Cool Things You Can Do With the Groovy Dynamic Language

Guillaume Alleon
Senior Researcher
EADS

Guillaume Laforge
Software Architect
OCTO Technology

Dierk König
Senior Architect
Canoo Eng. AG

http://groovy.codehaus.org

Session TS-1742

Overall Presentation Goal

Learn what's groovy about Groovy.

“A la carte …”

Groovy in the Java™ technology world

Groovy goodies

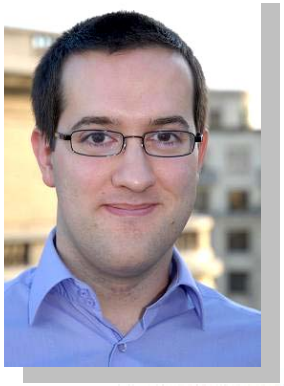
Groovy builders

Integrating Groovy

Speakers' Qualifications

- **Dierk König**
  - Committer to Groovy and Grails
  - Member of the Java Specification Request (JSR) 241 expert group
  - Author of "**Groovy in Action**"

- **Guillaume Alleon**
  - Committer to Groovy
  - Developer of **GroovySOAP**
- **Guillaume Laforge**
  - **Groovy Project Manager** and Grails initiator
  - JSR-241 Spec Lead standardizing the Groovy language
  - Co-Author of "Groovy in Action"



“A la carte …”

