Groovy-based Domain Specific Languages

Václav Pech

http://jroller.com/vaclav

http://www.vaclavpech.eu

@vaclav_pech

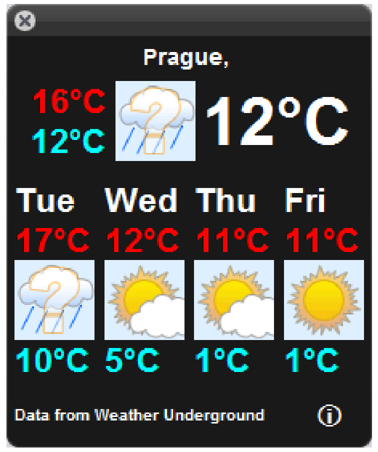
Agenda

- Domain-specific languages
- Builders
- Static meta-programming
- DSL frameworks – Grails, Griffon

Griffon

Rich-client applications on JVM

- Swing
- Groovy DSLs
- Scaffolding
- Convention over configuration
- DRY
- KISS



Prague,

16°C 12°C 12°C

Tue	Wed	Thu	Fri
17°C	12°C	11°C	11°C
10°C	5°C	1°C	1°C

Data from Weather Underground

Properties

```
class ProgrammingLanguage {
    String name
    String version
    boolean easy=true
}
def groovy=new ProgrammingLanguage(
        name:'Groovy', version:'1.5', easy:true)

def java=new ProgrammingLanguage(name:'Java')
java.version='1.6'
```

Closures

```
Closure multiply1 = {int a, int b -> return a * b}

Closure multiply2 = {int a, int b -> a * b}

Closure multiply3 = {a, b -> a * b}

def multiply4 = {a, b -> a * b}
```

Closures – implicit parameter

```
def triple1 = {int number -> number * 3}

def triple2 = {number -> number * 3}

def triple3 = {it * 3}
```

Groovy is functional

```
def multiply = {a, b -> a * b}
def double = multiply.curry(2)
def triple = multiply.curry(3)

assert 4 == multiply(2, 2)
assert 8 == double(4)
assert 6 == triple(2)
```

Collections

```
final emptyList = []
final list = [1, 2, 3, 4, 5]
final emptyMap = [:]
final capitals = [cz : 'Prague', uk : 'London']


final list = [1, 2, 3, 4, 5] as LinkedList
final emptyMap = [:] as ConcurrentHashMap
```

Scripting

Evaluate custom Groovy code

At run-time!!!

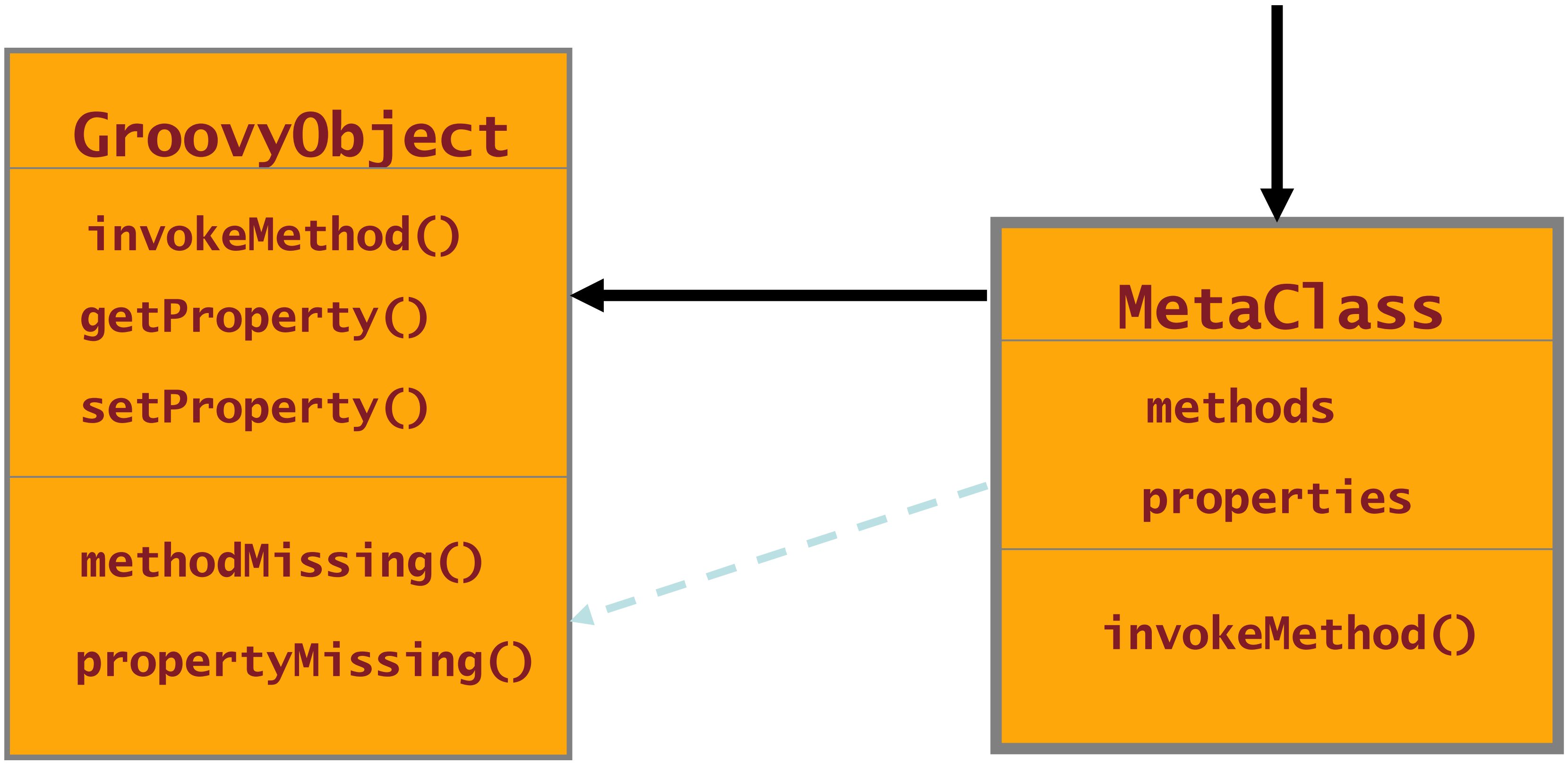
```
new GroovyShell().evaluate('println Hi!')
```

