Skriptsprachen im Vormarsch: Einsatz in Enterprise-Applikationen

W3L AG
info@W3L.de

2007

Inhaltsverzeichnis

zurück

Einleitung

- **Was ist daran neu?**
  - Skriptsprachen sind nicht neu!
  - Client- und Server-seitig bereits im Einsatz!
    - Mit PHP, CGI, Tcl, Perl (...) auch im Enterprise-Umfeld eingesetzt
    - SAP Scripting in a Box
- **Neu ist jedoch...**
  - Scripting for the Java™ Plattform
    - Herausgegangen aus JSR 223
  - Microsofts Dynamic Language Runtime (DLR)
- **Warum?**
  - Customizing
  - Flexibilität
  - Dynamische Änderbarkeit
  - Rapid Application Development (RAP)

Was sind Skriptsprachen?

- **Merkmale**
  - werden meistens interpretiert
  - Syntax und Sprachumfang meistens unkomplizierter als konventionelle Sprachen
    - Erlernung und Benutzung einfacher
    - Interpreter kompakt
  - Garbage Collector
  - Plattformunabhängig
  - Dynamische Typisierung
    - Fehlende Typunterscheidung
    - Manche unterstützen das Meta-Object-Protocol (MOP)

Was sind Skriptsprachen?

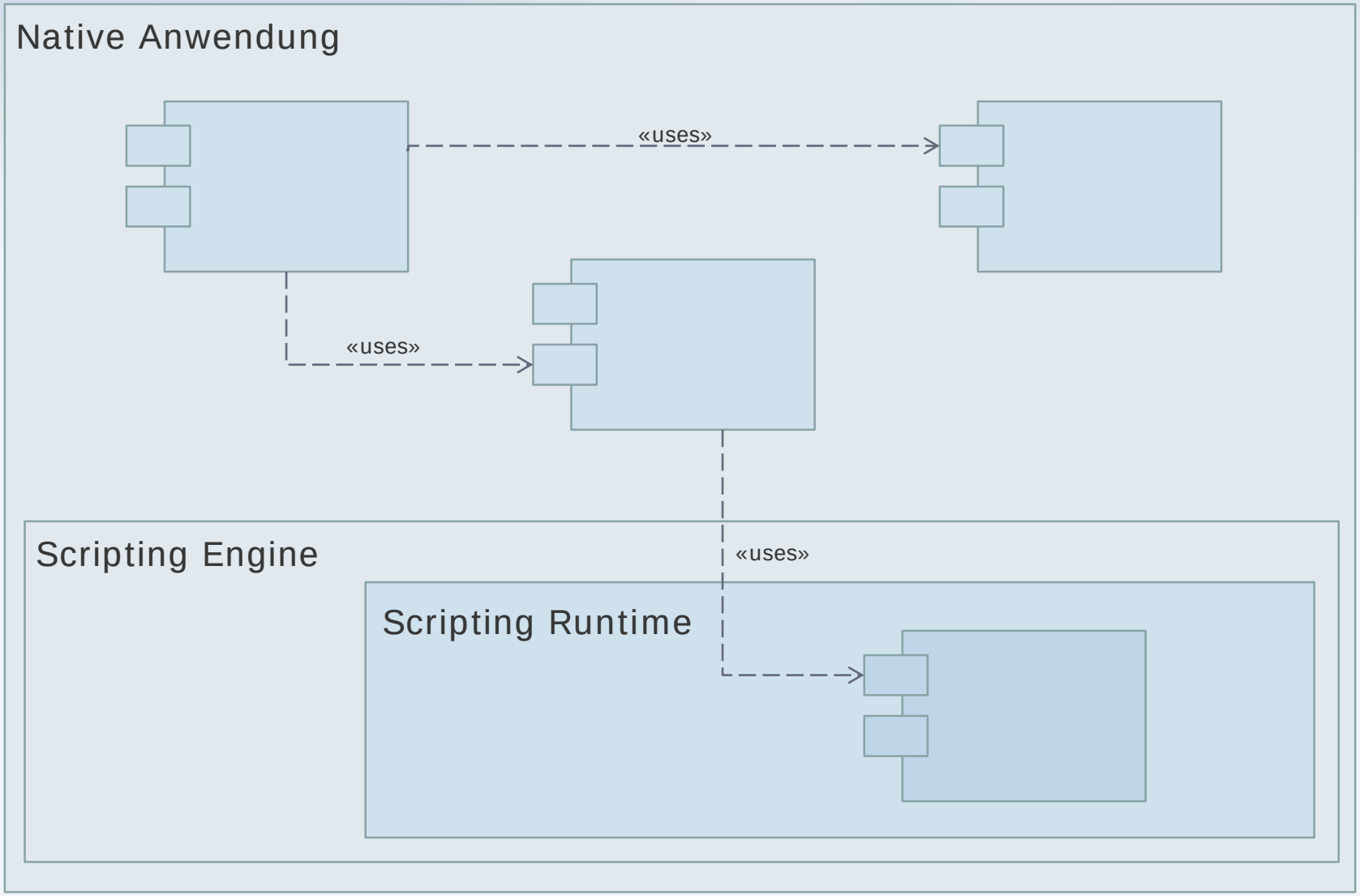
- **MOP**
  - Erlaubt die Schnittstellen-Änderung zur Laufzeit
  - Beispiel Javascript

```
var meinfeld = new Array("1", "2", "3", "4", "5", "6", "7");
alert(meinfeld.contains("1")); //Fehler

//Dynamische Änderung der Schnittstelle
Array.prototype.contains = function(e) {
  for (var i = 0; i < this.length; i++)
    if (this[i] == a)
      return true;
  return false;
}
alert(meinfeld.contains("1")); //OK

//Alternativ. Auf das Exemplar beschränkt
meinfeld.contains = function(e) {…}
```

