function, however we can let $istarml^{-1}_{\Omega}: \tilde{I} \rightarrow \mathcal{L}_{\mathcal{K}}$ be the function which maps any iStarML expression to the known language elements of the community \mathcal{A} . We can define the function of *ontology restriction* imposed by the community \mathcal{B} as $\mathfrak{B}: \mathcal{L}_{\mathcal{A}} \rightarrow \mathcal{L}_{\mathcal{A}}$, $\mathfrak{B}(m_{\mathcal{A}}) = istarml^{-1}_{\Omega_{\mathcal{B}}}(istarml(m_{\mathcal{A}}))$. Obviously $\mathfrak{A}(m_{\mathcal{A}}) = m_{\mathcal{A}}$. Thus $\mathfrak{B}(m_{\mathcal{A}})$ is the portion of model $m_{\mathcal{A}}$ which can be “understood” by \mathcal{B} . Note that $\mathfrak{B}(\mathfrak{B}(m_{\mathcal{A}})) = \mathfrak{B}(m_{\mathcal{A}})$ and, however, the iStarML representations would be different. Formally speaking, there are two different strategies to tackle iStarML representations of a model $m_{\mathcal{A}}$ which would be read by a community \mathcal{B} .

- To represent the iStarML corresponding to $m_{\mathcal{A}}$, i.e. $istarml(m_{\mathcal{A}})$. This means that all constructs are kept without any interpretation in the target model. Since the file is a well formed formula under both XML and iStarML, this is an admissible behaviour. Therefore, the technological problem of a tool defined for $\mathcal{L}_{\mathcal{B}}$ finding an unknown structure in $istarml(m_{\mathcal{A}})$ can be addressed either by simply skipping the structure (at the time of reading) or else to build an ad-hoc behaviour. Under this situation, handling an iStarML file coming from any i^* model does not necessarily mean loss of information but it implies some extra complexity at reading time because some action will be required for handling unknown structures. At this point, some mapping function can take place to translate part of the unknown constructs into a feasible interpretation for community \mathcal{B} .
- To represent the iStarML corresponding to $\mathfrak{B}(m_{\mathcal{A}})$, i.e. $istarml(\mathfrak{B}(m_{\mathcal{A}}))$. This implies to avoid unknown parts for the target i^* variant. This means that these unknown constructs are not allowed to appear in the target model and will be removed. Under this situation loss of information may occur, however, the \mathfrak{B}

application would include some mapping to partially avoid this loss of information and translate unknown formations to “some” \mathcal{L}_B 's structures if it is possible.

To illustrate this problem, we can consider the transformation of the model represented in the Fig. 4.6 into another i^* language variant representation, e.g., i^* -Tropos as illustrated in Fig. 4.4. It requires the interpretation of the extra attribute *value* existing in the *dependor* and *dependee* tags. To keep it or not only changes the point of a possible loss of information or variation on the meaning in the involved expressions.

4.6 Characterizing semantic preservation

Since our interoperability framework is intended to support interoperability among i^* language variants that may refer to different ontologies, then it will be natural to find semantic preservation differences. It is worth remarking that these semantic differences are not a problem introduced by iStarML but by the own nature of the considered variants. Therefore, as part of the proposal, we have included a way of characterizing the semantic-preservation of the chosen model transformation.

In Section 3.1.3 we have stated that the translation between models from the community \mathcal{A} to the community \mathcal{B} can be represented by a mapping function $\varphi_{\mathcal{A}\mathcal{B}}: \langle \mathcal{L}_{\mathcal{A}}, \Omega_{\mathcal{A}} \rangle \rightarrow \langle \mathcal{L}_{\mathcal{B}}, \Omega_{\mathcal{B}} \rangle$, $\varphi(m_{\mathcal{A}}) = m_{\mathcal{B}}$, which implies that $\Omega_{\mathcal{A}} \cap \Omega_{\mathcal{B}} \neq \emptyset$, i.e. there are common ontological distinctions in the communities \mathcal{A} and \mathcal{B} but it does not necessarily mean that $\Omega_{\mathcal{A}} = \Omega_{\mathcal{B}}$. Therefore, we have in the general case i^* variants which have a common core and, at the same time, they have ontological differences. If we apply the common contemporary operation (at least in some Software Engineering research lines) of matching ontologies and metamodels then we would use Waschmuth's categories to characterize semantic preservation. However, to us, it is just a reduction of a limited representation of a modelling language \mathcal{L} . An accurate match of both frameworks should consider a more complex situation. In Table 4.3 we outline such a match. Note that although Bunge's framework does not include the problem of model translations,

we have approached the definition of a mapping function $\varphi_{\mathcal{A}\mathcal{B}}$ in terms of Bunge's framework.

Table 4.3. An approach to a conceptual matching between Waschmuth's and Bunge's frameworks

Waschmuth's elements	$I(\mu)$	$C(\mu)$	Semantic is not part of the framework	Φ
Bunge's elements	$\xi(\Sigma)$	Ω	Δ	Adaptation (translation) is not part of the framework

Therefore, as part of our interoperability approach, a first algorithm is proposed for a generic translation of a model $m_{\mathcal{A}} \in I(\mu_{\mathcal{A}})$ to a model $m_{\mathcal{B}} \in I(\mu_{\mathcal{B}})$. It includes the characterization of the resulting semantic preservation (according to Waschmuth). It is important to remind the theoretical assumption of the existence of the i^* supermetamodel that we call i^*_{SM} .

In fact, we have also proposed a first approach to i^*_{SM} in [CARE'11a]. In it, we consider generic and partial formations, e.g. a model containing only intentional elements is a valid formation. In Figure 4.8 we reproduced the illustration of our i^* 's supermetamodel approach. In it, the $\ll XClass \gg$ stereotype allows having additional features from particular i^* variants by handling a list of attribute-value pairs.

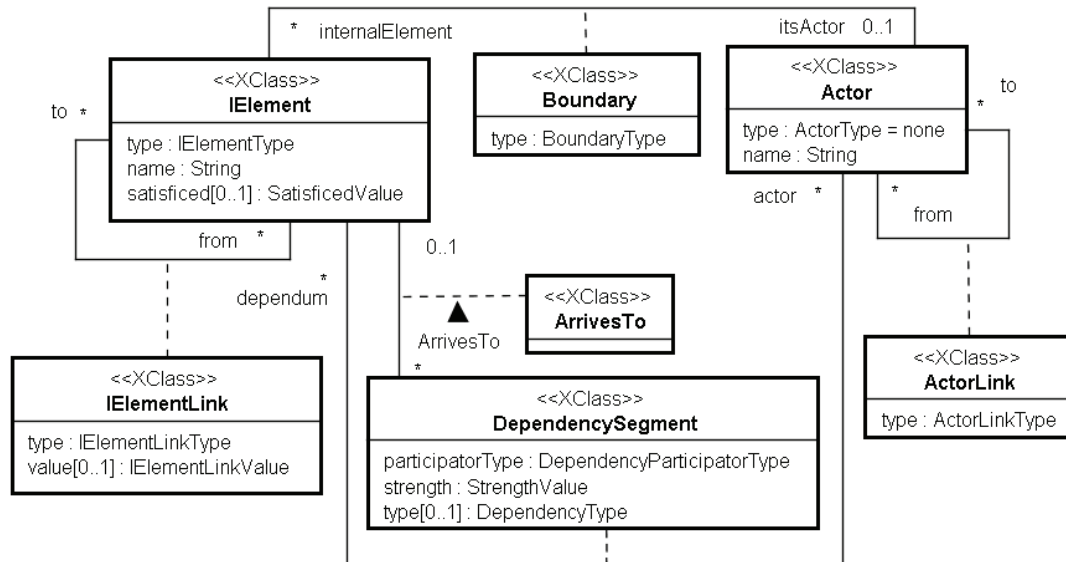


Fig. 4.8. The i^* supermetamodel

Thus, we have a version for i^*_{SM} , therefore, in our example $m_{\mathcal{A}} \in I(i^*_{SM})$. Moreover, if we assume that iStarML has the language structures to represent i^*_{SM} , which can be

seen by the match between supermetamodel concepts and iStarML tags, then $istarml(m_A)$ represents m_A without loss of information. Translating this model means to get an iStarML representation $istarml(m_B)$ which should preserve known elements from $istarml(m_A)$ and propose (possibly null) mappings for B 's unknown elements. Note that Waschmuth characterizes the semantic consequences of losing or changing information but he does not include a semantic theory of mapping, therefore specific mappings stay out of the scope of this proposal and we include them in a generic way.

The proposed transformation algorithm is shown in Fig. 4.9 as a UML activity diagram. The information about the semantic preservation consequences are established on it. The activities are:

- *Copy known formations.* The part of m_A that is also a valid instance of μ_A is directly considered as part of m_B . In other words, the concepts shared by both metamodels μ_A and μ_B are kept. In the case that the full model m_A is a valid instance of μ_B , the transformation finishes and it is classified as strictly semantic preserving translation. Example: a generic *actor* is always kept as a generic *actor*.
- *Translate using bijective mappings.* Let m_{A0} be the part of m_A that has not been treated in the previous activity. The part of m_{A0} from which a bijective mapping to the elements of μ_B exist, is translated using these mappings. In other words, the concepts which can be expressed in both metamodels μ_A and μ_B but with different constructs, are just translated. In case that the completed m_{A0} had been processed after this activity, the translation will be semantic preserving module variation (the variation introduced by the mapping). Example: a task can be translated into a plan and a plan into a task.
- *Translate using injective mappings.* Let m_{A1} be the part of m_A that has not been treated in the previous activity. The part of m_{A1} from which an injective mapping to instances of μ_B exist, is translated using this mapping. In the case that the completed m_A had been treated after this activity, then the translation will be decreasing module variation (the variation introduced by the mapping). Example: a constraint representation from some specific i^* variants can be translated to softgoal, but not any softgoal can be read as a constraint. Also a *make* contribution from GRL can be

translated into ++ contribution in the seminal i^* , but not any ++ contribution is a *make* contribution.

- *Forget non translated formations.* Finally, those constructs in m_{AI} that have not been translated in the previous activities, are just not considered. Example: a *belief* from GRL when translating into Aspectual i^* .

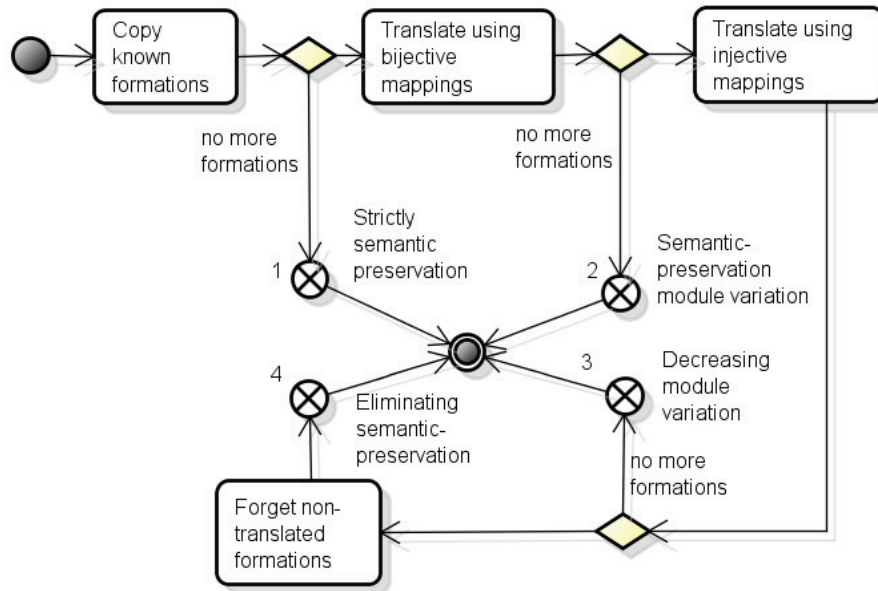


Fig. 4.9. Translation algorithm from the i^* supermetamodel to an i^* variant

4.7 Summary

To conclude with this chapter, we have formulated a technological approach that has called iStarML, it has been founded on different theoretical frameworks such as the concepts of ontology, semantic and human communication from diverse theoretical perspectives. We have provided a set of language requirements that have guided the interoperability proposal and, finally, we have sustained how they have been accomplished. Moreover, we have illustrated representational capabilities of iStarML and, as part of the proposed framework, we have provided an algorithm for model translations which categorizes the semantic preservation of the i^* model transformations.

