Mastering concept exploration in large industrial research projects

Simona Citrigno[1], Angelo Furfaro[2], Teresa Gallo[2], Alfredo Garro[2], Sabrina Graziano[3], Domenico Saccà[2]

[1]Centro di Competenza ICT-SUD, Piazza Vermicelli - 87036 Rende (CS), Italy
[2]Department of Informatics, Modeling, Electronics, and Systems Engineering (DIMES), University of Calabria, Via P. Bucci 41C, 87036, Rende (CS), Italy
[3]Open Knowledge Technologies s.r.l. Piazza Vermicelli - 87036 Rende (CS), Italy
simona.citrigno@cc-ict-sud.it,{angelo.furfaro, teresa.gallo, alfredo.garro, domenico.sacca}@dimes.unical.it, sabrina.graziano@okt-srl.com

Abstract. Concept exploration, requirement specification and analysis are activities of utmost importance in the development life-cycle of complex systems especially when the business domain is unfamiliar to the designers and many partners are jointly involved in a project, as in industrial research project funded by government institutions. A critical role is then played by the characteristics of the methods, notations and tools adopted for carrying out these activities. This paper proposes a goal-oriented methodology for addressing both early and late requirement engineering phases which is lean, easy to master and apply and, at the same time, clear and rigorous. The methodology is not the result of a pure speculative process but has been concretely shaped in the context of a real large research project where it is showing its effectiveness in fostering cooperation and clean project evolution.

I - Introduction

Large projects, such as those funded by government institutions, usually involve many partners that need to work concurrently and to cooperate among them to achieve project goals in an orderly and timely way. The organization and management of such projects are complex activities that are critical for a successful achievement of project results. Many are the risk sources: coexistence of different interpretations of project goals and requirements, conflicting specifications, late discovery of redundancy, fragmentation of efforts, weak focus on objectives, partner coordination issues, and work-product integration. In such complex and heterogeneous settings, a crucial role is played by the adopted methodologies, in particular during the requirements engineering phases (Sommerville, 2011; Yu, 1997). Most of the available modeling techniques are mainly focused on the identification, analysis and specification of the requirements that must be satisfied by the functionalities that are going to be realized. Often, little or no attention is paid to the motivations that lead stakeholders to demand for such features and this may result in late discovery of incompleteness and inconsistencies due to conflicting goals and needs. With the aim to overcome these issues, in the last few years, several goal-oriented methodologies have been proposed (van Lamsweerde and Letier, 2004; Quartel, Engelsman, Jonkers and Sinderen, 2009; Rolland, Horkoff, Yu, Salinesi and Castro, 2012) and limitations of the related existing tools have been addressed (Amyot et. all, 2012).

However, few methodologies support both early and late requirements engineering phases, most of them adopt proprietary or very formal notations and languages (Z.151, 2012) and are

difficult to integrate with the methodologies used in the downstream development process. In large research industrial projects with many partners and stakeholders, only one or few of them are industrial enterprises playing the role of target user whose requirements have to be addressed, while many of the partners are researchers and/or experts in specific areas, although precious for the reference application domain. In this case, the adoption of one of those methodologies and related tools do not facilitate communication and convergence of intents in mastering concept exploration. Thus, the project will end with important research results for every research partner in its specific area of interest, but with poor industrial applicable results for the target users. Incomplete or complex language notation is often the origin of an insufficient domain concept exploration and then of an unclear and inconsistent project requirement elicitation phase (Engelsman and Wieringa, 2012). As a consequence, misunderstandings in the way of achieving goals in the specific industrial contexts of the target users are indeed inevitable (Ali, Dalpiaz and Giorgini, 2013).

This paper proposes a novel, lean and easy to adopt approach that aims at addressing the above introduced issues in large industrial research projects. Specifically the paper presents a Goal-Oriented Requirements Methodology (GOReM) for mastering concept exploration, which seamlessly supports all the stages of requirements engineering. GOReM is structured in three main phases (Context modeling, Scenario Modeling and Use Case modeling) that lead to requirements specification analysis starting from stakeholders' goals. Key aspects of GOReM are the following: (i) clear understanding of the context where the system under definition is going to operate in terms of governing rules, stakeholders and their goals; (ii) modeling of business scenarios and analysis of the strengths, weaknesses, opportunities and threats of each business scenario; (iii) use case and process-based modeling of application scenarios. As a consequence, GOReM enables global/analytical views of the system and of its context at different abstraction levels. This provides a solid ground for partners' cooperation, efforts harmonization and outcomes validation. Moreover, GOReM is based on a lean process and on an UML-based notations (Object Management Group) so as to have a smooth learning curve and ease the integration and reuse of the released work-products in the subsequent design and development phases. It is worth to note that GOReM is not the result of a pure speculative process, but it derives from the needs and criticalities arising in the context of real projects. Indeed, the definition of GOReM has been carried out in a real large research project - DICET-INMOTO - funded by the Italian Ministry of Education, University and Research (MIUR), where it is showed its effectiveness in fostering cooperation and clean project evolution.

The remaining of the paper is structured as follows: Section II presents some proposals that share with GOReM aims and tools; Section III illustrates GOReM both in terms of reference process and meta-model; Section IV presents an exploitation of GOReM in a real case study with reference to the DICET-INMOTO project; Section V draws the conclusions and discusses some directions for future works.

