



„Ein Konzept zur Prüfung von Web Services auf kompositionelle Verklemmungen“

Martin Gebauer, Stefan Kühne, Maik Thränert

GEFÖRDERT VOM



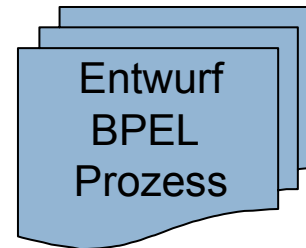
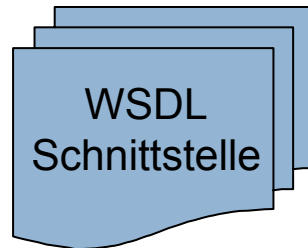
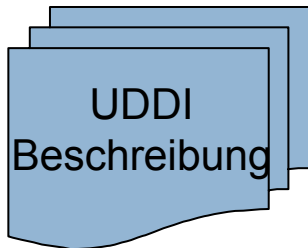
Bundesministerium
für Bildung
und Forschung

Das Forschungs- und Entwicklungsprojekt OrViA wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) gefördert, die innerhalb der zweiten Auswahlrunde der Forschungsoffensive „Softwareengineering 2006“ vergeben wurden, und vom Deutschen Zentrum für Luft- und Raumfahrt (DLR), Projektträger Informationstechnik/ Softwaresysteme betreut.

- Problemstellung
- Stand der Forschung
- Lösungsansatz
- Bewertung und Ausblick

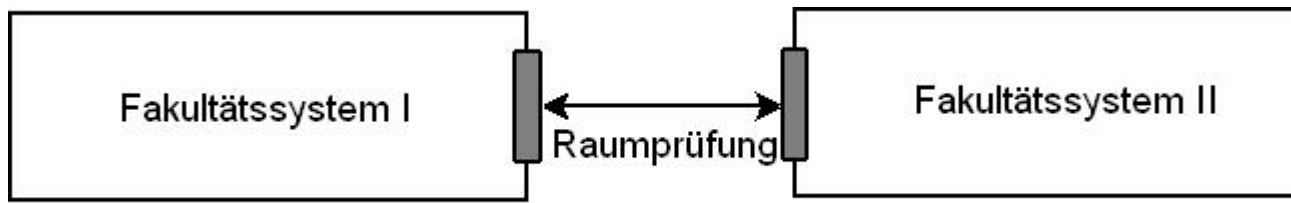
- Folgen der Virtualisierung von Unternehmen
 - neue Organisationsstrukturen
 - veränderte Anforderungen an Unterstützung durch Informations- und Kommunikationstechnologien
- Ansatz
 - Service orientierte Architekturen (SOA) + Realisation durch Web Services
- Verteilte Systeme
- Verteilte Geschäftsprozesse
- Integrationsprobleme (EAI, B2B)
- Zuverlässigkeit der einzelnen Komponenten immens wichtig
- Ausfälle von Prozessen durch Ausfall einzelner Services bergen hohes finanzielles Risiko
 - Ausfall ist nicht selbst zu verantworten aber zu „finanzieren“
 - Wunsch nach Sicherheit des Zusammenspiels der einzelnen Komponenten

- Möglichkeit für solche Ausfälle: Verklemmungen (Deadlocks)
- Eigene Services – unkritisch
- Nutzung fremder Services basiert auf einem Vertragsverhältnis
→ Service arbeitet in sich fehlerfrei
- Wie kann das Verhalten in einer Komposition sichergestellt werden?
- Welche Informationen liegen zum Kompositionszeitpunkt vor?