We have not suggested specific semantic equivalences because it appears to be a little bit risky due to it would be the case of pragmatic sub-communities more than

homogeneous ontological sub-communities, i.e. even the same language structure would have different meaning in different sub-communities.

Therefore, the interoperability proposal seems to accomplish the practical and theoretical points to be a satisfactory approach to *i** models interoperability beyond semantic differences, it considers the interpretations and usages, in words of the cybernetic reviewed approaches, it is an interoperability which enable new behaviours, therefore it attempts pragmatics issues beyond semantic variability.

In the next chapter a set of iStarML applications will be reviewed and a first community's perception will be reported.

CHAPTER 5. Applying the iStarML Interoperability Proposal

5. Applying the iStarML Interoperability Proposal

The goal of this chapter is to demonstrate that iStarML is a feasible technological proposal for i^* model interchange. In the first part, we illustrate two proof of concepts of using iStarML for sharing i^* models coming from different i^* variants. In the second part we show other possible scenarios where using iStarML as textual representation implies acquiring positive externalities. In the third part of the chapter, we show the outcomes of a survey conducted in 2008 about the i^* community to gather some empirical data about iStarML perception.

5.1 *iStarML Interoperability Proof of Concept*

In this section we show two proof of concepts of model interchange between different i^* variants. We briefly present the tools used to implement each case.

5.1.1 jUCMNav

jUCMNav [JUCM] is a tool for supporting the GRL language, which has a graphical editor and an analysis and transformation tool for the User Requirements Notation (URN). jUCMNav is available as an Eclipse *plugin*. JUCMNav is an open source software, therefore its source code is available which has made it possible to implement easily our proof of concept.

5.1.2 OME 3

The OME 3 modelling tool is a graphical *i** editor which allows handling *i** projects as sets of models. The models can be compliant to the *i** seminal proposal or to GRL. It is a Java application and uses its own file format. OME 3 is one of the most widespread tools in the *i** community, in fact, most of the *i** diagrams appearing on research proposals are shown as OME 3 diagrams. Although it is not as open source software, the storing format is legible; therefore, we used OME as part of our proof of concept anyway.

5.1.3 HiME

HiME [LOPE'09] is an *i** editor supporting the proposal appearing in [LOPE'08], which has the ability to deal with specialization between actors (*is-a* link) at the level of SR elements. It has particular language constructs for differentiating inherited elements. HiME's main graphical feature that distinguishes this editing system is that a model is not represented graphically following the symbols of the *i** framework, instead it is represented as a folder tree directory in a file system. HiME has been developed using Java and the Rich Client Platform (RCP) for Eclipse. Models in version v2.0 are stored using the iStarML.

5.1.4 First Case: from OME 3 Models (Yu's *i**) to jUCMNav Models (GRL)

This first case shows the interoperability problem involving two different *i** variants. According to Waschmuth's framework, we compared the two metamodels and searched for semantic categories. The metamodel behind jUCMNav is presented in [AMYO'09] and as *i** metamodel we use our own proposal [CARE'10;CARE'11c].

Some relevant differences are:

- *Actors*. In jUCMNav, only the generic type *Actor* is supported, the specializations *Position*, *Role* and *Agent* are not included.

- *Intentional elements.* jUCMNav includes, as GRL does, a special type of intentional element, namely *Belief*. However, in the metamodel itself, beliefs are not intentional elements but separate graphic nodes. This illustrates also a recurrent situation in which a tool implementing a variant may not support exactly the metamodel proposed in the theoretical work.
- *Dependencies.* In jUCMNav, *DependableNode* is always an *InternalElement*. Actors are not graphically linkable although the metamodel seems to be ready to allow it. Therefore, actors must contain internal elements to attach the dependency ends.
- *Intentional element links.* jUCMNav offers the possibility of adding some information on *Contribution Links* (quantitative contribution). This quantitative value may vary between -100 and 100 (integer units) and jUCMNav offers a mapping between these values and the qualitative ones.
- jUCMNav has only two specializations of *Links*, *Contributions* and *Decompositions*, instead of three. It means that there are not means-end links.
- *Actor links.* Apparently jUCMNav does not support actor links (*Relationship*), but in fact, it allows representing a nested structure of actors, which matches the concept of *is-part-of*.

The treatment of these misaligned situations is described below:

- Actor types. Since actor types are implemented in iStarML as attributes, OME 3 exportation keeps them. When importing into jUCMNav, actor types can be kept only as metadata.
- Dependencies. It is necessary to decide how to interpret an iStarML file coming from an OME 3 model in which some dependency end is an actor instead of an internal element. Basically there are two possibilities: to create an intentional element inside the actor to which attach the dependency; or to keep the dependum but deleting the dependency end(s), keeping that removed end as dependum's metadata.
- Intentional element links. In this case means-end links are interpreted as OR decompositions and decompositions are interpreted as AND decompositions. Note that we are taking a semantic decision, on behalf of the target community, given that other interpretations are also possible, for example, interpreting a means-end link as a XOR decomposition.

- Nested actors' structures. In order to represent nested actors' structures, the iStarML recommendation of using the aLink tag using the attribute *type* set to the value *is_part_of* is followed. The *is_a* language constructions are omitted since jUCMNav does not support them.

In the final step, the programming activities take place. As it was already mentioned, OME 3 stores its models using a Telos-based representation. This representation includes the classes and instances that are included in the model. In order to translate from this representation we have built a Java Applet which makes the transformation [CARE'11a].

In Table 5.1 we show four translation samples to illustrate the transition of *i** models from OME 3 to jUCMNav. In addition, for each case we have applied the algorithm proposed in Chapter 4 to categorize the semantic preservation. Below the rows of this table are explained.

- Row 1: the sample corresponds to a dependency from an intentional element into another. The translation is strictly semantic preserving: the complete model is translated without changes. It corresponds to the output 1 in the translation algorithm.
- Row 2: the sample corresponds to a task decomposition with dependency to an intentional element. The translation is semantic preservation modulo variation: the task decomposition in OME is translated into an AND-decomposition in jUCMNav. Note that it would be possible to recreate the original model. Therefore, this is a bijective mapping. It corresponds to the output 2 in the translation algorithm.
- Row 3: the sample corresponds to a dependency from an intentional element to an actor. Please note that jUCMNav does not admit dependencies with actors as dependers or dependees. The translation is decreasing modulo variation: it is possible to translate the dependency by creating an intentional element in the target actor and attaching dependency on it, but the original model cannot be recreated, since it is not known if the added intentional element is really new or not. In particular, note that this jUCMNav model is identical to the previous one,

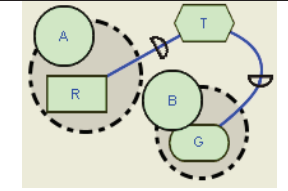
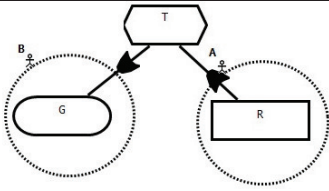
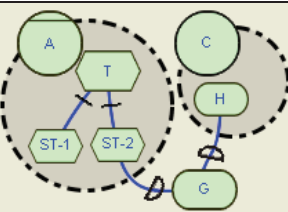
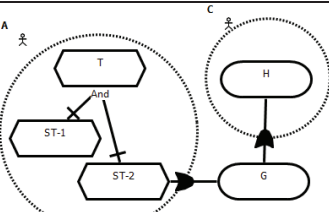
clearly showing the lack of bijection with respect to this particular point. The translation algorithm shows this case of semantic preservation in the output 3.

- Row 4: the sample corresponds to an agent as instance of actor. Although the agent is converted into actor (decreasing modulo variation), the instance link is lost. The translation is eliminating semantic preservation: the element cannot be kept and is removed. Note that this possible case translation implies loss of information. The translation algorithm shows this case of semantic preservation in the output 4.

It is necessary to remark that these samples are not proposing specific semantic equivalences from one variant to another, they only illustrate how these translation can take place by using iStarML and how to semantic preservation is categorized.

Under our theoretical approach the existence of many i^* variants implies the existence of different semantic-pragmatic communities and the equivalences or mappings among metamodels inside the specific community should be a matter of a meaning-making process. Just to mention an example, Row 3 and Row 4 are proposing two different strategies for dealing with one specific construct (dependency with an actor as dependee) that is supported in the departing metamodel but not in the target metamodel. Choosing one or another depends on the target community.

Table 5.1. Classification of specific model translations from OME to jUCMNav.

from OME	to jUCMNav	Semantic preservation
		Strictly semantics-preserving. There are no changes from the original representation
		Semantics-preserving modulo variation. A decomposition is mapped to an and-decomposition and an agent is mapped to an actor

from OME	to jUCMNav	Semantic preservation
	<p>Increasing modulo variation. A goal is added as internal element in the actor C and the decomposition is mapped to an and-decomposition</p>	
	<p>Eliminating semantics preservation. The INS association is removed and the dependency is lost</p>	

5.1.5 Second Case: Translation from jUCMNav to HiME

We find HiME supporting the same seminal i^* as OME 3, so we have a similar misalignment (ontological difference among variants). Therefore, all the already recognized differences between jUCMNav and OME metamodels are also presented here. In addition and concerning actors, HiME supports all the possible actor links. But a word of caution needs to be given for specialization: the *is-a* relationship between actors has a lot of implications in HiME's models, namely correctness conditions establishing which elements may be modified, deleted, moved around, etc. Fig. 5.1 illustrates particular concepts of HiME's metamodel related to inheritance.

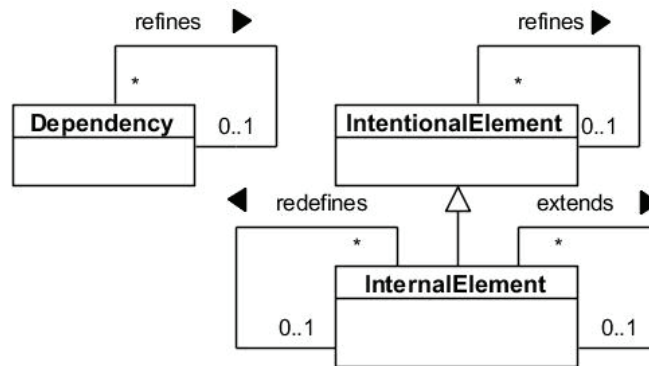


Fig. 5.1. Particularities of the HiME's metamodel related to inheritance

We have created models in both tools and then we have imported into each other. Fig. 5.2 shows a model created in jUCMNav (top) and the resulting model read from HiME (bottom). The model imported in HiME has the following differences respect to the original one:

- Some elements not included:
 - *Existence of Untreatable Diseases* because HiME does not recognize beliefs as a kind of intentional element.
 - Links for decomposition of *Get Treated* into *Follow Public Treatment* and *Buy Private Insurance* because HiME does not recognize xor-decompositions.
 - *Quantitative -75* associated to the contribution link from *Approve Treatment* to *Fast* because is not possible to add quantitative information to contributions.
- The default value *committed* has been assigned to all strengths (this default value is customizable through a configuration file).
- Or-decomposition for goal *Be Well* is read as a means-ends link.
- Or-decompositions for softgoal *Profitable* are read as contribution links with value equal to OR.

The complete proof of concept is in [COLO'11] and it includes the inverse case, i.e. the translation from i^* -HiME to GRL-jUCMNav, which is a similar case to the already reviewed in the previous point.

The interconnection of these three tools has fulfilled the objectives of showing that iStarML can sustain i^* models interchange. We have demonstrated the feasibility of tool interconnection and then model translations. Although there are some limitations, those are inherent to the semantic mismatch between the involved metamodels; however, these mismatches have been classified using a well-known metamodel-oriented semantic framework which allow categorize the semantic preservation if any.

