Information Sciences 180 (2010) 3894–3913

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Formal behavior modeling and effective automatic refinement

Shuaiqiang Wang, Jun Ma *, Qiang He, Jiancheng Wan

School of Computer Science and Technology, Shandong University, Shunhua Road, High-New Industrial Development Zone, 250101 Jinan, China

ARTICLE INFO

Article history:
Received 14 January 2008
Received in revised form 6 May 2010
Accepted 17 June 2010

Keywords:
Behavior modeling
Formal refinement
Automatic refinement
Formal method
Model-Driven Development

ABSTRACT

Modeling and refining behaviors of software systems are two crucial issues in the methodology of Model-Driven Development (MDD). Traditional methods include Unified Modeling Language (UML) based methods and formal methods. Recently integrated methods by taking full advantages of these two methods have received increasing attention. Unfortunately, there are still no effective formal modeling and automatic refinement methods for behaviors. Besides, difficulties exist in generating some structures, e.g., explicit loop structures, via existing approaches.

In this paper, we propose a B-notations based framework for formal modeling and automatically refining behaviors. In our framework, three issues are involved: automatic behavior modeling, theory of behavior refinement, and automatic behavior refinement. For automatic behavior modeling, we propose a B-notation based automatic modeling method for behaviors. For theory of behavior refinement, we propose a refinement theory focusing on behaviors that regards the behavior refinement problem as a search problem. For automatic behavior refinement, we propose an automatic refinement approach by combining top-down logic reduction based method and bottom–up local search based method, where explicit loop structures can be well generated.

1. Introduction

In the methodology of Model-Driven Development (MDD), a behavior is used to coordinate a series of basic actions and activities for achieving a certain function in a software system. The problem of behavior modeling essentially involves three issues: (1) When should a behavior be performed? (2) How will the states of a system be transferred step by step due to the behavior? (3) What will the final state be like?

Refinement is a technique used to transform the abstract model of a software system into a more concrete one. Without an accurate approach for specifying and refining their behaviors, software models will be regarded as incomplete and imprecise, and as a result, software systems cannot be generated automatically. Therefore, how to modeling such behaviors and how to stepwise refine these behavior models automatically, have become two critical problems in MDD.

Traditional methods can be classified as Unified Modeling Language (UML) methods and formal methods.

On one hand, UML provides a commonly accessible way of visualizing models, facilitating communication of ideas, and has been widely used for modeling in industry. In addition, automatic refinement for codes can be implemented using Model-Driven Architecture (MDA) [28] technology. Besides, Object Constraint Language (OCL) [23] is a complemental method of UML for specifying detailed invariants, assertions, and can be used to specify behaviors preciously. Unfortunately, OCL is only a specification language. The OCL specifications cannot be refined and transformed into codes directly. In addition, the mechanism for consistency verification is not provided by OCL, either.

* Corresponding author. Tel.: +86 0531 88391528; fax: +86 0531 88392498.
E-mail addresses: shqiang.wang@gmail.com (S. Wang), majun@sdu.edu.cn (J. Ma), uniheqiang@gmail.com (Q. He).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.06.024

On the other hand, formal specifications can be used to provide an unambiguous and precise supplement to natural language descriptions, and can be rigorously validated and verified, leading to the early detection of specification errors. Besides, refinement theories for various formal modeling methods has been developed [1,2,9,29]. Spite of these efforts, uptake within industry has mainly been limited to safety critical applications partially due to lack of automatic refinement facilities for formal models. It is quite difficult to achieve effective automatics for refinement without context and constraints, because there are too many potential directions at each refinement step to choose a proper one.

Recently integrated methods by taking full advantages of UML and formal methods have received increasing attention. First the framework of a system as well as behaviors are specified with UML. Then, these UML specifications are transformed into formal models. Finally, these formal models can be refined and implemented according to certain refinement theory.

The refinement and implementation of behavior models involve generating interactions with other behaviors and context, sequential steps, conditional executions, iterations, etc. Although some valuable efforts have been made in previous researches [17,26], there are still no effective formal modeling and automatic refinement methods for behaviors. Besides, some structures, e.g., explicit loop structures, are hard to generate via existing approaches.

In this paper, we propose a B-notations based general framework for formal modeling and automatically refining behaviors from their UML specifications.

The B-method, originally devised by Abrial [1], is a widely used formal method for software system development. The representations of B-method is called B-notations, that adopts a collection of mathematically based interdependent *Abstract Machines* for modeling, refinement and implementation of software components. Besides, some verification tools, such as Atelier-B,[1] B-Toolkit,[2] and ProB [18], have been well-developed for consistency checking and refinement checking. It has been used in major safety–critical system applications in Europe, such as the Paris Métro Line 14, and is attracting increasing interest in industry.

In our framework, three issues are involved: automatic behavior modeling, theory of behavior refinement, and automatic behavior refinement.

For automatic behavior modeling, we propose a B-notations based modeling method, where each behavior is viewed as a behavior model, and its context is viewed as a data model. Note that our modeling method adopts an incremental strategy targeting improvement in reusability for large systems. In addition, we provide a set of transformation rules, so that the specifications of behaviors in UML and OCL can be automatically transformed into behavior models, referring to automatic behavior modeling. It gets behaviors ready for automatic refinement.

For theory of behavior refinement, we propose a refinement theory focusing on behaviors rather than systems according to the corresponding principles of Event-B [2], where the behavior refinement problem can be viewed as a search problem, i.e., to search a potential trace that satisfies the requirements of behavior refinement. Besides, the complexity of behavior refinement problem is estimated. This theory is the primary principle of the automatic behavior refinement.

For automatic behavior refinement, we propose an automatic framework by combining top–down logic reduction based method and bottom–up local search based method, inspired by [25] that integrates local search methods in Inductive Logic Programming. It includes three steps: primary refinement, final refinement, and optimization. Firstly top–down logic reduction is used to carry out the primary refinement for simplifying problems and generating explicit loop structures. Then bottom–up local search is used to execute the final refinement according to the definition and requirements of behavior refinement. Finally an optimization method is used to optimize the refinement results.

Our main contribution is that we investigate the automatic mechanisms for formal modeling and refinement based on our proposed behavior models. In particular, in our framework, we introduce an automatic modeling method based on the UML and OCL specifications. In order to achieve automatic refinement, we provide a novel theory of formal refinement for behavior models, that regards the refinement problem as searching a potential trace satisfying certain requirements. Then based on the theory of behavior refinement, we propose an effective approach to implement behavior refinement by combining top–down logic reduction and bottom–up local search.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 states the motivation of our approach with an example. Section 4 describes the symbol systems as used in this paper. Section 5 presents the basic conceptions including data model and behavior. Section 6 introduces the theory of the behavior refinement. Section 7 proposes the automatic refinement mechanism. Finally, Section 8 concludes the paper and gives suggestions for future work.

2. Related work

Traditionally, there are two methods to model and refine behaviors, including UML-based methods and formal methods. Recently integrated methods have received increasing attention by taking full advantage of UML and formal methods. Some representatives are reviewed as follows:

2.1. UML-based methods

UML provides a commonly accessible way of visualizing models, facilitating communication of ideas. Based on the UML specifications, codes can be generated automatically through stepwise model refinement, using Model-Driven Architecture

[1] http://www.atelierb.eu/
[2] http://www.b-core.com/btoolkit.html