

Evaluation of Object Oriented Requirements Engineering Frameworks: A Study

Shreya Banerjee

Department of Computer Applications
National Institute of Technology,
Durgapur, India

Anirban Sarkar

Department of Computer Applications
National Institute of Technology,
Durgapur, India

ABSTRACT

With the growing demand of Object Oriented paradigm in the development of large scale information system, a “good” requirements engineering framework has become essential for analysis of requirements and related models for such system. Though various proposals of requirements engineering frameworks are exist in the literatures but they are varied in the specification mechanism of the requirements and the exploration of object oriented paradigm in the process. This paper has summarized the available object oriented requirements engineering frameworks with their relative advantages and disadvantages. On next, several features have been proposed and described which are essential to characterize the quality and usability of such frameworks. Based on those features a detailed comparative study also has been performed for the available frameworks. Finally, several potential issues have been described as future research directions for object oriented requirements engineering frameworks.

General Terms

Object Oriented Requirement Engineering, Requirement Engineering Framework, Framework Level Features, Quality Evaluation Level Features.

Keywords

Requirement Object, Requirement Analysis, Business Object, Requirements Engineering Framework, Object Oriented Analysis.

1. INTRODUCTION

The growing popularity of Object Oriented paradigm in the development of complex and large scale information system has led the influence of Object Oriented Requirements Engineering towards elicits and analyzes the requirements of such system. In general, Requirements Engineering is a structured process of eliciting, defining, negotiating, prioritizing and validating requirements of a system. Whereas, object oriented requirements engineering is an approach to encapsulating information about the process and product, as well as functionality into a requirements object [4]. In object oriented paradigm requirements are directly represented as first-class objects and the notation of a “requirements object” is used to represent the problem domain object [7]. A first-class object has the supports of all the features of object oriented paradigm and does not exhibit any difference between objects in the domain model and requirements objects in general. Requirements objects can be organized in generalization hierarchies that reflect different kinds of requirements [19]. They can be associated both with other requirements objects and domain objects, and they can have attributes. In any case, the requirements engineering process starts from requirements elicitation and where client or stakeholders of the system plays a key role. But it is

challenging for the stakeholders of the information system to produce the absolute, well-defined and comprehensive set of requirements when the process of requirements elicitation commence. That directly affects the analysis, design and other later stages of the information system development. For complex and large scale information system development process, the stakeholder’s requirements are often changed. Moreover, business processes of large system are dynamic in nature. Thus it is very difficult to anticipate any future requirements for such system in the scenario of evolving business processes over time. Further, requirements for complex and large systems are difficult to specify at the early stage because business processes are now days flexibly structured and shared across the different sectors of the large organization. In this scenario, Process Driven Requirements Engineering is proven to be more suitable for complex and large information system. It must also have support for reuse of domain level abstractions and step wise refinement mechanisms strictly for mapping the requirements in high level design. The use of object oriented paradigm to define and analyze the requirements is added advantageous in such cases. Further, for successful completion of complex and large scale information system development, a semantic representation of software requirements is required to remove the communication gaps exist between stakeholders, software engineers and project managers [20]. These raise the demand of user centric or demand driven requirements engineering methodology where the requirements objects can be specified and represented semantically. Moreover, the requirements objects can be represented using a common set of semantic notations which are understandable both to the stakeholders and domain engineers. At the same time, to accommodate the requirements changes, the representation of the requirements objects must be reusable. Thus, requirement engineering must start focusing with early requirements analysis and further it should move on detailed requirements analysis. Analyzing early requirements will significantly decrease the possibility of misunderstanding the stakeholder’s requests and consequently reduce the risk of failure for development of large scale information system.

Several research proposals of requirements engineering frameworks are exist in the literatures to analyze the object oriented system. Among them Use-Case modeling of UML are most popular and plays a central task towards the requirements analysis of object oriented software, though having several disadvantages compare to other approaches. This paper proposed and describe a set of features those are important for a “good” requirements engineering framework. Moreover, based on those features, a detailed comparative study of all existing object oriented requirements engineering frameworks also has been included in the paper. Besides this, broad areas of future research directions also have been described for requirements engineering domain.

