GBRAM-BASED MODEL FOR GOAL ORIENTED REQUIREMENT ENGINEERING: A UNIFYING FRAME WORK

[1]Dr. Shaik Subhani, [2]A. Ravikumar
[1]Associate Professor, [2]Assistant Professor
Department of Information Technology
Sreenidhi Institute of Science and Technology (Autonomous), Hyderabad.

Abstract: Goals are a logical system for identifying, organizing and mitigating software requirements. The word Goal is ever more being used in Requirement Engineering. Over the past ten years, the requirements engineering society has ever more extended its acceptance and adaptation of goal-oriented approaches to both functional and non-functional requirements? Goal-Oriented requirement engineering gives an advanced method for elicitation, investigation, clarification & modification, pattern and modeling of requirements. Different GORE methods exist for these requirement engineering processes based on confident underlying concepts and principles. Our literature review, we recognize that existing goal oriented necessities elicitation computation do not sustain to prioritize the supplies when the stakeholders view are often unclear and contain ambiguity. Therefore, in this paper, we obtainable Goal-Based Requirements Analysis Method for requirement engineering. The objective of GBRAM is to give practical direction towards the recognition and analysis of the future directorial goals that decide system requirements in the background of software systems growth. We talk about goals from the viewpoint of two themes: goal scrutiny and goal progression. Finally, our proposed method provides appropriate representation mechanisms to enhance stakeholder comprehension and facilitate communication between analysts and stakeholders.

Index Terms: Requirement Engineering, GORE, GBRAM, Goal Evaluation, Goal Analysis.

1. INTRODUCTION

The major determine of the achievement of a software system is the degree to which it reach its point. Therefore, identifying this purpose must be one of the main activities in the growth of software systems. It has been extended recognized that inadequate, incomplete, ambiguous requirements have a important impact on the excellence of software. Thus, Requirements Engineering, a stem of software engineering that handle with elicitation, modification, analysis, etc. of software system necessities gained a set of concentration in the academic world as well as in the commerce. Van Lamsweerde et.al presents the following intertwined functions that are enclosed by requirements engineering:

- **Domain analysis:** the environment for the system-to-be is studied. The relevant stakeholders are identified and interviewed. Issues with the current system are discovered and opportunities for improvement are investigated.
- **Elicitation:** substitute models for the goal system are analyzed to meet up the recognized objectives. Requirements and guesses on components of such models are recognized.
- **Negotiation and agreement:** substitute requirements and guesses are evaluated; risks are analyzed by the stakeholders by the most excellent alternatives are chosen.
- **Specification:** requirements and assumptions are formed carefully.
- **Specification analysis:** these are tartan for issues such as incompleteness, irregularity and feasibility.
- **Documentation:** different thinking's made during the requirements engineering procedure are documented jointly with the underlying logically and guesses.
- **Evolution:** requirements are customized to provide somewhere to stay corrections, environmental alters, or novel objectives.

The principle of goal-based methods is to concentrate on why systems are made, which gives the inspiration and rationale to good software requirements. Other advantages include: (1) serving to obtain requirements by elaborating what requirements are desirable to maintain the goals; (2) building simple justification and clarification of the occurrence of requirements in a expanded way by initial from system-level and organizational objectives from which such lower level descriptions are increasingly derived (3) providing the data for recognizing and resolving conflicts that come up from multiple viewpoints amongst agents.

[1]The fame of goal-oriented requirements engineering methods has enlarged radically. The major reason for this is the inadequacy of the traditional systems analysis approaches when dealing with more and more complex software systems. [2]. Goal-Oriented Requirements Engineering (GORE) attempts to solve these and other important problems. GORE concentrates on the actions that lead the formulation of software system requirements. [3] In this paper, we proposed GBRAM model for goal-oriented requirement engineering process (as shown in figure 1).