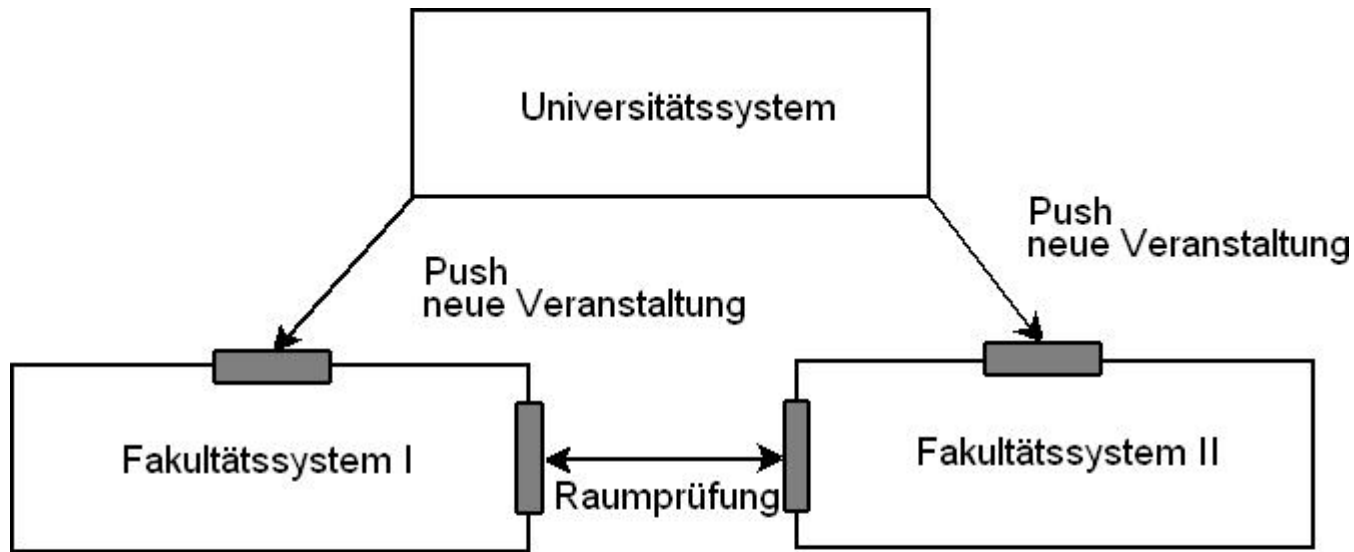
- nicht technische Lösung über Verträge zwischen Systemhersteller, Serviceanbieter und Systemanwendern
- Informationen über Abhängigkeiten unter den Services

```
<definitions>
  <types>
    <schema></schema>
  </types>
  <message>
    <part></part>
  </message>
  <portType>
    <operation>
      <input></input>
      <output></output>
    </operation>
  </portType>
  <binding>
    <operation>
      <input></input>
      <output></output>
    </operation>
  </binding>
  <service>
    <port></port>
  </service>
</definitions>
```






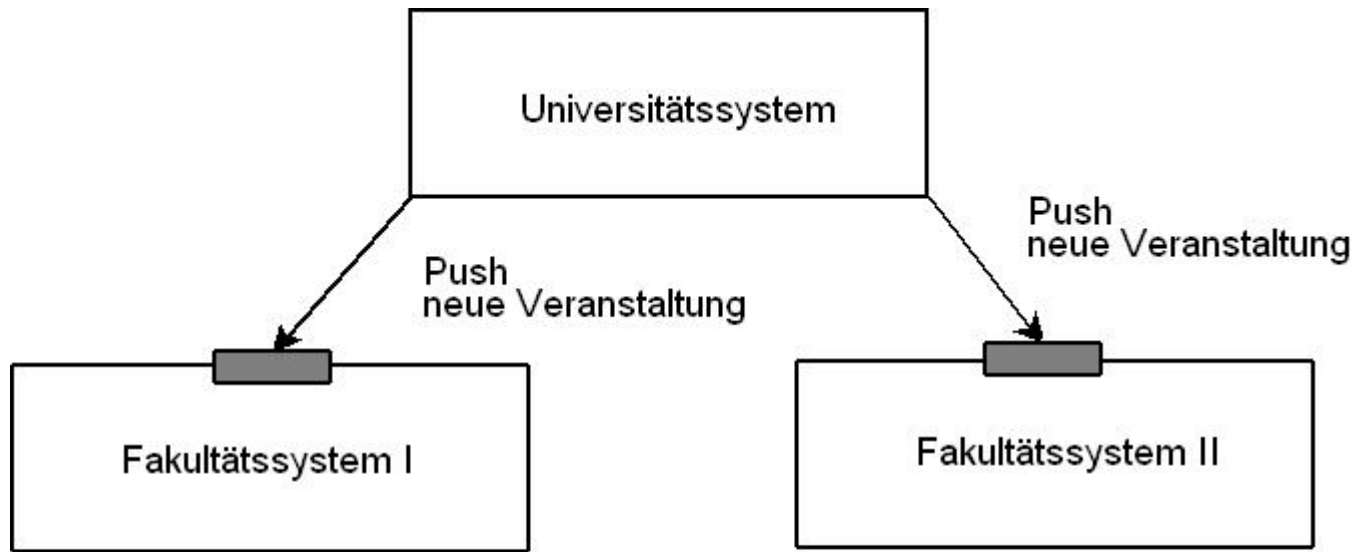
Legende:

-  Schnittstelle (z. B. WSDL)
-  System
-  Verknüpfung von Systemen (BPEL-Prozess, proprietär)






