Automatic User Support for Business Process Modeling

Stefanie Betz, Stefan Klink, Agnes Koschmider, Andreas Oberweis

Institute AIFB, Universität Karlsruhe (TH), Germany

Workshop “Semantics for Business Process Management”

Outline

1. Motivation
2. Foundations
3. Measuring Similarity between Process Elements
4. Analysis Methods for Petri nets
5. Conclusion and Outlook

Motivation - Application scenario

autocompletion of process models:

- process element names might differ in syntax or one process can be modeled in different ways even when utilizing the same modeling language

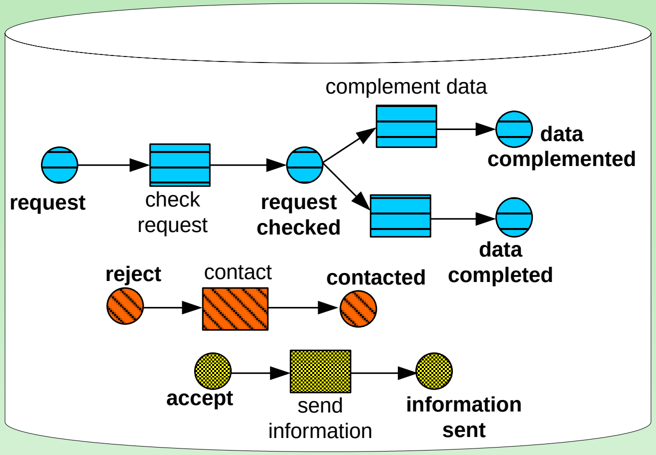
→ *how to solve ambiguity issues?*

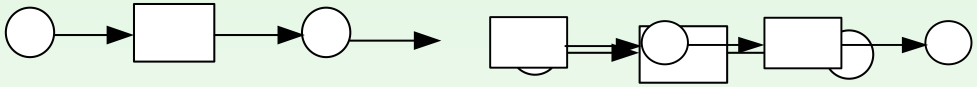
- typos and structural modeling errors make it particularly error prone to model processes manually

→ *which are appropriate subsequent elements? (find and proposition)*

Motivation

Process Templates Database





Automatic User Support for Business Process Modeling

Semantic Business Process Models

- missing semantic description of Petri net components hampers automated processing

- to uncover synonyms or homonyms of process element names a description of Petri nets in an unambiguous format is required (i.e. Web Ontology Language OWL)

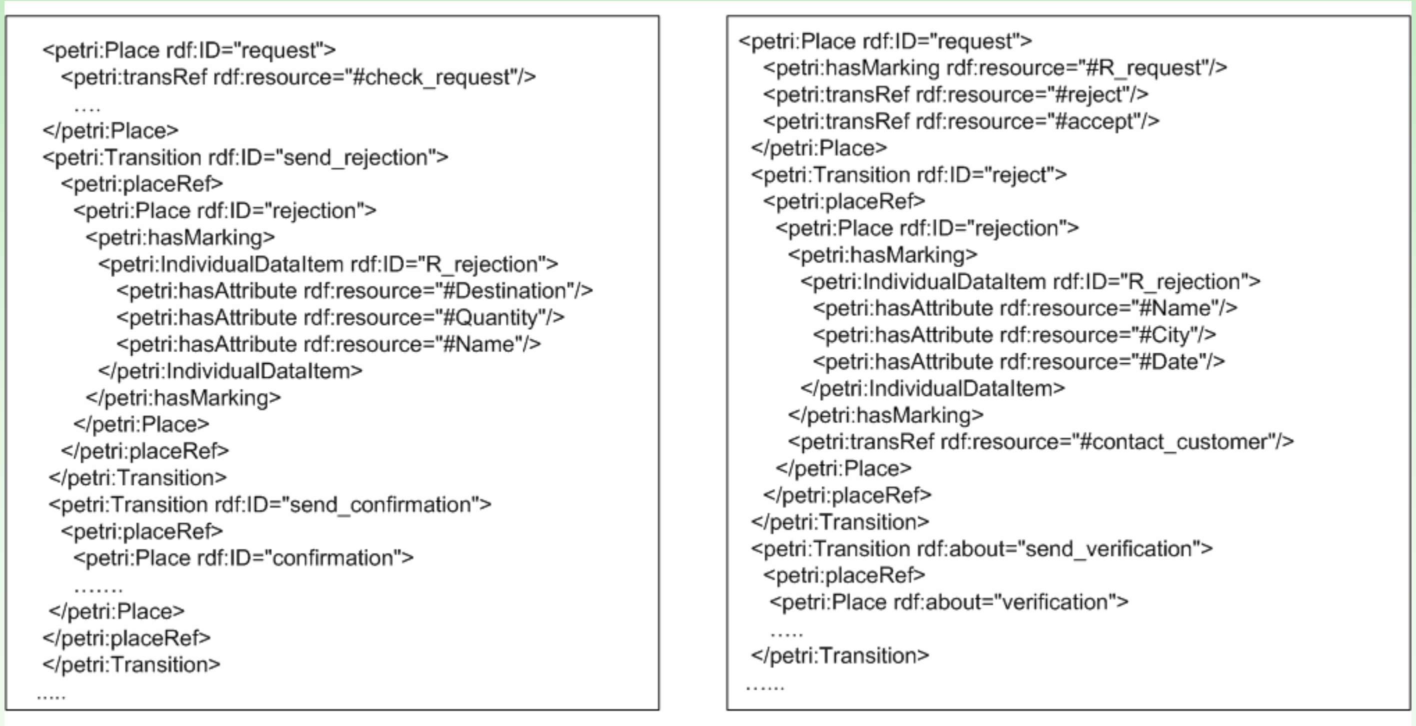
- machine readable and interpretable format, which might be used for ontological reasoning