With the aforementioned objectives, the paper has been organized in six sections. In section 2 previous researches related to the object oriented requirements engineering

frameworks have been summarized. In section 3, a set of general features has been described for a “good” requirements engineering framework for such system. The comparative study of existing frameworks has been discussed in section 4. In section 5, the future research directions of requirements engineering frameworks been summarized. Finally, the chapter has been concluded with the section 6.

2. RELATED WORK

Several facets and approaches for object oriented requirements elicitation and analysis has been discussed in [7] and [19]. In [7] an effort has been made to represent the requirements as object. In the approach requirements have been combined with domain object. But it suffers from user centric requirements representation.

Also several Process Oriented Requirements Engineering approaches have been proposed in recent years to minimize the communication gap between stakeholders and software developers for the complex and large scale system development. A thorough study and examination of the gap between classical requirements engineering approaches and process based requirements engineering has been made by [2, 13]. In [2] author also has proposed a new requirements information model to address the critical involvement of stakeholders in complex information system analysis. But the model is not semantic in nature. A requirements engineering approach which focuses on the visualization of requirements has been presented in [3]. But it also lacks from semantic representation of requirements and reusability of its specifications. In [5], an object oriented framework has been proposed to understand the business process. This approach described two models namely, business object models and role models to understand the interactions between deferent business processes and which are critical for requirements analysis of large and complex information system. Several advantages of business process oriented requirements engineering for large scale information system development has been addressed in [13]. The business process model based approaches allow the dynamic behavior of the organization to be understood by the stakeholders and information system developers. Thus it is more suitable for requirements engineering of large scale complex information system development. But majority of such proposed approaches are limited to early analysis phase of requirements only. Even though very few approaches are suitable for user centric requirements engineering due to less involvement of stakeholders. The major contributions are summarized in following sub-sections.

2.1 UML based Requirements Engineering

The representation of business requirements using Unified Modeling Language has been addressed in [8] and [9]. The central task in these approaches is to develop the Use-Case model to represent the usage of the system. However, use cases are actually not “object oriented” and does not specify the functional requirements of the system [19]. Moreover, the UML and extensions to UML represent software elements using a set of language elements with fixed implementation semantics (e.g. methods, classes) and more suitable represent the design and further implementation model. Requirement Object, however, represents the domain requirements. In other words, representation of requirements using class may blur the concepts of domain objects and software objects.

2.2 MORE – Model based Object-Oriented Requirements Engineering

In [15] a model based object oriented requirements engineering framework has been proposed to capture the

domain knowledge in well-defined model. It also supports the checking for several properties of requirements like completeness, consistency, traceability and reusability of requirements. The constructs of MORE are formal and promotes the software requirements development from document-centered to model-driven development. However, stakeholders requirements cannot be addressed in this approach and simply rely on representation using natural language for such requirements.

2.3 GORE – Goal Oriented Requirements Engineering

In [1] the Goal Oriented Requirements Engineering (GORE) has been introduced. Goal-oriented requirements engineering is concerned with the use of goals for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements. Goals may be formulated at different levels of abstraction, ranging from high-level strategic concerns to low-level technical concerns through iterative process. Goals are generally modeled by intrinsic features such as their type and attributes, and by their links to other goals and to other elements of a requirements model.

2.4 AGORA–Goal Oriented Requirements Analysis

An extension of Goal Oriented Requirements analysis has been proposed in [17] where several attributes like contribution values and preference matrices are used with goal concept. These attributes can help an analyst to choose and adopt a goal from the alternatives of the goals, to recognize requirements changes. A detail study of the requirements engineering methodologies based on goal oriented approaches have been discussed in [16]. Goal oriented analysis can greatly facilitate and rationalize early phases of the software design process for a large software system through analyzing the requirements and architectural design [10]. But perhaps it is not suitable for detailed analysis of requirements for such system. Even though the GORE is adopted, the object oriented requirements analysis are used to complement the detailed requirements analysis phase [10].

