



**IPROFS**

---

**Scala / Lift**

– Peter van Rijn

- Functioneel
- Object oriented
- Op JVM
  - Java bibliotheken beschikbaar
- Martin Odersky
  - javac 1.1 – 1.4 en 1.5 generics

- **Functionies als bouwstenen**
  - Embedded functionies
  - functie values
  - higher order functionies
  - DSL

- Academisch
- Concurrent is “killer app”
- Google
  - Map/Reduce, BigTable

### •Verdwijnen

- State
  - Geen variabelen
- while, for , if
  - Horen bij imperatief
- $\frac{3}{4}$  van de design pattern

- Alles is een object
  - Geen primitieve datatypen
  - Ook functies zijn objecten
- Mixins
  - Multiple inheritance zonder nadelen
- Pattern matching op objecten
  - Combinatie F en OO

- Verdwijnen
  - State
  - Interfaces
  - API's
- Workflow is een set transformaties
  - Een chain van functies

- OOAD werkt niet
  - Voor EE en webapplicaties
  - Wel goed voor standaard bibliotheken
- Objecten zijn vloeibaar
  - Bedenk een User klasse met 20 velden in een groep van 20 man
  - Objecten gaan niet over beschrijving
    - Syntax
  - Objecten gaan over betekenis, rol
    - Semantiek
- Gevolg: Meeste programmeerwerk is integratie i.p.v. functionaliteit

- OO is verworden tot state oriented
  - Data klassen
    - oplossing key/value
    - Iterate, filter, map
      - functioneel
- API's zijn autistisch
  - Applicaties zijn een éénrichting API stack
    - Oplossing: software componenten met tweezijdige communicatie

- David Pollak
  - ervaring met ruby, java
  - Ideeen uit: Rails, Wicket, Django
- <http://liftweb.net/>
- Lift 1.0 februari 2009

## •XML in Scala

```
val xml1 = <span>een tekst</span>
```

```
val xml2 = <span> {3 + 4} </span>
```

```
val xml3 = <a><b><c>d</c></b></a> \ "b"
```

- Layout
  - vlakverdeling
- Dynamic html
- Binding
  - form
- Message
  - foutmeldingen
- Session
- Model/DAO
- Validation
- Scaffolding
  - Forms, view
- menu/login
- ajax/comet

```
<lift:helloWorld.howdy />
```

```
class HelloWorld {  
  def howdy = <span>Welcome to me at {new Date}</span>  
}
```

```
<lift:MyTest.in>  
You are logged in  
</lift:MyTest.in>
```

```
<lift:MyTest.out>  
Please register  
</lift:MyTest.out>
```

```
class MyTest {  
  val loggedIn = true  
  
  def in ( content : NodeSeq ) :NodeSeq = {  
    if (loggedIn) content else NodeSeq.Empty  
  }  
  def out ( content : NodeSeq ) :NodeSeq = {  
    if (!loggedIn) content else NodeSeq.Empty  
  }  
}
```

```
</lift:boekUtil.add form="POST">
  <boek:title/> <br/>
  <boek:author/> <br/>
  <boek:submit />
</lift:boekUtil.add>
```

```
class BoekUtil {
  var title = "tt"
  var author = "aa"

  def add (content : NodeSeq) : NodeSeq = {
    def addAction() = { }

    bind ("boek", content ,
      "title" -> SHtml.text(title, title = _),
      "author" -> SHtml.text(author, author = _),
      "submit" -> SHtml.submit("Add", addAction)
    )
  } }
}
```

```
<lift:Msgs />
```

```
<lift:Msg id="titleError" />
```

```
S.warning("_warning")
```

```
S.notice("clicked")
```

```
S.error("titleError", "dit is de titel" )
```

```
<lift:embed what="menu" />

<lift:bind name="content" />

<lift:surround with="main" at="content">
  <head><script ... /></head>
  <h2>Welcome to your project!</h2>
</lift:surround>
```