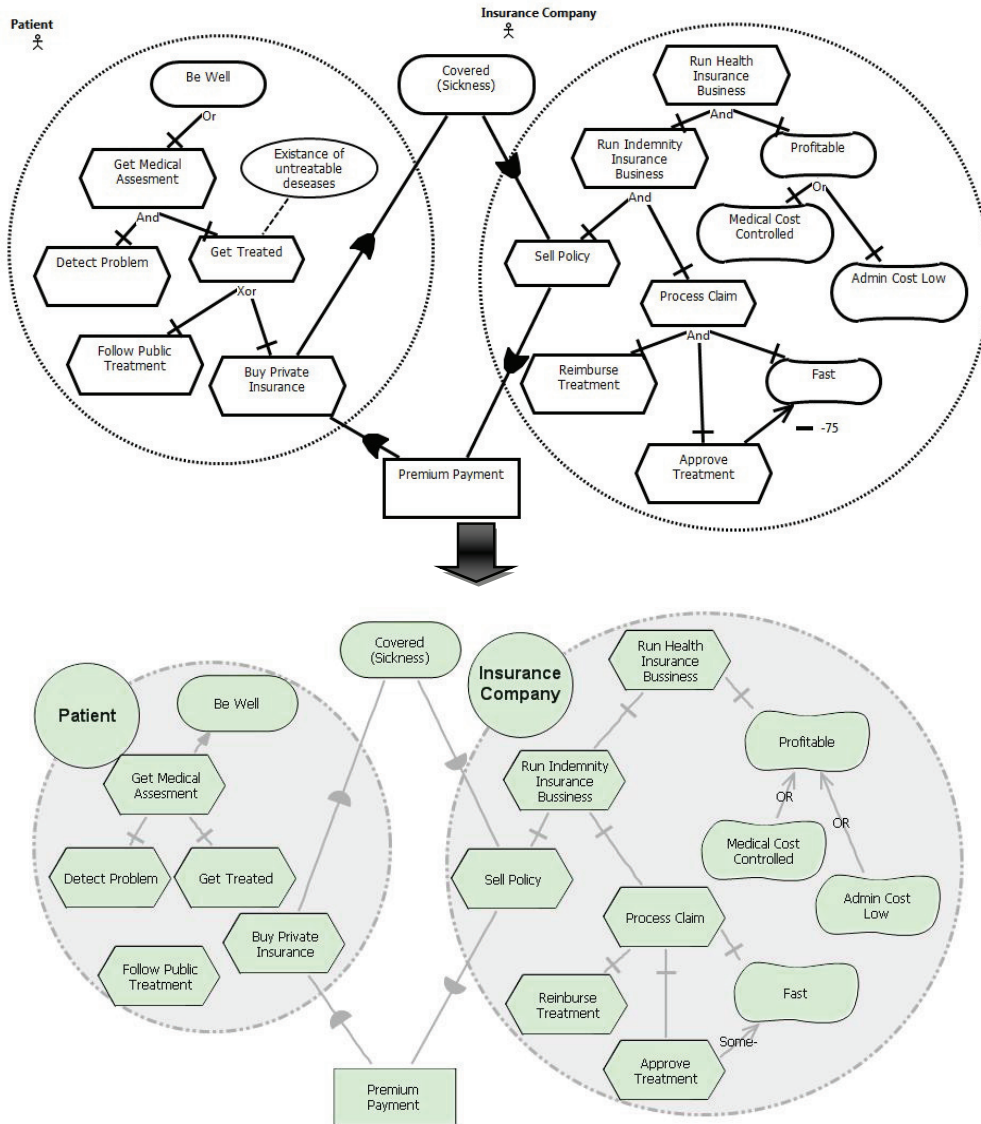


Fig. 5.2. Model created using jUCMNav (top) and the model read from HiME (bottom)

5.2 Interoperability Scenarios and Technological Support

Providing an interoperability approach for i^* models also implies to take into account the context under which it can be used. In the case of an RE modelling framework, we can sustain that the context is the software process where the RE stage takes place; therefore, we offer a perspective of how iStarML may be used in a software process. The simplest view is a basic model between two modelling tools; however, as part of

software process, interchanging and sharing models can enable different engineering practices. Here we have considered a set of recommended practices for dealing with teams in software processes [HUMP'00] such as design verification, measurement, reuse, change tracking which we have instantiated to specific practices using iStarML representations. Accordingly, we summarize a set of iStarML usages, firstly for model interchange, but also for *i** model verification (design verification) and metric calculation (measurement). Additionally, storing iStarML files enables models reuse and change tracking.

5.2.1 Generating and Reading iStarML by *i** Tools

For supporting interoperability, the most basic functionalities to be supported are the ability to import and export *i** models represented by iStarML. Our work in this direction has progressed from an individual initiative to a community joint effort:

- In the period 2008-2010, we implemented this functionalities in our HiME tool [LOPE'09] (the evolved model editor from J-PRiM [GRAU'06]). Also, we implemented the functionalities in the form of plug-in for the jUCMNav tool developed by the University of Ottawa [JUCM]. Last, we provided a translator from OME3 to iStarML [CARE'11a]. As a result of this first stage, we learned a bit more about the iStarML behaviour from different perspectives, and we were reassured that it can be used not just in our own tools but in others'.
- In the period 2010-2011, other groups in the community started to use iStarML. Remarkably, in the University of Leipzig, it has been generated an approach for checking *i** models using Prolog facts [LAUE'11a] and, in order to extend its application, they are including an iStarML exporter [LAUE'11b] on the tool OpenOME [HORK'11]. In CENIDET, an ontology oriented tool (Tagoon) which imports iStarML files in order to include *i** models as part of their analyses has been developed [NAJE'11]. Informal and separately, the software engineering research groups from Foundation FBK in Italy, and Universidade Federal de Pernambuco, in Brasil, have declared their intention of including iStarML in their modelling tools, TAOM4E [MORA'11] and iStarTool [MALT'11], respectively.

5.2.2 Metric Calculation

Metrics are a valuable analysis for performing analysis during several stages of the software process. In the case of a metric-calculation agent, we can sustain that it is possible to look for design anomalies using some goal and actor metric framework, e.g. [FRAN'06]. For example, an indicator could be the relation between the load of the most goal-loaded actor and the ideal situation of balanced goal loading. If we have an iStarML representation of the analysis domain then we may use XPath [CLAR'99] in order to calculate this metric value. In Fig. 5.3 we show a reduced view of a specific iStarML file (goals are hidden) and the XPath sentence which allows obtaining the value. On this result, we can say that the most goal-loaded actor is 2.4 times higher than a balanced situation. We observe the simplicity of the resulting query and its direct relation to constructors of the *i** language.

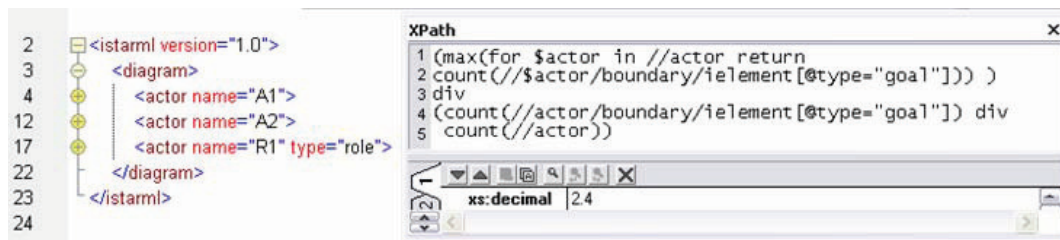


Fig. 5.3. Goal and actor metric calculation using XPath on iStarML files

5.2.3 Parsing and Handling iStarML Files

Under the same scenario, other simple example is parsing iStarML files (Layer 5). In order to show its feasibility, a Schematron [JELL'00] parser has been developed. Schematron is a rule-based syntax checker that allows customizing error messages. Applying an XSL transformation, the iStarML Schematron specification produces an XSL file (istarml.xsl), which allows verifying iStarML files. This transformation file can be downloaded from [DAM'03].

In addition, we have also developed a Java package for handling iStarML files. We have called it ccistarml package. The diverse functions in the package allow creating, reading

and modifying iStarML files. Therefore, the ccistarmml package allows checking the XML syntax and the specific iStarML syntax. Moreover, it allows handling and creating an iStarML structure by using a reduced set of Java classes. The Java classes and their functionality are explained in [CARE'11d].

In order to illustrate a simple use of the ccistarmml package, in Fig. 5.4 a Java code portion is shown. In this sample an iStarML is loaded and validated according to XML rules and then according to iStarML construction rules. Afterwards, the actor “Tutor” is added to the content and the new iStarML structure is saved into a modified iStarML file.

```
1 package ccmainssamples;
2 import ccistarmml.*;
3
4 public class Main {
5     public Main() {
6     }
7     public static void main(String[] args) {
8         ccistarmmlFile f = new ccistarmmlFile ();
9         f.loadFile("sample.istarmml.xml");
10        f.xmlParser();
11        f.istarmmlParser();
12        System.out.println("Cantidad de errores: "+f.errors());
13        if (!f.hasErrors()) {
14            ccistarmmlContent content = f.mainTagStructure();
15            content.add_actor("Tutor");
16            f.saveFile("sample02.istarmml.xml");
17        }
18        else
19            f.displayErrors();
20    }
21 }
```

Fig. 5.4. Using the ccistarmml package to parse and modify an iStarML file

5.2.4 Validating *i** Models by Using iStarML

The SEQUAL framework [KROG'95;KROG'06] argues that model quality depends on interpretation that involved people (stakeholders and software engineers) have built about it. It implies different type of model qualities, namely, syntax quality, semantic, quality, pragmatic quality and social quality. Moreover, in [KAVA'02] it is summarized that different goal-oriented proposals have different stages, in particular, it is asserted that *i** models have three sequential modelling times: (a) modelling the system as-is, (b) modelling the challenge system, and (c) modelling the to-be system. Under these two assumptions, i.e. quality types from SEQUAL and *i** modelling times from

[KAVA'02], we have proposed a framework for interactively assessing *i** models quality [CARE'09].

In order to perform model evaluation, a pattern-based schema is proposed. A pattern involves a type of symbol (e.g., a goal) to be validated, a specific metric, a query pattern for a question, and an answer scale. A prototype was developed to show the feasibility of measuring quality using these ideas. The *i** models were represented using iStarML and the query patterns were implemented by using XQuery language. In Fig. 5.5, an XQuery pattern on an specific iStarML question is presented. In the bottom part appears a sample of a generated question to-do in order to validate *i** models.

<pre> for g\$ in document("Tutoring.istarml") //ielement[@type="goal"] LET m\$:=g\$/ielementLink[@type="means-end"] /ielement[@name][@type] return <question type="yes-no"> It is possible to reach {g\$/@name} when {m\$/@name} ?</question> </pre>
<p>Is it possible to reach <i>Doubts solved</i> when <i>Solve doubts by email</i>?</p>

Fig. 5.5. Xquery-istarmml generation of questions for validating *i** models

Although this work is still under development, it illustrates that a textual representation as iStarML facilitates the validation of *i** models.

5.3 *iStarML: A Community's Perception*

In order to measure a first community perception about the interoperability problem and the proposed solution (iStarML), we conducted a survey regarding usages of the *i** framework in a first stage of our research. The population was defined as the *i** research and development community. This is the reason why the participants of a session in the Third International *i** Workshop in 2008 constituted the sample. The workshop had around 30 participants but there were more than 50 authors belonging to approximately 15 different working groups. The final sample size was 15. Traditional social variables such as gender or age were considered irrelevant with respect to the objectives of the

survey, then an homogenous population was assumed. Therefore, it is not necessary to stratify the population, hence the sample of researchers is representative: this sample may seem small but in fact it represents the core of the community of *i** researchers and developers.

In the survey, general knowledge about iStarML and a specific perception of the interoperability problem was investigated. It also included questions about iStarML possibilities for overcoming the interoperability problem and for the general willingness of adopting iStarML on researchers' work.

The instrument was mainly based on 5-degree Likert scale, from "strongly agree" to "strongly disagree". The questions about iStarML adoption were formulated describing different explicit adoption attitudes.

For data processing, the statistical recommendations given in [GOB'07] were followed, i.e. a multinomial approach for answer evaluation of the Likert scales was applied. This means that each processed question produced five proportions, the proportion of those who answered "strongly agree", the proportion of those who answered "agree", etc. However, following this procedure, no significant conclusion was obtained because the confidence intervals resulted quite wide and too overlapped. Then, a binomial approach was followed; this is a particular case of a multinomial one but considering two proportions only. This meant converting the 5-degree answers into success-fail answers as recommended in [PECK'98], which implied losing the grade of agreement or disagreement, but it allowed getting narrower confidence intervals.

Three cases of data interpretation were considered: answers in agreement were considered successful; answers in disagreement were considered fail answers; and, half of the answers checked "neither agree nor disagree" ("nor"-answers) were considered successful ones and the other half as fail ones. This option means choosing the maximum variance for a binomial case ($0.25=0.5 \times 0.5$). Under these considerations, a first statistical test by means of applying the simple and rough interval of Fitzpatrick and Scott recommended in [GOB'07] was used. Then, the interval of Agresti-Coull, recommended in [BROW'01] in order to get confidence intervals for binomial proportions was also used. Moreover, since Agresti-Coull confidence intervals allow

small sample sizes (starting from 12), a third analysis was added, this time discarding the “nor-answers”. The resulting Agresti-Coull’s confidence intervals using a probability of 95% ($\alpha = 0.05$) is shown in figure 5.6.