2.5 BORA – Business process Oriented Requirements Analysis

In [11] a business processes oriented requirement analysis model (BORA) has been defines using graphical representation language of business processes and its formal descriptions. It also has introduced the Software Requirements Automatic Generator (SRAG) from BORA.

2.6 Business-Object Oriented Requirements Analysis

In [22], a process driven requirements analysis framework based on Common Business Objects [6] has been proposed for large scale information system. A business object (BO) captures information about a real world (business) concept, operations, constraints, and relationships between those concepts. The advantage of using this concept is that, the set of BOs can be reusable in the context of business domain and can easily be transformed into system level objects for software realization of the specific business concept. With these perspectives, the framework consists of two phases, namely, (i) Early Requirements Analysis Phase and (ii) Detailed Requirements Analysis Phase. The former allows for modeling and analyzing the contextual setting of the business domain, in which the system will operate. In later phase, the early requirements specifications are refined with the structural, functional and nonfunctional features of the domain that is relevant to the stakeholders and their roles related to

the intended information system. The refinement process is largely influenced by the concepts of Feature Oriented Domain Analysis (FODA) [18]. The requirements analysis framework is supported with the object oriented features like abstraction, inheritance and reuse capability. Moreover, it supports user centric requirements analysis approach for large scale information system. In this context, the business object based requirements analysis for data warehouse system domain has been outlined in [21]. Nevertheless the framework proposed in [22] is more generic and comprehensive in nature.

3. FEATURES OF OBJECT ORIENTED REQUIREMENTS ENGINEERING FRAMEWORKS

Various proposed approaches for object oriented requirements engineering framework majorly varies on the representation of the requirements objects and on the degree of deployment of object oriented paradigm. In this section an essential set of features for requirements engineering frameworks are being listed, specifically which are crucial for such frameworks over object oriented system. For better clarifications, the features are divided into different groups and are discussed in following subsections.

3.1 Framework Level Features

These set of features include descriptions of related concepts those are used for the purpose for devising the framework for requirements engineering. The set of features are as follows,

- (a) *Formalism*: This feature indicates the formalism used to describe the requirements engineering framework's elements.
- (b) *Graphical Notation*: It is a framework level property in which requirements are specified by graphical notation.
- (c) *Extended Framework Used*: Some frameworks have redefined or extended the concepts of other general purpose models for the analysis and design which is reflected in this description. This property indicates the use of general purpose modeling languages like ER model, UML, Common Business Object etc.

3.2 Semantic and Construct Level Features

These set of features are used to describe the expressiveness of a framework towards realizations and analysis of the target domain. The set of features are as follows,

- (a) *Object orientation*: Object orientation is an approach that models a system as a group of interacting objects. Each object represents some entity of interest in the system being modeled, and is characterized by its classifier, its state (data elements), and its behavior. In object oriented framework, requirement specifications are specified as *Requirement Object*.
- (b) *Abstraction*: Abstraction mechanism is an essential property in object oriented paradigm to reduce the complexity of the system model. Such a representation is highly flexible for the user to understand and analyze the basic structure of the target system and to formulate the alternative design options.
- (c) *Reusability*: This is another important property in object oriented models. This can be achieved either using whole-part relationship or inheritance mechanism among the requirements objects.
- (d) *Classification of requirement Object*: Requirement Objects can be classified based on the different types of domain level system elements like process, data, actors, interactions etc.
- (e) *Elicitation*: During elicitation the requirement engineer identify the basic requirements objects for the target system and also defines the system boundary.

(f) *Specification*: The specification is the final document produced by the requirements engineering frameworks. The major objective is to provide the descriptions of the requirements objects unambiguously.

(g) *Validation*: Validation of the requirements modeling is one of the major objectives of any such framework. This process is important to validate the correctness and completeness of the specified requirements objects and models against the stakeholder's requirements achieved in the elicitation process.

(h) *Requirement Management*: It is a set of activities that guide the project team to identify, control and track requirements changes for the target software system as the project proceeds.