Legende:

-  Schnittstelle (z. B. WSDL)
-  System
-  Verknüpfung von Systemen (BPEL-Prozess, proprietär)



Legende:

-  Schnittstelle (z. B. WSDL)
-  System
-  Verknüpfung von Systemen (BPEL-Prozess, proprietär)

- Formale Repräsentation der Services
- Formale Repräsentation der verbindenden Prozesse
- Eignung für eine Untersuchung auf Verklemmungenfreiheit
- Abbildung mit XML
- Verfügbarkeit zum Kompositionszeitpunkt

```
<process name="ncname" targetNamespace="uri "
```

```
...  
xmlns="http://schemas.xmlsoap.org/ws/  
2003/03/business-process/">
```

```
<partnerLinks>...</partnerLinks>
```

```
<variables>...</variables>
```

```
<correlation Sets>...</ correlation Sets>
```

```
<faultHandlers>...</faultHandlers>
```

```
<compensationHandler>
```

```
...  
</compensationHandler>
```

```
<eventHandlers>...</eventHandlers>
```

```
activity
```

```
</process>
```

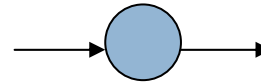
- entstanden als Kombination aus XLANG (Microsoft) und WSFL (IBM)
- aktuell Version 1.1 (Version 2.0 Public Review 10.09.2006 – 09.11. 2006)
- Prozessbeschreibung, Orchestrierung von Web Services



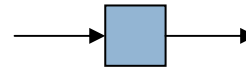
Carl Adam Petri

- „Kommunikation mit Automaten“ (1962 TU Darmstadt)

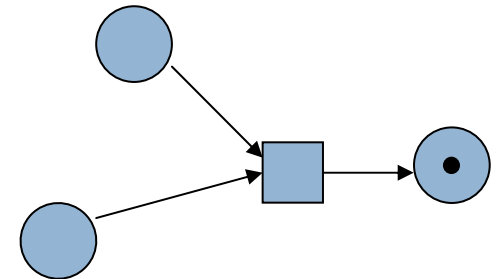
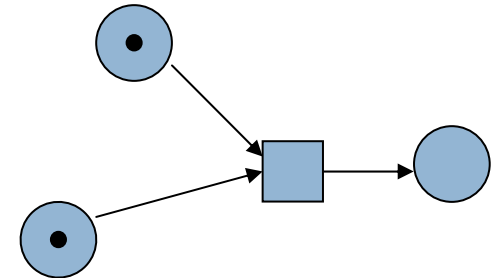
Stelle



Transition



Stelle mit Marken

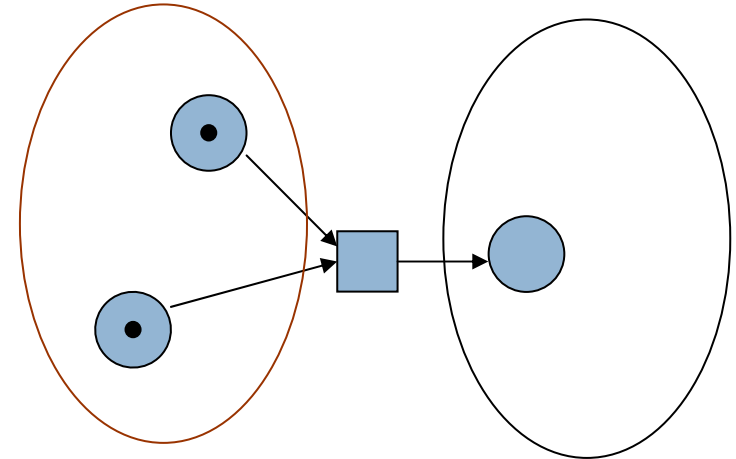


- *Ein Tripel $N = (P, T, F)$ ist genau dann ein Netz wenn es aus zwei disjunkten Mengen P und T und einer Relation $F \subseteq (P \times T) \cup (T \times P)$ besteht.*

Sei $X = P \cup T$ die Vereinigung der Mengen der Stellen und der Transitionen, so ist für ein Element $x \in X$

Vorbereich: $\bullet x = (y \in X \mid (x, y) \in F)$

Nachbereich: $x \bullet = (y \in X \mid (y, x) \in F)$



Das 6 Tupel $Y = (P, T, F, K, W, m_0)$ ist ein Stellen-Transitions-Netz sofern (P, T, F) ein Netz ist,

$K : S \rightarrow \mathbb{N} \cup \{\infty\}$ Kapazitäten der Stellen, möglicherweise

unbegrenzt

$W : F \rightarrow \mathbb{N}$ Kantengewichte der Kanten und

$m_0 : S \rightarrow \mathbb{N}$ Anfangsmarkierung, wobei $p \in P : m_0(p) \leq K(p)$.

