

Verteidigung Bachelorarbeit

# **Evaluierung der Konsistenz zwischen Business Process Modellen und Business Role-Object Spezifikation**

---

Lars Westermann

26.09.2019

Institut für Software- und Multimediatechnik, Professur für Softwaretechnologie

Prof. Dr. rer. nat. habil. Uwe Aßmann

Dr.-Ing. Thomas Küh

Hendrik Schön

Automatische Codegenerierung durch konsistente Modelle

Allerdings Diskrepanz zwischen Modellierungaspekten:

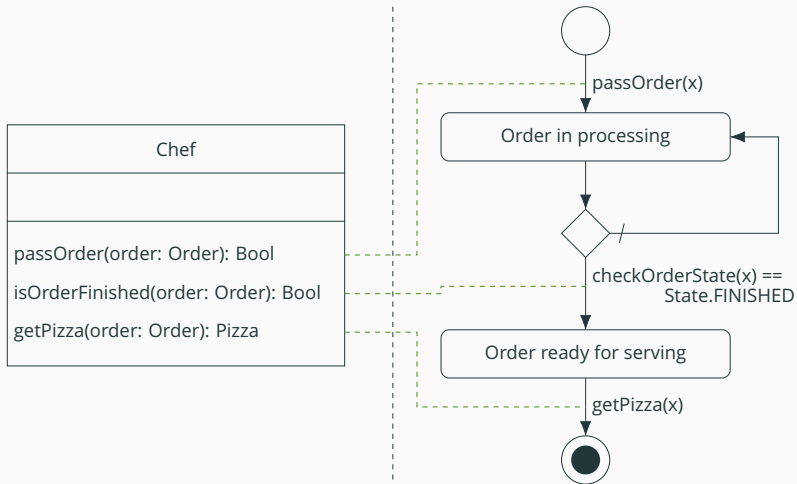
- **Strukturmodellierung**

- UML-Klassen- oder Komponentendiagrammen

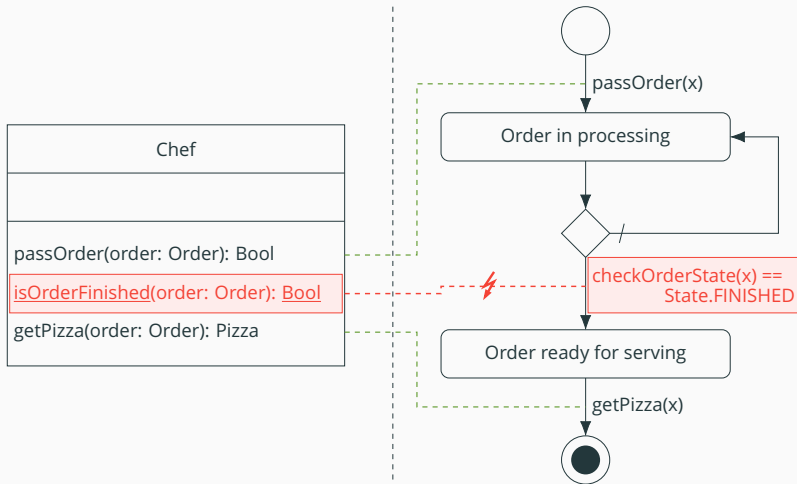
- **Verhaltensmodellierung**

- UML-Sequenzdiagrammen oder Petrinetzen

# Motivation



# Motivation



Bestehende Verfahren zur Konsistenzprüfung:

- UML-Klassendiagramm und UML-Zustandsdiagramm
- ...

Gilt nicht für die Konsistenz zwischen **BPMN** und **BROS**:

- Es existieren noch keine Konsistenzbeziehungen
- Damit auch kein automatisierbares Verfahren zur Konsistenzprüfung

- F1** Welche Konsistenzbeziehungen bestehen zwischen BPMN- und BROS-Modellen?
- F2** Wie lassen sich die Konsistenzbedingungen automatisiert überprüfen?
- F3** Mit welchem Aufwand ist dieses Verfahren erweiterbar?