These results confirm the initial hypothesis because they point out that there is a shared vision about the existence of an interoperability problem. Even the worst case indicates that more than a 60% of the population recognizes the problem. Moreover, at least a 52% of the population agrees that iStarML overcomes the problem. If the centre of the interval is considered, then the different population proportions appear to be relevant.

Regarding the answers related to adoption (either any generic interoperability mechanism or specifically the iStarML proposal) they also yield good results. The worst case represents, at least, a 42.9% of declared first adopters. Analyzing this proportion under the Rogers’s innovation adoption theory [ROGE'95] it is possible to see that first adopters normally became a 13.5% and the “early majority”, after first adoptions, became 34% of the population. Although this 42.9% of declared adopters does not seem to be very high in relation to the whole i^* community, on the light of Rogers’s theory it seems to be one of the most optimistic findings of this survey. The last inclusions of iStarML described in subsection 5.2.1 would be a symptom of these fostering intentions.

An additional analysis perspective is that the workshop attendees constitute a sample formed by “special people”: at least innovators and experts. Therefore, the adoption tendency showed by the survey would correspond to leaders’ attitudes (in opposition to followers) and their opinions would correspond to experts’ judgments, i.e. these judgments correspond to the acknowledged namesake qualitative research methodology. Therefore, the conclusions coming from this perspective, i.e. a qualitative one, seem to give additional support to these findings.

Finally, it is also remarkable, that most of the reviewed theoretical frameworks assume the existence of a generic community or a scientific community. This iStar Workshop represent the event where the core of the i^* community normally participates, therefore we can sustain that a relevant part of used theories is accomplished. We have analyzed this social perspective and implications in [CARE'11e].

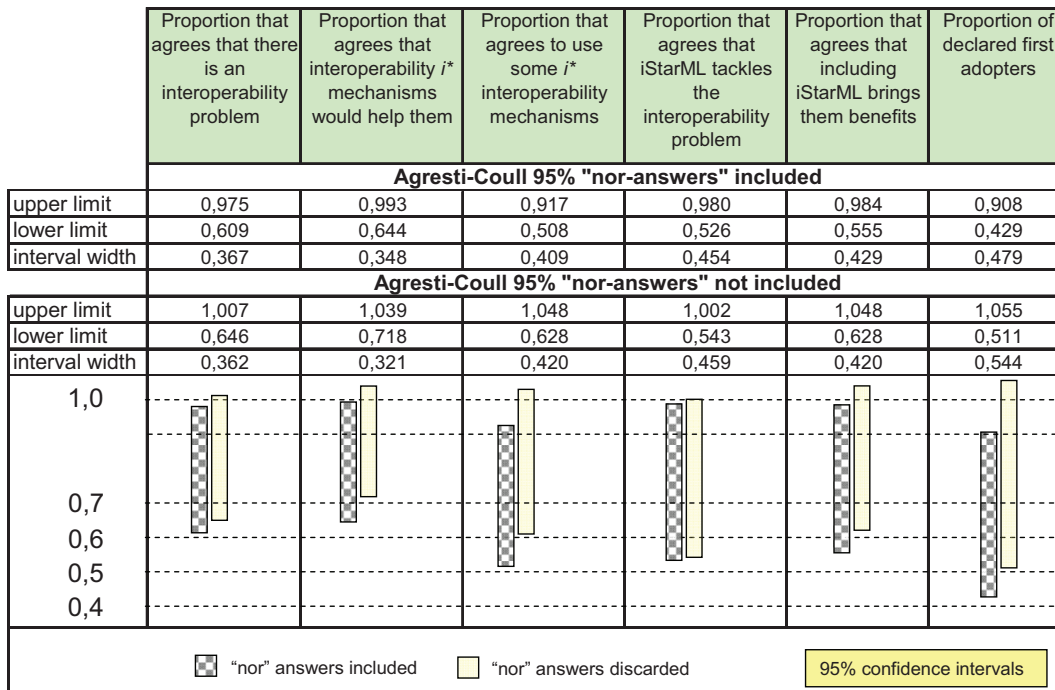


Fig. 5.6. Confidence intervals of the *i** community perception about interoperability and iStarML

CHAPTER 6. Conclusions

6. Conclusions

The origin of this thesis stems from the research performed by the GESSI research group centred on the i^* framework, its application in Requirements Engineering, and its implications on software architecture. Since 2004, when this research started, the group started different research threads. The first stage focused on the deep understanding of the syntax and semantics of the i^* framework. It was the time when the group reported systematically the particularities of the different variants supported by different explicit or underlying metamodels, which used and use different syntax rules and had and have variations on semantics and pragmatics. The first reports in the community talking about differences in i^* variants, modelling them and comparing them, come from this research group, which the author of this thesis belonged to from the very beginning of this work, 2004 (date of publication of the first joint paper in a regional conference).

A second step, built on the previous facts, was to question the tacit agreement that interoperability among i^* variants could not be built on a scenario in which these variants had syntactical, semantic and pragmatic differences. The idea of having a common core pushed the belief that, at least, some kind of interoperability may be reached. Therefore, we formulated this thesis objective as *to propose a framework to understand the variations of the i^* modelling language, and to generate a proposal to support the interoperability and integration of these variations.*

In order to attain this objective, we formulated a research project based on different questions and a mixed methodological approach. Consequently, we have arrived to conclusions which we have grouped as follows: answers to research questions and findings; conclusions about the interoperability approach; conclusions about the methodological point of view; and, finally, future work starting from the reached point at theoretical and technological levels.

6.1 Answers to Research Questions and Findings

A set of research questions was formulated according to three phases, namely: (i) understanding and characterizing the problem; (ii) approaching an interoperability solution; and (iii) evaluating this solution. In this section we review the phases (PH), the research questions (RQ), and the findings and theoretical approaches which have been presented as initial answers (IA) to them. In Table 6.1 we present a summary of results and next we explain each row.

Table 6.1. Summary of research questions and their approached answers

Research Phase (PH)	Research Question (RQ)	Initial Answer
PH.1. Understanding and characterizing the problem	RQ.1. What i^* variations are proposed and how can they be characterized?	Recognizing leading variants, characterizing differences by changes proposed (Section 2.2)
	RQ.2. Which is an appropriate framework for analysing i^* differences and similarities?	Metamodel-based characterization of variants and metamodel transformations (Section 2.3)
	RQ.3. Why an small community has generated different i^* variants? And, what are the implications?	Ontological evolution is a feature of Khun's paradigmatic view of science progress. Adaptations of theoretical frameworks happen on Bourdieu's theory of scientific fields (Section 3.2.1)
	RQ.4. What are the fundamentals concepts which could guide us to an interoperability approach?	Semiotics and Cybernetics provides us of communication approaches dealing with ontological variations (Section 3.2.2 and 3.2.3)
PH.2. Approaching to interoperability for i^* variants	RQ.5. Is it possible to formulate an interoperability framework which can support future i^* variants?	We answered affirmatively this question using semiospheres and the theory of scientific fields (Section 3.3)
	RQ.6. How to deal with semantic differences and to characterize lost of semantic if that occurs?	Wachsmuth's framework allows characterizing semantic preservation under a metamodel approach (Section 3.3)
	RQ.7. Which is the most efficient form that this semantic framework may take?	Wachsmuth's concept of supermetamodel allows a lineal behaviour to the quadratic problem of translations (Section 3.3). A semiosphera shape is proposed (Section 3.3) and implemented (Chapter 4)

Research Phase (PH)	Research Question (RQ)	Initial Answer
PH.3. Evaluating the proposed interoperability framework	RQ.8. Is it possible to implement the formulated semantic framework in the current scenario of i^* -related tools?	A couple of proofs of concepts illustrate the feasibility of the proposed interoperability approach (Section 5.1)
	RQ.9. Is there a perception in the i^* community about an interoperability problem? and how acceptable is our approach to it?	We have run a survey in the core of the i^* community. There is evidence that our approach is acceptable (Section 5.3) and also it is being adopted (Section 5.2.1)
	RQ.10. Are there other applications that could bring a positive evaluation of the proposed interoperability framework?	We show additional applications contextualized to software process (Section 5.2)

PH.1. Understanding and characterizing the problem.

RQ.1. What i^* variations are proposed and how can they be characterized?

IA.1. In Chapter 2 we have presented a genealogy of variants, we have detailed three of the relevant i^* variants and we have used metamodels to characterize them. We have recently updated these studies and we have classified the type and amount of i^* variants proposals.

RQ.2. Which is an appropriate framework for analysing i^* differences and similarities?

IA.2. Representing the variants in metamodels allows not only studying their differences but also having a measure of the differences among them, e.g., the amount of refactoring operations needed to derive a particular i^* variant from a reference metamodel.

RQ.3. Why it happens? In other words, why an apparently small research community has generated so many different i^* variants? And, what are the implications?

I.A.3. We have presented Khun's theoretical framework which seems to explain that when a scientific revolution occurs (no matter how big it is) a new ontology evolves and active research is an expected behaviour. The main implication is that, when the ontology matures, the revolutionary stage

stops and a normal science period arrives. A normal feature of this revolutionary stage is not having a semantic agreement about the evolved conceptual framework.

RQ.4. What are the fundamentals concepts which could guide us to an interoperability approach?

I.A.4. Fundamental concepts sustaining the feasibility of an interoperability framework come from different approaches: the contemporary concept of semantics from philosophy of language; intelligence amplifiers from cybernetics; structural coupling from neo cybernetics; semiospheres from semiotics; theory of scientific fields. We have used some of these principles to support and design our interchange proposal.

PH.2. Approaching to interoperability for i^* variants.

RQ.5. Is it possible to formulate an interoperability framework which includes structures to support future i^* variants?

I.A.5. We answered affirmatively this question. We sustained (following semiospheres and the theory of scientific fields) that evolved concepts are based on core concepts, therefore, we can add for example new types of intentional elements, new types of intentional links and, at least, the readers will know that these elements are a type of intentional element and a type of intentional link. However formulating new concepts that cannot be explained using the existing ones, will remain a problem.

RQ.6. How to deal with semantic differences and to characterize loss of semantic if that occurs?

I.A.6. To answer this question we have used Waschmuth's metamodel approach. It models semantic preservation on base of instance preservation. We have presented an isomorphism between this proposal and the interoperability among different i^* variants. Moreover, we have presented a general transformation algorithm which allows characterizing the semantic preservation of each corresponding translation.

RQ.7. Which is the most efficient form that this semantic framework may take?

I.A.7. Waschmuth's framework introduces the concept of supermetamodel which we have used to assume that a general i^* metamodel is possible. It basically implies that all existing i^* concepts are part of this supermetamodel. Assuming this, the quadratic problem of proposing translations (which imply interpretations) between every pair of existing i^* variants is reduced to a linear one, which only needs to propose a translation from the i^* supermetamodel to the specific metamodel of each variant.

PH.3. Evaluating the proposed interoperability framework.

RQ.8. Is it possible to implement the formulated semantic framework in the current scenario of i^* -related tools?

I.A.8. We have tackled the technical problem by some proof of concepts which show the technical feasibility of the proposal and also the semantic consequences of the translations. A broader application of the proposal will depend on the community and how to solve other technical problems, as the unavailability of some tool source code.

RQ.9. Is there a perception in the i^* community about an interoperability problem? And, how acceptable is our approach for it?

I.A.9. We have run a survey in the core of the i^* community. In spite of being a small sample, since the community itself is also small, we have got statistical confidence about the perception on our interchange proposal. Furthermore, our proposal presents favourable opinions beyond normal curves of technological adoptions.

RQ.10. Are there other applications that could bring a positive evaluation of using the proposed interoperability framework?

I.A.10. We have affirmatively answered this question. We have proposed an interoperability approach which includes the iStarML format. We have also included a set of other applications on our textual representation of i^* models such as metric calculation and model validation beyond a generic technological support for handling iStarML files.

Given the above questions and answers, and mainly founded on findings and the produced interoperability approach, we sustain that the research objectives have been reached, feasible and founded answers have been offered to research questions. Moreover, a community perception and proof of concepts have been offered as proposal's validation.

