

Scala: An OO Surprise

Apostolos Syropoulos
Xanthi, Greece

4th European Lisp Symposium
March 31st - April 1st, 2011
TUHH, Hamburg University of Technology,
Hamburg, Germany

Outline

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- 1 Introduction
 - What is Scala?
 - Object Orientation
 - Scalability
- 2 Functional Features
- 3 Parser Builders
- 4 XML Processing
- 5 XML Processing
- 6 Concurrent Programming

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

A versatile programming language

- A purely object-oriented language.

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- A purely object-oriented language.
- **Runs on the Java Virtual Machine.**

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.

A versatile programming language

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.
- Has features found in functional programming languages, but it is not a functional language!

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.
- Has features found in functional programming languages, but it is not a functional language!
- **Allows concurrent programming.**

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.
- Has features found in functional programming languages, but it is not a functional language!
- Allows concurrent programming.

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.
- Has features found in functional programming languages, but it is not a functional language!
- Allows concurrent programming.
- Includes facilities for GUI programming.

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.
- Has features found in functional programming languages, but it is not a functional language!
- Allows concurrent programming.
- Includes facilities for GUI programming.
- **Seamless integration of XML.**

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.
- Has features found in functional programming languages, but it is not a functional language!
- Allows concurrent programming.
- Includes facilities for GUI programming.
- Seamless integration of XML.

A versatile programming language

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- A purely object-oriented language.
- Runs on the Java Virtual Machine.
- A scalable language.
- Has features found in functional programming languages, but it is not a functional language!
- Allows concurrent programming.
- Includes facilities for GUI programming.
- Seamless integration of XML.
- **Provides parser combinators.**

Principles of Object Orientation

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Principles of Object Orientation

- Abstraction (i.e., a software object is an idealization of a real world object).

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Principles of Object Orientation

- Abstraction (i.e., a software object is an idealization of a real world object).
- Encapsulation (i.e., a *data protection mechanism*).

Scala: An
OO Surprise

Syropoulos

Introduction
What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Principles of Object Orientation

- Abstraction (i.e., a software object is an idealization of a real world object).
- Encapsulation (i.e., a *data protection* mechanism).
- Inheritance (i.e., an object may *extend* the functionality of another object).

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

Principles of Object Orientation

- Abstraction (i.e., a software object is an idealization of a real world object).
- Encapsulation (i.e., a *data protection* mechanism).
- Inheritance (i.e., an object may *extend* the functionality of another object).
- Polymorphism (i.e., software modules may have different instances with identical behavior).

Scala: An
OO Surprise

Syropoulos

Introduction
What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

What makes a programming language scalable?

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

What makes a programming language scalable?

- Any language that can be extended seamlessly is *scalable*.

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

What makes a programming language scalable?

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Any language that can be extended seamlessly is *scalable*.
- Gyu L. Steel Jr.: “[A] language design can no longer be a thing. It must be a pattern—a pattern for growth—a pattern for growing the pattern for defining the patterns that programmers can use for their real work and their main goal.”

What makes a programming language scalable?

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Any language that can be extended seamlessly is *scalable*.
- Gyu L. Steel Jr.: “[A] language design can no longer be a thing. It must be a pattern—a pattern for growth—a pattern for growing the pattern for defining the patterns that programmers can use for their real work and their main goal.”

What makes a programming language scalable?

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Any language that can be extended seamlessly is *scalable*.
- Gyu L. Steel Jr.: “[A] language design can no longer be a thing. It must be a pattern—a pattern for growth—a pattern for growing the pattern for defining the patterns that programmers can use for their real work and their main goal.”
- There two forms of *growth*: variable vocabulary or variable semantics.

What makes a programming language scalable?

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Any language that can be extended seamlessly is *scalable*.
- Gyu L. Steel Jr.: “[A] language design can no longer be a thing. It must be a pattern—a pattern for growth—a pattern for growing the pattern for defining the patterns that programmers can use for their real work and their main goal.”
- There two forms of *growth*: variable vocabulary or variable semantics.
- **Generic types, operator overloading, and user-defined light-weight types form the kernel of every scalable language.**

A Generic Stack

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

A Generic Stack

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

```
class Stack [γ] (n: Int) {  
  private var S = new Array[γ](n)  
  private var top = 0;  
  def push(elem: γ) {  
    top = top + 1  
    S(top) = elem  
  }  
  def pop () : γ = {  
    var oldtop = top  
    top = top - 1  
    S(oldtop)  
  }  
}
```

User-Defined Light-Weight Complex Numbers

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

User-Defined Light-Weight Complex Numbers

Scala: An OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

```
class Complex(val re: Double, val im: Double) {
  def + (x: Complex) =
    new Complex(re + x.re, im + x.im)
  def unary_- = new Complex(-Re, -Im)
  . . . . .
  override def toString = if(re == 0 && im == 0) "0" ...
}
object Complex{
  def apply(re: Double)(im: Double) =
    new Complex(re, im)
}
object i extends Complex(0.0, 1.0)
. . . . .
var a = 5 + 3.0*i // a complex number!!!
```

Call-By-Name or how to define new commands

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Call-By-Name or how to define new commands

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

```
def loop(pre: => Unit)(cond: => Boolean)
      (post: => Unit): Unit = {
  pre
  if (!cond) {
    post
    loop(pre)(cond)(post)
  }
  else
    ()
}

var x=6; var y=0
loop {
  x=x-1; y=y+1
} (x == 0 ) {
  println(x); println(y)
}
```

Scala's Functional Features

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

**Functional
Features**

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Scala's Functional Features

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

**Functional
Features**

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Functions are software modules (*objects* in Scala's parlance).

