December 1999

Problem and Solution Frameworks for Software Development Process Modeling,

Daniel Turk
Colorado State University

Vijay Vaishnavi
Georgia State University

Problem and Solution Frameworks for Software Development Process Modeling

Daniel E. Turk, Colorado State University, turk@lamar.colostate.edu
Vijay K. Vaishnavi, Georgia State University, vvaishna@gsu.edu

Abstract

Frameworks are valuable constructs in the development and growth of scientific disciplines. Frameworks identify important dimensions that should be studied, and guide researchers in appropriate techniques for problem-solving. Young disciplines frequently lack frameworks, since they have not developed sufficiently enough to have the maturity of well-thought-out frameworks. But as disciplines do mature, frameworks naturally result.

This paper presents and discusses two frameworks for use in software development process modeling. These frameworks organize in one place many of the problems and successful solution approaches identified over the past several years in the process modeling community[i], identifying general dimensions of the process modeling problem, and general dimensions that should be considered in any potential process modeling solution approach. In addition, several important problems and potential solution approaches that, to date, have received minimal focus are proposed and included in these frameworks.

Introduction

Numerous attempts have been made to define frameworks for software development process modeling. These frameworks generally focus on the specific constructs that must be represented in a process modeling system. Conradi et al. (1992) describe several sub-models for a process model: activity/task, product, tool, organization, and user. The Workflow Management Coalition (WfMC 1994) describes activities, roles, types, data, applications, and transition conditions. Curtis, Kellner, & Over (1992), however, describe more general dimensions that should be covered in a software development process modeling system: functional, behavioral, organizational, and informational.

Most frameworks of this type describe specific process modeling constructs and the interrelationships between them, and thus might better be termed process meta-models. While such meta-models are valuable and important to have, two other types of frameworks are valuable before building a process meta-model, and, subsequently, a process modeling system: one for the problems, and one for potential solution approaches.

This paper presents two such frameworks and discusses benefits associated with each one. These frameworks organize in one place many of the problems and successful solution approaches identified over the past several years in the process modeling community, identifying general dimensions of the process modeling problem, and general dimensions that should be considered in any potential process modeling solution approach. In addition, several important problems and potential solution approaches that, to date, have received minimal focus are proposed and included in these frameworks. It is upon these frameworks that sound process meta-models and modeling systems can be built.

SWD Process Modeling Frameworks

There are two general software development process modeling frameworks: one that organizes the types of problems that a software process model is intended to address, and one that organizes the types of solution approaches that may be helpful to apply when creating a software process model.

A Problem Framework

It is essentially impossible to describe a process without describing its activities, and for most processes to be valuable there must also be a description of the products (deliverables) that are used and produced by these activities. If a process is a partially-ordered sequence of activities intended to reach a goal, and a goal is intended to do, accomplish, or produce something, then the deliverable can be considered the realization of what the goal is trying to accomplish. Thus, activities and deliverables are *core issues* in the process modeling problem domain, and should thus be included in any framework that describes the problem(s) to be addressed by process modeling systems. *Goal issues* (goal specification, mapping between goals and activities and deliverables, and outcome measurement and assessment) are likewise inherent to fully describing processes and are a key part of a process modeling problem framework as well.

As soon as we focus on the activities required to accomplish a goal, the question of what order to carry out these activities arises. What is the sequence in which the activities should be performed? What are the interdependencies between multiple activities, multiple deliverables, and between activities and deliverables? As soon as we address sequencing and dependencies, we quickly realize that there are multiple types, loci, and levels of control that may come into play, and that there may be different types of rules that can be used to describe these dependencies, etc. These issues may be broadly categorized as *sequencing / rule / constraint issues*.

Core Process Modeling Issues:
• Activities
• Deliverables
• Process + Product
Sequencing / Rule / Constraint Issues:
• Activity Sequencing
• Levels of Control / Description
• Locus of Control
• Types of Control
• Dependencies
• Rule Types
Goal Issues:
• Goal specification • Mapping of goals to activities and/or deliverables • Outcome measurement and assessment
Process Improvement Issues:
• Continual Change
• Dynamic / Flexible / Changeable
• Feedback / BPR
Enactment Issues:
• Actors
• Tools
Miscellaneous Issues:
• Process Visualization
• Defined
• “Factored”

Figure 1: Problem Framework

Because real organizational processes continually change, rarely remaining static for very long, process improvement and continual change are important issues to remember. The broad dimension of *process improvement* encompasses these issues.

Typically a process is enacted by both manual and automated means. Various human actors carry out some parts of a process, while automated tools are used to perform other parts. The *enactment dimension* recognizes these process modeling problem issues.

Finally, there are a variety of *miscellaneous issues*: How are processes graphically visualized? How are processes precisely defined? How do we “factor” process modeling constructs so that they remain relatively uncoupled?

The process modeling domain addresses and must deal with issues from each of these dimensions. Process models frequently ignore one or more of the issues identified here[ii], focusing on a select set of issues at the expense of others. However, for real-world processes to be modeled effectively, we argue that all these issues should be given attention. Figure 1 summarizes our Problem Framework.

