Comparative Study of Goal-Oriented Requirements Engineering

Naeem Ur Rehman
Center of Research in Data Engineering (CORD)
Mohammad Ali Jinnah University (MAJU), Islamabad, Pakistan

Sarfraz Bibi
Department of Computer Sciences, SZABIST, Islamabad, Pakistan

Sohail Asghar
Center of Research in Data Engineering (CORD)
Mohammad Ali Jinnah University (MAJU), Islamabad, Pakistan

Simon Fong
Faculty of Science and Technology
University of Macau
Macau SAR

***Abstract*- Requirements Engineering is a sub discipline of Software Engineering, which is an important and vital phase in the overall software engineering life cycle. Software as a final product is deemed to fail if it does not fulfill the needs of its users. A number of contributions have been made to refine the requirements engineering process. Most of them attempted to link requirements to goals. This new trend is called as Goal-Oriented Requirement Engineering (GORE). A number of goal-oriented RE methods have been subsequently derived by various researches. This paper reviews some of the significant research works done in GORE. The comparative study may serve as a guide for readers to select an appropriate goal-oriented technique for fulfilling his/her requirements engineering needs.**

I. INTRODUCTION

Goals have been brought into RE activities for a number of reasons where customer/consumer satisfaction regarding their needs and intentions is one out of many. In some RE agendas, goals are of fundamental importance; while in others, they perform a supporting role in success of certain factors. Goal-Oriented RE has been supported by a great deal of research contributions. Researchers and practitioners recognized the valuable role of goals in requirements understanding, requirements gathering and requirements validation. Goals verify or validate requirements because they link requirements to user's needs. Where do goals come from? The answer to this question is the greatest challenge of GORE. Therefore, a great deal of efforts is required to identify goals and especially those goals that contribute most in fulfilling user's needs. Then the process of prioritizing those goals based on domain specific needs and justifications follows. In this review paper we discuss different approaches to Goal Oriented Requirement Engineering. The paper is organized in four sections. Section 2 gives details of each significant research work by showing its research question, proposed solution, explanation and its shortcoming(s) if any. In section 3 we provide our analysis over those contributions by discussing target natures, supports and problems. A conclusion of all our discussion in this review is drawn at the end.

II. DISCUSSION ON GORE CONTRIBUTIONS

Paolo Denzelli and Paolo Bresciani in [2] addressed the problem of refinement of high-level requirements (gathered from stakeholders) into detailed descriptions of system implementation (architecture, components, functions) and proposed a framework to gather requirements and to transform them into system implementation by proposing the notion of actor, goal, and intentional dependency. They called this, Requirements Engineering Framework (REF).

Explanation: REF includes three phases:

a. Organizational modeling

During organizational modeling, the analysis of overall organizational structure and architecture takes place. Besides, identification of actors, dependency among them, their soft and hard goals, and clashes among soft and hard goals of a single actor or multiple actors are carried out in this phase.

b. Hard goal modeling

During hard goal modeling, actor's hard goals are modeled in terms of decomposition to their sub-hard-goals and tasks.

c. Soft goal modeling

During soft goal modeling, soft goals are modeled in terms of decomposition to their sub-soft-goals, sub-hard-goals, tasks, and constraints. Constraints may be associated to soft goals that specify certain quality attributes.

This case study applies REF to an e-government project called Electronic Record Management System (ERMS) and at the end they propose a detailed requirements list and architecture to ERMS.

Potential Issues:

a. Can only be applied in socio-technical systems/setups
b. Addresses only early stages of RE

Sebastian Herold, et al in [3] addressed the problem of risk minimization in early requirements and architectural decisions during goal-oriented requirements engineering and architectural design and derived coarse-grained architecture alternatives from high-level requirements by using a repository of generic architectural drivers.

Explanation: Their approach is based on requirements-architecture relationship, making architectural knowledge explicit during early stages of software development i.e. RE. shown in Figure 1, this approach comes across three major phases:

