Process-Centered Model Engineering

Erwan Breton
Société Soft-Maint
4, rue du Château de l'Eraudière, BP 588
44074 Nantes cedex 3, France
ebreton@sodifrance.fr

Jean Bézivin
LRSG, University of Nantes,
2, rue de la Houssinière, BP 92208
44322 Nantes cedex 3, France
Jean.Bezivin@sciences.univ-nantes.fr

Abstract

A new information system landscape is emerging that will be more model-centered than object-oriented, characterized by many models of low granularity and high abstraction. These models may describe various aspects of a system such as products, structural organization and processes. This latter category may be refined in manufacturing, software and business processes for example. Each of these sub-domains share common concerns such the definition of work items, performers, inputs and outputs. However, as they may be represented using separate meta-models, there is therefore a risk of facing an uncontrollable number of meta-models that all redefine similar modeling constructs. It would be a wise initiative to put all these contributions together and to study what may be common and what should stay specific. In this paper we underline some similarities between process meta-models and we sketch out how they could be organized.

1. Introduction

Process management is today a core concern for software engineering. It promises a better quality of final products, an optimization of costs and delays, a faster adaptation to the new needs of the market, etc. After the adoption of a unified description formalism (i.e. UML [7]) by the OMG, the second stage has consisted in defining a common unified object-oriented analysis and design method, which has led to SPEM.

As meta-models define a specific viewpoint, a lot of process meta-models have been developed to address a specific domain. Indeed a manufacturing process meta-model and a software process meta-model may have different concerns. For example the manufacturing process meta-model may include real-time aspects, which may be missing in a software process meta-model. Process meta-models may also differ by their objectives. We may distinguish the pure act of modeling organizational behavior (descriptive modeling) from the act of constructing executable systems from these models (active modeling). We are more interested here in descriptive modeling than in active modeling although the boundary is not always very well-drawn between these domains. In any case, descriptive modeling is always a prerequisite to active modeling.

There are important differences between modeling techniques. For example some are fine-grained while other are more coarse-grained. Usually a modeling technique can be associated to an ontology for the purpose of abstraction and sharing. Abstraction means that the ontology defines what should be filtered out from a given situation. Sharing means that there is a collective agreement on the shared meaning of the various concepts (ontological agreement). On practical grounds, an ontology may be assimilated to a meta-model, constituted of conceptual categories and relations between them. To give a naive example, we may consider the simple ontology of Petri nets, composed of the concepts of *Place* and *Transition* together with usual relations between them. Another example in the domain of process modeling could be an actigram IDEF0 ontology.

So, a significant part of the behavioral information captured from a system can be stated in a process model. There is an infinite number of ways for constructing a model of a system, according to the intended purpose of the extracted model. As a consequence, there is also an infinite number of meta-models characterizing these modeling perspectives. We are interested in the practical applicability of process meta-modeling techniques.

The variation in the needs of the modeler are very large and we need to provide adapted techniques. This means that we cannot only offer the choice between Petri nets and IDEF0 actigrams for example. Each process model has its own characteristics. A workflow model is different from an ABC management model (Activity-Based Costing). One of the areas we are mainly interested in, is software process. Even in this area, the differences of perspective are quite important according to the interest, or non-interest, one may place in resource allocation to tasks, in delay measurements, in risk analysis, in consideration of non-functional requirements, and so on.

It is thus becoming clear that, if we want the process model engineering techniques to be practicable, we need to provide a framework to define, combine, use and reuse a set of generic process meta-models. This is the goal we have in the present investigation.

2. Organizing process meta-models

All types of processes share a common set of recurrent concepts:

- The work items that make up the process.
- The rules that constrain the execution of these work items.
- The objects (i.e. physical or human resources, information, products, tools, applications, etc.) that may be used or created while performing a work item.

For example, such elements may be found in the new SPEM [1] (Software Process Engineering Management) proposal, which addresses the SPE (Software Process Engineering) RFP [8] (see Figure 1).

