

canoo

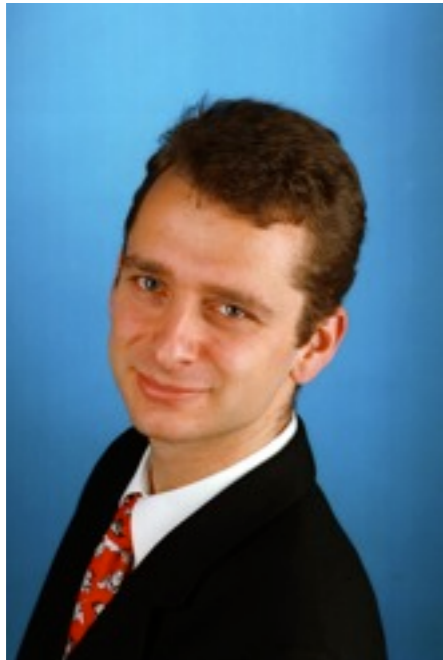
› your provider for business web solutions ›



Groovy Seven Usage Patterns

Dierk König

dierk.koenig@canoo.com
@mittie



Dierk König

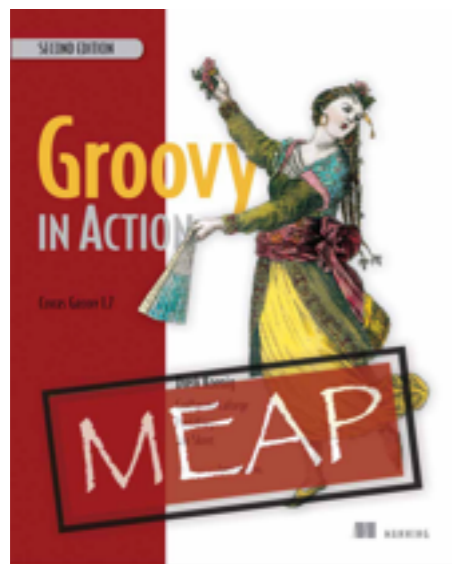
Fellow @ Canoo Engineering AG, Basel (CH)

Canoo RIA Suite

Projects, Consulting

www.canoo.com

Open-source committer Groovy, Grails, GParas, Grasp



The seven usage patterns

Super Glue

Liquid Heart

Lipstick

Keyhole Surgery

Unlimited Openness

Ghost Writer

House Elf

#1 Super Glue



- Build application from existing building blocks
- Java makes perfect infrastructure: middleware, frameworks, widget sets, services
- Scripting makes a flexible (agile) application layer: views und controller
- Grails, Gaelyk, Griffon, Ratpack, JBoss Seam, WebWork, Struts 2 Actions, Swing, ...
- Example: combination of XML parser, Java networking and Swing widget set in order to build a standard RSS feed reader

```
def url      ='http://www.groovyblogs.org/feed/rss'
def items   = new XmlParser().parse(url).channel.item
def cols     = 'pubDate title description'.tokenize()

groovy.swing.SwingBuilder.build {
  frame(id:'f', title: 'Groovy Blogs', visible:true) {
    scrollPane {
      table {
        tableModel(list: items) {
          cols.each { col ->
            closureColumn header: col, read: { it[col].text() }
          }
        }
      }
    }
  }
  f.pack()
}
```

#2 Liquid Heart



- Externalize business models
- Given application structure in Java
- Allow to learn about the business:
keep entities, relations, and behaviour flexible through scripting
- Usages: Spring beans, JBoss components, rule engines (groovyrules.dev.java.net, JSR-94), Grails, enterprise risk management, smart configuration
- Example: calculation rules for bonus allocation

Liquid Heart example: bonus allocation

```
revenue = employee.revenue
switch(revenue / 1000) {
  case 0..100 : return revenue * 0.04
  case 100..200 : return revenue * 0.05
  case {it > 200} : bonusClub.add(employee)
                  return revenue * 0.06
}
```



```
Binding binding = new Binding();
binding.setVariable("employee", employee);
binding.setVariable("bonusClub", bonusClub);
GroovyShell shell = new GroovyShell(binding);
File script = new File(filename);
float bonus = (float) shell.evaluate(script);
```



#3 Lipstick



- Make code read nicely
- Fluent transition from APIs to DSLs
- Usages: GDK, Groovy Beans, GPath, Lists, Maps, Callable, XML, SQL, Command chains, GPars, Scriptom, and pretty much all of Groovy...
- Example: calculation rules for bonus allocation (again)

```
def invite(subject) {  
  [to : { target -> target.add(subject) } ]  
}
```

```
invite employee to bonusClub
```



#4 Keyhole Surgery

- Minimal-invasive surgery "in vivo"
- Runtime inspection and modifications
- Ad-hoc queries are not foreseeable
- "Backdoor" for the live execution of scripts
- Particularly useful for product support, failure analysis, hot fixes, emergencies
- Usages: Oracle JMX Beans, XWiki, SnipSnap, Ant, Canoo WebTest, Grails Console, GAE, ULC Admin Console
- Example: a live Groovy Servlet



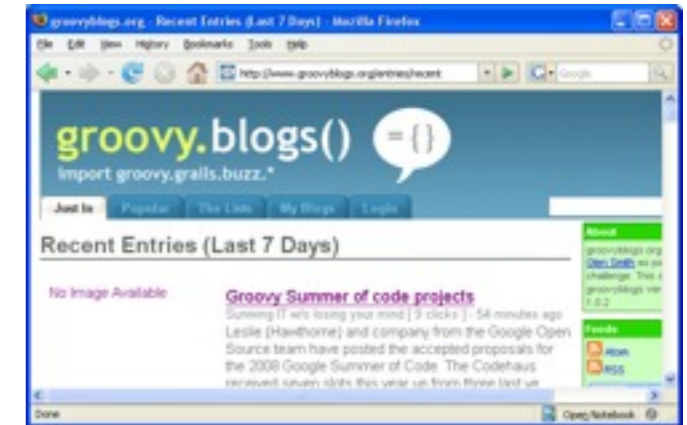
Remove malicious users from the servlet context

```
users = servletContext.getAttribute('users')
bad    = users.findAll { user -> user.cart.items.any { it.price < 0 } }
servletContext.setAttribute('users', users - bad)
```



#5 Unlimited Openness

- Every line of code may be changed
- It's impossible (and not desirable!) to design for every possible future requirement
- Allow for easily changing the code without tedious setup for compilation and deployment
- Follow the lessons of PERL, PHP, Python, ..
- Example: groovyblogs.org, miki



#6 Ghost Writer

- Code generation without source code generation
- Avoid repeating code constructs but still make them available in the bytecode
- Mark the code with annotations and provide an AST transformation or use existing ones
- Examples: `@Immutable`, `@Log`, `@Delegate`, `@Lazy`, `@Singleton`, `@Bindable`, `@AutoClone`, `@TimedInterrupt`, ...

#7 House Elf



- ◉ Delegate the housework
- ◉ Build automation, continuous integration, deployment, installer, service monitoring, reports, statistics, automated documentation, functional tests, HTML scraping, Web remote control, XML-RPC, WebServices
- ◉ Usages with Ant, Maven, AntBuilder, Gant, Gradle, Canoo WebTest, Grails scaffolding, ...
- ◉ Examples: `senv`, `groovy install`

The seven usage patterns

Super Glue

Liquid Heart

Lipstick

Keyhole Surgery

Unlimited Openness

Ghost Writer

House Elf

Discussion

dierk.koenig@canoo.com
@mittie



credits:
Paul King