

On the Gap between Business Processes and Business Software

Hermann Kaindl

Contributors: Dominik Ertl, Christian Gutschier, Ralph Hoch, Helmut Horacek, Roman Popp and Michael Rathmair

Acknowledgment: Part of this research has been carried out in the ProREUSE project (No. 834167), funded by the Austrian FFG.

Agenda

- Introduction
- Execution of Business Process Models
- Service Composition and Business Processes
- Formalizing Business Processes and their Properties
- Big Picture of Integration
- An Integrating Software Architecture
- Conclusion

Introduction

- Business Process Model and Notation (BPMN)
 - Graphical language for visually defining business processes
 - BPMN 2.0: metamodel and XML specifications
- BPMN 2.0 intends to address this gap.
- How to really execute such models?
- Semantic service specification based on formal logic
- Formal verification of composed services against the specifications of the single services
- Sufficient for business processes?

Introduction (cont.)

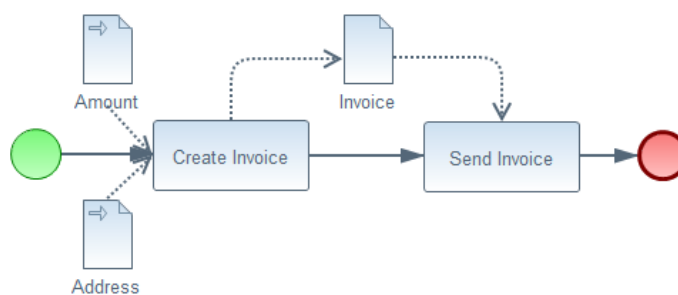
- Verification and validation (V&V)
- Additionally representing and including a certain kind of business rules, including *tacit* knowledge
- Consistently formalizing process models and their properties for model checking
- Closer integration of models, verification and execution
- How can we close the gap?

Execution of Business Process Models – Background

- BPMN 2.0 import mechanism for referencing existing service implementations through a Web Service Description Language (WSDL) file
- BPMN 2.0 *Service Task* specifies how WSDL references can be made and is thus the basis for automatic service execution.
- Different styles and encodings

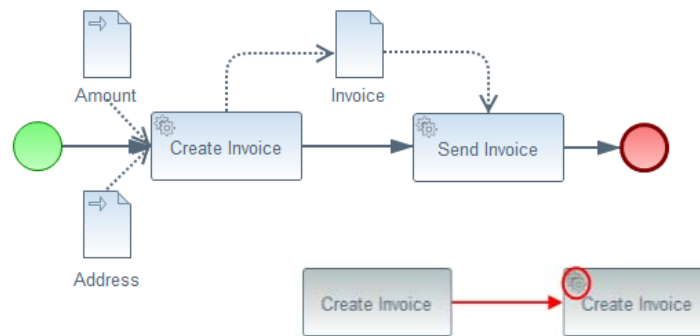
BPMN 2.0 Model Definition and Execution

- Running example (using BPMN Tasks):



BPMN 2.0 Model Definition and Execution – BPMN Service Tasks

- More specific model:

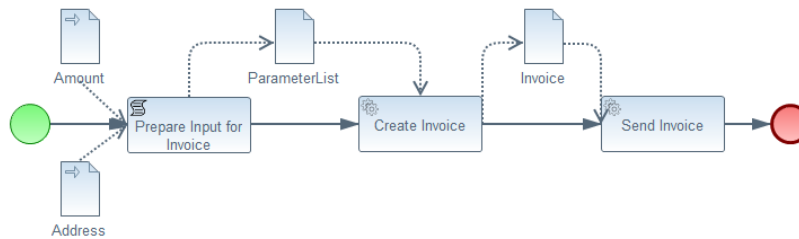


BPMN 2.0 Model Definition and Execution – A Pitfall Found

- Not compliant with the BPMN 2.0 standard!
- "The Service Task inherits the attributes and model associations of Activity (see Table 10.3). In addition the following constraints are introduced when the Service Task references an Operation: The Service Task has exactly one inputSet and at most one outputSet. It has a single Data Input with an ItemDefinition equivalent to the one defined by the Message referenced by the inMessageRef attribute of the associated Operation. If the Operation defines output Messages, the Service Task has a single Data Output that has an ItemDefinition equivalent to the one defined by the Message referenced by the outMessageRef attribute of the associated Operation." [Standard, p. 158]