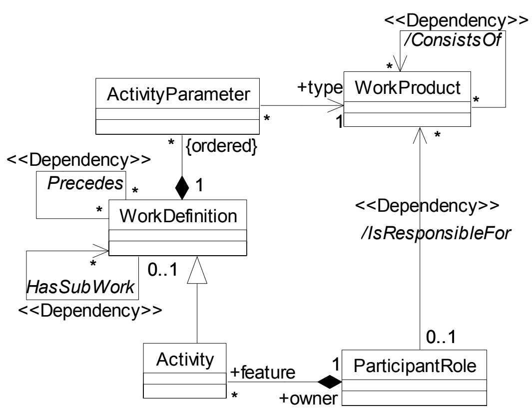

Figure 1: The *Process Structure* package of SPEM

WorkDefinitions and *Activities* define the work to be done. *WorkDefinitions* may be scheduled using *Precedes* dependencies. *WorkProducts* (i.e. piece of information, document, model, source code, etc.) may be produced, used or consumed. Finally an *Activity* is owned and performed by a *ParticipantRole*. As we may see the *Process Structure* package of SPEM contains classical process modeling constructs.

Since SPEM concentrates on software process modeling, it also introduces specific concepts such as *Phase*, *Iteration* or *Milestone* (the goal of a *Phase*). This is done in the *Process Lifecycle* package. Other process meta-models may focus on different aspects such as:

- Costs,
- Delays,
- Events,
- Resource consumption,
- Etc.

More than concepts, modeling patterns could also be shared between different process meta-models. The DSTC submission for EDOC proposes to separate the activity definition from its invocation, which allows an activity to be reused in different processes. Van der Aalst and al. [11] have identified some workflow patterns that cover some complex ordering of activities. It would be interesting that all process meta-models could benefit from such mechanisms instead of re-defining them.

PIF [4] (Process Interchange Format) and PSL [10] (Process Specification Language) already proposed some frameworks for process meta-model definition. These two proposals are today engaged in a merging process to integrate both business and manufacturing process concepts in a single ontology.

PIF stems from the need for various organizations (MIT, DEC, Stanford, etc) to share their process models. PIF core meta-model defines a more or less minimal set of entities. This basis set may be enriched using the extension mechanism of Partially Shared View (PSV). A PSV module inherits from the core module or from another PSV module (see Figure 2). It adds new local entities that specialize entities defined in the modules on which it is built.

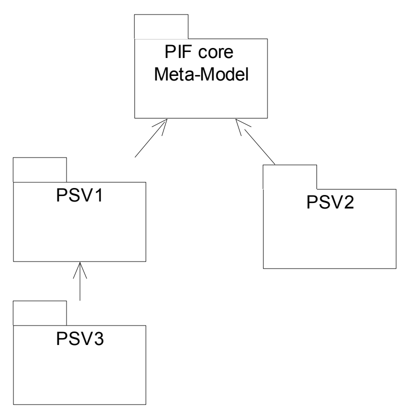

Figure 2: PIF Partially Shared View mechanism

PSL aims at defining standard ontology and format for exchanging manufacturing processes. PSL core ontology defines a process as a set of *activity* in which participate some *objects* at *timepoint*. In PSL, everything is an *activity*, an *object* or a *timepoint*. PSL core also introduces the concept of *activity _occurrence*. This minimal basis is refined in extension modules, defining for example non-deterministic *activities*, quantity for *objects* or temporal ordering on *timepoints*. Many predefined extension modules already exist (see Figure 3), each refining a single entity (e.g. the extension “Processor Actions” refines *activity*, the extension “Resource Pools” refines *object*, etc).