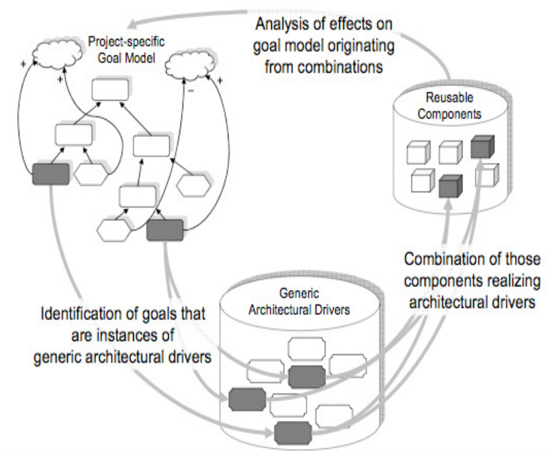

Fig. 1. Overview of envisioned approach [3]

a. Identifying goals and their relationships to various architecture drivers. During this phase a goal model is constructed, which models hard goals, soft goals, tasks and the relationships between goals and tasks. Goal model makes clear quantifiable intensions of certain stakeholders and possibly resolves clashes among stakeholders. Out of soft goals, various quality attributes are extracted that construct the foundation for system architecture.
b. Relating architecture drivers to generic architecture drivers taken from generic architecture driver's repository. Soft goals and relationships among them contribute to identification of various concrete architectural drivers. During this phase, these concrete architectural drivers are matched to generic architectural drivers from generic architectural driver's repository.
c. Selecting reusable architectural components based on generic architecture drivers. During this phase, reusable architectural components are compared against generic architectural drivers to decide which reusable architectural component is better than others in satisfaction of the generic architectural driver under-consideration.

Potential Issue: The "proven architectural solutions" may not properly suit the problem under consideration as similar problems may have the same nature but may also have different classes of stakeholders.

Ana Dias, Vasco Amaral and Joao Araujo in [4] addressed scalability issues of goal models due to the increase in number of goals. There is no possibility in already-proposed goal-oriented methods to handle complexity of goal models due to the increase in number of goals and proposed extension to the KAOS Language to integrate the notion of Compartments. Compartments classify goals based on their specific nature. They came up with the solution model called ECORE and a Domain Specific Language (DSL) called ModularKAOS.

Explanation: ECORE introduces the notion of compartments. Each compartment may contain goals of specific nature. There are seven types of compartments in ECORE as proposed:

a. Goal Compartment Node, which contains goals, requirements, and expectations
b. Soft Goal Compartment Node, which contains soft goals
c. Agent Compartment Node, which contains system agents and environment agents
d. Object Compartment Node, which contains entities and events
e. Domain Properties Compartment Node, which contains domain hypothesis and domain invariants
f. Operation Compartment Node, which contains operations
g. Obstacle Compartment Node, which contains obstacles

ModularKAOS provide the possibility to create goal models, responsibility models and object models. ModularKAOS support certain goal concepts while modeling

G	Goal	R	Requirement
E	Expectation	S	Softgoal
EA	Environment Agent	SA	System Agent
O	Obstacle	Op	Operation
Ev	Event	En	Entity
DH	Domain Hypothesis	DI	Domain Invariant

Fig. 2. Concepts supported by ModularKAOS [4]

Potential Issue: Users of ECORE and ModularKAOS should be familiar with KAOS Model and any KAOS tool e.g. Objective.

Haruhiko Kaiya, Hisayuki Horai and Motoshi Saeki in [5] addressed already-proposed goal-oriented methods/approaches

a. Selecting the goals to be decomposed
b. Prioritizing and solving the conflict of goals and the conflicts of stakeholders on a goal
c. Choosing and adopting a goal out of the alternatives of the goals as a requirements specification
d. Analyzing the impacts when requirements change
e. Improving the quality of the artifact developed by the method

They proposed to attach attribute values, to goals in goal graph. Attribute values are contribution values and preference values in preference matrix. They named this method Attribute Goal-Oriented Requirement Analysis (AGORA).

Explanation: AGORA introduces Preference Matrix and Contribution Value. Preference Matrix is used to find the gap of understanding a goal among stakeholders. Each goal has a preference matrix with it. Each preference matrix has a set of evaluators and a set of evaluatees. Evaluators and evaluatees are all stakeholders of the system under-consideration. Each evaluator fills a row (containing preference values of other stakeholders) according to his/her own judgment. Preference matrix helps in understanding how important a goal is and it also helps to resolve certain variances among stakeholders' views related to given goal.

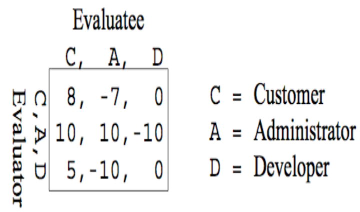

Fig. 3. Preference Matrix

Contribution value shows how many degrees a sub goal contributes to the attainment of its parent goal. The higher the contribution value of sub goal the greater will be its contribution in achievement of its parent goal. Negative value means that the sub goal blocks the achievement of its parent goal. Contribution value helps in goal prioritization.