- Workflow Netz
 - Ein endliches Petrinetz $N = (P, T, F)$ heißt Workflow Netz
 - ausgezeichnete Anfangsstelle $\alpha \in P$ mit $\bullet\alpha = \emptyset$
 - ausgezeichnete Endstelle $\omega \in P$ mit $\omega\bullet = \emptyset$
 - alle anderen Elemente liegen auf Pfaden zwischen α und ω

(Martens, Axel: *Verteilte Geschäftsprozesse - Modellierung und Verifikation mit Web Services*.
WiKu Verlag, 2004)

(Aalst, W.M. P. van d.: Structural Characterizations of Sound Workflow Nets. In: *Computing Science Reports/23* (1996), Nr. 96)

→ Diese Definition unterstellt ein endliches Verhalten

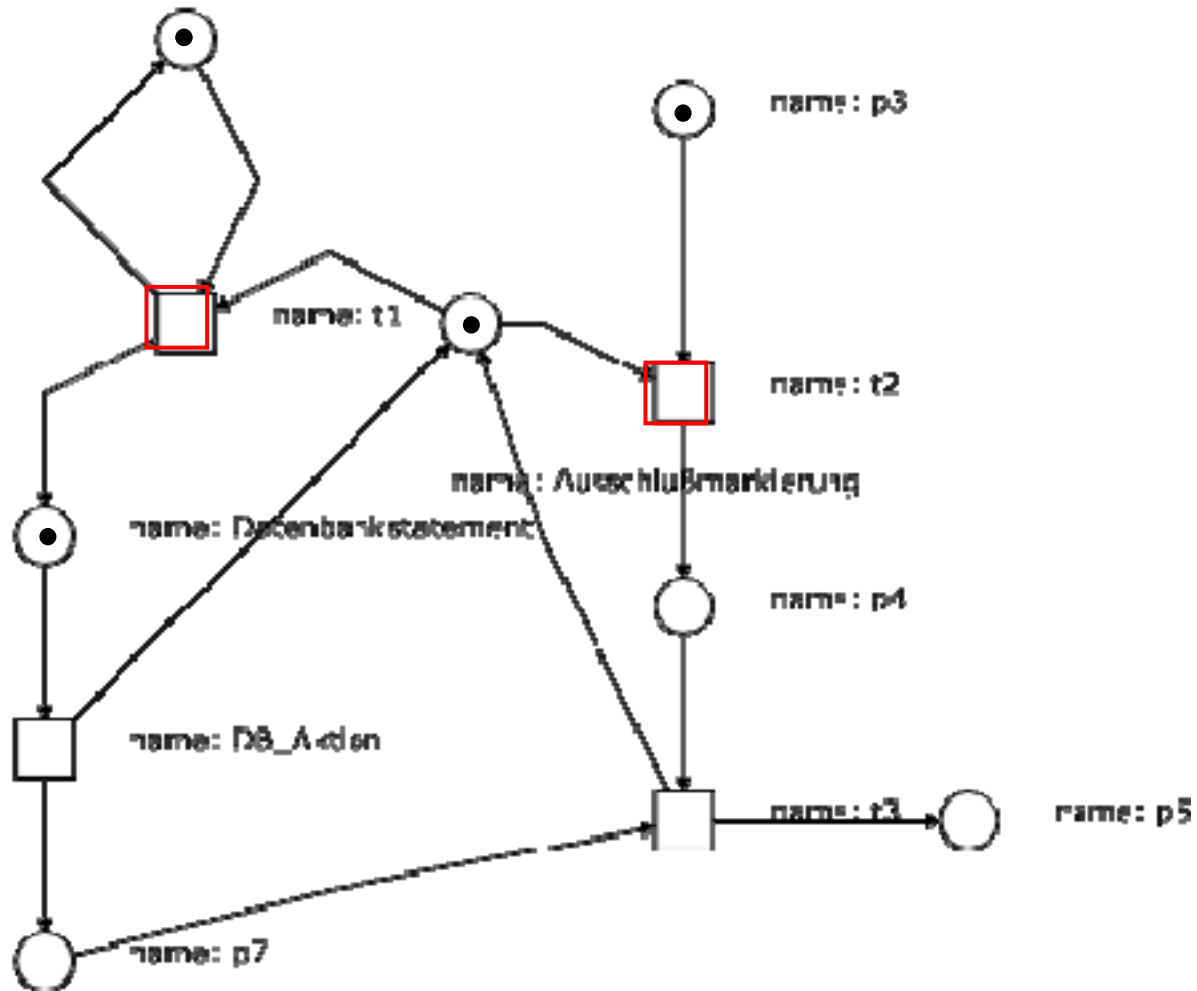
- „Wir betrachten einen Web Service als Implementierung eines lokalen Geschäftsprozesses.“

