

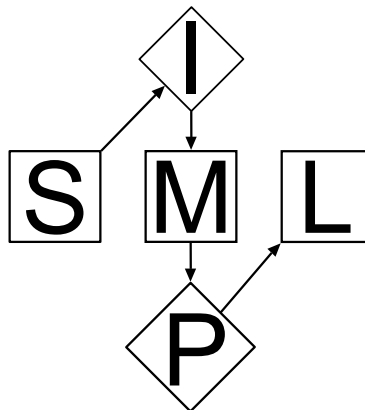
Spezifikation

Version 0.1

11. September 2009

Verfasser:

Wolfgang Huettig, Michael Hahn, Firas Zoabi, Michael Schneidt, Daniel
Brüderle, René Rehn



Dok-Status: neu

QS-Status: nicht QS-geprüft

Prüf-Status: nicht geprüft

Review-Status: kein Review durchgeführt

End-Status: -

Inhaltsverzeichnis

1	Einleitung	5
1.1	Zweck des Dokuments	5
1.2	Einsatzbereich und Ziele	5
1.3	Definitionen	5
1.4	Überblick	5
2	Allgemeine Beschreibung	6
2.1	Einbettung	6
2.2	Funktionen	6
2.3	Sprache	8
2.4	Distributionsform und Installation	9
2.5	Benutzerprofile	9
2.6	Einschränkungen	9
2.7	Annahmen und Abhängigkeiten	9
3	Nichtfunktionale Anforderungen	9
3.1	Mengengerüst	9
3.2	Benutzbarkeit	9
3.3	Robustheit	9
3.4	Sicherheit	9
3.5	Portabilität	9
3.6	Erweiterbarkeit	9
3.7	Wartbarkeit	9
3.8	Skalierbarkeit	9
3.9	10
4	Akteure	10
4.1	Prozess-Modellierer	10
4.2	Workflow-Administrator	10
4.3	Datenquellen-Administrator	10
4.4	ODE Workflow-Engine	10
4.5	Eclipse BPEL Designer	10
4.6	SIMPL Core	10
5	Anwendungsfälle (Use-Cases)	10
5.1	Diagramm aller Anwendungsfälle	10
5.2	Anwendungsfälle der Prozess-Modellierer	12
5.2.1	Data-Management-Aktivität erstellen	12
5.2.2	Data-Management-Aktivität bearbeiten	13
5.2.3	Data-Management-Aktivität löschen	13
5.2.4	Strategie auswählen	13
5.2.5	Datenquellen-Eigenschaften festlegen	13
5.2.6	Datenquelle auswählen	13
5.2.7	Annotation erstellen	14
5.2.8	Annotation bearbeiten	14
5.2.9	Annotation löschen	14
5.2.10	Annotationen speichern	14
5.2.11	Annotationen laden	14
5.2.12	Prozess ausführen	15
5.3	Anwendungsfälle der Workflow-Administratoren	15
5.3.1	Admin-Konsole öffnen	16

5.3.2	Monitoring aktivieren	16
5.3.3	Monitoring deaktivieren	16
5.3.4	Monitoring-Granularität ändern	16
5.3.5	Auditing aktivieren	16
5.3.6	Auditing deaktivieren	17
5.3.7	Auditing-Granularität ändern	17
5.3.8	Auditing-Datenbank festlegen	17
5.3.9	Neue Referenz in RRS einfügen	17
5.3.10	Referenz aus RRS bearbeiten	17
5.3.11	Referenz aus RRS löschen	18
5.4	Anwendungsfälle der Datenquellen-Administratoren	18
5.4.1	Datenquelle registrieren	18
5.4.2	Datenquelle entfernen	18
5.4.3	Datenquelle bearbeiten	19
5.5	Anwendungsfälle der ODE Workflow-Engine	19
5.5.1	Data-Management-Aktivität ausführen	19
5.5.2	Data-Management-Aktivität kompensieren	19
5.5.3	Data-Management-Aktivität rückgängig machen (rollback)	20
5.6	Anwendungsfälle des Eclipse BPEL Designers	20
5.6.1	Datenquellen abrufen	20
5.6.2	Annotation auswerten	20
5.6.3	Datenquellenliste filtern	21
5.6.4	Datenquelle per Strategie auswählen	21
5.7	Anwendungsfälle des SIMPL Core	21
5.7.1	BPEL-Datei transformieren	21
5.7.2	Neues Datenquellen-Plug-In in RRS einfügen	22
5.7.3	Referenz auflösen/dereferenzieren	22
6	Konzepte und Realisierungen	22
6.1	Reference Resolution System	22
6.2	BPEL-SQL	22
6.3	Eclipse BPEL Designer	22
6.4	Monitoring	22
6.5	22
7	Einzusetzende Technologien	22
7.1	Materialien	22
7.2	Werkzeuge	22
7.2.1	Entwicklungsumgebung	22
7.2.2	Sonstige Werkzeuge	22
8	Änderungsgeschichte	22

1 Einleitung

In diesem Abschnitt wird der Zweck dieses Dokuments sowie der Einsatzbereich und die Ziele der zu entwickelnden Software beschrieben. Weiterhin werden die, in diesem Dokument, verwendeten Definitionen erläutert und ein Überblick über das restliche Dokument gegeben.

