Integration of
SAP NetWeaver BPM and
Signavio Process Editor

A White Paper

February 2013
Authors

- Gero Decker, Signavio GmbH (gero.decker@signavio.com)
- Nico Herzberg, HPI at University of Potsdam (nico.herzberg@hpi.uni-potsdam.de)
- Vasudevan Kalyanaraman, SAP AG (vasudevan.kalyanaraman@sap.com)
- Stefan Krumnow, Signavio GmbH (stefan.krumnow@signavio.com)
- Matthias Kunze, HPI at University of Potsdam (matthias.kunze@hpi.uni-potsdam.de)
- Christian Loos, SAP AG (christian.loos@sap.com)
- Harald Schubert, SAP AG (harald.schubert@sap.com)
- Volker Stiehl, SAP AG (volker.stiehl@sap.com)
- Mathias Weske, HPI at University of Potsdam (mathias.weske@hpi.uni-potsdam.de)
Introduction

Modeling of business processes plays a crucial role to bridge the gap between company operations and the implementation using software systems. Recently, the Business Process Model and Notation, BPMN (Object Management Group, 2010), has been established as an international standard by the Object Management Group. Today, BPMN is supported by all major vendors in the business process management area, both related to modeling processes and to implementing them.

Starting from the current version, the BPMN standardizes business process diagrams, both with respect to their syntax and their semantics. In addition, and maybe most prominently, the standard also defines an XML-based serialization format of BPMN business process diagrams. Technologically this might be not too exciting, but from an integration perspective, this is a game changer. For the first time, business experts and IT professionals can use the same notation to express business processes. So far they have used such diverse languages like Event-driven Process Chain (EPC) for modeling business processes and web services standards like Web Services Business Process Execution Language (WS-BPEL) for implementing them. These different languages have led to significant miscommunication between the stakeholders involved, resulting in poorly implemented business processes.

This whitepaper reports on the BPMN2-based interchange of process models between Signavio Process Editor and SAP NetWeaver Business Process Management (SAP NetWeaver BPM). The scope comprises the Descriptive Modeling Conformance Subclass plus Multi Instance Activity, Event Based Gateway, and Catching Intermediate Event constructs from the Analytical Modeling Conformance Subclass of the BPMN standard, while we focus on model elements that are relevant for the execution in SAP NetWeaver BPM.

The Signavio Process Editor is a web based software tool that provides complete support for BPMN; it facilitates communication between different stakeholders to support discussion and agreement about process models. The Process Composer is an Eclipse-based tool to model executable business processes in SAP NetWeaver BPM.

The remainder of this paper is organized as follows. In the next section we provide the reader with preliminaries on the BPMN and the tools under investigation. We then look at integrating the Signavio Process Editor with the Process Composer, so that the business process models developed in Signavio can be imported to the SAP tool. We thoroughly analyze the different language elements of the BPMN and report on the lessons learned during the study.

Preliminaries

Business Process Model and Notation

Business Process Model and Notation (BPMN), maintained by the Object Management Group (OMG), is a standard towards formal modeling of business processes. As stated in the specification document (Object Management Group, 2010), the standard aims to “provide a notation that is readily understandable by all business users”, which includes domain experts to elicit and capture existing or desired business processes in a model, developers to implement the process in software or a process-oriented information system, as well as business analysts to monitor and evaluate process execution in the context of Business Process Intelligence (BPI). Hence, BPMN bridges “the gap between the
business process design and process implementation” (Object Management Group, 2010). Many business software vendors have been involved in the development and establishment of this standard which has since experienced wide adoption and can be seen as the de-facto standard for capturing business processes.

Besides capabilities to capture process models, BPMN also suggests conversation and choreography diagrams. The former captures different participants and their interaction structure, the latter addresses modeling of message based interaction, i.e., choreography of several participants. Both model types abstract and hide the internal control flow structures of a participant. As neither conversation nor choreography diagrams are relevant to process execution, currently, we will disregard them hereafter and solely focus on process models.