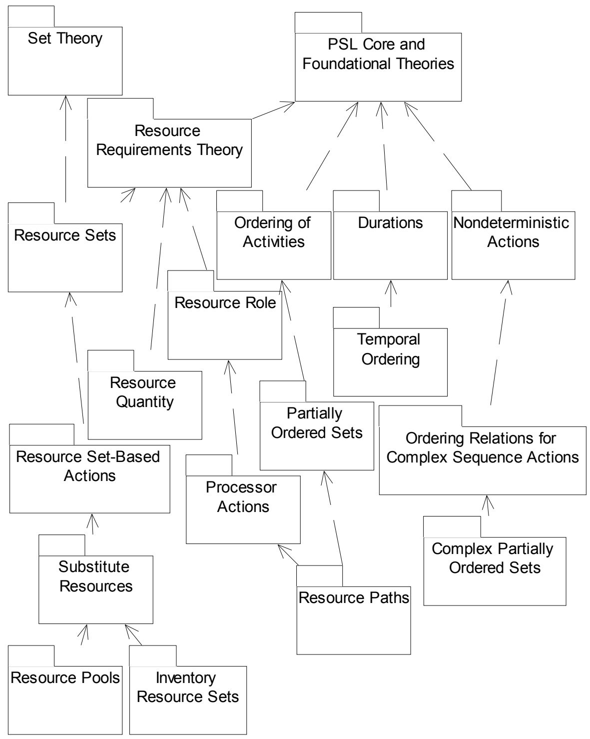

Figure 3: The modular architecture of PSL

A similar architecture could be defined within the MOF framework (see Figure 4). A generic process meta-model introduces common high-level concepts, such as activity, role, transition, etc. This core set is extended by domain-related meta-models, which may specialize the core concepts to address their specific needs. For example the SPEM meta-model would add the *Iteration* concept.

An architecture of this kind has already been defined in the MOF framework for organizing the products meta-models, i.e. UML and CWM [5] (Common Warehouse Metamodel). Both these meta-models are defined using separate packages. Some dependency relations stand between these packages. In this way many constructs may be reused. For example, the CWM Record package that defines the basic record and its structure depends on the UML foundation behavioural elements and model management packages and the CWM data types package. This package is reused for defining the COBOL Data Division package.

An important issue is the identification of the needed connections between process and product meta-models. Meta-modeling techniques may also provide some good answers for ensuring the coupling between meta-models. We have discussed it in other papers ([1], [2]).

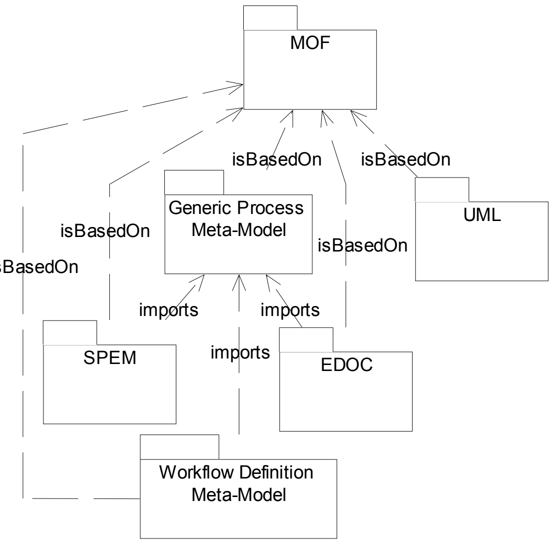

Figure 4: A hierarchy of meta-models

However, SPEM is not a meta-model but it has been defined as a UML profile, i.e. a set of stereotypes, tags and constraints added to the UML standard semantics (see [7], p. 407). In this way it benefits of UML existing features such as the activity diagram. A major drawback of such an approach is that SPEM is not a meta-model but a UML model as it is defined using UML concepts (*Stereotype*, *Tagged Values* and *Constraints*). Actually a SPEM model is a UML model that extends SPEM profile.

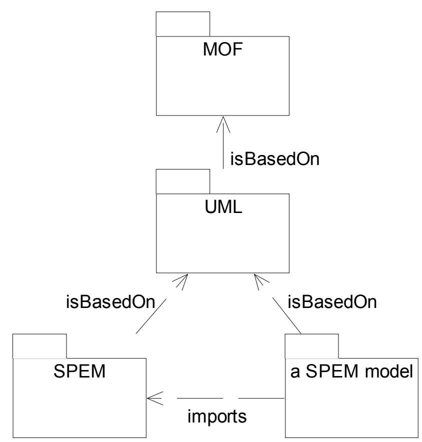

Figure 5: SPEM as a profile

The architecture presented Figure 4 may be defined using profiles (see Figure 6).

