Abstract of dissertation

In software design, it is important to encapsulate cross-cutting concerns, and the application of aspect-oriented technologies to design modeling is a significant challenge. We examined the design of software for embedded systems that exhibit complicated behavior and observed that aspect orientation is useful for designing such systems.
Aspect-orientation is one of the promising techniques not only for programming but also for analysis and design. For programming, we have a popular language, AspectJ, and the existence of this language facilitates the diffusion of aspect-oriented programming. Likewise, in order to actually utilize aspect-orientation in analysis and design phase, it is desirable to have good aspect-oriented modeling mechanism. Here, aspect-oriented modeling mechanism provides us the means for aspect-oriented modeling, in which the aspect-oriented concept such as “aspect” as well as conventional concepts such as “class” and “association” should be treated as the first class modeling elements.
We are studying the application of aspect-orientation to software development, especially to embedded software design, and developing an aspect-oriented modeling mechanism for it. In our study, an aspect is a unit to modularize a concern, and it includes fragments of software model. In other words, it is a structure to realize a concern. It is similar to a “hyperslice” of Hyper/J. In our modeling mechanism, an aspect contains one or more classes, each of which can have a state model that expresses its behavior. In our research, we found that this kind of modularizing mechanism is useful for embedded software design.
In this paper, we propose an aspect-oriented modeling mechanism. In this mechanism, aspects are defined in terms of structure model depicted by class diagrams and behavior model depicted by state diagrams, and we compose aspects by providing rules that define the interaction among behavior model in different aspects by means of elements defined in behavior model (such as events and transitions). We also introduce how this modeling mechanism is used to design embedded software.