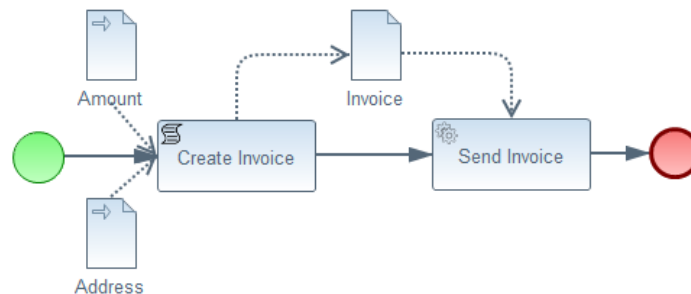
BPMN 2.0 Model Definition and Execution – Additional Wrapper

- Using BPMN *Script Task*



BPMN 2.0 Model Definition and Execution – BPMN Script Task

- Hiding the call to the Web service in a Script Task and thus avoiding the unusual constraint with just one parameter



Specifics of BPMN Execution Frameworks

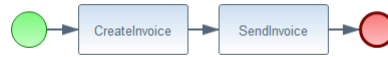
- BonitaSoft BPM
Connector implementation for the integration of external services
- Activiti
Possibility to pass multiple parameters (not standard compliant)
- jBPM
Enforcement of the only-one-parameter constraint of the BPMN 2.0 standard
- Camunda BPM
Only Java classes as reference structures

How to Use FLUX for Formal Verification

- FLUX is a tool implementing the Fluent Calculus.
- Fluent Calculus allows modeling *operations*
 - for transfer from one state to another;
 - through predefined predicates (*poss*, *state_update*).
- Verification of sequences of operations against their specifications
- Verification against goal conditions also possible

How to Use FLUX for Formal Verification — Verifying a Simple Business Process

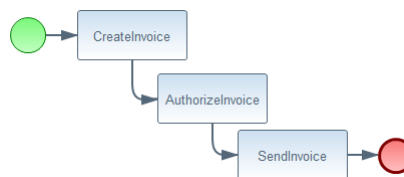
- Business Process: Create and Send Invoice



- Specifying an initial state
- Check whether CreateInvoice can be invoked (poss)
- Invoking it (state_update)
- Check whether SendInvoice can be invoked (poss)
- Invoking it (state_update)
- FLUX tells that the verification succeeds.

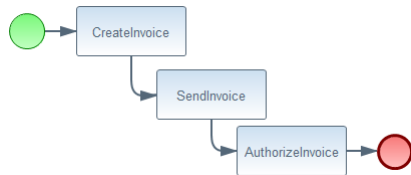
Specifying Semantic Knowledge for Service Composition

- Business process with additional authorization:

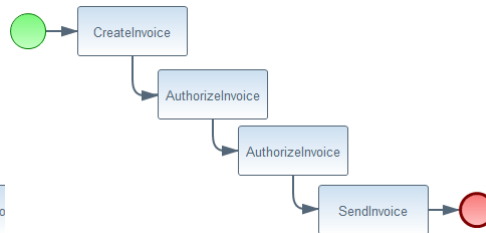
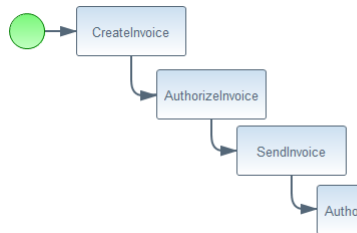


- Specifying this service as an operation for FLUX.
- Formal verification succeeds again.
- This is also a valid business process.

Specifying Semantic Knowledge for Service Composition — Invalid Business Processes



- Formal verification succeeds also for them.

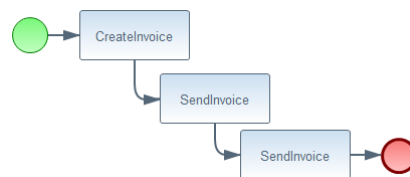


Specifying Semantic Knowledge for Service Composition — Extending Service Specs.