1.1 Zweck des Dokuments

Diese Spezifikation ist die Grundlage für alle weiteren Dokumente, die im Rahmen dieses Projekts entstehen. In ihr sind sämtliche Anforderungen an die zu entwickelnde Software festgelegt. Sie muss stets mit den anderen Dokumenten, insbesondere mit dem Entwurf und der Implementierung, konsistent gehalten werden. Die Spezifikation dient den Team-Mitgliedern als Grundlage und Richtschnur für die Entwicklung der Software und den Kunden als Zwischenergebnis zur Kontrolle.

Zum Leserkreis dieser Spezifikation gehören:

- Die Entwickler der Software,
- die Kunden und
- die Gutachter der Spezifikationsreviews.

1.2 Einsatzbereich und Ziele

Das Entwicklungsteam soll ein erweiterbares, generisches Rahmenwerk für die Modellierung und Ausführung von Workflows erstellen, welches den Zugriff auf nahezu beliebige Datenquellen ermöglichen soll. Bei den Datenquellen kann es sich beispielsweise um Sensornetze, Datenbanken und Dateisysteme handeln. Der Schwerpunkt soll klar auf wissenschaftlichen Workflows beruhen. Über das Rahmenwerk sollen beliebige Datenmanagement-Funktionen in einen BPEL-Prozess eingebunden werden können. Dafür werden bereits vorhandene Konzepte evaluiert und falls nötig erweitert oder angepasst. Es wird untersucht, inwiefern die Sprache BPEL ebenfalls erweitert werden muss. Für eine möglichst hohe Flexibilität soll ein dynamischer Ansatz gewählt werden, so dass erst während der Laufzeit des Systems die Datenquellen festgelegt werden können. Nichtsdestotrotz sollte auch die Möglichkeit bestehen, die Datenquellen statisch anbinden zu können. Eine Anforderung des Kunden ist, dass eine vorhandene BPEL-Engine sowie ein vorhandenes Modellierungstool um diese gewünschten Funktionen erweitert bzw. angepasst werden. Die BPEL-Prozesse sollen mit dem entsprechenden Modellierungstool spezifiziert und mit der BPEL-Engine ausgeführt werden können.

1.3 Definitionen

Die in der Spezifikation verwendeten Begriffe, Definitionen und Abkürzungen werden in einem separaten Begriffslexikon definiert und eindeutig erklärt. Durch das Begriffslexikon werden somit alle technischen und fachlichen Begriffe, Definitionen und Abkürzungen, die zu Missverständnissen innerhalb des Projektteams oder zwischen Projektteam und Kunde führen könnten, eindeutig definiert.

1.4 Überblick

In diesem Dokument werden alle Anforderungen des Kunden festgehalten. Nach einem kurzen Überblick über das Projekt selbst, folgen sowohl nichtfunktionale als auch funktionale Anforderungen an das zu entwickelnde Produkt. Die Beschreibung der funktionalen Anforderung, also des Funktionsumfangs, geschieht durch Use-Cases, sowie durch die Beschreibung der späteren Benutzeroberfläche. Ebenso wichtig ist die Beschreibung der nichtfunktionalen Anforderungen, die eventuelle Abhängigkeiten, Einschränkungen, Größenordnungen und Bedingungen im Bezug auf Portabilität, Robustheit, Sicherheit und weitere Aspekte beschreiben.

2 Allgemeine Beschreibung

Dieser Abschnitt liefert allgemeine Informationen über die zu entwickelnde Software. Dazu gehören beispielsweise die Beschreibung der späteren Systemumgebung, die wichtigsten Funktionen, die verwendete Sprache und Informationen über den Benutzerkreis der Software.

2.1 Einbettung

Das SIMPL Rahmenwerk soll in die, in Abbildung 1 dargestellte, Systemumgebung eingebettet werden. Die Systemumgebung besteht dabei aus den unterschiedlichen Datenquellen, Eclipse mit dem BPEL-Designer Plug-In und einem Web-Server, wie z.B. dem Apache Tomcat, in dem das Rahmenwerk und eine Workflow-Engine (z.B. Apache ODE) ausgeführt werden. Dabei läuft die benötigte Software auf dem lokalen Rechner des Benutzers, die Datenquellen können auf verschiedene Server verteilt sein.

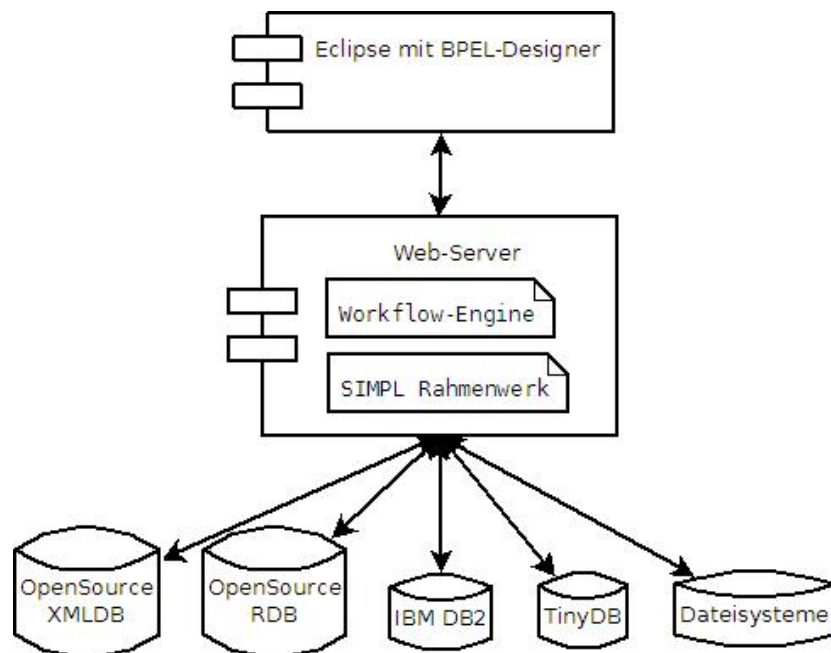


Abbildung 1: Übersicht über die Systemumgebung von SIMPL

2.2 Funktionen

In diesem Abschnitt folgen die wichtigsten Funktionen des Rahmenwerks, die später dessen Kernfunktionalität bilden sollen.