6.2 Theoretical Conclusions

From the theoretical point of view, we have reviewed a set of proposals coming from Linguistics, Cybernetics, and Philosophy of Science. From Linguistics, we have reviewed the semiotic proposal from Lotman, which sustains that monosemantic scenarios do not exist in isolation and it materializes the polysemantic dynamics under the concept of semiosphere. This proposal has given us the main ideas sustaining interoperation feasibility and also has given some ideas about the structure which could present an interchange language. The idea of interoperation feasibility has been also supported by the cybernetic perspective of biological communication, where it is affirmed that interoperation means a dynamic structural coupling where ontologies are internal and dynamic, therefore, the original assumption of having a common external ontology in order to reach high levels of interoperations do not seems to be a valid sentence. Other philosophy of science's positions, such Kuhn's theory of scientific revolutions, seems to be a good model to explain that a normal situation in paradigmatic shifts is to have an evolving and fuzzy ontology, therefore, the particular situation of the i^* community seems to be just another case of a phenomena already described. From the perspective of scientific behaviour, Bourdieu's theory of scientific fields also seems to explain the phenomena of adapting conceptual frameworks to different fields which is possible by relaxing or reinterpreting core concepts from a specific scientific field to a third one.

Moreover, we have reviewed a contemporary theory of semantics and its special meanings when we talk about scientific representational languages. Bunge's framework has constituted a useful formalization for handling the difference among semantics,

pragmatics and related concepts such as ontology, denotation, scientific language among others. We have used it to express some intermediate explanations and conclusions. Finally, in terms of semantic preservation, we have used Waschmuth's framework which offers a set of distinctions for characterizing the eventual semantic loss (if any) for model translations corresponding to different i^* variants. From this theoretical perspective the idea of proposing an i^* supermetamodel, i.e. a metamodel which is able of containing the different i^* variants metamodels, has emerged, which seems to be a key issue to improve our interoperability proposal.

Finally, from a theoretical point of view, we have tackled a problem in the Requirements Engineering community, specifically on the variants of a relevant modelling language, not only due to its usage but also because it corresponds to a contemporary and relevant perspective of analysis as is goal orientation. Moreover we have used novel and traditional theoretical elements from computer science such as languages and metalanguages, models and metamodels, model transformations and metamodel adaptations, and supermetamodels. Moreover, the foundations of our proposal have been supported by theoretical frameworks outside of the limits of computer science in a coherent and comprehensible way. From these external theories we have arrived to our computer science technological proposal, however, we see that additional interesting applications would be reach if we use these theoretical approaches in computer purposes related to models interoperability.

6.3 Technological Conclusions

From a technological perspective we have developed iStarML, the only existing proposal for interoperability in the i^* community. We have developed a set of technological tools such as parsers, translators, and a generic library for handling iStarML files. The general idea has been tried in different proof of concepts as part of this thesis work. Moreover, different tools have already included, are including, or have planned to include, iStarML as the interchange format for i^* models.

Therefore, we can sustain that the chosen technology has been adequate to the problem, we have produced a specification which, being founded on a solid theoretical background, delivers a practical and useful proposal which accounts not only with proof of concepts but also with real implementations on innovative software engineering software tools.

6.4 *Epistemological Conclusions*

Now, from an epistemological point of view, we have followed two traditional Software Engineering methodologies, such as proof of concept and conceptual analyses. But also we have followed classic qualitative techniques as descriptive analysis and, moreover, we have added a quantitative study which has been highly recommended by contemporary epistemological tendency in Software Engineering. Although we have just briefly included epistemological discussion, this thesis accomplishes contemporary epistemological recommendations of looking for different and mixed approaches to knowledge production, not following given prescriptions but reflecting about the topic, being open to react to intermediate findings, mixing qualitative and quantitative approaches according the particular research situation, and being aware of potential epistemological lacks in order to tackle new research stages.

Therefore, our epistemological approach has been multi-methodological, it has combined quantitative and qualitative approaches, it has followed different research stages, intermediate findings allowed redefining and adapting research plans. The effect of this epistemological approach can be noted in the different dates of studies that we have partially showed in this document.

In spite of that, we have not lost the engineering focus, because we have not only generated new knowledge; we have formulated, tried, and validated a technological approach, which allows classifying our research proposal as research and development, which accomplishes classical research in engineering fields.

6.5 Future Work

Many research threads have been opened by this thesis work, some technical and some theoretical. On the technical part, developing additional proof of concepts about iStarML seems necessary, not only to show feasible translations, but also in order to make visible other applications over iStarML representations. An open issue here is to use iStarML for translations to other software engineering representations. It would enable software processes which using some variant of i^* , can incorporate requirements models to some model driven development (MDD) approach.

Other perspective of analysis is based on the fact that more and more, software engineering work tends to be geographically distributed and carried out by distributed teams, therefore, i^* modelling co-design seems to be easier to implement having textual representations. However, other language elements could be needed in order to facilitate engineering communication. Therefore, a revised version of iStarML, including analysis engineering marks, would be a novel topic to develop. Anyway, other basic inclusions would be first developed such as module, inheritance marks, and additional graphical features to support concepts coming from other software engineering subfields.

From the theoretical point of view, there are research paths very interesting to develop. For example, the formalization of interoperability levels for polysemantic scenarios seems very interesting. To us, it seems clear that pragmatics would be better defined, because interoperation, i.e. *working together*, is a matter of behaviours more than matters of meanings. Tackling this problem would allow modelling the different software engineering behaviours acting under model driven approaches, which would improve model reuse and design performance at requirements stage.

In the specific case of i^* semantics and pragmatics, the fact of having studied the different i^* variants and their applications, gave us some background in order to participate in groups defining semantic integration on existing i^* variants and the

conceptual frameworks to link i^* representations with other organizational models. The problem here is not only technical in the sense of enabling experimenting with models transformations under an MDD approach, but also to propose the ontological convergence among organizational and technical representation, measuring e.g. the impact of the different cases of semantic preservation. Some initial work in this direction have been already started by the GESSI research group in both lines, the first one that we could call semantics alignment [FRAN'11b] and the second one on providing an ontological view to the means-end construct [LOPE'11].

6.6 Final Words

Finally, we want to say that scientific work should be never end, therefore it is hard to sustain that we have effectively solved some problem, we prefer the view that we have just advanced a little bit in a very specific research path. In this research path, as in every single other path, technological products represent milestones implying that the research has reached a non-return point. We see that iStarML 1.0 is a non-return point in the way of interoperability proposals aware of polysemantic and multi-ontological scenarios. Moreover, it appears as a very interesting scientific topic to be developed, having implications on computer supporting collaborative work, particularly on computer supporting collaborative software engineering and requirements engineering. We hope having the conditions to keep us on this line.

References

- [ABRA '04] Abran, A., J. W. Moore, P. Bourque and R. Dupuis, Eds.: Guide to the Software Engineering Body of Knowledge. Los Alamitos, CA, Washington, Brussels, Tokyo, IEEE Computer Society Press (2004)
- [ALEN '08] Alencar, F., J. Castro, et al.: Integration of Aspects with *i** Models, Lecture Notes in Computer Science (AOIS'08), vol 4898 pp. 183-201 (2008)
- [ALEN '09] Alencar, F., B. Marín, et al.: From *i** Requirements Models to Conceptual Models of a Model Driven Development Process. Proc. of the 2nd IFIP WG 8.1 Working Conference on The Practice of Enterprise Modeling (PoEM'09). Stockholm, Sweden, Nov 18-19 (2009)
- [ALON '04] Alonso, F., S. Frutos, L. Martínez and C. Montes: Towards a Natural Agent Paradigm Development Methodology, MATES 2004. Lecture Notes in Computer Science, vol 3187 pp. 155-168 (2004)
- [AMYO '02] Amyot, D. and G. Mussbacher: URN: Towards a New Standard for the Visual Description of Requirements. Telecommunications and beyond: The Broader Applicability of SDL and MSC, (SAM 2002).Aberystwyth, UK, June 24-26 (2002)
- [AMYO '09] Amyot, D., J. Horkoff, D. Gross and G. Mussbacher: A Lightweight GRL Profile for *i** Modeling. In Advances in Conceptual Modeling - Challenging Perspectives, LNCS, vol 5833 pp 254-264 (2009)
- [ASHB '57] Ashby, W. R.: An Introduction to Cybernetics. London, Chapman & Hall LTD. (1957)
- [AYAL '05] Ayala, C. P., C. Cares, et al.: A Comparative Analysis of *i**-Based Goal-Oriented Modelling Languages. Int. Workshop on Agent-Oriented Software Development Methodology, at the SEKE Conference. Taipei, Taiwan, July (2005)
- [BASI '86] Basili, V. R., R. W. Selby and D. H. Hutchens: Experimentation in Software Engineering, IEEE Transactions on Software Engineering, vol 12 pp. 733-743 (1986)
- [BASI '93] Basili, V. R.: The Experimental Paradigm in Software Engineering, Lecture Notes in Computer Science, vol 706, pp. 1-12 (1993)
- [BASI '96] Basili, V. R.: The Role of Experimentation in Software Engineering: Past, Current, and Future. Proc. of the 18th Int. Conference on Software Engineering (ICSE'96), March 25-29, Berlin, Germany, pp. 442-449, IEEE Computer Society Press (1996)
- [BEER '67] Beer, S.: Cybernetics and Management. London, English Universities P. (1967)
- [BERT '07] Bertolini, D., A. Novikau, A. Susi and A. Perini FBK-irst: TAOM4E: an Eclipse Ready Tool for Agent-Oriented Modeling. Issue on the Development Process Retrieved March 2008, from http://sra.itec.it/tools/taom4e/file/demo//TAOM4E_technical_report.pdf (2006)
- [BIES '06] Bieser, T., H. Fürstenau, S. Otto and D. Weiß: Requirements Engineering. In Multiagent Engineering. S. Kirm, O. Herzog, P. Lockemann and O. Spaniol, Springer, Berlin Heidelberg, pp 359-381 (2006)
- [BOUR '75] Bourdieu, P.: The Specificity of the Scientific Field and the Social Conditions of the Progress of Reason, Social Science Information, vol 14 (6) (1975)
- [BRAY '08] Bray, T., J. Paoli, et al., Eds.: Extensible Markup Language (XML) 1.0 (Fifth Edition), World Wide Web Consortium (W3C) obtained from <http://www.w3.org/TR/2008/REC-xml-20081126/> (2008)
- [BRES '04] Bresciani, P., A. Perini, et al.: Tropos: An Agent-oriented Software Development Methodology, Autonomous Agents And Multi-Agent Systems, vol 8 (3) pp. 203-236 (2004)
- [BROW '01] Brown, L. D., T. T. Cai and A. DasGupta: Interval Estimation for a Binomial Proportion, Statistical Science, vol 16 (2) pp. 101-133 (2001)