**Welche Konsistenzbeziehungen  
bestehen zwischen BPMN- und  
BROS-Modellen?**

---

Die **Konsistenz**: Zusammenhang, Widerspruchsfreiheit [Duden 2019]

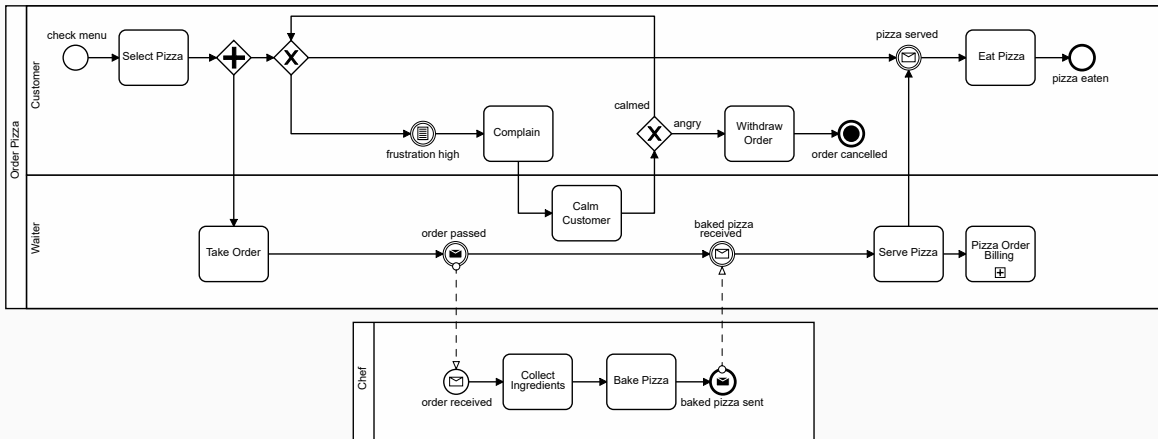
Eine **Inkonsistenz** tritt genau dann auf, wenn eine Konsistenzregel verletzt wird. [Nuseibeh 1996]

Eine **Konsistenzregel** ist eine Formalisierung von einem Aspekt der Konsistenz zwischen den betrachteten Modellen. [vgl. Nuseibeh 1996]

Das **Konsistenzproblem** beschreibt den Vorgang der Minimierung und Verhinderung von Inkonsistenzen mit Hilfe der Aufstellung und Prüfung von Konsistenzregeln.

