iStar'08 – 3rd International *i** Workshop

11-12th February, 2008

Recife, Pernambuco, Brazil

http://www.cin.ufpe.br/~istar08/site/

Jaelson Castro
Xavier Franch
Anna Perini
Eric Yu (eds.)

a CEUR Workshop Proceedings, ISSN 1613-0073

© 2008 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. Re-publication of material from this volume requires permission by the copyright owners.

Preface

In the enterprise of research, one of the most rewarding aspects is the sense of community arising from the interactions among researchers and research teams, who inspire and challenge each other to greater heights. The community of researchers exploring intentional strategic actor relationships modeling, as initiated by the i* framework, continues to grow and flourish worldwide.

In these proceedings you will find short papers summarizing the work to be presented at the Third International i* workshop. This series of workshops is dedicated to the discussion of concepts, methods, techniques, tools, and applications associated with i* and related frameworks and approaches. The objective of the workshop is to provide an opportunity for exchanging ideas, comparing notes, and forging new collaborations.

As with previous workshops in the series (2001 in Trento, Italy, and 2005 in London, UK), we aim to maintain the format of a small and informal gathering to maximize interaction. In this edition, however, we have decided to produce a proceedings from the workshop, with participants submitting short papers in advance of the workshop. The proceedings, published under CEUR, with an ISSN number, will serve as a tangible record of the event.

We look forward to lively conversations and debates at the workshop, in the beautiful surroundings in Recife, Brazil!

Jaelson Castro, Universidade Federal de Pernambuco, Brazil
Xavier Franch, Universitat Politècnica de Catalunya, Spain
Anna Perini, Fondazione Bruno Kessler – IRST, Italy
Eric Yu, University of Toronto, Canada

Index

Towards Aspectual i*

Fernanda Alencar[1] , Jaelson Castro[1], Cleviton Monteiro[1],
Ricardo Ramos[1], and Emanuel Santos[1]

[1] Universidade Federal de Pernambuco, Recife, Brasil
{fernanandaalenc, cleviton}@gmail.com, {jbc, rar2, ebs}@cin.ufpe.br

Abstract. The i* framework is a very popular approach in the Requirements Engineering community. However, crosscutting concerns are not handled explicitly in i*, compromising the modularity and consequently the complexity and evolution of such models. To deal with the complexity of i* models, the use of structuring mechanisms has been investigated. Moreover, the i* framework has been extended to incorporate the principles of aspect orientation, aiming at handling explicitly crosscutting concerns in i* models. Some metrics have been created to evaluate the understandability of requirements modeled using both the i* framework and the aspectual i*.

Keywords: requirements, modularity, graphical complexity, evolution, understandability, metrics, aspects.

1 Introduction

During the early stages of requirements engineering, it is necessary to identify and specify the stakeholders' needs. The i* framework provides expressive models to achieve this, wherein motivations and rationale are explicitly captured in a requirements model. Thus, the i* framework is becoming widely used by the Requirements Engineering community, capturing social and intentional characteristics of the system organisation context.

However, i* models, even for small applications, may become cluttered, compromising their evolution, scalability, and, consequently, its understandability. In large and complex projects, this problem increases significantly. Furthermore, traditional i* models tend to include scattered and tangled representations that are hard to understand and maintain. Therefore, as systems models evolve and grow in scale and complexity, better encapsulation and localization primitives are needed. Thus, we proposed to use some of the Aspect Oriented Software Development concepts to improve the i* models views.

The paper is organised as follows. Section 2 describes the objectives of the research. Section 3 presents the proposed tool to automate the identification of the candidate aspects on i* models. Finally, Section 4 concludes the work and points to future research.

2 Improving the understandability of the i* models

Since 1997 we have been investigating the use of i*, aiming at identifying its strengths and weaknesses. In the last ten years, this research group has proposed several approaches either to improve i* or to integrate it with other relevant techniques. For example, approaches have been proposed to reduce the gap between organizational and functional requirements by integrating i* and UML models [1,2]. To deal with the complexity of i* models, the use of structuring mechanisms has been investigated. Moreover, the i* framework has been extended to incorporate the principles of aspect orientation, aiming at handling explicitly crosscutting concerns in i* models.

We proposed an approach for identifying and separating crosscutting concerns (henceforth referred to as candidate aspects) in i* models [2, 3]. In particular, we provide means to discover and model tangled and scattered tasks, goals and softgoals as well as internal links (means-ends and contributions links). Composition is handled in a graphical way. Hence, we introduced (i) a set of guidelines to identify the candidate aspects in i* models; (ii) an extension of the i* metamodel by adding aspectual constructors to modularize candidate aspects and its composition with other system modules; and (iii) the concrete syntax of the aspectual i* modelling language (Fig. 1).

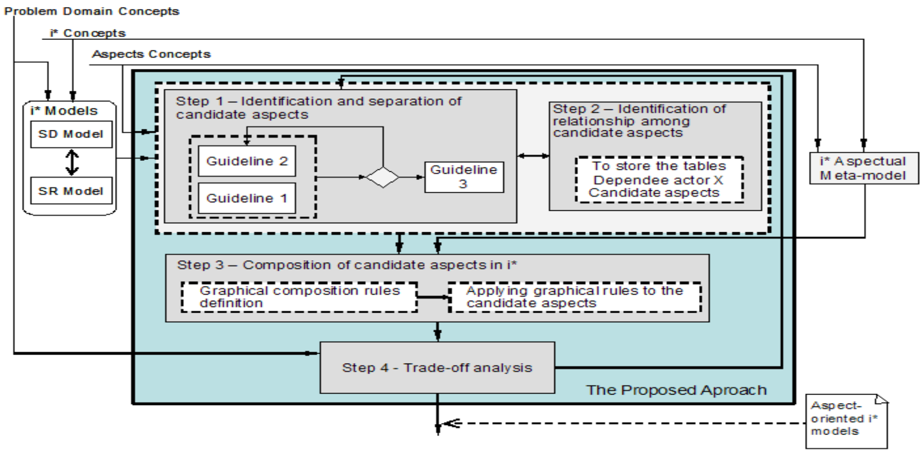

Fig. 1. The overview of the approach

We also evaluate the requirements document quality, finding pieces that can be improved with the application of requirements patterns and refactorings [5,6,7,8]. The refactoring opportunities that we have identified have the goal of making the requirements more understandable as well as to improve the overall organization of the project. But depending on the system quality attributes considered (for example: reusability, maintainability, etc) the requirements engineer might take actions that slightly differ from the guidelines proposed here. Also, the expected granularity of the requirements descriptions might influence the use of the refactoring opportunity. The requirements engineer should take these issues into consideration.

3 The iAspectPlugin

It is quite hard to manually identify crosscutting concerns directly from the organizational i* models. Thus we propose some means to automate this process. Therefore, the plug-in was developed with the objective of turning the identification faster and easier. The current version of iAspectPlugin [4] does not automates the whole process described in [3] yet; it only supports the activities of identifying and representing the crosscutting concerns.

When we developed the iAspectPlugin we chose to reuse the OME environment instead of creating a new tool exclusively to treat the crosscutting concerns identification, because this is a stable tool, largely used and allows a simple way to be extended.

Regarding the iAspectPlugin graphical interface, Fig. 2 shows a screen captured from the OME menu with the developed plug-in already installed. The circle is highlighting the button that was added to run the crosscutting concerns identification automatically. At all five buttons were added by the plug-in and are numbered in Fig. 2. The buttons 1, 2, 3 and 4 can be used to construct manually i* models with aspects. The user can do the organizational modeling with any i* element (Actors, intentional elements, links, etc) and at any time press the button 5 to run the crosscutting concerns identification. Then the plug-in performs the aspect identification and updates the model creating the crosscutting concerns stereotypes, composition rules inside the aspects and crosscuts links. It also moves the elements that were captured by the guidelines to the crosscutting concern internal representation (expansion).

A plug-in for the OME tool is a Java class that implements the OMEPlugin and interacts with other classes defined by the plug-in developers. The Java classes design's definition started using the GRASP (General Responsibility Assignment Software Patterns) guide for the requirement analysis.

Fig. 2. OME menu with the buttons added by the plug-in

4 Conclusions and future works

In order to improve the evolution, reuse and maintainability of i* models we have considered the modularization and localization of concerns. In particular, guidelines are required for the identification of crosscutting concerns. Moreover, we also need a notation for the description of improved models as well as compositions rules to keep a record of the relationships between the crosscutting concerns and the other model elements. The approach aims at reducing the graphical complexity of large i* models.

We also extended the i* metamodel to introduce two main new concepts: aspectual actor and crosscut relationship. Aspectual actors modularize candidate aspects and the crosscut relationship captures the information of source and target model elements, as well as, when and how an aspect crosscuts other model elements.

Currently, we are working on the inclusion of routines identification and their association to concerns which may be tangled and scattered across various actors in i* models. Another point to be investigated is how to deal volatile concerns that might not be necessarily scattered or tangled. Also some metrics have been proposed to evaluate the understandability of requirements modeled using both the i* framework and the aspectual i. Finally, we plan to redefine the current tool [4] to support the new guidelines and the entire process for identification and separation of candidate aspects.

References

1. Castro, J., Alencar, F., Cysneiro Filho, G.., Mylopoulos, J. Integrating Organizational Requirements and Object Oriented Modeling In: 5th IEEE International Symposium on Requirements Engineering, 2001, Toronto. Proceeding of 5th IEEE International Symposium on Requirements Engineering. IEEE Press, 2001. p.146 – 153.
2. Alencar, F., Cysneiros Filho, G.., Castro, J..: Support for Structuring Mechanism in the Integration of Organizational Requirements and Object Oriented Modeling. In. V Workshop de Engenharia de Requisitos (WER02). Espanha. Valência: Anais/Workshop de Engenharia de Requisitos (2002).
3. Alencar, F., Castro, J., Moreira, A., Araújo, J., Monteiro, C., Ramos, R., Mylopoulos, J.: Simplifying i* Models. Agent-Oriented Information Systems Workshop Proceedings (AOIS'07), Intl. Conf. on CAiSE'07. (eds.) Tapir Academic Press, pp. 635-649, Norway (2007).
4. Monteiro, C., Ramos, R. A., Castro, J. B., Araújo, J. Moreira, A., Alencar, F.: The iAspectPlugin to Automatize the Identification of Crosscutting Concerns on i* Models. In: I Latin-American Workshop on Aspect-Oriented Software Development 2007, João Pessoa. I Latin-American Workshop on Aspect-Oriented Software Development. João Pessoa : pp. 105-116, Sociedade Brasileira de Computação, (2007).
5. Ramos, R. A. ; Alencar, F. ; Araújo, J. ; Moreira, A. ; Castro, J. and Penteado, R.: i* with Aspects: Evaluating Understandability. In: Workshop on Requirements Engineering, 2007, Toronto, Canada (2007b).
6. Ramos, R.A., Araújo, J., Castro, J., Moreira, A., Alencar, F., Silva, C.: "Uma abordagem de instanciação de métricas para medir documentos de requisitos orientados a aspectos", in: 3º Brazilian Workshop on Aspect Oriented Software Development - WASP2006. Florianopolis, Brasil (2006)
7. Ramos, R. A.; Araújo, J. ; Moreira, A. ; Castro, J. ; Alencar, F. and Penteado, R.: "Um Padrão para Requisitos Duplicados", in: 6th Latin American Conference on Pattern Languages of Programming (SugarLoafPlop'2007), Porto de Galinhas, Recife, Pernambuco, Brasil (2007a)
8. Ramos, R., Piveta, E., Castro, J., Araújo, J., Moreira, A., Guerreiro, P., Pimenta, M., and Tom Price, R.. Improving the Quality of Requirements with Refactoring. In: VI Simpósio Brasileiro de Qualidade de Software – SBQS2007, Porto de Galinhas, Recife, Pernambuco, Brasil, Junho 27 – 30 (2007c)

Analyzing knowledge transfer in software maintenance organizations using an agent- and goal-oriented analysis technique – an experience report

Maria Carmela Annosi[1], Antonino De Pascale[1], Daniel Gross[2], Eric Yu[2]

[1] Ericsson Marconi Spa- via Madonna di Fatima 2,
84016 Pagani (SA)- Italy.
{MariaCarmela.Annosi, Antonino.DePascale}@ericsson.com
[2] Faculty of Information Studies, University of Toronto
{daniel.gross, eric.yu}@utoronto.ca

Abstract: Software maintenance is a knowledge intensive activity. Implementing effective knowledge management policies and structures during maintenance is therefore a key factor to improving software quality and maintenance productivity. This paper presents a novel agent and goal-oriented analysis technique that was used at Ericsson Marconi Spa to analyze the successful adoption of organizational structures and policies which greatly facilitated knowledge transfer. Utilizing this analysis technique helped to systematically represent, capture and analyze the strategic organizational relationships relevant to knowledge transfer. By capturing and analyzing such strategic relationships, it was possible to make visible the reasons why newly adopted policies and structures improved knowledge transfer, and thus helped the maintenance team to achieve significant improvements in their maintenance processes, and successfully achieve their software maintenance goals.

Keywords: Knowledge transfer, agent- and goal-orientation, maintenance

1 Introduction

In 2003 Ericsson Marconi Spa developed a set of organizational policies and structures geared towards improving the management and transfer of knowledge in its development organization. Since then the policies and structures were implemented with outstanding results during corrective maintenance activities.

To better understand why organizational maintenance goals were better achieved, an agent- and goal-oriented knowledge representation and analysis technique was applied. This paper reports on the insights gained, the benefits observed, and the lessons learned.

2 Objectives of the research

The objective of this research was to gain insights into the successful adoption of organizational polices and structures during corrective maintenance activities. Based on the assumption that the organizational structure and processes were key enablers of success, the i* modeling framework [1] was chosen to analyze how these structures and processes contributed to the successful achievement of the organizations maintenance goals. Another objective was to identify the utility of the i* modeling framework as a modeling and analysis technique, and, in particular, its use to document, communicate and explain among stakeholders the positive effects of the instituted organizational policies and structures.

3 Scientific Contributions

This paper includes two main contributions: a) the presentation of several organizational policies and structures that contributed at Ericsson Marconi Spa to significantly improved knowledge transfer and acquisition during software maintenance tasks, which in turn contributed to tasks' success, and b) the illustration of i*, as a modeling and analysis technique, that supports representing, capturing and analyzing these policies and structures, and explain how these contributed to success.

A key purpose of organizational policies and structures introduces was to create a knowledge environment which facilitates knowledge sharing and creation amongst individuals, and fosters the development of tacit knowledge.

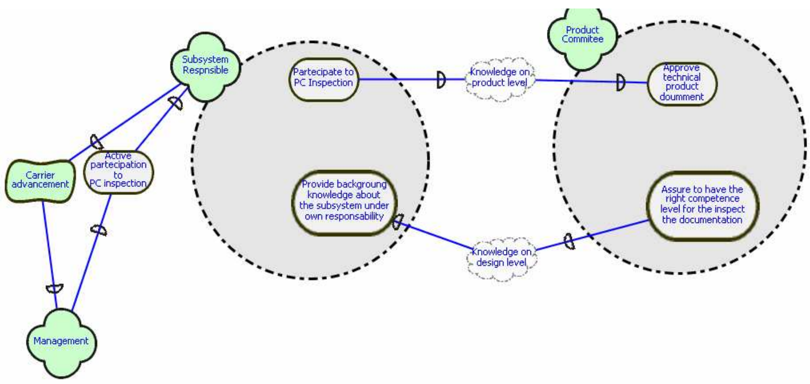

Figure 1: Knowledge transfer from Product Committee members to maintenance core team members responsible for a subsystem and vice versa

At Ericsson Marconi Spa several organizational decisions were made to facilitating the correlation and integration of the experience of individuals each working in different parts of the development organization.

First, four core members of the maintenance team (made up of nine people) where asked to take on responsibilities that went beyond maintaining selected subsystems, and included membership in the product development team and the product committee, the latter responsible for product and process quality. The effect of these multiple role assignments was that these members had to mingle and interact with individuals from other parts of the organization, not only learning first hand the particular challenges and issues others had to deal with, but also be part of the team that proposed solutions.

Maintainers therefore effectively gained hands-on experience with problems raised by customers in the field, participated in developing new product requirements, participated in proposing solutions to problems and faults, which then resulted in subsequent maintenance activities.

It is worthwhile to note that since same persons were assigned multiple responsibilities, the synergy between the diverse activities they performed increased the quality of the experiences they gained. On the other hand, also others in the development organization benefited from these "multitasking" members.

Members of the Product Committee took advantage of having colleagues with direct experience from the field, from product requirements specification, design and development, as well as maintenance activities. This raised the level of competence and assurance within the product committee product committee during inspections, and facilitated the knowledge transfer from various functional areas of the organization to product committee members.

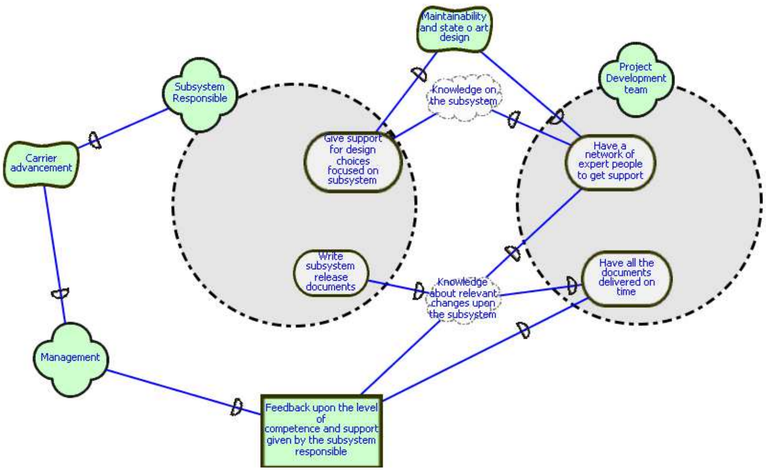

Fig. 2: Knowledge Transfer at design level from the Project Development team towards the Subsystem Responsible.

To encourage the participation in product committee meetings, an additional organizational policy was introduced requiring core members to attend a minimum number of meetings.

Another organizational policy required core members to update subsystem release documents. Writing and updating these documents was another way to acquaint maintainers with the product evolution from a product design perspective.

Another important synergist effect was that core team members often made suggestions during the design stages, as to how maintainability of subsystems under their responsibility could be improved. Suggestions often involved state-of-art design principles and approaches, which in turn improved the quality and maintainability of the products developed.

Figure 1 illustrates how the approval of technical documents by the product committee was made, as per organizational policy, dependent on the participation of an individual responsible for the maintenance of a subsystem, and in particular the latter's contribution of product knowledge. Figure 2 illustrates how management motivates participation. Management is not only in charge of advancement management also obtains feedback from the development team as to the degree of participation.

Given these motivations maintainers share knowledge with project team members, as well as write subsystem release documentation for them. Figure 2 further illustrates that maintainers responsible for subsystems depend on project development team to implement maintainability measures, which in turn increases the maintainer's productivity. This strategic relationship provides additional motivation for maintainers to share their knowledge so that development teams are motivated to include maintainability suggestions into the product.

4 Conclusions

Modeling organizational policies and structures that encouraged knowledge transfer has proven not only practical but of great value. The models obtained effectively captured how new organizational policies and structures enabled and successfully facilitated knowledge transfer amongst organizational stakeholders.

One limitation observed was the issue of readability and understandability of produced diagrams, which, when diagrams size increased, became more difficult to achieve. This is very important since lack of readability and understandability runs the risk of loosing the knowledge captured in diagrams. In this project, many simple diagrams were produced, rather than cluttering a smaller number of diagrams with too much detail.

The expressiveness of i* helped illustrate that the maintenance organization was well designed and managed, and helped mobilize tacit knowledge held by individuals by an effective combination of instituted organizational roles and positions.

5 Ongoing and future work

Given the successful use of i* as a graphical modeling technique in helping stakeholders understand why policies and structures introduced helped create a successful maintenance organization, future work will focus on disseminating this knowledge to other Ericsson Marconi Spa organizations. Future work will also focus on easing the learning curve of the i* modeling technique so that knowledge transfer mechanisms can become more readily understandable.

References

[1] E. Yu.:"Modeling strategic relationships for process reengineering", PhD dissertation, Department Of Computer Science, University of Toronto.

Analyzing software process alignment with organizational business strategies using an agent- and goal-oriented analysis technique - an experience report

Maria Carmela Annosi[1], Antonino De Pascale[1] Daniel Gross[2], Eric Yu[2],

[1] Ericsson Marconi Spa- via Madonna di Fatima 2,
84016 Pagani Italy.
{MariaCarmela.Annosi, Antonino.DePascale}@ericsson.com
[2] Faculty of Information Studies, University of Toronto
{daniel.gross, eric.yu}@utoronto.ca

Abstract. The continuous alignment of organizational processes with an organizations' business strategy is a key factor to its success. This paper reports on the application of the i* framework to support the continuous alignment of corrective software maintenance processes with the strategic goals of a Software Design Maintenance Organization at Ericsson Marconi Spa. Using the i* modeling framework, alignment of process elements with related business goals was successfully represented, captured and analyzed, which in turn facilitated process improvement efforts. To better deal with large i* models, a model slicing technique based on goals was introduced, which made models more readable, and contributed substantially to how the maintenance processes were analyzed and improved.

Keywords: Organization model, software maintenance organization, business process reengineering.

1 Introduction

Customer satisfaction is a key success criteria and driving force not only at Ericsson Marconi Spa, but for Ericsson worldwide also. Internal design centers world wide are as a matter of routine compared with each other on merit of customer satisfaction, effectively creating much internal competition.

While customer satisfaction is an important driving force for all departments in a local design center, it is the product maintenance department, and how it manages product maintenance requests, that are often most visible to the customer. Maintenance therefore most directly affects customer's satisfaction, and strategies employed by design centers on how to effectively design maintenance processes are key contributors to attaining the organizations most important business goals.

According to Michael Porter [2], strategy involves two main components: unique strategy and operational effectiveness, both of which are essential to the formulation of a successful business strategy.

Unique strategy concerns with what a company does differently from its rival. This includes differences in how they design their processes, use resources, etc, all of which provides a competitive advantage. Operational effectiveness is about performing activities and using technology according to known best practices.

2 Objectives of the research

The objective of this research is to provide modeling and analysis facilities that support representing, capturing and analyzing the key business drivers of Ericsson Marconi Spa's maintenance organization, and how these drive strategy formulation, and consequently the manner how maintenance processes are designed. This research adopts the i* modeling framework as the modeling technique for this purpose.

3 Scientific Contributions

This research offers two main contributions. First, it applies the i* modeling framework to represent, capture and analyzing the alignment of business goals and strategies with processes of Ericsson Marconi Spa's maintenance organization. Second, it introduces a model slicing mechanism to better support the modeling and analysis process.

Discussing the second contribution first, the slicing mechanism helps cope with the large amount of information typically found in i* process models. Instead of dealing with one large diagram, several smaller diagrams are produced, each capturing how a specific operational business goal or particular maintenance related purposes, are achieved. Operational goals are derived from business strategies identified earlier, while maintenance related purposes are derived from the maintenance organizations mission.

Operational goals and purposes serve as structuring criteria for the modeling of maintenance processes, effectively establishing different concern or viewpoints over the overall processes.

Furthermore, using goals and purposes as slicing criteria leads to the partitioning of the overall process descriptions into relatively independent viewpoints, with clearly defined intentional dependencies between the different views. In addition, we observer that by slicing along the lines of goals and purposes, no duplicate information is captured in different viewpoint diagrams, thereby eliminating the need for consistency checking across viewpoints, which improves the maintainability of these diagrams (changes in one diagram does not affect other diagrams). Finally, navigating process descriptions through the use of single views results in easier way to understanding the reasons (the "why") certain process elements where introduced (the "what") to the process.

Following additional advantages to having viewpoints can be identified:

a) each viewpoint is dedicated to a particular operational goal or maintenance purpose; this helps to further distinguish between whether process elements contribute to a business goals, or to particular maintenance purpose;
b) each viewpoint is dedicated to supporting a particular customer need; this helps identify whether processes exist that address particular customer needs;
c) with operational and maintenance goals derived from higher level business strategies, a clear link between process, goals and strategies can be identified
d) each viewpoint is independently be aligned to associated goal; misalignments within each viewpoint can thus more easily be identified.

Slicing process models into viewpoints therefore not only helps in overcoming the information overload problem found in large process diagrams, but also offer a better way to understanding, managing, changing and improving processes.

Considering the first contribution, following is a partial description of a corrective software maintenance process, as defined in one Ericsson Marconi Spa Design Maintenance Organization (DM).

