Automated Completeness Check in KAOS

Joshua C. Nwokeji, Tony Clark, Balbir Barn, and Vinay Kulkarni

1 Middlesex University, London, UK
Hendon the Burroughs, NW4 4BT
{J.Nwokeji,T.N.Clark,B.Barn}@mdx.ac.uk
2 Tata Consultancy Services, India
{vinay.vkulkarni}@tcs.com

Abstract. KAOS is a popular and useful goal oriented requirements engineering (GORE) language, which can be used in business requirements modelling, specification, and analysis. Currently, KAOS is being used in areas such as business process modelling, and enterprise architecture (EA). But, an incomplete or malformed KAOS model can result to incomplete and erroneous requirements analysis, which in turn can lead to overall systems failure. Therefore, it is necessary to check that a requirements specification in KAOS language are complete and well formed. The contribution at hand is to provide an automated technique for checking the completeness and well-formed-ness of a requirements specification in KAOS language. Such a technique can be useful, especially to business or requirements analysts in industries and research, to check that requirements specification in KAOS language is well formed.

1 Introduction

Requirements engineering (RE) is a core imperative in systems development, this is because it has the ability to cripple the entire system if incomplete or erroneous [4]. In fact, the vast majority of systems failures have been attributed to incomplete or erroneous requirements specification and analysis [2,18]. Goal oriented requirement engineering (GORE) refers to various requirements engineering methods, which uses goals, and other categories of intentions, to analyse and specify the requirements of a system [20]. Among varieties of competing GORE languages such as i* [19], GBRAM [1], and BMM [14], KAOS [3,8,11,12,15,17] is one of the most popular and widely used languages [8].

In order to facilitate a well formed requirements specification in KAOS language, we contribute by developing a tool that can be used to check the completeness of a requirements model in KAOS language. This tool is an extension of our previous work reported in [13], where we developed a comprehensive meta-model and a graphic editor (Ktool) for requirements specification in KAOS.

The rest of this paper is structured as follows: In Section 2 we describe a motivating example and use it to explain certain concepts of the KAOS language. This is also used as a case study for demonstrating the completeness check in KAOS. Section 3 gives a brief description of the KAOS framework. Related work is briefly explained in Section 4. In Section 5 the tooling process is described and a conclusion is given in Section 6.

2 Motivating Example

We describe a motivating example below, and use to explain the various concepts in Kaos and to demonstrate completeness check in Kaos. Consider the requirement analysis of a card payment system for Bank xyz. Bank xyz provides point of sale (POS) services to vendors, via wireless routers. A router can be encrypted but slow (E1), or unencrypted but fast (E2). Vendors sell products to customers but must upload daily transactions to Bank xyz. The bank desires to protect all transactions from hacker, who steal smart card information from customers.

3 KAOS

KAOS is a GORE language that uses the concepts of goals, actors and other intentional elements to elaborate, specify, model and analyse the requirements of a system [8] [10]. A goal is a statement that describes the intentions of a given actor in a system, e.g., steal credit card information [5]. A parent goal may be decomposed, or refined, into child sub-goals [12]; a leaf goal has no children. A leaf goal that has a clear criteria for its satisfaction is called a requirement, else it is called an expectation. A KAOS goal model is said to be complete when all leaf goals are assigned to agents-active entities, machine or human, in a system [15]. KAOS is a composite language/framework consisting of a Goal, Object, Operation and Responsibility models [8,15]. More details about the KAOS model and its elements can be found in [3,8,11,13,15,17].

4 Related Work

In the past, various approaches have been used to check the completeness of KAOS model. For instance, the use of pre, post, and trigger conditions based on temporal logic has been proposed in [16], while the goal question metric (GQM) has been proposed in [7]. Although these approaches can be useful, they involve some sort of rigour or formality, which can be difficult and thus discourage systems or business analysts from using them. However, our approach is different, we aim to provide a conceptual method that can be easier to use and less rigorous when compared with other methods. To achieve this, we consolidate the KAOS elaboration criteria proposed in [16], the completeness criteria proposed in [15], and the GQM in [7] into three completeness checks itemized below, and implement a tool that automates them:

- **Completeness Criteria 1**: All goals must be refined until they become either leaf goals (expectations and requirements) or domain property.
- **Completeness Criteria 2**: All leaf goals must be assigned to agents.
- **Completeness Criteria 3**: All agents, especially machine or software agents, must be assigned to operations.

In the following Sections we explain the processes involved in automating these checks.