KAOS: A Goal Oriented Requirement Engineering Approach

Mizbah Fatima
M. Tech Scholar
Department of Computer Science & Engineering
Al Falah School of Engineering & Technology, Faridabad, Haryana, India

Abstract

For a successful software project it is extremely crucial to properly extract all the requirements in the requirement engineering phase itself. Improper requirements elicitation generally results in a failed software product, eventually resulting in time and monetary loss. Of the many approaches that exist, the Goal Oriented Requirements Engineering (GORE) approach is generally used so that the requirements can meet the customer needs which may be defined as the ultimate goals of the software being developed. There are a number of methodologies under GORE approach, however this paper is focused on describing KAOS or Knowledge Acquisition in autOmated Specification as one of the GORE approaches for Requirement elicitation.

Keywords: Requirements Elicitation, Goal-Oriented Requirements Engineering (GORE) Approach, Goals, Knowledge Acquisition in Automated Specification, KAOS Models

I. INTRODUCTION

Requirements elicitation being the first sub phase of requirement engineering process is defined in literature as the process of finding, analyzing and evaluating, documenting, and comprehending the user's needs and constraints for the system [1].
The resulting product from the elicitation phase is a subset of the goals from the various stakeholders who are the actual sources from whom the requirements are elicited. This leads to the concept of Goal Oriented Requirements Engineering (GORE) approach that is useful in defining, eliciting, organizing, analyzing and refining the requirements, so that the system requirements can meet the customer needs [2]. The Goal oriented requirements elicitation helps in identifying the requirements in the form of high level goals that should be incorporated in the software while conforming to the stakeholders needs. The Goal Oriented Requirements Engineering (GORE) approach is concerned with the activities that lead to the ultimate requirements of the software. It mainly comprises of the following activities: goal elicitation, goal refinement and various types of goal analysis, and the assignment of responsibility for goals to agents. The main GORE approaches discussed in literature include: NFR framework, i*/ TROPOS, KAOS, GBRAM [2]. However this paper shall focus on KAOS in particular.

II. UNDERSTANDING KAOS & KAOS MODELS

KAOS stands for Knowledge Acquisition in autOmated Specification and is a goal oriented requirements engineering approach, developed by University of Oregon and University of Louvain [3]. This methodology for requirements engineering allows analysts to construct requirements models and to obtain requirements documents from KAOS models [4]. The KAOS methodology involves building up of the requirements model which includes the following steps:

1) Build a Goal Model depicting the requirements in the form of goals in AND/OR graph.
2) Build a Responsibility Model in order to achieve those goals through the help of agents.
3) Build an Object model along with building all the consistent and complete glossary of the problem-related terms that are used to write the requirements.
4) Build an Operational model describing the behavior of the agents responsible for achieving the goals they are responsible for.
5) Build the requirements document based on the requirements model.
6) Validate your requirements by first reviewing the model.

The KAOS uses the combination of four models: goal model, responsibility model, object model, operation model.

A. The KAOS Goal Model:

The goal model is considered the base and starting point of the whole method. It declares the goals of the composite system and thus forms the basis for obtaining all the other models through these goals. The goal model represents a set of interrelated goal diagrams that are used to deal with a problem. The main idea behind this approach is to represent system requirements as business goals and objectives and hence focus on realizing these business goals.

Goals are typically all the functional and non-functional requirements that should be incorporated in the system that is being developed, often through the assistance of some agents. The goals are the depiction of the customer needs in the form of properties that the system/application must contain. These needs may be elicited from the various stakeholders by means of any of the existing elicitation techniques and listed in the form of the functionalities required in the software. Functional Requirements are defined as the services/functionalities to be provided to stakeholders and the Non-Functional Requirements are the quality like Security, Performance, Flexibility, Reliability, Usability Scalability and Efficiency and so on that need to be included in the software [2]. These requirements in the form of goals are needed as they are useful in achieving requirements completeness, help in identifying irrelevant requirements, explaining the requirements to the stakeholders, refinement of the goals to sub goals etc.

After the preliminary analysis of the system and identification of the goals by the requirements engineer, these goals are refined into progressively simpler goals until they can be easily implemented. Thus sub goals are derived from the high level goals and are further refined into more concrete sub goals. Through asking how a goal can or should be realized, the goal could be decomposed into a set of sub-goals. The sub-goals could be independent or be associated with each other. KAOS allows the analysts to determine the system goals by interviewing current and future users and by examining the available systems, reading the existing technical documents, etc thus enabling the analysts to organize the gathered goals into directed, acyclic graphs. Each goal (except the leaves, the bottom goals) is refined as a set of sub goals telling how the refined goal can be reached [4]. The top nodes are strategic goals for the business and the lower level nodes represent the low level requirements with the root of the diagram being the ultimate business goals. The process of identifying the penultimate goals followed by the lower goals continues until the analysts reach the basic goals. The lower goals are linked with the parent goals indicating that the completion of the lower goal successfully will cause the definite completion of their parent goal.

