Goal Oriented Requirements Engineering for Non-Functional Factors

D. Francis Xavier Christopher
Director, School of Computer Studies
RVS College of Arts and Science
Coimbatore, India

E. Chandra
Phd, Director, School of Computer Studies
SNS Rajalakshmi College of Arts and Science
Coimbatore, India

ABSTRACT

Requirements phase is the most important stage in the Software Development Process. If the requirements have not been captured correctly, the whole development process will fail and will results in time and monetary costs. The Goal Oriented Requirements Engineering (GORE) approach helps in defining, eliciting, organizing, analyzing and refining the requirements, so that the requirements can meet the customer needs. This paper discusses about the application of Goal-Oriented Requirements engineering for eliciting the scalability, analyzing the Reliability requirements and eliciting and analyzing the security Requirements

Keywords

Requirements Engineering, Goal-Oriented Requirements Engineering (GORE) Approach, Non-Functional Quality Factors, Scalability, Reliability, Security.

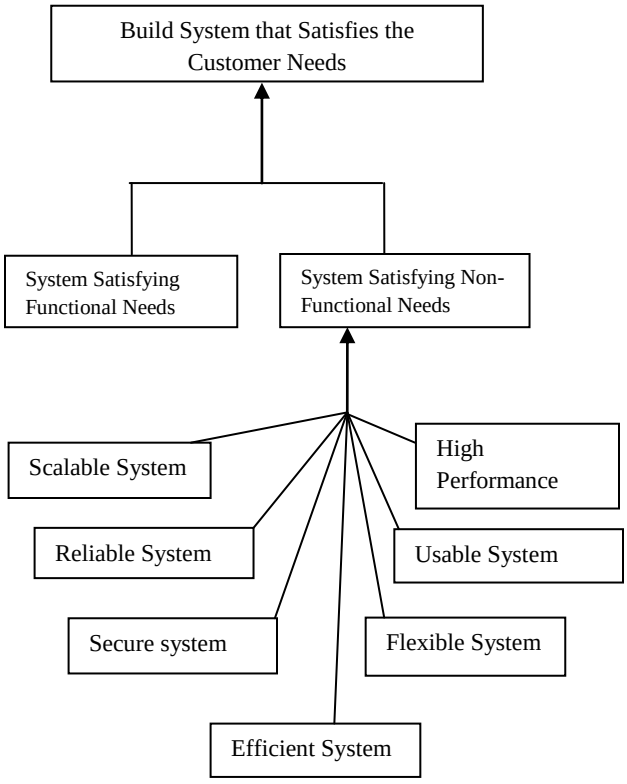
1. INTRODUCTION

Requirements Elicitation is the most important stage in the Software Development Life Cycle Process. If the requirements have not been captured correctly during the initial elicitation process, then the whole development process will fail and also results in time and monetary costs. Therefore the generation of an accurate requirements model is highly necessary for any kind of system. Hence the proper selection of Requirements Engineering Technique becomes a challenging work [13].

The Goal Oriented Requirements Engineering (GORE) [1] [4] [5] approach has been found appropriate for Eliciting, Defining, Analyzing and Refining the Requirements. The Goal-oriented requirements engineering (GORE) also supports a natural elicitation of software requirements in the context of high-level goals. Goals are the description of the customer needs, specified as the properties that the system/application must satisfy. The goal of the system is to build a system that satisfies all stakeholders' needs: functional and non-functional ones. Functional Requirements associated with the services/functionalities to be delivered to stakeholders. Non-Functional Requirements associated with the factors like Security, Performance, Flexibility, Reliability, Usability, Scalability and Efficiency and so on also known as quality factors [1].

This is clearly depicted in Fig.1.

Fig.1 Generic Goal Requirements View



The paper discuss about the background on Goal-Oriented Requirements Engineering approach and application of Goal-Oriented Requirements Engineering for eliciting the scalability, analyzing the reliability and eliciting and analyzing the security requirements

