

Reasoning on semantically annotated processes

Chiara Di Francescomarino Chiara Ghidini
Marco Rospocher Luciano Serafini Paolo Tonella



FBK-irst, Trento, Italy

6th International Conference on Service Oriented Computing
(ICSOC 2008)

December 1-5, 2008

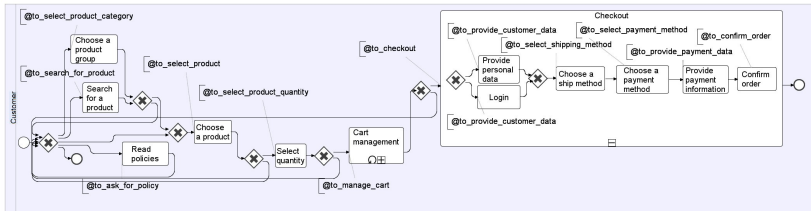
University of Technology, Sydney, NSW, Australia

Annotating business processes with concepts from domain ontologies allows to improve the automation of business process management.

Introduction & Motivation

Annotating business processes with concepts from domain ontologies allows to improve the automation of business process management.

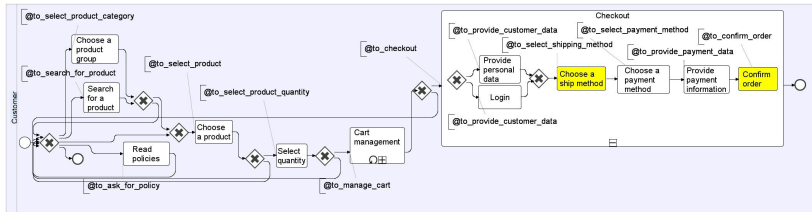
- Verifying the presence of critical patterns.



Introduction & Motivation

Annotating business processes with concepts from domain ontologies allows to improve the automation of business process management.

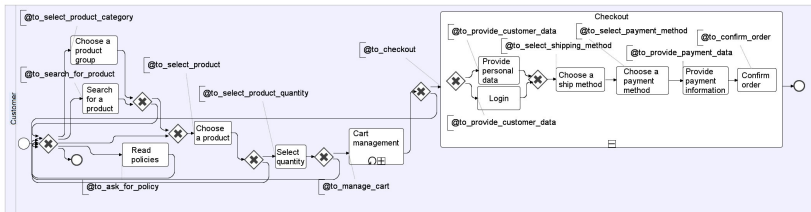
- Verifying the presence of critical patterns.



Introduction & Motivation

Annotating business processes with concepts from domain ontologies allows to improve the automation of business process management.

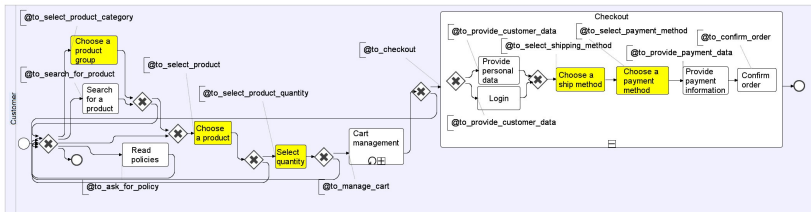
- Verifying the presence of critical patterns.
- Searching for crosscutting concerns.



Introduction & Motivation

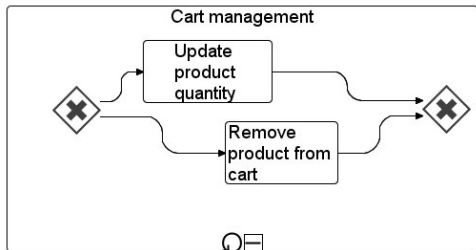
Annotating business processes with concepts from domain ontologies allows to improve the automation of business process management.

- Verifying the presence of critical patterns.
- Searching for crosscutting concerns.



However, semantic annotations will positively affect the creation of high quality process models only if they are correct.

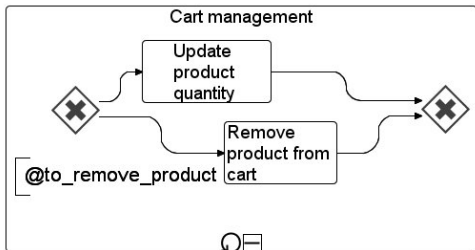
However, semantic annotations will positively affect the creation of high quality process models only if they are correct.



- ▼ ● **action**
 - ▼ ● **to_add**
 - **to_add_product**
 - ▶ ● **to_ask_for**
 - **to_checkout**
 - ▶ ● **to_compute**
 - ▼ ● **to_remove**
 - **to_remove_product**
 - ▶ ● **to_search_for**
- ▼ ● **object**
 - **customer**
 - ▼ ● **data**
 - **cart_data**
 - **checkout**

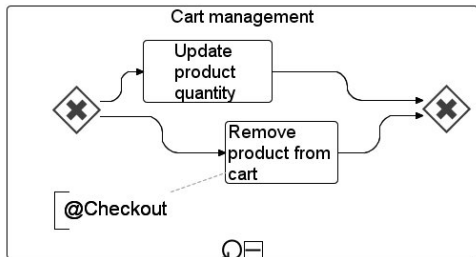
Research Problem

However, semantic annotations will positively affect the creation of high quality process models only if they are correct.



- ▼ ● action
 - ▼ ● to_add
 - to_add_product
 - ▶ ● to_ask_for
 - ▶ ● to_checkout
 - ▶ ● to_compute
 - ▼ ● to_remove
 - to_remove_product
 - ▶ ● to_search_for
- ▼ ● object
 - customer
 - ▼ ● data
 - cart_data
 - checkout

However, semantic annotations will positively affect the creation of high quality process models only if they are correct.



