

## Informe Técnico / Technical Report



### A Comparative Analysis of SPEM 2.0 and BPMN 2.0

Mario Cervera, Manoli Albert, Victoria Torres, Vicente Pelechano



Ref. #:	ProS-TR-2012-03				
Title:	A Comparative Analysis of SPEM 2.0 and BPMN 2.0				
Author (s):	Mario Cervera, Manoli Albert, Victoria Torres, Vicente Pelechano				
Corresponding author (s):	Mario Cervera, <a href="mailto:mcervera@pros.upv.es">mcervera@pros.upv.es</a> Manoli Albert, <a href="mailto:malbert@pros.upv.es">malbert@pros.upv.es</a> Victoria Torres, <a href="mailto:vtorres@pros.upv.es">vtorres@pros.upv.es</a> Vicente Pelechano, <a href="mailto:pele@pros.upv.es">pele@pros.upv.es</a>				
Document version number:	1.0	Final version:	Yes	Pages:	6
Release date:	May 2012				
Key words:	SPEM, BPMN, Comparative Analysis, Method Engineering				

This technical report presents a comparative analysis between the SPEM 2.0 [4] and BPMN 2.0 [5] standards. The comparative analysis is based on the evaluation framework presented in [3]. This framework is organized as a number of evaluation criteria, each describing a characteristic or requirement that a Process Modeling Language (PML) needs to address in order to be considered suitable for use in a Method Engineering (ME) context. These criteria are presented in the next sections and have been divided into two groups: *method modeling criteria* and *process support criteria*. For each criterion, first a short description is given. Then, the criterion is applied to SPEM 2.0 and BPMN 2.0.

## Method Modeling Criteria

In order to be suitable for use in a ME context, a PML must provide adequate mechanisms for modeling software development methods. The criteria of the evaluation framework that aim to evaluate the fulfillment of this requirement have been gathered in this subsection and are presented below.

**EC1. Coverage of ME Process.** This criterion refers to the extent to which the PML covers the ME lifecycle, which is composed of the following phases (see [3]): Method requirements analysis (MRA), method design (MD), method implementation (MI) and method test (MT).

- SPEM 2.0 partially covers MRA since it supports capturing functional requirements such as artifacts or activities, but does not support capturing non-functional requirements. MD is fully covered. MI is supported by those languages that can be enacted either by process engines or project planning tools, so MI is also covered. MT is not supported by SPEM 2.0.
- BPMN 2.0, like SPEM 2.0, covers MD and MI, and partially covers MRA since it does not support capturing non-functional requirements.

**EC2. Modeling Support.** This criterion refers to the ability of a PML to express the elements of a method in a clear and natural way. In order to establish which elements must be included in a method definition, some authors use the term *method dimensions*. In general, it is commonly agreed that a method must cover at least two dimensions: *product* and *process* [2]. However, other dimensions need to be considered. For instance, Seligmann *et al.* [7] consider a method as being composed of three dimensions: *product*, *process* and *tool*. Nonetheless, Henderson-Sellers *et al.* [1] enhance this definition by highlighting the existence of an additional and usually neglected dimension: the *people* (or *role*) dimension. According to these works, we advocate for a four-dimensional view of methods that covers all these aspects: the products created, the process followed, the roles involved and the tools used during the method enactment.

- SPEM 2.0 provides adequate concepts for modeling method elements. The main SPEM 2.0 concepts for representing the (1) product, (2) process, (3) people and (4)

tool dimensions of methods are respectively (1) *WorkProduct*, (2) *Activity*, *Task* and *Milestone*, (3) *Role*, *Qualification* and *RoleSet*, and (4) *Tool*.

- BPMN 2.0 provides obscure concepts for modeling method elements since it is oriented toward process modeling. The main BPMN 2.0 concepts for representing the (1) product and (2) process dimensions are respectively (1) *DataObject* and (2) *Activity* (and all its subclasses: *Task*, *Subprocess*, etc.). The (3) people dimension is usually modeled using the *Lane* primitive. Another possibility is to associate generic *Resources* to *Activities* via *Performer* elements. The (4) tool dimension can be modeled in different ways depending on the type of task. For *UserTasks* the *Rendering* element can be used as an extensible mechanism for specifying UI renderings. For *ScriptTasks* and *ServiceTasks* the attributes *script* and *operationRef* establish respectively the script and operation that will be executed during the task enactment.

**EC3. Abstraction/Modularization.** This criterion refers to the extent to which the PML supports the organization and modularization of reusable method content.

- SPEM 2.0 proposes powerful mechanisms for designing and managing maintainable, large scale, reusable, and configurable libraries or repositories of method content. These mechanisms enable SPEM 2.0 to be used as a framework for the construction of software development knowledge bases, where reusable method content can be stored in a standardized manner. Note that this feature is aligned with the most common approach in ME, i.e., the assembly-based approach [6], since it facilitates the construction of method bases that store method fragments.
- BPMN 2.0 processes are by definition reusable. They can be invoked by other processes by means of the *CallActivity* primitive. However, BPMN 2.0 does not define mechanisms for defining libraries or repositories of process content.

**EC4. Formalism.** This criterion refers to the way the PML represents method components. The more formal rigor is applied, the more automated support is possible, but the language gets less understandable. The best approach is to use an intuitive graphical representation built on top of a formal textual language.