Das Rahmenwerk soll als Eclipse Plug-In verwendet werden und als Laufzeitumgebung integriert sein. Ebenso soll die Verarbeitung von großen, heterogenen Datenmengen im Rahmen eines wissenschaftlichen Workflows möglich sein.

Um die Fehlertoleranz des Rahmenwerks zu realisieren, sollen Fehler der neu definierten Aktivitäten erkannt und durch entsprechende FaultHandler abgefangen werden können. Abgefangene Fehler werden dann durch entsprechende CompensationHandler kompensiert oder die vorher ausgeführten Aktionen rückgängig gemacht.

BPEL

Über den Eclipse BPEL Designer sollen innerhalb von BPEL-Prozessen Referenzen unterstützt werden, d.h. es soll möglich sein Referenzen (z.B. auf eine Tabelle) innerhalb eines SQL-Befehls über eine BPEL Variable anzugeben oder auch die Ergebnisse eines Queries per Referenz im Prozess bereit zu halten. Dazu ist es erforderlich, dass ein Reference Resolution System mit verschiedenen Adaptern implementiert wird. Dieses System dient dazu Referenzen/Pointer innerhalb eines BPEL Prozesses aufzulösen und so z.B. auch Daten in den Prozess-Cache zu laden. Bei Daten, die nicht für die Prozesslogik relevant sind, soll dabei die Referenz/der Pointer vom Service, zur Entlastung der Workflow-Engine, aufgelöst werden. Mithilfe eines Transformers übersetzt der BPEL Designer den BPEL-Code mit Referenzen in Standard BPEL-Code, der dann auf der Workflow-Engine ausgeführt werden kann.

BPEL Aktivitäten

Alle BPEL-Aktivitäten, die für den Umgang mit Datenquellen benötigt werden, müssen erstellt werden und sollen erweiterbar sein. Als Anhaltspunkt welche Aktivitäten benötigt werden, gelten die folgenden Datenmanagement Patterns: Das *Query Pattern* beschreibt die Notwendigkeit mithilfe von SQL-Befehlen externe Daten anfordern zu können. Die aus den Queries resultierenden Daten können dabei auf der Datenquelle extern gespeichert oder direkt im Prozessspeicher gehalten werden. Das *Set UID Pattern* beschreibt die Möglichkeit mengenorientiertes Einfügen (insert), Aktualisieren (update) und Löschen (delete) auf externen Daten durchführen zu können. Das *Data Setup Pattern* liefert die Möglichkeit benötigte Data Definition Language (DDL) Befehle auf einem relationalen Datenbanksystem auszuführen, um so während der Prozessausführung die Datenquelle zu konfigurieren oder neue Container (Tabellen, Schema, usw.) zu erstellen. Da die Verarbeitung von komplexen Daten meist durch Stored Procedures erfolgt, ist es bei der Verarbeitung von externen Daten unbedingt erforderlich, Stored Procedures auch aus einem Prozess heraus aufrufen zu können (*Stored Procedure Pattern*). Manchmal ist es nötig Daten innerhalb des Prozessspeichers verarbeiten zu können. Das *Set Retrieval Pattern* liefert dafür eine mengenorientierte Datenstruktur, in die man die angefragten externen Daten innerhalb des Prozessspeichers ablegen kann. Diese Datenstruktur verhält sich dabei wie ein Cache im Prozessspeicher, der keine Verbindung zur originalen Datenquelle besitzt. Das *Set Access Pattern* beschreibt die Notwendigkeit auf den erzeugten Datencache sequentiell und direkt (random) zugreifen zu können. Das *Tuple IUD Pattern* beinhaltet einfügen (insert), aktualisieren (update) und löschen (delete) von Daten im Datencache. Das *Synchronization Pattern* realisiert die Synchronisation eines lokalen Datencaches mit der originalen Datenquelle.

Anbindung von Datenquellen

Die Anbindung von Datenquellen soll generisch sein und es sollen so viele Anbindungen wie möglich realisiert werden. Als Minimum gilt dabei die IBM DB2, mindestens zwei OpenSourceDB's (eine RDB und eine XMLDB), mindestens eine SensorDB (z.B. TinyDB) und mindestens ein Dateisystem (z.B. ext3). Es sollen auch Zugriffe auf mehrere Datenquellen innerhalb eines Prozesses möglich sein, dazu besitzt jede Aktivität eine Variable, in der der logische Namen der Datenquelle auf die zugegriffen werden soll, hinterlegt ist. Alle Anfragesprachen für den Datenbankzugriff müssen unterstützt werden, mindestens SQL und XQuery. Zusätzlich sollen auch alle SQL-Dialekte unterstützt werden. Transaktionen sollen innerhalb von DB und BPEL-Prozessen unterstützt werden. Die Konzepte sind nach Absprache mit dem Kunden frei wählbar. Es soll auch möglich sein, dass Daten aus Datenquellen aus einem Prozess auch in lokale Dateien exportiert werden können und umgekehrt soll es auch möglich sein lokale Daten zu importieren. Dabei gilt XML als Standardformat.

