

Studienprojekt SIMPL

---

# Anforderungskatalog

Version 0.2

12. Juli 2009

Verfasser: Michael Hahn

---



# Inhaltsverzeichnis

<b>1</b>	<b>Nichtfunktionale Anforderungen</b>	<b>3</b>
1.1	Dokumente . . . . .	3
1.2	Benutzer . . . . .	3
<b>2</b>	<b>Funktionale Anforderungen</b>	<b>3</b>
2.1	BPEL . . . . .	3
2.1.1	BPEL Aktivitäten . . . . .	4
2.1.2	BPEL Monitoring . . . . .	4
2.2	ODE . . . . .	4
2.3	Eclipse BPEL Designer . . . . .	4
2.4	Anbindung von Datenquellen . . . . .	4
2.4.1	Datenbanken . . . . .	5
2.4.2	Dateien . . . . .	5
2.5	Autorisierung und Authentifizierung . . . . .	5
2.6	Monitoring . . . . .	6
2.7	Demo-Programm . . . . .	6
<b>3</b>	<b>Organisatorisches</b>	<b>6</b>
<b>4</b>	<b>Änderungsgeschichte</b>	<b>7</b>

# 1 Nichtfunktionale Anforderungen

1. Die Schnittstellen des Rahmenwerks sollen leicht erweiterbar sein und insgesamt soll ein modularer Aufbau zugrunde liegen.
2. Erweiterbarkeit ist ein wichtiges Qualitätsmerkmal.
3. Fehlertoleranz: Scopes mit Handler. CompensationHandler und FaultHandler implementieren Logik für die Kompensation von Fehlern (rollback).
4. Ausfallsicherheit soll so gut wie möglich sein.
5. Usability sollte sich an Zielgruppe orientieren und dazu die größtmögliche Transparenz liefern.
6. Es gibt keine bestimmte Infrastruktur in der das System später läuft. ODE läuft überall lokal und die Datenbanken sind verteilt.
7. Skalierbarkeit: keine bestimmte Infrastruktur => von schlechten PC's bis zu Supercomputern ist alles möglich.

## 1.1 Dokumente

1. Am Ende des Projekts soll es eine zentrale Spezifikation und auch Dokumentation für/mit alle Erweiterungen geben.

## 1.2 Benutzer

1. Benutzer haben keine bis wenig Vorkenntnisse (keine Informatiker).
2. Die Zielgruppe sind Wissenschaftler und Ingenieure.

# 2 Funktionale Anforderungen

1. Das Rahmenwerk soll als Eclipse Plug-In verwendet werden.
2. Verarbeitung von großen heterogenen Datenmengen im Rahmen eines wissenschaftlichen Workflows.

## 2.1 BPEL

1. Fehlerbehandlung und Kompensation im Workflow.
2. Unterstützung von Referenzen in BPEL-Prozessen
  - (a) Referenzen (z.B. auf eine Tabelle) im SQL Statement via BPEL Variable
  - (b) Implementierung eines Reference Resolution System in einer Testumgebung
    - i. Referenz/ Pointer im BPEL Prozess auflösen
    - ii. Bei Daten, die nicht für die Prozesslogik, soll die Referenz / der Pointer vom Service aufgelöst werden (zur Entlastung der Engine)
3. Momentan kein BPEL Einsatz in Scientific Workflows, Realisierung über Web-Services. Kepler und Andere definieren eigene Sprachen für Scientific Workflows.

### **2.1.1 BPEL Aktivitäten**

1. Alle generischen BPEL-Aktivitäten, die für den Umgang mit Datenquellen benötigt werden, müssen erstellt werden. Welche Datenmanagement-Patterns dabei abgedeckt werden müssen ist aus [1] zu entnehmen.
2. Auch das Anlegen, Löschen und Ändern von DB-Tabellen soll möglich sein, siehe hierzu auch [1].
3. Um neue BPEL-Aktivitäten zu erstellen, werden die BPEL Extension Activities genutzt.

### **2.1.2 BPEL Monitoring**

1. BPEL hat schon ein Monitoring, das für unsere Zwecke um die von uns neu erstellte Funktionalität erweitert werden muss und die Möglichkeit besitzt eine frei wählbare Monitoring-DB anzugeben.

## **2.2 ODE**

1. ODE hat bereits eine eigene Monitoring-DB, doch es soll möglich sein eine frei wählbare Monitoring-DB anzugeben.
2. Das bereits vorhandene Monitoring muss, um die von uns neu erstellten Funktionen, erweitert werden.

