Supporting Configuration Management for Software Documents in Cooperative Software Development

Masayuki Kotani (110044)

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

February 14, 2003

Keywords: Unified Modeling Language(UML), Model-based Translation, Meta model, Configuration Management.

1 Background and Purpose

In the cooperative software development, many artifacts are made by developers, referring to one made already. So, there is some Dependency among artifacts. Using and managing the Dependency, we can specify the artifacts to change, when we change one. But there is a large cost that developers define the Dependency. In this paper, I propose the method to create Dependency among artifacts automatically using Meta model about Change Management. The target of this study is the diagram set of Unified Modeling Language (UML). We call Change Management Model as the meta model. The system, which is realized this way, gets UML Diagrams or the components of UML Diagrams and corresponds them to the Meta Elements, which is the component of Change Management Model. Using Change Management Model, next, it adds the relation among Meta Elements. Using those relations, finally, it outputs Dependency among UML Diagrams.

2 Change Management Model for UML Diagrams

We investigate the particle size that is possible to define Dependency among UML Diagrams. We find the relation between the component of UML Diagrams and UML Diagram. Therefore, We define the Meta Elements using components of UML Diagram and UML Diagrams.

We define "Relationship Diagram", "Behavior Diagram" and " Communication Diagram" as the Meta Elements corresponding to UML Diagrams, and "Classifier", "Relationship", "Meta State", "Meta Transition", "Meta Instance" and "Meta Message" as the Meta Elements corresponding to Components of UML Diagrams. Relationship Diagram is the diagram expressing the relation among some thing. UML Diagrams corresponding to Relationship Diagram are Use Case Diagram, Class Diagram, Object Diagram, Component Diagram and Deployment Diagram. Components of UML Diagrams corresponding to Classifier are things that expressed the relation in each UML Diagrams. Behavior Diagram is the diagram that expressed behavior in Classifier, is Meta Element of Statechart Diagram and Activity Diagram. Communication Diagram is the diagram that expressed communications among Classifiers, is Meta Element of Sequence Diagram and Collaboration Diagram. And we define the Domain, that is set of Meta Elements corresponding to a UML Diagram.

We define "Same Related Copy", "Same Related Instanciation", and "Following About Existence" as the relation among Meta Elements. Same Related Copy and Same Related Instanciation are same between Meta Elements on both ends of those relations, including different from type, as Class and Object, of the Meta Elements. Following About Existence is the relation of existence that Following Meta Element is deleted when the other Meta Element is deleted. It is different from the content of both Meta Elements. Defined relations are effectiveness about Delete, because they have a restriction about existence.

We define the way to create Dependency among UML Diagrams. There is a relation, which have a restriction about existence, among Meta Elements over "Domain". This relation read Dependency among UML Diagrams, because Domain is corresponding to a UML Diagram.

We define the relation among Meta Elements as Change Management

Model. At First, we define "Model Element", which is a parent of all Meta Elements, has a Same Related Copy as a recursive relation. And, we define relations over Domain. The relation between Relationship Diagram Domain and Behavior Diagram Domain is corresponding to the relation between Classifier and Behavior Diagram. It is Following About Existence and Obeying Meta Element is Behavior Diagram. The relation between Relationship Diagram Domain and Communication Diagram Domain is corresponding to the relation between Classifier and Meta Instance. It is Same Related Instanciation and instanciated Meta Element is Meta Instance.

3 Examine the Effectiveness

We examine the effectiveness of the proposed method by simple case study. Many Dependencies are agreement, comparing the dependency created the proposed method with them defined by person. But any Dependencies are not. Dependencies not created the proposed method are two kinds Dependencies. One is because the relation is not defined between Classifier and Communication Diagram. There is two about the relation between Classifier and Communication Diagram Domain, such as the relation between Classifier and Meta Instance, and between Classifier and Communication Diagram. The way to support it is to detail the Classifier. The other is because the relation is not defined between Meta Instance and Behavior Diagram. This relation is be able to reason from the relation between Classifier and Behavior Diagram and the relation between Classifier and Meta Instance. It depends on the degree of expressing Model whether to define this relation or not. This problem is future works. The other hands, There is also Dependency that create by proposed method but not defined by person. In concretely, the person don't define all Dependencies between Class Diagrams have a same class and Collaboration Diagram has a object instanciated it. But, in purposed method, there are relations among all classes and object. So, There is Dependencies among Class Diagrams and Collaboration Diagram. This is an advantage using proposed method.

4 Conclusion and Future Works

In this paper, we proposed the method to create Dependency among UML Diagrams using the meta model about Change Management. First, we defined Meta Elements corresponding to UML Diagrams and Components in UML Diagrams, and "Same Related Copy", " Same Related Instanciate", and "Obedience About Existence" as relations between Meta Elements. Next, we defined the method to create Dependency using the relation over Domain corresponding to a UML Diagram, and relations between Meta Elements using defined relations. We applied the proposed method by simple case study. In the results, we got two problem points and one an advantage. First, about problem points, One is a Classifier was the rough particle size, the other is we found the relation which can be reasoned. Other hands, About An advantage point, If there was a Dependency to depend on the relation between Meta Element A and B, The Diagram, which has Meta Element C copied Meta Element A, has also Dependency with the Diagram has Meta Element B.

In the future works, we survey the relation about Addition and Updating. And we suggest the method of creating the change process by regarding the created Dependency as the workflow about the Change works.