

Feinentwurf

Version 0.1

11. Januar 2010



Inhaltsverzeichnis

1	Einleitung	3
1.1	Zweck dieses Dokuments	3
1.2	Gliederung	3
2	Komponenten	3
2.1	SIMPL Core	3
2.1.1	Paketstruktur	4
2.1.2	SIMPLCore	5
2.1.3	SIMPL Core Services	5
2.1.4	Datasource Plug-Ins	5
2.1.5	Web Services	5
2.2	Apache ODE	6
2.2.1	SIMPL DM Extension Activities	6
2.2.2	SIMPL DAO	6
2.3	Eclipse	6
2.3.1	BPEL DM Plug-In	6
2.3.2	SIMPL Core Plug-In	6
2.3.3	SIMPL Core Client	6
3	Kommunikation	6
3.1	SIMPL Rahmenwerk	6
3.1.1	Speicherung von Einstellungen über die Adminkonsole	7
	Literaturverzeichnis	8
	Abkürzungsverzeichnis	9
	Abbildungsverzeichnis	10

Änderungsgeschichte

Version	Datum	Autor	Änderungen
0.1	13.11.2009	zoabifs	Erstellung des Dokuments.
0.2	04.01.2010	schneimi	Überarbeitung der Struktur
0.3	09.01.2010	schneimi	SIMPL Core Beschreibung

1 Einleitung

Dieses Kapitel erklärt den Zweck des Dokuments, den Zusammenhang zu anderen Dokumenten und gibt dem Leser einen Überblick über den Aufbau des Dokuments.

1.1 Zweck dieses Dokuments

Der Feinentwurf beschreibt Details der Implementierung der Komponenten, die im Grobentwurf [1] in Kapitel 3 vorgestellt wurden. Die Komponenten werden ausführlich beschrieben und ihre Funktionalität durch statische und dynamische UML-Diagramme visualisiert. Der Feinentwurf bezieht sich im Gegensatz zum Grobentwurf nur auf die aktuelle Iteration und wird mit folgenden Iterationen vervollständigt. Grobentwurf und Feinentwurf bilden zusammen den Gesamtentwurf des SIMPL Rahmenwerks.

1.2 Gliederung

Der Feinentwurf gliedert sich in die folgenden Kapitel.

- Kapitel 2 “Komponenten” beschreibt die Realisierung der einzelnen Komponenten mit ihren Schnittstellen.
- Kapitel 3 “Kommunikation” beschreibt die Kommunikation im SIMPL Rahmenwerk über diese Schnittstellen.

2 Komponenten

In diesem Kapitel wird die Realisierung der einzelnen Komponenten im Detail beschrieben.

2.1 SIMPL Core

Abbildung 1 zeigt den Aufbau des SIMPL Cores, der in den folgenden Abschnitten beschrieben wird.

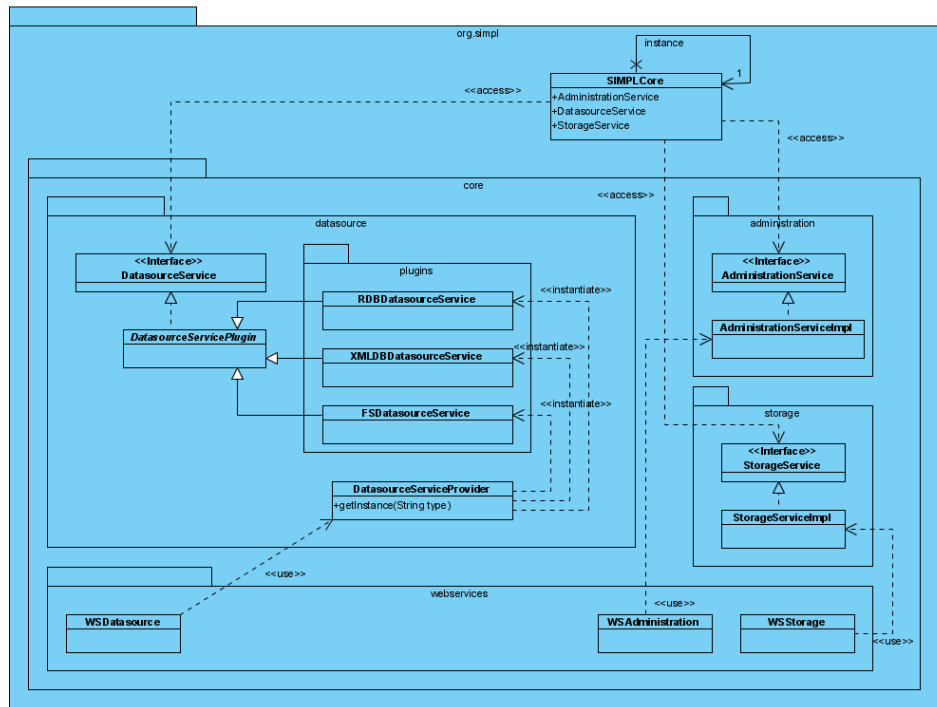


Abbildung 1: SIMPL Core Klassendiagramm

2.1.1 Paketstruktur

Der SIMPL Core besitzt folgende Paketstruktur, die sich in einen Kernbereich, sowie Bereiche für die Dienste (Services) und Web Services aufteilt.

