Title	振舞仕様によるUMLの意味解析に関する研究
Author(s)	宗像，一樹
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1441
Rights	
Description	Supervisor:二木 厚吉，情報科学研究科，修士

The semantic analysis of UML by behavioural specification

Munakata Kazuki

School of Information Science,
Japan Advanced Institute of Science and Technology

February 15, 2001

Keywords: UML, OCL, formal specification, CafeOBJ, behvaioural specification.

1 Introduction

Formal method and object-oriented technology have a long history as the main technology for the improvement in productivity and quality of software systems. In object-oriented technology, the arrangement of methodology and modeling language is progressing in addition to advanced reuse technology, and that practicality is high. Unified Modeling Language(UML) is a notation based diagrams for object-oriented modeling, and is easy to use for a practical developer. UML comprises an annotation language for UML models, Object Constraint Language(OCL). The OCL adds conformance to the diagrams. On the other hand, formal method has a sound mathematical basis, and offers a strict modeling method by formal specification language and automatic verification. formal method contributes to the improvement in the reliability of software greatly.

In such backgrounds, the researches about the fusion of object-oriented technology and formal method is performed on various levels. These are the trials in which a strict modeling and automatic verification of the formal method will be applied to object-oriented technology. Especially, the application to specification technique is an important subject, because the specification affects the quality of the software finally created. So, in object-oriented development, the advanced support by the computer for the modeling with UML is asked. However, in the present UML, its formal semantics are not arranged well. Therefore, it is not possible to apply rigorous automated analysis or to execute a UML model in order to test its behaviour.

The purpose of this research is to propose a framework for the analysis of UML models by computer, with the mapping UML to formal specification language. We assume that this framework can promote finding the misunderstanding of the software in the early stage of the development by spcification-based simulation and the verification.

2 Our approach

In our research, we use algebraic specification language CafeOBJ, because CafeOBJ can treat the object-oriented concept, and also can execute the specification by term rewriting engine, as the environment for supports of its verification. Especially, according to hidden algebra, CafeOBJ can treat object-oriented model naturally. In hidden algebra, a system is regarded as a kind of black box in the sense that we can observe its state only by using special operators called observation. The space of the states of a system is distinguished with the data used by the system and represented as a hidden sort. The data used by a system are represented by ordinary sorts. The state of an object can be changed only by using special operators called action. Specifications based on hidden algebra are called behavioural specification. Moreover, by composing the systems described as hidden algebra, it is possible to describe a bigger system. This composing of specifications is done by projection operator. A projection operator is defined for each subsystem to get its state from the state of the composite system. This mechanism for composing of systems by projection operator is called object composition.

The target UML of our research is UML class diagram with OCL constraints. In this paper, we propose 1) the definition of UML class diagram with OCL constraints by behavioural specification, and 2) the way to analyze the specifications converted according to the definition, using CafeOBJ interpreter. By this, the executability and the verification of CafeOBJ can be applied to UML models.

In the definition of UML model by behavioural specification, firstly, we take correspondence between behavioural specification and class diagrams which consist of class and association. We define a class as a module based on hidden algebra, and an association as object composition which composes each module (as a class) by projection operator. The behavioural specification derived from class diagram is only the signature of the specification. Secondly, we take correspondence between behavioural specification and OCL expression, and define OCL constraint as equations (as an axiom). As above, we derive the signatures of the specifications from class diagram, and the equations of the specifications from OCL constraints in order to define its behaviour.

In the analysis UML model with CafeOBJ interpreter, we present a method to execute UML model with specification-based simulation, and to check OCL phrases. The execution of UML model with the simulation is done by the reduction using equations derived from OCL constraints. And we present type checking, dynamic checking and invariant verification as OCL checking . The Type checking is a syntax check of OCL expressions, and the dynamic checking is a dynamical check of OCL constraints at runtime of a system. This dynamic checking is performed combining above simulation mechanism. And we show a verification which proves that OCL invariant is always satisfied in the all state of a system.

3 Conclusion

In this paper, we proposed how to apply a formal specification language to the analysis of UML model by a computer. The ideas can apply the executability and the verification technique of formal method to specifications with UML before its implementation. And we have prospects of the application as a UML tool. The simulation and verification of UML model are difficult in existing UML tools.

In OCL, the constraints are defined by using the attribute values of a class or the values of query operations, and in behavioural specification, its behaviour is defined by using the results of observation operator. So, behavioural specification can treat the concept of OCL very naturally. And, by expressing also the concept of a class and an association with behavioural specification, it is possible to treat class diagram and OCL with one formalism. Moreover, we showed the way to express the multiplicity of an association and OCL collection type (and its operations) derived from multiple association. We showed some easy examples of the translation from UML model to behavioural specification.

Generally, in order to do OCL type checking, it is necessary to acquire the information from a class diagram in addition to support for the standard OCL library. We showed that it is possible to check OCL model type in our approach, because a class diagram and OCL expression is treated with one specification language. In dynamic checking of OCL constraints, the execution of a specification is a key point. In our approach, the combination with specification-based simulation enables to check OCL constraints dynamically. Moreover, we showed the example of the verification which proves that OCL invariant was satisfied in the all state of a system.

Examples of UML model in this paper is a limited cases, but we have prospects to treat more complex cases, and to take more use of CafeOBJ verification technique.