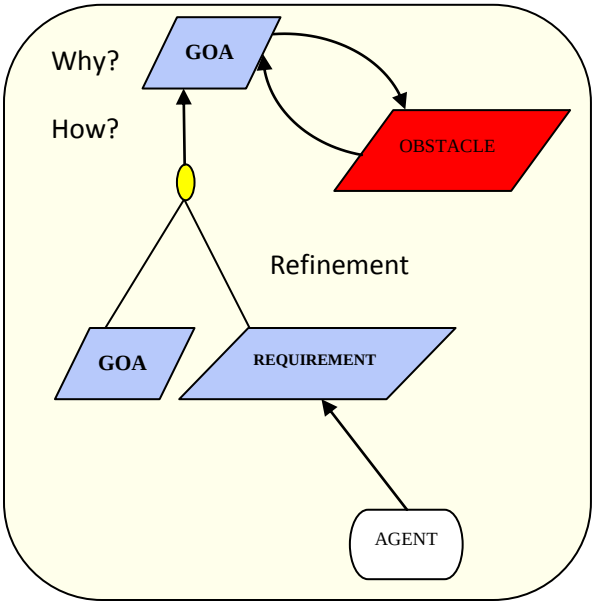
2. BACKGROUND ON GOAL ORIENTED REQUIREMENTS ENGINEERING

Requirements engineering involves requirements gathering, elicitation, analysis and specification of the requirements of a

system [1]. The requirements gathered during elicitation serves as a foundation for all the phases in software life cycle. Hence, clear understanding of these requirements helps in assessing and managing the requirements during development process. Goal-oriented requirements engineering methodologies, such as NFR, KAOS, i*/Tropos and GBRAM [8] [9] [11] focus on justifying why a system is needed through the specification of its high-level goals. These goals drive the requirements elaboration process, which results in the definition of proper requirements that can be implemented by the system under development.

Goal-oriented requirements engineering views the system-to-be and its environment as a collection of active components called *agents* [1]. Active components may restrict their behavior due to the constraints. These components are humans playing certain roles, devices, and software. In GORE, agents are assigned responsibility for achieving goals. There is a possibility that while achieving the goal, there occur some obstacle. This obstacle can be overcome by taking necessary actions and finding the way to attain that goal. This is clearly depicted in Fig 2 Goal Modeling Diagram.

Fig2. Goal Modeling Diagram [1]



A Goal is a prescriptive statement of intent about some (existing or to-be) system whose satisfaction generally requires the cooperation of some of the agents that constitute the system. A Requirement is a goal whose achievement is the responsibility of a single software agent [1] [11].

The following are the benefits for Goal Modeling [1]

1) Expectations for the Goal are captured during the early requirement phase.
2) The specifications of requirements are complete with respect to the set of defined goals.
3) A requirement is pertinent with respect to set of goals and hence a goal provides clear requirement pertinence.
4) A goal refinement provides the traceability links from high-level requirements to low-level requirements.
5) Goal Modeling provides a natural mechanism for structuring complex requirements documents. Also helps to communicate the requirements to the customers efficiently
6) A goal helps in identifying and managing the conflicts among the requirements.

3. GORE APPROACH FOR SCALABILITY

Scalability has been widely recognized as an important non-functional factor. It is necessary to undertake a clear, consistent and systematic exploration of the system's scalability goals in the application domain. The lack of techniques to elicit scalability requirements leads to number of problems.

Define the Scalability as "Ability of a system to maintain the satisfaction of its quality goals to levels that are acceptable to its stakeholders when characteristics of the execution environment ("the world") and design ("the machine") vary over expected ranges" [7].

It is apparent that it is necessary to derive the variables, scaling ranges, bounds on the ranges and characteristic functions in a clear, consistent and systematic manner from system goals in order to increase the precision and usefulness of scalability analysis results.

The following are the advantages of using Gore Technique for Scalability [7]:

1. Scalability of the application domain based on assumptions:

The scalability of the application domain should depend not only on the assumptions made during the current system environment but also for the future system environment. Making such assumptions is essential for scalability.

2. Provides rationale for requirements:

Lack of foundation in scalability requirements may results in unpredictable errors on the system design. These errors can be avoided by defining the proper scalability. Goal models helps in assessing the completeness of scalability requirements with respect to the defined goals.

3. Provides traceability:

Traceability is important in the context of scalability. If the assumptions made on the ranges of the application domain, in the future are found to be incorrect or no longer valid, then they can be traced easily to the system requirements and design.

4. Assignment of responsibilities:

Assign goals to respective agents. GORE can establish the responsibility for scalability as the agents takes the responsibility in achieving the goals.

5. Measurable quality variables and objective functions:

By using scalability factor, the untested boundary values can be identified and helps in measuring the quality variables and achieving the objective function i.e., to deliver the high quality software to the customer.

The following are the difficulties in applying GORE for eliciting requirements [7].

1) Handling ranges and thresholds in the application domain.
2) No clear taxonomy for the assumptions on the application domain
3) Handling conflicts on the range values

This paper reviews number of well-recognized advantages of GORE which are particularly useful for scalability. However, there is a lack of techniques for elaborating goal models with respect to scalability goals. In future, applying different models and techniques will help to overcome the difficulties in scalability.