Categories

```
StringUtils.countMatches(myString, 'Groovy')
```

```
use(StringUtils) {
    myString.countMatches('Groovy')
}
```

Dynamic method invocation



Querying objects' methods

```
o.respondsTo()
o.hasProperty()
o.metaClass.getMetaMethod(name, args)
o.metaClass.getMetaProperty(name)
```

DSL

- Limited purpose language
- Targeted to a particular domain
- Friendlier API to a framework
  - External
    - SQL, HTML, CSS, …
  - Internal

DSL – Account manipulation

```
Money money = new Money(amount: 350, currency: 'eur')
getAccount('Account1').withDraw money
getAccount('Account3').deposit money
```

```
"Account1" >> 350.eur >> "Account3"
```

order cake with plums and apples
and cream to "Malostranske namesti"

```
order(cake).with(plums).and(apples)
.and(cream).to("Malostranske namesti")
```

Builders

- Construct hierarchies

```
xml.records() {
    order(id: 'PL19826714', date: '21-01-2008') {
        item(quantity: 10) {
            product(id: '76327')
            price(base: 100) {
                volumeDiscount(value: 5)
            }
        }
```

Builders - GAnt

```
ant.sequential {
    myDir = "target/AntTest/"
    mkdir(dir: myDir)
    copy(todir: myDir) {
        fileset(dir: "src/test") {
            include(name: "**/*.groovy")
        }
    }
    List dirs = ['core', 'lib', 'engine', 'gui', 'db']
    for(String currentDir:dirs) {
        String targetDir="target/$currentDir"
        mkdir(dir:targetDir)
```