Die Datenbanken, die verwendet werden, 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.

Das Late-binding von Datenquellen soll unterstützt werden. Dazu müssen Kriterien zur Beschreibung von Datenquellen erstellt werden. Diese Kriterien werden dann als Anforderungen durch den Be-

nutzer modelliert und als Annotationen übertragen. Die 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. Die Annotationen dienen dazu, dass Datenquellen durch definierte Strategien automatisch ausgewählt werden können. Diese Strategien, die zur Auswahl einer Datenquelle anhand der vom Benutzer angegebenen Anforderungen verwendet werden, müssen von uns definiert und anschließend mit dem Kunden abgesprochen werden. Die Auswertung der Annotationen und die Verarbeitung der darin enthaltenen Informationen soll, ebenfalls selbst definiert und anschließend mit dem Kunden abgesprochen werden. Die Adressierung der Datenbanken soll statisch über konkrete Adressen oder auch dynamisch über logische Namen mithilfe der JNDI API möglich sein.

Admin-Konsole

Die Admin-Konsole ist ein Werkzeug mit dem verschiedene Einstellungen des Rahmenwerks auch während der Laufzeit noch geändert werden können. Dazu gehört beispielsweise das An- und Abschalten des Auditings und Einstellen der Granularität. Eine solche Komponente soll für die Administration des Rahmenwerks erstellt werden.

Autorisierung und Authentifizierung

Die Autorisierung und Authentifizierung soll momentan nur für Datenquellen bereitgestellt werden, allerdings soll eine spätere Erweiterung einfach realisiert werden können. Für die Authentifizierung und Autorisierung sind verschiedene Verfahren gewünscht, sodass in jeder Situation das Bestmögliche verwendet wird. Dazu soll es möglich sein, verschiedene konkrete Verfahren flexibel anbinden zu können. Die Autorisierung und die Authentifizierung soll dabei über einen Single-Sign-On in der Instanz beim Zugriff auf Datenquellen durchgeführt werden. Autorisierungs- und Authentifizierungsparameter für einen Prozess sollen auf Datenquellenebene spezifiziert werden, d.h. bei Datenbanken über Schema und bei Dateisystemen z.B. über extra Dateien. Die Autorisierung und Authentifizierung ist bei allen Datenquellen notwendig und soll auf verschiedene Arten möglich sein. Angaben über Autorisierung und Authentifizierung sollen extra im Eclipse BPEL Designer abgefragt und als Nachricht an die Datenquelle geschickt werden.

Auditing und Monitoring

Es soll ein Auditing der Prozessausführung geben, dafür soll die Ausführungshistorie in einer angebundenen Datenbank und nicht im lokalen Speicher der Engine gespeichert werden können. Die Datenbank für das Auditing soll über das Rahmenwerk frei wählbar sein. Die Auditing-Daten werden dabei auf einer Datenbank gespeichert und nicht über mehrere verteilt. Das vorhandene Auditing von Apache ODE muss dafür um das Auditing unserer Funktionalitäten und die Möglichkeit, eine variable DB als Auditing-DB anzugeben, erweitert werden. Die Speicherdauer der Auditing-Daten soll variabel als Parameter übergeben werden können. Das Auditing soll standardmäßig aktiv sein. Auf Benutzerwunsch soll die Granularität des Auditings veränderbar sein oder das Auditing auch komplett abschaltbar. Dies soll auch während der Laufzeit über die Admin-Konsole steuerbar sein.

Für unsere Zwecke muss ein komplettes Monitoring erstellt werden, d.h. ein Monitoring für alle Datenquellenaktionen und die Anpassung und Erweiterung des bestehenden Monitorings von Apache ODE. Dazu soll das Event-Modell von Apache ODE für das Monitoring, d.h. welche Informationen angezeigt werden können, für unsere Erweiterungen modifiziert und mit den Kunden abgesprochen werden.

2.3 Sprache

Generell gilt, dass alle Dokumente auf Deutsch und jeder Quellcode einschließlich Kommentaren auf Englisch verfasst und ausgeliefert werden. Eine Ausnahme bilden das Handbuch und die verschiedenen

Dokumentationen der durchzuführenden Erweiterungen, wie z.B. die Erweiterungen von Apache ODE oder dem Eclipse BPEL Designer. Diese Dokumente werden auf Deutsch und auf Englisch verfasst, um sie einem breiteren Leserkreis zur Verfügung stellen zu können.

2.4 Distributionsform und Installation

2.5 Benutzerprofile

Die Benutzer sind im Normalfall Wissenschaftler und Ingenieure. Sie haben meist keine bis wenig Vorkenntnisse im Bereich Workflow und Informatik und stellen so entsprechende Anforderungen an die Benutzbarkeit des Rahmenwerks (siehe Abschnitt 3).

2.6 Einschränkungen

2.7 Annahmen und Abhängigkeiten

3 Nichtfunktionale Anforderungen

In diesem Abschnitt werden die nichtfunktionalen Anforderungen an die zu entwickelnde Software beschrieben. Dafür werden die entsprechenden Software-Qualitäten aufgeführt und ihre Bedeutung für die zu entwickelnde Software erläutert.

