SKIL
Studentenkonferenz
Informatik Leipzig
25. September 2012

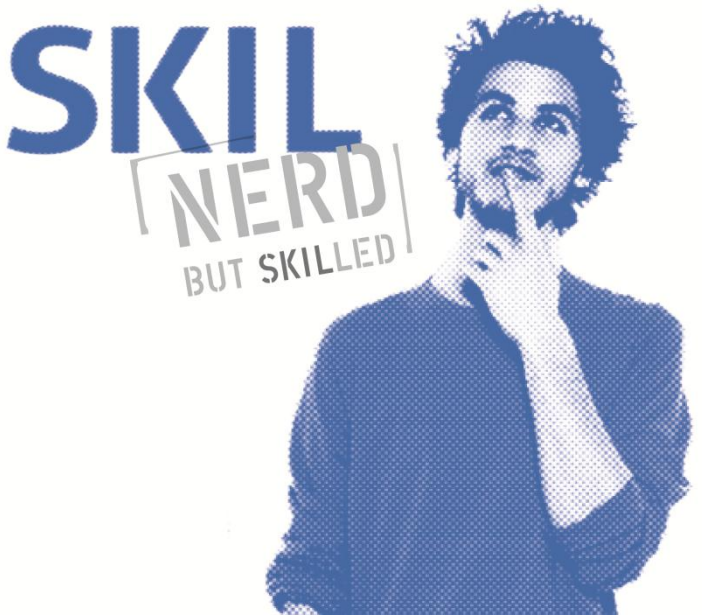
Generierung von Serviceverträgen auf Basis objektorientierter ereignisgesteuerter Prozessketten

Jörg Hartmann
Universität Leipzig
jhartmann@informatik.uni-leipzig.de

Agenda

- Grundlagen
- Problemstellung und Zielsetzung
- Vorstellung der Methode
- Anwendung der Methode
- Zusammenfassung und Ausblick

Grundlagen



Grundlagen – Serviceorientierte Architektur

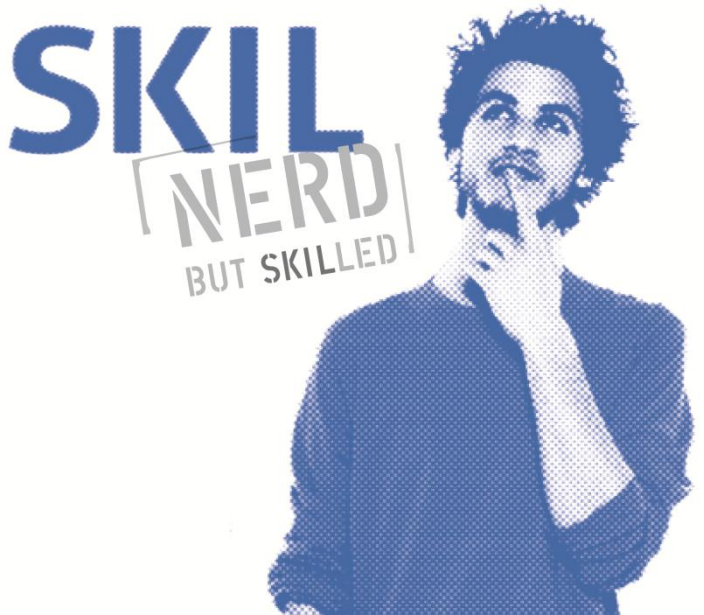
- Serviceorientierte Architektur (SOA)
  - Unternehmensweite IT-Architektur
  - Komposition loser gekoppelter Services
- Service
  - Realisierung von Geschäftsfunktionen
  - „Komponente mit wohldefinierter und wiederverwendbarer Funktionalität“ [1]
    - Spezifiziert durch Servicevertrag
    - Implementierung erfüllt Servicevertrag
- Servicevertrag
  - Vollständige Spezifikation eines Service [2]
    - Spezifikation (technischer Servicevertrag)
    - Service Level Agreement

