

Title	整合性を考慮したUMLモデル支援環境の研究
Author(s)	馬場, 茂雄
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1461
Rights	
Description	Supervisor:片山 卓也, 情報科学研究科, 修士

A Study of An Environment for UML Model Taking Consistency into Consideration

Shigeo Baba

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

February, 15 2001

Keywords: object-oriented, model consistency, UML, relational database, computer support.

Object-oriented methodologies are focused, today. In such methodologies, software is considered as a collection of discrete objects that incorporates both a data structure and a behavior. Up to this day, many kind of Object-oriented methodologies including Booch, OMT, and OOSE is proposed. In Object-oriented methodologies, we make a model for each development phase in a software development process. OMG(Object Management Group) standardized UML as model notation for Object-oriented methodologies. UML is widely pervaded now.

UML has nine kinds of diagram. In practical development, we construct a model using some diagrams. At that time, the model may set inconsistencies among diagrams. It is difficult that we detect inconsistencies among diagrams as an analysis and a design model get complex and large and large system developments. In this paper, we proposed a method for checking a consistency of the model. Consistency checking for the model use constraint rules. To express UML construction, Well-Formedness Rules are in UML specification. We also define the constraint on a methodologies and the constraint on user defined. We had never found a tool which has mechanism to customize constraint flexibly and to check it. In this study, we show being able to check the model consistencies with facility by means of dealing with model information and constraint description on a relational database. In proposed method by our, we allow us to define constraints flexibly and to check them.

We manage all UML model on a relational database. First of all, we must define the specification of tables. Structure of UML defined in UML metamodel. So we define tables based on logical structure of metamodel. We generate tables for each

metaclasses in metamodel, and define that attribute name of a table is property in metaclass. The property consists of attribute of metaclass and navigation property. Navigation is seeing property of another class associated with the class, and the way is that we invoke association end role name. Let navigation property be association end role name which is able to navigate. When metaclass has a generalization relation each other, child metaclass is inherit the property from the parent. The table consist of a set of tuples including the informations of metaclass instance. We can unify managing nine kind of UML diagrams by the means of making tables every metaclass. We can take three advantages as follow or above by virtue of using relational database system. To begin with we can comparatively introduce with facility, one of the reasons for relational database system is pervasive largely. Then we reduce considerably cost for developing software system, the reason for can use existing system. Finally, we can develop high reliable system on the grounds that can use mature system already.

The consistency conditions are constraints for UML and they equivalent invariant condition of UML metamodel. Those conditions are denoted by OCL(Object Constraint Language) which is the standard language for constraint notation of UML. We can check whether consistent or inconsistent. If it was inconsistent, we could pick up the point by virtue of the mechanism by which SQL querys are converted from OCL expressions. Then we are able to access to relational database by SQL querys via DBMS. Both of OCL expression and SQL query are not denoted a procedurly and both consist of statement having no side effects. And both have statements which manipulate a set of elements. Thus there are many similarity between OCL expression and SQL query, so that we can easily convert them comparatively. This approach is effective mechanism in checking consistency for UML syntax.

By dint of availing environment proposed in this study, we can easily check matched relation between model element of UML and the like. In this paper, we set up example used such constraints and describe the process and the consequence of detecting inconsistency for UML models.