3.3 Early Requirements Analysis Level Features

Early requirements analysis phase is important to understand how a business is perceived by its stakeholders as a set of related business processes and events. This phase is also the basis to identify the possible system level requirements of stakeholders from the business process level perspective. The important features of this phase are as follows,

- (a) *Identification of Requirements Objects*: Requirements are identified with their context and mapped in requirements objects.
- (b) *Identification of Interactions*: Several identified requirements objects are interact to achieve the functional goals of a system. Interaction is a framework level feature to realize this phenomenon.
- (c) *Identification of Roles*: In a system several elements plays different predefined roles to achieve the goals. In software engineering, the term role can be defined as separation of concerns i.e. separation of behavioral characteristic of elements in the system. Many of the requirements objects play multiple roles in the system.
- (d) *Collaborations*: Based on the roles, various requirements objects may collaborate with each other. Collaborations between those objects involve sending messages to each other and responsible for set of interactions.

3.4 Detailed Requirements Analysis Level Features

Detailed requirements analysis phase is essential to satisfy the functional and non-functional requirements of stakeholders in the targeted system for the domain of interest. In this phase the requirements and interactions identified in early requirement analysis phase are refined by adding the domain level feature like structural, functional and nonfunctional along with the attribute or operational level constraint specifications.

- (a) *Refinement of Requirements Objects*: Requirements objects and their interactions identified in early requirement analysis level are elaborated in this phase.
- (b) *Use of prototype CASE tools*: This feature is important as it provides the proof of the concepts of some requirements engineering framework. Also a prototype CASE tool is the basis for representing the requirements models and specifications using the graphical notations.
- (c) *User Interfaces*: For user centric requirements engineering, it is important to analyze that how user will efficiently interact with the system to perform the necessary activities. This feature provides the descriptions of system interfaces through which stakeholders can interact with the system.
- (d) *Mapping form Requirements to Design Model*: The requirements engineering provides the prime inputs for the design process. Thus such frameworks must support a

systematic mapping mechanism for representing the equivalent design models from the requirements objects and related models.

3.5 Quality Evaluation Level Features

Efficient system design depends on the quality of the requirements analysis modeling elements. Traditionally quality in requirement frameworks is a complex mix of factors that will vary across different applications and the customers who request requirements. However, a good requirements engineering framework must support the validation of certain quality features to evaluate the requirements objects and related models achieved from the framework. The quality factors are as follows,

(a) *Ambiguity*: It is a quality factor used to validate each requirements objects in the specification that those are atomic and cannot have more than one interpretation.

(b) *Correctness*: It is a quality factor which means how many requirements in requirement specification meet customer's need.

(c) *Completeness*: It is a quality factor which means necessary requirements objects are not lacking specification.

(d) *Consistency*: It is a quality factor used to check the presence of the inconsistency among requirements objects in the specification.

(e) *Traceability*: It is a quality factor used to specify the implementation of the requirements objects towards the related design components and vice versa.

(f) *Maintainability*: It is a quality factor which means the effort required to locate and fix an error in requirement specification.

(g) *Importance and stability*: This quality factor means how clearly the prioritization and stabilization of the requirements objects are described when they are necessary to be specified.

(h) *Reliability*: It is a quality factor which means the extent to which a software application can be expected to perform its intended functionalities as stated in requirement specification.

(i) *Efficiency*: This specifies the degree to which the requirement object makes optimal use of limited business resources.

(j) *Productivity*: It is the ratio between the functional value of software produced to the labor and expense of producing it.

(k) *Visibility*: It is the quality factor which is used to maintain end user access policies for the application.

(l) *Usability*: According to ISO/IEC 9126-1 [23], the quality characteristic called Usability includes the factors, such as operability, understandability, learnability, attractiveness, and compliance. These concepts are abstracts and cannot be measured directly, but the evaluation of the factors of usability at requirements engineering framework is important in order to produce an improved software system.

(m) *Interoperability*: This property means how efficiently coupling can be established from one system to another. The requirements related to interlinking of two systems are complex in nature

4. COMPARISON OF OBJECT ORIENTED REQUIREMENTS ENGINEERING FRAMEWORKS

The set of features described in subsections 3.1, 3.2, 3.3, 3.4 and 3.5 are used to compare the available requirements engineering frameworks those are based on object-oriented paradigm. The frameworks summarized in section 2 are used for the purpose. Each of the frameworks has deferent level of advantages and disadvantages. Majority of the frameworks are in general lacking from comprehensive detailed requirements analysis

The comparison has been summarized in Table I, Table II and Table III.