- ▼ ● action
 - ▼ ● to_add
 - to_add_product
 - ▶ ● to_ask_for
 - ▶ ● to_checkout
 - ▶ ● to_compute
 - ▼ ● to_remove
 - to_remove_product
 - ▶ ● to_search_for
- ▼ ● object
 - customer
 - ▼ ● data
 - cart_data
 - **checkout**

We propose an approach for

- the specification of constraints for correct annotations of business processes;
- the automatic verification of the correctness of annotated processes;
- the provision of reasoning services on annotated processes.

Still to come...

Annotation of BPMN Business Process Diagrams

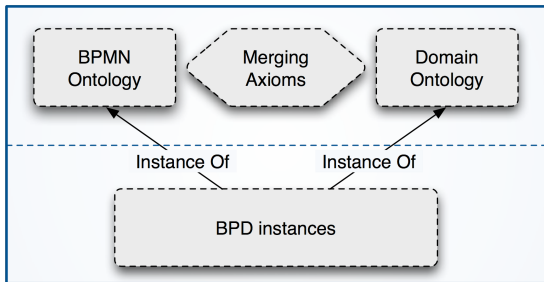
- Business Process Modelling Notation (BPMN) is a language for the specification of Business Process Diagrams (BPDs).
- BPMN is extended by allowing annotation of the objects of business processes with concepts taken from a domain ontology.
- The goal is to have correctly annotated business processes.
- Criteria for correct/incorrect annotation are statements that bridge the semantics of BPMN and the semantics of the domain ontology.
- Formally, we represent these criteria by inclusion axioms between the concepts of an ontology formalizing BPMN and the domain ontology.

The Business Processes Knowledge Base

We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.

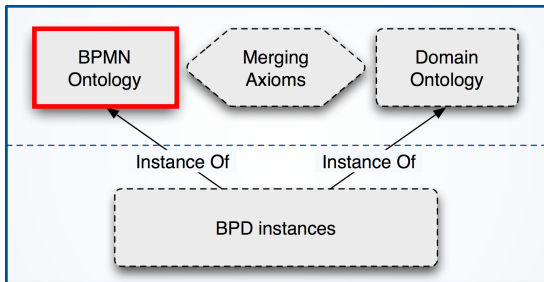
The Business Processes Knowledge Base

We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.



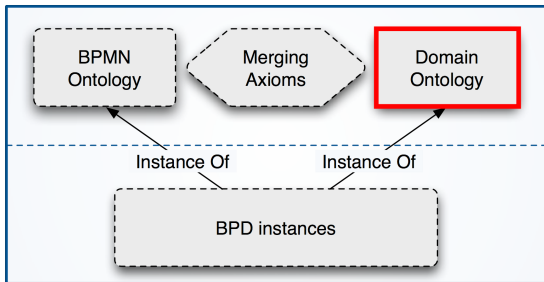
The Business Processes Knowledge Base

We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.



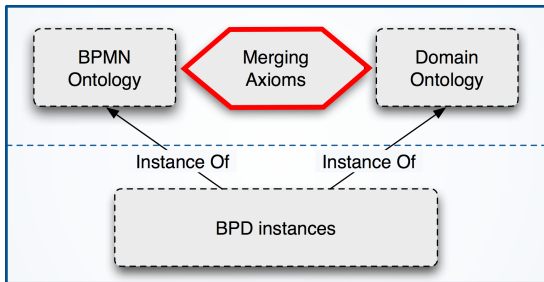
The Business Processes Knowledge Base

We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.



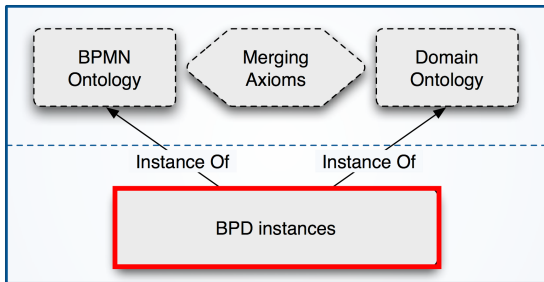
The Business Processes Knowledge Base

We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.



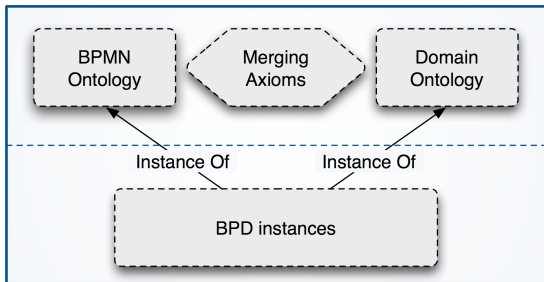
The Business Processes Knowledge Base

We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.



The Business Processes Knowledge Base

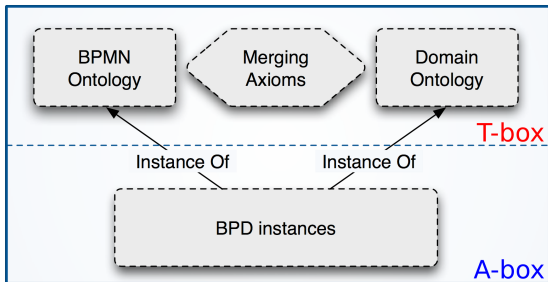
We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.



We implemented BPKB using OWL (Web Ontology Language), based on Description Logics (DL).