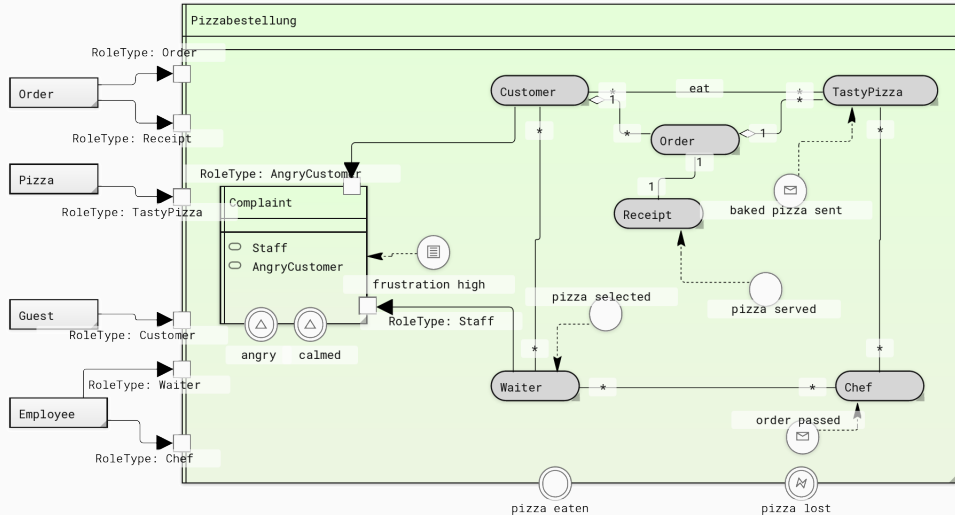


# Business Process Model and Notation<sup>1</sup>



<sup>1</sup><https://www.omg.org/spec/BPMN/2.0/About-BPMN/>

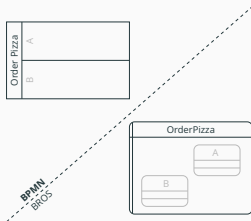
# Business Role-Object Specification<sup>2</sup>



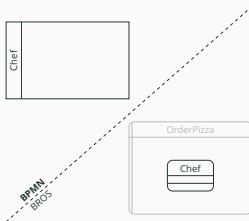
```
1  % Definition aller BPMN Elemente.  
2  bpmn(Bpmn, Type).  
3  
4  % Definition aller BROS Elemente.  
5  bros(Bros, Type).  
6  
7  % Definition aller Relationen.  
8  relation(Source, Target, Type).  
9  
10 % Definition der Eltern–Kind Beziehung.  
11 parent(Child, Parent).  
12  
13 % Orakel für das Matching von Modellelementen.  
14 match(Bpmn, Bros).
```