Vor- und Nachteile von konventionellen und Skriptsprachen

- **Konventionelle Sprachen**
  - Vorteil
    - Ausführungsgeschwindigkeit
    - Typsicher
    - Große Bibliotheken verfügbar
  - Nachteil
    - Keine Rapid Application Development (RAD) wegen Write-Compile-Run-Zyklus
    - Dynamische Anpassung des Codes nicht möglich
- **Skriptsprachen**
  - Vorteil
    - Bieten RAD
    - Dynamischer Code
  - Nachteil
    - Nicht umfassend: Setzen eine echte Programmiersprache voraus.
    - Ausführungsgeschwindigkeit

Konzeptionelle Architektur der Anbindung



Scripting for the Java™ Plattform

- **Neues API ab Java 6**
  - Anbindung verschiedener Skriptsprachen an das Java-Laufzeitsystem
  - Enthält Rhino
  - Idee
    - Auslagern spezialisierter Aufgaben in Skripte (Reportgenerierung und Textanalyse mit Perl, Komponenten-*Glue*-*Code* und Controller-Entwicklung mit Javascript)
    - Makromechanismus für jedermann
    - Implementierung eigener Skriptsprachen und –interpreter
  - Definiert Kommunikations-Architektur zwischen Java- und Skript-Programmen
  - Zentrale Schnittstellen und Klassen im Paket `javax.script`
    - `ScriptEngine`
    - `ScriptEngineFactory`
    - `ScriptContext`
    - `Invocable`
    - `ScriptEngineManager`