The Business Processes Knowledge Base

We propose to encode all the information about semantically annotated processes into a logical knowledge base, called **Business Processes Knowledge Base (BPKB)**.



We implemented BPKB using OWL (Web Ontology Language), based on Description Logics (DL).

An ontology for BPMN

The BPMN Ontology provides a formalization of the structural part of BPDs:

- it contains all the BPMN elements, organized in an *is-a* taxonomy;

An ontology for BPMN

The BPMN Ontology provides a formalization of the structural part of BPDs:

- it contains all the BPMN elements, organized in an *is-a* taxonomy;



An ontology for BPMN

The BPMN Ontology provides a formalization of the structural part of BPDs:

- it contains all the BPMN elements, organized in an *is-a* taxonomy;



- it contains attributes and properties which describe how to use these elements to compose business process diagrams.

An ontology for BPMN

The BPMN Ontology provides a formalization of the structural part of BPDs:

- it contains all the BPMN elements, organized in an *is-a* taxonomy;



- it contains attributes and properties which describe how to use these elements to compose business process diagrams.

A Start Event MUST NOT be a target for Sequence Flow

An ontology for BPMN

The BPMN Ontology provides a formalization of the structural part of BPDs:

- it contains all the BPMN elements, organized in an *is-a* taxonomy;



- it contains attributes and properties which describe how to use these elements to compose business process diagrams.

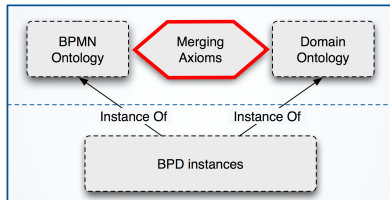
A Start Event MUST NOT be a target for Sequence Flow

$\text{sequence_flow} \sqsubseteq \forall \text{has_connecting_obj_target_ref.} (\neg \text{start_event})$

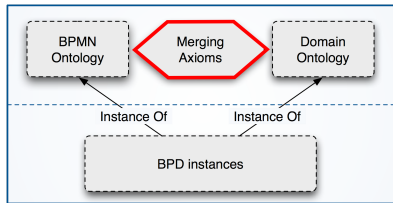
The BPMN Ontology provides a formalization of the structural part of BPDs:

- It is not intended to model the dynamic behaviour of BPDs.
If there are multiple outgoing Sequence Flow then only one Gate (or the DefaultGate) SHALL be selected during performance of the Process.
- There are a few documented properties which are not represented due to expressiveness limitation imposed by Description Logics.
- It is based on BPMN v1.1 specifications from OMG.
- Available for download at http://dkm.fbk.eu/index.php/BPMN_Ontology.

Merging Axioms



Merging Axioms



- To allow the business designer to specify constraints for the correct/incorrect annotation of BPD objects, we introduce four relations mapping pairs of concepts each belonging to one of the two ontologies.
- Their informal description provided via these relation is then translated into a formal set of DL axioms (class inclusion axioms).

Merging Axioms: annotatable only by

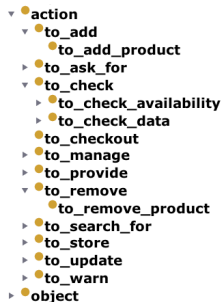
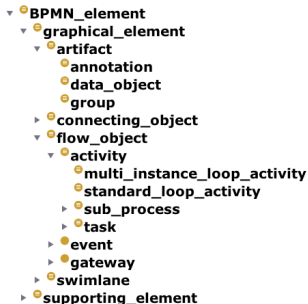
From BPMN ontology to BD ontology.

- **annotatable only by**: a BPMN element of type x can be annotated only with a concept of the BD ontology equivalent or more specific than y .

Merging Axioms: annotatable only by

From BPMN ontology to BD ontology.

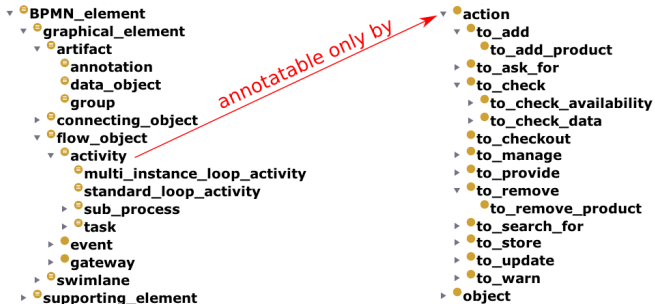
- **annotatable only by**: a BPMN element of type x can be annotated only with a concept of the BD ontology equivalent or more specific than y .



Merging Axioms: annotatable only by

From BPMN ontology to BD ontology.

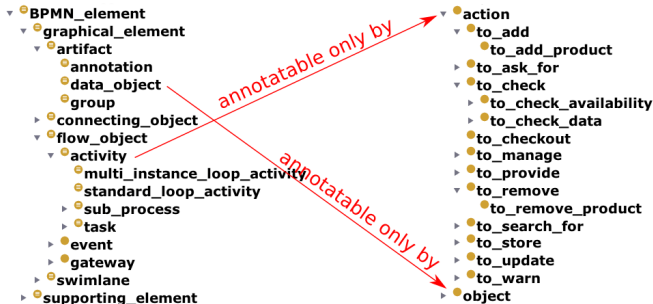
- **annotatable only by**: a BPMN element of type x can be annotated only with a concept of the BD ontology equivalent or more specific than y .



Merging Axioms: annotatable only by

From BPMN ontology to BD ontology.