- [BUHR '96] Buhr, R. J. A. and R. S. Casselman: Use Case Maps for Object-Oriented Systems, Prentice Hall (1996)
- [BUNG '08] Bunge, M.: Semántica I: Sentido y Referencia. Barcelona, GEDISA (2008)
- [CARE '07] Cares, C., X. Franch, A. Perini and A. Susi: iStarML Reference Guide, LSI Technical Report. ID: LSI-07-46-R, <http://www.lsi.upc.edu/~techreps/files/R07-46.zip>, (2007)
- [CARE '09] Cares, C. and X. Franch: Towards a Framework for Improving Goal-Oriented Requirements Models Quality. In Proc. of the 12th Workshop on Requirements Engineering (WER09), pp. 3-14. Valparaiso, Chile, July 16-17 (2009)
- [CARE '10] Cares, C., X. Franch, L. López, and J. Marco.: Definition and Uses of the *i** Metamodel. Proc. of the 4th Int. Workshop on *i** (iStar10), CEUR Workshop Proceedings, vol. 586 pp. 20-25, Hammamet, Tunisia, (2010)
- [CARE '11a] Cares, C. and X. Franch: A Metamodelling Approach for *i** Model Translations. In Proc. of the Int. Conf. on Advanced Information Systems Engineering (CAiSE). H. Mouratidis and C. Rolland, LNCS, vol. 6741 pp. 337-351, Berlin / Heidelberg, Springer, (2011)
- [CARE '11b] Cares, C. and D. Colomer The OME to iStarML translator Retrieved Jun 2011, from http://www.upc.edu/gessi/istarm/ometoistarm/remoto/online_tools.html
- [CARE '11c] Cares, C., X. Franch, E. Mayol and C. Quer: A Reference Model for *i**. In Social Modeling for Requirements Engineering. E. Yu, P. Giorgini, N. Maiden and J. Mylopoulos, Eds. Cambridge, MA, USA, London, England, MIT Press, pp 573-606 (2011)
- [CARE '11d] Cares, C. CCiStarML (v0.6), a Java Package for Handling iStarML files, Retrieved Oct 2011, from <http://www.essi.upc.edu/~gessi/iStarML/>
- [CARE '11e] Cares, C. and X. Franch: iStarML: Principles and Implications. CEUR Proceedings of the 5th International *i** Workshop (iStar 2011), 766 pp. 8-13, Trento, Italy, Aug 29-30 (2011)
- [CAST '01] Castro, J., M. Kolp and J. Mylopoulos: A Requirements-Driven Development Methodology, Proc. of the 13th Int. Conf. on Advanced Information Systems Engineering, CAiSE 2001, Interlaken, Switzerland, pp. 108-123 (2001)
- [CERN '05] Cernuzzi, L., M. Cossentino and F. Zambonelli: Process Models for Agent-based Development, Engineering Applications of Artificial Intelligence, vol 18 pp. 205–222 (2005)
- [CHEN '07] Cheng, B. H. C. and J. M. Atlee: Research Directions in Requirements Engineering. Proc. of Future of Software Engineering (FOSE 2007), pp. 285-303, Minneapolis, Minnesota, May 20-26, IEEE Computer Society, Washington, DC, USA (2007)
- [CHUN '00] Chung, L. K., B. A. Nixon, E. Yu and J. Mylopoulos: Non-functional Requirements in Software Engineering, Kluwer Academic Publishing (2000)
- [CHUN '09] Chung, L. and J. C. S. d. P. Leite: On Non-Functional Requirements in Software Engineering, Lecture Notes in Computer Science / Conceptual Modeling: Foundations and Applications, vol 5600 pp. 363-379 (2009)
- [CLAR '99] Clark, J. and S. DeRose World Wide Web Consortium (W3C): XML Path Language (XPath) Version 1.0, Retrieved March 2008, from <http://www.w3.org/TR/xpath> (2009)
- [COLO '11] Colomer, D., L. Lopez, C. Cares and X. Franch: Model Interchange and Tool Interoperability in the *i** Framework: A Proof of Concept. Proc of the 14th Workshop on Requirements Engineering (WER'11). Rio de Janeiro, Brasil, April 27-29, pp. 369-381 (2011)
- [DAM '03] Dam, K. H. and M. Winikoff: Comparing Agent-oriented Methodologies. In Agent-Oriented Information Systems., Springer-Verlag, Berlin, vol 3030 pp. 78-93 (2003)
- [DARD '91] Dardenne, A., S. Fickas and A. v. Lamswerde: Goal-directed Concept Acquisition in Requirements Elicitation. Proc. of the 6th Int. Workshop on Software Specification and Design, pp. 14-21. Como, Italy, Oct 25-26 (1991)

- [DARD '93] Dardenne, A., A. v. Lamsweerde and S. Fickas: Goal-directed Requirements Acquisition, *Science of Computer Programming*, vol 20 (1-2) pp. 3-50 (1993)
- [DONG '00] Dongwon, L. and W. C. Wesley: Comparative Analysis of Six XML Schema Languages, *SIGMOD Rec.*, vol 29 (3) pp. 76-87 (2000)
- [DONZ '02] Donzelli, P. and R. Setola: Handling the Knowledge Acquired During the Requirements Engineering Process - a Case Study -. *Proc. of the 14th Software Engineering and Knowledge Engineering Conference, SEKE'02*, pp. 673-679. Ischia, Italy, July 15-19 (2002)
- [DONZ '04] Donzelli, P. and P. Bresciani: Improving Requirements Engineering by Quality Modelling – A Quality-Based Requirements Engineering Framework, *Journal of Research and Practice in Information Technology*, vol 36 (4) pp. 277-294 (2004)
- [ECO '00] Eco, U.: *Tratado de Semiótica General*. Barcelona, Lumen. (2000)
- [ENGE '74] Engelfriet, J.: *Simple Program Schemes and Formal Languages*. Berlin, New York, Springer-Verlag (1974)
- [ERL '05] Erl, T.: *Service-Oriented Architecture: Concepts, Technology, and Design*. NJ, USA, Prentice Hall (2005)
- [ESFA '10] Esfahani, H. C., J. Cabot and E. Yu: Adopting Agile Methods: can Goal-Oriented Social Modeling help? *Proc. of the 4th Int. Conf. on Research Challenges in Information Science (RCIS'10)*, pp. 223-234, Nice, France, May 19-21 (2010)
- [ESTR '08] Estrada, H.: *A Service-Oriented Approach for the *i** Framework*. PhD. Thesis, Information Systems and Computation Department, University of Valencia, Spain (2008)
- [ESTR '10] Estrada, H., A. Martínez, et al.: Extending Organizational Modeling with Business Services Concepts: An Overview of the Proposed Architecture, *Lecture Notes in Computer Science (ER'10)*, vol 6412 pp. 483-488 (2010)
- [FRAN '06] Franch, X.: On the Quantitative Analysis of Agent-Oriented Models. *Proc. of the 18th Conference on Advanced Information Systems Engineering (CAISE'06)*, pp. 495-509. Luxembourg, Grand-Duchy of Luxembourg, Jun 5-6 (2006)
- [FRAN '10] Franch, X.: Fostering the Adoption of *i** by Practitioners: Some Challenges and Research Directions. In *Intentional Perspectives on Information Systems Engineering*. S. Nurcan, C. Salinesi, C. Souveyet and J. Ralyté. Berlin Heidelberg, Springer, pp 177-193 (2010)
- [FRAN '11a] Franch, X., R. Guizzardi, G. Guizzardi and L. López: Ontological Analysis of Means-End Links. *CEUR Proceedings of the 5th International *i** Workshop (iStar 2011)*, vol 766 pp. 37-42. Trento, Italy, Aug 29-30 (2011)
- [FRAN '11b] Franch, X., A. Maté, J. Trujillo and C. Cares: On the joint use of *i** with other Modelling Frameworks: a Vision Paper. *Proc. of the 19th IEEE International Requirements Engineering Conference (RE'11)*. Trento, Italy, Aug 29- Sept 2 (2011)
- [FUXM '01] Fuxman, A., P. Giorgini, M. Kolp and J. Mylopoulos: Information Systems as Social Structures. *Proc. of the 2nd Int. Conference on Formal Ontologies for Information Systems (FOIS01)*. Ogunquit, USA, Oct 17-19 (2001)
- [GAIL '08] Gailly, F., S. España, G. Poels and O. Pastor: Integrating Business Domain Ontologies with Early Requirements Modelling, *Lecture Notes in Computer Science (RIGiM'08)*, vol 5232 pp. 282-291 (2008)
- [GIOR '03] Giorgini, P.: Agent-Oriented Software Engineering Report on the 4th AOSE Workshop (AOSE 2003), *SIGMOD Record*, vol 32 (4) pp. 117-118 (2003)
- [GIOR '04a] Giorgini, P., F. Massacci, J. Mylopoulos and N. Zannone: Requirements Engineering Meets Trust Management, Model, Methodology, and Reasoning, *Lecture Notes in Computer Science*, vol 2995 pp. 176-190 (2004)
- [GIOR '04b] Giorgini, P., F. Massacci, J. Mylopoulos and N. Zannone: Filling the Gap between Requirements Engineering and Public Key/Trust Management Infrastructures, *Lecture Notes in Computer Science*, vol 3093 pp. 98-111 (2004)

- [GIOR '05] Giorgini, P., S. Rizzi and M. Garzetti: Goal-oriented Requirement Analysis for Data Warehouse Design. Proc. of the 8th ACM international workshop on Data Warehousing and OLAP (DOLAP'05), pp. 47-56. Bremen, Germany, Oct 31 - Nov 5 (2005)
- [GIOR '11] Giorgini, P., J. Mylopoulos, A. Perini and A. Susi: The Tropos Methodology and Software Development Environment. In Social Modeling for Requirements Engineering. E. Yu, P. Giorgini, N. Maiden and J. Mylopoulos Eds. Cambridge, MA, USA, MIT Press (2011)
- [GIUN '03] Giunchiglia, F., J. Mylopoulos and A. Perini: The Tropos Software Development Methodology: Processes, Models and Diagrams, AOSE 2002 Lecture Notes in Computer Science, vol 2585 pp. 162-173 (2003)
- [GLAS '02] Glass, R. L., I. Vessey and V. Ramesh: Research in Software Engineering: an Analysis of the Literature, Information and Software Technology, vol 44 (8) pp. 491-506 (2002)
- [GOB '07] Göb, R., C. McCollin and M. F. Ramalhoto: Ordinal Methodology in the Analysis of Likert Scales, Quality & Quantity, vol 41 pp. 601-626 (2007)
- [GOME '04] Gómez-Sanz, J., M.-P. Gervais and G. Weiss: A Survey on Agent-Oriented Software Engineering Research. In Methodologies and Software Engineering for Agent Systems. G. Weiss, Springer US, vol 11 pp 33-62 (2004)
- [GRAU '06] Grau, G., X. Franch and S. Ávila: J-PRiM: A Java Tool for a Process Reengineering *i** Methodology. Proc. of the 14th IEEE International Requirements Engineering Conference (RE'06), pp. 359-360. Minneapolis, MN, USA, Sept. 11-15 (2006)
- [GRAU '07] Grau, G. and X. Franch: A Goal-Oriented Approach for the Generation and Evaluation of Alternative Architectures, Lecture Notes in Computer Science (RIGiM'07), vol 4758 pp. 139-155 (2007)
- [GRL '01] GRL - Goal Oriented Requirement Language Retrieved September 20, 2010, from <http://www.cs.toronto.edu/km/GRL/> (2001)
- [HARE '04] Harel, D. and B. Rumpe: Meaningful Modeling: What's the Semantics of "Semantics"? IEEE Computer, vol 37 (10) pp. 64-72 (2004)
- [HOFE '07] Höfer, A. and W. F. Tichy: Status of Empirical Research in Software Engineering, Lecture Notes in Computer Science, vol 4336 pp. 10-19 (2007)
- [HOPE '69] Hopcroft, J. E. and J. D. Ullman: Formal Languages and their Relation to Automata. Reading, Mass., Addison-Wesley Pub. Co. (1969)
- [HORK '11] Horkoff, J., Y. Yu and E. Yu: OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool. CEUR Proceedings of the 5th International *i** Workshop (iStar 2011), vol 766 pp. 154-156. Trento, Italy, Aug 29-30 (2011)
- [HUMP '00] Humphrey, W. S.: The Team Software Process Report CMU/SEI-2000-TR-023, obtained from <http://www.se.cmu.edu/reports/10tr020.pdf> (2000)
- [ISML] The iStarML Web Site Retrieved October, 2011, from <http://www.essi.upc.edu/~gessi/iStarML/>.
- [ITUT '08] ITU-T Recommendation Z.151 (11/08), User Requirements Notation (URN) - Language Definition, from <http://www.itu.int/rec/T-REC-Z.151/en>
- [IWIK] The *i** wiki Retrieved July, 2011, from <http://istar.rwth-aachen.de>
- [JARK '11] Jarke, M., P. Loucopoulos, et al.: The Brave New World of Design Requirements, Information Systems, vol 36 pp. 992-1008 (2011)
- [JELL '00] Jelliffe, R. Academia Sinica Computing Centre: The Schematron Assertion Language 1.5 Retrieved September 2007, from <http://xml.ascc.net/resource/schematron/Schematron2000.html> (2000)
- [JENN '98] Jennings, N. R., K. Sycara and M. Wooldridge: A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems, vol 1 (1) pp. 7-38 (1998)
- [JENN '00] Jennings, N. R.: On Agent-based Software Engineering, Artificial Intelligence, vol 117 (2) pp. 277-296 (2000)