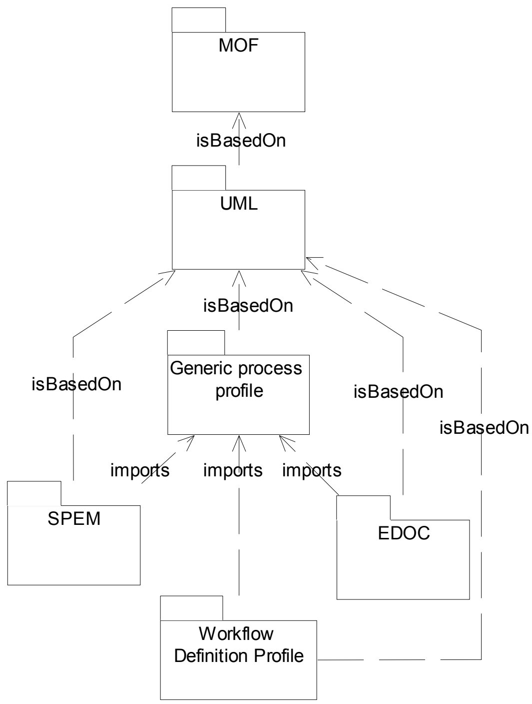

Figure 6: A hierachy of profiles

3. Conclusion

In this paper we have discussed the organization of process meta-models. The notion of a process can take many different aspects, and may involve many different attributes. There is however a commonly accepted idea that the variety of definitions and applications is referring to some common framework, yet to be precisely identified.

All process meta-models are built on a core set of concepts, which are always more or less the same (work items, products and resources). In addition to these common modeling constructs they integrate domain-related concepts, which are specifically dedicated to the needs of expression of the user.

As long as they were expressed using heterogeneous formalisms, it was difficult to observe the similarities between two proposals. With the standardization of MOF meta-meta-model all these meta-models may be described using the same formalism, which allows them to be put in relation and comparatively discussed using a common basis. In this way recurrent modeling patterns may be identified and factorized in generic meta-models. Therefore we may see an organization of process meta-models emerging. Such an organization could be hosted within the MOF framework. This could be achieved either using standalone meta-models or using profiles.

After the definition of SPEM, the OMG is now investigating the fields of business processes and workflow. It is thus urgent to rationalize the process meta-models architecture. If not we are running the risk to be faced with an uncontrollable number of slightly different process meta-models.

4. References

[1] Bézivin, J., Bouchet J.P. & Breton E. "Revisiting the P&P Pattern with Explicit Meta-Models" INCOSE/LA Spring Conference, Pasadena, April 1999.

[2] Breton E., Bézivin J. "Model Driven Process Engineering" to appear in COMPSAC 2001 Conference, Chicago, October 2001.

[3] IBM, Rational Software, Fujitsu/DMR, SOFTEAM, Unisys, Nihon Unisys Ltd., Alcatel, Q-Labs "The Software Process Engineering Management (SPEM) Revised Submission", OMG Document ad/2001-03-08, April 2001.

[4] Lee J., Grunninger M., Jin Y., Malone T., Tate A., Yost G. & other members of the PIF Working Group "The PIF Process Interchange Format and Framework Version 1.2" The Knowledge Engineering Review, Vol. 13, No. 1, pp. 91-120, Cambridge University Press, March 1998.

[5] OMG "Common Warehouse Metamodel (CWM) Specification" OMG Document ad/2000-01-10, February 2000.

[6] OMG "Meta Object Facility (MOF) Specification" OMG Document ad/97-08-14, September 1997.

[7] OMG "OMG Unified Modeling Language Specification version 1.3" OMG Document ad/99-06-08, June 1999.

[8] OMG "Software Process Engineering Request for Proposal" OMG Document ad/99-11-04, November 1999.

[9] OMG "Workflow Process Definition Request for Proposal" OMG Document bom/99-10-03, November 1999.

[10] Schlenoff C., Knutilla A. & Ray S. "A Robust Process Ontology for Manufacturing Systems Integration" Proceedings of 2nd International Conference on Engineering Design and Automation, Maui, Hawaii, August 7-14, 1998.

[11] Van der Aalst W.M.P., Barros A.P., ter Hofstede A.H.M., Kiepuszewski B. "Advanced Workflow Patterns" 7th International Conference on Cooperative Information Systems (CoopIS 2000), volume 1901 of *Lecture Notes in Computer Science*, pp 18-29. Springer-Verlag, Berlin, 2000, http://tmitwww.tm.tue.nl/staff/wvdaalst/Publications/p105.pdf.