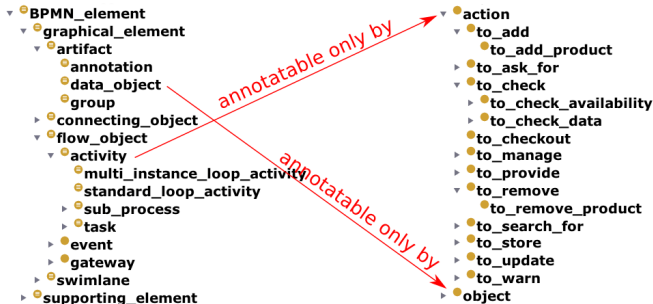
- **annotatable only by**: a BPMN element of type x can be annotated only with a concept of the BD ontology equivalent or more specific than y .



Merging Axioms: annotatable only by

From BPMN ontology to BD ontology.

- **annotatable only by**: a BPMN element of type x can be annotated only with a concept of the BD ontology equivalent or more specific than y .

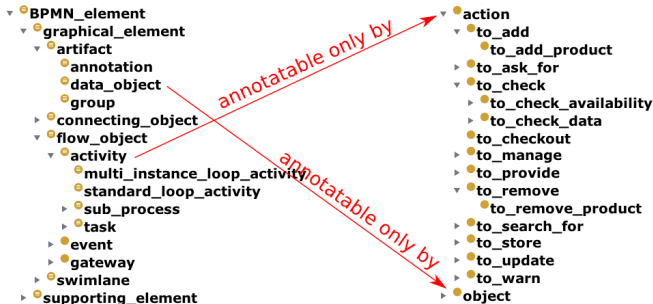


Formal axiom: $x \sqsubseteq y$

Merging Axioms: annotatable only by

From BPMN ontology to BD ontology.

- **annotatable only by**: a BPMN element of type x can be annotated only with a concept of the BD ontology equivalent or more specific than y .



Formal axiom: $x \sqsubseteq y$ activity \sqsubseteq action, data_object \sqsubseteq object

Merging Axioms: not annotatable by

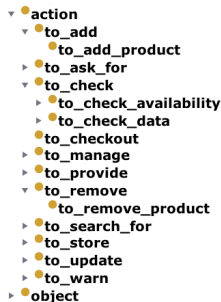
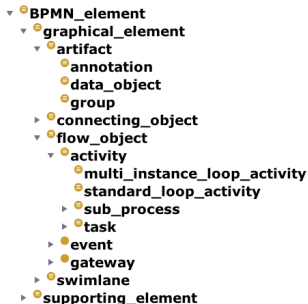
From BPMN ontology to BD ontology.

- **not annotatable by**: a BPMN element of type x cannot be annotated with a concept of the BD ontology equivalent or more specific than y .

Merging Axioms: not annotatable by

From BPMN ontology to BD ontology.

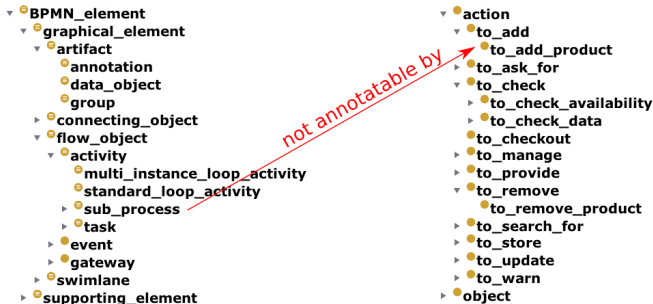
- **not annotatable by**: a BPMN element of type x cannot be annotated with a concept of the BD ontology equivalent or more specific than y .



Merging Axioms: not annotatable by

From BPMN ontology to BD ontology.

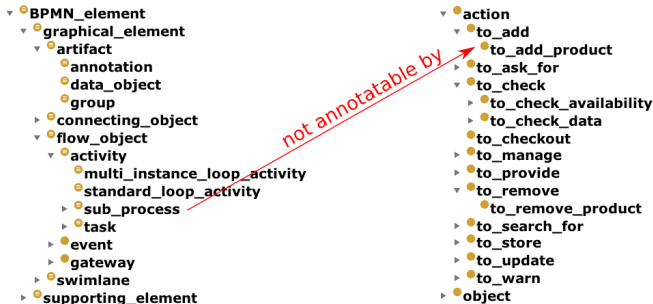
- **not annotatable by**: a BPMN element of type x cannot be annotated with a concept of the BD ontology equivalent or more specific than y .



Merging Axioms: not annotatable by

From BPMN ontology to BD ontology.

- **not annotatable by**: a BPMN element of type x cannot be annotated with a concept of the BD ontology equivalent or more specific than y .

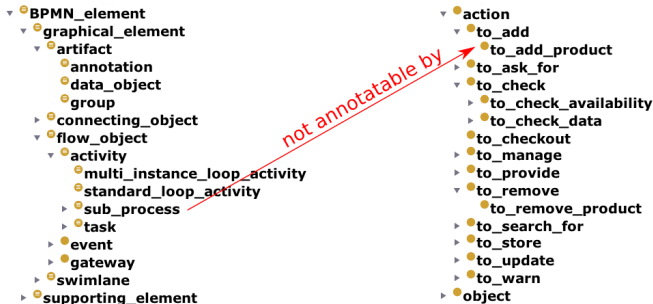


Formal axiom: $x \sqsubseteq \neg y$

Merging Axioms: not annotatable by

From BPMN ontology to BD ontology.

- **not annotatable by**: a BPMN element of type x cannot be annotated with a concept of the BD ontology equivalent or more specific than y .