Table I. Comparison of object oriented requirements engineering frameworks for Framework Level Features and Semantic and Construct Level Features

Model Name / Reference	Framework Level Features			Semantic and Construct Level Features							
	(a)	(b)	(c)	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
[8,9]	-	Y	UML	P	P	Y	-	Y	Y	-	Y
MORE[15]	Y	-	-	Y	Y	Y	-	Y	Y	Y	-
GORE[1]	Y	Y	-	P	P	-	Y	Y	Y	Y	Y
AGORA[17]	-	Y	-	P	Y	-	Y	Y	Y	-	Y
BORA[11]	P	Y	BPR	Y	-	-	-	Y	Y	-	-
[22]	-	Y	CBO	Y	Y	Y	Y	Y	Y	-	Y

Y: supported in the model, Hyphen: not supported or not explained how to support it, P: partially supported, BPR: Business Process Reengineering; UML: Unified Modeling Language, CBO: Common Business Object

Table II. Comparison of object oriented requirements engineering frameworks for Early Requirements Analysis Level Features and Detailed Requirements Analysis Level Features

Model Name / Reference	Early Requirements Analysis Level Features				Detailed Requirements Analysis Level Features			
	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
[8,9]	-	Y	-	Y	Y	Y*	-	P
MORE[15]	Y	Y	Y	-	-	-	-	-
GORE[1]	Y	Y	-	-	Y	-	-	-
AGORA[17]	Y	Y	-	-	Y	-	-	-
BORA[11]	Y	Y	P	-	-	Y	-	P
[22]	Y	Y	Y	Y	Y	-	Y	-

Y: supported in the model, Hyphen: not supported or not explained how to support it, P: partially supported
* Third Party tools support

Table III. Comparison of object oriented requirements engineering frameworks for Quality Evaluation Level Features

Model Name / Reference	Quality Evaluation Level Features												
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)
[8,9]	Y	-	-	-	-	-	-	-	-	-	-	-	-
MORE[15]	Y	-	P	P	P	-	-	-	-	-	-	-	-
GORE[1]	Y	Y	P	P	-	-	-	-	-	-	-	-	-
AGORA[17]	Y	Y	Y	Y	Y	P	P	-	-	-	-	-	-
BORA[11]	Y	P	P	P	P	-	-	-	-	-	-	-	-
[22]	Y	-	Y	P	-	-	-	-	-	-	-	-	-

Y: supported in the model, Hyphen: not supported or not explained how to support it, P: partially supported

Several interesting conclusions can be drawn from the comparison tables. Majority of the frameworks support the use of graphical notations but do not provide formalization of the frameworks elements. Thus majority of the frameworks in the literatures are not capable of rigorous validation of the requirements. Further, majority of the requirements frameworks do not support various features related to the detailed requirements analysis. Also many of them do not facilitate any systematic guidelines towards the mapping of requirements objects and related models to their equivalent top level design models.

Most importantly, majority of the proposed frameworks in the literatures do not provide any quality evaluation scheme to assess the quality of the requirements objects and related analysis models. Few of the quality features have been defined and discussed for the framework called AGORA in [17]. However those definitions are not generic and specific for the framework itself. Also few of the quality factors can be examined at the implementation level using CASE tools for MORE [15] and BORA [11]. Moreover, very few of the existing requirements engineering frameworks are supported with the prototype CASE tools. In respect to the requirements engineering frameworks, the prototype CASE tools are useful to specify the requirements graphically. Also it is the basis of the analysis of requirements objects and related models graphically along with their validation. The prototype CASE tools also can be useful towards automatic generation of top level design model for target system from its analysis models.