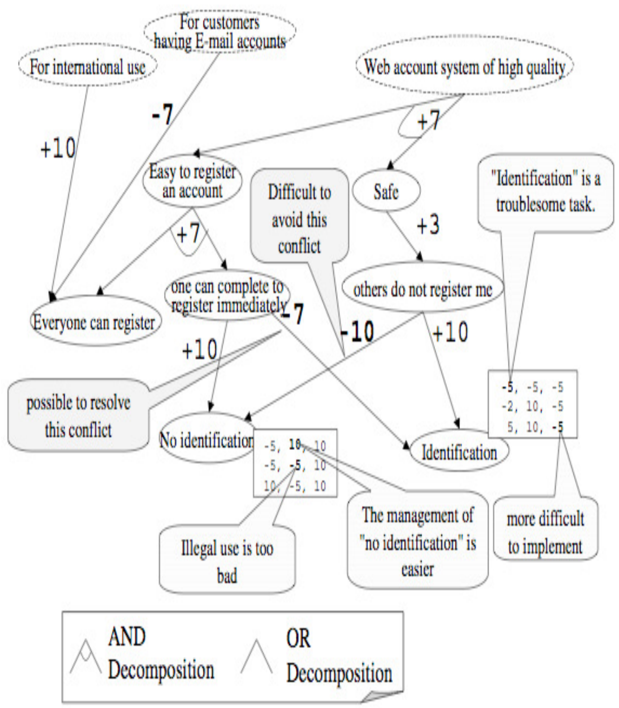

Fig. 4. Contribution values shown in graph (+10 etc) [5]

Subhas Misra, Vinod Kumar and Uma Kumar in [6] addressed the question, what should an analyst select between goal-oriented or scenario-based requirements engineering based upon the needs of problem under consideration? They derived a comparison Table based on different goal-oriented RE and scenario-based RE Techniques in the light of certain criteria.

Explanation: They proposed system analysts to use appropriate goal-oriented or scenario-based RE techniques depending on criteria mentioned in their comparison table.

Criterion 'driving methodology' in goal-oriented RE is goal analysis and that of scenario-based RE is scenario analysis, 'Objective' of goal-oriented RE concentrate on goals as goals describe user intentions while that of scenario-based RE concentrate on scenarios as scenarios make abstract user intentions clear by providing examples of how a new system may work in order to fulfill the goals of the users, 'example methodology' in case of goal-oriented RE is KAOS and that of scenario-based is SceneIC and SCRAM, 'Steps' involved in goal-oriented RE are goal elaboration, object modeling, responsibility assignment and operation while that in case of scenario-based RE are domain formalization and initial requirements capture, story boarding and design visioning, requirements elaboration, prototyping and requirements validation, 'complexity of analysis' requires less use cases than number of scenarios in scenario-based RE so goal-oriented analysis requires lesser time than scenario-based RE, 'critical problems' like modeling agent, conflict between requirement volatility and architectural stability and the gap between RE research and formal specification may occur in goal-oriented RE and that of scenario-based RE like sampling and coverage may occur.

Advantages of goal-oriented RE

a. From goals one can systematically derive requirement and object models
b. Goals give the rationale for requirements
c. A goal graph can provide traceability from strategic concerns to technical details
d. Goal formalization can prove if the refinements are correct and complete
e. The goal refinement structure can indicate a comprehensible structure that is helpful in the requirements document
f. Alternative system proposals also could be explored with the help of alternative goal refinements

Disadvantage of goal-oriented RE

a. Unless a rigorous automated reason is used with formal methods, an abstract model may go unquestioned.
b. Records contain vague intentions without thinking properly about the practical applications.

Advantages of scenario-based RE

a. From scenarios one can ground argument with specific examples
b. One can address the devil in the detail during the requirement specification and validation. (3) During the phase of requirements validation, the scenarios can help in testing the models and specifications

Disadvantages of scenario-based RE

a. Often being specific loses the generality

Potential Issue: The comparison table explains steps of KAOS (a goal-oriented technique), SceneIC and SCRAM (scenario-based techniques) while giving the advantages and disadvantages of many others. The comparison table should be in general comparing goal-oriented and scenario-based techniques.

A M Sen and S K Janin in [7] addressed the problem of elicitation/extraction of soft goals from stakeholders and proposed a methodology to elicit soft goals from different stakeholders involved in goal-oriented RE. They feed goal preferences of all stakeholders involved in an algorithm (on computer). Algorithm gives a list of prioritized goals.