Grundlagen – Sichten auf einen Servicevertrag

- Dekomposition des Informationsgehaltes

Modellsicht	Beschreibung	Notation
Kompositionssicht	Zeitlich-logische Abfolge der Systemfunktionen	oEPK
Datensicht	Aufbau und Struktur der Daten	UML-Klassendiagramm
Funktionssicht	Verknüpfung der Systemfunktionen mit Daten	eEPK, BPMN
Applikationssicht	Endpoints und benötigte IT-Systeme	UML-Deploymentdiagramm
Policy-Sicht	Nicht-funktionale Eigenschaften	WS-Policy-Editor

Problemstellung und Zielsetzung



Problemstellung

- Frage nach einer systematischen Identifikation
  - Beginn des Servicelebenszyklus
  - Grundlage der Spezifikation und Implementierung
  - „eine der herausforderndsten Aufgaben der Umsetzung einer SOA“ [1]

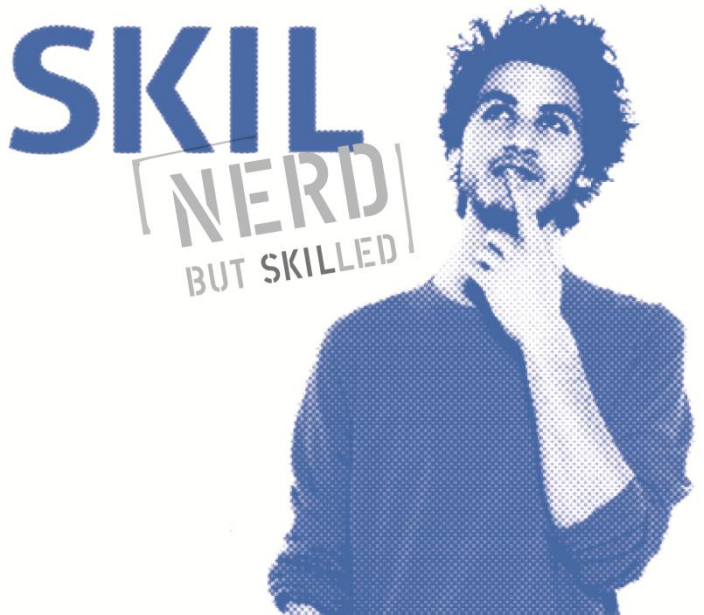
- Schwierigkeiten
  - Fehlende standardisierte Vorgehensweise
  - Keine Berücksichtigung von Anforderungen
  - Inkonsistenzen in den Geschäftsprozessmodellen
  - Konzentration auf funktionale Eigenschaften

Zielsetzung

- Weiterentwicklung bestehender Identifikationsansätze
  - Formalisierte, detaillierte und (möglichst) automatisierte Vorgehensweise
  - Ausrichtung an Anforderungen
  - Konsistente Prozessmodellierung
  - Berücksichtigung nicht-funktionaler Anforderungen

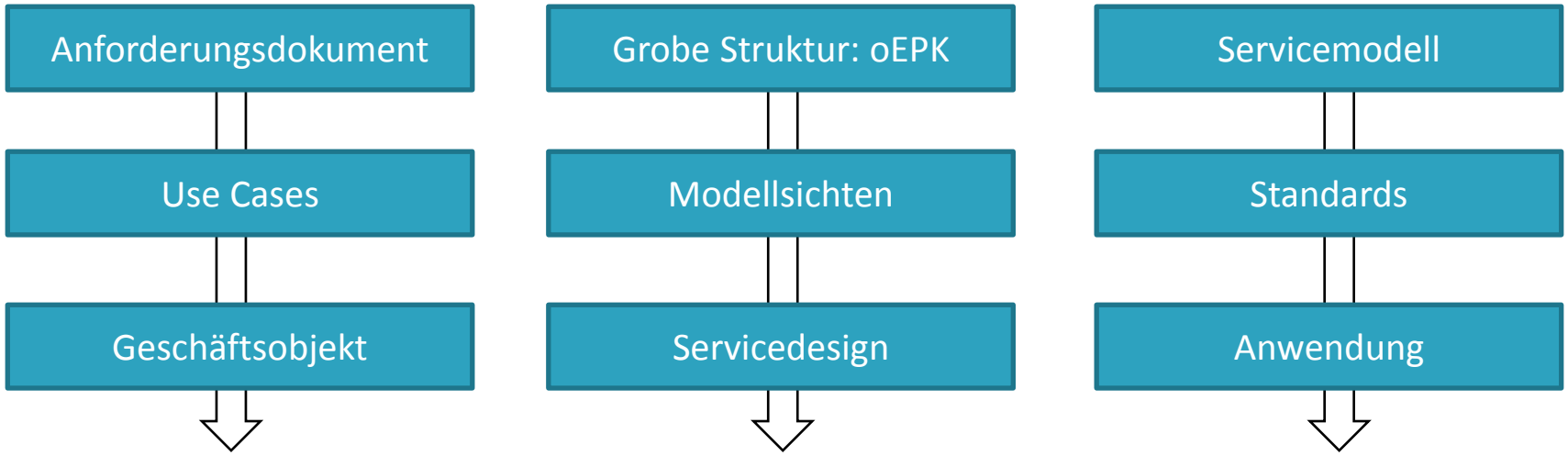
- Lösung – Methode zur Modellierung technischer Serviceverträge