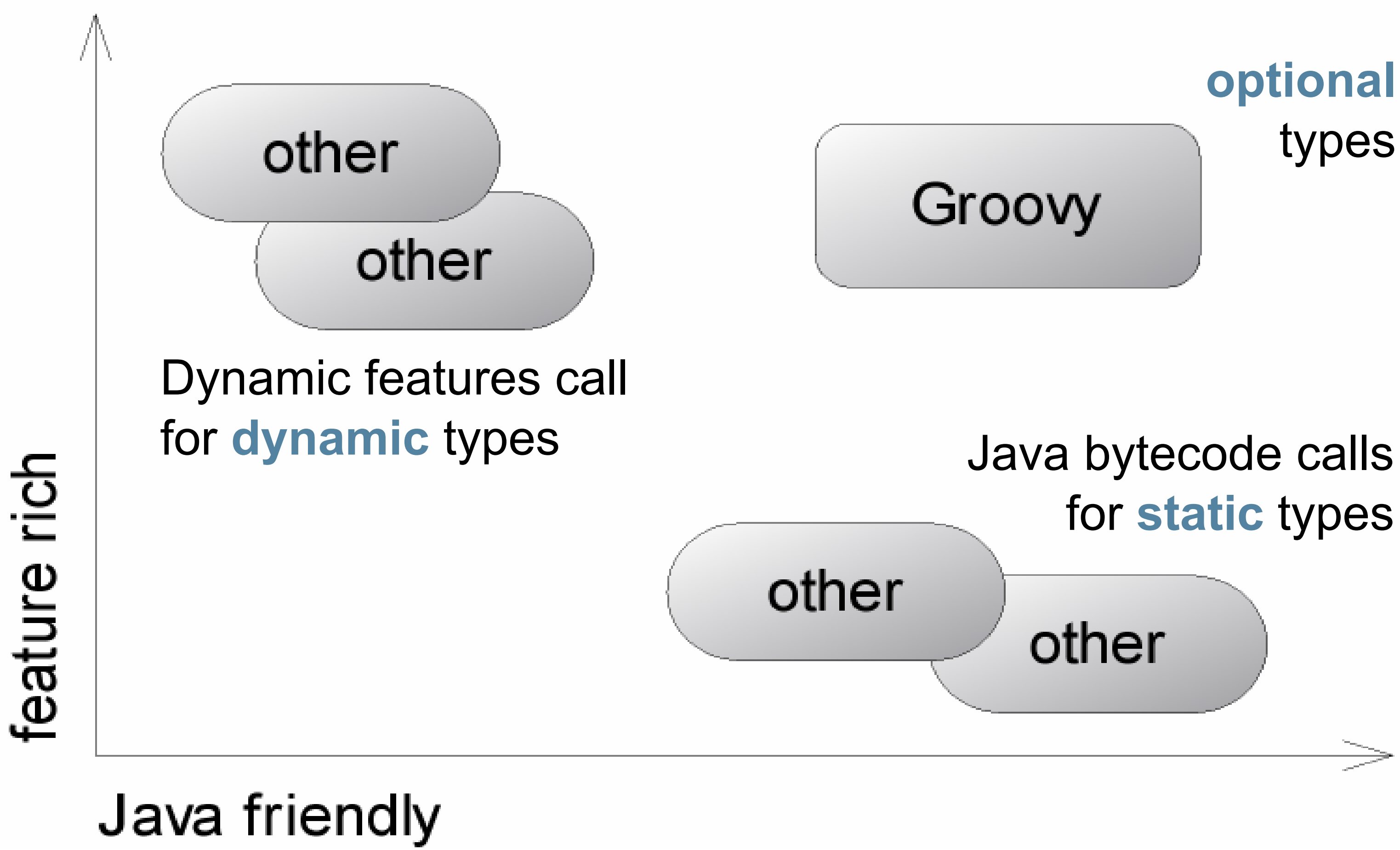
Groovy in the Java technology world

Groovy goodies

Groovy builders

Integrating Groovy

The Landscape of Java Virtual Machine (JVM™) Languages

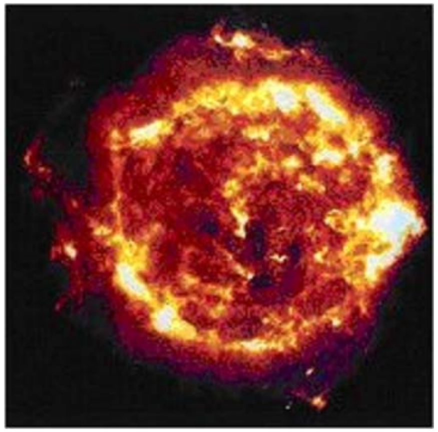


The terms “Java Virtual Machine” and “JVM” mean a Virtual Machine for the Java™ platform.

Usages of Scripting for Your Java Platform Project

Super glue

- Combine building blocks
- Groovy glues infrastructure and business logic together



Melted core

- Keep domain models adaptable
- Groovy supports smart configs and DSLs

Leverage Dynamic Language Features

- **Meta-Object Protocol** can change structure and behaviour at runtime
- Intercepting method calls
  - Compares to "AOP" or "Mixin"
  - Tracing and debugging
  - Mocks and stubs

- Enhance the Java Development Kit (JDK) with richer functionality (**GDK**)
- No more "*incomplete library smell*"
- Behaviour becomes injectable, even "non-intrusive"!
- Avoid duplication

Groovy Starter

```
System.out.println("Hello World!");
println 'Hello, World!'


def name = 'Guillaume'
println "$name, I get the car."
String longer = """ ${name}, the car
is in the next row."""


assert 0.5 == 1/2

def printSize(obj) {
     print obj?.size()
}
```

- optional ;
- GDK method

- Type optional
- GString
- java.lang.String

- BigDecimal equals

- Optional duck typing
- safe dereferencing

“A la carte …”

Groovy in the Java technology world

Groovy goodies

Groovy builders

Integrating Groovy

Java Platform’s Dynamic Friend

jax

INNOVATION AWARD 2007

1st Prize

Groovy

- **Flat learning curve**
  - **Leverage any Java library (JDK)**
  - **Syntax like Java (expressive++)**
  - **Java object and runtime model**
  - **Java architecture (threading, security, debugging)**
- Compiles down to **standard Java bytecode** precompiled or with runtime compilation
- **No impedance mismatch** with parent language!
- Supports “**duck typing**” and static typing
- Only dynamic language to **support annotations**

DEMO

@Transactional

Groovy Goodies Overview

- Fully **object** oriented
- **Closures** reusable and assignable pieces of code
- **Operators** can be overloaded
- Multimethods
- Literal declaration for **lists** (arrays), **maps**, **ranges** and **regular expressions**

- **GPath**: efficient object navigation
- **GroovyBeans**
- grep and switch
- **Templates**, **builder**, swing, Ant, **markup**, XML, SQL, XML-RPC, Scriptom, **Grails**, tests, Mocks

Regular Expressions

```
if ("Hello World!" =~ /Hello/)
if ("Hello World!" ==~ /Hello\b.*/)
~/Hello\b.*/




"1.23".replaceAll(/\d+/){ num ->
    num.toInteger() + 1
}

-> 2.24
```

- Find operator
- Match operator
- Pattern operator

- Replace with calculated values

Lists, Maps, Ranges

```
def empty = []
def full = [1, 2, 'J1']

assert full+full == full*2

assert full[0]   == 1
assert full[0..1]==[1, 2]

full[0..1]  = [0, 1, 2, 3]
assert full ==
        [0, 1, 2, 3, 'J1']
```

```
def empty = [:]
def full  = [a: 1, b: 2]

assert full['a'] == 1
assert full.a    == 1
full.a = 2
assert full == [a:2, b:2]

def inclusive = 'a'..'z'
inclusive.each {…}
def exclusive = 0..<10
```