Figure 1 captures relevant roles and positions identifiable within the Design Maintenance organization, while figure 2 captures one viewpoint of a maintenance process that specifically deals with the operational goal of keeping the approved correction cancellation rate less than 2.5% percent.

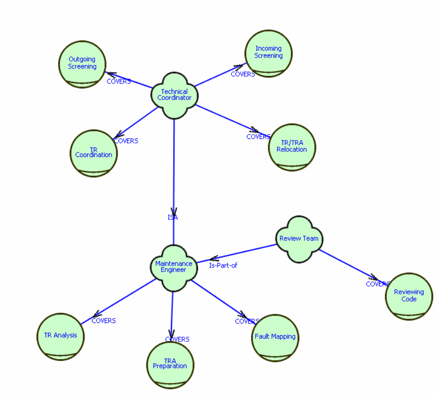

Fig. 1. Roles covered by some of DM positions

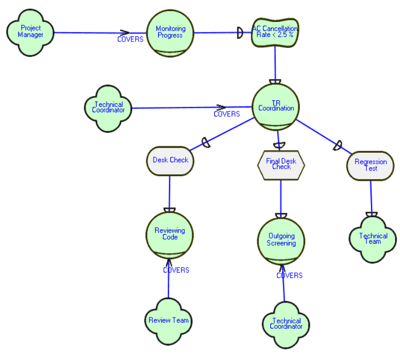

Fig. 2. Strategic dependencies to achieve the business goal to have a reduced AC cancellation rate

A different viewpoint (not shown here) was produced to capture the business process related to the operational goal to obtain a “TR” closure rate < 3 days (TR standing for Trouble Report, a formal notification of a fault).

4 Conclusions

Using the i* framework assisted in understanding how business process relate to operational business goals, which in turn were derived from business strategy. By slicing process diagrams along the lines of operational goals, guided the analysis process and helped improve the understandability of individual diagrams. Utilizing these diagrams help maintenance team members to better understand the maintenance processes, and offered opportunities to make suggestions for process improvements.

The increased understanding of the maintenance processes was particularly important for the engineers, in particular gaining insights into the reasons why certain process elements were needed, since it further motivated then in executing the process in the manner they were specified.

As an overall result this lead to the better management of corrective maintenance, to decreased lead time; and thereby, to increased productivity [4].

5 Ongoing and future work

Future work would focus on extending the framework as more experience is gained in modeling and analyzing processes in light of business strategy and process improvement. Another avenue of interest is aligning the framework with Object-oriented development processes and methodologies already established within Ericsson Marconi Spa.

References

[1] E. Yu.:"Modelling strategic relationships for process reengineering", Ph.D dissertation, Department Of Computer Science, University of Toronto.
[2] M. Porter.:"What is a strategic? Harvard Business Review, November-December 1996.
[3] Ericsson documentation: TR Handling Process – Organizational View ERI/XS-05:0034 Uen Rev. A
[4] Mira Kajko-Mattsson, Stefan Forssander, Ulf Olsson,'Corrective Maintenance Maturity Model (CM3):Maintainer's Education and Training', 13rd International Conference on Software Engineering (ICSE'01)

iStarML: an XML-based Interchange Format for *i** Models

Carlos Cares[1], Xavier Franch[1], Anna Perini[2] and Angelo Susi[2]

[1] Universitat Politècnica de Catalunya, C/Jordi Girona, 1-3, 08034 Barcelona, Spain
{ccares, franch}@lsi.upc.edu
[2] Fondazione Bruno Kessler (FBK-IRST), Via Sommarive 18, I-38050, Povo, Trento, Italy
{perini,susi}@fbk.eu

Abstract. There are several tools currently available in the *i** community with different purposes. This situation poses both benefits and difficulties. Benefits, because different groups may be able to share their models and results among their tools, and even connect different tools in order to perform complex processes. Difficulties, because most of these tools differ either in the underlying metamodel of the language, or the format in which they store the models, or in both. To overcome the difficulties and exploit the benefits, we have defined the iStarML model interchange format as a practical solution to this problem. In this paper we present the research line which supports this outcome. We present its motivation, objectives and current outcomes, the expected contributions and finally our on going and future work.

1 Introduction

The *i** framework has been recognized like both: a goal-oriented and agent-oriented modelling framework. Several extensions and variants to the original framework have been defined in order to handle different modelling situations. Most recent approaches aim at dealing with the increasing complexities in developing nowadays information systems, which need to operate in open, heterogeneous and evolving environments. This has implied a diversity of *i** applications on a wide research community[3].

In general, existing *i**-based tools and development frameworks are not capable to interoperate, which prevents taking advantage of existing functionalities. One of the main reasons related to the lack of interoperability of different *i** frameworks is that the different *i**-based proposals use, define, or redefine the syntax or even the semantics of the seminal *i** constructs. Among the main *i** variations are: the seminal proposal, the variation presented as the modelling language of Tropos, the Secure-Tropos (S-Tropos) proposal, the Formal-Tropos (F-Tropos) proposal and the URN/GRL proposal. Moreover different tools are supporting these variants which have configured a global scenario where reusing diagramming and/or analyzing tool capabilities are a serious difficulty. In table 1 we present a summary of the *i** tools and their main interoperability features. Table 1 mainly shows that in spite of using similar technologies the specific store formats, these are all different ones.

[3] For a more complete idea on the prolific community working around *i** you may visit the *i** wiki page at http://istar.rwth-aachen.de.

Table 1. Interoperability scenario of *i** tools, extracted from http://istar.rwth-aachen.de

*i** Tool	Institution	*i** variant	Importing file formats	Exporting file formats	Technology
OpenOME	University of Toronto	*i** / GRL	.tel .sml .vdx	.tel .vdx	Eclipse plug-in
OME		*i** / GRL	.tel .sml	.tel	Java
REDEPEN-REACT	City University & Tec. University of Catalonia	*i**	Visio XML	Visio XML	Visio plug-in
TAOM4E	Foundazione Bruno Kessler	Tropos	XMI (Tropos)	XMI (Tropos)	Eclipse plug-in
ST-Tool	University of Trento	S-Tropos / F-Tropos	Datalog files	Own XML F-Tropos, Datalog,	Java/Datalog
J-PRIM	Tec. University of Catalonia	*i**	No	No	Java/MySql
jUCMNav	University of Ottawa	GRL / URN	XMI (URN) own XML	XMI (URN) own XML	Eclipse plug-in
DesCARTES	Catholic Univ. of Louvain	*i** / Tropos	Own XML	Own XML, SQL, JACK	Eclipse plug-in

2 Objectives of the Research

Our work has focused on practical issues related to *i** interoperability. In particular, our main objective is to provide a formal representation where differences and similarities among *i** variants are explicit, generating a common representational framework for the *i** community and, in spite of the differences, enabling effective communication inside the community, tool interoperability and a common representation for repository of *i** models. Therefore we have worked with three specific objectives in mind: (i) To propose an interoperability language using a contemporary and easy-to-adopt technology and, at the same time, based on a core common set of *i** concepts, (ii) To propose an interoperability scenario to help understanding the roles of different *i** -related tools on software and organizational process, and (iii) To get a formal specification of the common and abstract conceptual framework to allow dealing with current and maybe future *i** differences and be helpful to design new and useful *i** variations.

We have already reached the first objective. A preliminary work was about understanding the *i** variations, their differences and similarities [1], then we extended this work including a study of the maturity of the common *i** concepts on most widespread *i** variations and we created an *i** Reference Model including many *i** variation communalities [2]. From here we have extracted a core set of abstract and common concepts which has been the platform for defining an XML-based language for *i** model interoperation named iStarML [3]. In table 2 we show the core concepts and their corresponding iStarML tags. Also we have included some of the main options in order to illustrate how particular *i** constructs can be represented.

Table 2. Core concepts of *i**-based modelling languages and proposed XML tags for iStarML

*i** core concept	iStarML Tag	Main attributes or subtags
Actor	<actor>	*type* attribute to specify different types of actors (e.g. agent)
Intentional element	<ielement>	*type* attribute to specify different kind of intentional elements (e.g. goal)
Dependency	<dependency>	Can contains two subtags: <dependee> and <depender>
Boundary	<boundary>	*type* attribute for representing future variations on boundary conceptualizations
Intentional element link	<ielementLink>	*type* attribute to specify types of intentional relationships (e.g. contribution) *value* attribute to specify values related to the relationship (e.g. +,++,-,++)
Actor association link	<actorLink>	*type* attribute to specify different types of actors' associations (e.g. is_part_of)

3 Scientific Contributions

We believe that proposing an interoperability solution to the *i** community can positively impact on it enabling the synergy among groups and among specific frameworks and tools. Besides, given that this proposal includes a format specification, it enables the transition from research proposals to industrial applications. Moreover we think that the definition and adoption of a reference format based on a limited set of concepts, flexible enough in order to consider the different existing *i** variations and also extensible in order to consider new variations, would help to reach interoperability objectives. In particular, we believe that the definition of iStarML allows: (i) having a file format for diagrams interchanging among existing and new tools; (ii) motivating the development and compliance of drawing tools to the defined format; (iii) developing *i**-based analysis algorithms independently of the graphic issues; (iv) extending existing tools with new *i**-based analysis components; (v) representing specific additional syntactic constraints to specify evolutions or new variations; (vi) having a common way of representing the differences and similarities between existing *i** variations.

However, this interesting set of benefits depends on the acceptability of the proposal which could be engaged offering tools that helps its implementation and mainly, having a flexible perspective for including the mainstream lines inside the *i** research activity. About the first point we have already developed a Java Library for checking, creating, importing and exporting iStarML files. About the second one, we are applying a survey to get information which would help to improve the proposal.

4 Conclusions

In this paper we have reviewed the join activity among the Technical University of Catalonia and IRST in order to propose an interoperability framework for the *i** research community. We have presented the scenario which motivates this initiative, which is characterized such as highly prolific, with conceptual *i** variants and derivations, and having a wide set of isolated tools. We have presented our objectives where the focus is on three points: (i) the proposal of a interoperation language: iStarML, which has been already reached, (ii) the proposal of an interoperability scenario which helps to understand the roles of different *i** -related tools on software and organizational process, and (iii) to get a formal specification of the common and abstract conceptual framework. Finally we have presented 6 expected benefits for *i** models interoperability which, we claim, can be achieved with the iStarML proposal.

5 Ongoing and future work

In order to finish this initial step of proposing iStarML, we have planned to test this first version following two experiences of interconnecting *i**-related tools. One of them is already working as a prototype, connecting the J-PR*i*M editor subsystem with the Decision King variability modelling tool [4]. Also we are applying a survey to analyze the perception on iStarML constructs. We hope it obtains interesting feedback in order to improve the current iStarML version and get a revised new version. About the other objectives we are formulating interoperability scenarios which use *i** itself like representational mechanism, i.e. we are proposing agent-oriented socio-technical perspectives for *i** interoperability. Finally, our future work is planned to get a formalization of the common structure of iStarML which would allow discovering or avoiding inconsistencies and projecting new *i** variations and applications which can be developed under a shared interoperability scenario.

Acknowledgements

This work has been supported by the Spanish research project TIN2007-64753.

References

1. Ayala C., Cares C., Carvallo J.P., Grau G., Haya M., Salazar G., Franch X., Mayol E. and Quer C.: A Comparative Analysis of *i**-Based Agent-Oriented Modeling Languages. *Proceedings of the 17th Conference on Software Engineering and Knowledge Engineering (SEKE'05)* (2005) 43-50
2. Cares C., Franch X., Mayol E. and Quer C.: A Reference Model for *i**. To be published in *Social Modelling for Requirements Engineering*, E Yu, J Mylopoulos, N Maiden, and P Giorgini eds. Boston: MIT Press (2008)
3. Cares C., Franch X., Perini A., Susi A.: iStarML. The *i** Mark-up language. *Technical Report LSI-07-46-R*, Technical University Catalonia, Barcelona (2007)
4. Clotet R., Dhungana D., Franch X., Grünbacher P., López L., Marco J., Seyff, N.: Dealing with Changes in Service-Oriented Computing Through Integrated Goal and Variability Modeling. *Proceedings of the 2nd Workshop on Variability Modelling of Software-Intensive Systems (VaMoS'08)*

Multi-Agent Abstractions and Organizations and the i* framework

Luca Cernuzzi[1] and Franco Zambonelli[2]

[1]Universidad Católica "Nuestra Señora de la Asunción", DEI
Campus Universitario, Asunción, Paraguay
lcernuzz@uca.edu.py
[2]Università di Modena e Reggio Emilia, DISMI
Via Allegri 13, Reggio Emilia, Italia
franco.zambonelli@unimore.it

Abstract. i* is one of the more promising goal-oriented modeling framework to capture and model multi agent systems (MAS) requirements. Moreover, i* has been incorporated as the foundation for one of the more important AOSE methodologies, that is, Tropos. Our research interests related with i* cover two mainly directions. The first, is to compare i* with other methods, techniques and notations for capturing and modeling MAS abstractions. The second is related to the computational organization theory to model the organizational perspective of the MAS. Specifically, we are analyzing the critical issue of coping with adaptive changes of MAS organizations whenever circumstances claim for changes in the very MAS structure. Then, we are surveying different relevant AOSE methodologies, Tropos (and therefore i*) among others, to discuss their suitability in dealing with adaptation in MAS organizations.

1 Introduction

A great deal of efforts in the Agent-Oriented Software Engineering (AOSE) area focuses on the definition of methodologies to guide the process of engineering complex software systems based on the multi agent systems (MAS) paradigm. AOSE methodologies, as they have been proposed so far, mainly try to suggest a clean and disciplined approach to analyze, design and develop MASs, using specific methods and techniques.

Those efforts have to consider the relevance of requirements engineering for agent based systems. Like any paradigm, Agents-Orientation introduces different abstractions which have to be captured in the modeling of the problem and solution domains. However, several of the AOSE methodologies do not (or partially) cover the requirements elicitation phase [Cernuzzi, et al. 2005].

Different aspects have been considered to capture and model requirements for design and development of MAS: organizational, behavioral, domain, and goals model. The two mainly approaches consider the actors (or scenario-based) perspective (i.e. CREWS, AUML) and the goal-based perspectives (GBRAM, KAOS, i*). Then, it may be useful to analyze the benefits and limitations of such approaches for the AOSE methodologies.

Moreover, among the abstractions characterizing MAS, the organizational perspective covers a special role. In effect, the general behavior of the MAS strongly depends on the interaction among the different agents composing the global organization. This introduces the need to capture and model at least the social structure (including its topology), the static and dynamics relationships among agents and the rules and norms governing the global behavior.

In MAS organizations the types of interactions among component vary a lot, and may change during the time, depending on the goal of the system and the objective of each agent. In effect, MAS, as well as the great majority of modern software systems, are likely to be subject to a large number of adaptive changes during their lifecycle. Since, some changes may affect the very structure of the MAS, an AOSE methodology should not only facilitate the effective development of a MAS answering to specific requirements, but should also facilitate engineers and developers work whenever adaptive structural changes in the overall organization of a MAS are required.

Therefore, it may be interesting to analyze how AOSE methodologies (Tropos among them) with their corresponding processes, models and notations, facilitate designers to cope with adaptive changes that may have a global impact on the overall design of a MAS.

2 Objectives of the research

Our research interests related with i* cover two mainly directions each one with its specific objectives. The first, is to compare i* with other methods, techniques and notations for capturing and modeling MAS abstractions. The goal of such comparison is to eventually suggest relevant methods and notations to extend the existing AOSE methodologies which do not cover the requirement elicitation phase in their process.

The second direction is related to the critical issue of coping with adaptive changes of MAS organizations whenever circumstances claim for changes in the very MAS structure. Then, we are surveying different relevant AOSE methodologies to discuss their suitability in dealing with adaptation in MAS organizations. Among others, we are analyzing the Tropos methodology and consequently the i* framework, for modeling the organizational perspective considering the adaptive changes.

3 Scientific contributions

Considering the critical issue of capturing and modeling requirements for agent based systems, we think that some kind of analysis (evaluation) of methods, techniques and notation for capturing MAS requirements may improve the reliability of their adoption in the AOSE methodologies and, more generally, in the design and development of agent based systems. Therefore, we have already presented a first comparison among three different methods, namely RETO, Agentis, and GBRAM [Rodriguez et al 2008] and we are planning to extend such comparison to different approaches like i*.

On the other hand, the issue of continuous design change/adaptation in MAS organizations has been the subject of several studies. However, the specific problem of how to properly analyze, design, and develop a MAS so as to make it ready to adaptation is definitely under-studied [Cernuzzi and Zambonelli 2006].

Some recently proposed AOSE methodologies explicitly face the problem of structuring the organization of the MAS introducing some degree of modularity separation of concerns that make them more suitable for adaptive change. Among others, an interesting approach is offered by Tropos.

The Tropos framework aims at building agent-oriented software that operates within a dynamic environment. A curios thing is the Greek etymology of the word which means "easily changeable/adaptable". Tropos is mainly requirement-driven, adopting the i* organizational modeling framework, and views the information systems as social structures that is a collection of social actors, human or software, which act as agents, positions, or roles and have social dependencies among them. To capture the organizational perspective, Tropos includes actors diagrams for describing the network of social dependency relationships among actors (modeling an agent, a role or a set of roles), and rationale diagrams for analyzing and trying to fulfill the specified goals of the actors. Also in the architectural design phase, more systems actors are introduced and goals and tasks assigned to the systems are deeper specified in term of sub-goals and sub-tasks. As presented in [Cernuzzi and Zambonelli 2006], this clear focus of Tropos on the definition of the organizational structure is a key requirement for promoting adaptive organizational changes.

4 Conclusions

The results of comparative analysis of requirement engineering frameworks for MAS introduce the opportunity of improving those AOSE methodologies which do not (or partially) cover the requirement elicitation process. In this direction, we are proposing such improvement for the Gaia methodology.

On the other hand, as presented in [Cernuzzi and Zambonelli 2006] several AOSE methodologies offer relevant insights to deal with the adaptive changes in MAS organizations. However, most of the AOSE methodologies are concerned with the analysis and design processes only [Cernuzzi et al. 2005]; few are trying to cover the development and deployment of the system; less yet are concerned with the maintenance stage of the system. Thus, even when a methodology is more suitable for a design-for-change perspective, a specific attention to the maintenance process and the definition of proper guidelines for change and adaptation are lacking, which is a great limitation for modern methodologies.

As a final point, it is also worth outlining that the dynamism of modern scenarios and need of nearly continuous adaptive changes claim for evolutionary process models and, more specifically, agile extreme process models. However, current agile and extreme software process models focus on small- to medium-size projects, and are not yet ready to tackle the complexity of developing large-scale adaptive MAS.

5 Ongoing and future work

In the requirement elicitation direction we have already presented a first comparison among three different methods, namely RETO, Agentis, and GBRAM [Rodriguez et al. 2008]. We are planning to extend such comparison including i*.

Moreover, being the organizational modeling a central point for the current AOSE methodologies, we are analyzing how AOSE methodologies facilitate designers to cope with adaptive changes that may have a global impact on the overall design of a system (i.e., on the overall architecture/organization of MAS). For this purpose, different aspects may have relevant impact. The application of principles like modularity and separation of concerns, the adopted process of the methodology, the explicit modeling of relevant abstractions for organizations (i.e. organizational structure and control regime), among other factors, may help designers to choose a different organization whenever circumstances claim for changes. Therefore, in this on-going work we are surveying some relevant AOSE methodologies (Gaia, Tropos, Ingenias and Passi) to discuss their suitability in dealing with adaptation in MAS organizations.

References

1. Cernuzzi, L., Cossentino, M., Zambonelli, F.: Process Models for Agent-based Development, Journal of Engineering Applications of Artificial Intelligence, Vol. 18, No.2, (2005) 205-222.
2. Cernuzzi, L., and Zambonelli, F.: Dealing with Adaptive Multi-Agent Organizations in the Gaia Methodology, Proceedings of the 6th International Workshop on Agent-Oriented Software Engineering AOSE-2005, LNCS No. 3950, 2006, 109–123.
3. Rodríguez, L., Hume, A., Cernuzzi, L., and Insfrán, E.: Análisis Comparativo de Métodos de Elicitación de Requisitos para Sistemas Basados en Agentes, (accepted paper) XI Iberoamerican Workshop on Requirement Engineering and Software Environment - IDEAS 2008, Recife – Brasil, 11 - 15 February 2008.

A service-oriented approach for the *i** Framework[1]

Hugo Estrada[1,2], Alicia Martínez[1,3], Oscar Pastor[1], John Mylopoulos[4], Paolo Giorgini[4]

[1] Valencia University of Technology, Valencia, Spain
{hestrada, alimartin, opastor}@dsic.upv.es
[2] CENIDET, Cuernavaca, Mor. Mexico
[3] ITZ, Zacatepec, Mor. Mexico
[4] University of Trento, Italy
{john.mylopoulos, paolo.giorgini}@unitn.it

Abstract. The *i** modelling framework is widely used for organizational modeling. The framework focuses on strategic relationships between actors in order to capture the social and intentional context of an enterprise. Nowadays, many research projects exist that use the *i** framework in different applications domain. However, despite well-known theoretical advantages of the *i** modeling approach, there are certain issues that still need to be improved to assure their effectiveness in practice. In this paper, we propose a service-oriented approach in order to address the detected weaknesses of *i**. The business services can be used as the basic granules of information that allow us to encapsulate a set of *i** business process models where actors participate in actor's dependency networks through interfaces defined in the business service specification.

1. Introduction

New application areas such as e-Business, application service provision and peer-to-peer computing all call for very complex software systems which effectively support "on line" enterprise processes. To build such systems, practicing software engineers are discovering the effectiveness of using organizational modeling techniques for facilitating the elicitation of requirements for organizational information systems and also for guiding and supporting the software production process.

In this context, the *i** Framework is one of the most well-founded organizational modeling techniques today. In this framework, the focus of the modeling activity is placed on: a) the representation of the social and intentional relationships among the network of actors of an Enterprise, and b) the representation of the internal behaviors required to satisfy actor's dependencies. The *i** Framework supports the description of organizational networks made up of social actors that have freedom of action, but that depend on other actors to achieve their objectives and goals.

[1] This project is partially funded by the Asociación Nacional de Universidades e Instituciones de Educación Superior ANUIES, México

The *i** modeling concepts have been used in a wide range of application domains. In all applications, *i** concepts have been used to capture social and intentional elements of each specific domain, thereby supporting software development. However, even considerable research has been devoted to use the *i** concepts of very different domains, less attention has been paid to propose mechanisms to manage the complexity of the modeling activity and for improving the usability and scalability of the *i** models

In this way, despite the advantages of the *i** modeling approach, there are certain issues that still need to be improved to assure their effectiveness in real-life case studies. Solutions for these weak points are proposed in this research work as an initial response to the results of a previous empirical evaluation [1]. This research work explores the use of a service-oriented architecture for the *i** Framework in order to give a partial solution to the detected problems. To make the practical application of the service-oriented approach possible, this work presents the definition of the conceptual modeling language, the service-oriented architecture and the modeling method associated with the service-oriented proposal.

2. Objectives of the research

As main conclusion of previous practical evaluations of *i**, what is clearly requested is the need to extend the *i** framework with mechanisms to manage granularity and refinement in real-life projects. These mechanisms must allow us to create and represent an organizational model in a modular way. As solution to this problem, in this research work we have introduced a method that deals with the current drawbacks of *i**. Our proposed solution is founded on the concept of business service as a high-level concept that encapsulates fragments of an organizational model as composite business processes.

The main idea of this proposal is the representation of an organizational model as a composition of business services, where these services represent the functionalities that the organization offers to potential customers. In this context, the business services become the basic building blocks that allow us to represent a business model in a high-level, three-tiered conceptual architecture. Business services, business processes, and business protocols are the hierarchically interrelated three tiers that compose our service-oriented architecture.

In the proposed approach, the organizational modeling process starts with the definition of a high-level view of the services offered and requested by the organization. Each business service is then refined into more concrete process models according to the business service method introduced in this research work. The main advantage of this proposal, is that it provides a solution to manage granularity, refinement and reuse in the *i** Framework.

The aim of this proposal was to attempt to make the modeling process simple by making the social an intentional characteristics of *i** hidden for novel analysts, at least in early elicitation stages [2]. In order to do this, the proposed method uses a well-known elicitation mechanism, such as goal analysis, to construct a goal structure that is built in such a way that contains the organizational knowledge without explicit social relationships. Thus, a method has been proposed as part of the service-oriented method in order to transform the goal structure into an *i** strategic models

We argue that expressive power of the conceptual primitives that we have introduced in the *i** Framework enables the analyst to the better managing of the complexity of organizational modeling in practice.