## **2.3 Eclipse BPEL Designer**

1. Syntax und Plausibilitätsprüfung für unsere Erweiterungen (z.B. für neue BPEL-Aktivitäten).
2. Test von Queries als Feature (falls zeitlich machbar).

## **2.4 Anbindung von Datenquellen**

1. Unterstützung von Transaktionen.
2. Late-binding bei Datenquellen: Kriterien zur Beschreibung von Datenquellen – Annotation von Datenquellen, Modellierung der Anforderungen an eine Datenquelle (durch den User), Strategien zur Auswahl einer geeigneten Datenbank.
3. Anforderungen an eine Datenquelle sollen, falls die Datenquelle über einen Web-Service eingebunden ist, mit Annotation mit/über “QUALITY OF SERVICE” angegeben werden, sonst statische Auswahl einer Datenquelle zur Modellierungs- oder Laufzeit.
4. Ein Nutzer soll angeben können, was die Datenquelle können muss und das Rahmenwerk wählt eine passende Datenquelle aus.
5. Spezifizierung der Anforderungen, die sinnvoll an eine Datenquelle gestellt werden können sollten, wird über die Definition eines Anforderungskataloges, der nach Fertigstellung mit dem Kunden abgesprochen werden muss, realisiert.
6. Ebenso sollen auch Strategien für die Auswahl von Datenquellen anhand definierter Anforderungen entwickelt und mit dem Kunden abgesprochen werden.
7. Die Strategien, die zur Auswahl einer Datenquelle anhand der vom Benutzer angegebenen Anforderungen verwendet werden, müssen selbst definiert und anschließend mit dem Kunden abgesprochen werden.
8. Die Auswertung der Annotationen und die Verarbeitung der darin enthaltenen Informationen soll ebenfalls selbst definiert und anschließend mit dem Kunden abgesprochen werden.

### 2.4.1 Datenbanken

1. Adressierung der Datenbanken: Statisch über konkrete Adresse oder dynamisch über logische Adresse.
2. Anbindung generisch, so viele Anbindungen wie möglich realisieren. Als Minimum gilt: IBM DB2 + OpenSourceDB's (mind. 1\*RDB + mind. 1\*XMLDB) + mind. 1\*SensorDB (z.B. TinyDB).
3. Es sollen auch Zugriffe auf mehrere Datenquellen innerhalb eines Prozess möglich sein, dazu besitzt jede Aktivität seine eigene Datenquellen-Referenz.
4. Alle Anfragesprachen für den Datenbankzugriff müssen unterstützt werden, mindestens SQL und XQuery.
5. Transaktionen sollen innerhalb von DB und BPEL-Prozessen unterstützt werden. Konzepte frei wählbar.
6. Es soll möglich sein, dass Daten aus Datenbanken aus einem Prozess auch in lokale Dateien exportiert werden können. Dabei gilt: XML als Standardformat (siehe IBM Ansatz: RDB-Tabellen <->XML).
7. Es müssen nicht zwangsweise alle Operationen auf Daten atomar sein, da z.B. Web-Services auch als atomar von Aussen angesehen werden. Als weitere Informationen hierzu dient auch das Thema *Atomic Scopes*.
8. Es gibt öffentliche wissenschaftliche Datenbanken, die für Testzwecke verwendet werden können: RDB Biowissenschaften (SwissProt, UniProt, blast). Weitere Informationen dazu gibt es bei Michael Reiter.
9. Datenbanken existieren bereits und müssen nicht durch das Rahmenwerk erstellt werden. Weiterhin sollen aber Schema-Definitionen möglich sein und das Erstellen, Ändern und Löschen von Tabellen innerhalb der Schemas.
10. Die Zugriffsrechte werden innerhalb der Datenbanken gesetzt und sind bereits vorhanden.

### 2.4.2 Dateien

1. Alle möglichen Dateiformate müssen unterstützt werden, wenn Einschränkungen stattfinden, müssen diese begründet werden.

## 2.5 Autorisierung und Authentifizierung

1. Autorisierung und Authentifizierung soll momentan nur für Datenquellen bereitgestellt werden, allerdings soll eine spätere Erweiterung einfach realisiert werden können.
2. Für die Authentifizierung sind verschiedene Verfahren gewünscht, sodass in jeder Situation das Bestmögliche verwendet wird.
3. Informationen über Authentifizierungsverfahren für wissenschaftliche Datenbanken gibt es bei Michael Reiter und als praktisches Beispiel das "Kepler-Projekt".
4. Die Autorisierung und die Authentifizierung sollen, über einen Single-Sign-On entweder im Prozess oder in der Instanz (unsere Entscheidung), beim Zugriff auf Datenquellen durchgeführt werden.
5. Autorisierung und Authentifizierung sollen auf die Ausführung von Prozessen beschränkt sein.