5. FUTURE RESEARCH DIRECTIONS

Various potential research agendas still exist in the area of requirements engineering. Several research proposals are there in literatures related to the requirements engineering frameworks based on object oriented paradigm and they mainly varies in the mechanism for representation of the requirements objects. Though Use-Case modeling of UML is most popular towards the requirements analysis of Object Oriented software, but the technique is lacking to provide support of several features for requirements engineering frameworks based on object oriented paradigm. Several related research directions are as follows,

(a) *Formalization*: Very few approaches are included formalization of requirements objects and related models [see Table I]. Though UML based techniques are still popular but UML itself is not formal in nature. Thus validation of UML based requirements analysis models are complex to realize. Major research trials are required to formalize the available requirements engineering frameworks along with the formal validation mechanism towards perceiving and realizing the domain concepts more formally along with.

(b) *Prototype CASE tools*: Very few models are supported with the CASE tools related to the requirements engineering frameworks [see Table II]. Moreover, those proposals also are at prototype levels. Further tool level researches are required towards the proposal for full-fledged CASE tools with the support of requirements specification based on object oriented paradigm, modeling of the requirements objects for analysis, validation of requirements analysis models etc. Moreover the research can be extends towards automatic generation of top level design models from the requirements objects and related models.

(c) *Analysis of non-functional requirements*: Majority of the proposed frameworks are useful for specifications analysis of functional requirements of the system. But as the software system in this days are becoming large and complex, research efforts are required to extend the requirements engineering

frameworks towards the specification, representation and analysis of non-functional requirements of the system as well.

(d) *Traceability towards design model*: Requirements analysis models are the key input for the design of software system. For the purpose, a systematic transformation mechanism is required for mapping the requirements objects and related models into the equivalent design model of the target system. The mechanism also needs to be capable enough of tracing the element wise mapping and validating the correctness of the transformation. The BORA framework [11] supports these partially. However, major research effort is required to achieve this goal in requirements engineering framework.

(e) *User-centric Requirements Engineering Framework*: On the process of requirements analysis very often a communication gaps are created between the software analyst and the stakeholders. Any requirements engineering framework must be supported with well-defined guidelines towards the establishment of communication among the analyst and the stakeholders at all the levels of requirements engineering process (in both early and detailed requirements analysis phase). This fact demands more research towards the user-centric requirements engineering frameworks.

(f) *Quality evaluation of requirements engineering process*: Efficient software design and subsequently efficient software product development depends on the quality of the requirements engineering process. It is evident from the previous researches that effective requirements engineering process can reduce the risk of failure for development of large scale information system. Very few available frameworks are included with quality evaluation schemes [See table III]. But those schemes are specific to the related frameworks and also not very comprehensive. An extensive list of quality factors has been listed in section 3.5 which are essential to measure the quality of requirements engineering process. Hence potential research agendas exist towards devising a general purpose quality evaluation framework for the requirements engineering process with the objective to assess the qualities of the requirements objects and related analysis models.

6. CONCLUSION

In this paper an evaluative study has been performed for the object oriented requirements engineering frameworks. Several such existing frameworks have been briefly described with their relative advantages and disadvantages. An extensive list of features of “good” requirements engineering framework for object oriented software analysis also has been proposed and described in this paper. The features are divided in different levels related to the requirements engineering process and frameworks for better clarification. While describing the features emphasize has been given on quality factors for assessments of the requirements and related analysis models. Further, a detailed comparative study based on those features has been done for the available requirements engineering frameworks. An interesting result has been found that many of the available frameworks do not provide any well-defined mechanism for detailed requirements analysis. Based on the result of comparative study, several potential research agendas also have been raised towards the broad area of requirements engineering process.

7. REFERENCES

- [1] A. V. Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour", 5th IEEE Intl. Symposium on Requirements Engineering, pp. 249-263, 2001.

