

Progression Report
of

A Study on the Translation from OTS/CafeOBJ Specifications to Java Programs

Jittisak SENACHAK
School of Information Science
Japan Advanced Institute of Science and Technology

1. Aim of Research

To develop reliable programs. In this case, reliable programs are the real implementations of their (formal) specifications. They must preserve the properties that have verified that they are held for a specification.

2. Approach/Idea

One way of developing reliable software for a system is to generate programs automatically from specifications and to guarantee that the programs are real implementations of the specifications. In other words, if any desired safety properties has verified that it is held for a specification, it can be guaranteed that that property will be preserved in the program generated from the specification.

In the formal method called "the OTS/CafeOBJ method"[], a system is modelled as an observational transition system (OTS). The OTS is written in CafeOBJ, an algebraic specification language, and the system modelled as an OTS can be verified that it has properties by writing proofs in CafeOBJ and checking the proof by the means of rewriting with the CafeOBJ system. In this method, the preservation of properties can be determined on the change of observational values.

Given any specification and a translator with the desired property, we can generate a Java reliable program, if the desired property (or called "Refinement Relation") is that for any OTS/CafeOBJ specification, the generated Java program is a real implementation of the specification. This relation is defined on the observing of observation values on the corresponding execution of a specification and corresponding program.

3. Progress of This Year

- Designed and implemented a translator that takes an OTS/CafeOBJ specification and generates a Java program. A translator must satisfy a desired property (having refinement relation).
- To show that a translator has such property, we have formalized OTS/CafeOBJ specifications, Java programs and the translator in CafeOBJ. Let call them as Cafe-OTS, Cafe-Java, and Cafe-Trans respectively.
- Define a refinement relation candidate, and formally verified that the translator has that relation with the CafeOBJ system.
- Also have experienced with a small concrete example: Bank Account.

4. Future Direction

In the current implementation, there are some considerable features that exist in the real system like a distributed system; for example,

- Extend the feature of specification to deal with an abstract data type which can be specified in CafeOBJ. It can be done by more extra work to gather the definitions of the new data type into the translator
- As an specification represents the whole system, all observation values and transitions are considered on the domain of one module. In fact, the system can design into many subsystems.
- Optimizing code should consider for the sake of performance, readability, and so on.

5. List of Publications and/or Systems having made

- J.Senachak, A Study on the Transformation from OTS/CafeOBJ Specifications to Java Programs, Master Thesis, Sep 2004.
- J.Senachak, T.Seino, K.Ogata, and K.Futatsugi, Provably Correct Translation from CafeOBJ into Java, Software Engineering Knowledge Engineering (SEKE05), 2005 (to appear).
- Translator (written in Java) from an OTS/CafeOBJ specification into Java classes
- Formalized version of the translator and proof in CafeOBJ