Besides its claim to support the complete process model life cycle (process design and analysis, implementation, enactment, and evaluation), the standard focuses on interoperability. That is, an XML-based interchange format is specified along with the modeling language that is intended to provide a means to conduct different steps of the process life cycle in various tools that suite a specific purpose. As such, it supports the idea to capture process models in a process modeling tool by domain experts, configure it in a more technical workbench by developers, and deploy it on a process-engine that enacts an implemented process providing for, among others, resource allocation.

**Conformance Levels**

The BPMN specification requires a set of concepts, syntactical rules and semantics to be preserved by software that claims to conform to BPMN. Therefore, different conformance classes have been defined that focus on different aspects, i.e., process modeling and process execution. The **Process Modeling Conformance** class requires a modeling tool to adhere to the visual appearance, structural prescriptions and process semantics as defined in the specification. Process Execution Conformance requires the import of a process model based on the interchange format mentioned above, and its proper performance of execution semantics, as defined in the according sections of the specification. Further conformance classes are BPEL Process Execution Conformance and Choreography Modeling Conformance.

The Process Modeling Conformance class is divided into three subclasses that are characterized by the purpose the models serve, namely *descriptive*, *analytical*, and *common executable*. These classes restrict both the number of modeling constructs and the number of attributes per construct that shall be supported. In BPMN each modeling construct, e.g., an activity, can be configured by a large and complex set of attributes. These include specific semantics of the constructs, e.g., specify whether an activity is instantiated only once, or multiple times, or technical details provided by developers and required for automatic enactment of a process by a process engine.

These different Process Modeling Conformance subclasses can be used as a basis for tools compliance with the BPMN specification. For example, modeling tools focusing primarily on drawing capabilities may take Descriptive Modeling Conformance subclass as its basis, since it includes a subset of BPMN elements that are relevant for modeling high-level business processes. On the other side, the Analytical Conformance subclass includes those elements needed to specify process models for analysis and provides initial features for their execution.
Scope of this Work

In this work, we focus on the Descriptive Modeling Conformance subclass and extend it with few elements enclosed in the Analytical Modeling Conformance subclass. Furthermore, we focus those constructs that are relevant for the process execution. The supported constructs can be classified by process modeling perspectives, function, data, and organizational information. Beside the exchange of model elements, we also considered layout information of the diagram, i.e., position and dimensions of these elements on the model canvas.

The function perspective addresses control flow aspects of a business process, i.e., activities, gateways, events, and sequence flow. A business process is defined as a set of activities that are conducted in coordination to provide added value. Hence, activities are the central aspects of a business process, referred to as tasks, atomic units of work, in BPMN. Here, tasks, userTasks (tasks that are carried out by humans), and serviceTasks (tasks that are carried out by a web service) are considered. Further expanded and collapsed subProcesses, i.e., tasks that contain (and hide) internal control flow shall be supported. In a similar fashion, callActivities capture subprocesses but offer them globally for reuse. A multiInstance Activity, from the Analytical subclass, creates several instances of one activity construct in the process model at runtime that can be executed in parallel or sequentially. Each activity can be configured to provide multiple instance behavior, as this is a particular attribute of the task construct.

Control flow generally addresses dependencies in the execution of activities. Sequence flow edges determine causal dependencies, i.e., an activity can only be started after its predecessor has been terminated. Advanced control flow routing is realized by gateways. The data-based exclusiveGateway (XOR) allows choosing from several alternatives, whereas the parallel gateway (AND) enables several paths, which may be executed concurrently or in interleaving order.

Events denote state changes that are relevant to the business process. Start events create a new process instance and start its execution; we address the types startEvent (none) that denotes manual process instance creation, messageStartEvent which starts a process upon receipt of a message, and timerStartEvent that starts a process at a certain point in time. End events determine the termination of a branch of a process instance, specific types of an event trigger according actions. As such, the endEvent (none) does not trigger any action, whereas the messageEndEvent triggers a final message to be sent. In case, several branches are active concurrently, the terminatingEndEvent cancels all activities and terminates the process instance.