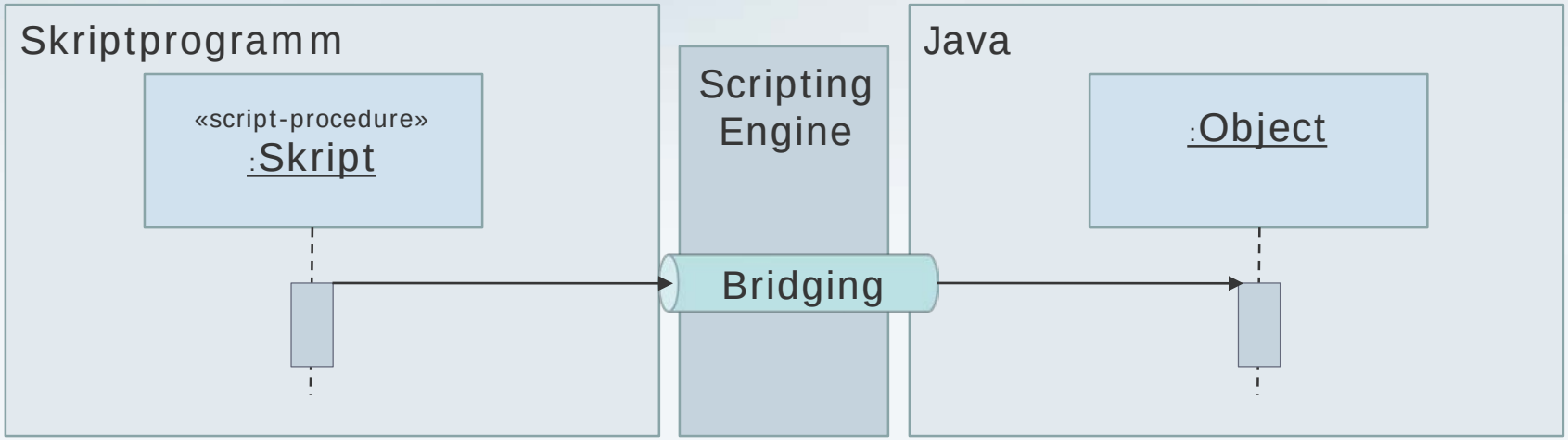
Scripting for the Java™ Plattform

- **Initialisierung und Ausführung**
  - Schritt für Schritt
    1. Ein Exemplar des `ScriptEngineManagers` erzeugen
    2. Entsprechende `ScriptEngine` suchen
    3. Skriptprogramm ausführen
  - Beispiel

```
import javax.script.*;
public class EvalScript {
    public static void main(String[] args) throws Exception {
        // 1. Schritt
        ScriptEngineManager factory =
                new ScriptEngineManager();
        // 2. Schritt
        ScriptEngine engine =
                factory.getEngineByName("JavaScript");
        // 3. Schritt
        engine.eval("print('Freundeskreis 37/H07')");
    }
}
```

Scripting for the Java™ Plattform

Zugriff auf globale Java-Objekte (Script-To-Java)