II - Related Work

Goal-orientation in requirements engineering has gained in popularity during recent years due to the limitations of traditional approaches in dealing with complex systems or complex organizations (Rolland, Horkoff, Yu, Salinesi and Castro, 2012; Ramirez, Vergne, Morandini, Sabatucci, Perini and Susi, 2012) . In particular, Goal-oriented requirements engineering (GORE) originates from the needs to model and to understand system requirements from high-level aspects of the domain of interest and to cope with non-functional requirements. Various GORE approaches have been proposed, applied, compared and reviewed in literature (Amyot et. all,

2012; Ramirez, Vergne, Morandini, Sabatucci, Perini and Susi, 2012; Sutcliffe and Sawyer, 2013; van Lamsweerde, 2001; UCMNav).

The NFR framework (Chung, Nixon, Yu, and Mylopoulos, 2000) is a process-oriented approach that focuses on the modeling of non-functional requirements (called *softgoals*) and on the identification of the influences (positive or negative) among them. Softgoal refinements and influences are represented in a *softgoal interdependency graph,* which allows evaluating the contributions of more specific goals with respect to higher level ones and to identifies and evaluate different alternatives.

The *i** framework (Yu, 1997) is a goal-oriented modeling technique which is based on an explicit representation of goals and actors and has been adopted in many requirement engineering contexts. The *i** framework defines two types of model: the Strategic Dependency model and the Strategic Rationale model. The first one describes the dependencies among actors involved in a given context where they depend on each other for goals to be achieved, tasks to be performed, and resources to be made available. The Strategic Rationale model identifies stakeholder interests and concerns, and shows how they can be dealt with by various configurations of systems and environments. The *i** framework is mainly concerned with the early-phase requirements engineering.

KAOS (van Lamsweerde and Letier, 2004) is a method for deriving operational requirements of a system starting from its goals. The notion of goal, intended as a "prescriptive statement of intent that the system should satisfy through cooperation of its agents" (van Lamsweerde, 2000), is the main concept underlying KAOS where an agent is any entity that may influence the fulfillment of a goal. KAOS supports the definition of goals at different level of abstraction by introducing suitable refinement relations among goals.

ARMOR (Quartel, Engelsman, Jonkers, and van Sinderen, 2009) is another goal-oriented modelling language which lays its basis both on *i** and on KAOS methods trying to overcome their respective limitations. In particular, ARMOR adds support for modeling stakeholders' domain during the early requirements specification and it adopts UML use cases for the elicitation of system specific requirements.

GRL (Goal-oriented Requirement Language) is part of the URN (User Requirements Notation) (Z.151, 2012), it is a standard notation for goal modeling and it is a simplified version of i*. GRL is supported by the jUCMNav (jUCMNav), a free Eclipse graphical editor, which is an analysis and transformation tool with limitations and possible improvements as discussed in (Amyot et. all, 2012).

In this context, GOReM aims to capitalize the above mentioned experiences by also resorting to concepts, abstractions, and methods coming from the AOSE (Agent-Oriented Software Engineering) domain (Lind, 2001) and, specifically, derived from both the Tropos (Bresciani, Perini, Giorgini, Giunchiglia and Mylopoulos, 2004) and Gaia (Zambonelli, Jennings and Wooldridge, 2003) methodologies. Moreover, it exploits an UML-based graphical notation, so as to produce a documentation which is both expressive and easy to read, embracing the thesis in (Caire, Genon, Heymans and Moody, 2013) where visual notation enhances user comprehension of requirements (i.e. cooperation and sharing) and it allows to move towards mastering concept exploration, at least for known unknowns as defined in (Sutcliffe and Sawyer, 2013) (i.e., expressing what is behind stakeholders' expressions as "obviously" and "of course") which is a typical situation in large industrial research projects.

III - The GOReM Methodology

In this section the GOReM Methodology is described by first explaining a reference meta-model that highlights the performed activities and the relationships among them. Then the GOReM reference process together with its work-products are presented.

The GOReM Methodology consists in three specific phases of modeling activities:

1) *Context Modeling*, where the system stakeholders are identified along with their goals as well as the relationships and dependencies between stakeholders and goals; moreover, the rules and norms that govern the context are specified.

2) *Scenario Modeling*, in which different business scenarios are specified in terms of roles that are played by the stakeholders involved in the scenario, their specific goals, and the rules and norms that govern the business scenario. A SWOT Analysis (Strengths, Weaknesses, Opportunities and Threats) is also performed (Hill and Westbrook, 1997) with the aim to guide future work decisions.

3) *Use Case Modeling*, in which application scenarios are introduced in order to fully specify the functionalities which should be provided by each business scenario resulting from the previous phase.

Figure 1 shows an UML based representation of the reference meta-model and all the involved entities and the associations among them. The three different colors used in the schema are useful to easily identify the corresponding three GOReM phases.