6. Autorisierungs- und Authentifizierungsparameter für einen Prozess sollen auf Datenquellenebene spezifiziert werden, d.h. bei Datenbanken über Schema, bei Dateisystemen z.B. über extra Dateien und bei Sensornetzen über spezielle Datenbanken (TinyDB?).
7. Autorisierung und Authentifizierung ist bei allen Datenquellen notwendig und soll auf verschiedene Arten möglich sein.
8. **Angaben über Autorisierung und Authentifizierung sollen extra abgefragt und als Nachricht an die DB geschickt werden.**

## 2.6 Monitoring

1. Monitoring der Prozessausführung: ExecutingHistory in einer angebundenen Datenbank speichern (nicht im lokalen Speicher der Engine).
2. Die Datenbank für das Monitoring muss über das Rahmenwerk frei wählbar sein und es soll nicht über mehrere Datenbanken verteilt sein.
3. Für unsere Zwecke muss ein komplettes Monitoring erstellt werden, d.h. ein Monitoring für BPEL, ODE und alle Datenquellenaktionen.
4. Das vorhandene Monitoring von BPEL und Apache ODE muss um das Monitoring unserer Funktionalitäten und die Möglichkeit eine variable DB als Monitoring-DB anzugeben erweitert werden.
5. Es soll ein Modell für das Monitoring (Informationen) erstellt und mit den Kunden abgesprochen werden.
6. Die Speicherdauer der Monitoring-Daten soll variabel als Parameter übergeben werden können.
7. Das Monitoring soll standardmäßig aktiv sein, aber auch auf Benutzerwunsch abschaltbar. Dies soll auch über den Eclipse BPEL-Designer mit Deployment-Deskriptoren definierbar sein.

## 2.7 Demo-Programm

1. Die Inhalte der Demo werden später geklärt. Es sollen darin die wichtigsten Use-Cases vorgeführt und ein kurzer Einblick in die Dokumentation gegeben werden.
2. In der Demo soll der Zugriff auf die IBM DB2 Datenbank, eine OpenSource RDB und eine OpenSource XMLDB, sowie evtl. auf Sensornetzdaten mittels TinyDB (falls öffentlich zugänglich) gezeigt werden. Ebenso soll eine kurze Vorstellung der Monitoring-DB-Anbindung mit einer der oben genannten Datenbanken gezeigt werden.

## 3 Organisatorisches

1. Vom Kunden erwartet:
  - (a) Angebot
    - i. mit Zeitplan
  - (b) Spezifikation
    - i. mit Prototyp
  - (c) Entwurf

- (d) Implementierung
  - i. mit Handbuch und Demo
- 2. Budget orientiert sich an der 1.Version des Angebots.
- 3. Verwertungsrechte sind OpenSource.
- 4. Regelmäßige Treffen zwischen Projektleiter und Betreuer/Kunde zur Fortschrittskontrolle im Projekt.
- 5. Arbeitsort des Projekts ist die Universität Stuttgart.
- 6. Liefertermin: April 2010, danach Abnahmetest und evtl. noch kleinere Änderungen/Anpassungen.
- 7. Urlaubsplan der Projektgruppe muss den Kunden geschickt werden.
- 8. Die Kunden stehen während des Projekts für Reviews oder ähnliches nach Terminabsprache zur Verfügung.

## 4 Änderungsgeschichte

- **Version 0.1**, 10. Juli 2009: Erstellung des Dokuments.
- **Version 0.2**, 11. Juli 2009: Ergebnisse der beiden Kundengespräche und Anforderungen aus den Kick-Off-Folien hinzugefügt.

## Literatur

- [1] Vrhovnik, M.; Schwarz, H.; Radeschütz, S.; Mitschang, B.: An Overview of SQL Support in Workflow Products. In: Proc. of the 24th International Conference on Data Engineering (ICDE 2008), Cancún, México, April 7-12, 2008
- [2] 1.Kundengespräch, Kundengespräch\_1.pdf, 10. Juli 2009
- [3] 2.Kundengespräch, Kundengespräch\_2.pdf, 10. Juli 2009
- [4] SIMPL Kick-Off Meeting Folien, SIMPL\_KickOff.pdf, 13. Mai 2009