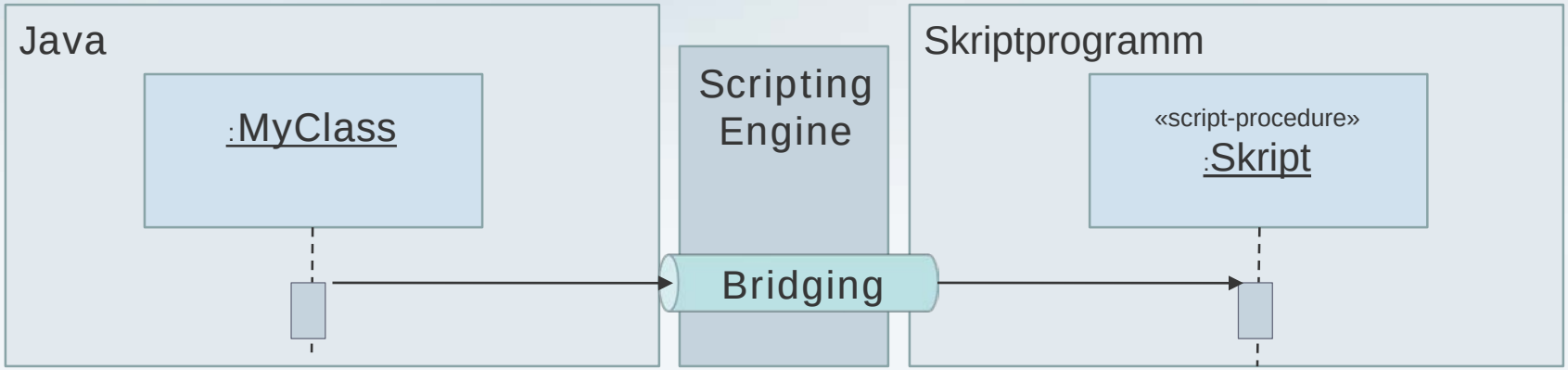


Zugriff auf globale Java-Objekte (Script-To-Java)

```
ScriptEngineManager factory = new ScriptEngineManager();
ScriptEngine engine = factory.getEngineByName("JavaScript");

//Globale Variable registrieren…
engine.put("myGlobalVariable", new File("test.txt"));

//Auf globale Variable zugreifen…
engine.eval("print(myGlobalVariable.getAbsolutePath())");
```

Scripting for the Java™ Plattform

Aufruf von Skript-Funktionen



Zugriff auf globale Java-Objekte (Java-To-Script)

```
ScriptEngineManager factory = new ScriptEngineManager();
ScriptEngine engine = factory.getEngineByName("JavaScript");

//Skriptprogramm laden…
engine.eval("function hello(n) {print('hello'+n);}");

//Globale Skript-Funktion aufrufen…
Invocable inv = (Invocable)engine;
inv.invokeFunction("hello", "hello");
```

Scripting for the Java™ Plattform

Implementierung von Java-Schnittstellen

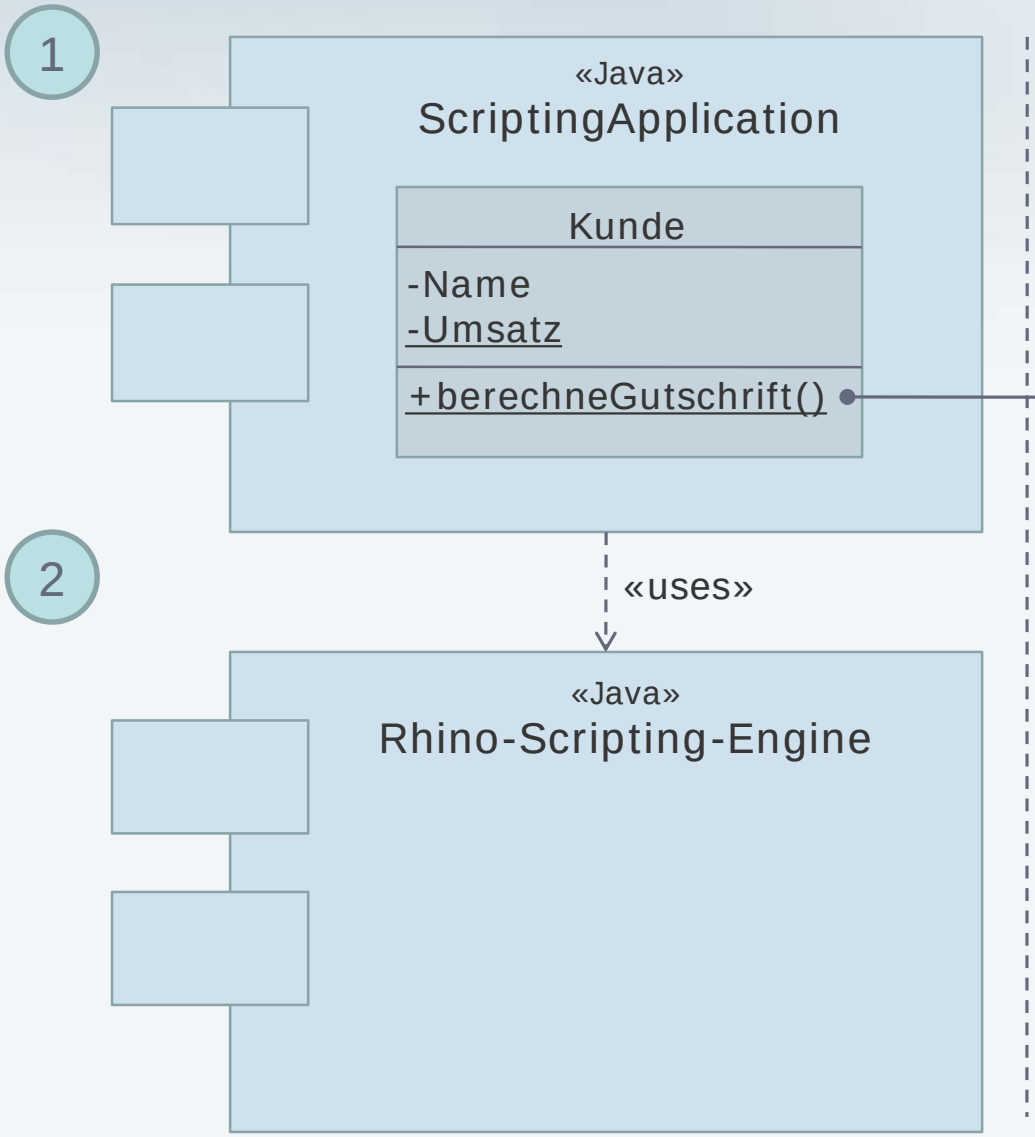

```
var r = new java.lang.Runnable() {
    run: function() {
        print("running...\n");
    }
};
var th = new java.lang.Thread(r);
th.start();
```