Explanation: They proposed an agile visualization technique to extract soft goals by doing an agent based goal refinement process repeatedly. These repetitions are done in sprints. Each sprint comes across develop, wrap, review, and adjust phase. Their proposed methodology involves some steps:

a. Planning for goals
b. Distributing activity cards to stakeholders involved
c. Preparation of goal preference model
d. Inputting goal names and their preference values in activity card compiler
e. Generation of prioritized goals
f. And a finite number of sprints (iterations) each having develop, wrap, review and adjust step

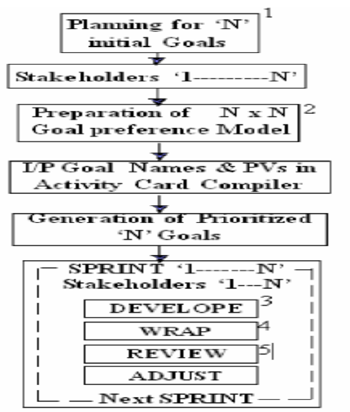

Fig. 5. Steps involved in their proposed method [7]

The above-mentioned steps are mainly divided in two phases. Pre-sprint phase and sprint phase. Pre-sprint phase is performed once while sprint is repeated until enough satisfaction is achieved. Pre-sprint phase involves preparation of an initial list of achievement and maintenance, distribution of prepared list among all the stakeholders involved, preparation of goal preference model by assigning a priority value by each stakeholder, analysis of goal preference model by analyst to generate a compilation table (using activity card compilation algorithm) and its distribution among all stakeholders to begin sprint.

Stake-holder / Name of Goals/ Sub-Goals	Sh-1	Sh-2	…..	Sh-M	Link with Predecessor Goals or Sub-goals	Priority After Compilation
A0	PV_{11}	PV_{11}	……	PV_{1M}	*No Link	
A1	PV_{21}	PV_{22}	……	PV_{2M}	*No Link	
A2	PV_{31}	PV_{32}	……	PV_{3M}	*No Link	
. . . . .	. . . . .	. . . . .	. . . .	. . . . .		
AN	PV_{11}	PV_{11}	……	PV_{NM}		

Fig. 6. Compilation table after analysis [7]

During sprint each stakeholder gets a chance to be part of a collective review of activity cards and to adjust his/her elicited soft goals given in his/her activity card. As a result of which he/she might elicit some more soft goals.

Potential Issue: Time consuming as number of sprints may increase due to conflicts. Also the higher the number of stakeholders involved the longer the time required carrying each sprint due to the greater number of activity cards.

Davide Bolchini and Paolo Palolini in [8] addressed a new conceptual tool for effectively supporting the activity of requirements analysis of web applications:

a. High-level communication and business goals have to be addressed and be carefully analyzed as a fundamental part of requirements management
b. Goals and needs of stakeholders have to be tied up with design in a consistent fashion as bridging the current gap between requirements and hypermedia specifications
c. Lightweight, usable, and unceremonious models needed for web designers and analysts with no engineering background

Explanation: Web applications today are primarily articulated means of communication between end-users and stakeholders who conceive web and their web resources. Poor requirements augment the risk of missing the opportunity to meet customers' needs and enhancing the user experience. The specific needs of requirements management for web application can be summarized as follows:

a. Managing complexity: High-level of communication and business goals have to be addressed and be carefully analyzed as an integral part of requirements management
b. Requirements-Design Gap: Goals and needs of the stakeholders required to be tied up with design in a coherent fashion as bridging the current gap between requirements and hypermedia specifications
c. Accessibility: Junior and less-experienced designers (often with no engineering background) need informal, lightweight, straightforward methods and best practices to dealing effectively with web application requirements.

Potential issue: Growing size and complexity of web applications due to service sophistication.

Paolo Giorgini, Stefano Rizzi and Maddalena Garzetti in [9] addressed a goal-oriented approach to requirement analysis for data warehouses, based on the Tropos methodology and proposed a requirements analysis approach with two perspectives. They are organizational modeling, centered on stakeholders, and Decisional modeling, focused on decision makers. This approach can be employed within both a demand-driven and a mixed supply/demand-driven design framework. In the second case, while the operational sources still explored to shape hierarchies, user requirements play a fundamental role in restricting the area of interest for analysis and in choosing facts, dimensions, and measures.

