Programming environment for the incremental software creation methodology based on data reification

Takashi ISHIKAWA

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

Februaty 13, 1998

Keywords: refinement, abstract interpretation, incrementa programming, programming environment.

Stepwise Refinement is a well-known effective method for developing a large and complex software. In traditional methods, a software is refined from its function point of view and refinements are only applied only on early stages of the project. In consequence, data defined at each step are not enough concreate for program execution. It is essential to execute the program in the intermediate stages otherwise it is difficult to find out potential errors and to refine further.

Yoshioka proposed a method: Incremental Software development method based on Data Reification (ISDR), in which a program is refined from its data reification point of view and can be executed using abstract interpretation. ISDR consists of two phases:

1. an abstraction phase, in which program inputs and outputs are abstracted into single values and the program is considered as a function defined on those values, named a primitive program,

2. a refinement phase, in which repeatedly we refine the primitive program by reification of its values until the data is refined enough to satisfy its requirements.

One of the advantages of ISDR is that we can interpret intermediate programs in which function are not defined completely, many errors can be detected at an earlier stages of software development than those in traditional methods.

To execute the intermidiate program, ISDR defines formal interpretation of it. Although there is no environment that help us to develop and execute programs based on

ISDR. In this paper, we propose a support environment for ISDR. Some examples with this environment are shown for demonstrating and consider efficiency of ISDR method.

We designed a language and implements whose interpreter in order to apply this method to some large practical problems. This environment consists named Alchemy of an interpreter for abstract programs should be reified.

Alchemy supplies functions for implementation as follows:

1. management revision.
2. define a data-domain and reify it.
3. edit program codes.
4. interface between AL and users.
5. guide for refinement.
6. suggestion for way of refinement functions.

ISDR assumes that we get a complete program when a series of the refinement finishes. We have to cancel some parts of the definitions if the specification are changed in intermediate stages. Programs can re-refined as as new branch and old definitions are saved as past. We extend a concept of versioning to revision in Alchemy. We can create a branchat any place in the version tree. Alchemy requires stronger constraints than ISDR, because consistency should be kept in any successive versions of programs and data domains through the refinement stages.

Finally, we demonstrate ISDR method using efficiency some middle-scale programs in AL and Alchemy.