Furthermore, the proposed method makes it feasible to use *i** as the starting point for a full software production process. In this process, the elaboration of the organizational model can be the cornerstone of the software process, because requirements modeling and conceptual modeling will be the result of a precise model transformation process where organizational aspects are correctly represented in the corresponding lower-level models. Given the advanced model-based software production tools that currently exist on the market, having extended tools to support a full software process that covers all the activities from organizational modeling to its corresponding final software product can become a reality [3][4].

3. Contributions

One of the main contributions of this work is the definition of a new methodological approach to address the enterprise modeling activity using *i**. The new approach is based on the use of building blocks for encapsulating organizational behaviors through the concept of business services.

In the context of the definition of the service-oriented modeling language, one of the contributions is the analysis of the current *i** modeling concepts in order to propose a revisited version of the concepts according to the proposed service-oriented architecture. The proposed modeling language overcome some of the problems that were detected in the empirical evaluation concerned with the current definition of the *i** modeling concepts.

Our research work proposes a specific modeling method according to the concept of business service. As a key point of the method, we proposed an extensive use of goals structures as an elicitation mechanism instead of starting the modeling process directly with the intentional concepts of *i**. The idea of hiding the intentional characteristics of *i** (at least in early elicitation stages) is to make the method more suitable for non-expert analysts in the use of *i** concepts. Therefore, another contribution of this work is the definition of a method to derive the goal refinement structures into the strategic models of the *i** framework in an automatic way. This proposal, that joins a goal-based elicitation process with the social aspects of the *i** strategic models, represents one of the contributions of this work over the current goal modeling techniques.

4. Conclusions

The *i** modelling framework is widely used for organizational modelling. The framework focuses on strategic relationships between actors in order to capture the social and intentional context of an enterprise. This paper presents our work on improving *i** as a business modeling technique based on a service-oriented approach.

Our solution is founded on the concept of a Business Service Architecture where encapsulated organizational units can only participate in actor dependency networks through well-defined interfaces. Our research work is based on the hypothesis that it is possible to focus the organizational modeling activity on the business services offered by the enterprise to their customers. Following this hypothesis, the proposed method provides mechanisms to guide the organizational modeling process based on the business service viewpoint. The proposed service-oriented architecture for the *i** framework permits that the monolithic structure of the *i** strategic rationale model can be broken down into several business services. The service are the building blocks that encapsulate a set of *i** business process models. This should help *i** to be successfully applied in real-life, complex projects.

5. Future work

With the proposed modifications made in this work, our intention is to overcome the current limitations that practitioners face when using *i** in its current state. In fact, these modifications are intended to both solve the problems that were detected and to make the practical application of the method easier. It is certainly necessary to evaluate whether these conclusions can be generalized in practice, making this the direction of our future work.

References

1. Estrada H., Martinez A., Pastor O., Mylopoulos J.: An experimental evaluation of the i* Framework in a Model-based Software Generation Environment. CAISE `06. LNCS, vol. 4001. pp. 513-527 Springer, Heidelberg (2006)
2. Estrada H., Martinez A., Pastor O.: Goal-based business modeling oriented towards late requirements generation (PDF). ER 2003. LNCS, vol. 2003. pp. 277-290 Springer, Heidelberg (2003)
3. Martinez, A., Pastor, O., Estrada, H.: A pattern language to join early and late requirements. Journal of Computer Science and Technology, special issue on Software Requirements Engineering. 2(5), pp. 64-70 (2005)
4. Martinez A., Pastor O., Mylopoulos J., Giorgini P.: From Early Requirements to Late Requirements: A goal-based approach. Proceedings of Eight International Bi-Conference Workshop on Agent-Oriented Information System (AOIS-2006), pp. 5-12. Luxembourg (2006)

A Metrics Definition Framework for *i**

Xavier Franch

Universitat Politècnica de Catalunya (UPC)
C/Jordi Girona, 1-3, 08034 Barcelona, Spain
franch@lsi.upc.edu

Abstract. Measuring is a central task in the any engineering discipline, and modelling is not an exception. The need for measuring properties over *i** models arises in different contexts. We have analysed this need over *i** models of organizations, projects, systems and architectures. As a result, we have proposed a framework which includes: (a) a metamodel of *i** that defines the elements that build the model; (b) a framework for the definition of metrics; (c) a collection of metrics (currently under construction) of different nature built with the framework. In this work, we present these basic concepts and discuss the applicability of the approach.

1 Introduction

Measuring is a central task in the Information Systems (IS) development process. Some measures are used to evaluate an already built IS, for instance, by establishing its size according to the number of classes or lines of code, or by checking that the resulting system accomplishes its non-functional requirements fit criteria (measuring the response time or failure rate, among others). However, measures can also be taken at the early stages of the IS development process, where they allow predicting some of the quality factors of the system-to-be, and planning corrective actions if needed. In this case, metrics are mainly defined over IS models of different kind. Having good suites of metrics allow not only analysing the intrinsic quality of an individual model, but also comparing different alternative models with respect to some properties in order to select the most appropriate alternative.

Among these models, we find the *i** framework. One of its strengths is versatility: it is currently used in different disciplines such as requirements engineering, business process modelling and reengineering, organizational modelling, or architecture representation, among others. In each of these contexts, an *i** model will be analysed with respect to different properties. The use of metrics may help to conduct these analyses. Therefore, the need for formulating metrics accurately and efficiently arises. In this paper, we report the work of the GESSI group in this subject, presenting a framework for the formulation of such metrics and some possible fields of application.

2 Objectives of the Research

Our work has the following objectives:

- To formulate a metamodel for *i** models (both SD and SR) allowing thus rigorous definition of the metrics on top of the structural form of those models. Our goal has been proposing a metamodel versatile enough to be adaptable and extendible to the different uses and variations of the framework that we may find in the literature. It has to be prepared also to enclose information that may useful for some metrics (e.g., time to perform some task), in addition to the structure itself.
- To define a framework for formulating metrics over *i** models. The framework is intended to provide some general forms of metrics such that defining a new metric basically means instantiating one or more of these forms to obtain the final definition.
- To build a catalogue of metrics using the framework. We basically address two types of metrics: general-purpose, or context-dependant. For context-dependant, we distinguish among domain-specific (e.g., for agent-based systems, databases, etc.) or activity-specific (e.g., for architecture modelling, project management, etc.).
- To propose and show different uses of the framework. We may use the metrics to compare solutions when exploring different alternatives to a problem, to analyse a model of a given system, etc.

3 Scientific Contributions

In [1] we formulated a metamodel for *i** capturing the specificities of the seminal proposal, GRL, and TROPOS variations, and we proposed refactoring as a technique to adapt the metamodel to a particular use. Remarkably, the metamodel includes all the concepts needed to build SD and SR models. Being a UML metamodel, the addition of information as mentioned in the first item of section 2, means just to add some generic attributes to the appropriate classes.

The framework was first outlined in [2] and refined in [3]. Remarkably, we distinguished 2 different axis to formulate metrics. The returned value axis for defining if a metrics is used for checking compliance (logical metrics), for measuring some concept (numerical metrics) or to obtain one or more model elements that fulfil some condition (model-element metrics). The subject of measure axis establishes which kind of model element is measured: we can measure the whole model (global metrics), individual elements (local elements) or even groups of individual elements (group metrics), e.g. pairs of actors). The same property may be measured with different metrics, therefore for strategic importance, we may have a numerical metrics to measure the strategic importance of each actor, a logical metrics to check if actors are over a given threshold, and a model-element metrics to obtain the most strategic actor. In [3], we also proposed the OCL as the language to formulate metrics over the metamodel.

We have formulated an initial (but yet incomplete) catalogue of metrics. In [3] we provide a comprehensive example of general-purpose, non-trivial metrics, predictability. In other works, we have addressed specific domains. In [4], we explored reengineering of software architectures over a documented case study (Home Service Robot) and for this purpose we defined over the *i** framework two classical metrics, coupling and cohesion. In [5], we targeted reengineering of software processes and we focused mainly on defining the functional size of a software system in the COSMIC-FFP framework (using then cfsu, COSMIC functional size unit, as metrics); we also included some results about process agility and ease of communication in the considered organizational alternatives. Other metrics are not currently available in the form of publications.

Concerning uses of the framework, our first interest was in comparing different requirements-oriented alternatives of COTS-based architectures [6]. Afterwards, as mentioned above, we have used *i**-based metrics as an important conceptual tool when comparing alternatives in reengineering systems. We have formulated a customizable reengineering framework [7] that has been customized both for software process reengineering [8] and architecture reengineering [4].

4 Conclusions

The use of metrics is very common in different type of models. For instance, there are some suites of metrics in the field of object-oriented modeling, which refer to structural properties like cohesion and coupling. Properties referring to the system itself, such as security, efficiency or cost, which mainly fall into the category of non-functional or organizational requirements, appear when considering models of the system architecture. For this reason, having metrics defined over *i** models is not surprising. In this paper, we have assumed this fact and then proposed a framework for formulating metrics over *i** models. The most significant contributions of the proposal are:

- Accuracy. We have provided a UML metamodel that is used as a baseline upon which the framework is built. Metrics are expressed with the OCL. We also have defined a method, R*i*SD, to build models in a systematic way [9], which also helps to get more accurate results.
- Expressiveness. The use of the OCL allows expressing metrics both in a comfortable and expressive way (although OCL is sometimes a bit messy).
- Sensitivity. Metrics can be defined more or less accurately depending on: 1) the expert judgement available; 2) the state of refinement of the model; 3) the effort we want to invest in model analysis.
- Easy tool support. The form that our framework takes allows implementation of tool support to drive metrics definition, model edition, generation of alternatives and evaluation of models. Our J-PR*i*M tool [10] is a first running prototype.
- Reusability. The metrics obtained may be reused in different projects of the same kind.

5 Ongoing and future work

As ongoing work, we mention: (a) Completing the catalogue with new, validated metrics constructed with them; (b) Incorporating the catalogue into J-PR*i*M going further than the current prototype.

Acknowledgements

This work has been supported by the Spanish research project TIN2007-64753. A significant part of the metrics framework and its application to reengineering activities has been developed jointly with Gemma Grau as part of her PhD thesis. The work related to the the metamodel formulation has been made jointly with other members of the GESSI group at the UPC.

References

1. Ayala C., Cares C., Carvallo J.P., Grau G., Haya M., Salazar G., Franch X., Mayol E. and Quer C.: A Comparative Analysis of *i**-Based Agent-Oriented Modeling Languages. *Proceedings of the 17th Conference on Software Engineering and Knowledge Engineering* (SEKE'05) (2005) 43-50
2. Franch X., Grau G., Quer C.: A Framework for the Definition of Metrics for Actor-Dependency Models. *Proceedings of the 12th IEEE International Requirements Engineering Conference* (RE'04) (2004)
3. Franch X.: On the Quantitative Analysis of Agent-Oriented Models. *Proceedings of the 18th International Conference on Advanced Information Systems Engineering* (CAiSE'06), LNCS 4001 (2006)
4. Grau G., Franch X.: A Goal-Oriented Approach for the Generation and Evaluation of Alternative Architectures. *Proceedings of the 1st European Conference on Software Architecture* (ECSA'07), LNCS 4758 (2007)
5. Grau G., Franch X.: Using the PR*i*M method to Evaluate Requirements Models with COSMIC-FFP. *Proceedings of the IWSM-Mensura 2007 Conference*
6. Franch X., Maiden N.: Modelling Component Dependencies to Inform Their Selection. *Proceedings of the 2nd International Conference on COTS-Based Software Systems* (ICCBSS'03), LNCS 2580, 2003.
7. Grau G., Franch X.: ReeF: Defining a Customizable Reengineering Framework. *Proceedings of the 17th International Conference on Advanced Information Systems Engineering* (CAiSE'07), LNCS 4495 (2007)
8. Grau G., Franch X., Maiden N.: PRiM: an *i**-based process reengineering method for information systems specification. *Information and Software Technology* (IST), 50(2), Jan. 2008.
9. Franch X., Grau G., Mayol E., Quer C., Ayala C., Cares C., Navarrete F., Haya M., Botella P.: Systematic Construction of *i** Strategic Dependency Models for Socio-technical Systems. *International Journal of Software Engineering and Knowledge Engineering* (IJSEKE), Vol. 17, N. 1, January 2007.
10. Grau G., Franch X., Ávila A.: J-PRiM: A Java Tool for a Process Reengineering i* Methodology. *Proceedings of the 14th IEEE International Requirements Engineering Conference* (RE'06) (2006)

AGFL - Agent Goals from Lexicon - Eliciting Multi-Agent Systems Intentionality

Antonio de Padua Albuquerque Oliveira [1, 2], Julio Cesar Sampaio do Prado Leite [2], Luiz Marcio Cysneiros[3]

[1] Universidade do Estado do Rio de Janeiro – UERJ
Rua São Francisco Xavier, 524 - 6 andar - Maracanã - Rio de Janeiro, Brazil

[2] Pontificia Universidade Catolica do Rio de Janeiro – PUC-Rio
Departamento de Informatica, Rua Marques de Sao Vicente 225 – Rio de Janeiro, Brazil

[3]York University – School of Information Technology
4700 Keele St. – Toronto, Canada

{padua, julio}@inf.puc-rio.br - cysneiro@yorku.ca

Abstract. "Agent-Oriented Software Methods" deals with agents' goals, commitments, beliefs, and abilities, but very few of these methods properly deal with the elicitation of these concepts. We believe that to build a goal model, first of all, we must accurately elicit and work with goals in several levels. Our approach named AGFL – Agent Goals from Lexicon is an indirect inquire process that can recognize goals via a simple elicitation strategy. In order to support this activity we will develop a software tool integrated with C&L, a software tool for lexicons and scenarios management. We posit that our approach helps the elicitation process and helps the production of more solid i* models.

Keywords: early requirements, GORE, Goal Oriented Requirements Engineering, MAS, Multi-Agent Systems.

1 Introduction

There seems to be a consensus that dealing with intentionality at early stages of software projects is a reasonable idea. When talking about intentionality we are directed to consider a goal-oriented approach and therefore, we need to understand and define why we are using goal modeling. The goal concept has come to play a critical role in Requirements Engineering (RE). In RE, goals are considered a significant construct. Various researchers consider GORE one of the best ways to produce quality software and therefore because MAS deals with agents' goals, the GORE approach seems particularly applicable to MAS.

This work is a summary of an earlier paper [4], the figures were adapted from there, and references were reduced in order to use four pages.

2 Objectives of the research

We face a common misuse, in the software engineering community, of the goal concept. Many people believe, wrongly, that a goal is like a function or an action that stakeholders can perform. Our research uses goals and softgoals in the same way used by the i* Framework [6]. And, in order avoid free style representations, which allow a goal to be represented like a function or an action; we adopted pre-defined frames that have the purpose of driving the requirements engineers to represent stockholder's intentionality.

One important gap in GORE approaches is the fact that GORE methods do not deal specifically with intentionality elicitation. All of them, no exception, are strongly oriented towards modeling.

Another motivation for this work relies on the common misuses of i* models pointed out by Estrada [1] and Pastor [5]. Ideally i* models should be divided into small pieces avoiding scalability problems and also improving the stakeholders' understanding.

3 Scientific contributions

In this work we introduce a method named AGFL – Agent Goals from Lexicon [4] showed in Figure 1, which is formed by tree steps. In the first activity ***"Elicit Actors' Goals"*** the engineer captures goals (and softgoals), separates them by actors, and organizes them in a chronological order.

For pushing up AGFL intentionality elicitation, the proposed method selected the Language Extended Lexicon (LEL) [2] because LEL promotes the capture of hints to find goals. As LEL captures the application vocabulary elements and classifies them as: subject, (someone who does the action) object, (something that receives the action) verb, (that means the action) and state (that is a result of the action), it provides a proper base to find application goals.

Our idea is simple: "ACTIONS CHANGE STATES AND STATES ARE GOALS". "A goal is a condition or state of affairs in the world that an actor would like to achieve" Yu [6].

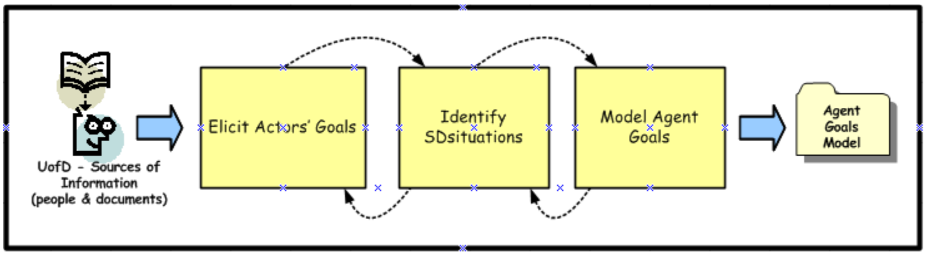

Figure 1 – Overview of the AGFL Method - (AGFL – Agent Goals from Lexicon)

In the second activity ***"Identify SDsituations"*** the engineer identifies goals (and softgoals) arrangements that are connected in order to implement situations of dependency called SDsituations – Strategic Dependency Situations [3].

Situations of dependency occur in the organizational environment and the central idea of SDsituations is: "each dependency link (goal, softgoal, task or resource) that involves actors is not isolated"; it is part of one well defined situation of collaboration called one "strategic dependency situation" or one SDsituation [3]. One SDsituation is composed by one or more dependency elements, and any SDsituation can be identified separately from other SDsituations forming a chain of interdependencies. An SDsituation can be characterized as part of the business unity. It means that we should identify several separate SDsituations but one depends on the others critically. Interdependencies among SDsituations may be physical, logical or temporal and can be represented in a specific diagram [3].

In the last one ***"Model Agent Goals"*** the requirements engineer builds diagrams, a kind of state charts that considers actors/agents, in order to represent chains of goals (and softgoals) relationships. The diagrams are called "INTENTIONALITY PANELS" and they are a simpler view of the i* Framework SR model.

The AGFL method suggests that intentionality should be drawn in parts based in SDsituations in a new diagram, called "Intentionality Panel" – IP diagram [4]. This diagram is a reduction of the SR model, it considers the i* "means-ends" structure being represented only by the structure end (goal or the softgoal) and the relationships between goals and softgoals are thus represented. An "Intentionality Panel" – IP diagram, is a kind of state-chart because it has different states linked together in a chain actors/agents' goals and softgoals.

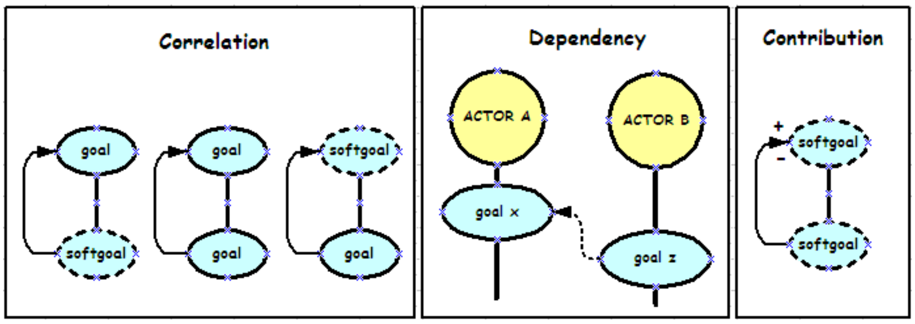

Figure 2 – It shows how the i* SR model can be reduced into one IP diagram.

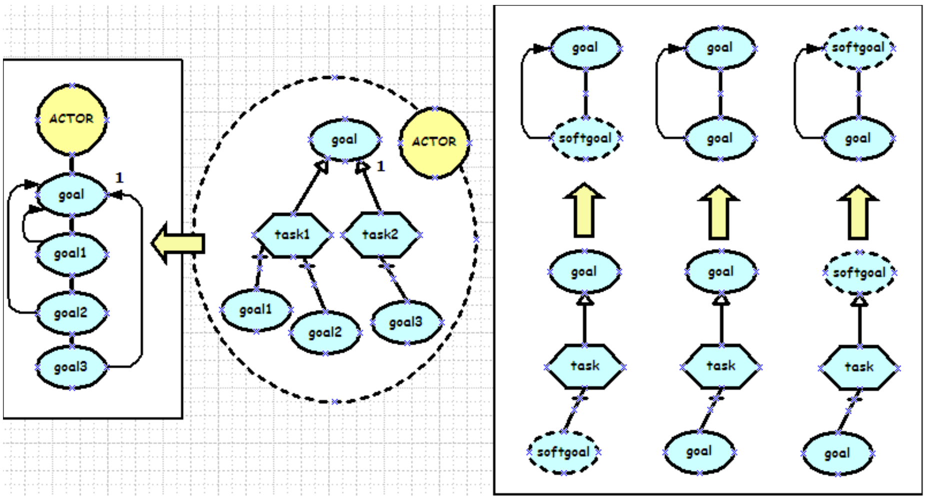

Figure 3 – In the right are represented the correspondences between SR model and the Intentionality Panel in the representation of **correlation**. In the left alternatives are represented. By one hand, goal[1] and goal[2] together they have correlation to the main goal be achieved but on the other hand, with ID=1, goal[3] has an alternative correlation to the main goal to be achieved

The first one activity ***"Elicit Actors' Goals"*** is partially supported by the C&L tool software, which is a management tool for lexicons and scenarios. C&L is an open tool developed by the Requirements Engineering Group at PUC-Rio and is available at http://pes.inf.puc-rio.br/cel/.

4 Conclusions

The method, AGFL, brings goal elicitation as the prime concern, towards properly supporting MAS development. The main contribution is to elicit agent goals by a method based on the Language Extend Lexicon [2] of the domain, which follows the simple idea represented bellow: "actions point to goals".

We have applied the AGFL method for the Insurance Company case study as a proof of concept. Our results are encouraging; however, more research in the use of the AGFL is necessary. We need to apply the strategy in different situations in order to get practical evidence of the benefits of applying the approach in real cases. While carrying out these experiments we will also evaluate how well the approach scales to more complex problems.

5 Ongoing and future work

We plan to continue the work in this direction by performing more analysis using the method AGFL. We need to apply the strategy in different situations in order to get practical evidence of the benefits of applying the approach in real cases. While carrying out these experiments we will also evaluate how well the approach scales to more complex problems.

Based in the method, we intend to implement a software tool supporting traceability and the baseline for requirements evolution. The baseline traceability support should allow the process **forward**: *UofD* → *LEL* → *SDsituations* → *IP diagram* and **backward**: *IP diagram* → *SDsituations* → *LEL* → *UofD*.

References

1. Estrada, H; Martínez, A; Pastor, O; Mylopoulos, J.; An Experimental Evaluation of the i* Framework in a Model-based Software Generation Environment; E. Dubois, K. Pohl (Eds.); CAISE 2006, LNCS 4001, pp.513-527, 2006. Springer-Verlag, Berlin Heidelberg, ISSN: 0302-9743 ; ISBN: 3-540-34652-X, 978-3-540-34652-4.
2. Leite, Julio C. S. P.; Franco, Ana P. M. A Client Strategy for Conceptual Model Acquisition, Proceedings of the International Symposium on Requirements Engineering, IEEE Computer Society Press, San Diego (1993), pp. 243-246.
3. Oliveira, A. Padua A.; Cysneiros, L. M.; "Defining Strategic Dependency Situations in Requirements Elicitation" The IX Workshop on Requirements Engineering; Rio de Janeiro, Brazil - July/2006.
4. Oliveira, A. Padua A.; Leite, J. C. S. P.; Cappelli, C.; Eliciting Multi-Agents Systems Intentionality: From Language Extended Lexicon to i* Models. In: International Conference of the Chilean Society of Computer Science, 2007, Iquique. Proceedings of the XXVI International Conference of the Chilean Computer Science Society. Los Alamitos: IEEE Computer Society Press, 2007. v. 16. p. 40-49 – ISBN 0-7695-3017-6.
5. Pastor, Oscar; Estrada, Hugo; Martínez, Alicia; The Strengths and Weaknesses of the i* Framework: an experimental evaluation i*, its Applications, Variations and Extensions. Eric Yu et als. (eds.) MIT Press (accepted for publication in 2006)
6. Yu, E. Modelling Strategic Relationships for Process Reengineering. PhD Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, Canada - 1995.

Fostering Investigation, Collaboration, and Evaluation: the *i** Wiki Experience

Gemma Grau[1], Jennifer Horkoff[2], Dominik Schmitz[3], Samer Abdulhadi[4], and Eric Yu[4]

[1] Universitat Politècnica de Catalunya, c/Jordi Girona 1-3, Barcelona E-08034, Spain
ggrau@lsi.upc.edu
[2] University of Toronto, Department of Computer Science, 10 King's College Road, Room 3302, Toronto, Ontario M5S 3G4, Canada
jenhork @cs.toronto.edu
[3] Fraunhofer FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
dominik.schmitz@fit.fraunhofer.de
[4]University of Toronto, Faculty of Information Studies, 140 St. George St.
Toronto, Ontario M5S 3G6, Canada
abdulhadisn@yahoo.com, yu@fis.utoronto.ca

Abstract. The *i** framework is becoming a widespread modeling technique that can be used in a wide variety of contexts. As a result, there is a large worldwide community of users. In order to allow better communication and dissemination of the work between different research groups, we have established a collaborative *i** wiki. This wiki allows *i** researchers, students and practitioners to obtain and publish information concerning related publications, *i** guidelines, existing *i** tools, case studies, and more. This paper presents our experiences setting up and managing the *i** wiki as well as its current and potential future offerings.