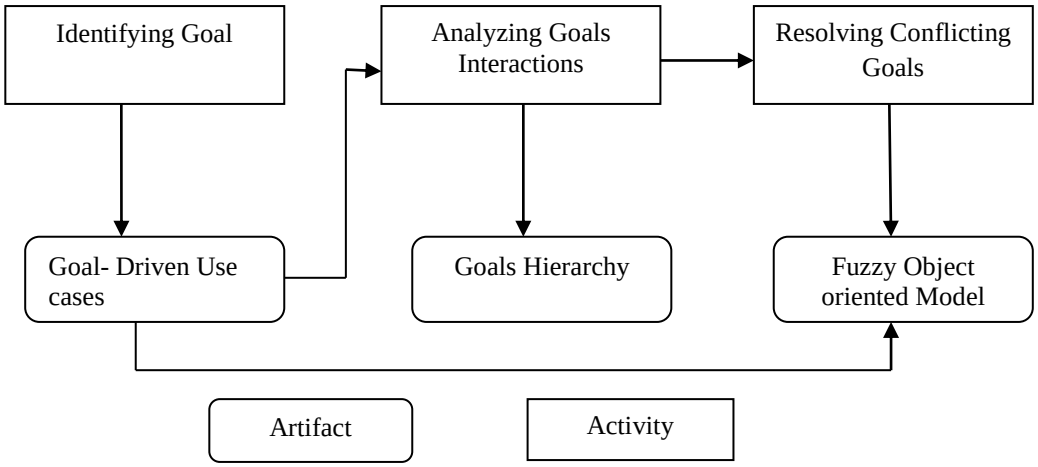

Figure 1 Proposed GBRAM Approach

This paper is organized as follows: Section 2 presents the background work. The concepts and process elements of our goal- based approach is discussed in section 3. In section 4, we present GBRAM based approach for requirement engineering process. Analysis and discussion is presented in section 5. Finally, conclusions are drawn in section 6.

2. BACKGROUND WORK

Although there has been relatively little attention paid to the process of acquiring goals for system development. Traditional systems analysis focuses on what features (i.e. activities and entities) a system will support. Subsequently, more number of researchers has explained about progress toward the development of goal-based methods. In particular, Dardenne et al. have presented a goal-directed procedure for model acquisition. Mylopoulos et al. have stated a structure for representing nonfunctional requirements in terms of goals, which can be evaluated in order to determine the degree to which a nonfunctional constraint is supported by specific design. Meanwhile, Anton has proposed a goal-based requirement analysis method to recognize, elaborate and refine goals for requirements specifications. Rather than using scenarios to concretize goals. The basic modeling component is a combination of goal and scenario where the scenario is authored for the goal. Finkelstein et. al. propose the use of viewpoints as both an organizing and a structuring principle in software development. Yu's strategic dependency model provides the rationale for networks of actors in which agents depend on each other to achieve goals, perform tasks and furnish resources. The strategic dependency model supports the process of suggesting, exploring and evaluating alternative solutions and helps identify what is at stake, for whom, and what impacts are likely if a dependency fails. [4] Lai et al. proposed a fuzzy based method to rank the customer necessities in a competitive situation.

3. PROCESS ELEMENTS

Goals are useful for organizing and justifying software requirements, but two practical questions need to be addressed: 1) how are goals identified? 2) What happens to the requirements when goals change?. In this paper we discuss goals from the perspective of these two themes: goal analysis and goal evolution. Goal analysis concerns the exploration of documentation followed by the organization and classification of goals. Goal evolution concerns the way goals change from the moment they are first identified to the moment they are operational in a system specification. The concepts and process elements of our goal- based approach are discussed in the following section.

3.1 Goal Analysis

Goals may be extracted from various types of gathered information including process descriptions such as flow charts or Entity Relationship (ER) diagrams. It is useful to identify goals from process descriptions by searching for statements which seem to guide design decisions at various levels within a system or organization. When used as the exclusive source of information, process descriptions are insufficient for achieving thoroughness and completeness. Therefore, if possible, the practitioner should consider other possible sources such as transcripts of interviews with stakeholders. However, stakeholders tend to express their requirements in terms of operations and action rather than goals. Thus, searching for action words is a useful way to extract goals from stakeholder descriptions. For example, in a meeting scheduler, stakeholders may use action words such as 'schedule' and 'reserve' which give rise to goals such as: Schedule Meeting and Reserve Room [5].