Formal axiom: $x \sqsubseteq \neg y$

$sub_process \sqsubseteq \neg to_add_product$

Merging Axioms: *annotates only*

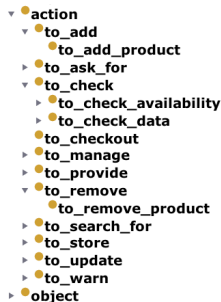
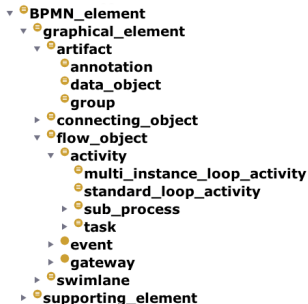
From BD ontology to BPMN ontology.

- *annotates only*: any concept of the BD ontology equivalent or more specific than y can be used to denote only BPMN elements of type x .

Merging Axioms: annotates only

From BD ontology to BPMN ontology.

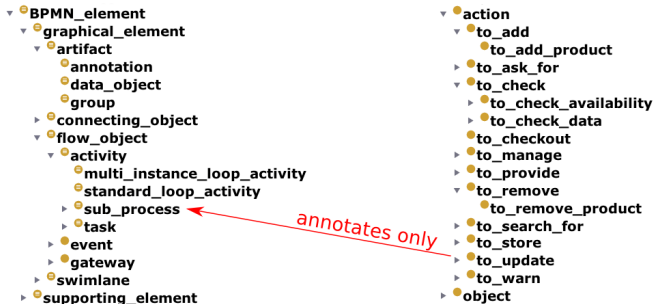
- **annotates only**: any concept of the BD ontology equivalent or more specific than y can be used to denote only BPMN elements of type x .



Merging Axioms: annotates only

From BD ontology to BPMN ontology.

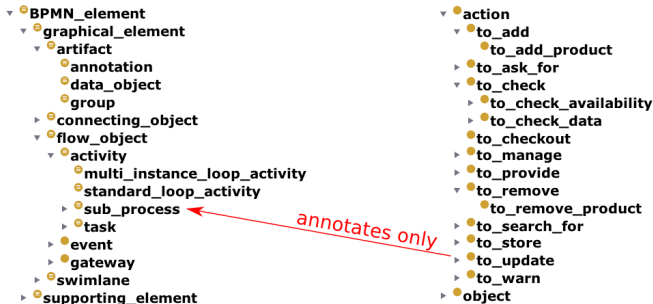
- **annotates only**: any concept of the BD ontology equivalent or more specific than y can be used to denote only BPMN elements of type x .



Merging Axioms: annotates only

From BD ontology to BPMN ontology.

- **annotates only**: any concept of the BD ontology equivalent or more specific than y can be used to denote only BPMN elements of type x .

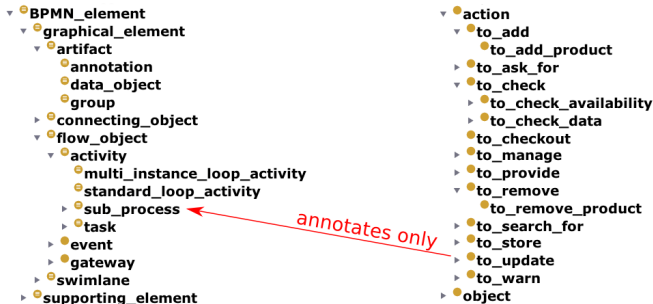


Formal axiom: $y \sqsubseteq x$

Merging Axioms: annotates only

From BD ontology to BPMN ontology.

- **annotates only**: any concept of the BD ontology equivalent or more specific than y can be used to denote only BPMN elements of type x .



Formal axiom: $y \sqsubseteq x$ $\text{to_update} \sqsubseteq \text{sub_process}$

Merging Axioms: cannot annotate

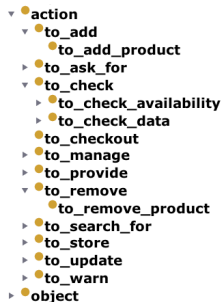
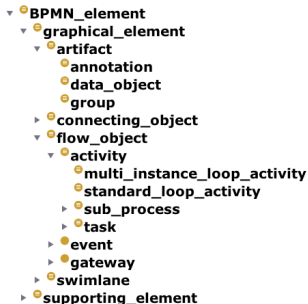
From BD ontology to BPMN ontology.

- **cannot annotate**: any concept of the BD ontology equivalent or more specific than y cannot be used to denote BPMN elements of type x .

Merging Axioms: cannot annotate

From BD ontology to BPMN ontology.

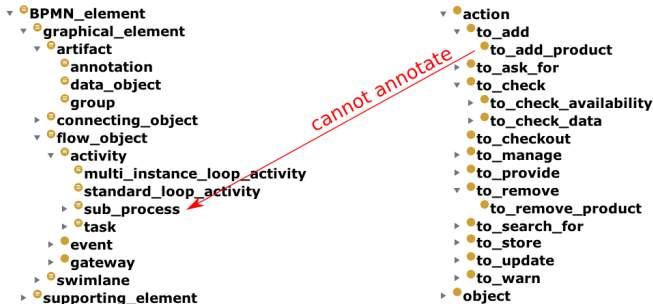
- **cannot annotate**: any concept of the BD ontology equivalent or more specific than y cannot be used to denote BPMN elements of type x .



Merging Axioms: cannot annotate

From BD ontology to BPMN ontology.