→ *Semantic Business Process Models (SBPM)*

Semantic Business Process Models

- Combination of process modeling methods with semantic technologies

two semantic business process models



```
<petri:Place rdf:ID="request">
  <petri:transRef rdf:resource="#check_request"/>
    ....
</petri:Place>
<petri:Transition rdf:ID="send_rejection">
  <petri:placeRef>
    <petri:Place rdf:ID="rejection">
      <petri:hasMarking>
        <petri:IndividualDataItem rdf:ID="R_rejection">
          <petri:hasAttribute rdf:resource="#Destination"/>
          <petri:hasAttribute rdf:resource="#Quantity"/>
          <petri:hasAttribute rdf:resource="#Name"/>
        </petri:IndividualDataItem>
      </petri:hasMarking>
    </petri:Place>
  </petri:placeRef>
 </petri:Transition>
 <petri:Transition rdf:ID="send_confirmation">
  <petri:placeRef>
    <petri:Place rdf:ID="confirmation">
    .......
 </petri:Place>
</petri:placeRef>
</petri:Transition>
.....
```

```
<petri:Place rdf:ID="request">
  <petri:hasMarking rdf:resource="#R_request"/>
  <petri:transRef rdf:resource="#reject"/>
  <petri:transRef rdf:resource="#accept"/>
 </petri:Place>
 <petri:Transition rdf:ID="reject">
  <petri:placeRef>
    <petri:Place rdf:ID="rejection">
      <petri:hasMarking>
        <petri:IndividualDataItem rdf:ID="R_rejection">
          <petri:hasAttribute rdf:resource="#Name"/>
          <petri:hasAttribute rdf:resource="#City"/>
          <petri:hasAttribute rdf:resource="#Date"/>
        </petri:IndividualDataItem>
      </petri:hasMarking>
      <petri:transRef rdf:resource="#contact_customer"/>
    </petri:Place>
  </petri:placeRef>
 </petri:Transition>
 <petri:Transition rdf:about="send_verification">
  <petri:placeRef>
   <petri:Place rdf:about="verification">
   .....
 </petri:Transition>
 ......
```

Outline

1. Motivation

2. Foundations

3. Measuring Similarity between Process Elements

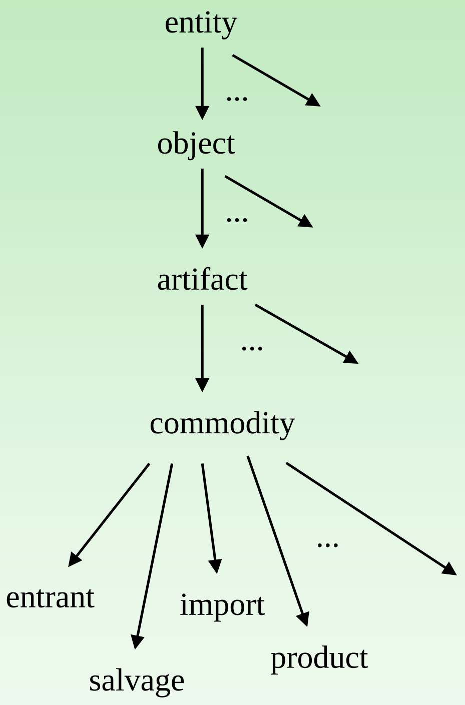
4. Analysis Methods for Petri nets

5. Conclusion and Outlook

Measuring Similarity between Process Elements

Input:

