Object-Oriented Meta Modelling

Motoshi Saeki
Dept. of Computer Science,
Tokyo Institute of Technology,
Ookayama 2-12-1, Meguro-ku, Tokyo 152, Japan

Abstract. For efficient system specification development, it is very important that the developers can compose the development methods suitable for their problem domain and environment before starting system development. Method Engineering, especially meta-modelling technique, provides us with the powerful devices to compose the methods effectively. In this framework, we develop new methods and their supporting tools easily from the data base of the existing methods and/or method fragments (meaningful constituents of methods), called method base. This paper discusses a meta-modelling technique based on object-oriented concept for the method base system. Information hiding and inheritance mechanism in object-oriented concept provide method fragments with high modularity and reusability. Method fragments are modelled and described in object-oriented formal description language Object Z. We also discuss how to customize method fragments and to integrate them to a new method.

1 Introduction

Many methods and their supporting tools, so called CASE tools, have been developed to support system specification and design processes, and some of them have been put into practice[7]. However, the effects of used methods greatly depend on problem domain where the system to be developed is. That is to say, some methods work well in specific problem domain and do not work well in the other domain. For example, it was pointed out that data flow diagrams (DFDs) are not suitable for developing real-time systems and incorporating state transition diagrams to DFDs is needed[14]. What method is the most suitable for specification development in every problem domain? We think that the answer is “none”. It is difficult to have an universal method which works well in every domain. The best solution that we can consider is that the developers compose suitable methods for their problem domain and development environment.

Method Engineering[4], especially meta-modelling technique, provides us with the powerful devices to develop methods effectively. According to [10], we could compose new methods and their supporting tools easily from a data base of the existing methods and/or method fragments (meaningful constituents of methods), called method base. How to extract method fragments and to store them in a method base is closely related to the meta-modelling technique that we employ. We should construct meta models of method fragments so that we can retrieve, customize, and integrate the method fragments into a new method which we aim at.
This paper discusses object-oriented meta modelling techniques to construct a method base which holds method fragments of high reusability and customizability. The reason why we have employed object-orientation concept is that it provides high modularity and reusability resulting from information hiding and inheritance mechanism. The organization of the paper is as follows. In the next section, we introduce our meta modelling technique based on object-oriented technique. In our approach, method fragments are described in object-oriented formal description language Object Z[3]. In the sections 3 and 4, we will illustrate how to customize method fragments and how to assemble and integrate method fragments into a method. Finally, we will discuss the research directions to developing a practical method base based on our technique.

2 Meta Modelling based on Object-Oriented Concept

2.1 Meta Modelling

Meta modelling is modelling system development methods or method fragments which they have as meaningful constituents. Until now, several meta modelling techniques based on Entity Relationship Model[11, 1], Attribute Grammer[5], and Predicate Logic[1]. Entity Relationship Model has been widely used on account of its simplicity and easiness to implement in conventional CASE tools.

Meta modelling techniques for a practical method base should provide high modularity and reusability on method-fragment descriptions. High modularity provides modifiability of method fragments. If method-fragment descriptions have high modularity, the effect of modifying them is not propagated to the other fragment modules. It results in customizable method fragments. When composing the new method from method fragments, we have to specify constraints on the method fragments[8]. So the meta modelling techniques must have enough expressive power for the constraints.

Entity Relationship Model approach does not necessarily satisfy the above requirements. In particular, it cannot represent more complex constraints although it can specify the cardinality of relationships, e.g. one-to-many. Thus we need another device such as Predicate Logic so that it can specify the varieties of complex constraints on method fragments. Furthermore almost all of the existing modelling techniques did not consider the modularity and reusability of the described method fragments.

Object-oriented approach for software development provides high reusability and modularity of software components. So we adopt object-oriented approach to meta-modelling and represent meta-models, i.e. description of methods, in an object-oriented language. We call our approach object-oriented meta modelling. Furthermore we use more expressive and formal language such as Object Z. Object Z is an object-oriented extended version of Z[12] based on ZF set theory and Predicate Logic. Logical formulas in Predicate Logic are very powerful to express any constraints on method fragments. Similar to complex constraints on a method fragment, intra-constraints can be expressed in logical formulas of Object Z.