Program development in Java based on CafeOBJ specifications

Ha Xuan Linh (1510214)
School of Information Science,
Japan Advanced Institute of Science and Technology

July 26, 2017

Keywords: formal specification, CafeOBJ, OTS, Java, concurrent programming.

In the formal specification, two basic styles that are often used to formally specify a system are the model-based and algebraic specification. In the model-based approach, the state of a system is represented by a set of data variables and transitions can be described by operations that manipulate these variables. All possible values of these variables, constrained by invariants over them define the state space of the system. Each operation playing the role of a transition is often defined with pre and post-conditions. This representation makes it likely easier for programmers to write programs as well as for automatic tools to generate codes in object-oriented programming languages from the specification. One of the well-known frameworks using this approach is VDM++, which has been applied to many industrial projects, such as the development of the "Mobile FeliCa" IC Chip Firmware, which is widely used in Japan. While it is convenient to implement specifications written in VDM++, there is one issue that can be dealt with algebraic specification languages better than VDM++: formal verification. Some model-based specification languages, such as Z and Event-B, are equipped with a model checker and/or an interactive theorem prover, but to the best of our knowledge, VDM++ is not. Testing is mainly used to check specifications written in VDM++.

In contrast, the algebraic specification is more powerful in applying verification techniques but it seems more difficult for programmers to read and implement it. OTS in CafeOBJ is a way of specifying systems in this approach, in which the semantics of a system is built from observers and transitions. From a certain point of view, the observers can be seen as data variables in the system. OTSs can be used for writing formal specifications of various software systems and verifying properties of them. It implements equational logic by rewriting and can be used as a powerful interactive theorem proving system. Despite its usefulness, up to present, few researches and engineers have applied CafeOBJ specifications to software development, while object-oriented specification techniques such as VDM++, Event-B and Z are successfully applied to many projects.

In this project, we briefly introduce some specification techniques which have been successfully applied to software development nowadays, the pros and cons of each technique and compare them with CafeOBJ, an algebraic specification language currently used in our research group. In addition, we have proposed a possible way to write a concurrent Java program based on an OTS specification in CafeOBJ. By annotating the OTS specifications, programmers have more information about the concurrent programs they write

such as the number of threads in the program, global and local variables and the methods in each thread. Because the systems specified in CafeOBJ are composed of several components interacting with each other, we have successfully applied state pattern in Java to design the programs where each component in the system can be developed as a separate state machine. This can help programmers easier to understand the programs. Some case studies on writing the simulator of communication protocols such as ABP, Simple Cloud and Qlock are also represented in this research project to demonstrate our method.

Besides, we also propose a method to test the concurrent programs based on their OTS specifications by applying some external support tools such as JPF and Maude. JPF can model check Java bytecode programs but one of the most serious problem when using a model checker like JPF is the state space explosion where the size of the system state space grows exponentially as the number of state variables in the system increases. Despite that, we could control JPF to model check a part of the whole state space. Maude is another model checker whose specifications can be converted from CafeOBJ ones by YAST. We have proposed a solution to combine these tools in our testing method to check if a trace from JPF can be a valid computation in Maude specification or not. From that result, we can have a conclusion to some extent that if the written program conform with the OTS specification.