A detail of each single phase is described as follows. The Context Modeling phase aims at delimitate the project scope within which the requirements should find precise boundaries. Context Modeling can be related to a large partnership which would like either to participate to a public call for proposal or to realize the starting phase of an industrial research project. This crucial activity has to be carried out by a (small set of) designer(s) who has to identify main elements useful to fix a clear, unique, specific and shared project scope. In the Context Modeling phase, the set of stakeholders (i.e., *Stakeholder* in Figure 1) and whichever their specialization (i.e., *specialize* association) should be identified. Each stakeholder is in turn interested in pursuing a set of softgoals (i.e., *Softgoal*) which is a common concept in the goal-oriented community for denoting generic goals and often non-functional requirements. The GOReM methodology usually identifies the following relationships among softgoals: *decompose*, *contribute*, *hinder* and *specialize*, but, if needed, other association stereotypes can be also added. Moreover, during this phase, it is important to clearly identify and carefully take into account the set of *Rule and Regulations,* which govern the specific context, as they (and their possible changes) can influence the achievement of stakeholder's goals.
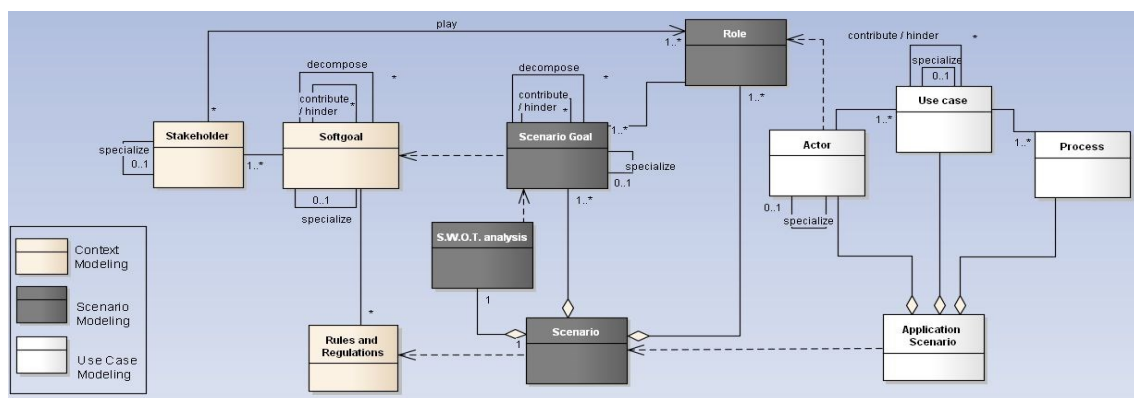

Figure 1: GOReM: The reference meta-model

The Scenario Modeling phase aims at specializing each context in many specific scenarios (i.e., *Scenario*). The decision process able to establish such set of scenarios covering the modeled context is not a formal process, but it is the results of several interactions among project partners based on the results of the Context Modeling phase.

Each scenario should be modeled by a (possible small set of) designer(s) specialized in the specific scenario domain. Starting from the stakeholders identified in the previous Context Modeling phase, a set of specific stakeholder roles (i.e., *Role*) is identified for each scenario. For each stakeholder playing a role (i.e., *play*) in the scenario that is going to be modeled, a set of specific scenario goals (i.e., Scenario Goal) is established as well as the association among them (i.e., *contribute, hinder, include, extend, specialize*). These scenario goals depend on one or more softgoals (see the dependency association in Figure 1) identified in the Context Modeling phase. Furthermore, it is important to define the subset of previously identified rules and regulations that govern each scenario (see the dependency association in Figure1 from *Scenario* to *Rules and Regulations*). For each scenario a *SWOT analysis* is also performed based on the scenario goal (see the dependency association in Figure 1 from *SWOT analysis* to *Scenario Goal*). This analysis may influence the decisions about "if and how" to proceed, for each scenario, with the subsequent Use Case Modeling phase.

In the *Use Case Modeling* phase, for each modeled scenario, an application scenario (i.e., *Application Scenario*) can be set. Note that it is possible to decide to not specify any application scenario for one or more already modeled scenarios (see the dependency association from *Application Scenario* to *Scenario* in Figure 1); this happens often in large industrial research projects where only a small subset of scenarios is considered for developing some prototypes aimed at demonstrating the value of specific project results. Thus, the Use Case Modeling phase guides the project in the direction of what has to be considered in the subsequent development phase. The decision process behind this phase is not a formal process, but it is the result of a strong interaction among project partners based on the results of the Scenario Modeling phase.

Each application scenario is defined in terms of its actors, use cases and processes (i.e., *Actor, Use Case, Process*). Actors and whichever of their specialization (i.e., *specialize* association) refer to roles identified in the previous phase (see the dependency association from *Actor* to *Role)*. Among use cases GOReM considers *contribute, hinder* and *specialize* associations, but also other association stereotypes can be added if needed.

Figure 2, shows a BPMN diagram (Business Process Model and Notation, 2011) that illustrates the GOReM reference process together with its main work-products. Modelers having different skills perform the three phases. As an example, the *Scenario Modeling* phase concerns the parallel definition of different scenario models that can be performed by different scenario experts. A scenario model can be considered ready for the *Use Case Modeling* phase independently by other scenarios. For each business scenario, several use cases can be introduced in the *Use Case Modeling* phase in order to define the application scenarios that are able to define a specific set of functionalities. In this phase, each *application scenario model* can develop itself independently from the others and thus can be released or can require going back to the Scenario Modeling phase, as further specifications are needed. The decision to go back or to proceed to the next process phase is a critical decision, which is taken by the teams working on the specific business and/or application scenario.

The *Context Model* aims at clearly representing the reference domain for the project under consideration. The work-products of this phase are: a *Stakeholder Diagram*, which shows a hierarchical specification of all the stakeholders involved in the specific context, each of which is in turn characterized by a set of *Softgoals* they intend to pursue; a *Softgoal Dependency*

Diagram, which shows the relationships between the stakeholders and the softgoals, as well as the relationships among softgoals (contributes, hinders, includes, extends, specializes); a *Rule and Regulations* table summarizing and shortly describing the rules and regulations governing the Context.