- SPEM 2.0 defines a visual, semi-formal UML-based language. Formal features are not considered, but seem feasible via using OCL.
- BPMN 2.0 defines a graphical notation and formalizes the execution semantics, bridging the gap between process design and implementation. In addition, BPMN 2.0 defines a mapping of BPMN 2.0 models into WS-BPEL, a formal XML-based language for the definition of executable business processes.

**EC5. Simplicity.** This criterion refers to the clarity, ease of use, and understandability of the PML. It is generally accepted that these features are related to the adoption of a graphical notation, since “pictures” are normally thought to be more related to the cognitive part of the human brain as compared to text [8].

- SPEM 2.0 does not address simplicity, due to its complex structure. This makes the language difficult to learn.
- BPMN 2.0 defines an intuitive graphical notation that has been especially designed for use by the people who design and manage business processes. This makes the language easy to learn.

## Process Support Criteria

In order to be suitable in a ME context, a PML must not overlook the process aspects of methods. The set of evaluation criteria that deal with process support of a PML have been gathered in this subsection and are presented below.

**EC6. Process Enactment.** This criterion refers to whether the syntax of the PML has an underlying executable semantics, allowing the processes to be executed.

- SPEM 2.0 does not provide concepts for executing process models, but proposes two alternative ways to do so: (1) mapping processes into project plans and enacting these processes with project planning tools such as Microsoft Project; or (2) mapping processes into executable languages such as BPMN 2.0, and execute these representations of the processes via a process engine.
- BPMN 2.0 fully formalizes execution semantics. Thereby, BPMN 2.0 process models can be executed by means of process engines that provide all enactment facilities, such as activity orchestration, transaction management, event and exception handling, etc.

**EC7. Process Elicitation.** This criterion refers to the extent to which the PML supports the representation of complete, understandable, unambiguous and well-structured processes.

- SPEM 2.0 provides limited support to represent processes, since it only allows the method engineer to establish precedence relationships between tasks, providing poor support to workflow patterns<sup>1</sup>.
- BPMN 2.0 defines powerful mechanisms to represent complete, understandable, unambiguous and well-structured processes. Thus, it provides adequate support to workflow patterns.

**EC8. Process Evolution.** This criterion refers to whether the PML provides mechanisms for facilitating the resumption of process model executions after the modification of the process model, without altering previous states of artifacts, process activities, etc.

- SPEM 2.0 does not address process evolution.

---

<sup>1</sup> <http://www.workflowpatterns.com/>

- BPMN 2.0 does not address process evolution.

**EC9. Process Evaluation.** This criterion refers to whether the PML provides some support for evaluation of the process models. For instance, process evaluation can be conducted by collecting process enactment data during execution and then comparing the data with predefined process objectives.

- SPEM 2.0 does not address process evaluation, since no support is provided for the collection of enactment data.
- BPMN 2.0 does not directly address process evaluation. However, since it is an executable language, implementations can perform evaluation based on the enactment data retrieved by the process engine.

## General Discussion

This technical report shows through a comparative analysis that both SPEM 2.0 and BPMN 2.0 have advantages and limitations. This is illustrated in Table 1, where the results of our study have been summarized. A “+” symbol indicates that a PML fulfils a specific evaluation criterion. A “-“ symbol indicates that the criterion is not fulfilled, and “+/-“ that it is partially fulfilled.

	Method modeling					Process support			
	EC1	EC2	EC3	EC4	EC5	EC6	EC7	EC8	EC9
<b>SPEM 2.0</b>	+/-	+	+	+/-	-	-	+/-	-	-
<b>BPMN 2.0</b>	+/-	+/-	+/-	+	+	+	+	-	+/-

**Table 1.** Summary of comparative analysis between SPEM 2.0 and BPMN 2.0

In order to take advantage of the benefits provided by both languages, we propose combining them. In particular, we suggest using SPEM 2.0 for method definition, and BPMN 2.0 for completing the method process part. The main reason for this is that SPEM 2.0 provides better support to model method elements (see EC2 and EC3 criteria), and BPMN 2.0 is better with respect to process elicitation and also enables the process to be enacted by a process engine (see EC6 and EC7 criteria).

## References

- [1] Henderson-Sellers, B., Ralyté, J.: Situational method engineering: State-of-the-art review. *J. UCS*. 16(3), 424-478 (2010)
- [2] Mirbel, I., Ralyté, J.: Situational method engineering: combining assembly-based and roadmap-driven approaches. *Requirements Engineering* 11, 58-78 (2006)

- [3] Niknafs, A., Asadi, M.: Towards a process modeling language for method engineering support. In: Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering - Volume 07. pp. 674-681. IEEE Computer Society, Washington, DC, USA (2009)
- [4] OMG: Software & Systems Process Engineering Metamodel (v2.0) (2007)
- [5] OMG: Business Process Model and Notation (v2.0) (2011)
- [6] Ralyté, J., Deneckère, R., Rolland, C.: Towards a generic model for situational method engineering. In: Eder, J., Missikoff, M. (eds.) Advanced Information Systems Engineering, Lecture Notes in Computer Science, vol. 2681, pp. 1029-1029. Springer Berlin / Heidelberg (2003)
- [7] Seligmann, P.S., Wijers, G.M., Sol, H.G.: Analyzing the structure of I.S. methodologies, an alternative approach. In: Proceedings of the First Dutch Conference on Information Systems. Amersfoort, The Netherlands (1989)
- [8] Zamli, K.Z., Mat Isa, N.A.: A survey and analysis of process modeling languages. Malaysian Journal of Computer Science. 17, 68-89 (2004)