



Einbindung dynamischer Sprachen in Java



Steffen Schluff

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

Gliederung

- Einleitung
- Einsatzszenarien
- Java Scripting API
- Zusammenfassung

Java und XML

) Software Factory)

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

) Object Rangers)

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

) Competence Center)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen



Steffen Schluff

~~— I'm a programar~~
~~— I am a programmer~~
~~— I'm a programor~~
I write code

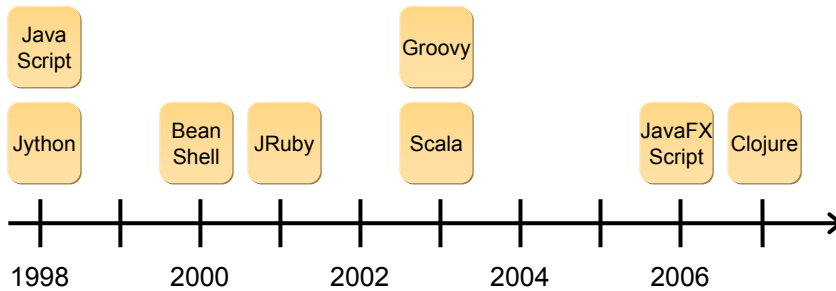
Beratung
Projekte
Training



- Einleitung
- Einsatzszenarien
- Java Scripting API
- Zusammenfassung

- [..] there are four completely different things that go by the name Java
- Language
 - Enormous class library
 - Virtual machine
 - Security model

(Jamie Zawinski, „java sucks.“, 2000)



7

- JVM Sprachen werden häufig als Skriptsprachen bezeichnet
 - Beispiel: JSR 223 - Scripting for the Java Platform
- „Skript“ ermöglicht Steuerung und Kontrolle eines „Programms“
 - Häufig interpretiert oder mit vernachlässigbarem Kompilierschritt
- Begriffe Skript und Programm heute schwer unterscheidbar
 - Eigene Benutzer sagen nicht Skriptsprache
- Skriptsprachen werden mit bestimmten Eigenschaften assoziiert
 - Vor allem Speicherverwaltung und dynamische Typisierung

8

„We got both kinds. We got Country and Western.“

- Viele dynamisch typisierte Sprachen sind älter als Java...
 - ... und die JVM Portierungen sind auch nicht neu
- Renaissance durch Web Frameworks als „Killer Apps“
 - Ruby on Rails, Grails, Django, ...
- Wiederentdecken für andere isolierte Zwecke
 - Shell Skripte, Buildmanagement, Testen, ...
- Einbindung in bestehende Java Anwendungen
 - Neues Steinchen im Java Architektur Baukasten
 - Zum Steuern und Kontrollieren, d.h. Scripting



9

Gliederung

- Einleitung
- Einsatzszenarien
- Java Scripting API
- Zusammenfassung

10

Die üblichen Verdächtigen

- Kommandozeilen Shell in Anwendungen
 - Graphische Konfiguration ist Standard
 - Administratoren bevorzugen häufig Kommandozeile
- Anpassung und Erweiterung von Anwendungen
 - Auslagern von Geschäftslogik und dynamische Konfiguration
 - Benutzermakros und Reporting
- Rapid Prototyping
 - Vermeidung des „edit-compile-deploy“ Zyklus
 - Ermöglicht schnelles Benutzer-Feedbacks



11

Wachs in meinen Händen

- Komplexe Anwendungen sollen häufig „customizable“ sein
 - Geschäftslogik, Reporting, Benutzermakros
 - Benutzerschnittstelle von Computerspielen
- Keiner ist wirklich geeignet
 - Entwickler ist zu teuer und hat kein fachliches Verständnis
 - Anwender kann nicht (genug) programmieren
- Auslagern von bestimmten Anwendungsteilen
 - Zugriff durch kundengerechte Abstraktion
 - Skriptsprachen oder Domain Specific Language (DSL)



12

- Experimentierkasten um Ideen auszuprobieren
 - Ziel ist das schnelle Evaluieren und Bewerten von Ideen
 - Lösung hat keinen Anspruch auf technische „Perfektion“
 - Gewisse Opfer werden gemacht (Architektur, Typisierung, ...)
- Dynamische Sprachen können unterstützen
 - Buildprozess schrumpft von „edit-compile-run“ auf „edit-run“
 - Dynamische Sprachen sind oft einfach und mächtig
 - Sprache ist aber nur Teil der Lösung (Infrastruktur, Frameworks, ...)
- Prototypen können gefährlich sein
 - Kunden können Prototyp und Produkt oft nicht unterscheiden
 - Nicht jedes dynamische Framework ist ein Prototyp-Framework

- Einleitung
- Einsatzszenarien
- **Java Scripting API**
- Zusammenfassung

Wo tut sich was?

