Abstract. The work relates to formal verification of requirements models using deductive reasoning. Elicitation of requirements has significant impact on the entire software development process. Therefore, formal verification of requirements models may influence software cost and reliability in a positive way. However, logical specifications, considered as sets of temporal logic formulas, are difficult to specify manually by inexperienced users and this fact can be regarded as a significant obstacle to practical use of deduction-based verification tools. A method of building requirements models, including their logical specifications, is presented step by step. Requirements models are built using some UML diagrams, i.e. use case diagrams, use case scenarios, and activity diagrams. Organizing activity diagrams into predefined workflow patterns enables automated extraction of logical specifications. The crucial aspect of the presented approach is integrating the requirements engineering phase and the automatic generation of logical specifications. Formal verification of requirements models is based on the deductive approach using the semantic tableaux reasoning method. A simple yet illustrative example of development and verification of a requirements model is provided.

Keywords: requirements engineering, formal verification, deductive reasoning, use case diagrams, use case scenarios, activity diagrams, workflows patterns, temporal logic, logical specifications, semantic tableaux method.

1 Introduction

Software modeling enables better understanding of the domain problem and of the system under development. Requirements engineering is an important part of software modeling. Requirements elicitation should lead into a coherent structure of requirements and have significant impact on software quality and costs. Thinking of requirements must precede the analysis, design, and code generation acts. Requirements models are descriptions of delivered services in the context of operational constraints. Identifying software requirements of the system-as-is, gathering requirements and formulation of requirements by users allows defects to be identified earlier in a life cycle.
UML, i.e. the Unified Modeling Language \cite{16}, which is ubiquitous in the software industry can be a powerful tool for the requirements engineering process. Use cases are central to UML since they strongly affect other aspects of the modeled system and, after joining the activity diagrams, may constitute a good vehicle to discover and write down requirements. Temporal logic is a well established formalism which allows to describe properties of reactive systems, also visualized in UML. The semantic tableaux method, which is a proof formalization for assessing logical satisfiability, seems intuitive and may be regarded as goal-based formal reasoning.

Formal methods enable precise formulation of important artifacts arising during software development and help eliminate ambiguity. There are two well established approaches to formal reasoning and system verification \cite{5}. The first is based on state exploration (“model checking”) and the second is based on deductive reasoning. However, model checking is an operational rather than analytic approach. Deductive inference enables the analysis of infinite computation sequences. On the other hand, one important problem of the deductive approach is the lack of automatic methods for obtaining logical specifications considered as sets of temporal logic formulas. The need to build logical specifications manually can be recognized as a major obstacle to untrained users. Thus, the automation of this process seems particularly important. Moreover, application of the formal approach to the entire requirements engineering phase may increase the maturity of requirements models.

Motivation, Contributions and Related Works. The motivation for this work is the lack of tools and practical applications of deductive methods for formal verification of requirements models. Another motivation, which is associated with the previous one, is the lack of tools for automatic extraction of logical specifications from software models. However, requirements models built using use case and activity diagrams seem to be suitable for such an extraction process. All of the aforementioned aspects of the formal approach seem to be an intellectual challenge in software engineering.

The contribution of the work is a method for building formal requirements models, including their logical specification, based on some UML diagrams. A complete deduction-based system which enables the automated and formal verification of requirements models is proposed. Another contribution is a method for automating the generation of logical specifications. The generation algorithm for selected workflow patterns is presented. The reasoning process is performed using the semantic tableaux method for temporal logic. The proposed method is characterized by the following advantages: introducing workflow patterns as primitives to requirements engineering and logical modeling, scaling up to real-world problems, and logical patterns once they are defined and widely used. All these factors are discussed in the work and summarized in the last section.

There are some fundamental works on requirements engineering, c.f. the work by van Lamsweerde \cite{15}, which is a comprehensive study of many fundamentals of this area. The work by Chakraborty et al. \cite{4} discusses some social processes associated with requirements engineering. In the work by Rauf et al. \cite{17}, a