org.simpl.core Hier befinden sich zentrale Klassen des SIMPL Cores, die auf die Dienste des SIMPL Core zurückgreifen, wie z.B. die SIMPLCore-Klasse, die in Kapitel 2.1.2 näher beschrieben wird.

org.simpl.administration Hier befinden sich alle Klassen zur Realisierung des Administration Service ([1] 3.3.1).

org.simpl.storage Hier befinden sich alle Klassen zur Realisierung des Storage Service ([1] 3.3.5).

org.simpl.datasource Hier befinden sich alle Klassen zur Realisierung des Datasource Service ([1] 3.3.5).

org.simpl.datasource.plugins Hier befinden sich die Plug-Ins für den Datasource Service, die für die verschiedenen Datenquellentypen entwickelt werden. Falls sich die einzelnen Plug-Ins auf mehrere Klassen verteilen, werden diese zusätzlich auf eigene Unterpakete verteilt. Das Plug-In-System wird in Kapitel 2.1.4 näher beschrieben.

org.simpl.webservices Hier befinden sich die Web Services des SIMPL Core, die den Zugriff von Außen auf den SIMPL Core ermöglichen. Alle Klassen werden mit JAX-WS-Annotationen versehen und können als Webservices in Apache ODE deployt werden.

2.1.2 SIMPLCore

Die SIMPLCore-Klasse bildet den zentralen Zugriffspunkt auf alle Dienste des SIMPL Cores für den Zugriff auf Klassenebene. Damit die Instanzen der Dienste nur einmal existieren und nicht bei jedem Zugriff erneut erstellt werden, ist die Klasse als Singleton ([1] Kapitel 3.3) ausgelegt. Diese Klasse wird von den Apache ODE Extension Activities (siehe 2.2.1) benutzt um DM-Aktivitäten auszuführen.

2.1.3 SIMPL Core Services

In diesem Abschnitt werden die Dienste des SIMPL Cores beschrieben.

Administration Service

Storage Service

Datasource Service

2.1.4 Datasource Plug-Ins

Die Unterstützung verschiedener Typen von Datenquellen wird durch Datasource Plug-Ins realisiert um eine Erweiterungsmöglichkeit zu garantieren. Dies wird durch die Bereitstellung einer abstrakten Klasse erreicht, von der sich die Plug-Ins ableiten lassen. Mit der Reflection API von Java ist es möglich, die Plug-Ins zur Laufzeit zu erkennen und zu verwenden, ohne dass bestehender Code angepasst werden muss.

DatasourceService (interface) Das DatasourceService-Interface schreibt alle Funktionen vor, die von den DatasourceServices (Plug-Ins) implementiert werden müssen.

DatasourceServicePlugin (abstract class) Bei der DatasourceServicePlugin-Klasse handelt es sich um eine abstrakte Klasse, die an das DatasourceService-Interface gebunden ist und damit das Grundgerüst für einen Datasource Service bildet. Ein Plug-In muss diese Klasse erweitern und wird dadurch gezwungen das DatasourceService-Interface zu implementieren.

DatasourceServiceProvider Über den DatasourceServiceProvider kann die Instanz eines DatasourceService angefordert werden. Dies geschieht über eine eindeutige Typkennzeichnung (z.B. RDB, XML, ...), mit der die entsprechende Instanz der Klasse über die Class.forName-Methode der Java Reflection API erzeugt und ausgeliefert wird.

2.1.5 Web Services

Die Web Services werden mit den JAX-WS annotierten Klassen wie folgt bereitgestellt. Zunächst wird mit Hilfe des Befehls wsgen.exe (..\Java\jdk1.6.0_14\bin\wsgen.exe), eine WSDL-Datei zu einer Klasse erzeugt. Die WSDL-Datei wird anschließend zusammen mit der kompilierten Klasse als JAR-Datei in Apache ODE hinterlegt (..\Tomcat 6.0\webapps\ode\WEB-INF\servicejars) und wird dann beim Start von Apache Tomcat von Apache ODE als Web Service bereitgestellt.

Datasource Web Service Der Datasource Web Service...

```
public DataObject queryData(String dsAddress, String statement)
```

```
public boolean defineData(String dsAddress, String statement)
```

```
public boolean manipulateData(String dsAddress, String statement, DataObject data)
```

Administration Web Service Der Administration Web Service...

```
public boolean saveSettings(String schema, String table, String settingName, Linked-  
HashMap<String, String> settings)
```

```
public LinkedHashMap<String, String> loadSettings(String schema, String table, String  
settingName)
```

```
public List<String> getDatasourceTypes()
```

```
public List<String> getDatasourceSubTypes(String datasourceType)
```

```
public List<String> getDatasourceLanguages(String datasourceSubType)
```

Storage Web Service Der Storage Web Service...

```
public boolean save(String serviceNS, String serviceName, List<LinkedHashMap<String,  
String>> data)
```

```
public List<LinkedHashMap<String, String>> load(String serviceNS, String ser-  
viceName)
```

2.2 Apache ODE

2.2.1 SIMPL DM Extension Activities

2.2.2 SIMPL DAO

2.3 Eclipse

2.3.1 BPEL DM Plug-In

2.3.2 SIMPL Core Plug-In

2.3.3 SIMPL Core Client

3 Kommunikation

In diesem Kapitel wird die Kommunikation im SIMPL Rahmenwerk beschrieben und wichtige Abläufe deutlich gemacht.