Using Closures

```
3.times { println 'Hi' }

[0, 1, 2].each { number ->
    println number
}

[0, 1, 2].each { println it}

def printit = { println it }
[0, 1, 2].each printit
```

```
def houston(Closure doit) {
    (10..1).each { count ->
        doit(count)
    }
}
houston { println it }
```

```
new File('/x.txt').eachLine {
    println it
}
```

GroovyBeans and GPath

```
class Dir {
    String name
    List   dirs
}

def root =
new Dir (name: '/', dirs: [
    new Dir (name: 'a'),
    new Dir (name: 'b')
])

root.dirs[0].name == 'a'
```

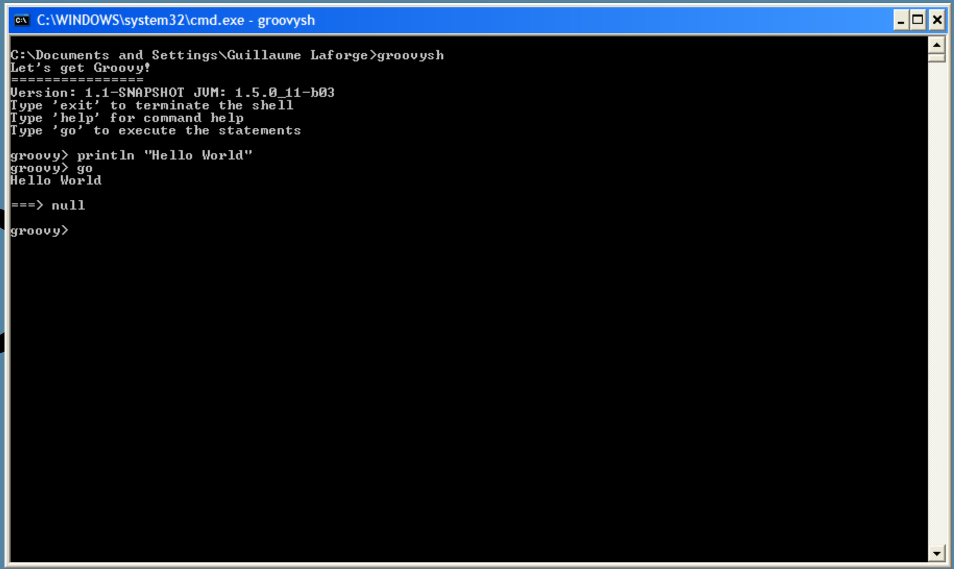
```
root.dirs.name ==
    ['a','b']

root.dirs.name*.size() ==
    [1, 1]

Dir.methods.name
     .grep(~/(g|s)et/)

-> [getName, setName, getDirs,
   setDirs, …]

find, findAll, grep,
every, any, …
```

Control Structures

```
if (1)
if (object)
if (list)

for (item in iterable) { }

myMap.each { key, value ->
    println "$key : $value"
}


throw, catch, finally
while, eachWithIndex,
  eachLine
```

```
switch (10) {
    case 0          : …; break
    case 0..9       : …
    case [8,9,11]   : …
    case Float      : …
    case {it%3 == 0}: …
    case ~/../      : …
    default :
}


implement
  boolean isCase(candidate)
```

DEMO



```
C:\Documents and Settings\Guillaume Laforge>groovysh
Let's get Groovy!
================
Version: 1.1-SNAPSHOT JVM: 1.5.0_11-b03
Type 'exit' to terminate the shell
Type 'help' for command help
Type 'go' to execute the statements

groovy> println "Hello World"
groovy> go
Hello World

===> null

groovy>
```

“A la carte …”