A Solution Framework

Given the set of issues described in the problem framework above, what, then, are some broad approaches to defining and modeling software processes? What are some general principles and techniques that – at the very least – should be given consideration when creating a software process model.

Models:
• Meta-Model (Data Dictionary)
• Requirements Model
• Domain Model(s)
• Application Model(s)
Concepts:
• Object-Orientation
• Procedure- and Rule-based
• Goal-based (GQM)
• Multi-level
• Process Improvement
• Open Meta-Model

Figure 2: Solution Framework

First, just as it is valuable to create a broad set of *models* during the process of actual system development, it is valuable to create a similar set of models when a software process model and a software process modeling system is under development. Typical models that have been found to be useful are a meta-model or data dictionary, a requirements model, one or more domain models, and one or more application models. Some contemporary realizations of these types of models include Use Case, Object, Interaction, State, Data Flow, and ER diagrams.

Second, there are a number of *concepts* that have helped various process modeling systems be successful or have been suggested as valuable to include. These include object-orientation (to support inheritance, etc.), hybrid specification (procedural as well as rule-based), goal support (e.g. GQM), multi-level (in order to describe activities, goals, and processes at multiple levels of detail and abstraction), process improvement (to provide direct process improvement constructs into process models), and use of an open meta-model (so that the underlying constructs are clear and obvious, and so that they may be more readily changed and adapted as necessary).

These broad solution approach dimensions are summarized in Figure 2, the Solution Framework.

Benefits of Frameworks for SWD Process Modeling

The concept of iterative incremental development has gained strong support in recent years, being driven by the object-oriented world but being accepted and adapted by the traditional structured development world as well. This paper has alluded to the idea that iterative incremental development is a valuable approach to process, process model, and process modeling system development as well, because of the inherent change continually experienced.

One major benefit that should accrue from first developing a general process modeling problem framework is that the problems of the process modeling domain will be more readily understood. This truism may seem obvious, but, currently the process modeling field does not have any broad, well laid out framework of the major issues that need to be addressed. The problems and general dimensions identified in this paper may be a first attempt at such a problem framework.

By identifying potentially useful solution approaches, a process modeling solution framework can help focus the activities of researchers, and help the process modeling community be more aware of the types of approaches that are currently being used or are presently under investigation. Such a framework can also be used to help developers identify diverse solution approaches to include within a given process modeling system. By including more than one solution approach, process modeling systems should become more flexible and capable of supporting a wider variety of process issues.

By identifying a broad set of relevant issues, problems, and potential solution approaches, researchers and developers may obtain guidance from frameworks and may be less likely to overlook important issues and potential solutions. We hope that the problem and solution frameworks presented above will lead in this direction.

References

Arbaoui, Selma; & Oquendo, Falvio. (1994). "Goal Oriented vs. Activity Oriented Process Modelling and Enactment: Issues and Perspectives." *3rd European Workshop on Software Process Technology* (EWSPT'94), Feb 7-9, 1994, Villard de Lans, France.

Barghouti, Naser S.; & Kaiser, Gail E. (1992). "Scaling Up Rule-Based Software Development Environments." *International Journal of Software Engineering and Knowledge Engineering, 2:1* (March), pp. 59-78.

Basili, Victor R. (1992, September). "Software Modeling and Measurement: The Goal/Question/Metric Paradigm." (Computer Science Technical Report Series CS-TR-2956, UMIACS-TR-92-96). College Park, MD: University of Maryland.

Conradi, Reidar; Fernstrom, Christer; Fuggetta, Alfonso; & Snowdon, Robert. (1992). "Towards a Reference Framework for Process Concepts." *2nd European Workshop on Software Process Technology* (EWSPT'92), Sep 7-8, 1992, Trondheim, Norway.

Curtis, Bill; Kellner, Marc I.; & Over, Jim. (1992). "Process Modeling." *Communications of the ACM, 35:9* (September), pp. 75-90.

Kellner, Marc I.; Briand, Loic; & Over, James W. (1996). "A Method for Designing, Defining, and Evolving Software Processes." *Fourth International Conference on the Software Process* (ICSP'96/ICSP4), Dec 2-6, Brighton, UK.

Sutton, Stanley M., Jr.; & Osterweil, Leon J. (1997). "The Design of a Next-Generation Process Language." *Proceedings of the 6th European Software Engineering Conference* (ESEC'97); 5th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE'97), Sep 1997, Zurich, Switzerland, pp. 142-158.

Workflow Management Coalition. (1994). *WfMC Workflow Reference Model*. Downloaded version 1.1, dated 29-Nov-94, from www.aiim.org/wfmc October 1997.

[i] The frameworks presented in this paper draw from many published articles (including Arbaoui & Oquendo 1994; Barghouti & Kaiser 1992; Basili 1992; Conradi et al. 1992; Curtis, Kellner, & Over 1992; Kellner, Briand, & Over 1997; Sutton & Osterweil 1997), but there are numerous others that we cannot reference directly here because of space limitations.

[ii] Business-level *goals* and software *process improvement* have been discussed in the literature, but have not yet been directly included in software process meta-models and process modeling systems. Thus, by including them in our frameworks here, we emphasize their importance and the need to focus more research on them.