An advanced control flow construct that allows concurrent waiting for one of several events to occur is the eventBasedGateway. Followed by catching intermediate events through sequence flow edges, it allows to wait for any of these events to be observed, however, the first event that occurs will activate the following branch of execution, all other branches become disabled. In this vein, the Catching message Intermediate Event and the Catching timer Intermediate Event are supported, here. These constructs originate from the analytical modeling conformance subclass and have been added to the scope of our work.

The organizational perspective of process models, i.e., roles and responsibilities within a process are represented by swim lanes in BPMN. A process is always constrained by the boundaries of an organization, which is referred to as participant (pool) in the descriptive modeling conformance
subclass. However, it may incorporate several roles, e.g., departments, which are represented by lanes and laneSets within a pool. Lanes can also be nested to refine organizational structures within an organization.

The manipulation and interchange of data is captured in the data perspective of business processes, of which the descriptive modeling conformance subclass requires the dataObject that represents any information or physical entity processed, and dataAssociations which connect data objects to activities and specify whether the data object is read, written or both. dataStoreReferences can be used to denote that an activity accesses a persistent data store, e.g., a relational database.

Several distinct processes cannot be connected through sequence flow edges, in particular, if they reside in different organizations, or pools. However, these processes can interact by means of messageFlow, i.e., an agreed choreography of messages that are passed back and forth to inform partners of the current state of interaction.

BPMN further allows enriching process models with informal information, e.g., notes, documentation, etc. As such, every construct required in the descriptive modeling conformance subclass must support the documentation attribute. Additionally, several constructs can be grouped logically by means of a group, and textAnnotations can be attached to any construct.

**Signavio Process Editor**

The Signavio Process Editor is a collaborative process modeling platform. It focuses on the early phases of process design, where organizational and business issues are at the center of interest. Therefore, the main target group of the Signavio Process Editor consists of business analysts, lines-of-business managers and process participants. These people are typically reluctant to use complicated, IT-oriented development tools and rather need lightweight support for their discussions around business processes.

From a technical point of view, the Signavio Process Editor is a purely HTML 5-based application that runs in the web browser and therefore does not require any software installation. This is particularly important for business users as it allows to get colleagues involved in the discussions in an ad-hoc way.

The tool fully supports Process Modeling Conformance of the BPMN standard, including all modeling constructs and attributes. It enforces basic syntactical correctness of diagrams while modeling. More advanced modeling errors such as deadlocks and lack of synchronization can be checked as well. Every BPMN diagram can be exported as BPMN 2.0 XML file, strictly following the specification. Custom attributes that a user might have added are exported using BPMN’s extension mechanism.
Signavio imports BPMN 2.0 XML files, too. Engine-specific attributes that go beyond the BPMN standard are also imported and stored so that exported BPMN files contain the same execution-relevant information again.

Beyond process modeling and inviting colleagues to comment on diagrams, the Signavio Process Editor also includes the following functionality:

- The process portal allows publishing process models and attached documentation to the entire organization.
- Various reports are available to produce Word, Excel and PDF files.

Repository structure and graphical model version comparison support handling of large process architectures.

**SAP NetWeaver BPM (Process Composer)**

SAP NetWeaver BPM is a process automation platform aimed at model-driven development of human-centric and integration-centric business processes. In that sense it not only supports modeling of business processes based on BPMN, but more importantly it also provides a runtime environment to execute these processes (Process Server).