---

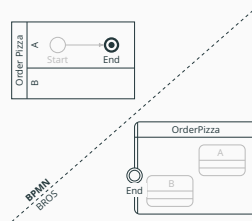
# Konsistenzbeziehungen



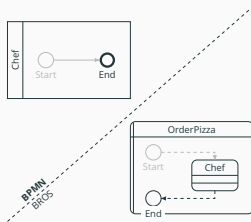
Regel 1: BPMN-Process



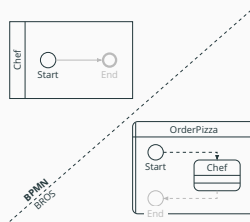
Regel 2: BPMN-Swimlane



Regel 3: BPMN-TerminationEvent

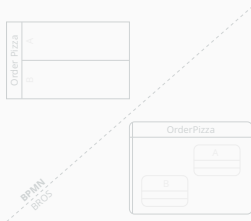


Regel 4: BPMN-EndEvent

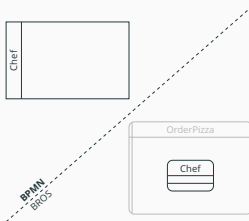


Regel 5: BPMN-StartEvent

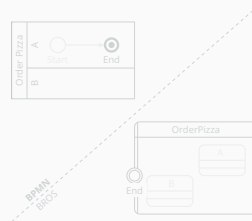
# Konsistenzbeziehungen



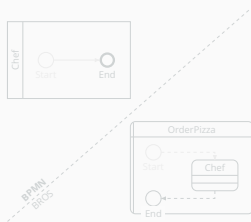
Regel 1: BPMN-Process



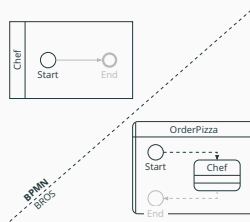
Regel 2: BPMN-Swimlane



Regel 3: BPMN-TerminationEvent

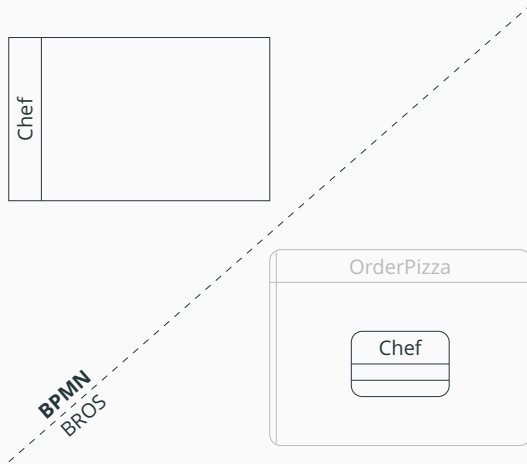


Regel 4: BPMN-EndEvent



Regel 5: BPMN-StartEvent

# BPMN-Swimlane - BROS-RoleType



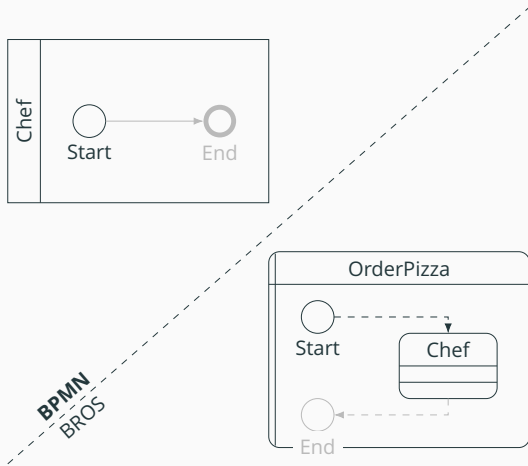
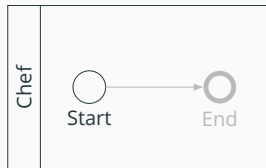
## BPMN-Swimlane - BROS-RoleType

---

```
1 rule_2(Bpmn) :- bpmn(Bpmn, "Swimlane") ->
2   (
3     bros(Bros, "RoleType"), match(Bpmn, Bros)
4   ).
```

---

# BPMN-StartEvent - BROS-Event



**BPMN**  
BROS



## BPMN-StartEvent - BROS-Event

---

```
1 rule_5(Bpmn) :- bpmn(Bpmn, "StartEvent") ->
2   (
3     bros(Bros, "Event"),
4     match(Bpmn, Bros),
5     (
6       relation(Bros, X, "CreateRelation"),
7       transitive_parent(Bpmn, BpmnParent),
8       match(BpmnParent, X)
9     )
10  ).
```

---

**Wie lassen sich die  
Konsistenzbedingungen  
automatisiert überprüfen?**

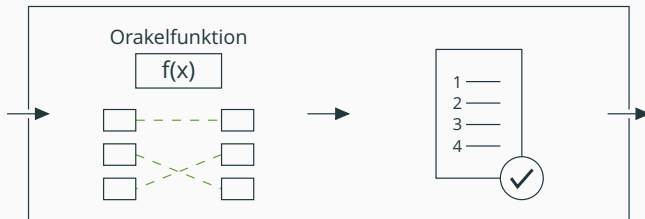
---

# Ablauf der Konsistenzprüfung

## Modelle



## Matching

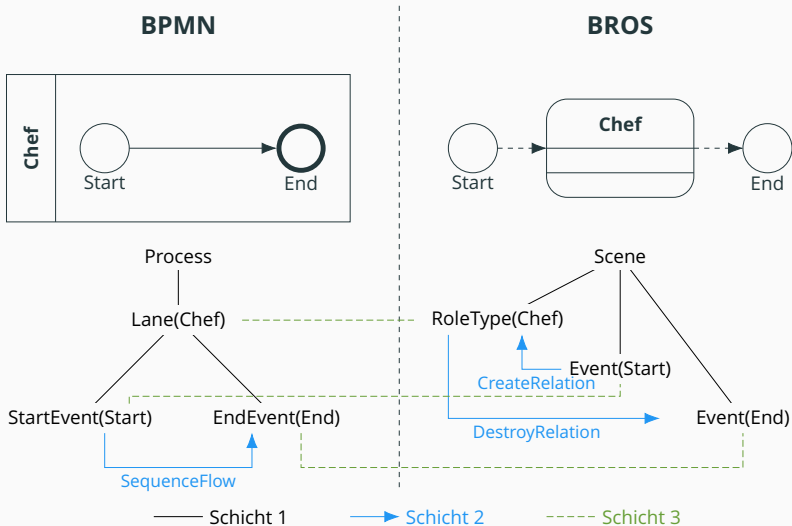


## Verifikation



## Ergebnis





Matching der Modellelemente anhand des Namen und des Typs

Algorithmus des Name-Matching:

1. Namen in Teilwörter aufteilen (Anhand von " " und Groß/Kleinschreibung)
2. Endung der Teilwörter entfernen (Letzte 2 Zeichen entfernen)
3. Alle Teilwörter des kürzeren Namens müssen im längerem Namen enthalten sein

'Select pizza' , 'pizza selected'  
{ 'select', 'pizza' } , { 'pizza', 'selected' }  
{ '**select**'  $\subseteq$  '**selected**' } , { '**pizza**'  $\subseteq$  '**pizza**' }

# Matching

---

```
1 Context.match<BpmnLane, BrosRoleType> { lane, role ->
2   return matchStrings(lane.element.name, role.element.name)
3 }
```

---

---

```
1 rule_2(Bpmn) :- bpmn(Bpmn, "Swimlane") ->
2   (
3     bros(Bros, "RoleType"), match(Bpmn, Bros)
4   ).
```

---



---

```
1 Context.verifyBpmn<BpmnLane> { bpmn ->
2   for (bros in bpmn.matchingElements) {
3     if (bros.checkType<BrosRoleType>()) {
4       return Result.match("...", bros = bros)
5     }
6   }
7   return Result.error("...")
8 }
```

---

## Positive und Negative Konsistenzmeldungen

- Referenz auf Modellelemente
- Regel die zur Konsistenzmeldungen geführt hat
- Textuelle Beschreibung der Ursache

BPMN ID: StartEvent\_0wk87jv  
BpmnStartEvent(check menu)

BROS

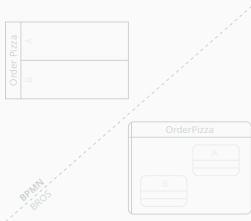
Module  
Rule 5 - BpmnStartEvent

Message  
Cannot find matching BrosElement for BpmnStartEvent(check menu)

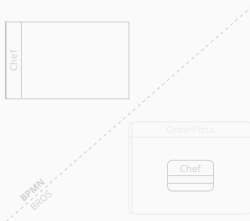
**Mit welchem Aufwand ist dieses  
Verfahren erweiterbar?**