3.1 Mengengerüst

3.2 Benutzbarkeit

Die Benutzbarkeit soll sich vor allem an Nutzer mit wenig Kenntnissen im Umgang mit Workflows und BPEL richten und dafür die größtmögliche Transparenz liefern, d.h. dass die interne Prozesslogik der Software bestmöglich vom Benutzer abgeschirmt wird und er eine möglichst einfache und schnell verständliche Schnittstelle zur Software erhält, um die Verwendung von SIMPL für alle Benutzergruppen zu ermöglichen.

3.3 Robustheit

3.4 Sicherheit

3.5 Portabilität

3.6 Erweiterbarkeit

Die Erweiterbarkeit des Systems spielt eine zentrale Anforderung, da es über einen langen Zeitraum genutzt und in Zukunft um die Anbindung weiterer Datenquellen, Konzepte für den Datenzugriff und den Umgang mit weiteren Datenformaten ergänzt werden soll. Um die Erweiterbarkeit des Systems zu gewährleisten, wird ein modularer Aufbau zugrunde gelegt und entsprechende Schnittstellen geschaffen.

3.7 Wartbarkeit

3.8 Skalierbarkeit

Die Skalierbarkeit des Systems muss eine sehr flexible Infrastruktur erlauben, da die Computersysteme, auf denen SIMPL später ausgeführt wird, in ihrer Leistung sehr weit auseinander gehen können, d.h. vom normalen Desktop-Computer bis zum Supercomputer kann und soll alles möglich sein.

3.9 ...

4 Akteure

In diesem Abschnitt werden die einzelnen Akteure der Software beschrieben und ihre Abhängigkeiten untereinander definiert.

4.1 Prozess-Modellierer

Der P

4.2 Workflow-Administrator

4.3 Datenquellen-Administrator

4.4 ODE Workflow-Engine

4.5 Eclipse BPEL Designer

4.6 SIMPL Core

5 Anwendungsfälle (Use-Cases)

Dieser Abschnitt beschreibt die funktionalen Anforderungen an die Software. Dazu werden alle Anwendungsfälle eines jeden Akteurs beschrieben und deren Zusammenhänge in entsprechenden Diagrammen graphisch dargestellt.

5.1 Diagramm aller Anwendungsfälle

Abbildung 2 zeigt das Diagramm aller Anwendungsfälle der gesamten Software. Dadurch wird die Funktionalität und die Akteure des späteren Gesamtsystems sichtbar.