- Stateless by default
  - Nieuwe snippet instance bij iedere request
- Stateful
  - RequestVar
  - SessionVar
  - StatefulSnippet

```
class BoekUtil {  
  object title extends SessionVar[String] ("tt")  
  object author extends RequestVar[String] ("aa")  
  
  def add (content : NodeSeq) : NodeSeq = {  
    bind ("boek", content ,  
      "title" -> SHtml.text(title.is, title(_)),  
      "author" -> SHtml.text(author.is, author(_)),  
    )  
  }  
}
```

```
class BoekUtil extends StatefulSnippet {  
  def dispatch : DispatchIt = {  
    case "add" => add _  
  }  
  
  var title = "tt"  
  var author = "aa"  
  
  def add (content : NodeSeq) : NodeSeq = {}  
}
```

```
class Boek extends LongKeyedMapper[Boek] with IdPK {  
  def getSingleton = Boek  
  object titel extends MappedString(this, 128)  
  object auteur extends MappedString(this, 128)  
  object isbn extends MappedInt(this)  
  
  object Boek extends Boek with LongKeyedMetaMapper[Boek]  
}
```

```
// in Boot  
Schemifier.schemify(true, Log.infoF _, Boek)
```

- create
- save
- delete
- count
- FindAll
- findAllByInsecureSQL
- findAllByPreparedStatement

```
object titel extends MappedString(this, 128) {  
  override def validations = checkTitle _ :: Nil  
  
  def checkTitle (title : String) = {  
    if (title.length < 3 ) { ... }  
  }  
}
```

- asHtml
- asJSON
- asJs
- toForm

- Menu en User/Login als componenten

```
// in Boot.scala
val entries =
  Menu(Loc("Home", List("index"), "Home")) :: User.sitemap

LiftRules.setSiteMap(SiteMap(entries:_*))

class User extends MegaProtoUser[User]
```

```
class AjaxSnippit {  
  
  def myFunc(html: NodeSeq) : NodeSeq = {  
    bind("hello", html,  
      "button" -> SHtml.ajaxButton(Text("Press me"), {() =>  
        println("Got an Ajax call")  
        SetHtml("my-div", Text("That's it"))  
      })))  
  }  
}
```

```
<lift:comet type="Clock" name="Other">  
  Current Time: <clk:time>Missing Clock</clk:time>  
</lift:comet>
```

```
class Clock extends CometActor {  
  override def defaultPrefix = Full("clk")  
  def render = bind("time" -> timeSpan)  
  def timeSpan = (<span id="time">{now}</span>)  
  
  ActorPing.schedule(this, Tick, 10000L)  
  
  override def lowPriority : PartialFunction[Any, Unit] = {  
    case Tick => {  
      println("Got tick " + new Date());  
      partialUpdate(SetHtml("time", Text(now.toString)))  
      ActorPing.schedule(this, Tick, 10000L)  
    } } }  
  
  case object Tick
```

```
bind("hello", html,  
    "submit" -> SHtml.submit ("Save", "onclick" ->  
  
        JsIf (JsEq (ValById (name), JsNull ),  
  
            Alert (" You must provide a name" )  
  
            & JsReturn (false))
```

- Scala is de beste taal op dit moment
  - Het ijkpunt voor andere talen
    - C# generics
    - Python compile time type checking
- Java is vastgelopen
  - Geen benevolent dictator for life
    - Design by committee
- Lift de laatste in voorschrijdend inzicht
  - Zal zeker ingehaald worden

4.

- Universiteiten gaan naar Scala
  - Onderwijskundig een betere taal dan Java
- Seniors stappen over naar Scala
  - En zijn dan niet meer beschikbaar voor Java
- OSS/Apache projecten gaan naar Scala
  - Geheel of gedeeltelijk
- Kritische massa
  - Ik schat over 1.5 jaar
  - En dan gaat het hard