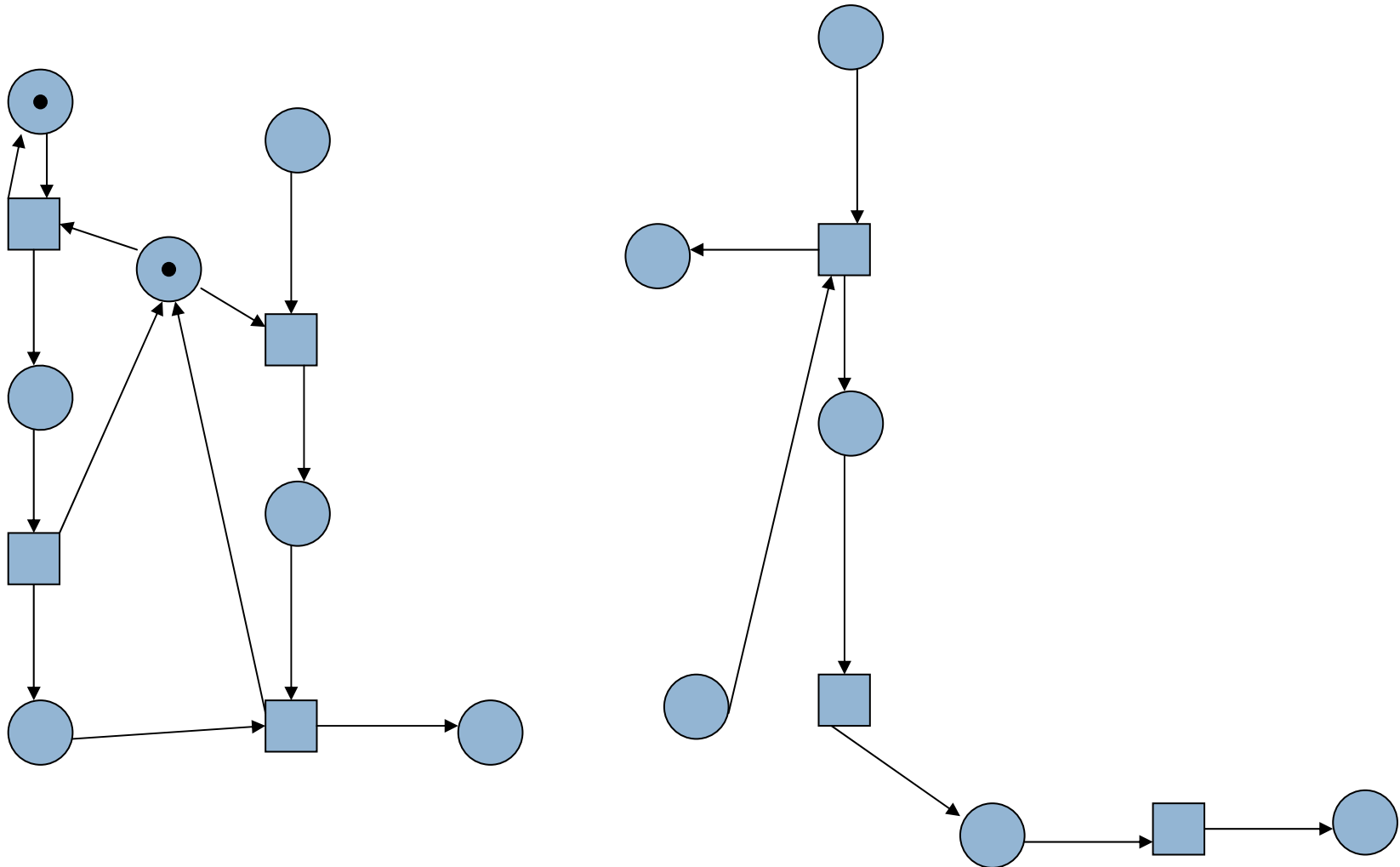
(Martens, Axel: *Verteilte Geschäftsprozesse - Modellierung und Verifikation mit Web Services*.
WiKu Verlag, 2004)