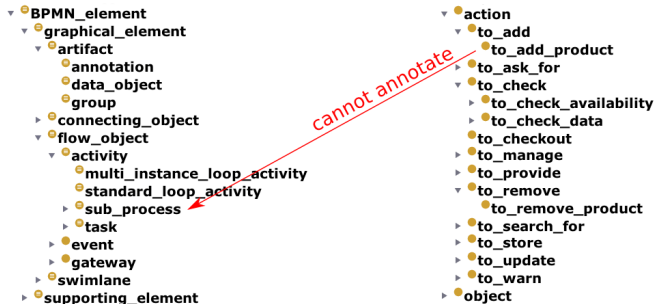
- **cannot annotate**: any concept of the BD ontology equivalent or more specific than y cannot be used to denote BPMN elements of type x .



Merging Axioms: cannot annotate

From BD ontology to BPMN ontology.

- **cannot annotate**: any concept of the BD ontology equivalent or more specific than y cannot be used to denote BPMN elements of type x .

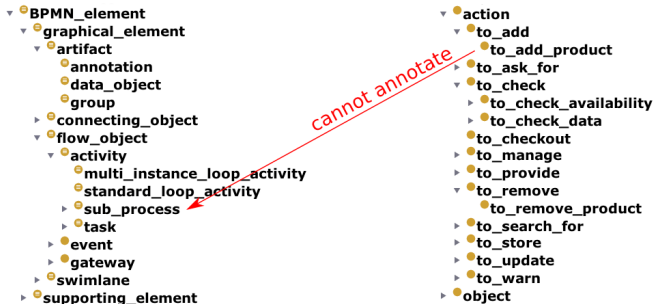


Formal axiom: $y \sqsubseteq \neg x$

Merging Axioms: cannot annotate

From BD ontology to BPMN ontology.

- **cannot annotate**: any concept of the BD ontology equivalent or more specific than y cannot be used to denote BPMN elements of type x .

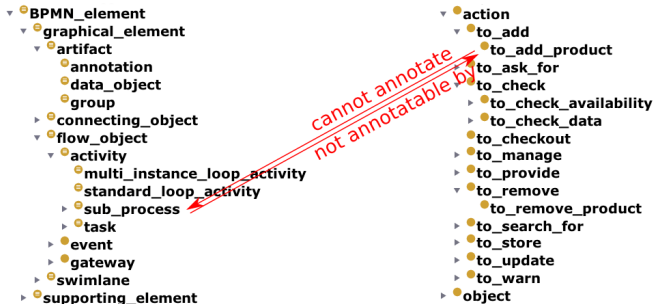


Formal axiom: $y \sqsubseteq \neg x$ `to_add_product` \sqsubseteq \neg `sub_process`

Merging Axioms: cannot annotate

From BD ontology to BPMN ontology.

- **cannot annotate**: any concept of the BD ontology equivalent or more specific than y cannot be used to denote BPMN elements of type x .



Formal axiom: $y \sqsubseteq \neg x$

$\text{to_add_product} \sqsubseteq \neg \text{sub_process}$

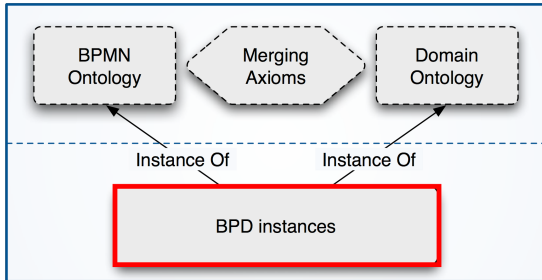
- We can automatically detect bindings specified by the modeller (via the four primitives) that may generate inconsistencies by verifying the consistency of integrated ontology

$$\text{BPMNO} \cup \text{BDO} \cup \text{Merging_Axioms}(\text{BPMNO}, \text{BDO})$$

via a DL reasoner.

- Suggestions for recovering from inconsistency can be given automatically.

Representing a semantically annotated BPD in the BPKB

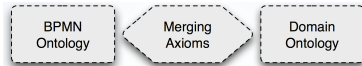
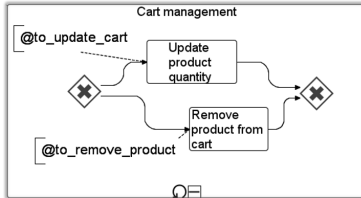


Representing a semantically annotated BPD in the BPKB

Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.

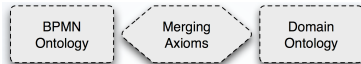
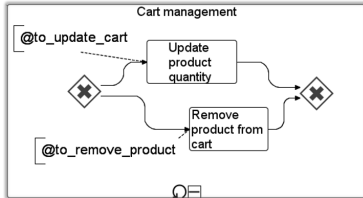
Representing a semantically annotated BPD in the BPKB

Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.



Representing a semantically annotated BPD in the BPKB

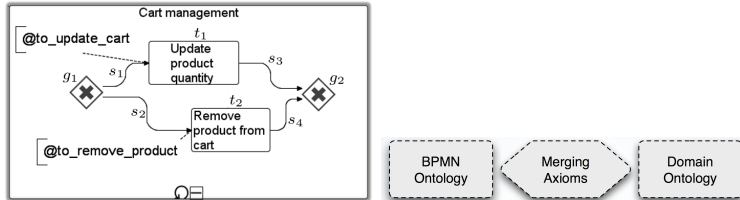
Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.



We create an individual for each graphical element of the business process.

Representing a semantically annotated BPD in the BPKB

Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.

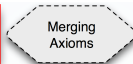
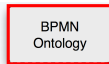
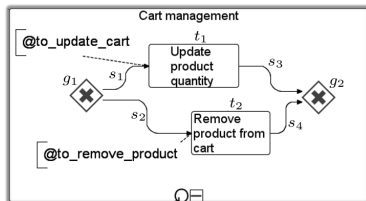


We create an individual for each graphical element of the business process.