1 Introduction

In April 2005 at the 2nd *i** workshop, several research groups working with *i** met in London in order to share their work and experiences. As a result of the interchange of ideas we realized that it would be useful to have a place to share our *i** experiences in terms of publications, *i** tools, and the construction of a body of knowledge on the use of the *i** constructs. In order to facilitate this work we, the authors, have constructed a wiki and have begun to gather content.

The *i** wiki [1] is currently hosted at the chair of computer science 5 at RWTH Aachen University, Germany. It uses the TikiWiki platform [2], an open-source, MySQL- and PHP-based environment with a huge number of features. The main features that are currently activated are *wiki pages* and *articles* (some kind of news) together with *comment, forum,* and *FAQ* features as well as *image* and *file galleries*. More technically, the *category* feature allows classifying an object (wiki page, article, file etc.) and the *structure* feature allows providing an ordering on Wiki pages alleviating navigation within the wiki. A sophisticated *rights and permission* module allows for the provision of different kind of accesses, e.g. readability for non-registered users (except for some sections that are protected), editing capabilities for registered users, and further managing features for administrators.

The first content was published in September 2005, when the first users begin to register. Initially, the wiki contained sections for 'Events' (i.e., the *i** workshops), 'Publications' (initially obtained from Eric Yu's publication pages [1]), '*i** tools', a 'Who is who' relation of researchers and affiliations, and case studies. In September 2006, the first version of the tool comparison table was added by Gemma Grau. Some months later the *i** Quick Guide was added with a description of the main *i** constructs (again Gemma Grau together with Jennifer Horkoff and Eric Yu). Lately, the guide has been extended by Samer Abdulhadi to include the *i** guidelines.

Currently, the wiki has 70 registered users, with the wiki homepage having received 6681 visits. In the following, we will summarize the current as well as potential future content to show how the *i** wiki can be of use to *i** researchers, students, and practitioners, including how to contribute.

2 Current Offerings

2.1 Publications

Building upon Eric Yu's personal homepage [3], the publication section is intended to present current and past work relating to *i** that has been published in journals, at conferences or workshops. Currently, 16 categories have been created: requirements engineering; process analysis and design, reengineering; evaluation, verification, and validation; agent-oriented systems development; trust in multi-agent systems; security requirements engineering; software engineering processes and organizations; data management processes; knowledge management; systems and organizational architecture; enterprise architecture; business modeling; intellectual property management; variability and personalization; *i** modeling techniques; and metamodels. The number of publications in each category ranges from 1 to 37, resulting in a total number of 144 publications (not including the *i** roadmap section). Initiatives are underway to make use of a database in the backend in order to provide more sophisticated access and search features.

2.2. An Overview and a Comparison of *i** Tools

Aiming at a comparison of *i** modeling tools, we have created a questionnaire to evaluate and compare the existing *i** modeling tools. The questionnaire has been designed by first creating a quality model of the functional and non-functional characteristics that an *i** modeling tool should have, and then transforming this quality model into a questionnaire. The following categories are included: general information about the tool; *i** modeling suitability provided by the tool; usability facilities provided; maturity of the tool; and, extensibility and operability with other tools, which includes the facilities for importing/exporting files and the development of new functionalities. Currently, there are 12 tool evaluations in the *i** wiki, submitted by developers of each tool. In order to allow a better comparison of the introduced tools, there is also a comparison table which summarizes the main features using the criteria of the questionnaire.

2.3 The *i** Quick Guide and *i** Usage Guidelines

The *i** community has recently grown and successfully applied the *i** graphical modeling notation in numerous applications and settings. Review of published literature, however, indicates that some modelers have deviated in various degrees from the original *i** notation, causing inconsistencies among *i** users, a situation that calls for finding ways to settle on a common practice. Consequently, the *i** Quick Guide provides a glossary of the *i** constructs and how they should be combined according to their semantics. In conjunction with the *i** Quick Guide, the *i** usage guidelines provide assistance for the modeling purpose. The benefits of the *i** quick guide and the guidelines include enhancing the overall consistency and effectiveness of the *i** modeling processes, reducing variation in practice among users of the *i** modeling framework, and reduced errors for new *i** users.

The *i** usage guidelines are integrated into the glossary of the *i** Quick Guide to help the reader relate between the presented glossary and the associated guidelines. Each guideline deals with a common modeling concept and, in addition to its explanation, provides examples and discussion components, making them more understandable and usable by less experienced *i** users. Each guideline is annotated with initial attributes that indicate the type of guideline (Concept, Naming, Notation, Layout, Methodology, or Evaluation) and the level of guideline difficulty (Beginner, Intermediate, or Advanced). Currently, most of the guidelines are attributed as Beginner. Current attributes, however, could evolve as new and more elaborate guidelines are discovered and added to the *i** Guide.

The *i** Usage Guidelines, are intended to be both an introduction to *i** for new users and a reference guide for experienced users. The guidelines are intended to be flexible recommendations, serving as a catalyst for reflective feedback and future development. To facilitate these objectives, individual wiki pages for all the guidelines are made accessible to all registered *i** wiki users to comment and provide suggestions on individual guidelines. This collaboration aspect fosters an open environment for *i** users and researchers to contribute to charting new and creative ways of presenting, employing, and developing the guidelines.

2.4 Who is Who and Events

The who is who lists all researchers, students, and practitioners that have registered to the *i** wiki, or are otherwise known to be working with *i**, together with their affiliation. It currently contains 87 people from 12 countries. The event section is accessible for registered users only and is partly used to document meetings such as the *i** workshops, including presentations, photos, and research discussions. As with any other wiki page, contributions are welcome.

4 Future Work and Conclusions

In addition to the currently available content, we have considered adding sections to collect case study information and example models. This would create a repository of *i** models, a helpful resource for students learning the notation and researchers looking for subject models. Such a repository would allow various groups to share concrete work not includable in publications. However, issues such as confidentiality must be resolved.

In order to allow sharing the created *i** models between the different users and their tools, the iStarML [4] is being defined as the specific XML format for the *i** framework. A new section will soon be added to the *i** wiki, publishing the iStarML format, and allowing a collaborative discussion about the language. We believe that, the existence of iStarML will allow, not only to make all the *i** tools interoperable, but also to share the different examples and case studies between the researchers.

The *i** wiki has the potential to become a valuable reference and collaboration tool for *i** users. However, the success of the tool depends on having active members. Although many of the pages can be viewed without registration, any researcher, practitioner or student can become a contributing member of the *i** wiki. Simply send an email request to "istarwiki@i5.informatik.rwth-aachen.de" with your name, affiliation, email address and how you have learned about the site. So, if you want to add your publications, your tools, or participate in any of the sections, just register. Contributions are always welcome.

Acknowledgements. The authors want to thank Reinhard Linde for maintaining the server and keeping the TikiWiki software up to date, the University of Aachen for hosting the *i** wiki, and, especially, all *i** wiki collaborators.

References

1. The *i** wiki at: http://istar.rwth-aachen.de. Last visited: January 2008.
2. The tiki-wiki platform at: http://tikiwiki.org. Last visited: January 2008.
3. Eric's Yu home page: http://www.cs.toronto.edu/~eric/. Last visited: January 2008.
4. Cares C., Franch X., Perini A. and Susi A.: "iStarML. The *i** Mark-up language". Research Report, Universitat Politècnica de Catalunya, Departament de Llenguatges i Sistemes Informàtics, LSI-07-46-R (2007).

Ontological Foundations for Agent-Oriented Organizational Modeling

Renata S. S. Guizzardi, Giancarlo Guizzardi, João Paulo A. Almeida, and Evellin Cardoso

Computer Science Department, Federal University of Espírito Santo (UFES)
Av. Fernando Ferrari, s/n, Vitória, ES, Brazil
{rguizzardi, gguizzardi}@inf.ufes.br jpalmeida@ieee.org evellinc@gmail.com

Abstract. This paper reports on our work on three complementary directions, aimed at developing: (i) a theoretical framework named UFO (Unified Foundational Ontology) for (re) design, evaluation and integration of conceptual modeling grammars and models; (ii) the ARKnowD agent-oriented software engineering methodology to develop knowledge-centered systems and processes; and (iii) methodological contributions to business process modeling and reengineering. The aforementioned theoretical framework is in the basis of our work. Here, we describe our research agenda, along with its main scientific contributions and future directions.

1 Introduction

Using agents as human abstractions is motivated by the fact that, for specific problems, such as software engineering and business process modeling, agents may aid the analyst to abstract away from some of the problems related to human complexity, and focus on the important issues that impact the specific *goals, beliefs* and *tasks* of agents of the domain. This often leads to a clear understanding of the current situation, which is essential for the proposal of an appropriate solution. In the consideration of these cognitive and social concepts lies the strength of Tropos/i*, which comprehend the analysis of agent's motivations, tasks, applied resources, and so on. In the past few years, we have often run into situations in which goal analysis is shown to be relevant in understanding the problem domain. This includes grasping the wants and needs of Knowledge Management users [1] and visualizing business process models in different abstraction levels, allowing the elicitation of different sets of business and system requirements.

Although presenting clear benefits, it is a fact that the adoption of the agent-oriented paradigm remains elusive in software engineering and business process modeling practice. We claim that part of the adoption problem lies with the fact that cognitive and social concepts underlying agents are subjective and complex for the average practitioner. Specifically, although there are many efforts related to the topic in the areas of philosophy, cognitive sciences and computer science, a uniform and well-founded semantic view on these concepts is currently lacking.

Our proposal for such a semantic view lies in a Foundational Ontology named the Unified Foundational Ontology (UFO) [2]. This ontology has been successfully used to evaluate and interoperate two prominent agent-oriented languages, namely, TROPOS and AORML. The combination of these two languages is the core of the ARKnowD methodology [1], specifically targeted at developing Knowledge Management systems. UFO is now being applied to guide the integration of Tropos/i* to organizational modeling languages focused on the process modeling viewpoint (e.g., ARIS-EPC). This integration allows for establishing a principled link between the composition structure of business goals, on one side, and business events and activities, on the other. This integration enables us to address some important issues in business process modeling, such as goal-driven evolution of processes, and reasoning about processes on different levels of granularity.

2 Objectives

Our research agenda can be summarized as follows. Our first objective is to investigate the ontological nature of the social entities underlying the agent-oriented modeling paradigm. By doing this with the help of an interdisciplinary approach, we aim at defining a ***stable and sound formal theory*** which can be used as a foundation for agent concepts. Having this theory at hand, we intend to contribute to uncovering the concepts and viewpoints which are sufficient and necessary for creating ***agent-oriented organizational models*** in specific concern areas. In particular, we are interested in the areas of Knowledge Management and Business Process Modeling. Thus, a second objective of our research is to contribute to the *language and methodology unification efforts* of the agent community. Finally, we intend to apply the languages and methodologies which are the outcome of our evaluation, (re)design and unification processes in ***industrial case studies***. These case studies shall work as testbed and possibly empirical support both for these languages/methodologies as well as for their underlying ontological concepts. In the next section of this paper, we elaborate on these objectives discussing the scientific contributions of our work to the TROPOS/i* community. Our ongoing and future in these three research lines are elaborated in section 4.

3 Scientific Contributions

3.1 Ontological Foundations for Agent-oriented Modeling

In recent years, there has been a growing interest in the application of Foundational Ontologies, i.e., formal ontological theories in the philosophical sense, for providing real-world semantics for conceptual modeling languages, and theoretically sound foundations and methodological guidelines for evaluating and improving the individual models produced using these languages. In a number of publications (e.g., [2][3][1][4][5]), we developed a Foundational Ontology named the Unified Foundational Ontology (UFO).

In the past years, UFO has been tested as a theoretical framework to evaluate, (re)design and integrate concepts from many different conceptual modeling languages (e.g. UML [2]), reference models (e.g., RM-ODP [6]) and domain ontologies (e.g., the ODE Process Ontology [5]). Particularly in line with this workshop, UFO has been successfully employed to evaluate, propose modifications and provide real-world semantics to TROPOS/i* modeling constructs. Moreover, it has been used to promote the integration of TROPOS/i* with the AORML modeling language [1]. This result is the core of the ARKnowD methodology mentioned in section 3.2 and fits well with the on-going effort of the agent-oriented community towards a unification of existing modeling languages and methodologies. Motivated by our interest in Tropos/i*, our research has recently focused on the concept of goal, aiming at disambiguating its definition, discussing its different manifestations, and clarifying its relation to other important agent-related concepts [4].

3.2 Agent-oriented Knowledge Management

In [1][7][8], we propose the ARKnowD methodology to develop Knowledge Management systems. ARKnowD is an integrated agent-oriented methodology, which represents as agents all humans, organizations and information systems of the domain. This enables the analyst to understand their relations and interactions, guiding him on finding appropriate solutions to target the idiosyncrasies of that particular environment. Note that 'system' is defined as a *general set of interacting entities*, including but not being restricted to that of information system. This opens the possibility to consider several outcomes resulting from the application of our methodology, such as: changing organizational structures, modifying business processes, and adopting technological or non-technological tools.

One of the main principles of ARKnowD is the realization that there is no silver bullet when pursuing an agent-oriented engineering methodology, so the best approach is combining existing work according to the given domain or situation. This allows us to benefit from these works' modeling concepts and viewpoints, besides reusing modeling tools and other related developments. As previously mentioned, ARKnowD results of the integration of two modeling languages which are used in different phases of the agent-oriented development processes, namely, TROPOS/i* (for requirements engineering and architectural design) and AORML (for detailed design). Moreover, ARKnowD applies Constructivist theories as a theoretical background to guide the proposal of Knowledge Management solutions.

3.3 Business Process Modeling and (Re)Design

Business Process Modeling is about the description of sequence of business activities carried out in organizations in order to make them explicit. Hence, these processes can be effectively analyzed, communicated and used as knowledge assets of documented enterprise practices and activities. Together with modeling, Business Process Engineering also evokes two other areas of concern, namely *process evolution (or reengineering)* and *process abstraction (granularity)*. Reengineering is about the modification of current enterprise processes in order to make their results better fit to organizational strategies and goals. In a different perspective, it is important that the same business process and its composing activities can be seen in different levels of granularity in different phases of the process of analysis and (re)design, or for the sake of carrying out different problem solving tasks. For example, in a situation in which parts of an existing process are unstable, the designer may choose to increase the abstraction level of the existing process model before attempting at producing a redesigned model. The same can be the case for the transition between a conceptual model to an implementation model, i.e., for instance, one may choose to implement as a workflow specification a more abstract version of the initial conceptual model.

In our research program, the establishment of systematic relations between these aforementioned concern areas is made through the use of the UFO framework. In a nutshell, this is accomplished via the formal understanding of the semantic ties between, on one side, a viewpoint focused on the temporal ordering and compositional structure of processes and, on the other side, a viewpoint focused on the compositional structure of goals and the relations between goals and alternative activities that can fulfill those goals.

4 Ongoing and Future Work

In our first line of work, we intend to continue with the development of the ontological theories that constitute UFO. Moreover, as result of these developments, we expect to be able to provide real-world semantics to further agent-oriented and organizational concepts as well as contribute to the principled unification of additional agent-oriented modeling languages.

In our second line of work, currently, our efforts are focused on experimenting with the methodology in the health care domain. This is being pursued in the context of a project entitled Constructivist Knowledge Management in Health Care [9], funded by FAPES (Fundação de Apoio à Ciência e Tecnologia do Espírito Santo). The project proposes the analysis of the public health care sector in the state of Espírito Santo, with the main goal of enhancing the knowledge and information flow within the involved organizations.

Finally, in our third line of research, besides continuing with the analysis and integration of the process evolution (reengineering) and process abstraction (granulariry) concern areas, our current and near future activities in this area can be summarized as follows. Firstly, in the same spirit of section 3.2, we are producing an integrated modeling language that covers the aforementioned viewpoints by integrating TROPOS/i* (goal modeling viewpoint) with ARIS-EPC (temporal ordering viewpoint). Additionally, we are currently applying this integrated modeling language in industrial cases within the Petroleum and Gas industry [10].

Acknowledgements: This research has been partially supported by the FAPES DCR Grant #37274554/2007 and INFRA-MODELA Project FAPES/MCT/CNPq/CT-INFRA #019/2006.

References

1. Guizzardi, R.: Agent-oriented Constructivist Knowledge Management. PhD thesis, University of Twente, The Netherlands (2006)
2. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. PhD thesis, University of Twente, The Netherlands (2005)
3. Guizzardi, G., Wagner, G.: Towards Ontological Foundations for Agent Modeling Concepts using UFO. In: Agent-Oriented Information Systems (AOIS), selected revised papers of the Sixth International Bi-Conference Workshop on Agent-Oriented Information Systems. Lecture Notes on Artificial Intelligence. Volume 3508 of LNAI. Springer-Verlag (2005)
4. Guizzardi, R., Guizzardi, G., Perini, A., Mylopoulos, J.: Towards an ontological account of agent-oriented goals. In Choren, R., Garcia, A., Giese, H., Leung, H.f., Lucena, C., Romanovsky, A., eds.: Software Engineering for Multi-Agent Systems V: Research Issues and Practical Applications. Volume 4408 of LNCS. Springer-Verlag (2007) 67–81
5. Guizzardi, G., Falbo, R., Guizzardi, R.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. In: Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS'08), Recife/PE, Brazil (2008)
6. Almeida, J., Guizzardi, G.: On the Foundation for Roles in RM-ODP: Contributions from Conceptual Modelling. In: 4th International Workshop on ODP for Enterprise Computing (WODPEC 2007), IEEE International EDOC Conference (EDOC 2007), Maryland, USA (2007)
7. Guizzardi, R., Perini, A.: Analysing Requirements of Knowledge Management Systems with the Support of Agent Organizations. Journal of the Brazilian Computer Society (JBCS) - Special Issue on Agents Organizations **11**(1) (July 2005) 51–62
8. Guizzardi, R., Dignum, V., Perini, A., Wagner, G.: Towards an Integrated Methodology to Develop KM Solutions with the Support of Agents. In: Proceedings of the International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS'05), Waltham/MA, USA (2005)
9. Guizzardi, R.: Constructivist Knowledge Management in Health Care. Technical Report DCR Project number 37274554/2007, Fundação de Apoio à Ciência e Tecnologia do Espírito Santo, Vitória/ES, Brazil (2006)
10. Cardoso, E., Almeida, J., Guizzardi, G.: Uma Experincia com Engenharia de Requisitos baseada em Modelos de Processos. In: Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS'08), Recife/PE, Brazil (2008)

Qualitative, Interactive, Backwards Analysis of i* Models

Jennifer Horkoff[1], Eric Yu[2]

[1]Department of Computer Science, University of Toronto
[2]Faculty of Information Studies, University of Toronto
{jenhork@cs.utoronto.ca, yu@fis.utoronto.ca}

Abstract. Deriving the full benefits of i* models requires analysis and iteration beyond initial construction. This work outlines a procedure which allows backwards (top-down), qualitative, interactive analysis of i* models using SAT solving techniques. This approach expands on work in goal model reasoning, expressing i* models and qualitative evaluation values as a SAT problem. The nature of the interaction with the user will be explored. Future work will include an implementation of this procedure into the Eclipse-based OpenOME tool and application to a detailed case study.

1 Introduction

Much attention in the requirements modeling community has been paid to the use of i* models as tools for early requirements analysis. Deriving the full benefits of such models requires analysis and iteration beyond initial construction. To this end, various i* analysis methodologies and techniques have been proposed.

Previously proposed techniques for i*, or related goal model analysis, provide approaches which range in various dimensions including the level of automation, the direction of analysis, and the role of human intervention. Techniques introduced for reasoning with goal models allow qualitative evidence to be propagated in a backwards (top-down) manner, in a fully automated way using SAT solving techniques, and by separating positive and negative evidence [1]. In contrast, a method introduced for analysis of i* models allows qualitative evidence to be propagated in a forwards (bottom-up) manner, requiring interaction with the user to resolve combinations of positive and negative evidence ([2], [3]).

This ongoing work attempts to combine these two approaches, employing SAT techniques to allow backwards, qualitative, interactive analysis of i* models. This will allow for a new type of analysis to be used with i* models, providing the ability to ask new types of analysis questions.

2 Objectives of the Research

In order to develop a backwards, qualitative, interactive analysis procedure for the i* Framework, we intend to do the following:

- Expand work which expresses goal model analysis as a SAT problem [1] in order to express i* model analysis as a SAT problem.
- Determine what information is needed from the user, including how this information can be used, stored and reused.
- Analyze computational issues such as correctness and termination.

This work will briefly describe our progress toward these specific objectives.

3 Scientific Contributions

In the following section, we outline a procedure and framework which combines the work of [1] and [2], allowing for backwards, qualitative, interactive, i* analysis. We define the use of a SAT solver in an iterative procedure then briefly sketch how we express the required formal constructs. Finally, we provide an example for illustrative purposes.

3.1 Iterative Procedure

Generally, the procedure iteratively runs a SAT solver, given a SAT formula which encodes the forward and backward propagation of i* constructions, including target evaluation values for certain elements, and input elements, whose values, though propagation, produce these target values (line 1). The representation of i* models in a SAT formula allows elements to have both positive and negative evidence, as in [1]. However, to follow the conventions of [2], such evidence must be combined using human judgment. Therefore, once the SAT procedure is run with the formula encoding an i* model (line 3), the results are checked for softgoals that would normally require human judgment (have some combination of positive, negative, partial, conflicting or unknown evidence) (line 9). Starting with the softgoals which are closest to the targets, the procedure asks the user for one or more combinations of contributing elements which would produce the values for these softgoals required by the SAT solver (lines 10-12). These decisions are encoded in new clauses which are added to the SAT formula, replacing the previous clauses which applied to the softgoal (line 13). The entire procedure is run again (lines 14, 3). If the newly entered decisions do not result in a satisfying result, the user is prompted again to enter further possible combinations, if such combinations exist (line 5). Finally, if a satisfying assignment is found with SAT results such that no softgoals require human judgment, the procedure completes successfully, providing the necessary input values needed to produce the desired analysis values for the target elements (lines 15-16). The procedure is outlined in the high-level pseudocode shown in Fig. 1.

```
1    Create Φ, SAT Formula
2    Start
3    Run SAT(Φ)
4    If no satisfying assignment
5        If the user has been previously prompted for judgment, starting from the
         most recent, for each such softgoal prompt again, if new input is given
6            Update SAT Formula with new user input
             Goto Start
7        Else
             No satisfying assignment found, END
8    Else, a satisfying assignment is found
9        If human judgment is required for any softgoal
10           Find the softgoals requiring human judgment closest to target elements
11           Find the necessary analysis values for these softgoals according to SAT
12           Prompt the user for combinations of contributing analysis values which
                 would achieve these necessary values
13           Update SAT Formula with these values
14           Goto Start
15       Else, human judgment not needed on any softgoal
16           Success, return SAT results
```

Fig. 1. High-Level Summary of Analysis Procedure

3.2 Expressing i* Propagation as a SAT Formula

In order for the procedure outlined in the above section to work correctly, we must be able to represent the propagation of an i* model in a SAT formula. Similar to the approach in [1], we must formally define the construction of an i* model, define axioms for forward and backward propagation and describe how this information is combined into a SAT formula. The details of these formulizations are omitted due to space constraints. Generally, the formulations provided in [1] are modified to take into account i* syntax which is not used in goal graph constructs, including dependencies and differing types of contribution links. Currently the presence of actors or actor boundaries does not effect propagation as described in [2]. In addition, the formalizations are adjusted to account for the additional evaluation values of conflict and unknown. Once the axioms for forward and backward propagation are adjusted for use with i*, the SAT formula can be constructed in the same manner as in [1]. However, if more complex i* syntax, such as a mixture of incoming link types, were to be considered this formula may need to be altered.