- Additional preconditions for specifications of AuthorizeInvoice and SendInvoice
- “Create and Send Invoice” Process cannot be verified anymore!
- Mismatch of semantic specification and service implementation, more precisely an overspecification
- Additional knowledge encoded not directly related to these services per se

Making Business Rules Explicit

- Business knowledge, business rules
- Modeled in FLUX like services
- Minor changes to service specifications required, e.g., being ready for sending
- Still, invalid processes verify:



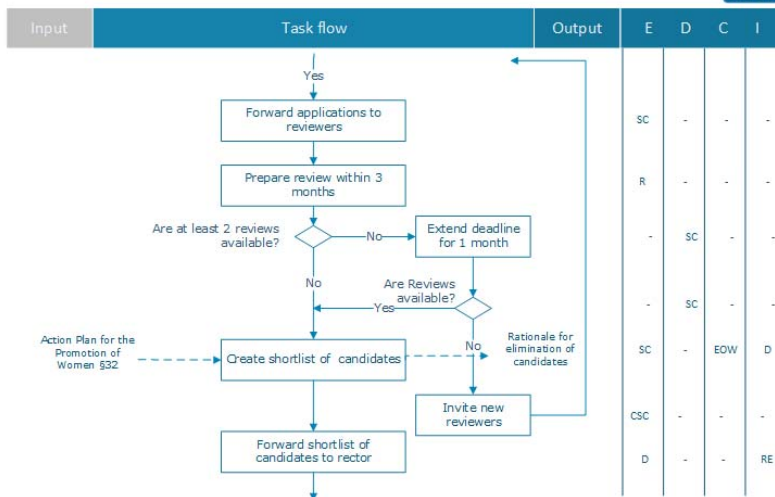
- *Tacit* business rules

Formalizing Business Processes and their Properties

- Formal verification of business process models
- Model checking (also known as property checking)
- Key issue: to consistently formalize properties and business process
- Semi-formal representation of properties
- Step-wise and incremental formalization

Model Checking — Part of Process Model

MAN-03-02-S Process of Appointing a Professor

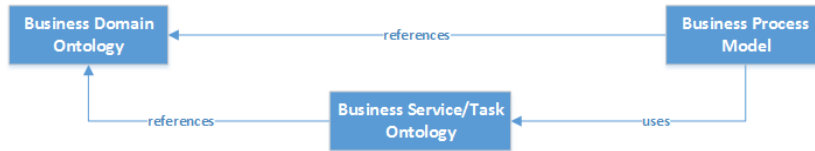


Model Checking — Example Property

- Property derived from the Guideline:

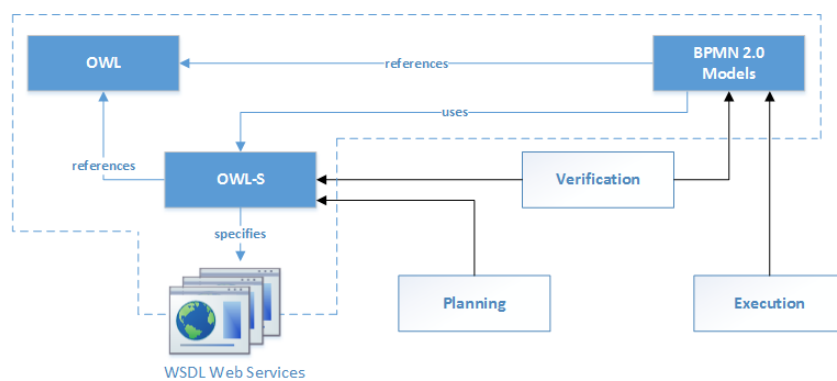
§7 (1)	The chairperson of the search committee forwards this list as soon as possible to the responsible dean, who forwards it to the rector.
	$G (CSC \text{ of } SC = TRUE \rightarrow F SC.state = \text{forward candidate list to dean})$
	No refined formalization possible, since no such state exists in the FSMs.

Big Picture of Integration – Conceptual View



- Business Domain Ontology
 - Concepts/objects of the business domain
- Business Service/Task Ontology
 - Semantic specification of services and tasks
- Business Process Model
 - Specification of behavior through sub-processes/tasks
 - Specification of service orchestration (and operation on objects)

Big Picture of Integration – Concrete Technologies



Potential Use of the Integration

- Domain-driven development of business software
- Top-down development
- Automated generation of (certain kinds of) business process models
- Formal verification of (certain kinds of) business process models
- Automated generation of (parts of) user interfaces for business software

An Integrating Software Architecture



- BPMN Execution Engine integrated in the second tier of the architecture

Conclusion

- Tacit business knowledge needs to be made explicit and even formally represented for automated V&V of service composition and business processes.
- Certain business rules need to be specified additionally and separately.
- Consistently formalizing is key for successful formal and automated verification of business processes.
- Closer and more comprehensive integration will be needed.
- The gap is still wide ...



Thank you for your attention!

???