Groovy Java technology context

Groovy goodies

Groovy builders

Integrating Groovy

Builder Pattern Inclusive

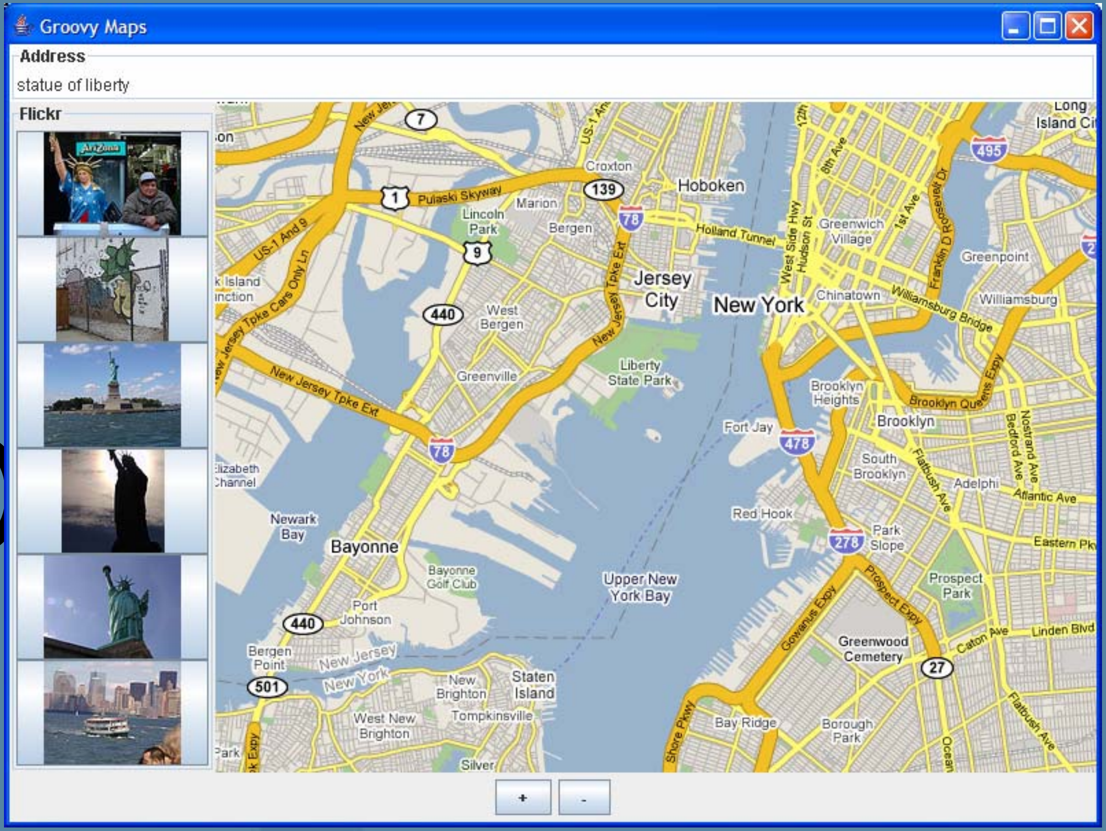
- Builder pattern from the GoF at the syntax-level
- Represents easily any **nested tree-structured data**

```
import groovy.xml.*
def page = new MarkupBuilder()
page.html {
  head { title 'Hello' }
  body {
    ul {
      for (count in 1..10) {
        li "world $count"
} } } }
```

NodeBuilder, DomBuilder, SwingBuilder, AntBuilder, …

- Create new builder
- Call pretended methods (html, head, ...)
- Arguments are Closures
- Builder code looks very declarative but is ordinary Groovy program code and can contain any kind of logic

DEMO



Address

statue of liberty

“A la carte …”

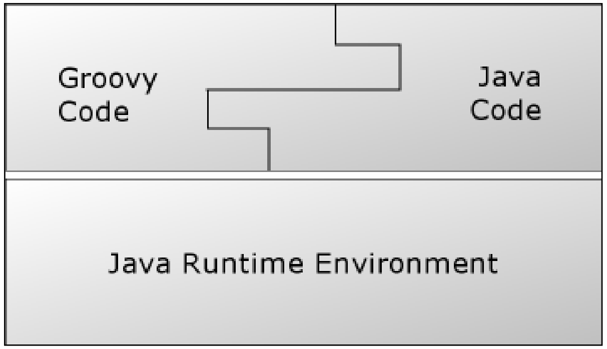
Groovy in the Java technology world

Groovy goodies

Groovy builders

Integrating Groovy

Integration Options



- Groovy integrates with Java platform on the bytecode level. They cannot only call each other - they are the same thing: java.lang.Class objects.
- 1) Compile with groovyc
- 2) Evaluation at runtime
  - Eval.x (users, "x.grep{ it.salary > 100000 }.address.town");
  - GroovyShell, GroovyScriptEngine, GroovyClassLoader, Spring Beans
  - Bean Scripting Framework / JSR-223 (Java platform 6)

Integration With Existing Native Apps

- Scriptom allows you to script any ActiveX or COM Windows component from within your Groovy scripts

```
import org.codehaus.groovy.scriptom.ActiveXProxy

def outlook = new ActiveXProxy("Outlook.Application")
def message = outlook.CreateItem(0)
def emails = "galleon@codehaus.org;glaforge@codehaus.org"
def rec = message.Recipients.add(emails)
rec.Type = 1
message.Display(true)
```