3.2 Example

Consider the simple model in Fig. 2, representing password implementation choices. Take the example of the user choosing the partial satisfaction of Attract Users as a Target, with input goals of the tasks related to password implementation. During the first iteration of the procedure the user would be told that Attract Users needs to be partially satisfied and asked to input one or more combinations of analysis values for Security and Usability which produces this value. If the user indicates that both these elements need to be satisfied, the procedure will come back to the user asking, individually, what combinations of the input tasks would cause Security and then Usability to be satisfied. Assuming the user does not manually catch the conflict, she might say that Restrict Structure of Password and Ask for Secret Question need to both be Satisfied for the first question, and then Denied and Satisfied, respectively, for the second question. The procedure will then indicate that a satisfying assignment was not found, and return to the previous questions asking for further viable combinations for Security and Usability, and then, if no viable combinations are found for these elements, for Attract Users. After several iterations, the user may come to an agreeable combination, for example, with both tasks satisfied Security is fully satisficed while Usability has conflicting value, or, a satisfying assignment may not be found, indicating that the desired target values are not feasible.

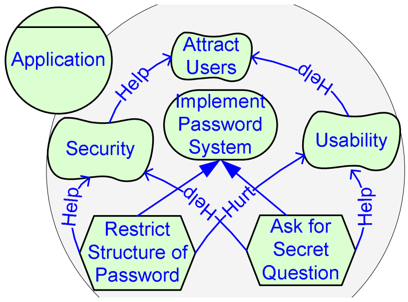

Fig. 1: Example i* Model and Forward Evaluation

4 Conclusions, Ongoing and Future Work

Preliminary investigations indicate that it is possible to implement a reasonable qualitative, backward, interactive analysis procedure for i*; however much future work is needed. Currently, we are developing proofs of correctness for the algorithm and SAT formula, including termination. We are also investigating ways to reuse the information provided by the users in future rounds of analysis, as well as ways to make the user interaction more user-friendly. Future work will include an implementation of this procedure into the Eclipse-based OpenOME tool and application to a detailed case study, testing its practical applicability.

References

1. Sebastiani, R., Giorgini, P., Mylopoulos, J.: *Simple and Minimum Cost Satisfiability for Goal Models*. CAiSE'04 (2004)
2. Horkoff, J.: *Using i* Models for Evaluation*. MSc Thesis, University of Toronto (2006)
3. Horkoff, J., Yu, E.: *Using the i* Evaluation Procedure for Model Analysis and Quality Improvement*. Presented at the Second i*/Tropos Worshop (2005)

Continuous, Requirements-Driven Support for Organizations, Networks, and Communities

Matthias Jarke[1,2], Ralf Klamma[1], Gerhard Lakemeyer[1], and Dominik Schmitz[2]

[1] RWTH Aachen University, Informatik 5, Ahornstr. 55, 52056 Aachen, Germany
{jarke,klamma,lakemeyer}@cs.rwth-aachen.de
[2] Fraunhofer FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
dominik.schmitz@fit.fraunhofer.de

Abstract. Next to requirements engineering, business process reengineering and organizational impact analysis have been discussed in Yu's original thesis as application areas of i*. In our research, we elaborate and combine these three fields by aiming at continuous, requirements-driven support for organizations, networks, and communities. While i* is well suited to capture the static relationships and rationales in these cases, extensions have been introduced to cope with the dynamics of these forms of organization, such as delegation discussions, evolution, the growth (or shrinkage) of trust, etc. We integrate a dedicated speech act perspective and provide a mapping to ConGolog, a logic-based high-level process modelling environment, to enable simulations. Analysis means are extended furthermore to include dynamic social network analysis.

1 Introduction

Modern forms of organization such as networks or communities of practice as well as traditional hierarchical ones do not automatically evolve smoothly and well in a self-organized fashion. All types have varying pitfalls and shortcomings that need to be addressed explicitly.

In organizations, strict processes tend to hinder effective interactions over time. Business process reengineering was a first attempt to address this problem and also one of the early application areas of i*. Currently, approaches that treat interchangeable business rules explicitly are considered to be promising.

On the other hand, inter-organizational or strategic networks have emerged as a new paradigm to address the requirements of nowadays volatile markets, culminating in the idea of the virtual enterprise. Various partners are intended to dynamically come together to temporarily join forces to answer market needs. Instead of stable hierarchies dynamic, trust-based relationships are supposed to provide the foundation of these networks thereby enabling flexibility and stability at the same time. But trust needs to be built up and maintained.

And finally, communities of practice are a currently heavily investigated field, mainly driven by the so called "Web 2.0" emphasizing the interaction of users in wikis, blogs, etc. Again such communities where people with similar interests exchange ideas are not free from problems. Disturbances such as trolls, i. e. persons that only interact in single-edge discussions they initiated themselves, need to be considered and coped with, especially if it is as easy and "cheap" to enter a community as in the web.

2 Objectives of the Research

All these forms of organization profit from a careful requirements-based investigation of conditions that influence their success or failure. But as research on equilibrium analysis has shown, it does not suffice to analyse only a steady-state, e. g. before the set-up. Continuous support that takes the dynamics and transitions into account is required to ensure a sustainable success.

Thus, the objective of our research is to provide a foundational framework for setting-up, managing, maintaining, and evolving organizations, networks, and communities involving human as well as technological actors. i* has proven to be suitable to capture the various relationships and rationales of the actors involved. Our multi-perspective modelling methodology [2] furthermore integrates a dedicated speech act perspective as well as a planning and simulation perspective. Additionally, we aim at a general integration of agent based approaches, including actor network theory and social network analysis to capture these complex problems by considering, resolving, and integrating multiple viewpoints. Building on earlier work, we use the knowledge representation language *Telos*, that also underlies, i* as the integration platform.

In addition to enabling formalized modelling and simulation of the object of investigation, we provide means to analyse situations and settings in order to gain insight into the dynamics, for example, regarding trust relationships. Dynamic social network analysis allows to investigate the effect of measures on the structure of the organization, network, or community as well as on other relationships, extending the currently available analysis support for i* and enabling a comparison with the intended outcome.

3 Scientific Contributions

Multi-Perspective Modelling Methodology to Capture Trust Together with sociologists we have developed a model of trust in networks that varies from existing approaches in that it considers three kinds of ingredients: trust, confidence (system trust), and distrust. Only a suitable balance enables the success of a network. A multi-perspective modelling methodology [2] has evolved around this model to provide support from a computer science perspective building on earlier work to capture cooperative processes in enterprises.

i Extensions Covering Dynamics to Enable Simulations* i* has been extended via a precondition/effect element and sequence links to operationalize the presentation in order to enable simulations. Furthermore, we provide a quantitative interpretation of softgoal contributions that enables the agents within our simulations to autonomously reason about various alternatives at run-time [3].

Mapping to ConGolog The above extensions aim at alleviating the automated mapping to the simulation environment especially in regard to clarifying ambiguities. ConGolog is a logic-based process simulation environment that builds on the situation calculus. The user of our SNet modelling and simulation tool starts by modelling the relationships of the organization, network, or community in a role based manner in i*. In the next step these generic roles are instantiated by agents that vary in regard to some details such as duration or contributions. After finally specifying the initial trust setting, simulations can be run that initiate the proactivities of agents which themselves result in delegations, i. e. complex interactions of the agents involved.

Combined Analysis Approach Although ConGolog has a formal foundation, it still only allows for simulations and is thus complementary to other existing approaches that build on model-checking (Formal Tropos) or Datalog axioms (Secure Tropos) [10]. From our perspective, Formal Tropos and Secure Tropos can be of help to analyse initial, intermediate, or final situations, i. e. snapshots from the dynamic simulations. For example, the user can be supported in evaluating the outcome of simulations and in adapting a setting to investigate a finding in more detail. The other way round, our simulation approach overcomes the instantiation limitations of a model-checking based approach. Also, the ability to consider the dynamic process of how a "system-to-be" can earn the trust of it users could be helpful to smoothly put the new or adapted system at work.

High-Level, Social Analysis Support Next to the above mentioned analysis means via Formal and Secure Tropos, social network analysis opens up a broad field of further analysis means that concern, for example, investigations on clustering, centrality etc. including extensions that consider their dynamic evolution over time. A graphical representation is often at the heart of these approaches and a corresponding toolkit has been developed that eases the application of these approaches [1]. Furthermore, we model our expectations in regard to emerging structures with i* and enable a matching with the outcome of social network analyses via a suitable pattern language [8, 9, 7].

4 Conclusions

We have applied our research approach in several projects. In a health care organization [6], trust relationships regarding the transfer of stroke patients between an acute ward and a rehabilitation ward have been analysed. Already a static investigation revealed considerable trust problems. Entrepreneurship networks [5] also involve trust relationships, but the dynamics are getting more important here. Due to the many complex features that need to be available such as network rules, agent evolution, monitoring, etc., it has not yet been possible to run a complete real world case study but simplified examples hint already on the potential of the approach. And finally, regarding communities disturbance patterns such as trolls on mailing lists have been investigated. They were easily captured in i* and the modelling was used to analyse large repositories of mailing lists.

Altogether the extension towards capturing dynamics seems a valuable step, that opens up new kinds of analysis. The means to cope with the large amount of data that results from this are already at hand (social network analysis) and currently integrated in our system (see next section).

5 Ongoing and Future Work

Current research concerns understanding and elaborating the intertwining of the various agent related theories, i*, actor network theory, and social network analysis and the potential of such an integrated methodology. Especially, the integration of the developed simulation facilities with the analysis means from dynamic social network analysis is currently targeted. Other areas include enhancing the modelling and simulation features by providing implementations for agent evolution, explicit consideration of network rules, and enabling monitoring to enable more realistic simulations [4]. Finally, we aim at applying our tool to a real world evaluation example.

Acknowledgment. This work was partially supported by the excellence initiative of the German government (UMIC cluster) and the Deutsche Forschungsgemeinschaft (SFB 427 and GK 643).

References

1. Y. Cao, R. Klamma, M. Spaniol, and Y. Leng. A toolkit to support dynamic social network visualization. In *VISUAL*, LNCS 4781, pages 512–523. Springer, 2007.
2. G. Gans, M. Jarke, S. Kethers, and G. Lakemeyer. Continuous requirements management for organization networks: A (dis)trust-based approach. *Requirements Engineering Journal*, 8(1):4–22, 2003.
3. G. Gans, M. Jarke, G. Lakemeyer, and D. Schmitz. Deliberation in a metadata-based modeling and simulation environment for inter-organizational networks. *Information Systems*, 30(7):587–607, 2005.
4. G. Gans, D. Schmitz, T. Arzdorf, M. Jarke, and G. Lakemeyer. SNet reloaded: Roles, monitoring, and agent evolution. In *AOIS*, LNCS 3508. Springer, 2004.
5. M. Jarke, R. Klamma, and J. Marock. *Zu den Wirkungen des regionalen Kontexts auf Unternehmensgründungen*, chapter Gründerausbildung und Gründernetze im Umfeld technischer Hochschulen: ein wirtschaftsinformatischer Versuch, pages 115–154. EUL-Verlag, 2003.
6. S. Kethers, G. Gans, D. Schmitz, and D. Sier. Modelling trust relationships in a healthcare network: Experiences with the TCD framework. In *ECIS*, 2005.
7. R. Klamma, M. Spaniol, Y. Cao, and M. Jarke. Pattern-based cross media social network analysis for technology enhanced learning in europe. In *EC-TEL*, LNCS 4227, pages 242–256. Springer, 2006.
8. R. Klamma, M. Spaniol, and D. Denev. Paladin: A pattern based approach to knowledge discovery in digital social networks. In *I-KNOW*, pages 457–464. Springer, 2006.
9. R. Klamma and S. Steinfels. Innovationsmanagement und Web 2.0. In *Analyse sozialer Netzwerke und Social Software - Grundlagen und Anwendungsbeispiele*, pages 273–302. GITO-Verlag, 2007.
10. D. Schmitz, G. Lakemeyer, and M. Jarke. Comparing three formal analysis approaches of the Tropos family. In *AOIS*, LNAI 4898. Springer, to appear.

Exploring i* characteristics that support software transparency

Julio Cesar Sampaio do Prado Leite[1], Claudia Cappelli[1]

[1] PUC-Rio – Pontifícia Universidade Catolica do Rio de Janeiro
www.inf.puc-rio.br/~julio
ccappelli@inf.puc-rio.br

This work presents the idea of software transparency. It posits that software transparency must be based on requirements, which will be the baseline for downstream traceability as well as upstream traceability. In that context, i* models are viewed as providing the support for several of the non-functional requirements that impact the software transparency NFR. In particular, we will explore the SA (strategic actor) model.

Keywords: Transparency, Softgoal, non-functional requirements, i-star, SA model.

1 Introduction.

Transparency has been, for long, a general requirement for democratic societies. The right to be informed and to have access to the information has been an important issue on modern societies. The demand for trust based on transparency has been increased in the context of global transformations. The importance of openness in the flow of information is creating an open society in which the very idea is to establish a democratic society with engaged citizens able to understand and use the information that is accessible to them [1]. However, is not sufficient to wish to be transparent. The organizations have to know what transparency is exactly and how they can demonstrate it.

Quoting Wordnet[1], transparency is: "(n) transparency, transparence, transparentness (the quality of being clear and transparent)" and "(adj) transparent [Related to: transparency] (easily understood or seen through (because of a lack of subtlety)) "a transparent explanation".".

We have been studying transparency as a non-functional requirement, and in [2] we have produced an initial mapping of several NFRs as listed in Chung et al. [6]. Figure 1 shows the transparency network we have mapped.

1. [1] WorldNet – A lexical database for the English language http://wordnet.princeton.edu/

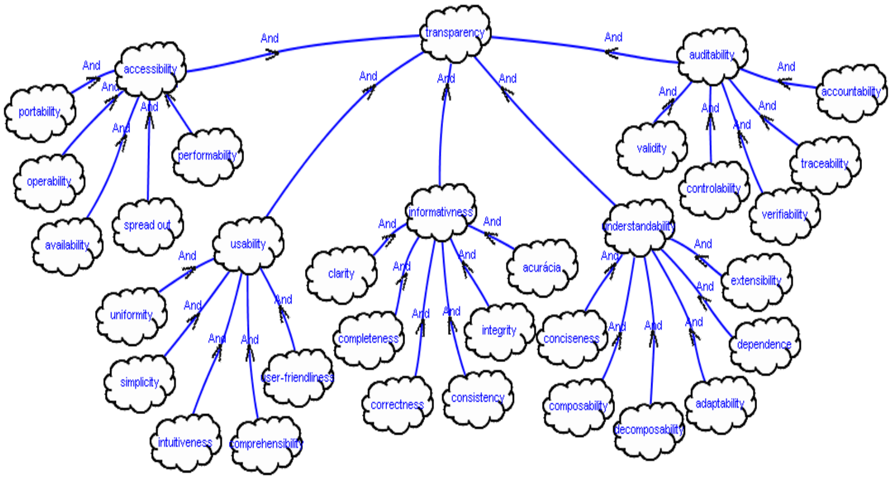

Fig. 1. Transparency Network [2]

From the transparency network we have posit an initial “transparency ladder”, which must be climbed as to achieve transparency. Figure 2 shows such ladder.

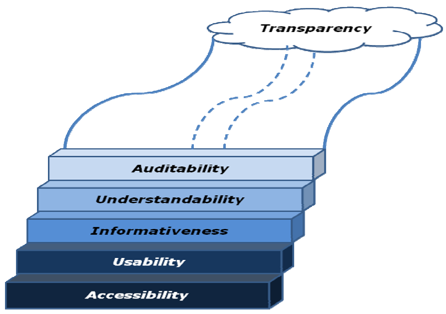

Fig. 1. Transparency Ladder

Software is deemed transparent if it makes the information it deals with transparent (*information transparency*) and if it, itself, is transparent, that is it informs about itself, how it works, what it does and why (*process transparency*). We tackle the problem of software transparency using the idea of requirements that are readable for both general stakeholders as developers’ stakeholders.

Our vision that software transparency should be based on requirements is best described by an observation by Professor John Mylopoulos. He states[2]:

[2] Personal Communication

"Transparency is an interesting quality because it makes it necessary to attach requirements models to software."

With this in mind, a requirements framework that allows both pre-traceability and post-traceability becomes central to the step Auditability as in the Transparency ladder, which has traceability as one of its components. Accordingly, we can refer to an upstream transparency to general stakeholders and downstream transparency to developers (code).

2 i* models as support for transparency

In [2] relations between transparency qualities (Figure 1) and "Quality Questions" (5W1H) were identified. Three business process modeling meta-models were compared and i* model ranked better since it covered most of the Quality Questions for each of the softgoals that compose the transparency network (Figure 1).

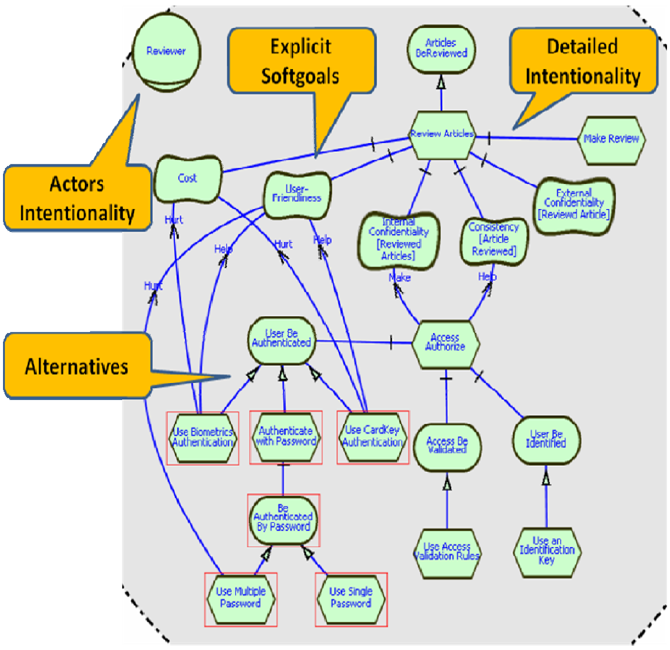

Fig. 3. An Instance of SD Transparency [3]

For instance, the SR diagram of Figure 3 presents 4 major points that address the "transparency ladder": actors' intentionality, explicit softgoals, alternatives, and detailed intentionality. By describing actors' intentionality we are addressing the NFRs (Figure 1) of traceability and verifiability, which contribute to the auditability step. By addressing explicit softgoals we are addressing the NFRs of completeness, clarity and accuracy, which contribute to the informativess step. By addressing detailed intentionality we are addressing the NFRs of decomposability and composability which contribute to the understandability step. The description of alternatives is important to the NFRs of integrity, extensibility and validity each contributing to a different step in the ladder.

3 The Strategic Actor Model and its role on Upstream Transparency

We understand that the Strategic Actors Diagram must be considered first class citizens as the other i* diagrams [4]. We understand that the SA model has to be first produced once information sources are being identified. In [5] we have proposed an influence graph upon which requirements engineering discussed and plot the relevant information sources. These information sources should be analyzed and those information sources that are actors should be modeled by an SA model.

Note that the SA model is used in different parts of the requirements process. First it is used as a map of information sources of the type actor and later on it is used to map the actors that will be related to the elicited goals.

Our research is focusing in SA model as an instance of upstream transparency. We will explore the relationship of the SA model with other representations such ontologies and business processes models as to improve the fitness of SA models to the steps of the transparency ladder

References

1. Holzner B., Holzner L. Transparency in Global Change: The Vanguard of the Open Society. University of Pittsburgh Press; 1 edition, 2006.
2. Cappelli, C., Pádua, A., Leite, J.C.S.P, Exploring Business Process Transparency Concepts, RE 2007, IEEE Computer Society Press, pp. 389-390, 2007.
3. Oliveira, A.P, et. al., Engenharia de requisitos intencional: tornando o software mais transparente, Tutorial presented at SBES´07, http://www.sbbd-sbes2007.ufpb.br/tuto3.pdf
4. Leite, J. C. S. P. ; Werneck, V. ; Padua, A. ; Cappelli, C. ; Cerqueira, Ana ; Cunha, H.S. ; González-Baixauli, B . Understanding the Strategic Actor Diagram: An Exercise of Meta Modeling. In:. Proceedings of the 10th Workshop on Requirements Engineering. Toronto : York University, 2007. v. 10. p. 2-12 (wer.inf.puc-rio.br/WERpapers)
5. Leite, J. C. S. P. ; Moraes, E. ; Castro, carlos . A Strategy for Information Source Identification. In: Proceedings of 10th WER, York University, 2007. v. 10. p. 25-34. (wer.inf.puc-rio.br/WERpapers)
6. Chung, L.; Nixon, B.; Yu, E.; Mylopoulos, J.; Non-Functional Requirements in Software Engineering – Kluwer Academic Publishers 2000 – Massachusetts, USA.

On Using i^* for Modeling Autonomy, Reasoning, and Planning in Adaptive Systems

Yves Lespérance[1] and Alexei Lapouchnian[2]

[1] Department of Computer Science and Engineering, York University, Toronto, ON, Canada, lesperan@cse.yorku.ca
[2] Department of Computer Science, University of Toronto, Toronto, ON, Canada, alexei@cs.toronto.edu

Abstract. Many systems are being developed where components have some form of autonomy and can adapt to change. However, it is not clear how these aspects of autonomous systems can be modeled by existing software development frameworks. How can we model how much autonomy an agent/component has? How can we specify when autonomous behavior, reconfiguration, or reasoning/planning is triggered? How can we model the constraints that control autonomy? We need a framework and process for developing such systems. In this paper, these questions are discussed and some possible approaches are outlined.

1 Introduction

More and more systems are being developed where components have some form of autonomy, be it autonomic computing systems that reconfigure themselves in response to changing conditions, workflow systems that adapt, multiagent systems (MAS) that coordinate, or individual agents that perform planning, all to better achieve their goals. In most of these systems however, when dynamic reconfiguration or reasoning/planning occurs and what autonomy the system has is constrained by the designer or user. For instance, planning is usually constrained by control knowledge and behavior in MAS is usually constrained by social rules. It is not clear how these aspects of autonomous/adaptive systems, can be modeled by frameworks such as i^* [1], Tropos [2], Gaia [3], etc., either at the requirements or design stages. How can we model how much autonomy an agent has? How can we specify when autonomous behavior, reconfiguration, or reasoning/planning is triggered? How can we model the constraints that control autonomy? In this paper, these issues and questions are discussed and some possible approaches are outlined.

In the development of such systems, a *requirements-driven* process as in Tropos should be followed. The choice of how much autonomy and adaptiveness an agent should have and constraints on it should be evaluated with respect to the system requirements, functional and non-functional. The objectives of individual agents should be derived from the system requirements, something that existing approaches to adaptive/autonomous systems development, e.g. [4], often fail to address. The model should support requirements traceability. In this paper, we discuss these questions and make some proposals for an appropriate development process and associated models.

2 Background

Wooldridge [5] defines the notion of an autonomous agent as follows: "An agent is a computer system that is capable of independent action on behalf of its user or owner. In other words, an agent can figure out for itself what it needs to do in order to satisfy its design objectives rather than having to be told explicitly what to do at any given moment." Adaptiveness is closely related to "reactivity", an attribute of agents that perceive their environment and respond in a timely manner to changes. Autonomous action and adaptation may require activities such as reasoning, planning, scheduling, optimization, coordination, and negotiation, especially if the set of objectives and environment conditions that the agents have to deal with cannot be enumerated in advance.

Some work has tried to identify conditions under which using an agent architecture is appropriate for an application [5, 6]. There have been many proposals for agent-oriented software engineering methods, e.g. Gaia [3], Prometheus [7], Tropos [2], etc. Some of these are closely tied to a particular type of agent programming framework (e.g. BDI (Belief-Desire-Intention) agent programming languages) or even a particular programming platform. Such models may implicitly assume a particular control regime, e.g. models of goal decomposition rules in BDI agent programming languages.

i^* already has some features that support modeling autonomous agents. The distinction between a goal dependency and a task dependency turns on whether the delegatee has the autonomy to select the means to achieve a goal (rather than being required perform a specific task). The presence of a goal in an SR diagram may mean that the agent has the choice of how to achieve it. However, it may be expected that the agent will use one of the tasks specified as means in the model to achieve it. Generally, SR diagrams are not assumed to be complete, and additional means to achieving a goal may be derived, even at runtime. But there is often an implicit assumption that one of the specified means will be used. It seems clear that in some cases, the modeler takes the intentional stance [8] towards an agent, ascribing some goals (and beliefs) to it and expecting it to behave rationally and attempt to achieve these goals. In other cases, the modeler takes a design stance, expecting the agent to make decisions and act according to the way it was designed. Which of these stances is taken typically depends on how much of the agent's design is known.

In i^*, an agent is viewed as capable of achieving a goal if it has a "routine" for achieving it. A routine is "an interconnected collection of process elements serving some purpose for an agent" [1], i.e. a plan skeleton. It seems clear that to support the specification of truly autonomous and adaptive agents and MAS, one needs models that are much less restrictive.