- [JIAN' 06] Jiang, L., T. Topaloglou, A. Borgida and J. Mylopoulos: Incorporating Goal Analysis in Database Design: A Case Study from Biological Data Management. Proc. of the 14th IEEE International Requirements Engineering Conference (RE'06), pp. 199-207. Minneapolis/St. Paul, Minnesota, USA, Sept 11-15 (2006)
- [JUCM] The jUCMNav website, from <http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/>
- [KAIS '05] Kaisler, S. H.: Software Paradigms. Hoboken, New Jersey, John Wiley & Sons Inc. (2005)
- [KAVA '02] Kavakli, E.: Goal Oriented Requirements Engineering: A Unifying Framework, Informática, vol 6 (4) pp. 237-251 (2002)
- [KICZ '97] Kiczales, G., J. Lamping, et al.: Aspect-Oriented Programming., Lecture Notes in Computer Sciences (ECOOP'97), vol 1241 pp. 220-242 (1997)
- [KITC '02] Kitchenham, B. A., S. L. Pleeger, et al.: Preliminary Guidelines for Empirical Research in Software Engineering, IEEE Transactions on Software Engineering, vol 28 (8) pp. 721-734 (2002)
- [KITC '04] Kitchenham, B. A., T. Dyba and M. Jorgensen: Evidence-based Software Engineering. Proc. of the 26th International Conference on Software Engineering (ICSE'04), pp. 273-281 (2004)
- [KRIS '09] Krishna, A., S. A. Vilkomir and A. K. Ghose: Consistency preserving Co-evolution of Formal Specifications and Agent-oriented Conceptual Models, Information and Software Technology, vol 51 (2) pp. 478-496 (2009)
- [KROG '95] Krogstie, J., O. I. Lindland and G. Sindre: Towards a Deeper Understanding of Quality in Requirements Engineering, Lecture Notes in Computer Science (CAiSE'95), vol 932 pp. 82-95 (1995)
- [KROG '06] Krogstie, J., G. Sindre and H. Jørgensen: Process Models Representing Knowledge for Action: a Revised Quality Framework, European Journal of Information Systems, vol 15 pp. 91-102 (2006)
- [KUHN '96] Kuhn, T. S.: The Structure of Scientific Revolutions, The University of Chicago Press (1996)
- [LAMS '00] Lamsweerde, A. v.: Requirements Engineering in the Year 00: A Research Perspective. Proc. of the Int. Conf. on Software Engineering, ICSE 2000, pp. 5-19. Limerick, Ireland, Jun 4-11 (2000)
- [LAMS '01] Lamsweerde, A. v.: Goal-Oriented Requirements Engineering. Proc. of the 5th IEEE Int. Symposium on Requirements Engineering (RE'01), pp. 249-262. Toronto, Canada, 27-31 August (2001)
- [LAMS '08] Lamsweerde, A. v.: Requirements Engineering: from Craft to Discipline. Proc. of the 16th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 238-249. Atlanta, Georgia (2008)
- [LAUE '11a] Laue, R. and A. Storch: A Flexible Approach for Validating *i** Models. CEUR Proceedings of the 5th International *i** Workshop (iStar 2011), vol 766 pp. 32-36. Trento, Italy, Aug 29-30 (2011)
- [LAUE '11b] Laue, R. and A. Storch: Adding Functionality to OpenOME for Everyone. CEUR Proceedings of the 5th International *i** Workshop (iStar 2011), vol 766 pp. 169-171. Trento, Italy, Aug 29-30 (2011)
- [LETI '02] Letier, E. and A. v. Lamsweerde: Agent-Based Tactics for Goal-Oriented Requirements Elaboration. Proc of the 24th International Conference on Software Engineering (ICSE '02), pp. 83-94. Orlando, Florida, May 19-25 (2002)
- [LIND '01] Lind, J.: Issues in Agent-Oriented Software Engineering, Agent-Oriented Software Engineering: First International Workshop, AOSE 2000, Limerick, Ireland, June 10, 2000. LNCS, vol 1957 pp. 45-58 (2001)

- [LI '11] Li, T., T. Li, H. Xie, and L. Liu: I*-Prefer: A Tool for Preference Model-Driven Decision Making. In Proc. of the 5th. Int. i* Workshop (iStar'11), CEUR-Workshop vol 766 pp. 151-153, Trento, Italy, Aug 29-30 (2011)
- [LIU '06] Liu, L., C. Chi, Z. Jin and E. Yu: Strategic Capability Modelling of Services. Conference. Proc. of the 2nd International Workshop on Service-Oriented Computing: Consequences for Engineering Requirements (SOCCER'06), pp. 3-15. Minneapolis, MN, USA, Sept 12 (2006)
- [LOCK '08] Lockerbie, J. and N. Maiden: REDEPEND: Tool Support for i* Modelling in Largescale Industrial Projects. Proc. of the Forum at the CAiSE'08 Conference, Montpellier, France, Jun 18-20, CEUR Workshop Proceedings, vol 344 pp. 69-72. (2008)
- [LOPE '08] López, L., X. Franch and J. Marco: Defining Inheritance in i* at the Level of SR Intentional Elements. Proc. of the 3rd i* Int. Workshop, CEUR Workshop Proceedings, 322 pp. 71-74, (2008)
- [LOPE '09] López, L., X. Franch and J. Marco: HiME: Hierarchical i* Modeling Editor, Revista de Informática Teórica e Aplicada (RITA), vol 16 (2) pp. 57-60 (2009)
- [LOPE '11] López, L., X. Franch and J. Marco: Making Explicit Some Implicit i* Language Decisions, Lecture Notes in Computer Sciences (ER11), vol 6998 pp. 62-77 (2011)
- [LOTM '03] Lotman, J.: On the semiosphere, Sign Systems Studies, vol 33 (1) (2003)
- [LUCE '08] Lucena, M., E. Santos, et al.: Towards a Unified Metamodel for i*. Proc. of the IEEE Int. Conf. on Research Challenges in Information Science, (RCIS 2008), pp. 237–246. Marrakech, Morocco, June 3-6 (2008)
- [MALT '11] Malta, Á., M. Soares, et al.: iStarTool: Modeling Requirements using the i* Framework. CEUR Proceedings of the 5th International i* Workshop (iStar 2011), vol 766 pp. 163-165. Trento, Italy, Aug 29-30 (2011)
- [MANS '04] Mansell, R.: Political Economy, Power and New Media. New Media & Society, vol 6 (1) pp. 74-83 (2004)
- [MAO '05] Mao, X. J. and E. Yu: Organizational and Social Concepts in Agent Oriented Software Engineering, Lecture Notes in Computer Science, vol 3382 pp. 1-15 (2005)
- [MART '06] Martín-Barbero, J.: A Latin American Perspective on Communication/Cultural Mediation, Global Media and Communication, vol 2 (3) pp. 279-297 (2006)
- [MASS '07] Massacci, F., J. Mylopoulos and N. Zannone: An Ontology for Secure Socio-Technical Systems. In Handbook of Ontologies for Business Interaction. P. Ritggen, IDEA Group, (2007)
- [MATU '98] Maturana, H. R. and F. J. Varela: De Máquinas y Seres Vivos: Autopoiesis la Organizacion de lo Vivo. Santiago de Chile, Editorial Universitaria (1998)
- [McNA '82] McNaughton, R.: Elementary Computability, Formal Languages, and Automata. Englewood Cliffs, N.J., Prentice-Hall (1982)
- [MEYE '85] Meyer, B.: Formalism in Specifications, IEEE Software, vol 2 (1) pp. 6-26 (1985)
- [MORA '11] Morandini, M., C. D. Nguyen, et al.: Tropos Modeling, Code Generation and Testing with the TAOM4E Tool. CEUR Proceedings of the 5th International i* Workshop (iStar 2011), vol 766 pp. 172-174. Trento, Italy, Aug 29-30 (2011)
- [MOUR 2002] Mouratidis, H., P. Giorgini, G. Manson and I. Philp: A Natural Extension of Tropos Methodology for Modelling Security. Proc. of the Agent Oriented Methodologies Workshop (OOPSLA 2002), Seattle, USA, November (2002)
- [MOUR '07] Mouratidis, H. and P. Giorgini: Secure Tropos: A Security-Oriented Extension of the Tropos Methodology, International Journal of Software Engineering and Knowledge Engineering, vol 17 (2) pp. 285-309 (2007)
- [MYLO '90] Mylopoulos, J., A. Borgida, M. Jarke and M. Koubarakis: Telos: Representing Knowledge about Information Systems, Information Systems, vol 8 (4) pp. 325-362 (1990)

- [MYLO '92] Mylopoulos, J., L. Chung and B. Nixon: Representing and Using Non-functional Requirements: A Process-Oriented Approach, IEEE Transactions on Software Engineering, vol 18 (6) pp. 483-497 (1992)
- [MYLO '99] Mylopoulos, J., L. Chung and E. Yu: From Object-Oriented to Goal-Oriented Requirements Analysis, Communications of the ACM, vol 42 (1) pp. 31-37 (1999)
- [NAJE '11] Najera, K., A. Perini, A. Martínez and H. Estrada: Supporting *i** Model Integration through an Ontology-based Approach. CEUR Proceedings of the 5th International *i** Workshop (iStar 2011), vol 766 pp. 43-48. Trento, Italy, Aug 29-30 (2011)
- [OXFO] Oxford University Press: Oxford Dictionaries Retrieved August 2011, from <http://oxforddictionaries.com/>
- [PECK '98] Peck, R., L. D. Haugh and A. Goodman: Statistical Case Studies: A Collaboration Between Academia and Industry, SIAM. (1998)
- [ROGE '95] Rogers, E. M.: Diffusion of Innovations. New York, Free Press (1995)
- [ROLL '06] Rolland, C.: From Conceptual Modeling to Requirements Engineering, Lecture Notes in Computer Science (ER'2006), vol 4215 pp. 5-11 (2006)
- [ROSS '77] Ross, D. T. and K. E. Schoman: Structured Analysis for Requirements Definition, IEEE Transaction on Software Engineering, vol 3 (1) pp. 6-15 (1977)
- [ROY '07] Roy, J.-F.: Requirement Engineering with URN: Integrating Goals and Scenarios. Thesis, University of Ottawa (2007)
- [SANN '02] Sannicolo, F., F. Giunchiglia and A. Perini: The Tropos Modeling Language. A User Guide Report # 0204-13, FBK-irst obtained from <http://sra.itec.it/tr/SPG02.pdf> (2002)
- [SANT '02] Santander, V. and J. Castro: Deriving Use Cases from Organizational Modeling. Proc. of the IEEE Joint Int. Conf. on Requirements Engineering (RE'02), pp. 32-39. Essen, Germany, Sept 9-13 (2002)
- [SEID '02] Seidewitz, E.: What Models Means, IEEE Software, vol 20 (5) pp. 26-32 (2002)
- [SETH '96] Sethi, R.: Programming Languages: Concepts and Constructs (2nd edition). Addison Wesley (1996)
- [SHAW '90] Shaw, M.: Prospects for an Engineering Discipline of Software, IEEE Software, vol 7 (6) pp. 15-24 (1990)
- [SIEN '08] Siena, A.: Engineering Law-compliant Requirements. The Nòmos Framework. PhD. Thesis, International Doctorate School in Information and Communication Technologies, University of Trento (2008)
- [SIEN '09] Siena, A., A. Perini, A. Susi and J. Mylopoulos: A Meta-Model for Modelling Law-Compliant Requirements. Second International Workshop on Requirements Engineering and Law (RELAW'09), pp. 45-51, Sept 1 (2009)
- [SMIT '93] Smith, G. F. and G. J. Browne: Conceptual Foundations of Design Problem Solving, IEEE Transactions on Systems, Man and Cybernetics, vol 23 (5), pp. 1209 - 1219 (1993)
- [SOMM '04] Sommerville, I.: Software Engineering, Addison Wesley (2004)
- [SOMM '05] Sommerville, I.: Integrated Requirements Engineering: A Tutorial, IEEE Software, vol 22 (1) pp. 16-23 (2005)
- [STUD '98] Studer, R., R. Benjamins and D. Fensel.: Knowledge Engineering: Principles and Methods. Data and Knowledge Engineering, vol 25 pp. 161-197 (1998)
- [SUDE '05] Sudeikat, J., L. Braubach, A. Pokahr and W. Lamersdorf: Evaluation of Agent-Oriented Software Methodologies – Examination of the Gap Between Modeling and Platform, AOSE 2004. Lecture Notes in Computer Science, vol 3382 pp. 126-141 (2005)
- [SUSI '05] Susi, A., A. Perinni, J. Mylopoulos and P. Giorgini: The Tropos Metamodel and its Use, Informatica, vol 29 pp. 401-408 (2005)
- [TOLK '03] Tolk, A. and J. A. Muguira: The Levels of Conceptual Interoperability Model. Fall Simulation Interoperability Workshop. Orlando, Florida, September (2003)