Starting from the results of the *Context Modeling* phase, the *Scenario Modeling* phase aims at identifying and modeling various business *scenarios* of interest for the project.
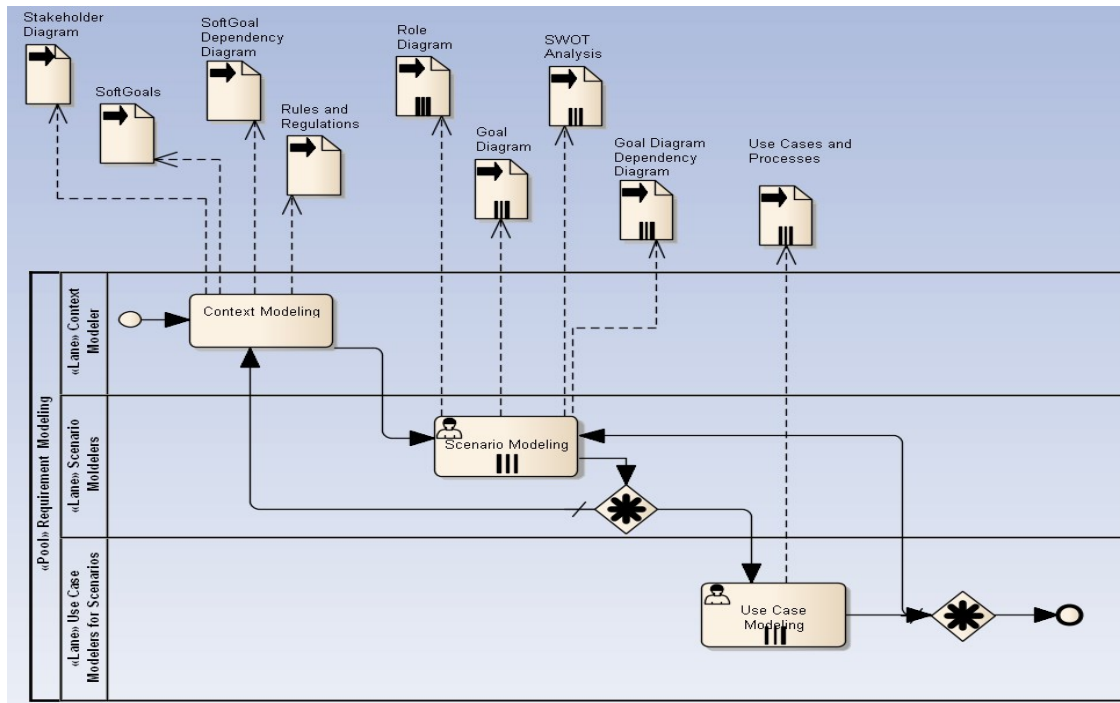

Figure 2: GOReM: The reference process

A *Scenario Model* consists in the following work-products: a *Role Diagram,* that highlights roles played by the stakeholders involved in the scenario; a *Goal Diagram,* that highlights the specific goals assigned to the identified roles as well as the relationships among them*;* a table describing such goals; a table for summarizing and shortly describing the specific rules and regulations that govern the scenario (*Scenario Rules and Regulations*); a SWOT analysis, that produces a 2x2 matrix that highlights internal Strengths, internal Weaknesses, external Opportunities, and external Threats (Hill and Westbrook, 1997). From the Scenario Modeling phase, for each specific scenario, it is always possible to go back to the Context Modeling phase to better define or elicit some context elements of interest for the scenario (i.e., stakeholders, softgoals, rules and regulations). When a specific scenario has been modeled in the proper way, then the related *Use Case Modeling* phase can begin.

The *Use Case Modeling* phase aims at specifying the application scenarios, which will constitute (a part of) the system to be implemented. Each application scenario is characterized by functionalities that are modeled by UML-based *Use Cases and Processes*. From this phase it is always possible to go back to the *Scenario Modeling* phase to better define or elicit some scenario elements (i.e., goals, roles, scenario rules and regulations, SWOT analysis). The result of this phase is a collection of use case specifications grouped by scenario and constitute an application scenario. Each use case is specified in terms of its objectives, primary and

secondary involved actors, flow of events, and, if necessary, secondary use cases related to it through the inclusion/extension relationships.

IV – Exploiting GOReM in a real project

The GOReM Methodology has been used in some recent research projects. As a case study, it is showed the exploitation of GOReM in an ongoing large industrial research project named "DICET- INMOTO" - ORganization of Cultural HEritage for Smart Tourism and Real-time Accessibility (OR.C.HE.S.T.R.A.), funded by the Italian Ministry of Education, University and Research (MIUR) within the PON Project - Research and Competitiveness 2007-2013.

This Project involves 13 partners including universities, public research centers, small and medium-sized enterprises (SMEs), and large enterprises. The Project, which started on November 2012 and whose conclusion is planned on May 2015, aims at defining and implementing models, processes and tools for sustainable development of an intelligent territory through the exploitation of its cultural heritage and environmental resources and the promotion and marketing of its tourist offer. In particular, the INMOTO Project deals with technologies and innovative methods for goods and cultural contents exploitation and for the promotion of the linked territories for a sustainable tourism development.