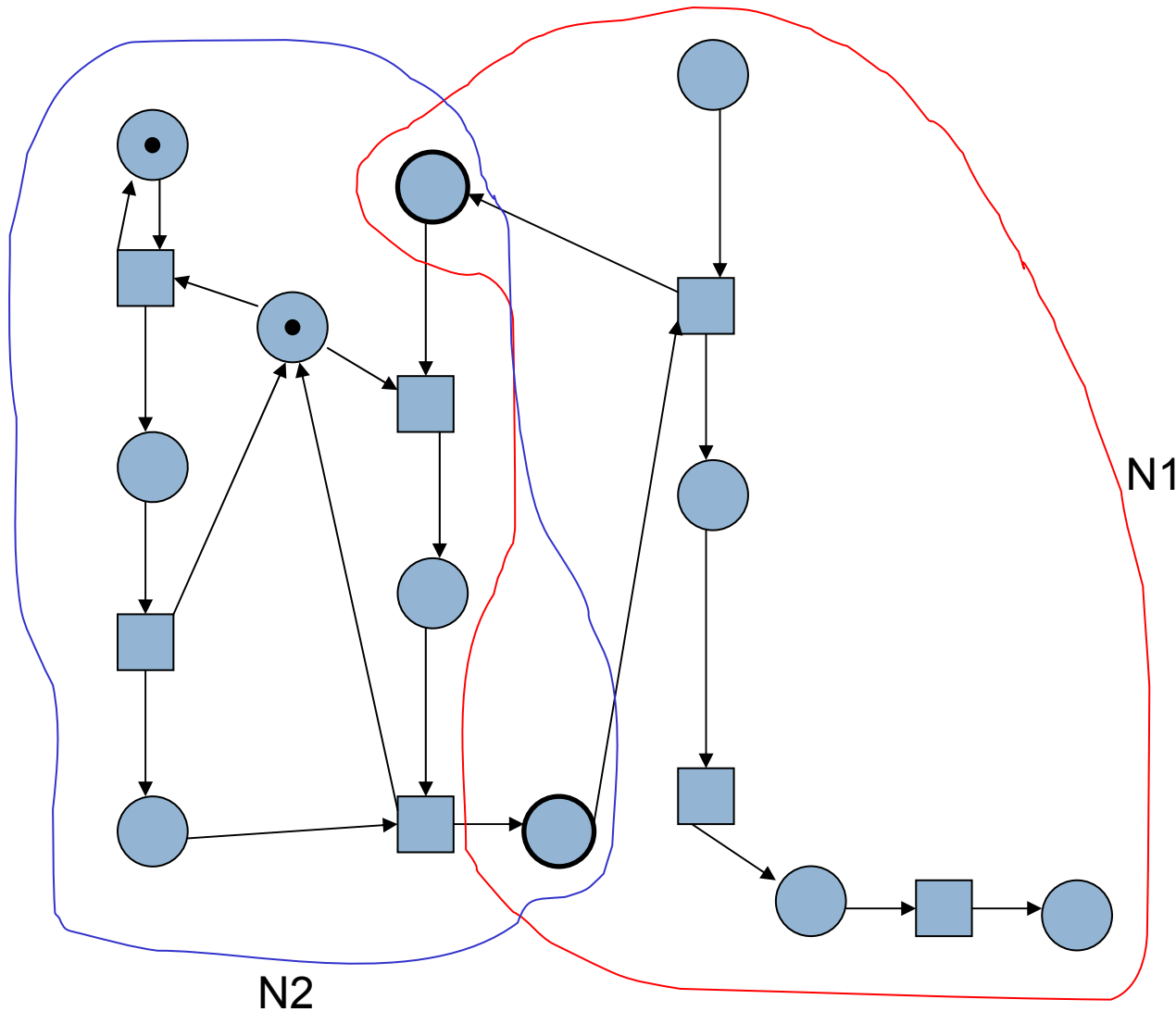
- Zusammenspiel der Komponenten
- Verklemmung (Deadlock):
 - „Von einer Verklemmung spricht man dann, wenn sich nichts mehr bewegen kann.“
(Starke, Peter H.: *Analyse von Petri-Netz-Modellen*. B.G. Teubner, 1990 S. 54)
- Netzeigenschaften
 - Erreichbarkeit
 - Lebendigkeit
 - Verklemmungsfreiheit (schwache Lebendigkeit)
 - Soundness