- [TSAI '05] Tsai, W. T.: Service-oriented System Engineering: a New Paradigm. IEEE Int. Workshop on Service-Oriented System Engineering (SOSE'05), pp. 3-6, Beijing, China, Oct 20-21 (2005)
- [WACH '07] Wachsmuth, G.: Metamodel Adaptation and Model Co-adaptation, Lecture Notes in Computer Science, vol 4609 pp. 600-624 (2007)
- [WASS '96] Wasserman, A. I.: Toward a Discipline of Software Engineering, IEEE Software, vol 13 (6) pp. 23-31 (1996)
- [WEBS '05] Webster, I., J. Amaral and L. M. Cysneiros: A Survey of Good Practices and Misuses for Modelling with *i** Framework, Proc. of the 8th Workshop on Requirements Engineering. Porto, Portugal, pp. 148-160 (2005)
- [WIER '05] Wieringa, R.: Requirements Researchers: Are We Really Doing Research? Requirements Engineering, vol 10 (4) pp. 304-306 (2005)
- [WIKP] Wikipedia Interoperability Retrieved Jun, 2011, from <http://en.wikipedia.org/wiki/Interoperability>
- [WOOL '01] Wooldridge, M. and P. Cincarani: Agent-Oriented Software Engineering: The State of the Art, First International Workshop on Agent-Oriented Software Engineering, (AOSE 2000), Limerick, Ireland, June 10, 2000. LNCS, vol 1957 pp. 1-28 (2001)
- [WYNE '97] Wynekoop, J. L. and N. L. Russo: Studying System Development Methodologies: an Examination of Research Methods, Information Systems Journal, vol 7 (1) pp. 47-65 (1997)
- [YU '93] Yu, E.: Modeling Organizations for Information Systems Requirements Engineering, Proc. of IEEE International Symposium on Requirements Engineering. San Diego, CA, USA, pp. 34-41 (1993)
- [YU '94] Yu, E. and J. Mylopoulos: Understanding “why” in Software Process Modelling, Analysis, and Design. Proc. of the 16th Int. Conf. on Software Engineering. Sorrento, Italy, pp. 159-168 (1994)
- [YU '95] Yu, E.: Modelling Strategic Relationships for Process Reengineering. PhD. Thesis, Computer Science Dept., University of Toronto (1995)
- [YU '97] Yu, E.: Why Agent-Oriented Requirements Engineering. Proc. of the Third Int. Workshop on Requirements Engineering: Foundations for Software Quality, Barcelona, Spain (1997)
- [YU '02] Yu, E.: Agent-Oriented Modelling: Software versus the World, Lecture Notes in Computer Science (AOSE'01), vol 2222 pp. 206-225 (2002)
- [YU '09] Yu, E.: Social Modelling and *i**, Lecture Notes in Computer Science, vol 5600 pp. 99-121 (2009)
- [YU '11] Yu, E., P. Giorgini, N. Maiden and J. Mylopoulos: Social Modeling for Requirements Engineering. Cambridge, MA, USA, MIT Press (2011)
- [ZANN '06] Zannier, C., G. Melnik and F. Maurer: On the Success of Empirical Studies. In Proc. of the 28th International Conference on Software Engineering (ICSE'06), pp. 341-350. Shanghai, China, (2006)
- [ZAVE '97] Zave, P.: Classification of Research Efforts in Requirements Engineering, ACM Computing Surveys, vol 29 (4) pp. 315-321 (1997)

APPENDIX A. Publications

Publications

A.1 Studying and Understanding the *i** Framework

Publication	Used in/for	Venue
1. Ayala, C., C. Cares, J. Carvallo, G. Grau, M. Haya, et al. Análisis Comparativo de Lenguajes de Modelado Orientados a Objetivos Basados en <i>i*</i> . in Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC'04). 2004. Madrid, España.	This study influences Section 2.3 analysis and Section 4.4's metamodel	Regional (Latino-American) conference
2. Ayala, P., C. Cares, J. Carvallo, G. G., M. Haya, et al. <i>A Comparative Analysis of i*-Based Goal-Oriented Modeling Languages</i> . in <i>17th International Conference on Software Engineering and Knowledge Engineering, SEKE'05</i> . 2005. Taipei, Taiwan.	This study influences Section 2.3 analysis and Section 4.4's metamodel	International conference
3. Cares, C., X. Franch, L. López, and J. Marco. <i>Definition and uses of the i* metamodel</i> . in <i>Proc. of the 4th Int. Workshop on i* (iStar10)</i> . <i>CEUR Workshop Proceedings</i> vol.586, pp.20-25. 2010. Hammamet, Tunisia.	Presentation of the definitive <i>i*</i> reference metamodel showed in sections 2.3 and 4.4	International workshop
4. Cares, C., X. Franch, E. Mayol, and C. Quer, <i>A Reference Model for i*</i> , in <i>Social Modeling for Requirements Engineering</i> , E. Yu, et al., Editors. 2011, MIT Press: Cambridge, MA, USA, London, England. p. 573-606.	Foundations of the <i>i*</i> reference metamodel. It also includes de <i>i*</i> genealogy (Section 2.1), and recognizing <i>i*</i> variants (Section 2.2.1)	<u>Book chapter (by invitation and later review)</u>
5. Franch, X., A. Maté, J.C. Trujillo, and C. Cares. <i>On the joint use of i* with other Modelling Frameworks: a Vision Paper</i> . in <i>Proc. of the 19th IEEE Int. Requirements Engineering Conference (RE11)</i> . 2011. Trento, Italy.	This collaboration included a study of <i>i*</i> extensions (Section 2.2.3)	<u>International conference (CORE A)</u>

A.2 Trying *i** on Software Engineering Contexts

These studies have not been included in the thesis document, however they have heavily contributed to understand *i** power and limitations.

Publication	Used in/for	Venue
6. Grau, G., X. Franch, E. Mayol, C. Ayala, C. Cares, et al. <i>RiSD: A Methodology for Building <i>i*</i> Strategic Dependency Models</i> . in <i>The 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'05)</i> . 2005. Taipei, Taiwan.	Participation on a prescriptive methodology for building <i>i*</i> models	International conference
7. Cares, C., X. Franch, and E. Mayol. <i>Extending Tropos for a Prolog Implementation: A Case Study Using the Food Collecting Agent Problem (First Place Award in the CLIMA Contest)</i> . in <i>Sixth International Workshop Computational Logic in Multi-Agent Systems</i> . 2005. London, UK.	Experimenting a logic-based representation for <i>i*</i> models in agent-oriented contexts	International workshop
8. Cares, C., X. Franch, and E. Mayol, <i>Using Antimodels to Define Agents's Strategy</i> . <i>Lecture Notes in Computer Science, Series, vol 4371, 284-293</i> . Lecture Notes in Artificial Intelligence (Second Place Award in the CLIMA Contest), 2006. 4371 : p. 284-293.	Experimenting a logic-based representation for <i>i*</i> models in agent-oriented contexts	International workshop
9. Cares, C., X. Franch, E. Mayol, and E. Alvarez. <i>Goal-Driven Agent-Oriented Software Processes</i> . in <i>32nd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO'06)</i> . 2006. Cavtat/Dubrovnik, Croatia.	Using <i>i*</i> for software process modelling and improving	International conference
10. Franch, X., G. Grau, E. Mayol, C. Quer, C. Ayala, et al., <i>Systematic Construction of <i>i*</i> Dependency Models for Socio-Technical Systems</i> . <i>International Journal of Software Engineering and Knowledge Engineering</i> , 2007. 17 (1): p. 79-106.	Participation on a prescriptive methodology for building <i>i*</i> models	<u>SCI-indexed journal with IF</u>
11. Cares, C. and X. Franch. <i>Selecting mobile office devices using a goal-oriented approach</i> . in <i>Proc. of the 3rd Int. Conf. on Research Challenges in Information Science (RCIS09)</i> . 2009. Fez, Morocco: IEEE Computer Society.	Extending the <i>i*</i> framework for a hardware selection proposal.	International conference
12. Cares, C. and X. Franch, <i>3MSF: A Framework to Select Mobile Office Devices</i> . <i>International Journal of Computer Sciences and Application</i> , 2009. 6 (5): p. 121-144.	Extending the <i>i*</i> framework for a hardware selection proposal.	International journal (invited as one of best papers in RCIS'09)
13. Hernández, P., A. Castro, J.-N. Mazón, J. Trujillo, and C. Cares. <i>Modelling requirements with <i>i*</i> in the development of a data warehouse for an university</i> . in <i>The iStar Showcase: Exploring the Goals of your Systems and Businesses at CAiSE'11</i> . 2011. London, UK.	Using <i>i*</i> in an industrial case	International workshop

A.3 Theoretical Foundations for an Interoperability Proposal

Publication	Used in/for	Venue
14. Cares, C. <i>A Semiosphere-based Approach for Modelling Pre-semantic Engineering Communication</i> . in <i>Workshop on Philosophy & Engineering (WPE'08)</i> . 2008. London, UK: Royal Academy of Engineering.	It sustains feasibility of polysemantic scenarios on engineering modelling which is included in section 3.2.3	International workshop
15. Cares, C., X. Franch, and E. Mayol. <i>Perspectives about paradigms in Software Engineering</i> . in <i>Second Workshop on Philosophical Foundations of Information Systems Engineering (PHISE'06)</i> . 2006. Luxembourg.	Kuhn's paradigmatic principles, under an engineering perspective, are used in section 3.2.1	International workshop
16. Cares, C. and X. Franch. <i>IStarML: Principles and Implications</i> . in <i>Proc. of the 5th International iStar Workshop</i> . 2011. Trento, Italy.	This paper summarizes section 3.2 and 3.3.	International workshop

A.4 Proposing iStarML

Publication	Used in/for	Venue
17. Cares, C., X. Franch, A. Perini, and A. Susi, <i>iStarML Reference Guide</i> . LSI Technical Report. ID: LSI-07-46-R, http://www.lsi.upc.edu/~techreps/files/R07-46.zip , 2007.	The iStarML specification, mainly presented on sections 4.3 and 4.4	Technical report
18. Cares, C., X. Franch, A. Perini, and A. Susi. <i>iStarML: an XML-based Interchange Format for i* Models</i> . in <i>Proc. of the 3rd Int. i* Workshop (iStar08)</i> . <i>CEUR Workshop Proceedings</i> . 2008. Recife, Brasil.	Presentation of basic metamodel foundations of iStarML. Used in section 4.3.	International workshop
19. Cares, C., X. Franch, A. Perini, and A. Susi, <i>Towards interoperability of i* models using iStarML</i> . <i>Computer Standards and Interfaces</i> , 2011. 33 (1): p. 69-79.	A revised presentation of iStarML principles (sections 4.3 and 4.4), and the i* community survey (section 5.3)	<u>SCI-indexed journal with IF</u>
20. Cares, C. and X. Franch, <i>A Metamodelling Approach for i* Model Translations</i> . <i>Lecture Notes in Computer Science (CAiSE'11)</i> , 2011. 6741 : p. 337-351.	It includes transformations principles included in section 3.3, the transformation proposal in sections 4.5 4.6 and the transformation cases in section 5.1.4	<u>International conference (CORE A)</u>

A.5 iStarML Support and Usages

Publication	Used in/for	Venue
21. Cares, C. <i>CCISarML (v0.6), a Java Package for Handling iStarML files.</i> 2007; Available from: http://www.essi.upc.edu/~gessi/iStarML/ .	The services of iStarML Java package has been summarized in section 5.2.3	Downloadable software
22. Cares, C. and X. Franch. <i>Towards a Framework for Improving Goal-Oriented Requirement Models Quality.</i> in <i>12th Workshop on Requirements Engineering (WER'09)</i> . 2009. Valparaiso, Chile.	It details the quality proposal for <i>i*</i> models summarized in section 5.2.4	Regional workshop
23. Colomer, D., L. Lopez, C. Cares, and X. Franch. <i>Model Interchange and Tool Interoperability in the i* Framework: A Case Study.</i> in <i>14th Workshop on Requirements Engineering (WER'11)</i> . 2011. Rio de Janeiro, Brasil.	It shows the <i>i*</i> model transformations of section 5.1.5	Regional workshop
24. Cares, C., X. Franch, D. Colomer, and L. López. <i>Tool Interoperability using iStarML.</i> in <i>Proc. of the 5th International iStar Workshop. Tool demo fair.</i> 2011. Trento, Italy.	Summary of iStarML technological applications presented in Chapter 5	International workshop

A.6 *i** Interoperability Related Work

Publication	Used in/for	Venue
25. Giachetti, G., X. Franch, B. Marin, et al. <i>Using Measures to Improve i* Models for Automatic Interoperability in Model-Driven Development Processes.</i> in <i>XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2011)</i> . 2011. La Coruña, España.	Participation which includes interoperability issues on <i>i*</i> models	National conference