In accumulation to goals, the agents, stakeholders and constraints must also be recognized. The most sensible method is to recognize the agents as early as possible by determining what agents are ultimately responsible for the achievement or maintenance of a goal. For example, the goal Schedule Meeting is the responsibility of the Meeting Scheduler. Constraints are helpful because they offer additional data concerning requirements that must be met in order for a specified goal to be finished. As a universal rule, we recognize constraints by searching for temporal connectives, such as during, before and after, or any variants thereof. Constraints may also be recognized by looking for dependency relations. Consider the goal Meeting prearranged in the scheduler system with the constraint: Meeting room must be obtainable during the date/time. Once the goals, agent responsibilities and stakeholders are identified and specified, the goals are then classified according to their target conditions and begin to evolve [5].

3.2 Goal Evolution

Goal evolution is affect via goal elaboration and refinement. Useful methods for goal elaboration are: identifying goal obstacles, analyzing scenarios and constraints, and operational goals. Recognizing goal obstacles, in order to believe the possible ways for goals to be unsuccessful, enables one to expect exception cases. When goal priorities modify, scenarios make possible the evaluation of these novel priorities. Goals are additional elaborated by considering the possible ways in which goals can be infertile and by recognizing scenarios to develop an understanding of how the goals can be operational [5]. In the GBRAM, achievement goals are compound and listed according to their priority relations and dependencies. This ordering enables us to decide a goal's pre and post conditions. It is helpful to consider goal precedence relations such as our method differs from Yu's model in that dependency relations are used primarily to order goals so they can be consequently refined. We expect that further consideration of goal and agent dependency relations will yield deeper insights for conflict resolution but as yet we have not addressed this.

Goals are sophisticated by eliminating redundancies and integration identical goals. For example, the goals Meeting arranged and scheduled are synonymous and can be integrated. In our experience, the best procedure is to eliminate redundancies after the goals have been merged into one ordered goal set. It is then easier to identify synonymous goals because they typically are listed adjacent to each other in the ordered set since they tend to share ordinary precedence relations. Goals are also modified via elaboration. The operational goals, responsible agents, stakeholders, constraints and scenarios are ultimately consolidated into a set of goal schemas that can be easily translated into a requirements specification. The outcome artifact, while not formal in the strict sense, provides a textual representation of system requirements organized according to system goals.

4. GBRAM-BASED FRAME WORK

The objective of GBRAM is to give practical support towards the recognition and analysis of the future organizational goals that determine system requirements in the framework of software systems growth. Therefore, GBRAM concerns the shortly stages of RE in that require for alter has been analyzed and the choice to expand a number of software application has been completed.