In previous work, our group has shown how i^* can be combined with ConGolog [9], a formal MAS specification/programming language, to support formal analysis/verification. Complete ConGolog models are executable and can also be validated by performing simulation. In this approach, i^* models are mapped into ConGolog by using an intermediate notation, annotated SR (ASR) diagrams [10], where process specification annotations are used to increase the precision and level of details of SR models.

Ordinary ConGolog does not support the specification of the intentional features of i^* models, i.e., the mental states of the agents in the system/organization modeled; these must be operationalized before they are mapped into ConGolog. But there is an extension of ConGolog called the Cognitive Agents Specification Language (CASL) [11] that supports formal modeling of agent mental states, incomplete agent knowledge, etc. Mapping i^* models into CASL gives the modeler the flexibility and intuitiveness of the i^* notation as well as the powerful formal analysis capabilities of CASL. We have extended the i^*-ConGolog approach to combine i^* with CASL and accommodate formal models of agents' mental states. Our intermediate notation has been generalized to support the intentional/mental state modeling features of CASL [12], in what we call intentional annotated SR (iASR) diagrams. With our i^*-CASL-based approach, a CASL model can be used both as a requirements analysis tool and as a formal high-level specification for a MAS that satisfies the requirements. This model can be formally analyzed using the CASLve [11] verification tool or other tools and the results can be fed back into the requirements model. One of the main features of this approach is that goals (and knowledge) are assigned to particular agents thus becoming their subjective attributes as opposed to being objective system properties as in many other approaches, e.g., Tropos [2] and KAOS [13]. This allows for the modeling of conflicting goals, agent negotiation, information exchange, complex agent interaction protocols, etc. However, this work does not support the modeling and analysis of many aspects of autonomous agent behavior, such as planning and reasoning.

Lapouchnian et al. [14] proposed a requirements-driven approach for designing adaptive systems with KAOS-like goal models (enriched with control flow annotations) that captured the variability in the way high-level system goals could be achieved. Various alternative ways of attaining these goals were analyzed with respect to their contribution to important quality criteria represented by softgoals. The system at runtime supports some or all of these alternatives and is able to switch from one alternative to another a) in response to changing user preferences over softgoals; b) in attempt to improve quality of service; c) as a result of a failure. This idea was further applied to the design and configuration of business processes (BPs) [15]. In this approach, the adaptivity of systems is limited to the alternative behaviors specified in the goal model. This favors predictability and trust in the system over adaptivity and autonomy. This technique may be a sensible choice for BP management and other applications where limited adaptivity may suffice, but it is not flexible enough to be used in a wide variety of adaptive systems.

3 Objectives

Here are some objectives that a framework for developing autonomous/adaptive systems should satisfy. First, it should support the specification of a wide range of types of autonomy/adaptivity (or lack thereof) that agents may possess, of what is known (and not known) about their design and decision making process, of the conditions under which reasoning or adaptation is triggered, and of what constraints apply to their decisions/behavior. This should rely on various models/stances that one can use to specify behavior while abstracting over design details.

Secondly, it should support analysis, allow predictions about agent behavior, and with a sufficiently detailed specification, formal verification, while remaining abstract. For instance, BDI-style specifications of agents together with specifications of their capabilities should allow reasoning about what goals will eventually be achieved under various conditions, even when the means cannot be specified in advance.

Thirdly, the framework should support the analysis of the merit of various alternative architecture designs with or without runtime reasoning and/or adaptation given the functional and non-functional requirements on the system. It should be possible to understand the benefits of doing more runtime reasoning in terms of increased robustness and improved solution quality, and its costs in terms of increased reaction time and unpredictability. The method should be requirements-driven. The objectives of individual agents and of the MAS itself should come from system requirements and the method should support requirements traceability. This is where using an i^*-based approach can help.

4 Some Proposals

There are two types of autonomy/adaptation that need to be modeled and analyzed: autonomy in individual agents, for instance through planning and reasoning, and adaptation in groups of agents, for instance through negotiation and coordination. Note that even the latter has an individual component as the negotiating/coordinating agents generally make individual decisions.

Here are some modeling and analysis techniques that could be exploited in a framework to achieve the objectives described earlier:

1. Tropos-style analysis of which parts of the system should be specified in advance and which should be left to be reasoned about at runtime; there are tradeoffs involved in making decisions about how much to specify, with effects on quality attributes such as predictability, trust, responsiveness, robustness, adaptiveness, autonomy, etc.;
2. extensions to the modeling language to support specification of actors or components as black boxes with behavioral constraints derived from high-level softgoals, and triggering conditions for reasoning and adaptation;

3. modeling extensions to support incomplete system specifications, such as weak constraints on the number of instances of an agent type and optional agent types/roles, and coordinator actors that constrain autonomy;
4. modeling extensions to support specification of the information that agents use to reason and make decisions.

In working on the design of the methodology/framework and its validation, it would be useful to model and analyze the use of various existing platforms for the implementation of autonomous agents, for instance:

- BDI agent programming languages that select plans to execute at runtime;
- agent programming languages that support runtime plan generation, such as IndiGolog [16] and CanPlan [17];
- other agent programming frameworks that support decision-theoretic planning, game-theoretic planning, or the use of deontic rules to constrain behavior;
- negotiating agents frameworks.

Moreover, there are many common applications where autonomous agents have been exploited that could be used to experiment with the framework:

- meeting scheduling systems;
- travel planning systems;
- systems that perform server load balancing or dynamic task allocation.

5 Conclusion

Autonomy and adaptiveness are qualities that are often required in state-of-art computer systems. Agent technology has been used to implement such systems. Agent-oriented software engineering methods have been proposed to help in designing them. Requirements engineering frameworks have also addressed the specification of such systems, while also incorporating agent notions such as "goals". Yet these development frameworks remain inadequate for modeling the features associated with autonomy and adaptiveness in systems. What is needed is a framework where these features and their connection with system requirements can modeled and analyzed. In this paper we have sketched how one might try to address this problem.

References

1. Yu, E.S.: Modelling Strategic Relationships for Process Reengineering. PhD thesis, Dept. of Computer Science, University of Toronto (1995)
2. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi-Agent Systems **8**(3) (2004) 203–236
3. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems **3**(3) (2000) 285–312

4. Kephart, J., Chess, D.: The vision of autonomic computing. Computer **36**(1) (2003) 41–50
5. Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley (2002)
6. Wooldridge, M., Jennings, N.R.: Pitfalls of agent-oriented development. In: Proc. of the 2nd Int. Conf. on Autonomous Agents (Agents 98), Minneapolis/St. Paul, MN, USA (1998) 385–391
7. Padgham, L., Winikoff, M.: Developing Intelligent Agent Systems: A Practical Guide. Wiley (2004)
8. Dennett, D.C.: The Intentional Stance. MIT Press, Cambridge, MA, USA (1987)
9. De Giacomo, G., Lespérance, Y., Levesque, H.J.: ConGolog, a concurrent programming language based on the situation calculus. Artificial Intelligence **121** (2000) 109–169
10. Wang, X., Lespérance, Y.: Agent-oriented requirements engineering using congolog and i^*. In Wagner, G., Karlapalem, K., Lespérance, Y., Yu, E., eds.: Agent-Oriented Information Systems 2001, Proceedings of the 3rd International Bi-Conference Workshop AOIS-2001, Berlin, iCue Publishing (2001) 59–78
11. Shapiro, S., Lespérance, Y., Levesque, H.J.: The cognitive agents specification language and verification environment for multiagent systems. In Castelfranchi, C., Johnson, W.L., eds.: Proc. of the 1st Int. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002), Bologna, Italy (July 2002) 19–26
12. Lapouchnian, A., Lespérance, Y.: Modeling mental states in agent-oriented requirements engineering. In Dubois, E., Pohl, K., eds.: Advanced Information Systems Engineering, 18th International Conference, CAiSE 2006, Luxembourg, Luxembourg, June 5-9, 2006, Proceedings. Volume 4001 of LNCS., Springer (2006) 480–494
13. Dardenne, A., Fickas, S., van Lamsweerde, A.: Goal-directed requirements acquisition. Science of Computer Programming **20** (1993) 3–50
14. Lapouchnian, A., Yu, Y., Liaskos, S., Mylopoulos, J.: Requirements-driven design of autonomic application software. In: Proc. of 16th Annual International Conference on Computer Science and Software Engineering CASCON 2006, Toronto, Canada, Oct 16-19. (2006)
15. Lapouchnian, A., Yu, Y., Mylopoulos, J.: Requirements-driven design and configuration management of business processes. In Alonso, G., Dadam, P., Rosemann, M., eds.: Proc. 5th International Conference on Business Process Management (BPM 2007), Brisbane, Australia, Sep 24-28. Volume 4714 of LNCS., Springer (2007) 246–261
16. De Giacomo, G., Levesque, H.J.: An incremental interpreter for high-level programs with sensing. In Levesque, H.J., Pirri, F., eds.: Logical Foundations for Cognitive Agents. Springer-Verlag (1999) 86–102
17. Sardiña, S., de Silva, L., Padgham, L.: Hierarchical planning in BDI agent programming languages: A formal approach. In: Proc. of the 5th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'06), Hakodate, Japan, ACM Press (May 2006) 1001–1008

From *i** to an Integrated Requirements Knowledge Representation, to Requirements for Services

Lin Liu

School of Software, Tsinghua University, Beijing, China, 100084
linliu@tsinghua.edu.cn

Abstract. This paper aims to review the research results from the past three years from the RE group at the school of software, Tsinghua University. Major threads of work include: (1) study the common body of knowledge on requirements modelling, and integrate *i** with other requirements languages, such as problem frames, UCM and UML; (2) work on a service capability and requirements modelling ontology – SRMO, based on goal and agent-oriented concepts from *i**; (3) build a double feedback loop control framework (*ASREF*) to achieve optimal service demand - supply relationship based on service models. (4) Tool development attempts related to *i** framework within the Tsinghua group.

Keywords Service capability, requirements model, feedback loop, requirements tools

1. Introduction

In retrospect, the *i** framework has gained growing attention in software requirements engineering research and industry in the past decade. It is recognized as a very different way of thinking that can be adopted by requirements engineers to understand the origin and focal point of a software problem within the organizational and social setting. It helps tackle problems at a different level of abstraction and depth. First, it provides us a set of graphical modelling constructs, so that the different kinds of elements identified from the original problem descriptions can be categorized and structured. Then, it provides us the reasoning mechanisms, such as task decomposition, means-ends analysis, softgoal satisficing level evaluation and dependency network exploration, based on which one can derive what questions to ask, identify where the missing bits and pieces of information are, and balance out the inequalities and conflicts among agents. Finally, it sets up a basic weltanschauung to look at the world, which is distributed, networked, social, strategic and intentional. Bearing such a viewpoint in mind, one can map the real world problem context to an analysis model based on *i** with minimum effort and difficulty. At all three levels, one can benefit from the *i** framework.

2. Research objectives

We have extended our research towards the following directions.

(1) Requirements Engineering Body of Knowledge (REBOK), and an integrated requirements modelling language based on i.*

Based on the above understanding to the *i** framework, it is natural to adopt *i** as a basic requirements knowledge representation language, and try to find how other existing requirements modelling languages relate and complement to it. The ultimate objective is to build a requirement ontology that incorporates as many perspectives as possible. So following the first attempt in integrating *i**(GRL) with UCM, we move to integrate *i** with the Problem Frames[1, 3]. It seems that there is considerable overlapping between the two languages. E.g. they both look into entities external to the system under development (actor vs. domains), and both focus on relationships between external entities and the system (dependency vs. interaction phenomena). The two also differs from each other obviously in that they emphasis on different aspects of the problems, one is at high-level, subjective, design-time decision making, and the other is at implementation-level, objective, run-time behaviors of the future system.

(2) Service capability and requirements modelling ontology – SRMO, based on goal and agent-oriented concepts from i.*

Since service orientation is becoming a dominant paradigm of the web-based software applications, a common feature of service orientation is the need to understand and characterize what the customer wants and to design services that can meet those requirements effectively. At present, user's requirements are often represented in certain existing standard interoperable service description languages such as WSDL/OWL-S. General service requestors may find such languages hard to use directly due to the reason that service requirements are often partially elicited and fragmented. The objective of this line of research is to develop a service requirements ontology SRMO, which extends the agent-oriented requirements modeling framework i* for early-phase requirements analysis with necessary language constructs for services requirements and capability modelling. [2,9,10].

(3) A double feedback control framework (ASREF) to achieve optimal service demand - supply relationship based on service capability and requirements models derived from i.*

This line of research aims to formulates the service-oriented requirements analysis process as a feedback control system, in which a classical "once for all" philosophy is replaced with a continuous negotiation and adaptation process based on existing requirements model and new service request.

3. Scientific contributions.

(1) An integrated requirements analysis approach based on i and problem frames.*

One of the difficulties that goal-oriented requirements analyses encounters is that the efficiency of the goal refinement is based on the analysts' subjective knowledge and experience. To improve the efficiency of the requirements elicitation process, engineers need approaches with more systemized analysis techniques. This work integrates the goal-oriented requirements language *i** with concepts from a structured problem analysis notation, Problem Frames (PF). The PF approach analyzes software design as a contextualized problem which has to respond to constraints imposed by the environment. The proposed approach is illustrated using the meeting scheduler exemplar. Results show that integration of the goal and the problem analysis enables simultaneous consideration of the designer's subjective intentions and the physical environmental constraints.

(2) Service capability and requirements modelling ontology – SRMO, based on goal and agent-oriented concepts from i [8].*

Along this line of work, formalism for service requirements and capability modeling is proposed. It adopts concepts from the agent-oriented requirements modeling framework *i**, which can be used as a means of studying the requirements and architecture for distributed agent systems. A social modeling framework such as *i**, extended with necessary service-related concepts and formal reasoning mechanisms, offers a better understanding of the social/organizational relationship in an open services world. By representing explicitly the underlying assumptions and the essential factors of services, a semi-formal requirements model in *i** can automatically evolve and be refined into a service requirements and capability reasoning framework. Eventually, it will assist intelligent agents with certain knowledge and intentions to make intelligent, rational decisions during service discovery, publication, selection, and binding within an open services community.

(3) A Service Requirements Elicitation Mechanism SREM based on SRMO and a double feedback control framework (ASREF) to achieve optimal service demand - supply relationship based on service capability and requirements models derived from i.*

An automated Service Requirements Elicitation Mechanism (SREM) is also proposed to help extract and accumulate relevant knowledge on service requirements. First, the SREM elicitation approach proposes to use a list of questions to narrow generic service requirements down to specific expressions of user preferences. Then, a service requirements and capability ontology is adopted to capture services requirements in breadth and precision. By integrating service requirements issued by different requestors, SREM provides non-trivial requirements guidelines and heuristic rules on service publication and discovery, also provided is a service requirements analysis mechanism that improves the accuracy of service discovery and efficiency of service composition continuously.

*(4) Tools under Development related to i**

A web-based modelling tool for i* is under development, which adopts a similar user interface as OME, but is operable with the web browser. We are also building tool prototypes to support the research works introduced above [10].

4. Conclusions

In summary, Ongoing work of the Tsinghua group include the investigation to a common requirements engineering body of knowledge, requirements engineering for services[5,7,8], and requirements engineering for trustworthy software [4, 6, 11]. The *i** framework provides a foundation of requirements knowledge representation and reasoning mechanism.

References

[1] Lin Liu, Zhi Jin. Requirements Analyses Integrating Goals and Problem Analysis Techniques, Tsinghua Sciences and Technology, 12(6). 2007.

[2] J. Xiang, L. Liu, W. Qiao, J. Yang. SREM: A Service Requirements Elicitation Mechanism Based on Ontology. . In Proceedings of the 31th Annual International Computer Software & Applications Conference (COMPSAC'07), Beijing, China. July 23-27, 2007:196-203.

[3] L. Liu, Z. Jin. Integrating Goals and Problem Frames in Requirements Analysis. (Extended Abstract) Proceedings of 14th IEEE International Requirements Engineering Conference (RE'06). Minnesota, Minneapolis, USA. September 11-15, 2006.

[4] L. Liu, E. Yu, J. Mylopoulos. Security Design Based on Social Modelling. COMPSAC'06. In Proceedings of the 30th Annual International Computer Software & Applications Conference, Chicago, Illinois, USA. September 17-21, 2006:71-76.

[5] P. Wang, Z. Jin and L. Liu. An Approach for Specifying Capability of Web Services Based on Environment Ontology. In: Proceedings of IEEE International Conference on Web Services. Chicago, Illinois, 18-22, 2006. 365-372.

[6] Lin Liu, Eric Yu: Modeling Identity Management Architecture within a Social Setting. APWeb 2006: 917-922. X. Zhou et al. (Eds.): Frontiers of WWW Research and Development - APWeb 2006, 8th Asia-Pacific Web Conference, Harbin, China, January 16-18, 2006, Proceedings. Lecture Notes in Computer Science 3841 Springer 2006, ISBN 3-540-31142-4. 917-922.

[7] Puwei Wang, Zhi Jin, Lin Liu: Environment Ontology-Based Capability Specification for Web Service Discovery. Formal Methods and Software Engineering, 8th International Conference on Formal Engineering Methods, ICFEM 2006, Macao, November 1-3, 2006, Proceedings. Lecture Notes in Computer Science 4260 Springer 2006, ISBN 3-540-47460-9: 185-205.

Peer Reviewed Workshop Papers

[8] L. Liu, Z. Jin, R. Lu. Towards Controllable Requirements Engineering Processes Based on Cybernetics. In Proceedings of the 4th IEEE International Workshop on Software Cybernetics (IWSC'07), Beijing, China. July 23-27, 2007:229-232.
[9] L. Liu, C. Chi, Z. Jin, E. Yu. Strategic Capability Modelling of Services. In: Procceedings of 2nd Workshop of Service-Oriented Computing Consequences of Engineering Requirements (RE-SOCCER'06). Minnesota, Minneapolis, USA. September 11-15, 2006.
[10] J. Xiang, W. Qiao, Z.. Xiong, T. Jiang, L. Liu. SAFARY: A semantic web service implementation platform. In: Proceedings of APSEC-SOPOSE'06. Bangalore, India, December 9, 2006.
[11] M. Zhu, L. Liu, Z. Jin, A Social Trust Model for Services, In: Proceedings of the 11the Australian Workshop on Requirements Engineering, Adelaide, Australia, 9th, December, 2006.

Extending *i** to Fit with the Requirements World

James Lockerbie & Neil Maiden

Centre for HCI Design, City University London, UK
[N.A.M.Maiden@, J.Lockerbie@soi.]city.ac.uk

Abstract. Whilst the *i** approach has been applied to case studies for some time, its wider uptake in industrial requirements processes and projects necessitates integration with established methods and techniques. However, there has been little reported integration. This paper will report how *i** has been integrated into a wider scenario-based requirements process, and summarize industrial uses of *i** on a recent food traceability project. Each of these projects has necessitated process and tool extensions to *i** to enable its uptake and use. The paper will report these extensions.

1. Introduction

Whilst the *i** approach has been developed and applied to case studies for some time, its wider uptake in industrial requirements processes and projects necessitates integration with established methods such as the Rational Unified Process (RUP) and effective requirements techniques such as scenario walkthroughs. However, there has been little reported integration so far, and this lack of integration threatens future uptake and industry-based evaluation of *i** and its underlying concepts. In this short paper we summarize previous and current research to integrate the *i** approach with other reported requirements methods and techniques.

2. Objectives of the Research

The objectives of the reported research are to investigate and evaluate the integration of the *i** approach with established requirements methods and techniques. If these objectives are successfully met, the outcomes will include an agenda of future research and knowledge transfer for the wider uptake and effective use of *i** in requirements processes and projects. Three specific research questions to which we are currently seeking answers to achieve the objectives are:

Q1: Can the strengths of the *i** approach in large-scale requirements projects relative to other requirements techniques be identified?

Q2: Can the strengths of the *i** approach in large-scale requirements projects be harnessed in established requirements methods to deliver quantitative and qualitative benefits to these projects?

Q3: Can we develop new software-based tools and techniques with which to use the *i** approach successfully in large-scale requirements projects?

Scientific contributions that seek to provide answers to these 3 questions are summarized in the next section.

3. Scientific Contributions

We have sought to answer the 3 research questions by developing and evaluating new requirements methods, techniques and tools that exploit the *i** approach.

The RESCUE process is a concurrent engineering process in which different modeling and analysis processes, including use of *i*,* take place in parallel. Each stream has a unique and specific purpose in the specification of a socio-technical system:

1. Human activity modelling provides an understanding of how people work, in order to baseline possible changes to it;
2. System modelling enables the team to model the future system boundaries, actor dependencies and most important system goals;
3. Use case modelling and scenario-driven walkthroughs enable the team to communicate more effectively with stakeholders and acquire complete, precise and testable requirements from them;
4. Managing requirements enables the team to handle the outcomes of the other 3 streams effectively as well as impose quality checks on all aspects of the requirements document.

The RESCUE process was reported at length in [1]. It is supported with an *i** modelling tool called REDEPEND, which is designed to provide systems engineers with *i** modelling and analysis functions, coupled with additional functionality and reliability of Microsoft Visio. It provides a graphical palette from which systems engineers can drag-and-drop *i** concepts to develop Strategic Dependency (SD) and Rationale (SR) models. REDEPEND also provides systems engineers with simple model checking functions for SD and SR models. It implements modelling constraints that, if activated, forbid a user to add or change a model element that violates *i** model constraints. Usability has been enhanced by, for example, adding new check features to highlight and shade-out model elements using layers, to partition and mark up models during analysis and review tasks, and to support *i** model colour-coding, which highlights model features during walkthroughs. Most of these features emerged from feedback on REDEPEND use in large-scale requirements projects.

In contrast, new productivity features were added to REDEPEND as results of academic research. For example we researched simple patterns – recurring syntactic and semantic structures in the *i** models – that can be applied automatically to any SD model expressed in REDEPEND to generate textual requirement statements. Our patterns are not traditional in the design sense – a solution to a problem in context. Rather each pattern defines one or more desired properties (requirements) on the future system that must be satisfied for the SD model dependency to hold for the future system. As such, the SD model, which has been signed off as complete and correct, informs further discovery and specification of requirements statements. The concepts and patterns underlying this approach are described at length in [2] and an application of the approach is reported in [3].

Research undertaken with NATS, the UK's air traffic service, resulted in a new version of REDEPEND to support the specification of satisfaction arguments [4] for *i** means-end links and the procedure to analyse the impact of software requirements on system-wide goals and soft goals. The procedures extend existing *i** model propagation rules [5] with domain knowledge imported through the satisfaction arguments, thus enabling effective use of domain assumptions in such propagation techniques for the first time. These procedures also addressed a pressing industrial need in NATS by providing techniques to relate system-wide safety-related goals to functional requirements of new software systems. Further details are in [6]

We have been applying RESCUE and REDEPEND on large-scale requirements projects including air traffic management projects reported elsewhere. The most recent application of *i** and REDEPEND has been on the TRACEBACK project. Assuring the total traceability of food and feed along the whole chain from production to consumption is a cornerstone of EU policy on the quality and safety of food. This is a complex procedure involving identification, detection and processing of a vast amount of information. TRACEBACK is developing innovative solutions based on micro-devices and innovative service-based architectures to provide innovative new information services to actors from primary food producers to consumers and health authorities. Solutions, which will include new micro-devices and a service-oriented reference architecture for food traceability called RATIS, are to be trialled on two major product chains – feed/dairy and tomatoes.

During the application of the RESCUE process a team of 3 analysts, all experienced with *i** and REDEPEND, produced *i** SD and SR models describing actors in the diary food chain, and the introduction of RATIS and micro-devices into this food chain. The models were developed using information from descriptions of current processes and workflows in the dairy food chains in Europe, one-on-one interviews with stakeholders who fulfil modelled actor roles in these food chains, *i** modelling workshops at project partner sites, and electronic distribution of SD and SR models to stakeholders for comment and feedback. Overall the process lasted 6 months. Key results are reported in *i** models.