Process Composer is the design time environment for modeling business processes and integrates with many other model-driven SAP tools inside SAP NetWeaver Developer Studio such as:

- Rules Composer for describing business rules (including decision tables)
- Service Composer for composed services built atop other services
- Composite Application Framework for application services and business objects
- UI tools such as Web Dynpro Java/ABAP, Visual Composer, and SAPUI5 tools
- ESR tools to connect and import services from the Enterprise Service Repository
- Standard Eclipse tools like for example JEE tools

---

1. To be installed separately from SAP Service Marketplace
Process Composer supports all stages of process modeling from high-level definitions of the process down to the enrichment for actual execution in Process Server. This includes the definition of process context data, data mappings, service calls, user interfaces, authorization rules, etc. A model checker ensures the consistency of the model. Graphical debugging of processes is supported and integrated into the overall Eclipse debugging environment which includes, among other things, introspection of process context data. Finally, life cycle aspects of process models (versioning, transport, and so on) are fully supported through the SAP NetWeaver Development Infrastructure.

Seen in a larger context, a process model created in Signavio can be executed by Process Server (within the bounds of the BPMN subset it supports). The model is exported from Signavio and imported into Process Composer. After a successful import, the process is manually enhanced to arrive at an executable model. Further on, the process is compiled and packed into a deployment archive that can then be deployed to and executed on Process Server.

**Modeling vs. Execution**

There are different levels of process modeling, each with different goals and semantics (compare Figure 3).

Signavio Process Editor mainly focuses on operational process models, with a clear business orientation. The primary goal is to understand the process and clarify details, not to necessarily automate it.

On the other hand, SAP NetWeaver BPM is a full BPM suite. Here, the goal is to model processes which can be executed by a process engine. Therefore, the operational process model has to be mapped or transformed to a technical process model, which also takes into account execution semantics of the process engine. As an example, manual tasks are important when documenting an
operational process model, but are not relevant in the technical model as they are performed outside of the process engine.

The following figure (Figure 3) highlights the different levels of process models (Freund & Rücker, 2012).

---

**Forward Engineering**

**Sample Scenario: Patient examination**

The functionalities of the import and export capabilities of both tools are shown with respect to the sample scenario of a patient examination (Stiehl, 2013). Once the process has started the admission of the patient and the assignment to a doctor is processed. The doctor is doing the initial examination. If further examinations are required the patient is assigned to certain specialists. The pool of the Medical Specialists represents a proxy for all additional examinations needed to be done. Once the examinations have been finished the responsible doctor will create the treatment plan. In cases where the responsible doctor does not get any responses from the additional examinations he will check the status of the examinations on a regular basis. After the treatment plan is created the patient is released by getting all the required documents from the personal of the admission. The patient examination process model (designed in BPMN) is shown in Figure 4.
Considered Elements

As stated in the chapter “Scope of this Work” the evaluation targets on elements of descriptive modeling conformance subclass and a few elements from the analytical modeling conformance subclass that are relevant for the process execution. These elements are evaluated by exporting and importing specific created process models from Signavio to SAP Process Composer and vice versa.

As an example the test model for testing pools is shown in Figure 5. The test models kept simple to reduce the complexity, concentrate on the elements focused on and enable easier recap of certain issues.

Results

In general, most modeling elements of the BPMN could be imported and exported in both directions, meaning from Signavio Process Editor to Process Composer and vice versa. While doing this a couple of points need to be considered. A list of the elements investigated is shown below; the term “partly supported” indicates that the element is supported with restrictions. Not imported means that the
element is supported in general, but gets lost during import. Finally, not supported designates an element which is not considered on the modeling palette of the respective environment.

<table>
<thead>
<tr>
<th>BPMN Element</th>
<th>Support in Signavio</th>
<th>Support in SAP NW Process Composer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagram</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>SequenceFlow</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Task (none)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Service Task</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>User Task</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>StartEvent (none)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>EndEvent (none)</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Pool</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>laneSet</td>
<td>✔</td>
<td>- not supported</td>
</tr>
<tr>
<td>Lane</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>exclusiveGateway</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>parallelGateway</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>messageFlow</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>messageStartEvent</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>messageEndEvent</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>DataObject</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>DataStore</td>
<td>✔</td>
<td>- not supported</td>
</tr>
<tr>
<td>DataStoreReference</td>
<td>- partly supported</td>
<td>- not supported</td>
</tr>
<tr>
<td></td>
<td>(cannot be modeled</td>
<td></td>
</tr>
<tr>
<td></td>
<td>without a data store)</td>
<td></td>
</tr>
<tr>
<td>Data Association</td>
<td>✔</td>
<td>- not supported</td>
</tr>
<tr>
<td>Association</td>
<td>✔</td>
<td>- not imported</td>
</tr>
</tbody>
</table>