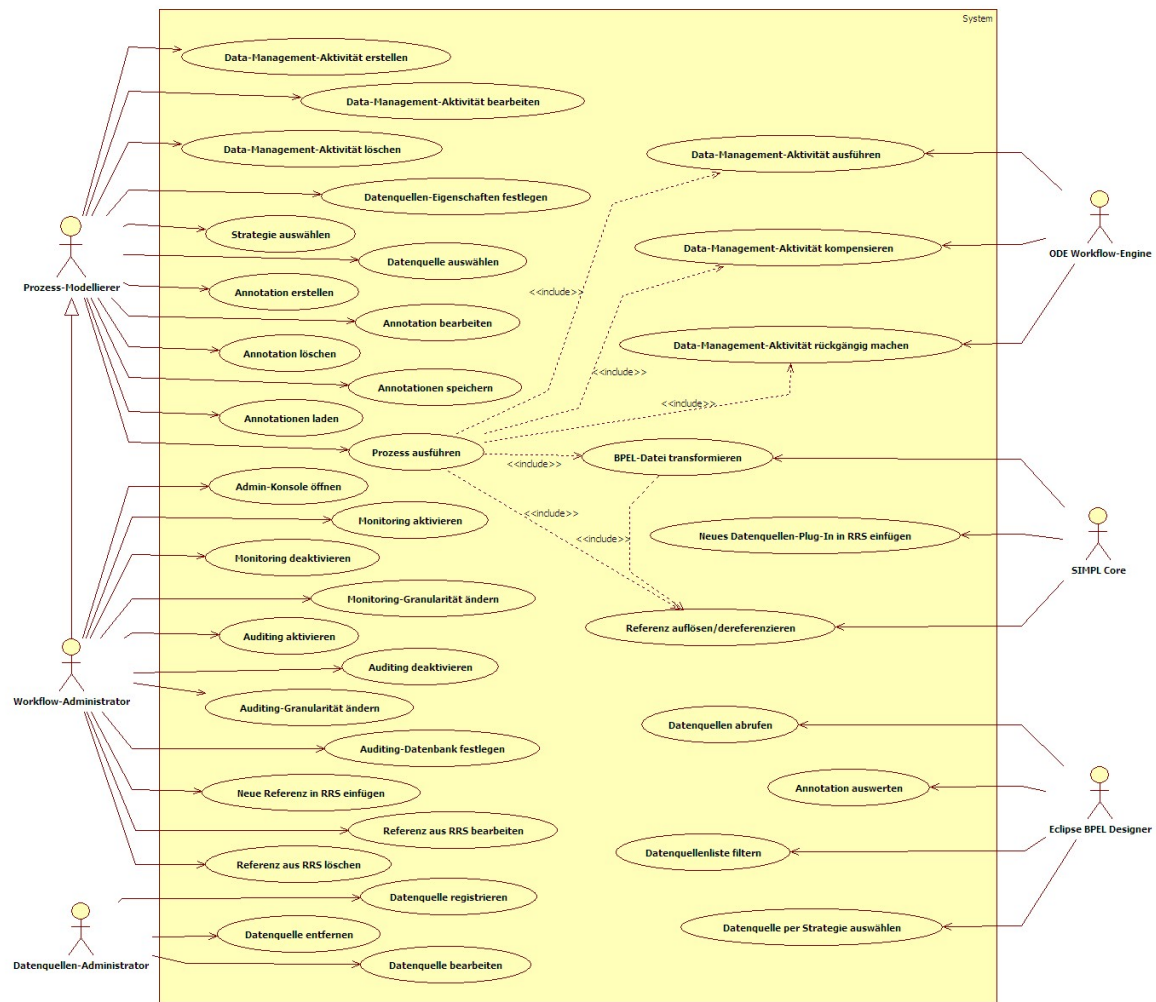


Abbildung 2: Anwendungsfall-Diagramm des gesamten Softwaresystems

5.2 Anwendungsfälle der Prozess-Modellierer

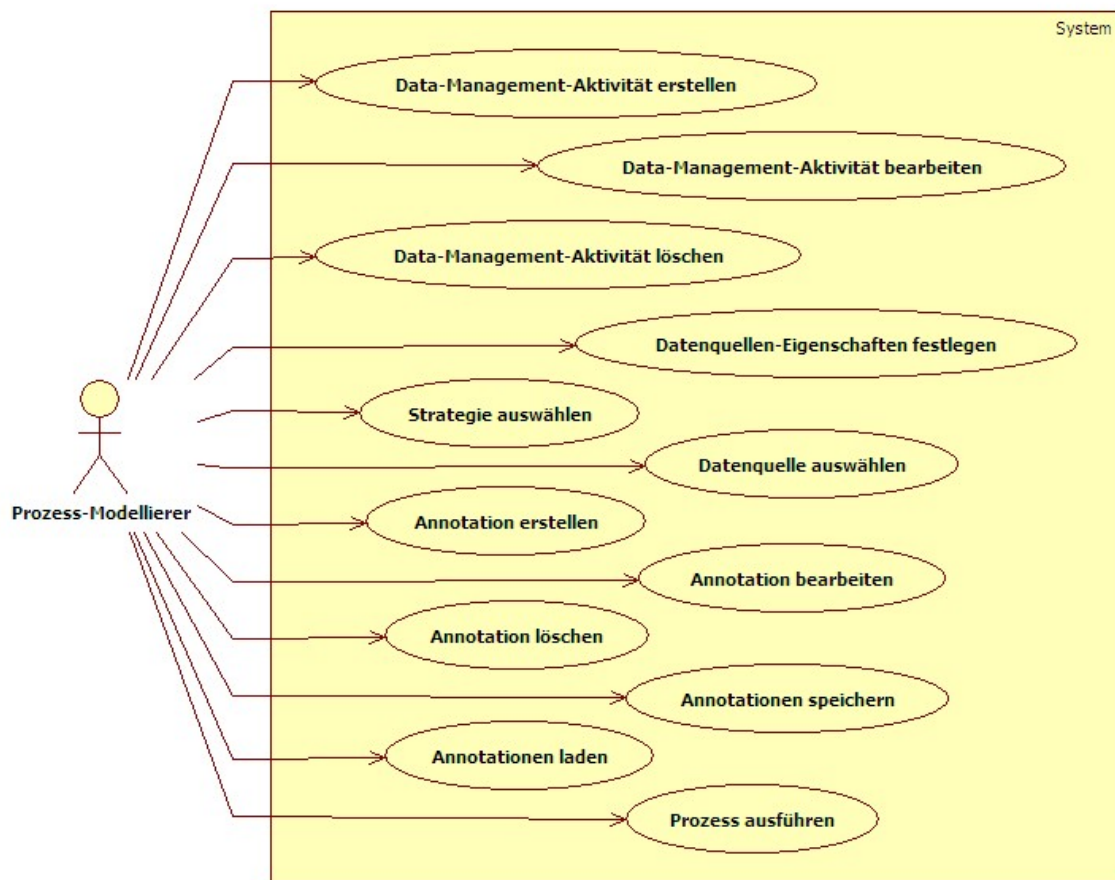


Abbildung 3: Anwendungsfall-Diagramm für die Prozess-Modellierer

5.2.1 Data-Management-Aktivität erstellen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.2 Data-Management-Aktivität bearbeiten

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.3 Data-Management-Aktivität löschen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.4 Strategie auswählen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.5 Datenquellen-Eigenschaften festlegen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.6 Datenquelle auswählen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.7 Annotation erstellen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.8 Annotation bearbeiten

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.9 Annotation löschen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.10 Annotationen speichern

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.11 Annotationen laden

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.2.12 Prozess ausführen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3 Anwendungsfälle der Workflow-Administratoren