GOReM has been used for the Context Modeling, Scenario Modeling and Use Case Modeling in the INMOTO Project. In particular, in the following sections an example of exploitation of the Methodology is outlined. An UML-based notation is used to represent many of the previously explained work-products. The example shows the *Tourism in mobility* Context Model, where it is possible to view stakeholders involved in the Tourism domain together with their hierarchical decomposition. For each stakeholder it is possible to view their softgoals and the relationships among softgoals. This is represented in the Softgoal Dependency Diagram work-product in Figure 3 where stakeholders are represented by using the standard UML actor symbol but with a yellow-filled head and softgoals are represented by rectangles.
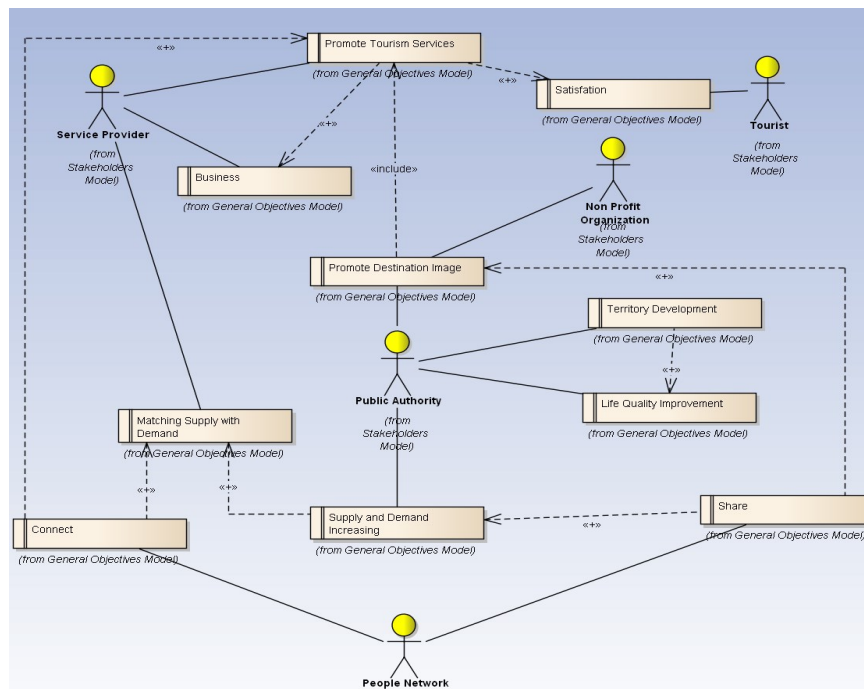

Figure 3: "Tourism in mobility" Context Model: Softgoals Dependency Diagram

In the Project, starting from the Context Model, the following scenarios have been identified: "Business to network for Tourism in Mobility", "Tourism in Mobility Brokerage", "Tourism in Mobility Knowledge Exploitation". As an example of Scenario Modeling, the "Tourism in Mobility Brokerage" scenario is investigated, where stakeholders and their roles played in the scenario are highlighted.

Brokerage activity in Tourism in Mobility fosters mutual collaboration, services and level of quality monitoring through coordinating policies and supporting activities made by actors belonging to the relational network. This scenario is particularly effective when the involvement of local institutions is foreseen in order to guarantee a real economic and organizational impact on the territory. Thus, the "Tourism in Mobility Brokerage" scenario derives from a best practice analysis of the sector that validate the idea that tourism policies are more effective when they come along with technological solutions seen as an organized collection of applications that make use of social networks. Tourism Broker enhances tourist offer on a territory and becomes supporter of the integration needed among the different system components and, in particular, among public and private actors both during managing phase and project development phase. The *Role Diagram* of "Tourism in Mobility Brokerage" scenario is depicted in Figure 4.
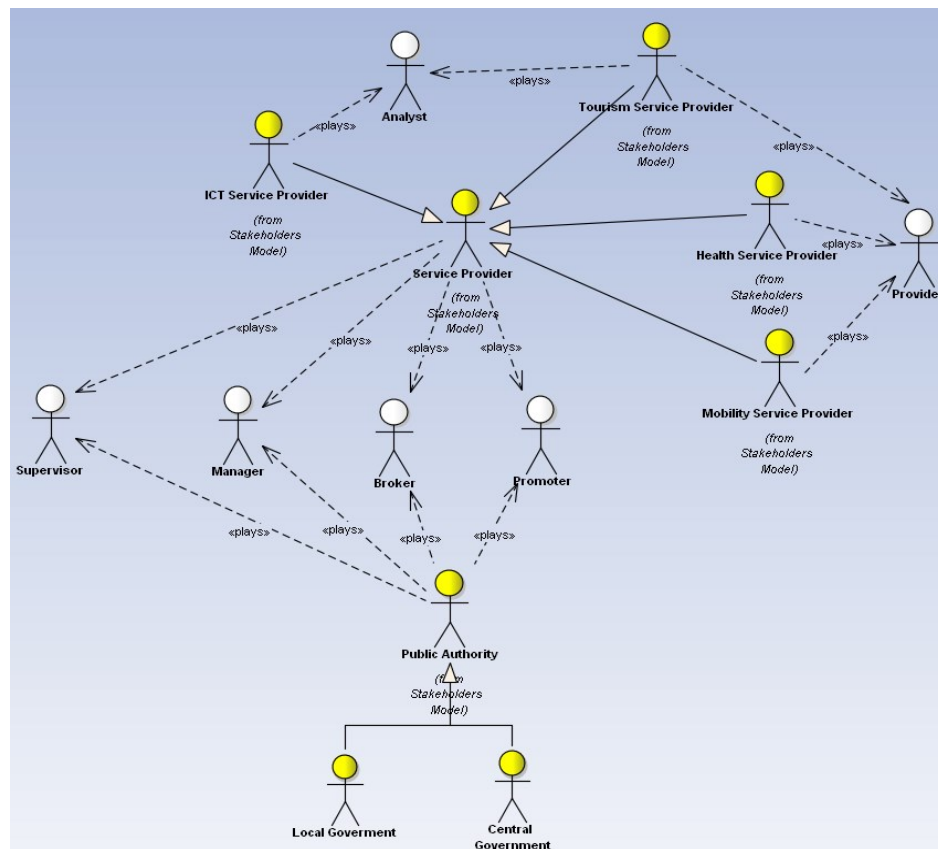

