Conceptual Issues of an Object-Centered Process Model

Makoto Matsushita[†] Makoto Oshita[†] Hajimu Iida[‡] Katsuro Inoue[†]
matusita@ics.es.osaka-u.ac.jp oshita@ics.es.osaka-u.ac.jp iida@itc.aist-nara.ac.jp inoue@ics.es.osaka-u.ac.jp

[†]Osaka University
Dept. of Information and Computer Sciences
Faculty of Engineering Science
1-3 Machikaneyama, Toyonaka
Osaka 560 JAPAN

[‡]Nara Institute of Science and Technology
Information Technology Center
8916-5 Takayama, Ikoma
Nara 630-01 JAPAN

Abstract

In this paper, we propose an object-centered software process description model. We also present the idea of software development management environment based on the model. To use this model and environment, we illustrates software development environment as it is, and provides a framework for software process description, management, and improvement.

1 Introduction

Software process description and its enaction help the software development to proceed effectively and to produce high quality software. However, most of process-centered software engineering environments tend to enforce specific types of development activities to the developer. Also, they require proprietary and exclusive systems/environments which are completely different from existing software development environment [2, 3]. Therefore those systems are not yet widely used in real software development.

Moreover, most of these software process description languages[1] focus on the description of "how to produce a product"; i.e., a procedure of software development. However, recent software development methods such as object-oriented programming, software reuse, component-based programming mainly focus on "what should be made"; i.e., artifacts in software development environment. In process-centered software engineering environment, these artifacts-centered idea of software development should be supported, to make more effective support for software development.

In this paper, we propose a new object-centered software process description model. We also present the idea of software development management environment based on the model.

The major contributions of object-centered process model and its environment are as follows:

- Granularity of the object representation is easily managed by using the object grouping and inheritance features. Multi-grained operations are established straightforward.
- Any partial information within an object easily extracted.
- The representation of the described objects is flexibly changed corresponding to the change of the project structure.
- System is easily installed to existing development environments.

2 Object-Centered Process Model

The model consists of a set of objects, which represent artifacts and resources in the software development, such as design documents, compilers, developers, and so on. The goal of the model is to illustrate software development environment as it is.

2.1 Object Overview

An object consists of attributes and methods. An attribute represents characteristic of the object. An attribute has an attribute label and an attribute value. An attribute label is a unique name for attribute and it indicates what kind of information is needed to the object. Information itself is represented by the attribute value. The type of the attribute values is a number, string, label, or list of these types.

A method is a function applied to the objects. A method has a method label and a method function. The method label is a unique name for this method function, showing what operation is done with this method. Actual operation is defined by the method function. Operations of the method functions are: operation to the information in the objects, operation to the outside of the model such as tool invocation, typical numerical, literal, and collection operation found in a programming languages, and so on.

2.2 Software Development Process

We introduce a status object, which represents a status of software development environment in some viewpoint. A status object is composed of a set of partial object, which is defined as an object composed of a subset of attributes and methods of a certain object. They represents typical characteristics of an object in some viewpoint. Then, software development process is defined as a transition sequence of those status objects.

2.3 Features

The model provides has various features to support project management, software development, and cooperation for developers.

- Reference scope of objects
  Each object can set a scope of reference with object own attribute. This is achieved by object grouping and corresponding permission for the group. The permission represents allow/deny operation to the object.

- Operation history
  Any operation (referring an attribute value, invoke a method, and so on) to all objects are automatically recorded as history. The history is stored in an attribute of objects, and it is referred by the methods.

- Object creation and inheritance
  Object creation is done by inheriting attributes or methods from any existing object or predefined skeleton objects. The inheritance is accomplished by copying or referring attributes value and method functions from existing objects, and by modifying or adding attributes and methods to a new object.

There is no "class" feature found in common object-oriented language in the model. However, we use an object just like a class. For example, at first we create an object of "generic source code", then create an object of "source-code.c" or whatever object for source code file from "generic source code" object. Since it may be complex to describe objects in the software development environment, we think it is not suitable to employ recent object oriented programming language.

3 Design Overview of System

The system allows to the developers to operate object, execute a method to develop a software, invoke tools for doing an activity, and so on. The system also allows project managers to browse the whole software project, store the progress information, show what is done in the project, and other helpful information about software development environment. We have already made a prototype system, and it is used with an example process.

4 Conclusion

In this paper, we propose a new object-centered process management model. With this model, software process is illustrated clearly; that is powerful capability of the process management. We also designed the software development environment based on the object-centered process model. A prototype system of the system has been implemented.

As a further work, a full implementation of system has to be completed. Validation of the model and more support for process enaction based on validation is also planned.

References

[1] P. Armenise, S. Bandinelli, C. Ghezzi, and A. Morzenti. Software Process Representation Language: Survey and Assessment. In *Proceedings of the 4th Conference of Software Engineering and Knowledge Engineering*, pages 455–462, 1992.

[2] S. Bandinalli, E. Nitto, and A. Fuggetta. Supporting Cooperation in the SPADE-1 Enviornment. *IEEE Transaction on Software Engineering*, 22(12):841–865, 1996.

[3] I. Ben-shaul and G. Kaiser. A Paradigm for Decentralized Process Modeling and its Realization in the Oz Environment. In *Proceedings of 16th International Conference on Software Engineering*, pages 179–188, 1994.