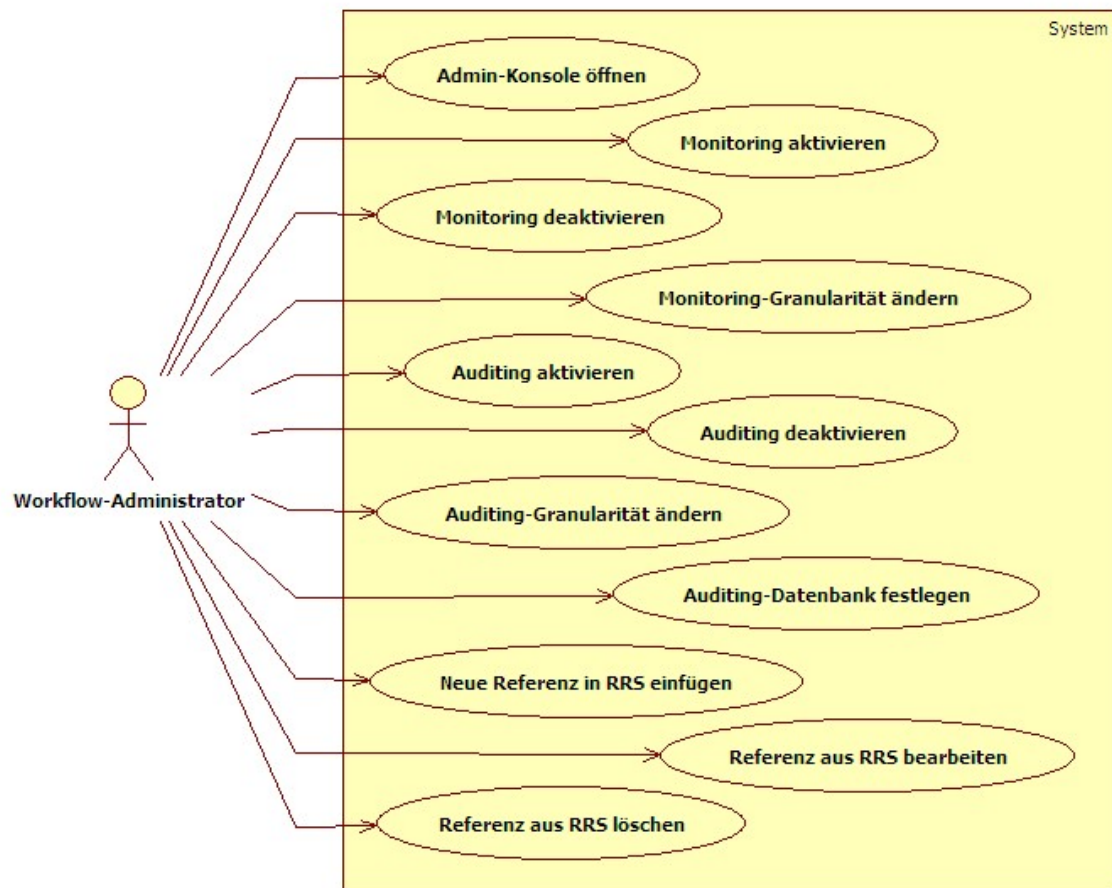


Abbildung 4: Anwendungsfall-Diagramm für die Prozess-Modelierer

5.3.1 Admin-Konsole öffnen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.2 Monitoring aktivieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.3 Monitoring deaktivieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.4 Monitoring-Granularität ändern

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.5 Auditing aktivieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.6 Auditing deaktivieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.7 Auditing-Granularität ändern

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.8 Auditing-Datenbank festlegen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.9 Neue Referenz in RRS einfügen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.10 Referenz aus RRS bearbeiten

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.3.11 Referenz aus RRS löschen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.4 Anwendungsfälle der Datenquellen-Administratoren

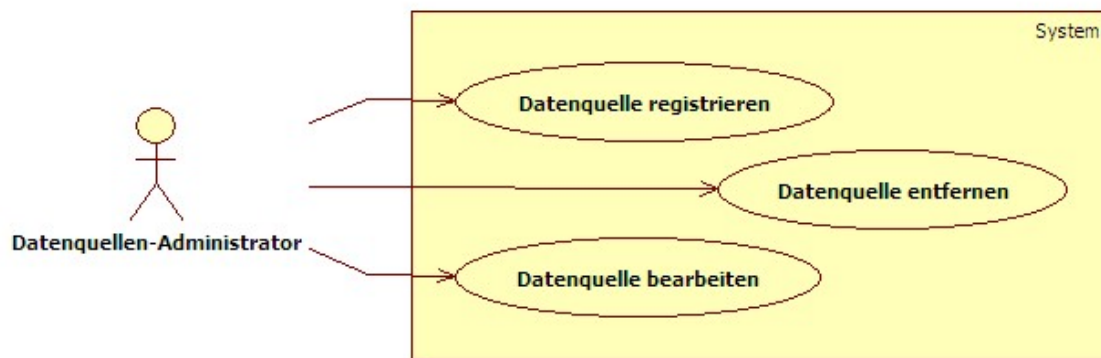


Abbildung 5: Anwendungsfall-Diagramm für die Prozess-Modelierer

5.4.1 Datenquelle registrieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.4.2 Datenquelle entfernen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.4.3 Datenquelle bearbeiten

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.5 Anwendungsfälle der ODE Workflow-Engine