Alternativ

```
Object run = engine.get("r");
Invocable inv = (Invocable) engine;
Runnable r = inv.getInterface(run, Runnable.class);
Thread th = new Thread(run);
th.start();
```

Fallstudie: Integration der Rhino-Scripting-Engine

Fallstudie: Integration der Rhino-Scripting-Engine



3 Javascript

```
function berechneGutschrift()
{
  var u = _this.getUmsatz();
  return u * u;
}
```

Benchmark: Rhino vs. Java

- **Schleifen-Iteration von 1 bis 1.000.000**
  - Java: 12ms
  - Rhino: 4s
- **Integer-Vergleich von 1.000.000-Werten**
  - Java: 10ms
  - Rhino: 8s
- **Erzeugung und Initialisierung eines Array mit 100.000-Elementen**
  - Java: 10ms
  - Rhino: 1s
- **Speicherverbrauch nach Initialisierung des Rhino-Interpreters**
  - ca. 1MB

Überblick verfügbarer Skriptsprachen für Java – nicht vollständig!

- **BeanShell**
  - Kleiner, kostenloser embeddable Interpreter für Java-Sourcecode
  - Enthält Features einer objektorientierten Skriptsprache
- **Jess**
  - Regel-Engine ursprünglich von der Experten-Shell CLIPS inspiriert
  - Skripting in deklarativen Regeln
- **JudoScript**
  - Java-artige Skriptsprache
  - Kostenlos, Opensource
  - Unterstützt XML, XSLT, JDBC-Skripting, Sendmail, Java-GUI
- **Rhino**
  - Open-Source-Implementierung von Javascript in Java. Anbindung an das gesamte Laufzeitsystem
- **Jython**
  - Python für Java. Nachfolger von Jpython
- **JRuby**
  - Versuch, einen Ruby-Interpreter in Java umzusetzen

Microsoft‘s Dynamic Language Runtime

- **Setzt auf dem Common Language Runtime (CLR) auf**
- **Entspricht funktional CLR für dynamische Sprachen**
  - Interoperabilität zwischen Objekten
  - Typsystem
- **Anbindung dynamischer Sprachen – auch Skriptsprachen**
- **Unterstützte Sprachen**
  - Python
  - Javascript
  - Dynamic Visual Basic
  - Ruby
- **Unter Microsoft Permissive License (MsPL) veröffentlicht**
  - Open Source

Microsoft‘s Dynamic Language Runtime

- **Beispiel: Ruby-Code mit Zugriff auf Javascript- und VBX-Code**

```
require 'Silverlight.Samples, Controls,Version=0.0.0.0,
JS = require '3Dtext.js'
VB = require 'technorati.vbx,

def initialize(s, e)
  button = Controls.Button.new
  button.Text = "Click Me!„

  JS.initialize

  root.children.add button

  button.click do ls, el
     JS.clearList
     items = VB.GetTitles("silverlight")
  end
end
```