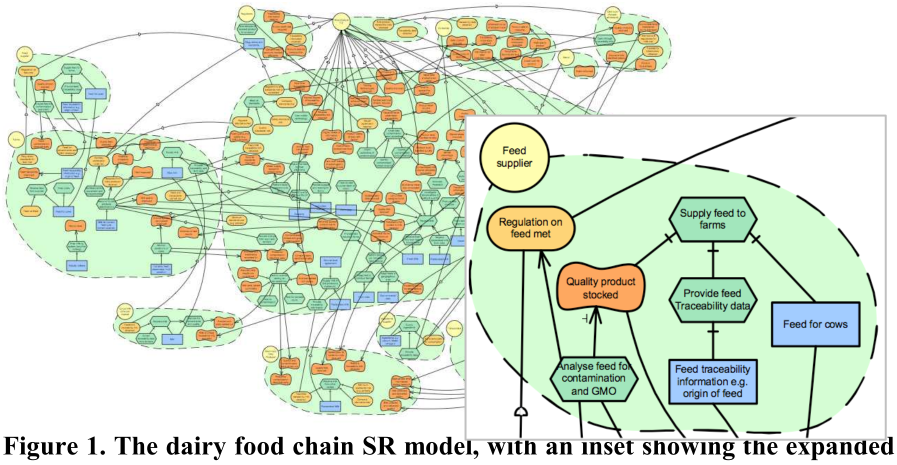

Figure 1. The dairy food chain SR model, with an inset showing the expanded food supplier actor

The SR model for the dairy food chain is depicted in Figure 1. The model specifies 14 actors, 251 different process elements and 257 different associations between these elements. The inset demonstrates part of the model – the *Feed supplier* actor – in a readable form. Using the requirements generation functionality of REDEPEND, we generated a set of requirements prompts directly from the SR model which we reviewed and refined. These prompts were integrated into the use case modelling and scenario-driven walkthrough phase of the RESCUE process. For example, the requirement prompts were used to create additional "what if" questions for the ARTSCENE scenario walkthroughs [7] and were also used by the facilitators as background reference material to aid the walkthrough facilitation.

4. Conclusions

The research is not complete, and we still need to refine the use and hence effectiveness of REDEPEND features including pattern-based requirements generation and refining *i** means-end links with satisfaction arguments. Another ongoing research challenge is to understand the trade-off between the simplicity and usability of the *i** notation, to understand the number and types of *i** modelling elements that requirements analysts can model and analyse effectively on requirements projects.

5. Future Research

Future research will continue to seek answers to the 3 research questions, in particular by trying to developing RESCUE and REDEPEND for effective use in large-scale requirements projects. If successful we will make both available to new exponents of *i** for use in their own requirements projects.

References

1. Maiden N.A.M.,, Jones S.V., Manning S., Greenwood J. & Renou L., 2004, 'Model-Driven Requirements Engineering: Synchronising Models in an Air Traffic Management Case Study', Proceedings CaiSE'2004, Springer-Verlag LNCS 3084, 368-383
2. Maiden N.A.M., Manning S., Jones S. & Greenwood J., 2005, 'Generating Requirements from Systems Models using Patterns: A Case Study', Requirements Engineering Journal 10(4), 276-288.
3. Ncube C., Lockerbie J. & Maiden N.A.M., 2007, 'Automatically Generating Requirements from i* Models: A Case Study with a Complex Airport Operations System', Proceedings 13th International Working Conference, REFSQ'2007, Trondheim Norway, Springer-Verlag Lecture Notes on Computer Science LNCS 4542, 33-47.
4. Hammond J., Rawlings R. & Hall A., 2001, 'Will It Work?', Proceedings 5th IEEE International Symposium on Requirements Engineering, IEEE Computer Society, 102-109.
5. Horkoff J. Yu E. & Lui L., 2006', 'Analysing Trust in Technology Strategies', Proceedings Privacy, Security, Trust Conference, Toronto, Canada, October 2006.
6. Maiden N.A.M., Lockerbie J., Randall D., Jones S. & Bush D., 'Using Satisfaction Arguments to Enhance i* Modelling of an Air Traffic Management System', Proceedings 15th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, 49-52.
7. Mavin A. & Maiden N.A.M., 2003, 'Determining Socio-Technical Systems Requirements: Experiences with Generating and Walking Through Scenarios', Proceedings 11th International Conference on Requirements Engineering, IEEE Computer Society Press, 213-222

Defining Inheritance in *i** at the Level of SR Intentional Elements[†]

Lidia López, Xavier Franch, Jordi Marco

Universitat Politècnica de Catalunya, C/Jordi Girona, 1-3, 08034 Barcelona, Spain
{llopez, franch, jmarco}@lsi.upc.edu

Abstract. The is-a relationship among actors has been introduced since the very beginning in the *i** framework. However, the effect of this construct at the level of intentional elements and dependencies is not always completely determined. In this paper, we explore the semantics of inheritance in *i** with focus on SR models. Aligning with its usual meaning in object-orientation, we distinguish 3 main notions to be defined: extension, refinement, and redefinition. For each of them, we have studied its effects on the different types of intentional elements and their links, and also dependencies, making explicit what can be and cannot be done. We have also analysed the proposal with an example that makes intensive use of inheritance, a multi-stakeholder distributed system in which different types of related stakeholders co-exist.

1 Introduction

Several variations of the *i** framework exist, for instance Yu's seminal proposal, GRL, and Tropos. They all agree on a core of main concepts whilst not addressing in much detail other related concepts (see [1] for an analysis). One of the elements whose definition is not complete is the concept of inheritance, despite the fact that it was incorporated in the framework since its early definition. Several authors make use of inheritance but they have not clearly defined this concept nor provided guidelines for usage. The reason for this lack of rigor in inheritance definition is that the construct is not needed often for some modeling tasks, and when needed, normally it just suffices with establishing inheritance of actors at the level of SD diagrams. But there are domains that need a more precise definition of inheritance.

As one of these domains, we have started to use the *i** language to model service-oriented multi-stakeholder distributed systems (MSDS). MSDS are distributed systems in which subsets of the nodes are designed, owned, or operated by distinct stakeholders. Using basic *i** modeling concepts such as intentional elements, links, and actors we experienced some limitations of *i** when specifying the needs of heterogeneous stakeholders in a particular example of system, a web-based travel agency [2]. A significant problem we faced when modeling this MSDS was caused by the need to use inheritance for building hierarchies of actors without knowing accurately the consequences on their rationale of doing so. Specifically, when

[†] This work has been supported by the Spanish research project TIN2007-64753.

modeling our MSDS system, we aimed to model a common rationale in the superactor and a specific rationale in the subactors. Using inheritance as defined by Yu, we felt the need to determine which model transformation operations are valid and which are their implications in the context of specialization of actors.

This paper reports our results in the definition of inheritance in *i** that were presented in the AOIS'07 workshop [3]. Also we have discussed the applicability of inheritance in the MSDS domain, reported in the VaMoS'08 workshop [4].

2 Objectives of the Research

The main objective of this research is: Presenting a complete and non-ambiguous definition of inheritance for the i* framework. This general goal may be split into:

- Studying the meaning of inheritance in the *i** framework. We are interested in exploring in which part of *i** models, and under which conditions, inheritance may be applied. Also, how inheritance affects subactor goals and dependencies. How the subactor goals can be modified to achieve these new dependencies or if it is possible that this modified behaviour can create new outgoing dependencies.
- Proposing a way to model inheritance in the *i** framework. As important as to define inheritance is how to model it. *i** makes intensive use of graphical elements to express actors' goals. So, we need an easy way to model inheritance. Of course, tool support is an important issue so that inheritance can be useful.
- Exploring how does inheritance affect to *i** treatments and properties. Treatments (e.g., backward reasoning) and properties (e.g., workability) are used to analyse models, so it is important that models that use inheritance can be also analysed.

We aim at validating the inheritance definition both formally and methodologically. For formal validation, we mean verifying that the proposal is sound and complete. For methodological validation, we mean to find out if *i** users like it, knowing if they would use it in their models. We are interested in:

- Knowing if the proposal is easy to learn. For this validation, we will ask some non-expert *i** users to use inheritance to model some academic examples.
- Knowing if our proposal is useful. For this property validation, we will present our proposal to *i** community using scientific events and the other means (e.g., the *i** wiki). We aim at applying this proposal to huge examples and even real projects.

3 Scientific Contributions

A goal of our proposal is to align *i** inheritance with the general concept of inheritance as known in OO approaches. After an analysis of existing options, we have decided to adhere to Meyer's *Taxomania rule*: "Every heir must introduce a feature, redeclare an inherited feature, or add an invariant clause". Upon adopting this rule in the *i** framework we obtain three different specialization operations on IE: extension (i.e., introducing a feature), redefinition (i.e., redeclaring an inherited feature), and refinement (i.e., adding an invariant clause):

- ***Extension***. In the OO paradigm, one of the most frequent ways of specializing a class is adding some information such as attributes and methods to a subclass. We extrapolate this idea into the *i** modeling framework and call it extension. Extension in *i** means adding new IEs to the SR model of a subactor together with relationships to other IEs. Any kind of IE, together with links and dependencies, may be added to the SR model of a subactor.
- ***Redefinition***. Redefinition ("redeclaration" in the Taxomania rule) allows redefining IEs and their relationships. The main difference among redefinition and refinement is that redefinition does not allow changing satisfactibility predicates. In the case of goals, tasks and resources, redefinition implies that the redefined IE (in the superclass) needs some decomposition using task-decomposition or means-ends links to make sense (otherwise, we would use extension or refinement). In the case of softgoals it is only possible to redefine the interpretation of the condition to be fulfilled (fit criterion), since they are not decomposed but just contributed.
- ***Refinement***. Refinement captures the third situation stated in Meyer's Taxomania rule, adding an invariant clause. We interpret adding an invariant as restricting the satisfactibility predicate of the IE being refined, in other words, satisfactibility of the new IE implies satisfactibility of the refined IE. More specifically, this means for the four types of IEs: (1) goals and (2) softgoals: the set of states attained by the new IE is a subset of the states attained in the refined IE; (3) tasks: the procedure to be undertaken in the new IE is more prescriptive than the procedure to be undertaken in the refined IE; (4) resource: the entity represented by the new IE entails more information than the entity represented by the refined IE.

As a result, specialization of an actor consists of several specialization operations applied to the inherited SR diagram. Extensions, refinements and redefinitions cannot not be arbitrary; conditions of applicability are explored in detail in [3].

Two important things that play a fundamental role in our approach are:

- ***Satisfactibilty***. Intuitively, an IE states some objective that may be satisfied or not. We assume that satisfactibility is denoted by a Boolean predicate. The exact meaning of satisfactibility depends on the type of the IE [3]. In extension and redefinition, the satisfactibility predicate does not change. However, by refinement, the satisfactibility predicate of an inherited IE is changed but not arbitrarily.
- ***Syntax***. The *i** framework heavily relies on the use of a graphical representation of its constructs. We want to apply a economy rule: Non-modified inherited IEs will not be included in the subactor unless strictly necessary. On the other hand, new IEs and modified inherited IEs will be included in the subactor SR using a solid line shape using the standard notation. When needed, non-modified inherited IEs will be included in the subactor SR using a dotted line shape; this is the only change in the standard use of *i** in our approach.

4 Conclusions

We have presented our first results towards defining in detail the concept of inheritance in the *i** framework. The main strengths of our approach are:

- It relies on the theory of inheritance as defined by some milestone references. Thus, it is compliant with the most recognised principles in this context.
- We avoided adding new constructs to *i**. This is an important issue since we avoid committing our approach to a particular version of the language. The only exception is in syntax (dotted lines for representing replicated elements).
- We have analyzed the effects of the several specialization constructs to the diversity of intentional elements, links and dependencies that are in *i** definition.

This work has been developed in the context of a collaboration with the Johannes Kepler University at Linz, Austria. In this context, the use of inheritance as a way of identifying candidate variation points in variability modelling is reported in [4].

5 Ongoing and future work

Our future work includes formalisation and the addition of inheritance into the *i** metamodel [1]. We will also focus on the specialization of dependencies in SD models and the transitivity of actor specialization. Another research question is to investigate the joint application of redefinition and refinement.

We are currently addressing these challenges also including research on adequate tool support for *i** inheritance. We are currently extending the model edition part of the J-PR*i*M tool [5] for supporting the inheritance concept, including exportation using iStarML [6].

References

1. Ayala C., Cares C., Carvallo J.P., Grau G., Haya M., Salazar G., Franch X., Mayol E. and Quer C.: A Comparative Analysis of *i**-Based Agent-Oriented Modeling Languages. *Proceedings of the 17th International Conference on Software Engineering and Knowledge Engineering (SEKE'05)*, 2005
2. Clotet R., Franch X., Grünbacher P., López L., Marco J., Quintus M., Seyff N.: Requirements Modelling for Multi-Stakeholder Distributed Systems: Challenges and Techniques. *Proceedings of the 1st International Conference on Research Challenges for Information Systems (RCIS'07)*, 2007
3. Clotet R., Franch X., López L., Marco J., Seyff N. and Grünbacher P.: The Meaning of Inheritance in *i**. *Proceedings of the 17th International Workshop on Agent-Oriented Information Systems (AOIS'07)*, 2007
4. Clotet R., Dhungana D., Franch X., Grünbacher P., López L., Marco J., Seyff, N.: Dealing with Changes in Service-Oriented Computing Through Integrated Goal and Variability Modeling. *Proceedings of the 2nd Workshop on Variability Modelling of Software-Intensive Systems (VaMoS'08)*, 2008
5. Grau G., Franch X., Ávila A.: J-PRiM: A Java Tool for a Process Reengineering i* Methodology. *Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06)*, 2006
6. Cares C., Franch X., Perini A. and Susi A.: iStarML Reference's Guide. Technical Report. LSI-07-46-R, UPC, Barcelona (2007)

Improving the Syntax and Semantics of Goal Modelling Languages

Raimundas Matulevičius

PReCISE, Computer Science Faculty, University of Namur, Belgium
rma@info.fundp.ac.be

Abstract. One major obstacle to requirements engineering (RE) is the growing complexity of today's systems. Such a complexity can only be fought efficiently by powerful abstraction mechanisms as incarnated by goal-modelling techniques. Unfortunately, the research efforts in this area are fragmented, which is a major impediment to a wide adoption by practitioners. In this work, we describe our approach how to aim the fragmentation by adopting a rigorous and novel approach for comparing and integrating goal modelling languages (GMLs). We investigate both syntax (using the principles for the effective communication) and semantics (using the UEML approach and the ISSRM reference model) and apply GMLs to solve domain specific problems (e.g. for security risk management). We hope to improve the coordination of research in this field, so that a comprehensive, sound, efficient, standard and tool-supported goal modelling language can emerge, be put into the hands of IS developers, and that the overall quality of IS developments can be improved.

1 Introduction

Goal-modelling languages (GMLs) have been a subject of research and experimentation for more than 15 years and have proved extremely valuable tools in a great number of situations. We can observe a host of GMLs and their variants– *i**, GRL, Tropos, NFR and KAOS. However, each GML comes with its own terminology, syntax, semantics, and process. In [1] Kavakli and Loucopoulos examined 15 GMLs and classified them along four dimensions: "usage", "subject", "representation" and "development". The authors identified that fragmentation appears at all levels. The languages have constructs that force developers to emphasise some aspects of the problem and neglect others. The more people work with one particular language, the more their thinking is influenced by this language, and their awareness of those aspects of the world that do not fit in, may consequently be diminished thus resulting in incomplete specification of the problem. Also, different issues within a problem situation may be relevant for different people at the same time, however not supported by the same GML.

Due to this fragmentation, we have not yet observed a widespread adoption of GMLs by practitioners. This is regrettable since RE is where GMLs are expected to have the highest payoff. In [1] authors have stressed the importance of more integration efforts to obtain a stronger GML that takes advantage of the many streams of goal-oriented research. In the literature we can find a number of attempts to unify GMLs at different levels, as well as to compare the meaning of their concepts following various approaches. However, none of them results in the systematic approach relating different GML aspects into the unified view.

In this paper we propose to yield a comparison and integration of GMLs. We present an on-going research, which analyses different GML quality aspects. The purpose is to develop an integrated and tool-supported GML, which would help improving the RE process. The structure of the paper is as follows: in Section 2 we introduce a research objective. Section 3 presents the recent contributions. Section 4 summarises our work and points out some future work.

2 Research Objective

The overall objective is the *comparison and integration of GMLs*. The objective is divided into four subgoals shown in Fig. 1. Firstly (*i*) we intend to assess the GMLs quality at the coarse-grained level using systematic evaluation frameworks. Secondly (*ii*) we evaluate the GML quality at the fine-grained level and define a precise syntax and semantics of GML constructs. Thirdly (*iii*) we compare tools that support modelling with GMLs. Finally (*iv*) we use the results of our comparisons to determine rules for language integration and model translation at both syntactic and semantic levels. The latter subgoal includes development of the integrated GML supported by a (prototype) tool.

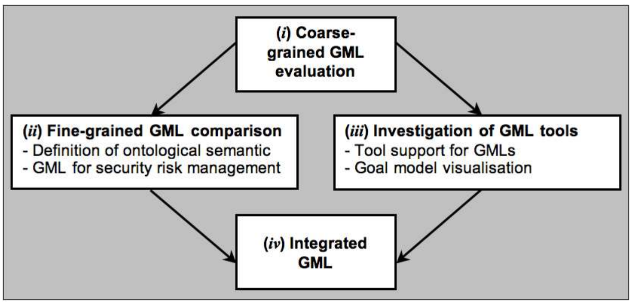

Fig. 1. Research method

3 Contribution

In this section we briefly present the contributions we achieved so far.

3.1 Coarse-grained GML Evaluation

In [2] we report on the experiment where two GMLs, namely *i** and KAOS, and models created using them are evaluated following the semiotic quality framework. This framework separates between different quality types (like semantic, syntactic, pragmatic and others). The experiment showed that the quality of individual goal models depends on particular language characteristics with respect to a given context. Even if one language is evaluated better than the other, this does not guarantee that the quality of the goal model would be better. Model quality much depends on the user's experience, the effort spent for model creation and the evaluator's subjective judgment.

3.2 Fine-grained GML Comparison

Definition of Ontological Semantics. In [3] we have applied the UEML approach [4] to investigate meaning of the GRL and KAOS constructs. Here a language construct can be described by (*i*) decomposing it to the represented classes, properties, states and transformations and by (*ii*) mapping them to the common UEML ontology. The study introduces a set of correspondences between two analysed languages. These correspondences can be used to translate GRL and KAOS models to each other based on their explicit semantics. This can help improve the traceability between models and tools used at different development stages. For instance, between early requirements elicitation using GRL, and late requirements specification using KAOS (or Tropos).

GML for Security Risk Management. We also investigated how GMLs are applicable for specific domains. We applied the reference model [5] of the information system security risk management (ISSRM) in order to check if concepts of Secure Tropos (which utilises *security constraints*) are adequate and sufficient for security risk management. The results indicate that Secure Tropos, firstly, has to be provided with guidelines as to when and how to use the constructs to avoid misinterpretations of ISSRM. Secondly, it should be improved with additional constructs to cover ISSRM better. In addition to Secure Tropos we have analysed Misuse cases [6]. We envision that after analysing a number of security languages it will be possible to facilitate model transformation to represent system security using different perspectives.

3.3 Investigation of GML supporting Tools

Tool support for GMLs. In [7] we have investigated goal modelling tools (e.g. OME, TAOM4E). We have observed that most of them are prototypes, thus requiring serious improvements before acquiring them to practice. To become more mature tools should be able to prepare and maintain not only the goal models, but the requirements specifications, too.

Model Visualisation. The problem with the goal models is that for the humans they quickly become difficult to comprehend the displayed information. In [8] we considered how to reduce the complexity of KAOS models using principles for effective communication. The current ongoing research involves analysis of the *i** framework languages and their supporting tools (TAOM4E, OME, and ST-Tool). We investigate scenarios which modellers could apply to create effectively communicating goal models. We also look for the visual cues (and supporting tool functionality) that facilitate preparation of the effective goal models and comprehension of the concepts provided in the *i** framework faster.

4 Conclusion and Future work

This paper presents an on-going research which aims to create the integrated GML. Currently we develop a metaCASE tool [9] using which we intend to generate a prototype tool supporting the integrated GML. The overall expected results of our study would contribute with (*i*) a thorough systematic scientific investigation and comparison of GMLs; and (*ii*) an integrated and tool-supported GML. The expected long-term benefits of GML analysis are improvement of the quality of the RE process. We hope to drive the research community towards a more rigorous way to define and extend (goal) modelling languages.

Acknowledgment. The author would like to acknowledge prof. Patrick Heymans and all the Requirements Engineering group at the University of Namur for contributing to and supporting this research.

References

1. Kavakli, E., Loucopoulos, P.: Goal Modeling in Requirements Engineering: Analysis and Critique of Current Methods. In: Information Modeling Methods and Methodologies. IDEA Group Publishing (2005)
2. Matulevičius, R., Heymans, P.: Comparison of Goal Languages: an Experiment. In: Proceedings of REFSQ 2007, Springer-Verlag Berlin Heidelberg (2007) 18–32
3. Matulevičius, R., Heymans, P., Opdahl, A.L.: Comparing GRL and KAOS Using the UEML Approach. In: Enterprise Interoprability II. New Challenges and Approaches. Springer-Verlag (2007) 77–88
4. Anaya, V., Berio, G., Harzallah, M., Heymans, P., Matulevičius, R., Opdahl, A.L., Panetto, H., Verdecho, M.J.: The Unified Enterprise Modelling Language – Overview and Further Work. In: Accepted at the 17th IFAC World Congress as invited keynote paper. (2008)
5. Mayer, N., Heymans, P., Matulevičius, R.: Design of a Modelling Language for Information System Security Risk Management. In: Proceedings of RCIS 2007. (2007) 121–131
6. Matulevičius, R., Mayer, N., Heymans, P.: Alignment of Misuse Cases with Security Risk Management. In: Proceedings of SREIS 2008, IEEE Computer Society (2008)
7. Matulevičius, R., Heymans, P., Sindre, G.: Comparing Goal-modelling Tools with the RE-tool Evaluation Approach. Journal of Information Technology and Control **35A**(3) (2006) 276–284
8. Matulevičius, R., Heymans, P.: Visually Effective Goal Models using KAOS. In: Proceedings of RIGiM 2007, Springer-Verlag Berlin Heidelberg (2007) 265–275
9. Englebert, V., Heymans, P.: Towards More Extensive MetaCASE Tools. In: Proceedings of CAiSE 2007, Springer-Verlag Berlin Heidelberg (2007) 454–468

Applying the i* framework to the development of data warehouses*

Jose-Norberto Mazón[1], Jesús Pardillo[1], Emilio Soler[2], Octavio Glorio[1], and Juan Trujillo[1]

[1] Universidad de Alicante, Spain
{jnmazon,jesuspv,oglorio,jtrujillo}@dlsi.ua.es
[2] Universidad de Matanzas, Cuba
emilio.soler@umcc.cu

Abstract. Data warehouse design has been traditionally guided by an in-depth analysis of the underlying operational data sources, thus overlooking an explicit stage in which information requirements of decision makers are addressed. This scenario has prompted that the deployed data warehouse often fails in delivering the expected support of the decision making process. To overcome this problem, we propose to use the i* framework for modeling goals and requirements within our *model driven architecture* (MDA) approach for the development of data warehouses. Our current and short term research also includes the reconciliation between information requirements and data sources, and the modeling of quality-of-service requirements (e.g., security). Finally, an Eclipse-based tool is being implemented as a proof of concept of our research.

1 Introduction

Data Warehouse (DW) systems are used by decision makers to analyze the status and the development of an organization. These systems are based on large amounts of data integrated from heterogeneous sources into multidimensional (MD) models, which are special data models allowing data access in a way that comes more natural to human analysts. Generally speaking, designers depict data into facts and dimensions in a conceptual MD model. Facts are usually measures of business processes (e.g., how many products are sold, how many patients treated, how long something takes, etc.), and dimensions represent the context for analyzing these measures (e.g., time, customer, product, etc.).

Since the DW integrates several operational data sources, the development of conceptual MD models has been traditionally guided by their detailed analysis. However, several studies have pointed out that most of these conceptual MD models fail to address the required information as a result of a poor communication between DW developers and decision makers. Actually, information

* This work has been supported by the ESPIA (TIN2007-67078) project and by the grants AP2005-1360 and AP2006-00332 from the Spanish Ministry of Science, and by the DADS (PBC-05-012-2) project from the Castilla-La Mancha Ministry of Science.

needs cannot be understood by only analyzing the operational data sources, and a requirement analysis stage is needed in order to model the information requirements of decision makers. Moreover, from this requirement model, a suitable conceptual MD model can be derived and reconciled with the available data sources. Importantly, this stage should be based on a *goal-oriented requirement engineering* (GORE) framework since (i) the DW aims at providing adequate information to support the decision making process, thus helping to fulfil goals of an organization, (ii) requirements for DWs are difficult to specify from scratch, since decision makers often only express general expectations about which goals the DW should support, and (iii) DW systems have a lot of kind of stakeholders with different interrelated goals that must be modeled to easily obtain a conceptual MD model that satisfy them.

The remainder of this contribution is structured as follows: objectives of our research are described in the next section. Our scientific contributions are presented in section 3. Our conclusions are reported in section 4. Finally, in section 5, ongoing and future work are sketched.