---

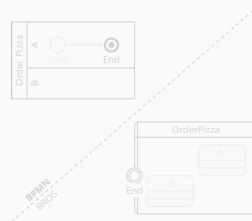
# Konsistenzbeziehungen



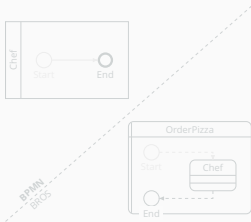
Regel 1: BPMN-Process



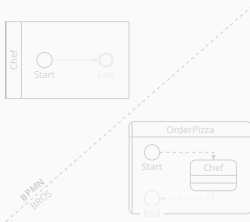
Regel 2: BPMN-Swimlane



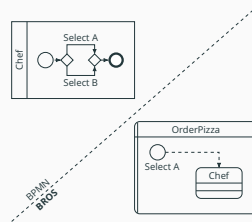
Regel 3: BPMN-TerminationEvent



Regel 4: BPMN-EndEvent

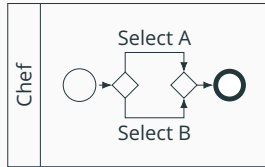


Regel 5: BPMN-StartEvent

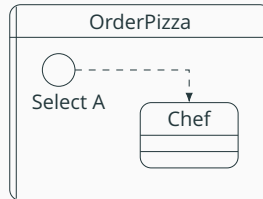


Regel 6: BROS-Event

# BROS-Event - BPMN-Element



BPMN  
BROS



## BROS-Event - BPMN-Element

---

```
1 rule_6(Bros) :- (bros(Bros, "Event"); bro(Bros, "ReturnEvent")) ->
2   (
3     bpmn(Bpmn, "StartEvent"), match(Bpmn, Bros);
4     bpmn(Bpmn, "EndEvent"), match(Bpmn, Bros);
5     bpmn(Bpmn, "TerminationEvent"), match(Bpmn, Bros);
6     bpmn(Bpmn, "Event"), match(Bpmn, Bros);
7     bpmn(Bpmn, "Activity"), match(Bpmn, Bros);
8     bpmn(Bpmn, "Gateway"), match(Bpmn, Bros)
9   ).
```

---

## BROS-Event - BPMN-Element

---

```
1 fun Context.setupRule6 {
2     match<BpmnTask, BrosEvent> { bpmn, bros ->
3         return matchStrings(bpmn.element.name, bros.element.desc)
4     }
5     verifyBros<BrosEvent> { bros ->
6         for (bpmn in bros.matchingElements) {
7             if (bpmn.checkType<BpmnElement>()) {
8                 return Result.match("...", bpmn = bpmn)
9             }
10        }
11        return Result.error("...")
12    }
13 }
14 val activeModules = listOf(... , Context::setupRule6)
```

---

## **Ergebnisse und Ausblick**

---



**F1** Welche Konsistenzbeziehungen bestehen zwischen BPMN- und BROS-Modellen?

- Fünf Konsistenzregeln aufgestellt

**F2** Wie lassen sich die Konsistenzbedingungen automatisiert überprüfen?

- Tool als Referenzimplementierung erstellt

**F3** Mit welchem Aufwand ist dieses Verfahren erweiterbar?

- Niedrig, Aufwand exemplarisch verdeutlicht

# Klassifikationsschema

	Diagrams	Consistency Type	Consistency Strategy	Intermediate Representation	Case Study	Automatable	Tool Support	Model Extensibility	Rule Extensibility
Rasch 2003	CD, SM	Intra	Monitoring	CSP/OZ	✓	●	✗	●	◐
Shinkawa 2006	UCD, CD, SD, AD, SC	Inter	Analysis	CPN	✗	●	✗	◐	○
Mens 2005	CD, SD, SC	All	Monitoring	Extended UML	✓	●	✓	●	◐
Egyed 2001	CD, OD, SD	Intra, Inter	Construction		✗	●	~	◐	◐
Egyed 2006	CD, SD, SC	Intra	Monitoring		✓	●	✓	○	◐
BBV	BPMN, BROS	Intra	Monitoring		✓	●	✓	○	●

● mit geringem Aufwand

◐ mit mittlerem Aufwand

○ mit hohem Aufwand

✓ ja

✗ nein

~ teilweise

**CD** Class Diagram

**SM** State Machine





**USC** Use Case Diagram




**SD** Sequence Diagram

**AD** Activity Diagram

**SC** Statechart

**BBV** BPMN-BROS-Verifier

-  Duden (2019). *Duden | Konsistenz*. URL:  
<https://www.duden.de/rechtschreibung/Konsistenz> (visited on 09/24/2019).
-  Egyed, Alexander (2001). ?Scalable consistency checking between diagrams-The VIEWINTEGRA approach? In: *Proceedings 16th Annual International Conference on Automated Software Engineering (ASE 2001)*. IEEE, pp. 387–390.
-  – (2006). ?Instant consistency checking for the UML? In: *Proceedings of the 28th international conference on Software engineering*. ACM, pp. 381–390. DOI:  
10.1145/1134285.1134339.
-  Mens, Tom, Ragnhild Van Der Straeten, and Jocelyn Simmonds (2005). ?A framework for managing consistency of evolving UML models? In: *Software Evolution with UML and XML*. IGI Global, pp. 1–30.