4. GORE APPROACH FOR RELIABILITY

Reliability is one of the most important criteria for the Non-Functional quality factor. IEEE STD 982.2 states "A software reliability management program requires the establishment of a balanced set of user quality objectives, and identification of intermediate quality objectives that will assist in achieving the user quality objectives."[6]

The main goal for Reliability is to build high quality reliability software which depends on the application/system quality attributes in the requirements phase. The business requirements are gathered from the business analysts/clients which are the basic building blocks for the development process upon which the entire application is built. Hence requirement validation has to be done in order to ensure that the functionality delivered correctly [12]. However, when there is any misunderstanding between the developer and the client, then the requirements delivered are not satisfied, results in the accepting changes to the existing functionality. The better approach is to get the requirements complete, concise and clear. This provides a clear picture for the developer to build the system without any misunderstanding between the Business Analyst, Developer, Quality Assurance team and the client [6].

The Requirement Reliability Metric is based on the requirements specification and keeps track of the requirements in scope of the project [6] [12]

This metric make sure that the project is staying on track by counting the number of requirements that plan to implement during the current release and tracking their status during construction. This would help in focus on meeting the client's needs and delivery targets by monitoring requirements change requests. Further helps in estimating future projects better by tracking the time spent on requirements engineering and correlating requirements definitions with development effort. This can be achieved by applying Gore Approach as mentioned in below points.

- Specify a set of goals based on the needs of the organization, project and client.
- Generate a set of quantifiable questions to achieve the specified goal/target in time.
- Define the set of measures that provide the quantitative information needed to answer the questions.

More number of questions and Measures can be identified and formulated based on the organizational needs and client's needs to achieve the target i.e., to achieve the high quality reliable software. First step is to identify the goals and generate the quantifiable questions to achieve the goal and then identify the metrics to answer the questions. This helps for the project manager to keep track of the status of the requirements. Given below the sample set of questions and Measures to attain the Requirements Reliability [6] [12].

Questions	Measures
1. What is the current status of each requirement	Status of each requirement in the current phase
2. Is the requirements are stable	Number of Initial Requirements Number of Changes per Requirement Number of Changes per Requirement per Release Total Number of Changes in the release Number of Final Requirements Number of requirements added Number of requirements modified Number of requirements rejected
3. Is the requirements are feasible for the current development technology	Number of Requirements identified as feasible Number of Requirements identified as infeasible Total Number of Requirements
4. Has the Effort Estimation have been documented for each requirement	Number of Hours estimated per Requirement. Actual Number of Hours spent per requirement Type of Documentation

5. Has the Scheduling Changes for the requirement have been documented	Initial Number of hours Scheduled per Phase Number of Hours Scheduled per Phase after changes Actual Number of Hours Spent per Phase
6. How many other requirements are affected by the change	Number of requirements affected by change due to dependency relation
Questions	**Measures**
7. How many incomplete requirements have been identified	Number of Incomplete Requirements
8. How many identified incomplete requirements made to be complete	Number of Complete Requirements which are incomplete before
9. Has the all the requirements are documented including change, incomplete and so on	Type of Documentation
10. How many missing requirements have been identified	Number of Missing Requirements
11. Does the product developed has satisfy all the customer needs	Functionality of the software Number of Initial Requirements Number of Final Requirements Number of Tests per Requirement Number of Passed Tests per Requirement
12. Has all the Change proposals have been approved by Change Control Board(CCB)	Total Number of Requirement Changes Number of Requirement changes proposed by developer Number of Requirement changes proposed by client/System Analysts Number of Requirement Changes Approved and Rejected by CCB per release
13. Has the test cases have been documented for requirements	Number of Test Cases written per Requirement Number of Test Case changed per Requirement Type of Documentation
14. Has the communication between client and developer regarding requirements clarification has been documented	Type of Requirement Clarification Document