- two semantic business process models
- background ontology:



Measuring Similarity between Process Elements

Similarity Computation

- syntactical similarity: how many changes are required to transform one string into the other one

- linguistic similarity: via a background ontology

- structural similiarity: considering context of process model elements

→ *combined similarity*

Measuring Similarity between Process Elements

Output

combined similarity results

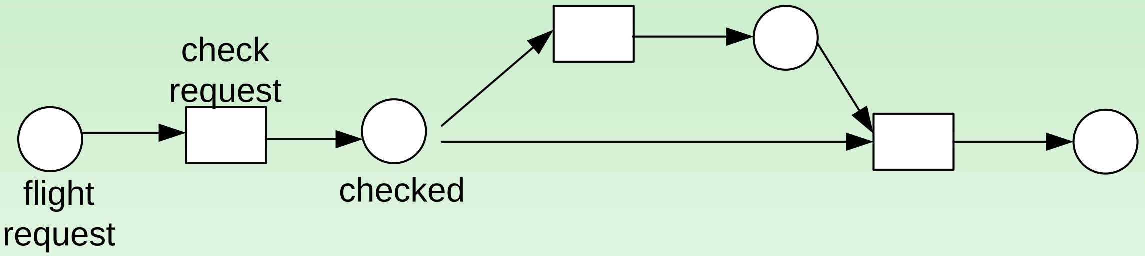
name	sim_{com}	sim_{syn}	sim_{ling}	sim_{str_p}	$sim_{str_{At}}$	sim_{str_V}	sim_{str_T}
#confirmation	0.8	0.64	0.5	0.95	0.0	0.0	1.0
#verification	0.8	0.64	0.5	0.95	0.0	0.0	1.0

Outline

1. Motivation
2. Foundations
3. Measuring Similarity between Process Elements
4. ⇨ Analysis Methods for Petri nets
5. Conclusion and Outlook

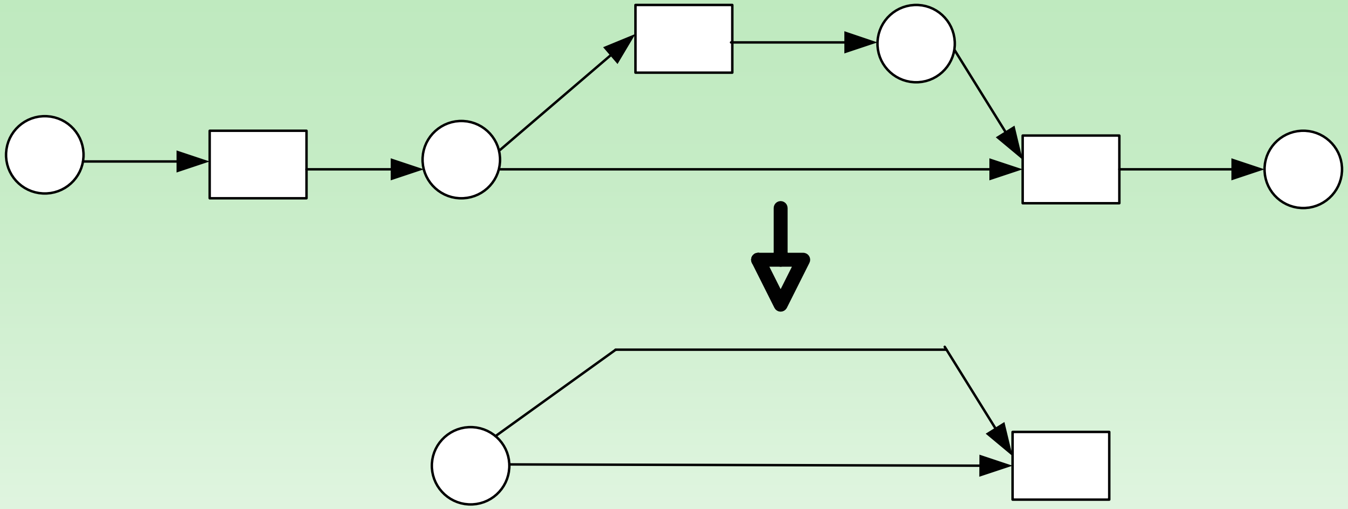
Analysis Methods for Petri nets

- proposed process fragments should not cause deadlocks and synchronization errors



→ free of structural errors if the reduction results in an empty graph

Analysis Methods for Petri nets



→ two nodes left over, thus the modeled process is not free of deadlocks

Conclusion

- Assisting users in process models modeling to improve the reusability of business processes
- Similarity measurement to solve ambiguity issues caused by the use of different names for describing the same tasks
- Recommendation system validates process properties of the automatically completed process to avoid deadlocks and synchronization errors.