$s_1, s_2, s_3, s_4, t_1, t_2, g_1, g_2$

Representing a semantically annotated BPD in the BPKB

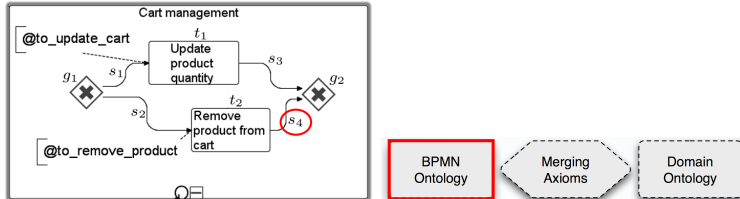
Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.



BPM-type assertions: for every graphical element g of BPMN type T occurring in the process, we add the assertions $T(g)$.

Representing a semantically annotated BPD in the BPKB

Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.

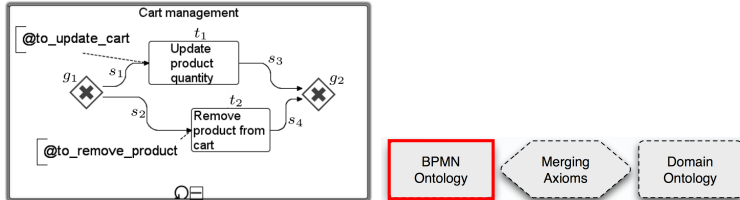


BPM-type assertions: for every graphical element g of BPMN type T occurring in the process, we add the assertions $T(g)$.

$sequence_flow(s_4)$

Representing a semantically annotated BPD in the BPKB

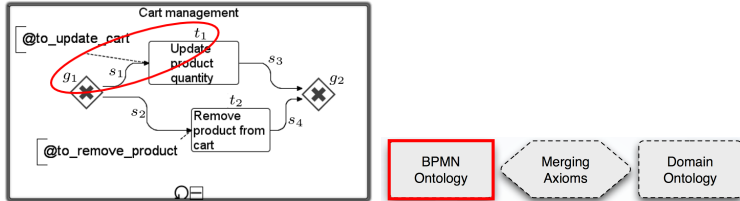
Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.



BPM-structural assertions For every connecting object c , that goes from a to b , we add two structural assertions of the form $\text{SourceRef}(c, a)$ and $\text{TargetRef}(c, b)$.

Representing a semantically annotated BPD in the BPKB

Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.

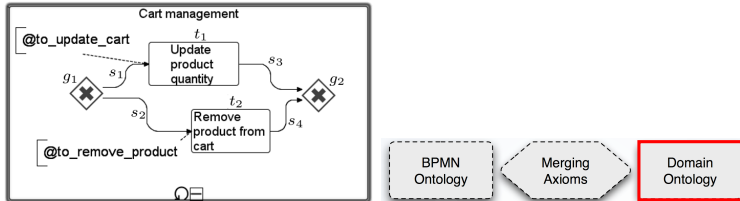


BPM-structural assertions For every connecting object c , that goes from a to b , we add two structural assertions of the form $\text{SourceRef}(c, a)$ and $\text{TargetRef}(c, b)$.

$\text{has_sequence_flow_source_ref}(s_1, g_1)$
 $\text{has_sequence_flow_target_ref}(s_1, t_1)$

Representing a semantically annotated BPD in the BPKB

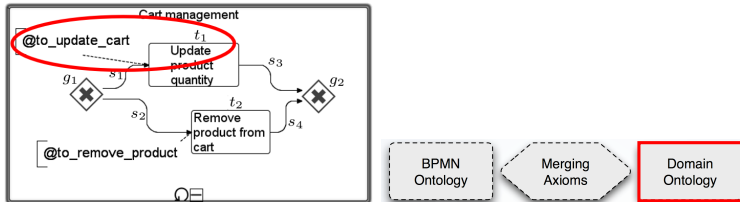
Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.



BPM-semantic assertions For every graphical element g of the diagram which is annotated with C (where C is a complex concept expression of the domain ontology), we add the assertion $C(g)$.

Representing a semantically annotated BPD in the BPKB

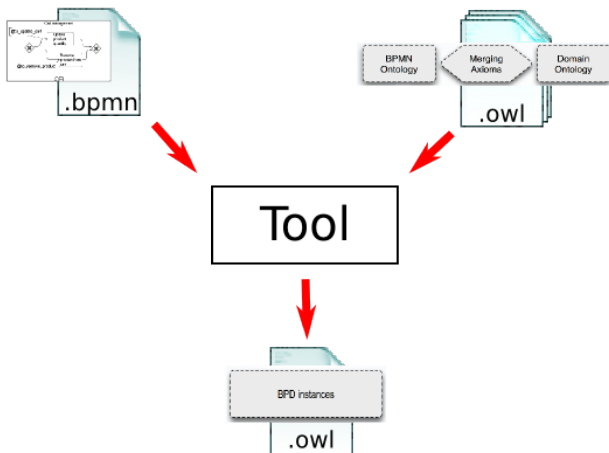
Given a semantically annotated business processes, we formalize it as an A-box in our BPKB.



BPM-semantic assertions For every graphical element g of the diagram which is annotated with C (where C is a complex concept expression of the domain ontology), we add the assertion $C(g)$.

`to_update_cart(t_1)`

Automatically Encoding a BPD into an A-Box



Representing a semantically annotated BPD in the BPKB

Given an OWL representation of a BPKB we can reduce the problem of checking the correctness of the semantic annotation of a BPD to a satisfiability problem in DL. In particular we can reformulate it as the fact that

$\text{BPD Instances} \cup \text{BPMNO} \cup \text{BDO} \cup \text{Merging_Axioms}(\text{BPMNO}, \text{BDO})$

is a consistent knowledge base.