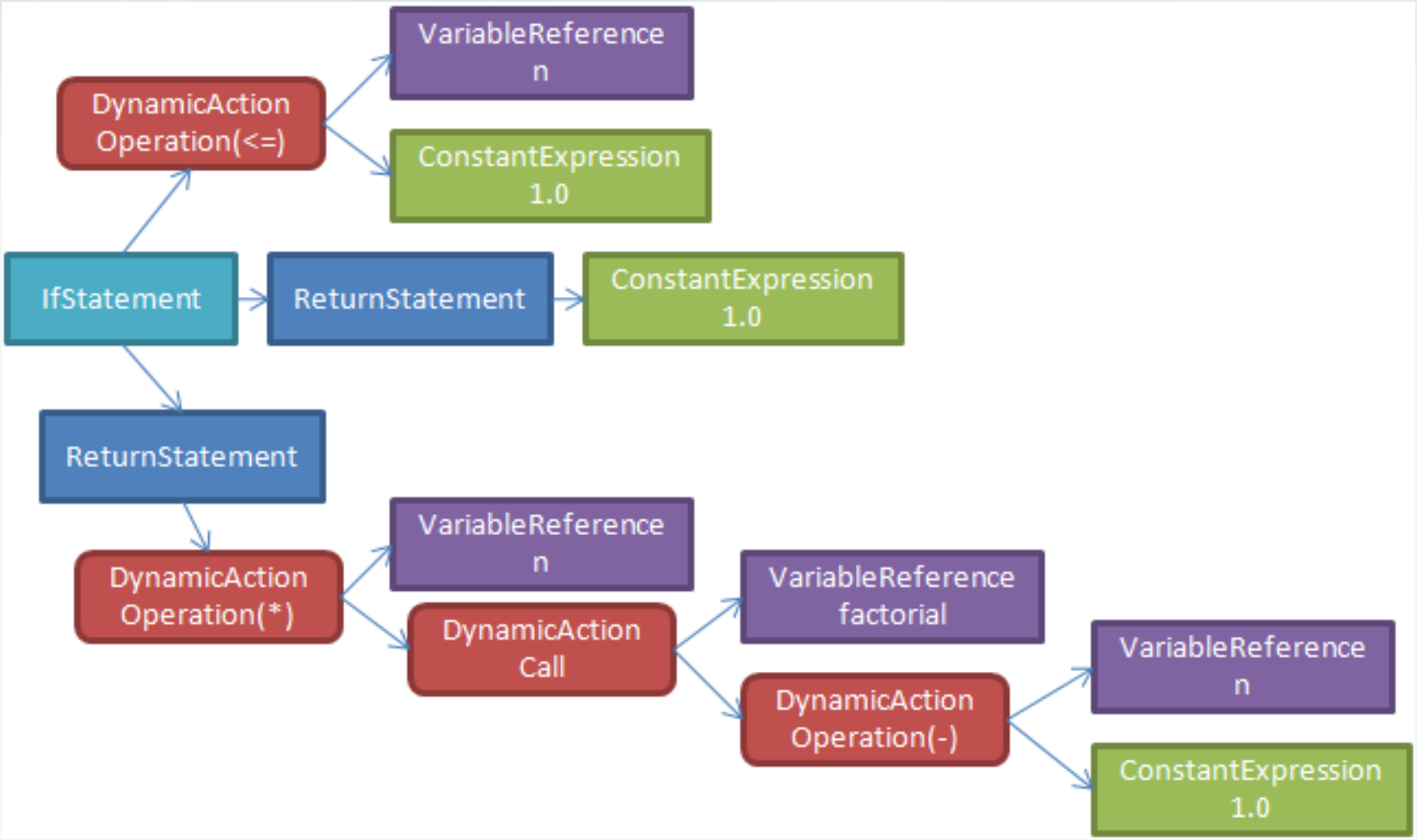
Microsoft‘s Dynamic Language Runtime

- **Trick**
  - Programm ist keine Abfolge von Befehlen,
  - sondern wird als Objektstruktur abgelegt (Syntaxbaum)
- **Grenzen zwischen Daten und Programmen vermischen sich**
  - Programme werden manipulierbar wie Daten
- **Beispiel**

```
function fakueltaet(n) {
   if (n <= 1)
      return 1;
   else
      return n * fakueltaet(n – 1);
}
```