Outlook

- Use of machine learning techniques

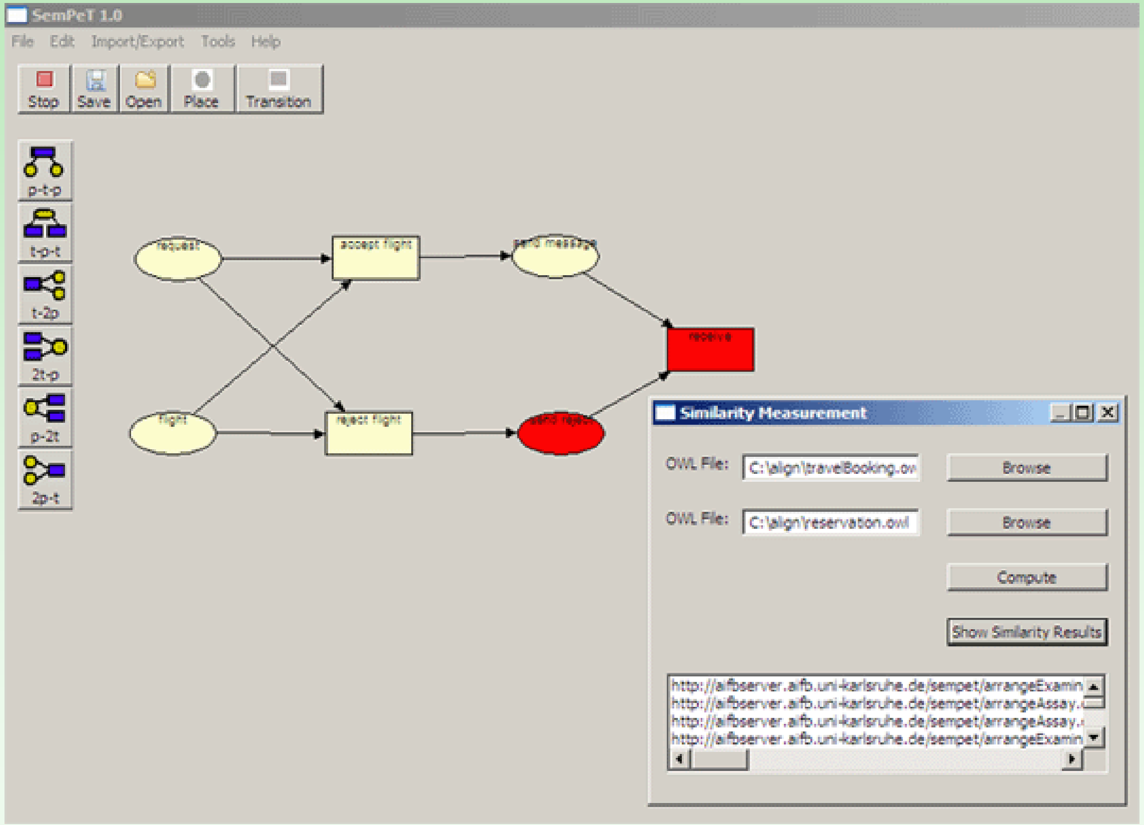
  Neural Nets: learn a threshold for combined similarity – instead of using a threshold with fixed value

  Information Filtering: generate and compare user-profiles, which help to categorize users

  Content-based Information Filtering: combined similarity of process template elements can be used to rank the recommendations

Implementation

http://aifbserver.aifb.uni-karlsruhe.de/sempet/index.htm



OWL File: C:\align\travelBooking.ow

OWL File: C:\align\reservation.owl

```
http://aifbserver.aifb.uni-karlsruhe.de/sempet/arrangeExamin
http://aifbserver.aifb.uni-karlsruhe.de/sempet/arrangeAssay.
http://aifbserver.aifb.uni-karlsruhe.de/sempet/arrangeAssay.
http://aifbserver.aifb.uni-karlsruhe.de/sempet/arrangeExamin
```