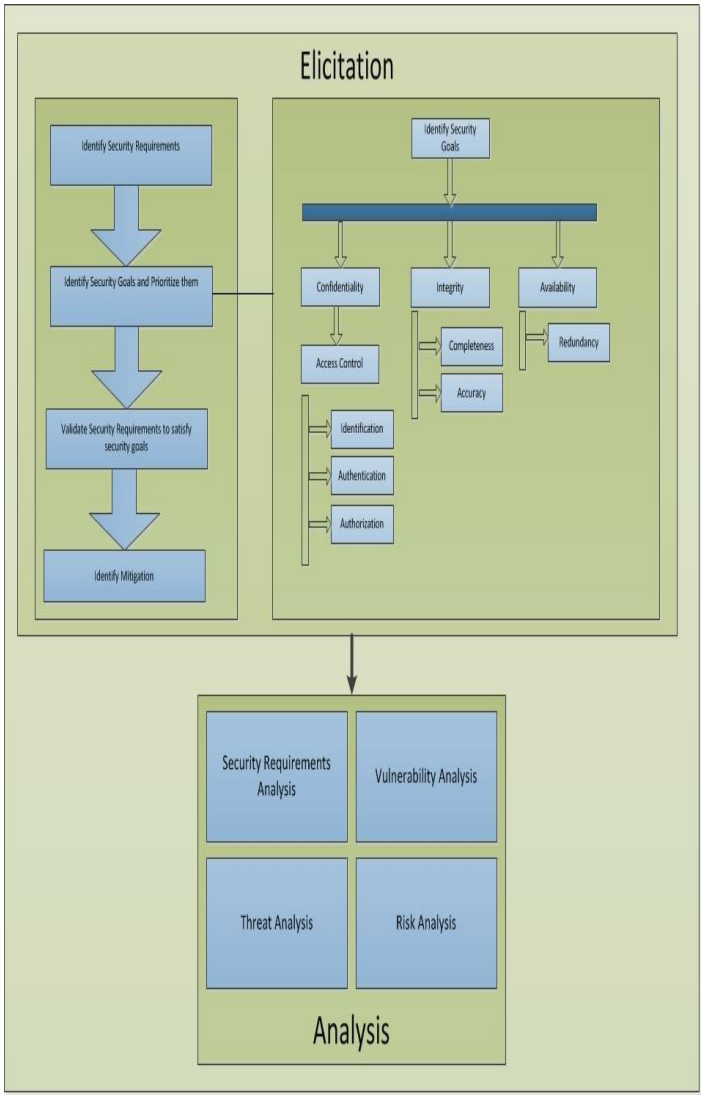
The following are the advantages of applying GORE technique for Reliability:

1) Ease of Use.
2) Validate and verify the requirements if it meets the customer needs.
3) Completeness of requirements.
4) Setting the goal and identifying the path to achieve the target.
5) Calculating the reliability of the software.

5. GORE APPROACH FOR SECURITY

Security is an integral part of the Software Development Life Cycle. Security Requirements Engineering deals with the protection of assets from malicious attacks that may harm the security of the system. GORE approach to security requirements concentrates on identifying potential obstacles for satisfying security goals. The security constraints should be identified and defined at the application level during the early development phase itself for protecting the system from harm. The Security Requirements Model consists of two phases: Elicitation phase and Analysis Phase [2] [3].

Fig3. Goal Oriented Security Requirements Model



In Elicitation Phase, the steps to be followed [3]:

- Identify Security Requirements based on the needs of the clients/organization
- Identify the Security Goals(Confidentiality, Integrity, Availability and so on) based on the Security Requirements and prioritize them
- Validate the Security Requirements for satisfying the security Goals
- Identify the Mitigation

In Analysis Phase, Security Requirements Analysis, Vulnerability Analysis, Threat Analysis and Risk Analysis have to be done to achieve the security Goals that are defined during the Elicitation Phase [3]. The Security Requirements Analysis explores and evaluates the security requirements based on the threats, vulnerability and the risks. A threat is simply an event that may have the negative impact of the system. Threat Analysis describes the potential attacks on the system and helps in making critical decisions and countermeasures for avoiding threats. Vulnerability Analysis describes the weakness of an asset, error in the specification which can exploits the system's security. Risk analysis determines the chances that the threats, vulnerability will occur and helps in assessing their impact and the risk [3].

This can be clearly depicted in Fig3. Goal Oriented Security Requirements Model.

The following are the benefits of using Goal Oriented Security Requirements Modeling:

1) Identifying the Security Goals for each Security Requirements at the early development phase which helps in attaining the high quality of the system.
2) For every threat T identified in the system, there will exists a mitigation or countermeasure M that helps in neutralizing the threats and vulnerabilities.
3) It helps to build a powerful and secure system based on the client's needs.
4) It helps in making critical decisions which protects the system from malicious attacks.

6. CONCLUSION AND FUTURE WORK

The paper discusses the importance of GORE approach in early Requirements phase. This paper reviews number of well-recognized advantages of Goal Oriented Requirements Engineering approach particularly useful for the Non-Functionality factors such as Scalability, Reliability and Security. By applying the GORE Approach, goals that have been defined based on the organization/client needs can be achieved which in turn achieves high quality software.