Vorstellung der Methode



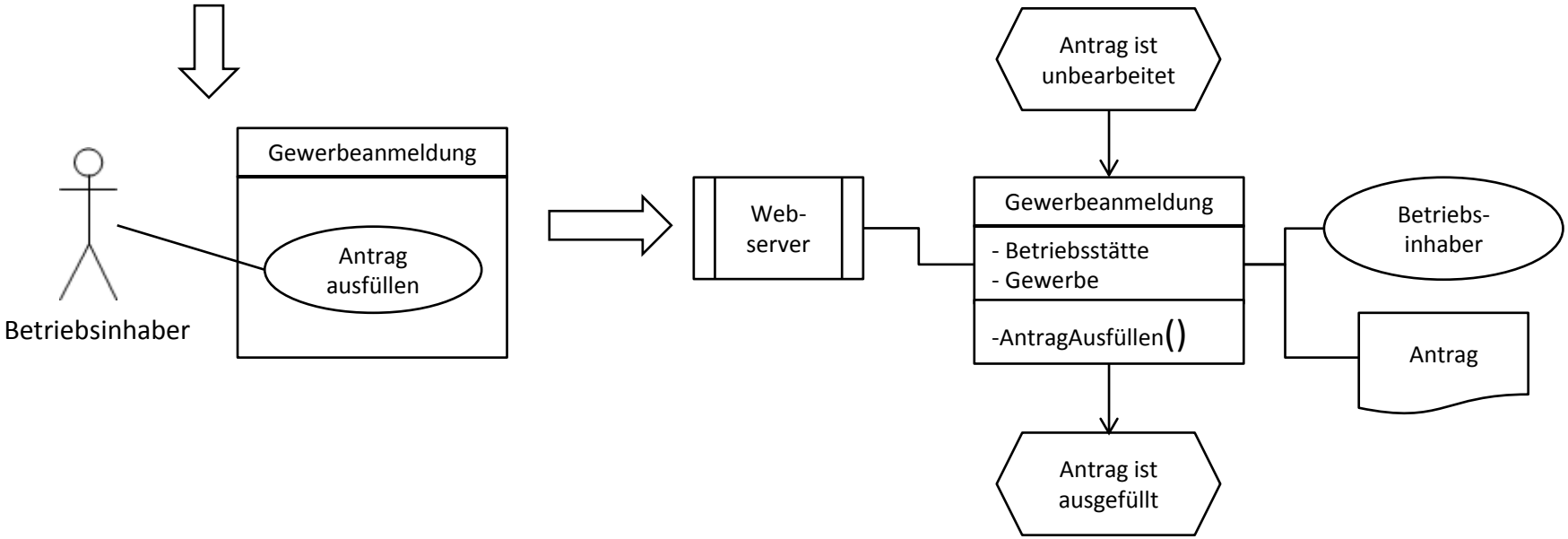
Vorstellung der Methode – Architektur

- Anforderungsebene: Transformation fachlicher Anforderungen zu Geschäftsobjekten
- Designebene: Verfeinerung der Geschäftsobjekte in Modellsichten
- Serviceebene: Generierung technischer Serviceverträge