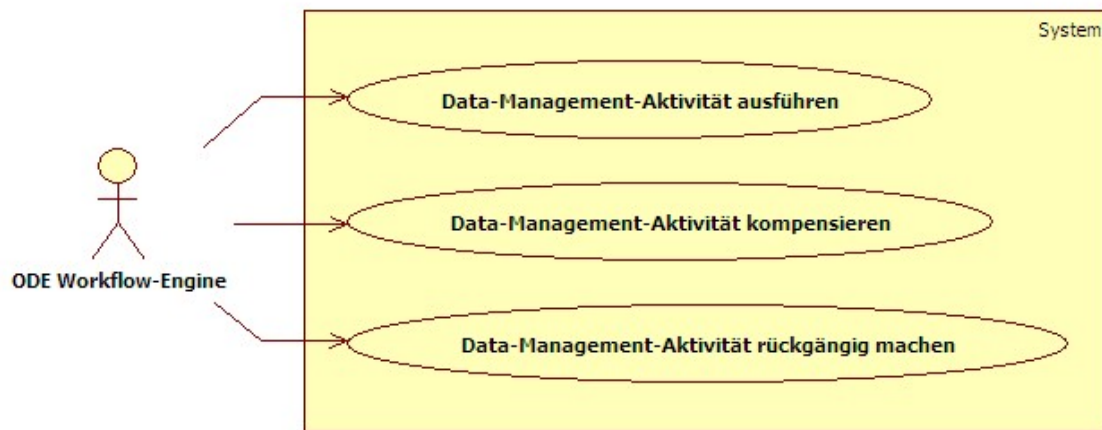


Abbildung 6: Anwendungsfall-Diagramm für die Prozess-Modelierer

5.5.1 Data-Management-Aktivität ausführen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.5.2 Data-Management-Aktivität kompensieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.5.3 Data-Management-Aktivität rückgängig machen (rollback)

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.6 Anwendungsfälle des Eclipse BPEL Designers

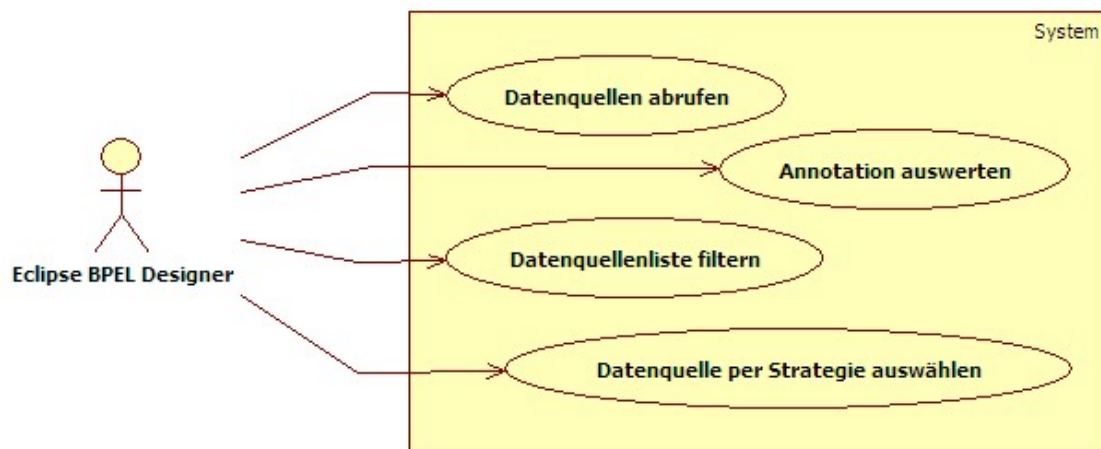


Abbildung 7: Anwendungsfall-Diagramm für die Prozess-Modelierer

5.6.1 Datenquellen abrufen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.6.2 Annotation auswerten

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.6.3 Datenquellenliste filtern

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.6.4 Datenquelle per Strategie auswählen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.7 Anwendungsfälle des SIMPL Core

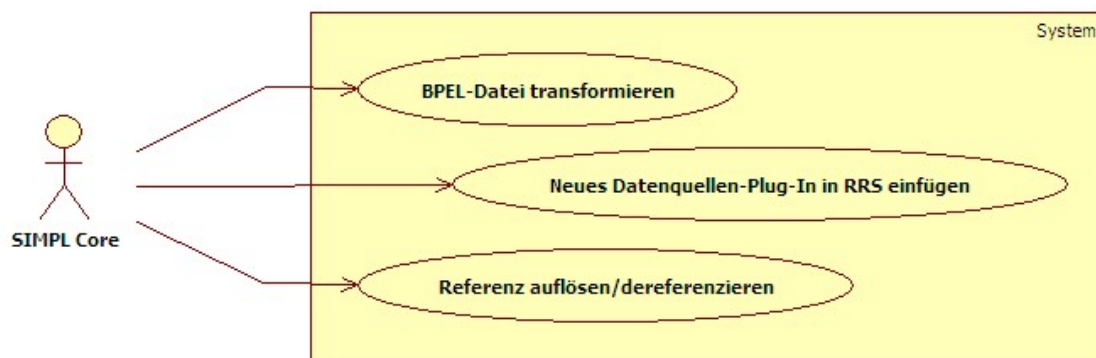


Abbildung 8: Anwendungsfall-Diagramm für die Prozess-Modelierer

5.7.1 BPEL-Datei transformieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.7.2 Neues Datenquellen-Plug-In in RRS einfügen

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

5.7.3 Referenz auflösen/dereferenzieren

Ziel	
Vorbedingung	
Nachbedingung	
Nachbedingung im Sonderfall	
Normalablauf	
Sonderfälle	

6 Konzepte und Realisierungen

6.1 Reference Resolution System

6.2 BPEL-SQL

6.3 Eclipse BPEL Designer

6.4 Monitoring

6.5 ...

7 Einzusetzende Technologien

7.1 Materialien

7.2 Werkzeuge

7.2.1 Entwicklungsumgebung

7.2.2 Sonstige Werkzeuge

8 Änderungsgeschichte

- **Version 0.1**, 11. September 2009: Erstellung des Dokuments.

Literatur

[1] Begriffslexikon