Builders – Spring config

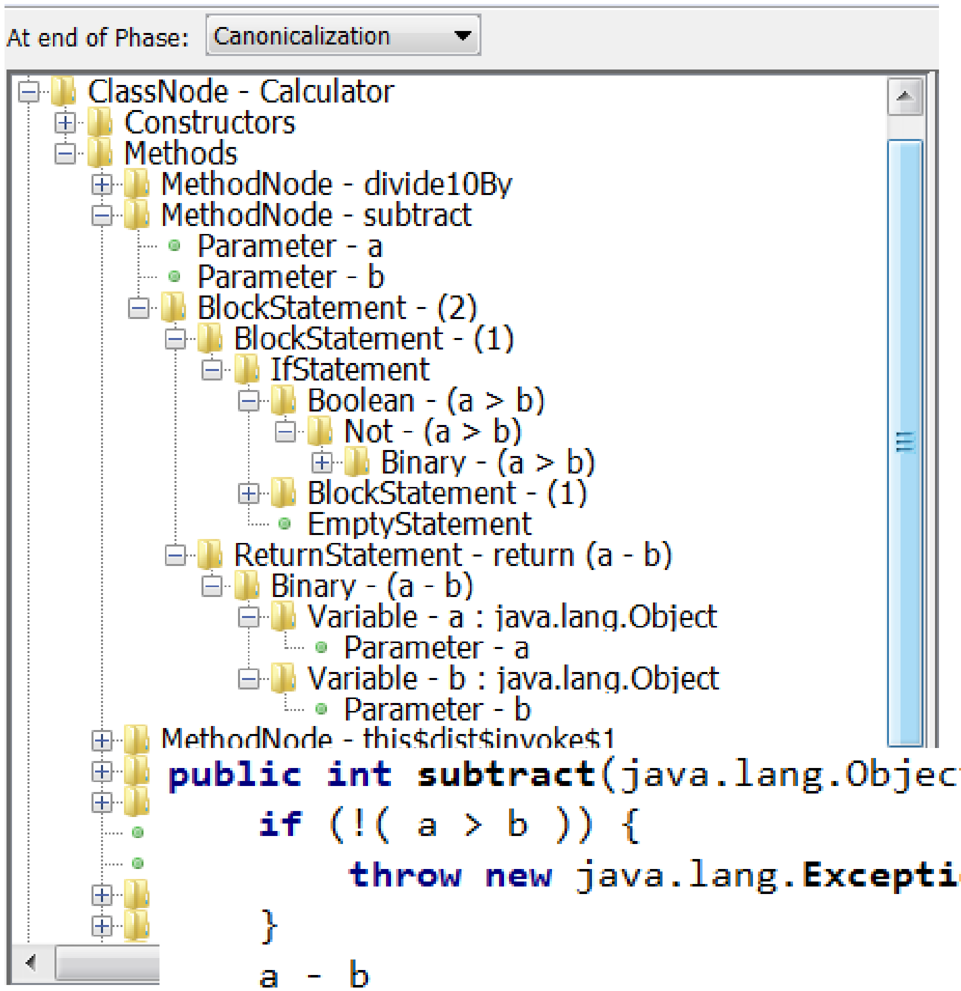
```
dataSource(BasicDataSource) {
    driverClassName = "org.hsqldb.jdbcDriver"
    url = "jdbc:hsqldb:mem:shopDB"
}

sessionFactory(ConfigurableLocalSessionFactoryBean) {
    dataSource = dataSource
    hibernateProperties = ["hibernate.hbm2ddl.auto": "create-drop",
            "hibernate.show_sql": true]
}

calculator(demo.shop.CalculatorImpl) {bean ->
    bean.singleton = true
    bean.autowire = 'byType'
}
```

BDD - Spock

```
class DataDriven extends Specification {
    def "maximum of two numbers"() {
        expect:
        Math.max(a, b) == c
        where:
        a << [7, 4, 9]
        b << [3, 5, 9]
        c << [7, 5, 9]
 } }
```

http://code.google.com/p/spock/

AST



At end of Phase: Canonicalization

```
ClassNode - Calculator
  Constructors
  Methods
    MethodNode - divide10By
    MethodNode - subtract
      Parameter - a
      Parameter - b
      BlockStatement - (2)
        BlockStatement - (1)
          IfStatement
            Boolean - (a > b)
              Not - (a > b)
                Binary - (a > b)
            BlockStatement - (1)
            EmptyStatement
        ReturnStatement - return (a - b)
          Binary - (a - b)
            Variable - a : java.lang.Object
              Parameter - a
            Variable - b : java.lang.Object
              Parameter - b
    MethodNode - this$dist$invoke$1
```

```
public int subtract(java.lang.Object a, java.lang.Object b) {
    if (!( a > b )) {
        throw new java.lang.Exception('Precondition violated: {a > b}')
    }
    a - b
}
```

AST Transformations

```
class Registrations {
    @Delegate List items = []
}

def people = new Registrations()
people.addAll(["Joe", "Dave"])
assert ["Dave", "Joe"] == people.reverse()
```

@Delegate, @Immutable, @Singleton

@Lazy

@TupleConstructor

@InheritConstructors

@Canonical

@ToString

@EqualsAndHashCode

@Log, @Log4j, @Commons

@Synchronized

@WithReadLock

@WithWriteLock

@AutoClone, @AutoExternalize

...

Creating AST Transformations

```
new AstBuilder()
                    .buildFromString()
                    .buildFromCode()
                    .buildFromSpec()
```

```
.buildFromString ("""
                    Integer.parseInt("$param")
""")
```

```
.buildFromCode (
                    Integer.parseInt("$param")
)
```

```
.buildFromSpec {
   method('convertToNumber', ACC_PUBLIC, Integer) {
            parameters { parameter 'parameter': String.class }
            exceptions {}
            block {
               returnStatement {
                  staticMethodCall(Integer, "parseInt") {
                     argumentList {
                        variable "parameter"
                     }
      }  }  }  }  }
```

Grails

Web applications on JVM

- Hibernate, Spring, ...
- Groovy DSLs
- Scaffolding
- Convention over configuration
- DRY
- KISS

Summary

Lots of power and fun for Java programmers

http://jroller.com/vaclav
pech@d3s.mff.cuni.cz

References

http://glaforge.appspot.com/article/groovy-ast-transformations-tutorials

http://www.groovy.cz

http://groovy.codehaus.org

http://grails.org

http://groovyconsole.appspot.com/