Research of diagraming of the text file using formal specification description

Takayuki Tezuka (610703)

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

February 10, 2008

Keywords: UML, Software engineering, object-oriented analysis, Dependency analysis.

In recent years, embedded systems are often employed in such various devices as cellular phones, automobiles and Multi-Function-Printers (MFP), and the size of which has increased significantly. It is usual for the number of program lines in such items to exceed one million, and in some cases it may reach ten million. As the size of program increases, so does the time required to develop the software in embedded devices. At the same time, the product development cycles for such items have tended to decrease. Thus, the developers of the embedded software are pressed to speed up the overall pace of development.

To address this situation, the author proposes a tool, which generates a UML diagram automatically from a given specification written in natural language, to help the developers to speed up the programming stage of embedded devices.

In this study, the author considers cooking recipe as an example of specification documents, which describe a step-by-step plan that helps cookers to guide the whole process. It specifies which, how, and when the ingredients should be added, boiled or baked, and how it should be completed. In order to clarify this process, the authors aim at representing it in UML diagram. The author has included the following four ideas.

Idea 1 Sequential numbers are given to sentences of natural language, and are regarded variables of arguments for methods of object-oriented framework.

Idea 2 When a noun is a class, those contiguous nouns are treated as an attribute of the noun.

Idea 3 An explanation of an action are given to the class diagram as comment sentences of Java program.

Idea 4 The UML modeling tool is invoked by key events from applications, in the way of reverse engineering.

As a result, the contiguous arguments as consecutive numbers [Idea 1] could ease the generation of sequence figure, though the sentences should have been arranged in the sequential order. In the actual situations, human writes specifications by hand in the contingent order, and furthermore, the specification is often changed; thus, the specification documents are not always arranged in the procedural order.

The attribute of the noun class [Idea 2] has contributed to the visualization of ingredients and condiments.

[Idea 3], i.e., the representation of extraneous comments on the diagram by method comments of Java, should be closely investigated since it may contribute various modeling tools, including JUDE, which has been employed in this study. If this is versatile this method is quite convenient for software developers; those comments often appear in multiple documents, but if it appears once in the diagram it would be much helpful.

[Idea 4] does not directly affect this study. However, the key events could automate operations and also prevent misoperations.