Reasoning services over semantically annotated business processes

Using logical reasoning over BPKB we can implement the following services:

Reasoning services over semantically annotated business processes

Using logical reasoning over BPKB we can implement the following services:

- Reasoning on the BPD instances. Queries may involve either the domain ontology, the BPMN ontology or both.

Reasoning services over semantically annotated business processes

Using logical reasoning over BPKB we can implement the following services:

- Reasoning on the BPD instances. Queries may involve either the domain ontology, the BPMN ontology or both.
- Verification of semantic annotation. Verifying whether the semantic annotation satisfies the constraints specified using the merging axioms.

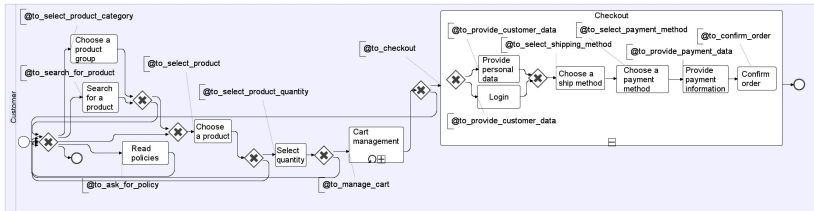
Reasoning services over semantically annotated business processes

Using logical reasoning over BPKB we can implement the following services:

- Reasoning on the BPD instances. Queries may involve either the domain ontology, the BPMN ontology or both.
- Verification of semantic annotation. Verifying whether the semantic annotation satisfies the constraints specified using the merging axioms.
- Suggestions for correct annotation of the process. Merging axioms can be used to suggest (sets of) of **correct** annotations on-the-fly during process annotation.

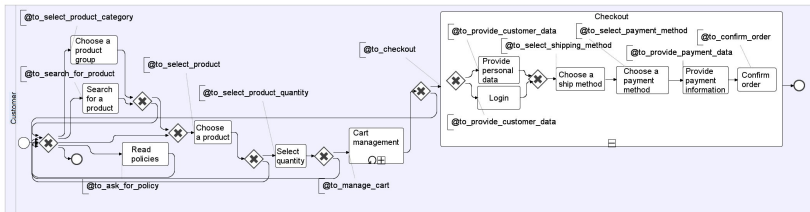
Searching for Crosscutting Concerns

Managing the customer's preferences in an On-line Shop process.



Searching for Crosscutting Concerns

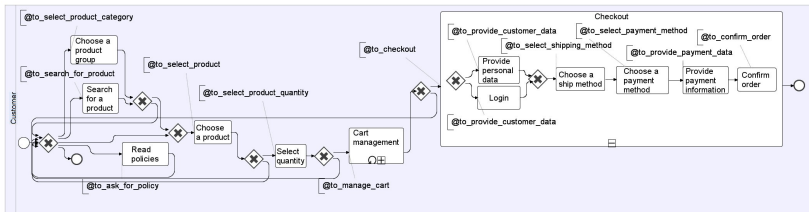
Managing the customer's preferences in an On-line Shop process.



We can retrieve all the different points in the workflow where a customer expresses her preferences by formulating a SPARQL query matching the “customer choice” concern:

Searching for Crosscutting Concerns

Managing the customer's preferences in an On-line Shop process.

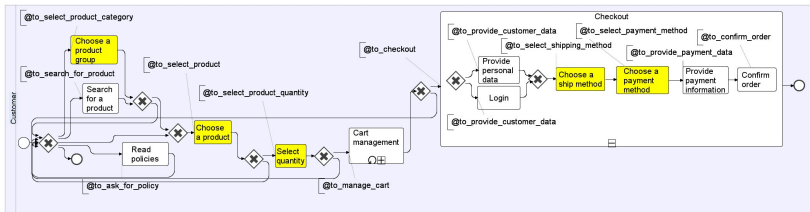


We can retrieve all the different points in the workflow where a customer expresses her preferences by formulating a SPARQL query matching the “customer choice” concern:

retrieve all BPMN elements of type **activity** in the **Customer** pool which are labelled with BDO concept **to_select**.

Searching for Crosscutting Concerns

Managing the customer's preferences in an On-line Shop process.



We can retrieve all the different points in the workflow where a customer expresses her preferences by formulating a SPARQL query matching the “customer choice” concern:

retrieve all BPMN elements of type **activity** in the **Customer** pool which are labelled with BDO concept **to_select**.

Conclusions

- We have proposed a method to add **correct** semantic annotations to a business process, based on a set of merging axioms that connect a BPMN ontology and a business domain ontology.
- Semantic annotations allow formal, automated reasoning on the elements and properties of a business process.
- Structural and domain specific constraints can be expressed as axioms and can be verified as ontology consistency violations.
- Queries on the instances (i.e. actual process elements) can be defined to match relevant process features, such as crosscutting concerns.

Some directions for future works

- Simplify the task of ontology merging for the final user by means of tools and algorithms that handle inconsistencies.
- Investigate user friendly notations for constraint and query specification.
- Moving from specification to executable process description languages, such as BPEL.
- Validate the approach further, on larger case studies.
- Design a collaborative tool (wiki-style) for (guided) annotation of business processes.

Thank you!

Marco Rospoche

Data and Knowledge Management Unit (DKM)¹

FBK-irst

rospoche@fbk.eu

¹<http://dkm.fbk.eu>