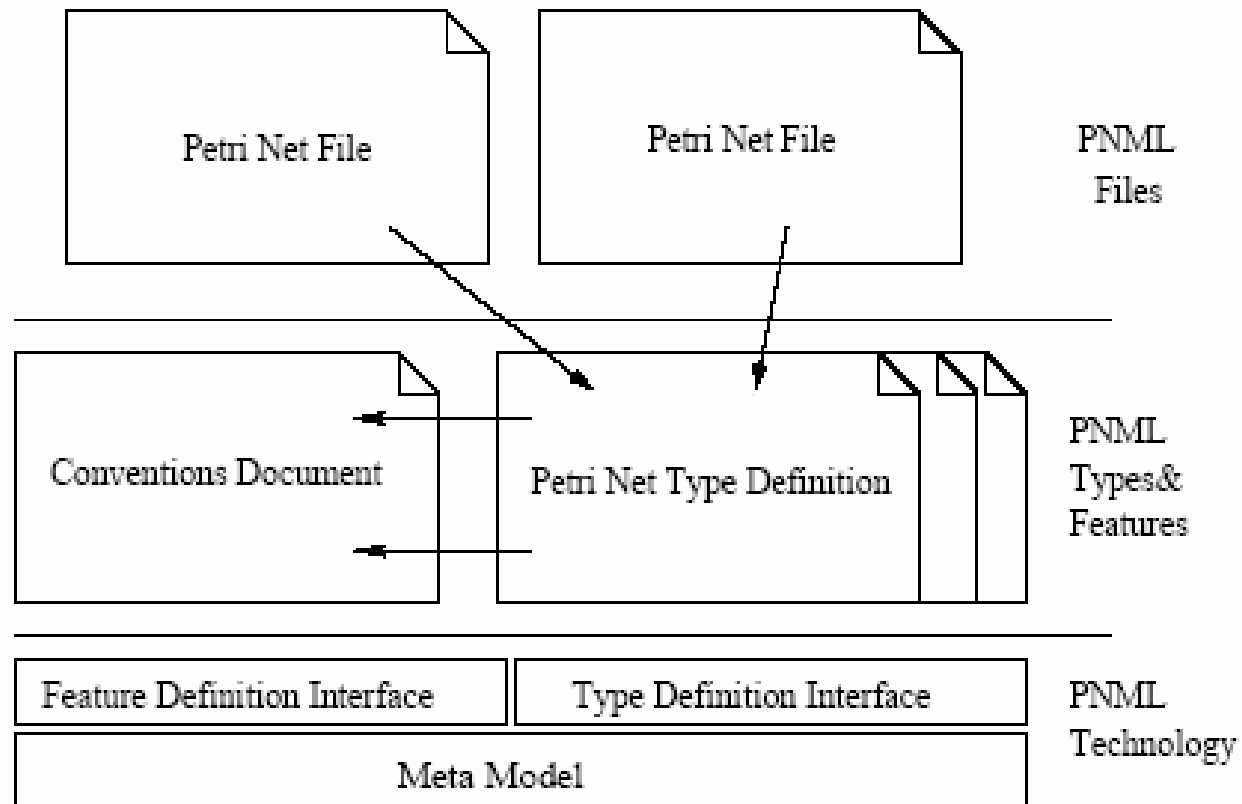
name: Wetterservice

name: Sensorwert









- ISO/IEC - 15909, International Standard on Petri Net Markup Language (PNML)

```
<pnml>
  <net id="n1" type="PTNet">
    <name>
      <value>PNML_GRUNDSTRUKTUR</value>
    </name>

    <place id="p1">
      <marking>
        <graphics>
          <offset page="1" x="20" y="16"/>
        </graphics>
        <value>0</value>
      </marking>
      <name>
        [...]
        <value>p1</value>
      </name>
      <initialMarking>
        [...]
        <value>0</value>
      </initialMarking>
      [...]
    </place>
```

```
<transition id="t1">
  <name>
    [...]
    <value>t1</value>
  </name>
  [...]
</transition>
<arc id="a1" source="p1" target="t1">
  <inscription>
    [...]
    <value>1</value>
  </inscription>
  [...]
</arc>
</net>
</pnml>
```

- Erweiterungsbedingungen aus Sicht von WSDL 2.0
 - Open content Model
 - Erlaubt die Verwendung von Elementen und Attributen aus nicht WSDL Namespaces
 - Features und Properties
 - Features beschreiben abstrakte Funktionalitäten im Zusammenhang mit dem Nachrichtenaustausch
 - Properties sind sozusagen Laufzeitvariablen
- Entscheidung für das Open content Model