Microsoft‘s Dynamic Language Runtime

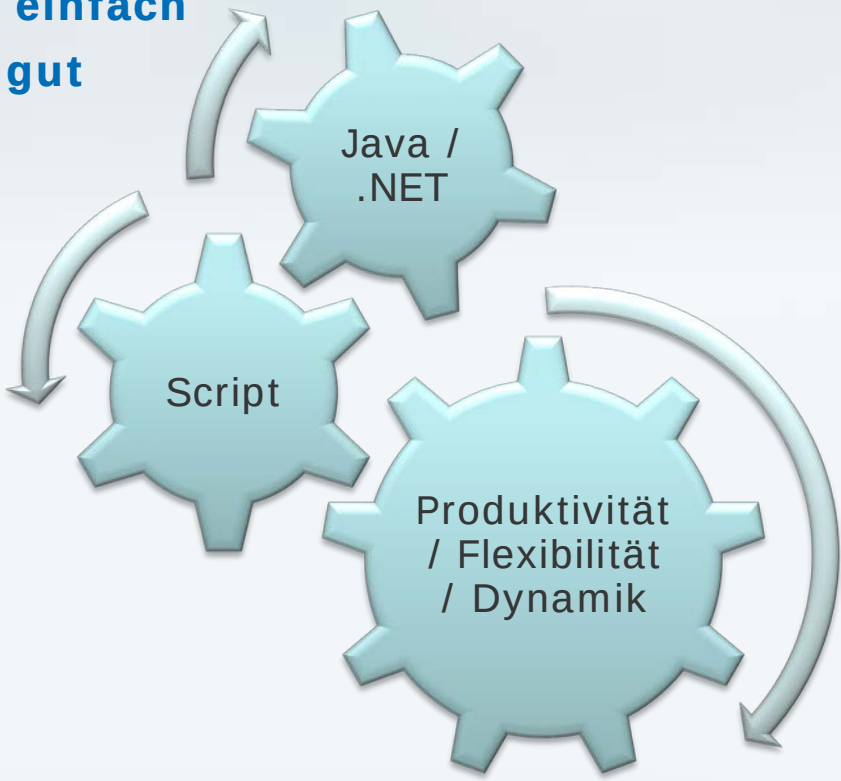
- **Syntaxbaum**



(vgl. Jim Hugunin, Thinking Dynamic, 2007, http://blogs.msdn.com/hugunin)

Fazit

- **Integration einer Scripting-Engine einfach**
- **Vorteile beider Welten lassen sich gut kombinieren**
- **Stabile und ausgereifte Technik**
- **Zusammenfassend**
  - Flexibilisierung
  - Rapid-Application-Development
  - Dynamische Änderbarkeit



Vielen Dank!

Inhouse-Schulungen

Wir bieten Inhouse-Schulungen und Beratung durch unsere IT-Experten und –Berater.

Schulungsthemen

- Softwarearchitektur (OOD)
- Requirements Engineering (OOA)
- Nebenläufige & verteilte Programmierung

Gerne konzipieren wir auch eine individuelle Schulung zu Ihren Fragestellungen.

Sprechen Sie uns an!
Tel. 0231/61 804-0, info@W3L.de

W3L-Akademie

In Zusammenarbeit mit der Fachhochschule Dortmund bieten wir

zwei Online-Studiengänge

- B.Sc. Web- und Medieninformatik
- B.Sc. Wirtschaftsinformatik

und 7 Weiterbildungen im IT-Bereich an.

Besuchen Sie unsere Akademie!
http://Akademie.W3L.de