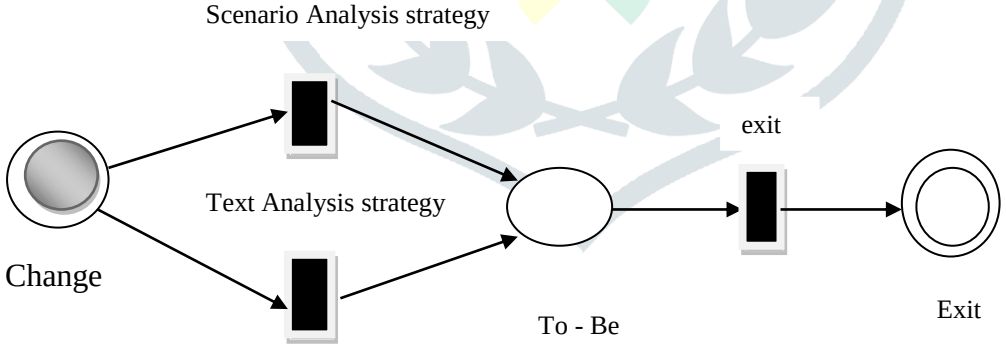

Figure 2 the GBRAM way-of-working

The GBRAM way-of-working is shown in Figure 2. Discovering enterprise goals with respect to the future system is based on guided analysis of existing documents. This meth is termed text analysis strategy in Figure 1. Such documents may explain enterprise policies, requirement stipulation of in sequence systems as well as transcripts of interviews with stakeholders. The recognition of goals is guided by heuristic rules and questions. In addition, GBRAM suggests a situation analysis strategy based on analysis of difficult scenarios that explain the circumstances in which a goal may fail or blocked, thus leading to the design of solutions that determine these problems. Scenarios are useful means for linking with stakeholders, offering a natural way to demonstrate how user requirements may be in a future situation. GBRAM include the following activities: goal analysis and goal refinement. The different activities of GBRAM technique are shown in the following figure 2.

Goal analysis is about the information sources for goal recognition followed by organization and classification of goals. This action is further separated into explore activities that explore the obtainable data, recognize activities about extracting goals and their responsible agents from the information and manage activities that classify and arrange the goals according to goal dependence relations. The goal analysis activities can be listed below as follows:

- Explore activities require the examination of the inputs.
- Recognize activities require extracting goals and their accountable agents from the available documentation.
- Systematize functions involve the categorization of goals and arrangement of those goals according to goal dependency relationships.