- Attributbezogene Erweiterbarkeit
 - Elementerweiterungen mit alternativen Namespaces
 - `<service extension-attribute="?">`
 - Keine Möglichkeit solche Erweiterungen als optional zu kennzeichnen
- Elementbezogene Erweiterbarkeit
 - Basistyp für Erweiterungen
 - `<xs : complexType name= ' Extensibility Element ' abstract= ' true '>`
 - `<xs : attributeref = ' wsdl : required ' use=' optional ' />`
 - `</xs: complexType>`
- Element für Erweiterungen `<service>`
- Wiederverwendbarkeit abstrakter Spezifikationsteile
- Gleiche abstrakte Teile zweier Services aber unterschiedliche konkrete Ausprägungen

```
<? xml version="1.0" encoding ="ISO-8859-1" ?>
  <definitions>
    <types>
      <schema xmlns := "http://www.w3.org /2001/XMLSchema ">
        <complexType abstract ="true" name="TNetlocation">
          <attributeref=" wsdl:required" use="optional"/>
          <attribute location ="anyURI"/>
        </complexType>
        <element name="Netlocation" type="TNetlocation" />
      </schema>
    </types>
    [...]
    <service>
      <Netlocation location =" http://192.168.1.5/Netze /EVV.pnml"/>
      <port>[...] </port>
    </service>
  </definitions>
```

- Automatisierung?
- Offenlegung der Geschäftslogik?
- Anbieten mehrerer Beschreibungen unterschiedlicher Abstraktionsniveaus?
- Werkzeugunterstützung?

1-a.1	Codeanalyse	manuell
1-a.2	Netzerzeugung aus den Analysedaten	manuell
1-b.1	Serviceprogrammierung	werkzeuggestützt
1-b.2	Erstellung einer Netzrepräsentation der Services	manuell
2.	Integration der Netze in die Schnittstellen	manuell
3.1	Komposition durch Erstellung des Prozesses	werkzeuggestützt
3.2	Erstellung der Netzrepräsentation des Prozesses	manuell (Werkzeugprüfung steht aus)
4.	Generierung des Gesamtnetzes	manuell
5.	Analyse des Gesamtnetzes	automatisch

- Modellierung der Netze – Petri Net Kernel (PNK) Version 2.2
(<http://www2.informatik.hu-berlin.de/top/pnk/>)
- BPEL Prozessmodellierung – Oracle BPEL Process Manager
- Validierung von BPEL Prozessen www.activebpel.org
- BPEL2PNML entwickelt an der Queensland University of Technology
(<http://www.bpm.fit.qut.edu.au/projects/babel/tools/>)
- Analyse des Gesamtnetzes – INA (Integrated Net Analyzer)
 - Transformation der PNML Daten in INA Format notwendig
 - Erreichbarkeitsgraph des Beispiels 1299 Elemente
 - 4.174 Sekunden (Pentium III mit 700 Mhz und 512 MB RAM)(<http://www2.informatik.hu-berlin.de/~starke/ina.html>)

- kein technisches Problem
- Mögliche Offenlegung von beteiligten Organisationseinheiten, Personen, Werkzeugen
- Gegenläufig zum Gedanken der Kapselung durch definierte Schnittstellen
- Lösungsansätze
 - Vertrauensbasierte Modelle
 - Netztransformation oder Verschleierung

- „Eine weitere Methode um zu kleineren Erreichbarkeitsgraphen zu kommen, besteht in der schrittweisen Reduktion des Netzes. Ein Reduktionsschritt besteht dabei im Ersetzen eines Unternetzes durch ein anderes Unternetz (im allgemeinen mit weniger Plätzen) in einer solchen Weise, dass die interessierenden Eigenschaften des Netzes invariant bleiben.“

(Starke, Peter H.: *Analyse von Petri-Netz-Modellen*. B.G. Teubner, 1990 S. 83)

- lokale Verklemmungsfreiheit vorausgesetzt genügt eine schnittstellenerhaltende Reduktion
- ***schnittstellenerhaltende Transformation***
Ein Transformationsschritt, zum Beispiel eine Reduktion, ist schnittstellenbewahrend genau dann, wenn keine Eingangsstellen und keine Ausgangsstellen entfernt werden. Für ein zu streichendes x , $x \in P$ muss also gelten, dass $\bullet x \neq \emptyset$ und $x \bullet \neq \emptyset$.

- Ausgangspunkt: Kompositionszeitpunkt
- Welche Probleme ergeben sich?
- Sind die vorhandenen Ansätze ausreichend?
- Petri Netz Modellierung
- Integration in die WSDL Schnittstelle
- Durchführung an einem Beispiel
- Untersuchung der vorhandenen Werkzeugunterstützung
- Ausräumen der Offenlegungsproblematik

Vielen Dank
für
Ihre Aufmerksamkeit