Figure 4: "Tourism in Mobility Brokerage" scenario: Role Diagram

In the end, Use Case Modeling focuses on a particular Application Scenario: "Event Tourism", including a main Use Case diagram where actors, corresponding to specific roles, can perform a basic set of functionalities for events organization (creation, planning, production and follow up). As explained in (Getz, 2008), organization of new events can be considered as resources to exploit since they are highly valued as attractions, catalysts, animators, place – marketers and image - marketers for increasing the value of a destination.

The following Top Level Use Case diagram has been identified: "Event Tourism". This functionality involves as primary actor the Event Broker and, as primary process, the composite process "Event Production" which includes an internal (sub) process named "Packaging with collateral services", from which other related use-cases arisen.

These information are sketched in the Use Case Diagrams showed in Figure 5 and Figure 6.
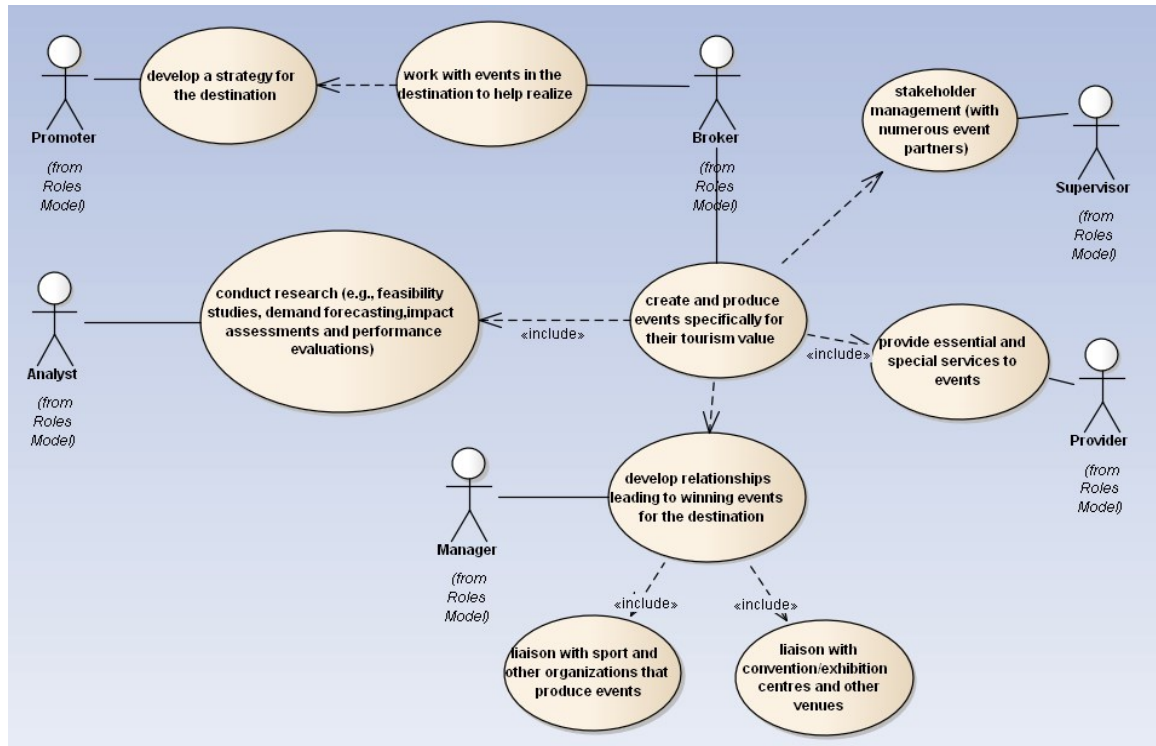