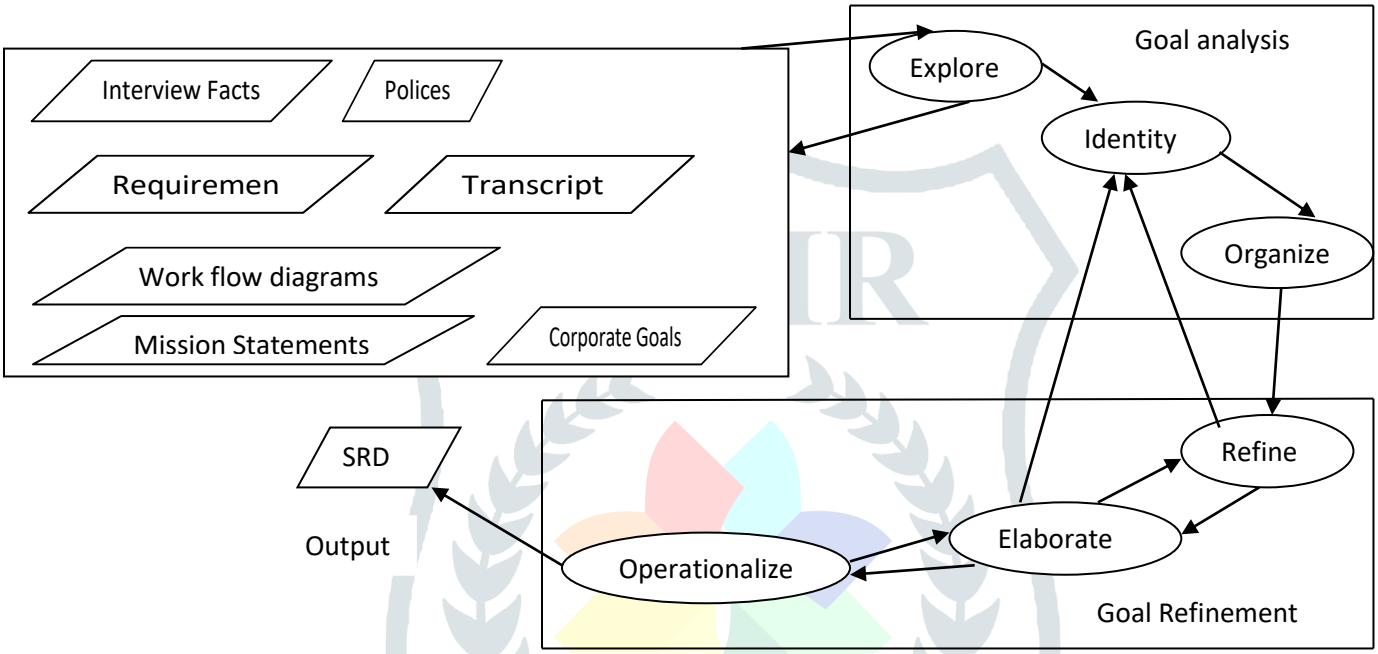

Figure 3 GBRAM Activities

The GBRAM approach support in goal elicitation and modification by practicing requirements engineers with typical questions. For example, one possible question to resolve if a goal is a maintenance goal. GBRAM, goals, agents, stakeholders are particular in the textual form in goal schemas. The goal refinement activities can be listed below as follows:

- Refine activities require the actual pruning of the goal set.
- Elaborate refers to the procedure of analyzing the goal set by bearing in mind possible goal obstacles and constructing scenarios to uncover hidden goals and necessities.
- Operationalize refers to transforming goals into operational necessities for the last requirements specification.

The box in the top left corner of figure 2 contains the possible inputs, which may differ in accordance with the documentation initially available to analysts. The output of GBRAM is always a software requirements document. The SRD includes the functional and non-functional requirements and should be differ specific with regard to the external activities of the system. A generalized synopsis of the inputs and output of every GBRAM activity is presented in Table 1.

Table 1 Input and Outputs of GBRAM Activities

Activity	Inputs	Outputs
Explore	• Requirements Organized artifacts • Interview transcripts Goals • Work flow diagrams • Corporate goals • Policies • Interview facts • Mission statement	• Organized artifacts • Goals
Identify	• Requirements • Interview transcripts • Work flow diagrams • Corporate goals • Mission statement	• Goals • Stakeholders • Agents
Organize	• Goals	• Achievement goals • Maintenance goals • Dependency relations • Reduced goal set • Goal topography
Refine	• Goal set	• Goal obstacles • Scenarios • Constraints
Operationalize	• Goal set	• Requirements • Goal schemas • Action definitions • Software Requirements Document

5. ANALYSIS AND DISCUSSION

In this section analyses the following facts about the requirement engineering process using GBRAM Approach:

Goal Conflicts View

Different stakeholders (clients, users, requirements engineers, developers, etc.) in common have dissimilar objectives, requirements, concerns, perceptions, knowledge, and skills. In order to generate sufficient and complete requirements specification, all applicable view points on the system need to be captured and integrated, with their differences determined suitably. The significance of viewpoints has been identified since the in the early hours of requirements engineering. While inconsistencies may be sources of novel data, eventually, they require being resolved.

Obstacle Analysis

In GBRAM, once an obstruction is recognized, a scenario for it must be constructed. Anton notes that while obstacles indicate the reason why a goal failed, scenarios indicate real circumstances under which a goal may be unsuccessful. Scenarios can be measured instantiations of goal obstacles. GBRAM, scenarios are used to examine obstacles. They may assist uncover hidden goals or other goal obstacles.

Refinement view

The GBRAM method offers strategies and heuristics for goal modification. It does not offer an algorithm for formal or semi-formal analysis of goal models or refinements. In i* and Tropos goals tasks are sophisticated through means-ends and job decompositions. For refining obstacles KAOS provides officially proven modification patterns.

Goal Assigning view

In GBRAM, it is promising for some agents to be accountable for the similar goal at dissimilar times. It allows requirements engineers to analyze option configurations of the border between the system and its environment through the use of OR responsibility relations. Thus, it is possible to compare different system configurations..

