Study on a design constraint verification method using aspect-oriented language

Shinsuke Terashima (410084)

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

February 9, 2006

Keywords: AspectC++, Aspect-oriented, Embedded system, constraint .

As the complexity and the scale of embedded system become larger and larger, development style of embedded system had been changed a lot. In the conventional development style was implementation centri. Therefore many problems arose on software quality. To solve these problems, object-oriented software development approaches are going to be adopted in this domain.

The object-oriented development style consists to three phases: analysis, design, and implementation phase. In an analysis phase, we analyze the target system and domain to develop logical model. Next, in a design phase, we develop design model under the constraints of logical model. Lastly we implement the system under the constraint of design model.

In embedded system, there are severe constraints on resources and timing For example, concurrency, resource management and so on. Consiering the phase of object-oriented development style, we can observe the followings: In analysis phase, we develop logical model, in which we depict concurrency and resource management that are essentially needed from the requirement. In design phase, we develop design model considering how to realize concurrency and resource management. In implementation process, we actually implement the system based on the design model. As just described, software is developed under the constraints imposed from

the previous phase. Therefore, it is important to check if the software is actually follows these design constraints.

Though object-oriented method is a good one, there still remains drawbacks, and there proposed other additional ideas to the method. Aspect-oriented method is one of them. If we use aspect oriented programming language, we can add functionalities to the program, without modifying the original code. This characteristics is quite good for design constraitns checking, because we can add check program without modifying the code. Furthemore, we may weave checking aspects to original code in cross-development environment, and then remove aspects before we bring the program into target system.If we used aspect-oriented method, it ha many advantages in checking design constraints.

Our objectives are to establish how to check design constraint by aspect oriented language, and we implement the tool that reflected in these techniques.

In this paper, firstly, we have summarized the relationship between design constraints and aspect. We have categorized a design phase into architecture design phase and detail design phase. We have examined design constraints for each design phase. As many design constraints relate to , message flow and data flow and these flow can be consider to be cross-cutting concerns, we examine the way to check these consitraints utilizing aspect-orientation.

Next, we have proposed an expression technique for design constraint. The expression technique uses sequence diagram of UML2.0 and OCL (Object Design constraint Language). The sequence diagram consists of a variety of frame each of which represents different types of behaviors, and are called combined fragment. In addition, OCL are used for representing constraints that cannot be expressed by sequence diagram. Finally, we implemented a checking tool based on the expression technique. We have evaluate ou idea by checking design constrains of a case provided by an industry.

Using our technique, we can check design constraints represented as message flow and data flow. We have also demonstrate the usefulness of our proposal based on actual case. design.