- [2] Arao T., Goto E., Nagata, T., "Business process oriented requirements engineering process", 13th IEEE Intl. Conf. on Requirements Engineering, pp. 395 - 399, 2005.
- [3] Pichler M., Rumetshofer, H., "Business Process-based Requirements Modeling and Management", 1st Intl. Workshop on Requirements Engineering Visualization, pp: 6, 2006.
- [4] Joseph E. Kasser, "Object-Oriented Requirements Engineering and Management", Systems Engineering Test and Evaluation (SETE) Conference, 2003.
- [5] Artur Caetano, Antonio Ritó Silva, José Tribolet, "Using roles and business objects to model and understand business processes", ACM Symposium on Applied computing, pp 1308-1313, 2005.
- [6] OMG, Business Object DTF – Common Business Objects. OMG Document bom/97-11-11, ftp://ftp.omg.org/pub/docs/bom/97-11-11.pdf, 1997.
- [7] H. Kaindl, "A practical approach to combining requirements definition and object-oriented analysis", Annals of Software Engg., Vol. 3, pp 319–343, 1997.
- [8] Vidgen, R., "Requirements analysis and UML use cases and class diagrams", IET Journal of Computing & Control Engineering, Vol. 14(2), pp 12 - 17, 2003.
- [9] Vidgen, R., "Requirements analysis and UML - Interaction diagrams and state transition diagrams", IET Journal of Computing & Control Engineering, Vol. 14(3), pp 7 - 11, 2003.
- [10] J. Mylopoulos, L. Chung, E. Yu, "From object-oriented to goal-oriented requirements analysis", Comm. of the ACM, Vol. 42(1), pp. 31-37, 1999.
- [11] Z. Gan, D. Wei, J. Zhang, V. Varadharajan, "Business-process-oriented software requirements automatic generator", 3rd International Conference on Information Technology and Applications, Vol.1, pp: 95 - 98, 2005.
- [12] Liegl P., Schuster R., Zapletal M., Huemer C., Werthner H., Aigner M., Bernauer M., Klinger B., Mayr M., Mizani R., Windisch M., "[vem:xi:] - A Methodology for Process Based Requirements Engineering", 17th IEEE Intl. Requirements Engineering Conf., pp.193-202, 2009.
- [13] Cardoso E.C.S., Almeida J.P.A., Guizzardi G., "Requirements engineering based on business process models: A case study", 13th Workshop on Enterprise Distributed Object Computing Conf., pp.320 - 327, 2009.
- [14] Gaur V., Soni A., Bedi P., "An agent-oriented approach to requirements engineering", IEEE 2nd Intl. Advance Computing Conference (IACC), pp: 449 - 454, 2010.
- [15] Lu C., Chu W.C., Chang C., Wang C. H., "A Model-based Object-oriented Approach to Requirement Engineering (MORE)", 31st Annual Intl. Computer Software and Applications Conf., (COMPSAC 2007), Vol. 1, pp: 153 - 156, 2007.
- [16] Anwer S., Ikram N., "Goal Oriented Requirement Engineering: A Critical Study of Techniques", 13th Asia Pacific Software Engineering Conf., pp. 121 - 130, 2006.
- [17] Kaiya H., Horai H., Saeki M., "AGORA: attributed goal-oriented requirements analysis method", IEEE Joint Intl. Conf. on Requirements Engineering, pp. 13 - 22, 2002.
- [18] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, A. Spencer Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study", Tech. Report, Software Engg. Institute, Carnegie Mellon Univ. (USA), 1990.
- [19] H. Kaindl, "Is object-oriented requirements engineering of interest?", Journal OF Requirements Engineering (Springer-Verlag), Vol. 10(1), pp. 81-84, 2005.
- [20] Y. Yang, F. Xia, W. Zhang, X. Xiao, Y. Li, X. Li, "Towards Semantic Requirement Engineering", IEEE International Workshop on Semantic Computing and Systems, pp: 67 - 71, 2008.
- [21] A. Sarkar, S. Choudhury, N. Chaki, S. Bhattacharya, "Business-Object Oriented Requirements Analysis Framework for Data Warehouses", 22nd Intl. Conf. on Software Engineering and Knowledge Engineering (SEKE 2010), pp: 34 – 37, 2010.
- [22] A. Sarkar, N. C. Debnath, "Business Object Oriented Requirements Analysis for Large Scale Information System", 20th Intl. Conf. on Software Engineering and Data Engineering (SEDE 2011), PP 103 – 108, 2011.
- [23] ISO/IEC 9126-1, *Software engineering - Product quality - Part 1: Quality model*. International Standard Organization, 2001.