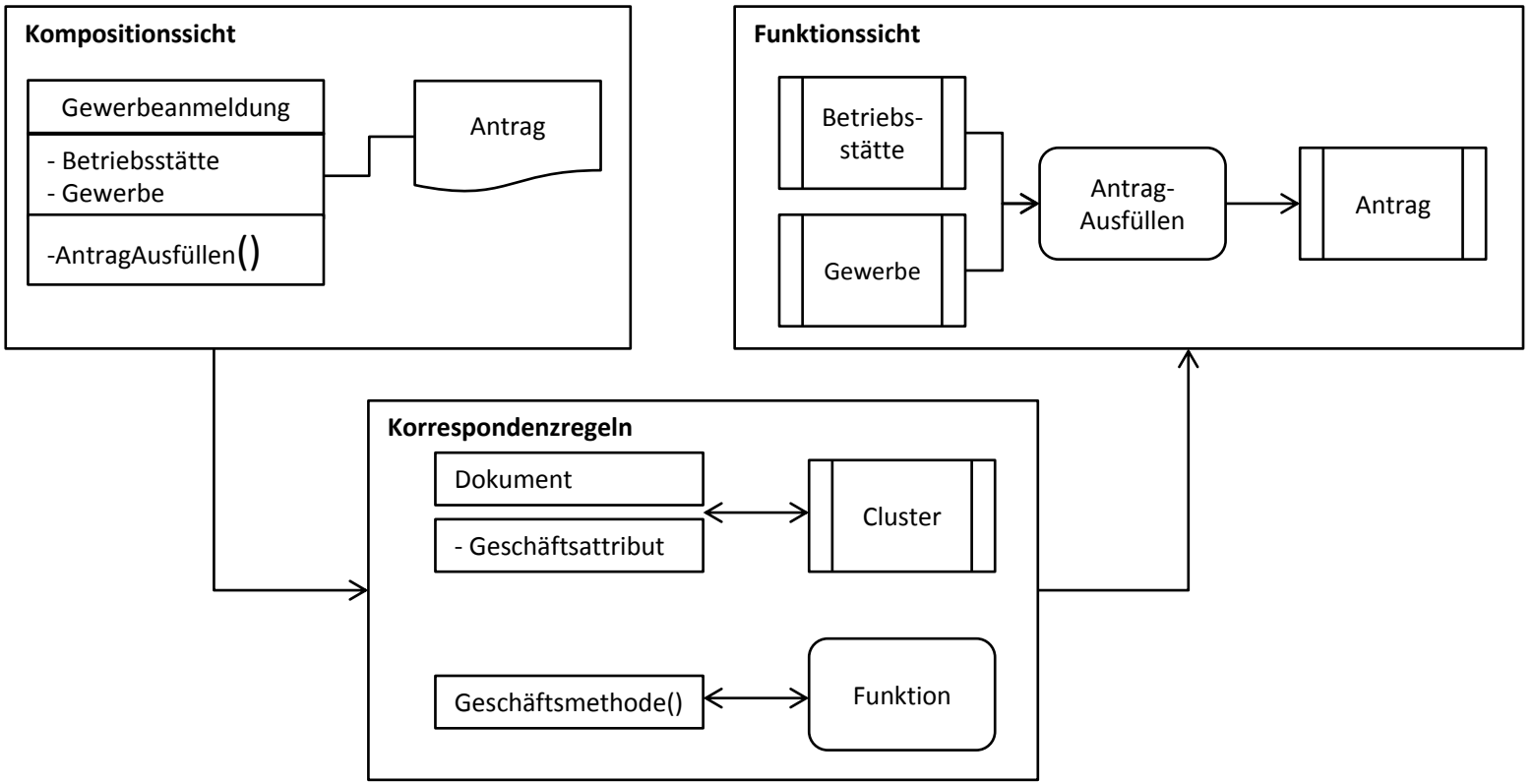


Anwendung der Methode – Anwendungsfall

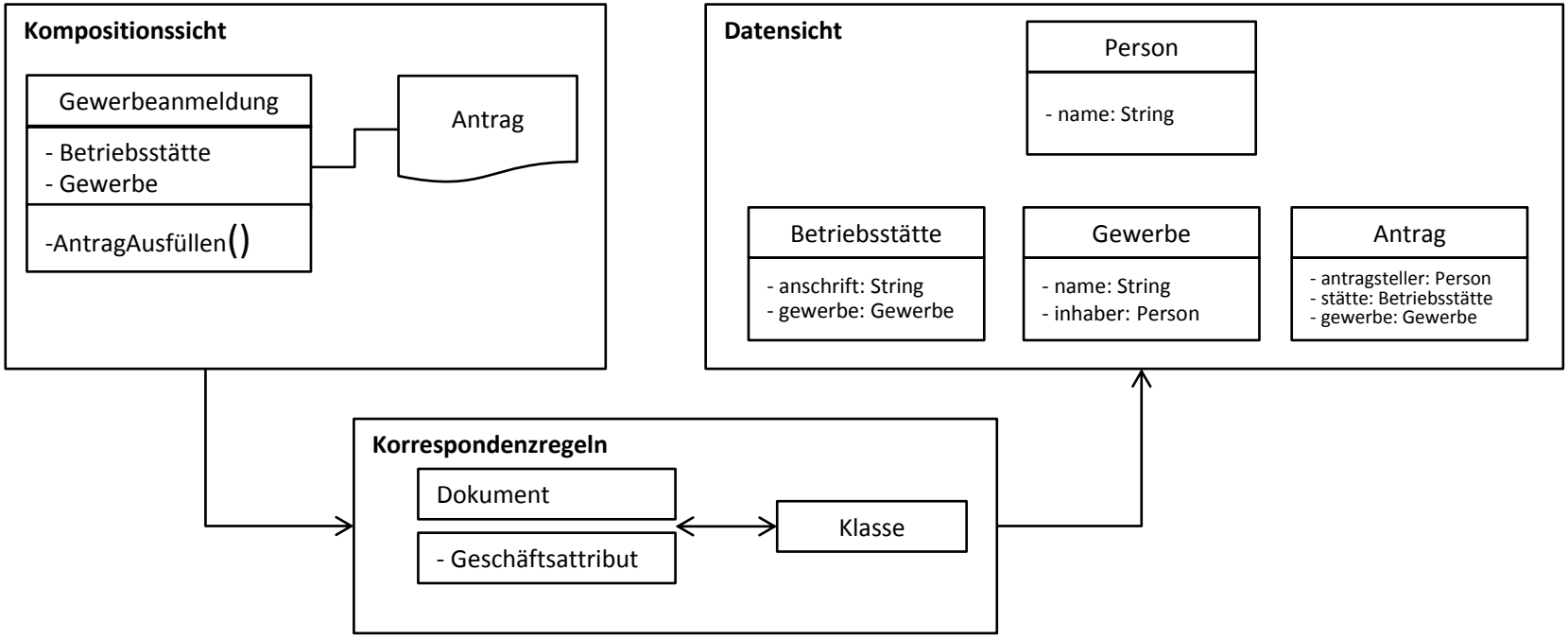
- Anforderungen - Anmeldung eines Gewerbes
  - Das System erlaubt einem Betriebsinhaber das Ausfüllen eines Antrages.
    - Ein Antrag benötigt Informationen zur Betriebsstätte und einem Gewerbe.
  - Das System ist über einen Webserver erreichbar.
  - Das System ist stets verfügbar.



Anwendung der Methode – Verfeinerung der Funktionssicht



Anwendung der Methode – Verfeinerung der Datensicht



Anwendung der Methode – Generierung

```
<!--  type declaration Betriebsstaette -->
<xsd:element name="Betriebsstaette" type="tns:BetriebsstaetteType" />
<xsd:complexType name="BetriebsstaetteType">
      <xsd:sequence>
            <xsd:element name="Anschrift" type="tns:AnschriftType“
                minOccurs="1" maxOccurs="1">
            </xsd:element>
            <xsd:element name="Inhaber" type="tns:PersonType“
                minOccurs="1" maxOccurs="1">
            </xsd:element>
       </xsd:sequence>
</xsd:complexType>
```