Explanation: Warehousing projects are long-term, and most requirements cannot be stated from the beginning:

a. Information requirements for DW applications are difficult to specify since decision processes are flexibly structured, poorly shared across large organizations, jealously guarded by managers, and unstable in time to keep pace with evolving business processes
b. Requirements for decision making often refer to information that does not exist in required form, and must be derived from data sources.

Farida Semmak, Christophe Gnaho and Regine Laleau in [10] proposed a Component-based approach to specify trustworthy systems from the requirements phase to the specification phase. This paper mainly deals with the improvement of requirements elicitation and proposed to apply reuse-based techniques at goal level in order to improve requirements engineering in the context of Cycab domain. These techniques are inspired of the field of software product lines engineering and domain engineering. The aim is to provide a requirements model that captures the commonality and variability of domain from which requirements model for specific systems can be derived according to some options selected by the stakeholders.

Explanation: The domain level provides Requirements Family Model (RFM), which enables the description of diverse applications of the same domain by identifying and expressing the common and variable requirements at goal level.

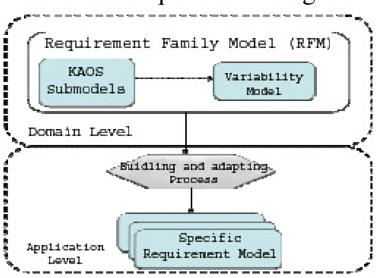

Fig. 7. Extended KAOS to support Visibility for Goal Oriented RE [10]

To specify RFM, they choose the KAOS Goal-Oriented Requirement Engineering approach. However, the KAOS approach has not been designed to address a class of systems of a given domain and then it does not explicitly take into account variability concerns.

Potential Issue: The main purpose is to derive the specific requirement model from the RFM according to stakeholders' needs. The paper mainly focuses on the domain level.

Anthony Finkelstein and Andrea Savigni in [11] addressed the fundamental problems for both requirements engineering and software architecture and their relationship and proposed reflection-based framework for requirements engineering. The framework addresses the key difficulties such as changing context and changing requirements.

Explanation: "Context-awareness" means the ability of a particular service to adapt itself to a changing context. One classical example is mobile commerce applications, which should run equally well on full-fledged web browsers running on desktop computers.

a. Requirements engineering in the area of context-aware services, especially when these are targeted towards mobile devices, poses new and very challenging problems.
b. In case of context-aware mobile services, changing context may entail:
c. Changing location. This means not only that the absolute location of a device can change, but also that the relative locations of two devices must be taken into consideration
d. Changing display characteristics e.g., graphics PDAs. Text-only mobile phones, color vs. monochrome displays, etc
e. Changing usage paradigm. For example, from a user perspective having a full-screen, button-centered PDA is very different from using a scroll-centered mobile phone
f. Target platforms unknown in advance. Note that this problem is not implied by any of the preceding points. Platforms may be unknown in advance, and the service should anyway be able to dynamically adapt itself to this aspect of the new context. This means of performing a very hard abstraction job in order to express the common set of characteristics in a general, uniform way.

Potential Issue: Context-aware services problem especially available to mobile devices

III. COMPARISON

This section encompasses the analysis on previously mentioned research contributions and provides analysts with the target nature where he/she is willing to use a particular technique/method. Provided support by that method/technique and also problems that he/she might trade off.

TABLE I

No	Nature	Support	Problem
[2]	Socio-economic systems and e-organizations' systems where actors perform to achieve goals and actors are dependent on each other in achievement of goals	- Understanding of the whole system in easy ways of models - Models are organizational models (actors and their dependency), hard and soft goal models (goals, their dependency and hidden quality factors in soft goals)	- No use in real-time systems - Can only be used in early stages of requirements engineering
[3]	Systems where goals have a great impact on system architecture	- Extraction of architectural attributes out of goals even in very early stages of requirements engineering	- Given architectural solutions out of which a suitable architectural solution taken may not suit the problem under-consideration