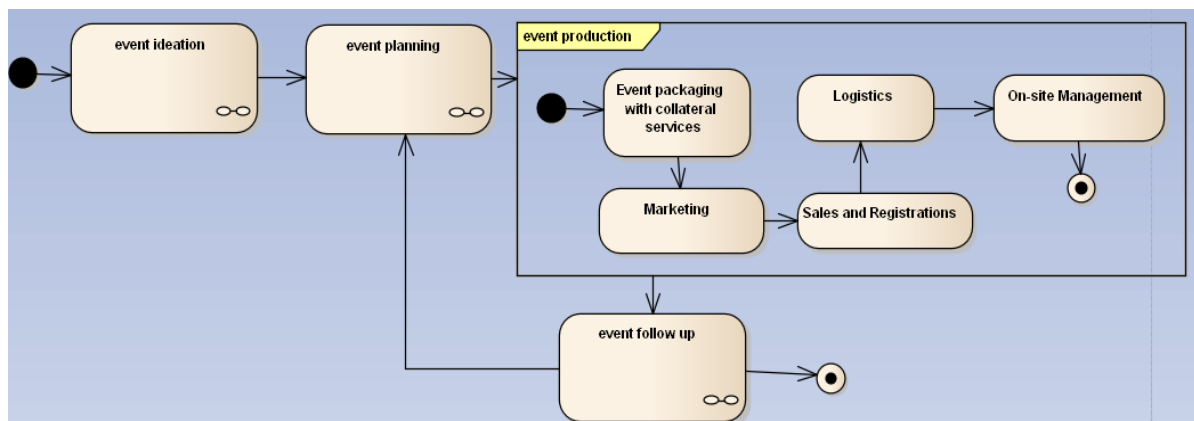Figure 5: "Event Tourism" application scenario: Use Case Diagram

Figure 6: "Event Production" process: Process Diagram

The Process Diagram of Figure 7 represents the specific action flow related to the "Event Packaging with collateral services" (sub) process.
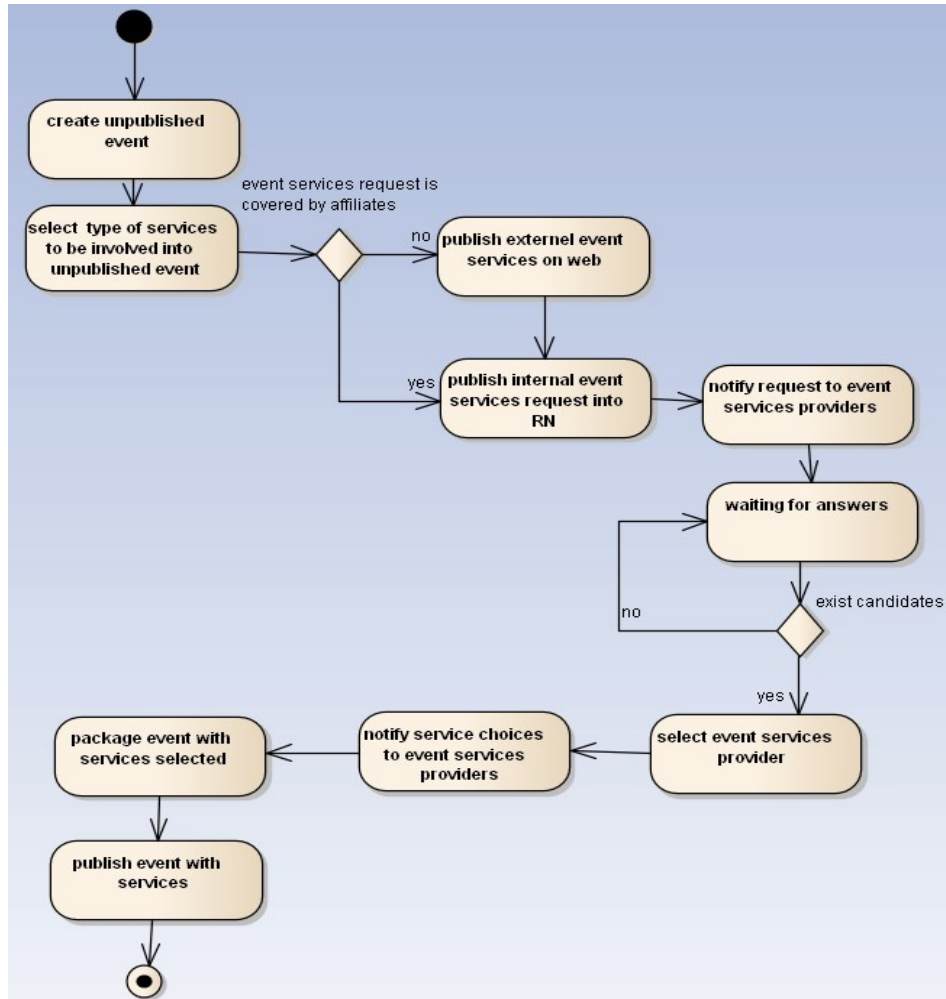
Figure 7: "Event packaging with collateral services" (sub) process: Process Diagram

V - Conclusions

The paper has proposed a Goal-Oriented Requirement Methodology (GOReM) that aims to address the needs and criticalities that often arise in the context of real large research projects. Indeed, this kind of projects, that usually involve several and heterogeneous partners working concurrently to achieve project goals in an orderly and timely way, require methodologies to be exploited in the early phases in order to clearly represent the reference context and to provide a solid ground for partners' cooperation, effort harmonization and outcomes validation. GOReM enhances good practices as those on i*, KAOS or GRL (see section II), by means of a stronger separation between the concept exploitation phase and the solution definition (e.g. Context and Scenario models are well separated from Use Case models). Moreover, in contrast with the above mentioned approaches, where proprietary modeling languages and notation are often introduced, GOReM is based on the popular UML notation; this promotes cooperation and sharing among involved partners of goals and results and allows to be aware of "who does, what and why". This seems particularly efficient when many partners work together in large industrial research projects. In particular, GOReM, which has been derived from a real experience related to the DICET-INMOTO project, has proved its effectiveness in fostering cooperation and clean project evolution due to its capability to fully support the requirements engineering phase by offering system models that, although easy to draw and understand, can be used as a valid ground to feed the design and development phases. Key features of the

GOReM Methodology are: a focus on system stakeholders and their goals; a clear distinction between business scenarios and application scenarios modeled through use cases; an explicit representation of the norms and rules that control the system evolution and the identified business scenarios; the focus on the strengths, weaknesses, opportunities and threats that can affect a business scenario.