Anwendung der Methode – Generierung

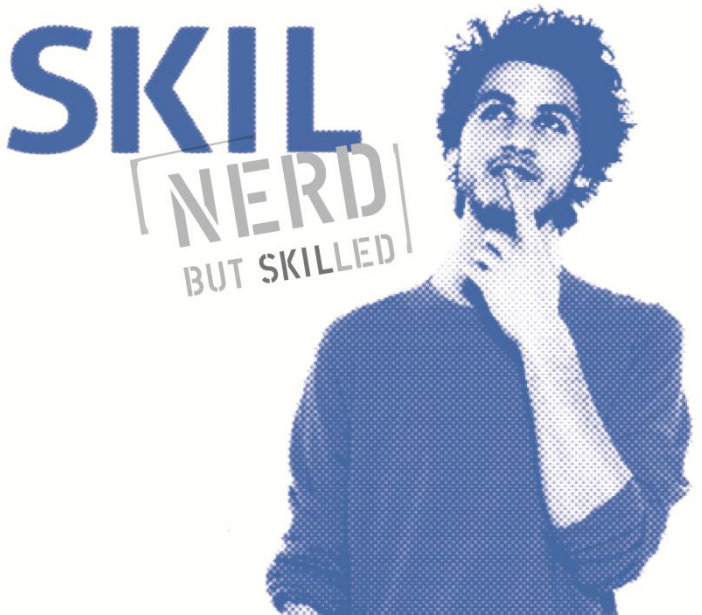
```
<!-- message declaration AntragAusfuellen -->
<wsdl:message name="AntragAusfuellenRequest">
    <wsdl:part name="Betriebsstaette" element="types:Betriebsstaette" />
    <wsdl:part name="Gewerbe" element="types:Gewerbe" />
</wsdl:message>
<wsdl:message name="AntragAusfuellenResponse">
    <wsdl:part name="Antrag" element="types:Antrag" />
</wsdl:message>
```

Anwendung der Methode – Generierung

```
<!-- interface Gewerbeanmeldung -->
<wsdl:portType name="Gewerbeanmeldung">
        <!-- policy declaration -->
        <wsp:PolicyReference URI="#SecurityPolicy" />
        <!-- exposed function AntragAusfuellen -->
        <wsdl:operation name="AntragAusfuellen">
                <wsdl:input message="tns:AntragAusfuellenRequest" />
                <wsdl:output message="tns:AntragAusfuellenResponse" />
        </wsdl:operation>
</wsdl:portType>

<!-- service Gewerbeanmeldung -->
<wsdl:service name="Gewerbeanmeldung">
        <wsdl:port binding="tns:gewerbeanmeldungSOAP" name="Gewerbeanmeldung">
                <soap:address location="localhost:8080" />
        </wsdl:port>
</wsdl:service>
```

Zusammenfassung und Ausblick



Zusammenfassung und Ausblick

- Zusammenfassung
  - Problemstellung um Serviceidentifikationsmethoden
  - Weiterentwicklung bisheriger Ansätze
  - Anforderungsspezifikation/Modellierung/Generierung
  - Nachverfolgbarkeit und Reflektion

- Ausblick
  - Vervollständigung: Versionierung
  - Erweiterung: Implementierung der Services
  - Prototypische Entwicklung

Literatur

[1] Thomas Müller. Der Weg zum guten Service. In Gernot Starke und Stefan Tilkov, Hrsg., SOA-Expertenwissen: Methoden, Konzepte und Praxis serviceorientierter Architekturen, Seiten 141–159. dpunkt.verlag, 2007.

[2] Erl, Thomas: Web service contract design and versioning for SOA. Upper Saddle River, NJ : Prentice Hall, 2009.

[3] August-Wilhelm Scheer, Markus Nüttgens und Volker Zimmermann. Objektorientierte Ereignisgesteuerte Prozesskette (oEPK) – Methode und Anwendung. Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 141, 1997

[4] THESEUS.TEXO – Infrastruktur für internetbasierte Dienste, http://theseus-programm.de/de/texo.php, 2012. letzter Zugriff am 24.09.2012.