Integration With Existing Services

- WS or XML/RPC allow seamless connection to existing services …

```
import groovy.net.soap.SoapClient
proxy = new SoapClient(
   "http://www.webservicex.net/CurrencyConvertor.asmx?WSDL")
rate = proxy.ConversionRate("USD", "EUR")
println rate
```

  - No need to generate stubs,
  - Complex types are supported

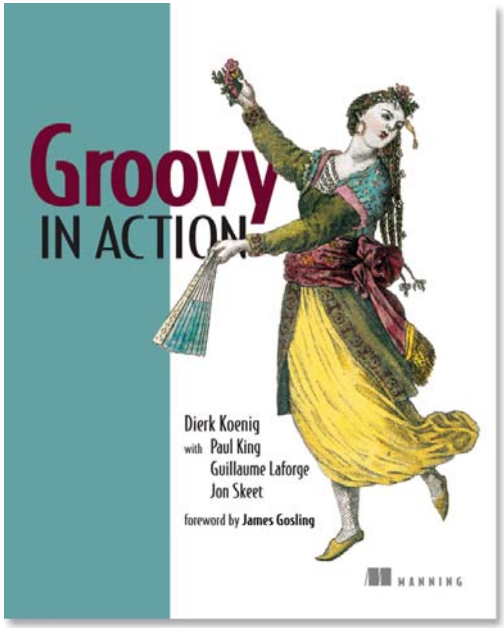
Groovy Users

- Financial industry, USA / UK **Fortune 500** supply chain, biochemistry, EADS, Siemens, ...
- Education www.knowledgepool.com
- Pepsi, UUZone, www.pj-smoothies.co.uk
- **Oracle** clustering OC4J
- OSS: **Spring**, RIFE, NanoContainer, Drools, SnipSnap, XWiki, eXoPlatform, Turbine Fulcrum, Groovestry, Luxor XUL, Canoo WebTest, ...

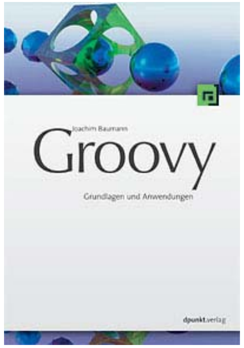
Toolbox

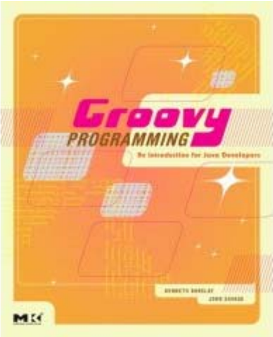
- Genuine Java support for profiling, debugging, code coverage, etc.
- IDE Integrations
  - Eclipse: 4 developers
  - IntelliJ IDEA: JetBrains
  - NetBeans: Sun Microsystems
- Smart editors
  Emacs, Vim, UltraEdit, Textmate, ...

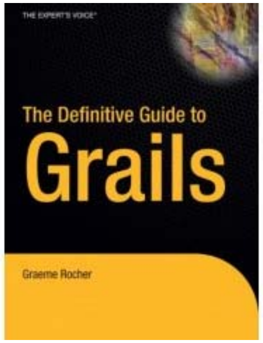
Further Information

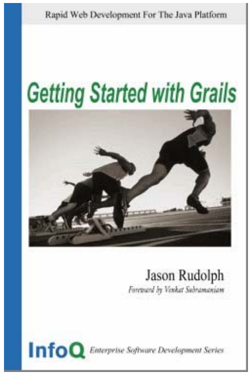


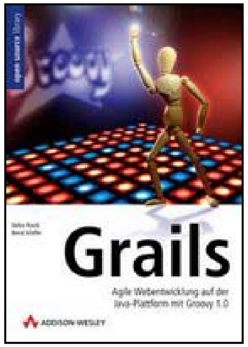
- Web resources
  - http://groovy.codehaus.org
  - http://grails.org
- **Groovy in Action**
  Manning, 2007
  Foreword by **James Gosling**
  König with Glover, Laforge, King, Skeet








Rapid Web Development For The Java Platform
Getting Started with Grails
Jason Rudolph
Foreword by Venkat Subramaniam
InfoQ
Enterprise Software Development Series



Groovy Recipes

Pragmatic Bookshelf

Groovy Summary

- Be feature rich
- Be Java technology friendly
- Be groovy

Q&A

Cool Things You Can Do With the Groovy Dynamic Language

Guillaume Alleon

Senior Researcher

EADS

Guillaume Laforge
Software Architect
OCTO Technology

Dierk König
Senior Architect
Canoo Eng. AG

http://groovy.codehaus.org

Session TS-1742