- Proprietäre Mechanismen der einzelnen Sprachen
- Apache Bean Scripting Framework (BSF)
- Scripting Support in Spring
- Java Community Process Scripting JSRs
 - JSR 223 - Scripting for the Java Platform
 - JSR 241 - Groovy Programming Language
 - JSR 274 - BeanShell Scripting Language
 - JSR 292 - Dynamically Typed Language Support
- Hauptentwickler von Jython und JRuby arbeiten bei Sun
 - Gleiches gilt für IronPython und Microsoft

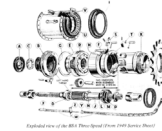
15

Java Scripting API

- „Java Scripting API“ erlaubt Java Einbindung von Skriptsprachen
 - Framework zur „Script Engine“ Anbindung von Drittanbietern
 - Einbindendes Programm muss konkrete Sprache nicht kennen
- Basiert auf JSR-223 „Scripting for the Java Platform“
 - „[...] background to bridge the scripting and the Java community.“
- Seit Java 6 fester JDK Bestandteil
- Java 6 beinhaltet „Mozilla Rhino“
 - JavaScript Implementierung für Java (Rhino 1.6R2)
 - Weitere Sprachen per „scripting project“ (<http://scripting.dev.java.net>)

16

- Überschaubare API
 - 6 Interfaces, 5 Klassen, 1 Exception
- Enthalten in Package javax.script
 - Weitere Tools und Beispiele im JDK (jrunscript, jconsole-plugin, ...)
- Zwei grobe Aufgabenbereiche
 - Scripting Funktionalität für die jeweiligen Sprachen
 - Kontext in dem ein Script ausgeführt wird

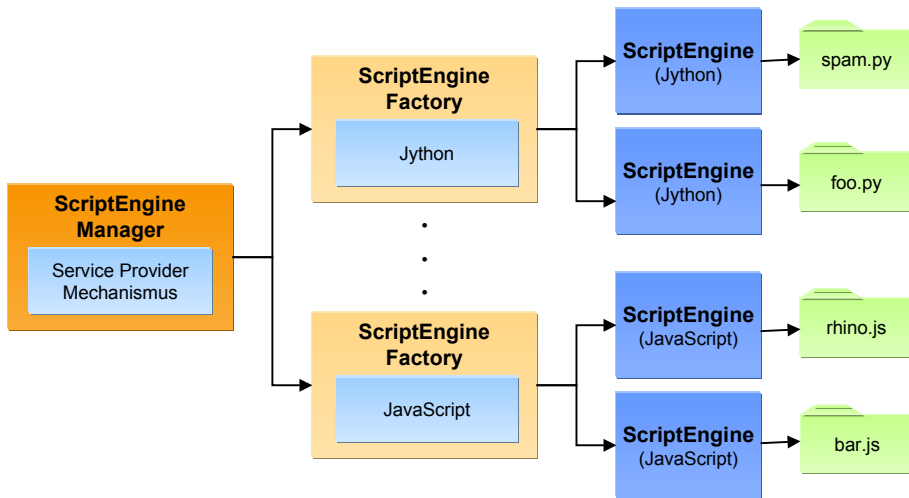


17

- Interface ScriptEngine
 - Übersetzung und Evaluierung (Ausführung) von Skripten
 - Überladene Methode eval() für verschiedene Quellen (String, Reader)
- Interface ScriptEngineFactory
 - Erzeugt ScriptEngine Instanzen
 - Enthält ScriptEngine Metadaten (Name, Version, File Extension, ...)
 - Enthält Methoden zur Skripterstellung in der jeweiligen Sprache
- Klasse ScriptEngineManager
 - Sucht vorhandene ScriptEngineFactory Implementierungen
 - Verwendet Service Provider Mechanismus aus Jar Spezifikation
 - Konkrete Klasse als programmatischer Einstiegspunkt

18

Motoren, Fabriken und Manager (2)