In future, the research concentrates on the other Non-Functional factors which help in improving the quality of software. Many software quality models such as ISO/IEC 9126 Quality Model, McCall's model, Boehm's Model exists in the real world for quantifying the Non-functional factors. Software quality factors needs to be quantified and measured so that the quality of the software can be measured. This can be done by formulating a mathematical model in organizing all the Non-Functional Factors based on Goal Oriented Requirements Engineering Approach.

REFERENCES

[1] Alexei Lapouchnian, "Goal-Oriented Requirements Engineering: An Overview of the Current Research" Depth Report, University of Toronto, 2005

[2] Bashar Nuseibeh, Charles B. Haley, Craig Foster, "Securing the Skies: In Requirements we trust", IEEE proceedings, September 2009

[3] Hui Wang, Shulan Gao, Bibo Lu, Zihao Shen, "A Novel Model for Security Requirements Process" ISECS '10, Guangzhou, China, pp 29-33, July 2010.

[4] Huzam Saud Faleh Al-Subaie " Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Method", Research thesis submitted to the University of London, 2007

[5] Jennifer Horkoff and Eric Yu, "Analyzing Goal Models – Different Approaches and How to Choose Among Them" SAC'11 Proceedings of the 2011 ACM Symposium on Applied computing, pp 675-682, Mar. 2011.

[6] Dr. Linda Rosenburg, Ted Hammer, Jack Shaw, "Software Metrics and Reliability", ISSRE, November 1998.

[7] Leticia Duboc, Emmanuel Letier, David S. Rosenblum, Tony Wicks, "A Case Study in Eliciting Scalability Requirements", International Requirements Engineering, 2008, 16th IEEE, pp 247-252, Sep. 2008.

[8] Miguel A. Teruel, Elena Navarro, Victor Lopez-Jaquero, Francisco Montero, Pascual Gonzalez, " A Comparative of Goal-Oriented Engineering Approached to Modelling Requirements for Collaborative Systems", 6th International Conference on ENASE'11, Springer, pp 131-142, March 2011

[9] Miguel A. Teruel, Elena Navarro, Victor Lopez-Jaquero, Francisco Montero, Pascual Gonzalez, "Assessing the Understandability of Collaborative Systems Requirements Notations: An Empirical Study", Empirical Requirements Engineering, 2011 First International Workshop on IEEE proceedings, pp 85-92, Aug. 2011.

[10] Miguel A. Teruel, Elena Navarro, Victor Lopez-Jaquero, Francisco Montero, Pascual Gonzalez, "CSRML: A Goal Oriented Approach to Model Requirements for Collaborative System", Proceeding ER'11 Proceedings of the 30th international conference on Conceptual modeling, Springer, pp 33-46, 2011.

[11] Ponsard, Christophe and Devroey, Xavier, "Generating High-Level Event-B System Models from KAOS Requirements Models" InforSID 2011, May 2011.

[12] Rohit Gupta, "Software Reliability Metric Taxonomy", TECHNIA – International Journal of Computing Science and Communication Technologies, Vol 3, No.2, Jan 2011.

[13] Dr. Sohail Asghar, Mahrukh Umar, "Requirements Engineering Challenges in Development of Software Applications and selection of Customer-off-the-Shelf (COTS) Components", International Journal of Software Engineering (IJSE), Volume 1, Issue 2, 2010.

AUTHOR'S PROFILE

D.Francis Xavier Christopher received his B.Sc., in 1996, M.Sc., in 1998 from Bharathiar University; Coimbatore .He obtained his M.Phil, in the area of Networking from Bharathiar University, Coimbatore in 2002. At present he is working as a Director, School of Computer Studies in RVS College of Arts and Science, Coimbatore. He has published more than 08 research papers in National, International journals and conferences. He has guided for more than 40 M.Phil, scholars. His research interest lies in the area of Software Engineering.

Dr. E.Chandra received her B.Sc., from Bharathiar University, Coimbatore in 1992 and received M.Sc., from Avinashilingam University, Coimbatore in 1994. She obtained her M.Phil., in the area of Neural Networks from Bharathiar University, in 1999. She obtained her PhD degree in the area of Speech recognition system from Alagappa University Karikudi in 2007. At present she is working as a Director, School of Computer Studies, in SNS Rajalakshmi College of Arts and Science, Coimbatore. She has published more than 20 research papers in National, International journals and conferences. She has guided for more than 30 M.Phil, scholars. At present, she is guiding 5 PhD research scholars. Her research interest lies in the area of Data Mining, Artificial intelligence, neural networks, speech recognition systems and fuzzy logic. She is an active member of CSI, Currently management committee member of CSI, Life member of Society of Statistics and Computer Applications.