-  Nuseibeh, Bashar (1996). ?To be and not to be: On managing inconsistency in software development? In: *Proceedings of the 8th International Workshop on Software Specification and Design*. IEEE, pp. 164–169.
-  Rasch, Holger and Heike Wehrheim (2003). ?Checking Consistency in UML Diagrams: Classes and State Machines? In: *International Conference on Formal Methods for Open Object-Based Distributed Systems*. Springer, pp. 229–243. DOI: 10.1007/978-3-540-39958-2\_16.
-  Schön, Hendrik et al. (2019). ?Business Role-Object Specification: A Language for Behavior-aware Structural Modeling of Business Objects? In: AIS, pp. 244–258.



Shinkawa, Yoshiyuki (2006). **Inter-Model Consistency in UML Based on CPN Formalism**? In: *2006 13th Asia Pacific Software Engineering Conference (APSEC'06)*. IEEE, pp. 411–418. DOI: 10.1109/apsec.2006.41.

## Demo

---

**Fragen?**

# Implementierung des Name-Matching

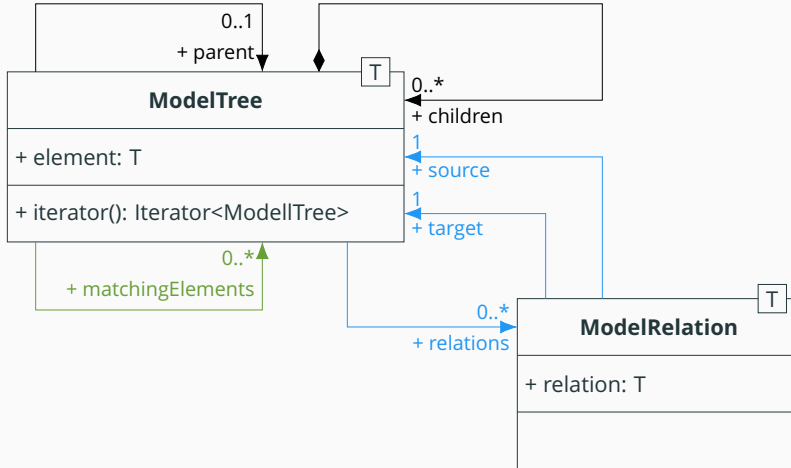
---

```
1 fun matchStrings(shorterName: String, longerName: String): Boolean {
2     val shorterNameSet = splitNameToSet(shorterName)
3     val longerNameSet = splitNameToSet(longerName)
4
5     return shorterNameSet.all { short ->
6         val sh = trimEnding(short)
7
8         return longerNameSet.any { long ->
9             val lo = trimEnding(long)
10
11             return long.startsWith(sh) || short.startsWith(lo)
12         }
13     }
14 }
```

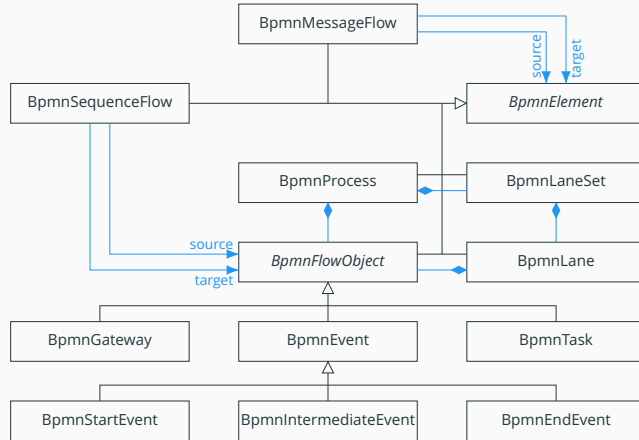
---



# Metamodell der Graphstruktur



# Metamodell der Business Process Model and Notation



# Metamodell der Business Role-Object Specification