19

Hello World in JavaScript

```
ScriptEngineManager mgr = new ScriptEngineManager();

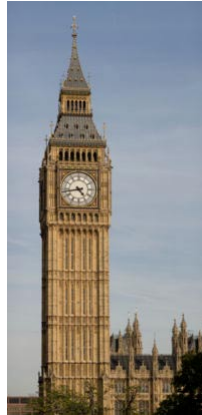
ScriptEngine engine = mgr.getEngineByName("JavaScript");

String sayHello = "println('Hello World')";

// Throws ScriptException
engine.eval(sayHello);
```

20

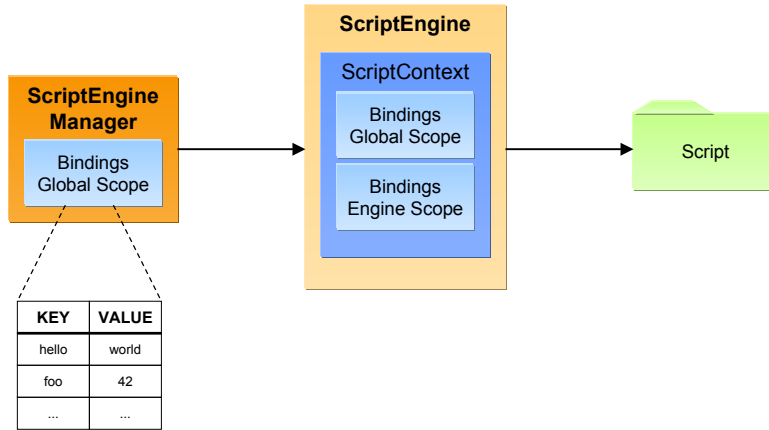
- Wenn noch Zeit ist...



21

- Interface ScriptContext
 - Verbindet ScriptEngine und ausführende Applikation
 - Hat Reader und Writer für ScriptEngine Ein-/Ausgabe
 - Hat ein oder mehrere Bindings
- Interface Bindings
 - Subtyp von Map<String, Object>
 - Für Wertaustausch von ScriptEngine und Applikation
- Scope ordnet die Bindings eines ScriptContext
 - Global Scope - Sichtbar in allen Engines
 - Engine Scope - Sichtbar in einer bestimmten Engine Instanz
 - Suche nach Werten vom kleinsten Scope aufwärts

22



23

- Manche Features von ScriptEngine sind optional
 - Wenn vorhanden, implementiert ScriptEngine zusätzliche Interfaces
- Interface **Compilable**
 - Skripte sind mehrfach ausführbar ohne Neucompilierung
 - Verwendung eines Abstract Syntax Tree (AST) oder Java Byte Code
 - ScriptEngine Instanz wird umgewandelt in CompiledScript Objekt
- Interface **Invocable**
 - Einzelne Methoden eines Skripts sind gezielt ausführbar
 - Methoden eines Skript können Java Interfaces implementieren

24

Demo „Confluence Wiki mit Scriptix Plugin“



25

- Einleitung
- Einsatzszenarien
- Java Scripting API
- **Zusammenfassung**

26

„Every sufficiently large Java program, anything beyond medium-sized, needs a scripting engine, whether the authors realize it or not.“

(Steve Yegge, Stevey's Blog Rants: „The Universal Design Pattern“, 2008)

27

- Java Scripting Programmer's Guide
 - http://java.sun.com/javase/6/docs/technotes/guides/scripting/programmer_guide/index.html
- The scripting project
 - <http://scripting.dev.java.net>
- Confluence - Enterprise Wiki Software
 - <http://atlassian.com/software/confluence>
- Scriptix - Confluence Extensions
 - <http://confluence.atlassian.com/display/CONFEXT/Scriptix>

28



Vielen Dank für Ihre Aufmerksamkeit !

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0



Fragen ?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Version: 1.0

- Schulung: JavaScript
 - <http://www.oio.de/seminar/web-design/seminar-training-javascript-schulung.htm>
- Schulung: Einführung in Groovy
 - <http://www.oio.de/kurs-java-training-groovy-schulung.htm>
- Artikel: Einführung in Groovy
 - <http://www.oio.de/public/java/groovy/groovy-einfuehrung.htm>
- Beratung durch OIO zum Einsatz von Groovy und Grails
 - <http://www.oio.de/beratung-consulting/open-source-software/java-frameworks/groovy-grails-consulting.htm>