3.1 SIMPL Rahmenwerk

In Abbildung 2 wird die Kommunikation zwischen den Komponenten beschrieben und die entsprechenden Funktionsaufrufe gezeigt.

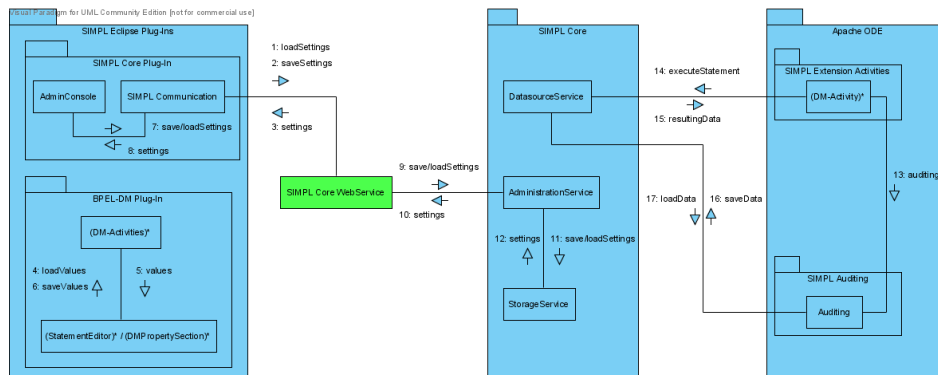


Abbildung 2: Kommunikation im SIMPL Rahmenwerk

3.1.1 Speicherung von Einstellungen über die Adminkonsole

Abbildung 3 zeigt den Ablauf bei der Speicherung von Einstellungen der Adminkonsole und das Zusammenspiel der beteiligten Komponenten.

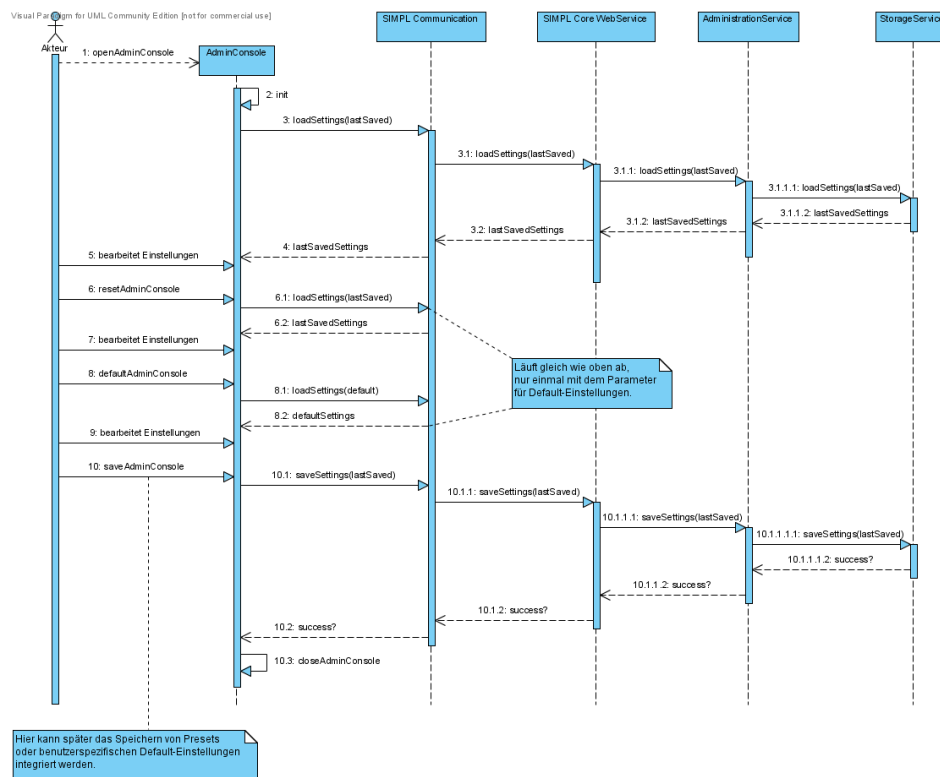


Abbildung 3: Speicherung von Einstellungen über die Adminkonsole

Literatur

- [1] *Grobentwurf v1.5*. Stupro-A SIMPL (2009)

Abkürzungsverzeichnis

UML	Unified Modeling Language

Abbildungsverzeichnis

1	SIMPL Core Klassendiagramm	4
2	Kommunikation im SIMPL Rahmenwerk	7
3	Speicherung von Einstellungen über die Adminkonsole	7