2 Objectives of the research

Our research is focused on defining a GORE approach, based on *i**, for modeling goals that the DW supports, thus easier obtaining information requirements. Furthermore, this approach is combined with our *model driven architecture* (MDA) framework for the development of DWs that has been described in [4]. This framework is based on defining a *computation independent model* (CIM) which addresses goals and requirements, a *platform independent model* (PIM) to specify MD properties at the conceptual level, and a *platform specific model* (PSM) tailored to a specific database technology. Therefore, *i** is used for defining a CIM, while a PIM for MD modeling is derived by establishing a formal transformation between these models via the *query/view/transformation* (QVT) language. The main advantage is that the conceptual MD model, represented in a PIM, meets every goal and requirement defined in the CIM. Furthermore, this PIM obtained from requirements is reconciled with data sources to obtain a hybrid PIM that provides the adequate information to fulfil business goals without disagreeing with data sources [3]. It is worth noting that combining *i** and MDA in DW development, via the use of the QVT language, assures the traceability between goals, requirements and the necessary MD elements related to them. This is an advantage of our proposal, since other works only propose informal guidelines to obtain a conceptual MD model from information requirements which also prevents the automatization of the process.

Finally, the DW is not just *data* but a whole system, where users require that the information has some characteristics when it is provided (security, performance tuning, etc.). These characteristics are constraints that the DW must fulfil to satisfy user expectations. We have named them quality-of-service (QoS) requirements, because they are additional issues that must be fulfilled by the DW to add quality in the way that the information is supplied and used. Informally speaking, information requirements answer *what* information the DW is expected to provide, and QoS requirements answer *how* this information should be provided for a right use. Therefore, these QoS requirements must be considered in the CIM by extending the i* notation. We have first focused on security requirements, since the extreme importance of the information managed by DW systems makes essential to specify security issues from the early stages of the MD modeling process, and enforce them [5].

3 Scientific contributions

To fulfil our research objectives, the *i** modeling framework and MDA have been integrated via the profiling mechanism of the *unified modeling language* (UML). In this way, *i** has been adapted to requirement analysis in DWs, allowing us to model both information [1, 2] and security requirements [6] at the CIM level (see Fig. 1). Moreover, in [3], we have developed an approach for reconciling data sources and requirements based on a set of *multidimensional normal forms* which assure several desirable properties in the conceptual MD model.

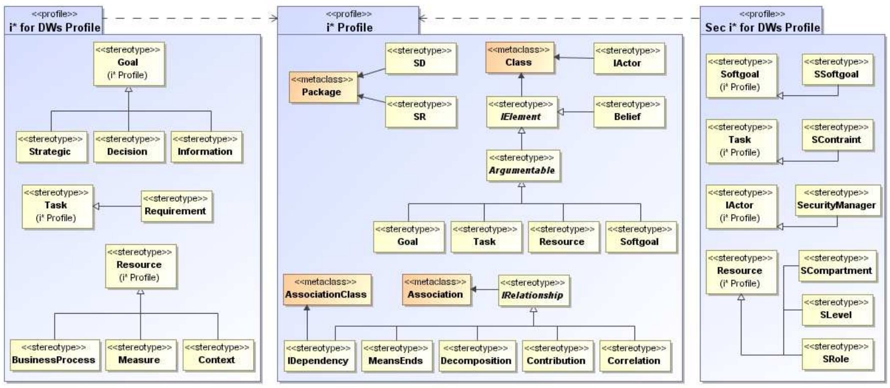

Fig. 1. UML profiles for i* modeling in the DW domain.

This UML profile for i* has been implemented in an Eclipse-based tool that provides support for our MDA-based approach for the development of DWs (see Fig. 2). By using this tool we can define a CIM and apply a set of QVT transformations to obtain the corresponding PIM.

4 Conclusions

DW projects overlook an explicit requirement analysis phase when MD models are defined. Therefore, DW fails to give the adequate support to decision making.

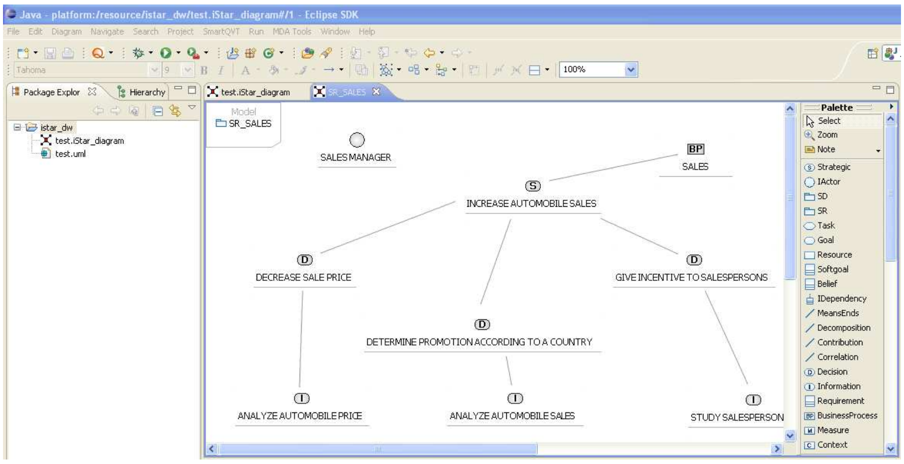

Fig. 2. Snapshot from our tool for i* modeling in the DW domain.

Our research aims to use the *i** framework within our MDA approach for the development of DW in order to avoid this important drawback.

5 Ongoing and future work

Our immediate future work comprises the improvement of our proposal by considering other further issues of GORE (e.g., more complex mechanisms for reasoning about goals or prioritization of goals). Furthermore, other QoS issues (apart from security) should be considered.

References

1. Mazón J.N., Trujillo J., Serrano M., Piattini M.: Designing data warehouses: from business requirement analysis to multidimensional modeling. REBNITA 2005
2. Mazón J.N., Pardillo J., Trujillo J.: A Model-Driven Goal-Oriented Engineering Approach for Data Warehouses. RIGiM 2007. LNCS 4802, pp. 255–264
3. Mazón J.N., Trujillo J., Lechtenbörger J.: Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. Data Knowl. Eng. **63**(3) (2007) 725–751
4. Mazón, J.N., Trujillo, J.: An MDA approach for the development of data warehouses. Decision Support Systems (in press, **doi:10.1016/j.dss.2006.12.003**)
5. Soler E., Trujillo J., Fernández-Medina E., Piattini M.: A Framework for the Development of Secure DWs based on MDA and QVT. ARES 2007, pp. 294-300
6. Soler E., Stefanov V., Mazón J.N., Trujillo J., Fernández-Medina E., Piattini M.: Towards Comprehensive Requirement Analysis for Data Warehouses: Considering Security Requirements. ARES 2008

Tropos[1] at the Age of Eight: On-going Research at FBK, UniTN and UT

Paolo Giorgini[1], John Mylopoulos[1,2], Loris Penserini[3], Anna Perini[3], Angelo Susi[3]

[1] University of Trento, Trento, Italy
{paolo.giorgini,jm}@disi.unitn.it

[2] University of Toronto, Toronto, Canada

[3] Fondazione Bruno Kessler, Povo, Trento, Italy
{perini,penserini,susi}@fbk.eu

Abstract. The Tropos project was launched in the Spring of 2000. Its aim has been to establish a methodology for building agent-oriented software systems. The methodology that has emerged is founded on the *i** modelling framework to support four phases of software development: early and late requirements, as well as architectural and detailed design. The purpose of this report is to offer an overview of on-going work on the project at Fondazione Bruno Kessler (FBK), the University of Trento (UniTN) and the University of Toronto (UT).

1 Introduction

The Tropos project was launched in the Spring of 2000 at the University of Toronto (hereafter UT), the University of Trento (UniTN) and the Fondazione Bruno Kessler (FBK) known as IRST back in those days. Its aim has been to establish a methodology for building agent-oriented software systems. The methodology that has emerged is founded on the *i** modelling framework to support four phases of software development: early and late requirements, as well as architectural and detailed design. Its initial contributors (… founding fathers and mothers) included at UT Jaelson Castro[2], Manuel Kolp and John Mylopoulos; at UniTN/FBK Paolo Bresciani, Paolo Giorgini, Fausto Giunchiglia, Anna Perini, Marco Pistore and Paolo Traverso.

The first major milestone of the project was to lay out a methodology for building agent-oriented software. This milestone was achieved within the first year with the help of two case studies, leading to the most cited publications of the Tropos project [Castro02], [Bresciani04]. The next milestones focused on developing formal reasoning techniques to support the Tropos methodology. One thread of research

[1] "tropos", in Greek τροπος, is an ancient word. The very first words in Homer's Odyssey are "Ανδρα μοι εννεπε μουσα πολυτροπον …" – "Muse, help me tell the story of the man of many ways" (…"the man" is Ulysses).

[2] On leave from the Federal University of Pernambuco (Brazil).

aimed to develop a tool that would enable verification of Tropos models through model checking. This work led to the Formal Tropos specification language and the T-tool [Fuxman04]. In parallel, the UniTN team developed formal reasoning techniques for goal (and softgoal) models, along with the G-tool that implemented these techniques [Giorgini03], [Sebastiani04]. Publications on and running versions of these tools can be found at http://www.troposproject.org/.

The one-and-only purpose of this report is to offer a guide to some of the research threads at FBK, UniTN and UT that followed the original milestones outlined above. There have been other significant threads of research at other universities, but they fall outside the scope of this report.

The rest of the document is structured as follows. Sections 2, 3 and 4 overview respectively on-going but reasonably mature research threads at the three institutions. Section 5 concludes and offers some hints on future directions for the project.

2 FBK

Research on Tropos is conducted within the Software Engineering (SE) unit at FBK.[3] More generally, the research carried out by the SE unit addresses the development of complex software systems, having large size, operating in a distributed environment, exhibiting autonomic behaviours, expected to fulfil high quality standards, and realized using innovative technologies and approaches. The SE unit focuses on two strategic areas of software development, namely *Requirements engineering* and *Code analysis and testing.* In the first area, the scientific challenges deal with the explicit representation of requirements for autonomic behaviours (e.g., those of self-adaptive systems), of the normative constraints and of the flows. Here, agent-oriented approaches seem particularly promising. In the area of software testing, the challenge is to automate the generation of the test cases and their execution.

Research results contributed to the extension of the agent-oriented modelling tool TAOM4E (http://sra.itc.it/tools/taom4e/). Advanced functionalities include test case derivation and execution (see the eCAT framework) and automated BDI code derivation [Morandini07a].

Normative i* modelling. A distinguishing feature of socio-technical organisations over ad hoc groups of interacting individuals is the existence of norms. Various types of norms exist in the real world, but those that are more relevant at requirements time are behavioural norms that impose actions to be performed, goals to be achieved, resources to be delivered or principles to be respected. We propose to use a goal-oriented approach, based on i*, for modelling such kind of norms and introduce a limited set of additional abstractions and diagrams for modelling norms. More specifically, our idea is to model contextually and homogeneously, but separately, the normative context of a domain and its stakeholders with their intentionality [Siena07a]. A recent application of normative i* modelling to a food-chain scenario gave promising results towards proving its effectiveness [Siena07b].

[3] More details on research activities, projects and collaboration at http://se.fbk.eu.

High-Variability Design for Software Agents: Extending Tropos. High-variability design has been proposed to generate generic software solutions and to support self-configuration in autonomic software. Complementing research developed in UT, we focused on designing software agents [Penserini07, Morandini07b]. We extended the Tropos methodology, enhancing its ability to support high variability design, through the explicit modelling of alternatives, by adopting an extended notion of agent capability. A tool-supported process founded on the Model-Driven Architecture (MDA) framework and standards, supports goal-oriented analysis of requirements of self-configuring software and the derivation of BDI agent code which realizes them.

Goal-Oriented Testing. Goal-oriented specifications are particularly appropriate for distributed, concurrent systems, which communicate by means of messages and have been designed to behave autonomously (like agents). Testing of these kinds of systems remains an unexplored area, of great importance for their adoption in SE practice. We are studying testing techniques for goal-oriented systems. In particular we address the problem of automating test case generation as well as their execution.

Main results of this research include a goal-oriented testing methodology that complements Tropos analysis and design [Nguyen07a]. Test cases are derived directly from the goal-oriented specification of the system under test; a novel testing framework, called eCAT[4], which integrates manual and automated test cases generation techniques, so that it can generate and evolve test cases automatically, and run them continuously [Nguyen07b,c].

3 UniTN

At UniTN, research on Tropos is done within the Software Engineering and Formal Methods research group[5]. Three are the most relevant research activities: Security Modelling and Analysis, Goal-based Risk Analysis and Automated Design.

Security Modelling and Analysis

Managing high-level user requirements is a key issue for the successful and cost effective development of IT systems, but managing security requirements is almost completely ignored. We propose a requirements engineering methodology, Secure Tropos [Giorigni05a, Giorgini05b, Giorgini06c], to support IT designers in the capture of high-level security and trust requirements and their implementation. In particular, we have extended and refined the i*/Tropos methodology with basic primitives suitable for capturing security aspects of organizations. In particular, we introduced primitives for modelling entitlements of actors and making explicit their capabilities. Moreover, the notions of delegation and (dis)trust are used to model the transfer of entitlements and responsibilities between actors, and the expectation of an actor about the behaviour of other actors. Once the security and trust model has been captured, our purpose is to automatically verify security and trust requirements [Giorgini06a]. To provide automated reasoning support with a quick prototyping lifecycle we use Datalog. In this setting, each concept/relation occurring in graphical

[4] See http://sra.fbk.eu/people/cunduy/ecat/. eCAT has been integrated with TAOM4E.

[5] More details about the group can be found at http://dit.unitn.it/research/rp.xml?rpid=3

diagrams is represented as a Datalog predicate. The collection of these predicates represents the extensional description of the system. The formal framework is comprised of rules that define the semantics of primitive concepts and are used to make explicit the information that are necessary for the verification of security requirements. Such information is then used to define constraints whose violation points out inconsistencies in the system [Giorgini06b]. These constraints are essentially in form of patterns that represent system vulnerabilities.

Goal-based risk analysis

Goal models have been proved to be useful to model and analyze stakeholder objectives to elicit requirements of information systems. However, a goal model also needs to anticipate uncertain circumstance that can affect the achievement of stakeholder objectives. Therefore, Goal-Risk Framework [Asnar06a, Asnar06b] are introduced extending Tropos goal model with 3 layers of conceptual analysis: goal, event, and treatment layer. Goal layer is meant to analyze strategic interest of stakeholders, event layer analyzes the impact of uncertain events to the goal layer (i.e., a risk is uncertain event with negative impact), and treatment layer analyzes a course of actions that are meant to treat uncertain events (e.g., mitigate risks). Using this framework, an analyst can model and reason about IS requirements that have encompassed risks and their mitigation besides stakeholder objectives [Asnar07a]. The framework has been implemented and enhanced for analyzing safety critical systems (e.g., Air Traffic Management [Asnar07b]) and goal deliberation process of autonomous agent systems [Asnar07c].

Automated Design

The focus of the work is on exploring the space of alternative choices during requirements analysis and design of information systems. Namely, the problem is in how to find an optimal/good-enough set of delegations and assignments of goals (to be fulfilled by a system) to the system actors. The approach taken consists of two parts: generating alternative design structures with the help of AI (Artificial Intelligence) planning techniques, and evaluating the generated alternatives with respect to the local strategies of system actors [Bryl06b]. The problem of constructing a design structure that guarantees the fulfilment of system goals is framed and formalized as a planning problem. An off-the-shelf planning tool is used to generate an alternative design structure, which is then evaluated, amended and finally adopted [Bryl06a]. Evaluation schema is inspired by game-theoretic ideas; basically, system actors are seen as self-interested and rational players that are trying to maximize their local utilities, i.e. the benefit they could gain from the adopted alternative. The prototype tool (P-Tool) implements the approach, and is supposed to support the designer in selecting good-enough alternative design structures. The described planning-and-evaluation approach has a number of applications, e.g. it was applied to the problem of self-configuring systems [Bryl06c], which change their structure in response to internal or/and environmental changes.

4 UT

We present three mature research threads.

Variability in Goal Models. Goal models describe a set of alternative ways for fulfilling a requirement. We are interested here in making the design of such models more systematic by identifying the origins of variability. For example, variability may arise from a choice of the agent assigned to fulfil a goal, the medium to be used, or the time of the fulfilment [Liaskos06]. Once variability is identified, it can be used to support personalization [Liaskos05].

Goal-oriented design. Goal-oriented design is characterized by an explicit consideration of design alternatives, and a selection based on non-functional requirements (a.k.a. softgoals). However, the space of design alternatives is based partly of the solution space for the problem-at-hand (dealt with by goal models) and partly on the nature of the artifact-to-be. We have been exploring two threads of research on this.

Lei Jiang, Alex Borgida and Thodoros Topaloglou have been exploring goal-oriented database design. Here, the idea is to start from stakeholder goals, identify plans for fulfilling them, pinpoint information needs for these plans, and design a database on that basis. Variability is an important parameter here: there are many possible designs for a given set of stakeholder informational goals. So are data quality considerations that can make-or-break an information system [Jiang07].

Along a different path, Alexei Lapouchnian is developing a methodology for design that starts from stakeholder requirements expressed as goal models and refines them to generate business process designs [Lapouchnian07]. The proposed methodology exploits the variability inherent in goal models to generate business process designs that that can fulfil root level goals in multiple ways.

5 The Future

Future trends for the Tropos project are largely dictated by the emerging focus on run-time software behaviour. This trend is manifested under different buzzwords: autonomic, adaptive, dynamic, etc. Independently of the buzzword, the theme is the same: software in the future will have to self-manage itself and adapt to changes in its environment through monitoring, diagnosis and compensation components.

The other major trend influencing Tropos is the broadened scope of modelling, analysis and design techniques to support not just software systems through their lifecycle, but also the organizational environment within which they live and operate.

Acknowledgements. The Tropos project benefited tremendously from generous funding provided by the Provincial Government of Trentino through the STAMPS (a.k.a SMTPPS) project and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[Asnar06a] Y. Asnar and P. Giorgini. Ensuring Dependability in Socio-Technical System by

Risk Analysis. In *Proceedings of the 6th European Dependable Computing Conference*, 2006.

[Asnar06b] Y. Asnar and P. Giorgini. Modelling Risk and Identifying Countermeasures in Organizations. In *Proceedings of 1st International Workshop on Critical Information Infrastructures Security* (CRITIS '06), 2006.

[Asnar07a] Y. Asnar, P. Giorgini, F. Massacci, and N. Zannone, From Trust to Dependability through Risk Analysis. In *Proceedings of the Second International Conference on Availability, Reliability and Security* (AReS 2007), Vienna, April 10th-13th, 2007

[Asnar07b] Y. Asnar, R. Bonato, P. Giorgini, F. Massacci, V. Meduri, C. Riccucci, and A. Saidane, Secure and Dependable Patterns in Organizations: An Empirical Approach. In *Proceedings of the 15th IEEE International Requirements Engineering Conference* (RE '07), India Habitat Center, New Delhi, India, October 15-19th, 2007

[Asnar07c] Y. Asnar, P. Giorgini, and N. Zannone, Reasoning about Risk in Agents Deliberation Process: a Jadex Implementation. In *Proceeding of 8th International Workshop on Agent Oriented Software Engineering* (AOSE '07), Honolulu, Hawai'i, May 14th, 2007

[Bresciani04] Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., and Perini, A., "Tropos: An agent-oriented software development methodology", *Journal of Autonomous agents and Multiagent Systems* (JAAMAS), Kluwer, 203–236, 2004.

[Bryl06a] Bryl, V., Massacci, F., Mylopoulos, J., Zannone, N.: Designing security requirements models through planning. In: *CAiSE'06*, Sprinter (2006) 33–47.

[Bryl06b] Bryl, V., Giorgini, P.: Designing cooperative IS: Exploring and evaluating alternatives. In: *CoopIS*, Springer (2006) 533-550.

[Bryl06c] Bryl V., Giorgini P.: Self-configuring socio-technical systems: Redesign at runtime. *ITSSA*, 2(1):31-40, 2006.

[Castro02] Castro, J., Kolp, M., and Mylopoulos, M., "Towards Requirements-Driven Information Systems Engineering: The Tropos Project", *Information Systems (27),* Elsevier, Amsterdam, The Netherlands, 365–389, 2002.

[Fuxman04] Fuxman, A., Liu, L., Mylopoulos, J., Pistore, M., , Roveri, M., Traverso, P., "Specifying and Analyzing Early Requirements in Tropos", *Requirements Engineering Journal 9(2)*, Springer-Verlag, 132-150, July 2004.

[Giorgini03] Giorgini, P., Mylopoulos, J., Nicchiarelli, E., Sebastiani, R., "Formal Reasoning Techniques for Goal Models", *Journal of Data Semantics 1(1)*, Springer LNCS 2800, 1-20, 2003.

[Giorgina05a] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modeling Security Requirements Through Ownership, Permission and Delegation. *In Proc. Of RE'05*, pages 167–176. IEEE Press, 2005.

[Giorgini05b] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modelling Social and Individual Trust in Requirements Engineering Methodologies. In *Proc. of iTrust'05*, volume 3477 of LNCS, pages 161–176. Springer, 2005.

[Giorgini06a] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Detecting Conflicts of Interest. In Proc. of RE'06, pages 315–318. IEEE Press, 2006.

[Giorgini06b] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Requirements Engineering for Trust Management: Model, Methodology, and Reasoning. Int. *J. of Inform. Sec*., 5(4):257–274, 2006.

[Giorgini06c] P. Giorgini, H. Mouratidis, and N. Zannone. Modelling Security and Trust with Secure Tropos. In *Integrating Security and Software Engineering: Advances and Future Vision*, chapter VIII. IDEA Group, 2006.

[Jiang07] Jiang, L., Topaloglou, T., Borgida, A., Mylopoulos, J., "Goal-Oriented Conceptual Database Design", 16th Int. IEEE Conf. on Requirements Engineering (RE'07), Delhi, October 2007.

[Lapouchnian07] Lapouchnian, A., Yu, Y., Mylopoulos, J., "Requirements-Driven Design and Configuration Management of Business Processes", 5th International Conference on Business Process Management (BPM'07), Brisbane, September 2007.

[Liaskos05] Liaskos, S., Lapouchnian, A., Wang, Y. Yu, Y. and Easterbrook, S., "Configuring Common Personal Software: A Requirements-Driven Approach", 13th IEEE International Conference on Requirements Engineering, August 2005, Paris.

[Liaskos06] Liaskos, S., Lapouchnian, A., Yu, Y., Yu, E., Mylopoulos, J., "On Goal-Based Variability Acquisition and Analysis", 14th International IEEE Conference on Requirements Engineering, September 2006, Minneapolis/St. Paul.

[Morandini07a] M. Morandini, D. C. Nguyen, A. Perini, A. Siena and A. Susi, *Tool-supported Development with Tropos: the Conference Management System Case Study*, AOSE-07 Postproceedings LNCS 4951.

[Morandini07b] M. Morandini, L. Penserini, A. Perini, and A. Susi, *Refining Goal Models by Evaluating System Behaviour*, AOSE-07 Postproceedings LNCS 4951.

[Nguyen07a] Nguyen C. D., Perini A., Tonella P. "A Goal-Oriented Software Testing Methodology", AOSE-07 Postproceedings LNCS vol. 4951.

[Nguyen07b] Nguyen C. D., Perini A., Tonella P., "Automated Continuous Testing of Multi-Agent Systems", Fifth European Workshop on Multi-Agent Systems (EUMAS), December 2007.

[Nguyen07c] Nguyen C. D., Perini A., Tonella P., "Ontology-based Test Generation for Multi-Agent Systems", short paper at AAMAS 2008.

[Penserini07] Penserini, L., Perini, A., Susi, A., Mylopoulos, J., "High-Variability Design for Software Agents: Extending Tropos", *ACM Transactions on Autonomous and Adaptive Systems (TAAS) 2(4),* ACM Press, November 2007.

[Sebastiani04] Sebastiani, R., Giorgini, P., Mylopoulos, J., "Simple and Minimal-Cost Satisfiability for Goal Models", Sixteenth International Conference on Advanced Information Systems Engineering (CAiSE'04), Riga, Springer-Verlag LNCS 2003, June 2004.

[Siena07a] Siena, A., "Engineering Normative Requirements.", In Proceedings of the First International Conference on Research challenges in Information Science (RCIS). Ouarzazate, Morocco. 22-26 April, 2007.

[Siena07b] Siena, A., Maiden, N., Lockerbie, J., Karlsen, K., Perini, A., and Susi, A., "Exploring the Effectiveness of Normative i* Modelling: Results from a Case Study on Food Chain Traceability", Tech. Rep. FBK-IRST1207.