ISSN:2320-0790

An International Journal of Advanced Computer Technology

A Survey of Various Object Oriented Requirement Engineering Methods

Anandi Mahajan[1], Dr. Anurag Dixit[2]

[1]Research Scholar, Mewar University, Rajasthan, India
[2]Research Supervisor, Mewar University, Rajasthan, India

Abstract: In current years many industries have been moving to the use of object-oriented methods for the development of large scale information systems The requirement of Object Oriented approach in the development of software systems is increasing day by day. This paper is basically a survey paper on various Object-oriented requirement engineering methods. This paper contains a summary of the available Object-oriented requirement engineering methods with their relative advantages and disadvantages. On next, several features have been proposed. These features are the characteristics of a good model of Object-oriented requirement engineering.

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

I. 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.

II. 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.

A. 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.

B. 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.

C. 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.

D. 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].

E. 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.

F. 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.

III. CHARACTERSTICS OF A GOOD OBJECT ORIENTED REQUIREMENTS ENGINEERING MODEL:

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

(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

*(i) Identification of Requirements Objects***:** Requirements are identified with their context and mapped in requirements objects.

(j) 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.

(k) 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.

(l) 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.

IV. FUTURE RESEARCH DIRECTIONS

Various potential research agendas still exist in the area of requirements engineering. 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) *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.

(c) *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.

V. 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. Based on the result of literature survey, several potential research agendas also have been raised towards the broad area of requirements engineering process.

VI. 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.