Object-Oriented Modelling of Software Processes and Artifacts: Promises and Challenges

Jun Han
Peninsula School of Computing and Information Technology
Monash University, McMahons Road, Frankston, Vic 3199, Australia
phone: +61 3 99044604, fax: +61 3 9904124, e-mail: jhan@monash.edu.au

1 Introduction

Over the years, there have been significant efforts in modelling and representing software processes and artifacts, especially in the software engineering environments community, eg. [6, 4, 5]. It has long been recognised that focusing on the artifacts alone does not deliver the full benefit for software practice since software development should follow an orderly process. On the other hand, considering processes without much regard to the artifacts is also limited, eg. in dealing with fine-grained technical processes. As such, a promising direction is to integrate the modelling and representation of software processes and artifacts. Encapsulation of data and behaviour is one of the key characteristics of the object-oriented approach. As such, it seems natural to use the object-oriented approach to model software artifacts and processes in an integrated manner. Due to the specific nature of software processes and artifacts and the limitations of the object-oriented approach, however, object-oriented modelling of software processes and artifacts still requires much investigation.

In this paper, we outline some of the issues, promises and challenges in developing integrated models of software processes and artifacts in the object oriented approach. It also aims to achieve uniform treatment of fine-grained processes and high-level processes. The discussion is set in the context of our preliminary work in integrating software processes and artifacts [3], based on our approach to representing software artifacts [1, 2], and oriented towards full-fledged modelling and representation of software processes and artifacts.

In section 2, we briefly discuss some of the basic characteristics of software processes and artifacts. Then in section 3, we contemplate an object-oriented approach to modelling software processes and artifacts, and identify the relevant issues, promises and challenges.

2 Characteristics of Software Processes and Artifacts

The software process in general relates to all activities involved in the development and maintenance of a software system. The process has different characteristics in different phases and at different granularity levels. While acknowledging that there are other classifications, here we distinguish two types of process:
1. *fine-grained process* concerning the manipulation of an artifact's content, and
2. *high-level process* concerning the scheduling and coordination of activities performed on one or more whole artifacts.

It should be noted that an atomic activity in the high-level process may correspond to a complicated fine-grained process. In general, there may be hierarchical organisations of activities where higher-level activities are lower-level processes.

Software artifacts are the products of the software process, and their structures and contents are determined by the project. Therefore, software processes and artifacts should be treated together with equal importance. Besides, it is felt that the prominent role that tools take in some of the existing process modelling approaches is unnecessary and rather a hindrance to properly address certain issues such as fine-grained processes.

The next section will introduce some additional characteristics of software processes and artifacts, including views, active artifacts, informal communications, and modelling styles.

## 3 O-O Modelling of Software Processes and Artifacts

In modelling software processes and artifacts, we focus on the following aspects: fine-grained processes, high-level processes, and artifacts reflecting necessary process features. We aim to achieve integrated modelling of processes and artifacts, and uniform treatment of fine-grained and high-level processes. As mentioned above, the object-oriented approach seems natural in meeting these objectives.

In general, the object-oriented approach suggests that we model artifacts and their manipulation (ie. process) in an integrated manner. That is, the definition of an object in the model should involve the structural and behavioral aspects of the artifacts.

**Artifact Structures.** The structural formulation of an artifact should take into account the following factors: its components, the relationships among the components, and the consistency constraints about the components and their relationships. As such, this formulation presents a static definition of what a *consistent* artifact of this kind is. To a certain degree, the captured relationships form a static reflection of some process characteristics such as traceability of the process [3]. The consistency constraints are additional information about the artifact structure. They further restrict the space of the artifacts that can be constructed, and can assist the developer in understanding and constructing the artifacts. A corresponding model for artifact structures has been proposed in [1].

The current object-oriented approach can readily cope with the definition of components. However, the support for the specification of relationships and constraints are somehow limited, ie, they are not treated as first class. We believe that first-class treatment of relationships and constraints in modelling is necessary and beneficial.
**Fine-Grained Processes.** A fine-grained process deals with the manipulation of an artifact. It should be defined in terms of the operations that can be applied to the artifact structure, and the (sequencing) rules that govern their application. In general, it is the developer who drives the manipulation process of a specific artifact by applying the operations as required. However, rules do exist regarding the applicability of operations at a given time, and the sequencing of the operations. While operations can be readily defined, the rules can not be easily specified in a formal notation in the current object-oriented approach (although there are notations like state transition diagrams).

**High-Level Processes.** High-level processes focus on scheduling and coordination of activities performed on whole artifacts, and should be defined based on the group of relevant artifacts. The grouping of the artifacts together with their relationships and constraints forms the structure of the high-level object that the high-level process applies to. The high-level process is defined in terms of the activities performed on the component artifacts, and the (sequencing) rules that govern the individual activities. As stated earlier, the current object-oriented approach does not provide effective support for the specification of process rules.

Among the artifacts manipulated by a high-level process may be a process-oriented artifact, such as a project plan. The process may manipulate the internal components of such artifacts, e.g., making changes to a project plan. An operation applied to a process-oriented artifact may not only cause changes to its content, but also cause "executional" effects to the project. That is, the content change to the artifact will be interpreted (or executed) in the project in the sense that it effects process changes. (But there is no such effect from content changes of an ordinary artifact such as a design document.) Because of their executional effects, we call the process-oriented artifacts active artifacts. In addition, relationships originating from active artifacts may also have executional effects, and are consequently called active relationships. In general, it is not clear how to naturally model active artifacts and relationships in the current object oriented approach.

As mentioned earlier, object oriented modelling of artifacts and processes may take a hierarchical form. One level provides the context (e.g., resources) for the next lower level. The contextual relationships between the levels need to be further investigated. Besides, the top-most level should give a high-level view of the project and how it starts and progresses.

**Further Issues.** Following the grouping of a number of artifacts into a high-level object in relation to a high-level process, it is possible for an artifact to appear in a number of high-level objects in the sense that each object context offers a specific perspective of the artifact from the relevant high-level process viewpoint. The consistency and relationships among the perspectives of an artifact need to be further investigated.

Another issue of interest is the process modelling style. One may model the explicit sequencing relationships of activities and operations in an imperative
manner. For implicit sequencing based on status or content change of artifacts, it seems more natural to use an event-based approach. How to naturally integrating these two styles poses another challenge for process modelling.

Process modelling in general should also deal with the issues of informal discussions and negotiations in the context of supporting coordination and communication. How to handle these issues in the object oriented approach presents yet another challenge.

4 Conclusions

In this paper, we have briefly analysed a number of issues in using the object-oriented approach to model software processes and artifacts. The major issues identified include: relationships and constraints as part of artifact structures (to capture process characteristics), fine-grained processes, high-level processes, active artifacts and relationships, hierarchical multi-perspective modelling, process modelling styles, and the treatment of different forms of coordinations and communications. In general, the object-oriented approach provides a way to integrate process and artifact modelling and to treat fine-grained and high-level processes in a uniform manner. While some modelling requirements can be readily met by the current object-oriented approach, others are identified as requiring further investigation.

Acknowledgments. The author would like to thank Se-Il Choi, Hardeep Singh, Jim Welsh and Yun Yang for their discussions in formulating some of the views expressed in this paper.

References