[4]	Systems with large number of goals and stakeholders involved	- Easily adjust large number of goals in different categories called Compartments, hence reducing a great of complexity - Use of familiar goal concepts and domain specific language	- Previous knowledge of KAOS modeling and any KAOS tool is necessary
[5]	Systems where goals are not clear cut or composite and need to be decomposed for the purpose of understanding. Also where goals need to be prioritized	- Goal decomposition for the purpose of better understanding - Conflict resolution among stakeholders using preference matrix - Understanding goal dependency on its sub-goal(s) and easy and quick achievement of goal by achieving its sub-goals using contribution values	- Subjective evaluation in preference matrix may be faulty causing time-consuming conflict resolution
[6]	Socio-technical, e-organizations etc	- Understanding where and why goal-oriented or scenario-based requirements engineering technique should be used - Comparison among goal-oriented and scenario-based requirements engineering techniques using a set of criteria	- The comparison table does not differentiate goal-oriented and scenario-based RE techniques generally rather different techniques and their steps are compared
[7]	Systems where a large number of stakeholders are involved and elicitation of soft goals is a problem and due this large number conflict occurrences	- Simple and quantitative way of soft goals elicitation - Iterative nature of soft goal elicitation and conflict resolution - Goal prioritization	- Long sprint sessions due to large number of stakeholders
[8]	Web Systems	- Reducing and managing complexity of bigger web projects - Reducing the gap between requirements and hypermedia design	- Growing size of web application due to service sophistication
[9]	Data extensive systems where large data sources are involved	- Organizational modeling focusing stakeholders - Decisional modeling focusing decision makers - Integration to demand-driven and mixed design framework	
[10]	System where goal-oriented requirements come across the process of reuse due to varying requirements	- Identification of varying requirements at goal level	- Only domain level application is involved
[11]	Service based systems	- Handling of changing context and changing requirements - Service auto-adaptation	- Context-aware services problem especially in case of mobile devices

IV. CONCLUSION

This review work encompasses selected research contributions in the field of goal-oriented requirements engineering. Each method has its own way to deal with the problem i.e. GORE. A unified solution to all the problems is not possible as new problems emerge with the passage of time and with the progress in technology, collaboration, growth and many of these revolutionary attributes. Therefore, it is observed that it is the responsibility of an analyst to select a method that best suits his/her organizational needs or the problem under consideration and the stakeholders involved.

REFERENCES

[1] Evangelia Kavakli, “Goal-Oriented Requirements Engineering: A Unifying Framework”, Requirements Engineering Journal, Springer-Verlag London, 2002, pp.237-251.

[2] Paolo Denzelli, Paolo Bresciani: “Goal Oriented Requirements Engineering: A Case Study in E-Government”, 15th International Conference of Advanced Information Systems Engineering (CAiSE 2003), Klagenfurt, Austria, June 2003, pp.601-616.

[3] Sebastian Herold, Andreas Metzger, Andreas Rausch, Heiko Stallbaum, “Towards Bridging the Gap between Goal-Oriented Requirements Engineering and Compositional Architecture Development", 29th International Conference on Software Engineering Workshops (ICSEW'07), pp.7-12.

[4] Ana Dias, Vasco Amaral, Joao Araujo, “Towards a Domain Specific Language for a Goal-Oriented Approach based on KAOS”, 3rd International Conference on Research Challenges in Information Science (RCIS 2009), pp.409-420.

[5] Haruhiko Kaiya, Hisayuki Horai, Motoshi Saeki, “AGORA: Attributed Goal Oriented Requirements Analysis Methods”, IEEE Joint International Conference on Requirements Engineering, 2002, pp.13-22.

[6] Subhas Misra, Vinod Kumar, Uma Kumar, “Goal Oriented or Scenario Based Requirements Engineering Technique- What a practitioner should select?”, Canadian Conference on Electrical and Computer Engineering, May 2005, Sask, pp.2288-2292.

[7] A M Sen, S K Janin: “A Visualization Technique for Agent Based Goal Refinement to Elicit Soft Goals in Goal Oriented Requirements Engineering”, Second International Workshop on Requirements Engineering Visualization (REV 2007), Oct. 2007, pp.2.

[8] David Bolchini, Paolo Paolini, “Capturing Web Application throught Goal Oriented Analysis”, Proceedings of the Workshop on Requirements Engineering (WER 02), 2002, pp.16-28.

[9] Paolo Giorgini, Stefano Rizzi, Maddalena Garzetti, “Goal-Oriented Requirement Analysis for Data Warehouse Design”, In Proc. DOLAP, 2005, ACM Press, pp.47-56.

[10] Farida Semmak, Christophe Gnaho, Regine Laleau, “Extended KAOS to Support Variability for Goal Oriented Requirements Reuse”, International Workshop on Model Driven Information Systems Engineering: Enterprise, User and System Models (MoDISE-EUS'08), June 2008, France, pp.22-33.

[11] Anthony Finkelstein, Andrea Savigni, “A Framework for Requirements Engineering for Context-Aware Services”, In Proceeding of 1 st International Workshop From Software Requirements to Architectures (STRAW 01), 2001, pp.200-206.