This acyclic directed graph called the goal graph or the AND/OR graph has nodes which represent the goals to be achieved by the system and its edges express logical dependency relationships between the connected goals.

The goal graph has two types of goal decomposition: the AND decomposition and the OR decomposition. The AND decomposition signifies that if all of the sub goals are achieved, their parent goal can be achieved or satisfied while in OR decomposition, the achievement of at least one sub goal leads to the achievement of its parent goal. Root goals, having no parents in a graph represent the requirements that the customer would like to realize ultimately and the analyst tries to accomplish them by combining the sub goals.

B. The KAOS Responsibility Model:

The KAOS Responsibility model is a compilation of derived responsibility diagrams. It involves entities called agents which may be humans or automated components that are concerned with achieving the goals/requirements. The assignment of the agents to fulfill the particular goal is done according to the goal model.

The goals are always assigned to a number of agents. However, whenever there is a single agent response for the goal, it indicates that there is no room for any further goal refinement and this difference gives the analyst a criterion to stop refining goals into sub goals.

C. The KAOS Object Model:

The Object model is basically concerned with linking the application domain and establishing constraints on the operational system.

The objects could be categorized as entities, agents and associations where “entities” describe and translate the state of the object but do not carry out operations; the “agents” are concerned with performing the operations whereas “associations” are entities that are dependent on the object and do not have the ability to carry out the operations.

D. The KAOS Operation Model:

The Operation model represents all the behaviors that agents must have to accomplish their needs. Behaviors are basically operations performed by agents. These operations are used to manipulate the objects described in the object model: they can create objects, provoke object state transitions or trigger other operations through sent and received events [4]. These operations depend on the needs of stakeholders. In KAOS an operation diagram typically comprises of operations performed by one or several agents to fulfill a requirement. Compositions are made through data flows (the output of an operation output becomes the input of another operation) or control flow (an event sent by an operation triggers or stops another operation). An operation diagram thus describes how the agents need to cooperate in order to make the system work.

With KAOS, the operation model is connected to the goal model: the analysts justify operations by the goals they “operationalize”. An operation with no justification means that either there is still missing goals in the model or that the operation is not necessary. Conversely if some requirements are left without “operationalization”, they may just be wishful thinking [4].

E. Obtaining Requirements Document from the Requirements Model:

In KAOS, there is a template document which contains all the information extracted from the four models to specify the requirements document. The glossary part is derived from the object model; the requirements are specified according to the goal

model from top (the business/strategic goals) to bottom (the requirements). Requirements on the system architecture are obtained from the responsibility model and requirements for the system behavior from the operation model. This entire process leads to extracting the system requirements and ultimately resulting in a complete, consistent and unambiguous document. Thus the output of the four models is a complete requirements document.

F. Validate Your Requirements by First Reviewing the Model:

The requirements gathered through the KAOS methodology can be validated by reviewing them so as to build a high quality product by organizing collective reviews of the KAOS model. Virtual meetings using shared screens on distant stakeholder's locations can also be used.

III. CONCLUSION

KAOS finds its application in various industries such as mechanics, telecommunication, and health care and hence can be used for any type of information system. It is considered an efficient goal oriented requirement elicitation method that uses the concept of building requirements models. KAOS provides a systematic and sound way to organize requirements and uses graphical way to tackle a problem [5]. It allows defining concepts relevant to the problem description. KAOS could clarify the responsibilities of all stakeholders as each requirement is related with an agent who is responsible for achieving it. Finally, KAOS also provides a clear hierarchy for stakeholders enabling easy and efficient communication [4].

REFERENCES

[1] Sommerville (1997a) Sommerville, I. & Sawyer, P. Requirements Engineering: A Good Practice Guide. New York, NY: John Wiley & Sons, 1997.
[2] Alexei Lapouchnian, "Goal-Oriented Requirements Engineering: An Overview of the Current Research" Depth Report, University of Toronto, 2005
[3] A. Dardenne, A. van Lamsweerde and S. Fickas, "Goal Directed Requirements Acquisition," Science of Computer Programming, Vol. 20, No. 1-2, April 1993, pp. 3-50
[4] Respect IT, A KAOS Tutorial, Objectiver, 2007.
[5] Huzam Saud Faleh Al-Subaie "Evaluating the Effectiveness of a Goal-Oriented Requirements Engineering Method", Research thesis submitted to the University of London, 2007.