2 defect arrowhead, missing properties, names only shown for flows originating from XOR gateway
3 order of lanes is changed and names are discarded during import
<table>
<thead>
<tr>
<th>Feature</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Annotation</td>
<td>✔️</td>
</tr>
<tr>
<td>Group</td>
<td>✔️</td>
</tr>
<tr>
<td>SubProcess</td>
<td>✔️</td>
</tr>
<tr>
<td>Call Activity</td>
<td>✔️</td>
</tr>
<tr>
<td>timerStartEvent</td>
<td>✔️</td>
</tr>
<tr>
<td>terminateEndEvent</td>
<td>✔️</td>
</tr>
<tr>
<td>MultiInstance Activity</td>
<td>✔️</td>
</tr>
<tr>
<td>eventBasedGateway</td>
<td>✔️</td>
</tr>
<tr>
<td>timerIntermediateEvent</td>
<td>✔️</td>
</tr>
<tr>
<td>messageIntermediateEvent</td>
<td>✔️</td>
</tr>
</tbody>
</table>

An investigation during importing BPMN files into SAP NW Process Composer showed that SAP NW Process Composer is changing the element IDs. This will have an impact when trying to re-import the model again in Signavio Process Editor.

Figure 6 shows the resulting diagram in SAP NetWeaver BPM after a few adjustments.

![Figure 6: Imported BPMN File of Patient Examination Process (SAP NetWeaver BPM)](image)

Upon importing a BPMN model in SAP NetWeaver BPM, a list of warnings and errors identified during the import may be presented to the user. For instance, if a field that is required by SAP NetWeaver BPM has no value in the imported document, it will be provided with a default value.

---

4 supports annotation to flow nodes and data objects
during import. If not supported elements are identified in the importing file, those elements will be ignored as well as the source and the target sequence and message flows.

While testing all the elements in scope with specific process models, the following specifics were detected. These need to be considered when exporting models from Signavio to SAP NetWeaver BPM:

- *Call Activities* are not used in SAP NetWeaver BPM.
- *Sequential Multiple Instance Activities* could not be modeled in Process Composer. However, this can easily be circumvented by modeling an explicit loop.
- *Data Store, Data Store References, Association, and Groups* are not imported in Process Composer, because they are not required to execute business processes modeled with BPMN.
- The *timerStartEvent* is not supported in Process Composer
- *textAnnotation* is working for flow nodes and data objects in Process Composer, e.g., activities and events
- Merge of process models is not supported in SAP NetWeaver BPM. Each BPMN import creates a new process model.

**Concluding Remarks**

In this report, we show that the integration of business process modeling and analysis tools on the one hand and BPM tools for execution on the other side can in fact be realized. The basis for importing and exporting process models between the Signavio Process Editor and SAP Netweaver BPM is provided by the standardization of the exchange format. Still, there are subtle differences in the interpretation of process models, most of which have been eliminated by SAP and Signavio during this study.

Roundtrip engineering is often seen as the holy grail of implementing business processes. This means that changes to the business process models will have implications on the implementation models, and vice versa. These implications are not of an automatic nature, where business requirements are, by magic, translated to the implementation level. However, scoping where the implemented process needs to be adapted or to which parts of the business process a change in an implementation will have implications, is of crucial importance to improve the link between business level and the implementation level, ultimately leading to narrowing the business-IT-gap. This study provides the basics to a better understanding of the specific challenges imposed by round-tripping between business aspects and technical implementation aspects. Further information on what needs to be considered to keep the executable process as close as possible to the initial process model provided by the business experts can also be found in (Stiehl, 2013).
Bibliography