Future research efforts will be devoted to improve GOReM on the basis of the feedback coming from its users, to further formalize the meta-model that is at the base of GOReM work-products, to develop an integrated tool-chain, based on open source software, for seamlessly supporting all phases of the methodology and thus further promoting its exploitation.

Acknowledgment

References

D. Amyot, A. Shamsaei, J. Kealey, E. Tremblay, A. Miga, G. Mussbacker, M. Alhaj, R. Tawhid, E. Braun, and N. Cartwright, 2012. *Towards Advanced Goal Model Analysis with jUCMNav*, Proc. of Fourth International Workshop on Requirements, Intentions and Goal in Conceptual Modeling (RIGiM 2012)

R. Ali, F. Dalpiaz and P. Giorgini, 2013. *Reasoning with contextual requirements: Detecting inconsistency and conflicts*, in Journal Information and Software Technology, Volume 55 Issue 1, January, pp. 35-57

P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, 2004. *Tropos: An agent-oriented software development methodology*, Autonomous Agents and Multi-Agent Systems, vol. 8, no. 3, pp. 203–236

Business Process Model and Notation (BPMN), 2011. OMG, version 2.0

P. Caire, N. Genon, P. Heymans and D. L. Moody, 2013. *Visual Notation Design 2.0: Towards User Comprehensible Requirements Engineering Notations*, Proc. of the 21st IEEE International Conference on Requirements Engineering (RE'13), Rio de Janeiro, Brazil, pp. 115 – 124

L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, 2000. *Non-Functional Requirements in Software Engineering*, Kluwer

W. Engelsman and R. Wieringa, 2012. *Goal-Oriented requirements engineering and enterprise architecture: two case studies and some lessons learned*, Proceedings of the 18th international conference on Requirements Engineering: foundation for software quality (REFSQ'12), pp. 306-320

D. Getz, 2008. *Progress in Tourism Management Event tourism: Definition, evolution, and research*, ScienceDirect, Tourism Management, pp. 403–428

T. Hill and R. Westbrook, 1997. *SWOT analysis: It's time for a product recall*, Long Range Planning, vol. 30, no. 1, pp. 46 – 52

jUCMNav v6.0.0 http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/WebHome

A. van Lamsweerde and E. Letier, 2004. *From object orientation to goal orientation: A paradigm shift for requirements engineering*, Proc. of Radical Innovations of Software and Systems Engineering in the Future, ser. Lecture Notes in Computer Science

A. van Lamsweerde, 2000. *Requirements engineering in the year 00: A research perspective*, Proc. of the 22nd International Conference on Software Engineering, (ICSE '00). New York, pp.5–19

A. van Lamsweerde, 2001. *Goal-oriented requirements engineering: a guided tour*, Proc. of the 5th IEEE International Symposium on Requirements Engineering, Toronto, pp. 249 – 262

J. Lind, 2001. *Issues in agent-oriented software engineering, in Agent- Oriented Software Engineering, ser. Lecture Notes in Computer Science*, Eds. Springer Berlin Heidelberg, vol. 1957, pp. 45–58

Object Management Group, *Unified Modeling Language (OMG),* www.omg.org/spec/UML/

D. Quartel, W. Engelsman, H. Jonkers, and M. van Sinderen, 2009. *A goaloriented requirements modelling language for enterprise architecture*, Proc. of IEEE International Enterprise Distributed Object Computing Conference (EDOC '09), pp. 3–13

I.M. Ramirez, M. Vergne, M. Morandini, L. Sabatucci, A. Perini and A.Susi, 2012. *Where Did the Requirement Come from? A Retrospective Case Study*, Proc. of Fourth International Workshop on Requirements, Intentions and Goal in Conceptual Modeling (RIGiM 2012)

C. Rolland, J.Horkoff, E. Yu, C.Salinesi and J. Castro, 2012. *Preface to the Proceeding of Fourth International Workshop on Requirements, Intentions and Goal in Conceptual Modeling (RIGiM 2012)*

Sommerville, 2011. *Software Engineering*, 9th ed. Addison-Wesley.

A. Sutcliffe and P. Sawyer, 2013. *Requirements Elicitation: Towards the Unknown Unknowns*, Proc. of the 21st IEEE International Conference on Requirements Engineering (RE'13), Rio de Janeiro, Brazil, pp. 92 – 104

E. Yu, 1997. *Towards modelling and reasoning support for early-phase requirements engineering*, Proc. of the Third IEEE International Symposium on Requirements Engineering, pp. 226–235

F. Zambonelli, N. R. Jennings, and M. Wooldridge, 2003. *Developing multiagent systems: The Gaia methodology*, ACM Trans. Softw. Eng. Methodol., vol. 12, no. 3, pp. 317–370

Z.151: Recommendation ITU-T Z.151 (10/2012), User Requirements Notation (URN) - Language definition, https://www.itu.int/rec/T-REC-Z.151-201210-I/en