

# A UML-Based Process Meta-model Integrating a Rigorous Process Patterns Definition

Hanh Nhi Tran<sup>1</sup>, Bernard Coulette<sup>1</sup>, and Bich Thuy Dong<sup>2</sup>

<sup>1</sup> University of Toulouse 2 -GRIMM

5 allées A. Machado F-31058 Toulouse, France  
{tran, coulette}@univ-tlse2.fr

<sup>2</sup> University of Natural Sciences, VNUHCM

227 Nguyen Van Cu, Q5, HoChiMinh Ville, Vietnam  
thuy@hcmuns.edu.vn

**Abstract.** Process Pattern is an emergent approach to reuse process knowledge. However, in practice this concept still remains difficult to be exploited due to the lack of formalization and supporting methodology. In this paper, we propose a way to formalize the process pattern concept by introducing it into a process meta-model. We provide a general definition to cover various kinds of process-related patterns in different domains. We define rigorously process concepts and their relations to allow representing processes based on process patterns and to facilitate the development of supporting tools. By distinguishing process patterns at different abstraction levels, we aim to develop a systematic approach to define and apply process patterns.

## 1 Introduction

Recently, process patterns approach has been adopted by process communities to capture and reuse proven processes. The expected interests of reduced development time and improved process quality make process patterns approach attractive. However, up to now works on process pattern have not been enough developed to permit applying this concept efficiently. Firstly, it lacks a definition that can cover the diversity of process patterns (c.f. [8][9] for more detailed discussions on process patterns definitions and taxonomy). Secondly, the representation of process patterns is not adequately formal to be directly reused in process modeling. Finally, a process patterns based methodology for developing processes is still needed to guide process designers applying process patterns systematically.

To cope with these issues, we believe that the first step is to formalize process patterns. This objective can be achieved by introducing the process pattern concept into a process meta-model. We present in this paper a UML-based meta-model integrating the concept of process pattern into the description of software processes. Our meta-model provides rigorous concepts to describe process patterns and processes based on process patterns. The meta-model will be presented in the following section and exemplified in section 3. Related works and contributions are summed up in the final section.

## 2 A Meta-model for Representing Process Patterns

Our objective is to define process patterns as patterns for modeling processes. Thus, we provide a wide definition to cover all the related notions.

We introduce multi-abstraction levels<sup>1</sup> into process representation to promote process reuse. The relations between process models at different abstraction levels are defined in order to clarify the way of defining process models (and then process patterns which capture them). To allow process modeling based on patterns and facilitate patterns organization, we also define explicitly the aspects of process pattern concept as well as the relationships among patterns, between process patterns and process models. To attain the requirement on standardization, initially we aimed at integrating the process pattern concept into the software process meta-model SPEM 1.1[4]. However, SPEM 1.1 has some deficiencies and is progressing towards a next version [5]. So, we decided to develop a meta-model which is strongly inspired by SPEM 1.1 for process description, but based directly on UML<sup>2</sup>.

To present our meta-model, we use UML class diagrams<sup>3</sup> for the abstract syntax. The semantics is expressed in natural language and reinforced by OCL expressions<sup>4</sup>.

### 2.1 ProcessModel

- This concept is used to describe (part of) a process. It is composed of process tasks (*Task*), the required products (*Product*) and the participating roles (*Role*) of these tasks (Figure 1a).

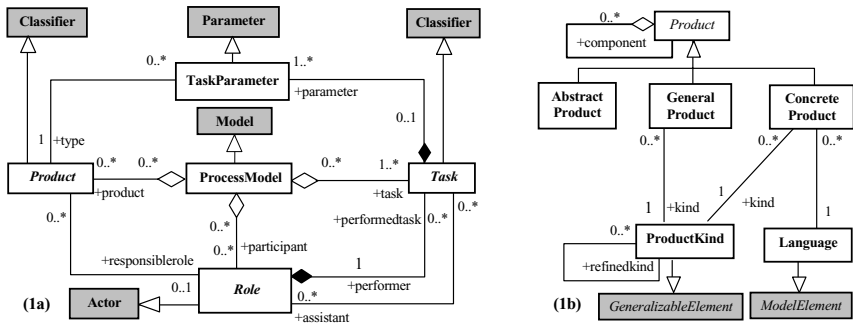


Fig. 1. Meta-model for *ProcessModel* and *Product*

*Product* is an artifact created, consumed or modified during the development. A *Task* is a unit of work realized to create or modify products. Necessary products for the execution of a task are described explicitly as its parameters (*TaskParameter*). A *Role*

<sup>1</sup> The abstraction level of a process model reflects how detailed its content is described.

<sup>2</sup> In our meta-model, the part describing processes (which is inspired by SPEM 1.1) is defined separately with the part introducing process patterns. Thus, in the future, we just need to change the first part to conform our meta-model to the stable version of SPEM.

<sup>3</sup> Light grayed classes represent ones from UML; white classes represent new defined concepts.

<sup>4</sup> Due to the space constraint, we cannot present here the OCL well-formedness rules.