Automatic Generation of Dependency Relationships between UML Elements

Kentaro Kanai (710018)

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

February 5, 2009

Keywords: UML, Dependency Relationships, Automatic Generation, Change Impact Analysis.

1 Introduction

In software development, various documents such as requirement definitions and source codes are written. A huge number of the documents are made as the development progresses. In order to change a huge number of UML documents safely and efficiently, Kotani[1] proposed a set of auto-generable basic dependency relationships (BDRs) which is useful for change impact analysis for UML1.5. The method that generates BDRs uses the Dependency Generation Model which consists of comparison rules, addition rules and selection rules.

In this paper, we aim to extend his results to deal with UML2.0. When decomposition of a concept [2] arises in an UML element, Kotani's method is not effective. So, we improve his method to deal with the decomposition of a concept.

[2]We define a "concept" as the action or function which the designer gives to UML modelings and elements.

We define "decomposition of a concept" as dividing into a number of UML elements to fill the action or function.

2 Improvement of the Dependency Generation Model

On the handling to UML2.0, we extend the Dependency Generation Model which auto-generates BDRs. Specifically, we improve the comparison rules and the addition rules.

- The comparison rules are the rules that search for the combinations of UML modelings and elements that are candidates for appending dependency relationship. We need to extend the comparison rules on the UML modelings and elements newly added in UML2.0. For example, Composite Structure diagram is newly added in UML2.0. This diagram can express the inside of a class. There are a certain class 'A' and a certain Composite Structure diagram 'B'. If the name of A and B is alike, the comparison rule is applied between A and B. Thus, BDRs are appended between A and B.
- The addition rules define which BDRs can be appended between UML modelings and elements. We need to upgrade addition rules about the UML modelings and elements newly added in UML2.0.

About the handling to the decomposition of a concept, Kotani's method[1] can auto-generate BDRs from the decomposition of a concept that appears in the late phase and success in change impact analysis. This can be explained as follows. About the decomposition of a concept that appears in the late phase, the names between UML elements is alike in many cases, and we can apply comparison rules because of the inclusive relation between UML elements. So, we can auto-generate BDRs. But, on the decomposition of a concept that appears in the early phase, the names between UML elements is not alike in many cases. If we fail to set BDRs in the early phase, we may overlook the elements in the late phase which are affected by the ones in the early phase in terms of change impact.

In this paper, in the decomposition of a concept of use cases, we propose that a designer should append dependency relationships between UML elements. If we append dependency relationship between UML elements, since Kotani's method can append dependency relationship between UML elements in the late phase from this phase, we can analyze change impact in the early phase and the late phase.

3 Evaluation

We compare and evaluate Kotani's result and our result. We conduct the evaluation experiment through a case study of an elevator control system.

When a use cases are changed, we count the number of UML elements affected by the change impact.

When a designer appends a few dependencys relationship in the early phase, the UML elements more than the number of the added dependency relationships have been extracted. As a result, the recall of our method is over 94% though Kotani's recall is about 70%.

4 Conclusion

In this paper, we improve the dependency relationship model and propose the method of auto-generating BDRs for UML2.0. We can handle the decomposition of a concept by a designer's appending dependency relationships.

References

[1] 小谷正行, 落水浩一郎: "UML 記述の変更波及解析に利用可能な依存関係の自動生成法", 情報処理学会論文誌 Vol.49 No.7 pp.2265-2291, 2008.7