Capturing Variability

The inconsistency is modeled by the OR decompositions of goals in the usual AND/OR goal graph. A ranking algorithm for selecting the most excellent system configuration among this huge space of alternatives is planned. It takes into deliberation user preferences, which is modeled as soft-goals and are accompanied by the donation relations relating them to the functional goals, and user skills, which are tough constraints on the leaf-level goals. Our procedure has drawn upon different ideas from goal-based procedures, methods in manage conflicts and formulations of imprecise requirements (as shown in Table 2).

Table 2 Goal-based Requirements Engineering Approaches

Category	Anton	Finkelstein	Dardenne	Mylouplos	Proposed Method
Relationships between goals	Dependent	Cooperative	Conflicting	Support, against	Conflicting, cooperative, irrelevant, counterbalanced
Types of goals	Achievement goal, Maintenance goal	Nonfunctional requirement, satisfying goal	System goal, Privacy goal	Nonfunctional requirement, Goal, satisfying goal, argument goal	Rigid, soft, actor-specific System-specific, functional, nonfunctional
Roles of goals	Requirement. Acquisition, Requirement evolution	Requirement Analysis	Requirement. Acquisition, Requirement Analysis	Nonfunctional requirement analysis	Use case structuring, Requirement evolution, models structuring

6. CONCLUSION

In the current years, the fame of goal-oriented requirements engineering procedures has increased radically. The major cause for this is the insufficiency of the established systems analysis methods dealing with complex software systems. Goal-Oriented requirement engineering offer an incremental procedure for elicitation, analysis, elaboration & modification, condition and modeling of requirements. Different GORE methods survive for these requirement engineering processes based on positive underlying concepts and ideology. Our review, we recognize that obtainable goal oriented necessities elicitation computes do not support priority wise requirements when the stakeholders estimation are often unclear and ambiguity. Therefore, in this paper, we offered Goal-Based Requirements Analysis Method for requirement engineering. It offer appropriate representation procedure to improve stakeholder comprehension and facilitate link between analysts and stakeholders while concurrently offering a reasonable demonstration which can be without difficulty transformed from the language and conventions of the stakeholder's workplace to the language and conventions of analysts and developers. We talk about goals from the viewpoint of two themes: goal analysis and evolution. The objective of GBRAM is to provide practical direction towards the recognition and analysis of the future organizational goals that decide system requirements in the background of software systems development.

REFERRENCES

[1] M. Bano, D. Zowghi, N. Ikram, "Systematic reviews in requirements engineering: A tertiary study", *EmpiRE*, pp. 9-16, 2014.

[2] M. Daneva, D. Damian, A. Marchetto, O. Pastor, "Empirical researchmethodologies and studies in requirements engineering: How far did we come?", *J Syst Software*, vol. 95, pp. 1-9, 2014.

[3] Zhu M X, Luo X-X, Chen X H, and Wu D D, "A Non-functional Requirements Tradeoff Model in Trustworthy Software" Information Science-Elsevier, Vol. 191, pp.61-75, 2012.

[4] Thakurta R, "A Framework for Prioritization of Quality Requirements for Inclusion in a Software Project", Software Quality Journal, Springer, 2012.

[5] S. Ghanavati, D. Amyot, L. Peyton, "A systematic review of goal-oriented requirements management frameworks for business process compliance", *RELAW*, pp. 25-34, 2011.

[6] Zickert F, "Evaluation of the Goal Oriented Requirements Engineering Methods KAOS", American Conference on Information system", pp.1-9, 2010.

[7] Sadiq M, Shahid M, "Elicitation and Prioritization of Software Requirements", International Journal of Recent Trends in Engineering, Vol. 2, No. 3, November, 2009.

[8] P. Zave. Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys, 29(4), 1997.

[9] E. Yu, J. Mylopoulos. Why Goal-Oriented Requirements Engineering. Proc. 4th International Workshop on Requirements Engineering: Foundations of Software Quality, Pisa, Italy, June 1998.

[10] G. Kotonya, I. Sommerville. Requirements Engineering: Processes and Techniques. Wiley, 1998.