Scala's Functional Features

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Functions are software modules (*objects* in Scala's parlance).
- Users can define anonymous functions.

Scala's Functional Features

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Functions are software modules (*objects* in Scala's parlance).
- Users can define anonymous functions.

```
var inc = (x:Int) => x+1
```

Scala's Functional Features

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Functions are software modules (*objects* in Scala's parlance).
- Users can define anonymous functions.

```
var inc = (x:Int) => x+1
```

- Supports closures.

Scala's Functional Features

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Functions are software modules (*objects* in Scala's parlance).
- Users can define anonymous functions.

```
var inc = (x:Int) => x+1
```

- Supports closures.

```
var mul = (x: Int) => (y:Int) => x*y  
var mul3 = mul(3)
```

Scala's Functional Features

Scala: An
OO Surprise

Syropoulos

Introduction
What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Functions are software modules (*objects* in Scala's parlance).
- Users can define anonymous functions.

```
var inc = (x:Int) => x+1
```

- Supports closures.

```
var mul = (x: Int) => (y:Int) => x*y  
var mul3 = mul(3)
```

- Pattern matching.

Data Structures with Class Hierarchies

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

**Functional
Features**

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Data Structures with Class Hierarchies

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

**Functional
Features**

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Haskell binary tree

Data Structures with Class Hierarchies

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

**Functional
Features**

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Haskell binary tree

```
datatype BinTree = Empty |  
                  Node(Int, BinTree, BinTree)
```

Data Structures with Class Hierarchies

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Haskell binary tree

```
datatype BinTree = Empty |  
                  Node(Int, BinTree, BinTree)
```

- Scala binary tree

Data Structures with Class Hierarchies

Scala: An
OO Surprise

Syropoulos

Introduction
What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Haskell binary tree

```
datatype BinTree = Empty |  
                  Node(Int, BinTree, BinTree)
```

- Scala binary tree

```
abstract class BinTree  
case object EmptyTree extends BinTree  
case class Node(elem : Int,  
                left : BinTree,  
                right: BinTree) extends BinTree
```

Pattern Matching in Action

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

**Functional
Features**

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Pattern Matching in Action

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

**Functional
Features**

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Depth of a binary tree

Pattern Matching in Action

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Depth of a binary tree

```
def depth(t: BinTree): Int = {  
  t match {  
    case EmptyTree => 0  
    case Node(_,EmptyTree,r) => 1 + depth(r)  
    case Node(_,l,EmptyTree) => 1 + depth(l)  
    case Node(_,l,r) => Math.max(depth(l),  
                                  depth(r)) + 1  
  }  
}
```

Constructing Parsers with Parser Builders

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

**Parser
Builders**

XML
Processing

XML
Processing

Concurrent
Program-
ming

Constructing Parsers with Parser Builders

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

**Parser
Builders**

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Tools that can be combined to build real parsers.

Constructing Parsers with Parser Builders

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

**Parser
Builders**

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Tools that can be combined to build real parsers.
- A toy parser that accepts binary numerals.

Constructing Parsers with Parser Builders

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Tools that can be combined to build real parsers.
- A toy parser that accepts binary numerals.

```
import scala.util.parsing.combinator._
class BinDigit extends JavaTokenParsers {
  def D = "0" | "1"
  def B = D~rep(D)
  def parse(text : String) = parseAll(B,text)
}
var P = new BinDigit
println("input : "+args(0))
println(P.parse(args(0)))
```

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

**XML
Processing**

XML
Processing

Concurrent
Program-
ming

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

**XML
Processing**

XML
Processing

Concurrent
Program-
ming

- Variables can be assigned XML content.

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Variables can be assigned XML content.

```
var poem =  
  <poem> <title> Magic Everywhere </title>  
  <poet realname="yes"> Tom Jones </poet>  
  <stanza> <verse> magic everywhere </verse>  
  </stanza> </poem>  
println(poem)
```

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Variables can be assigned XML content.

```
var poem =  
  <poem> <title> Magic Everywhere </title>  
  <poet realname="yes"> Tom Jones </poet>  
  <stanza> <verse> magic everywhere </verse>  
  </stanza> </poem>  
println(poem)
```

- Or they can be assigned XML content stored in files.

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Variables can be assigned XML content.

```
var poem =  
  <poem> <title> Magic Everywhere </title>  
  <poet realname="yes"> Tom Jones </poet>  
  <stanza> <verse> magic everywhere </verse>  
  </stanza> </poem>  
println(poem)
```

- Or they can be assigned XML content stored in files.

```
val x = XML.loadFile("listdir.html")
```

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

**XML
Processing**

Concurrent
Program-
ming

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

**XML
Processing**

Concurrent
Program-
ming

- Saving XML content.

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Saving XML content.

```
var z = <person>
<name>Blaise</name> <surname>Pascal</surname>
</person>
XML.save("persons.xml",z)
```

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

- Saving XML content.

```
var z = <person>
<name>Blaise</name> <surname>Pascal</surname>
</person>
XML.save("persons.xml",z)
```

- Example: Find all poems that have been published after 1960 and print the title and the poet of each such poem.

XML Processing

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- Saving XML content.

```
var z = <person>
<name>Blaise</name> <surname>Pascal</surname>
</person>
XML.save("persons.xml",z)
```

- Example: Find all poems that have been published after 1960 and print the title and the poet of each such poem.

```
for ( val poem <- poems \ "poem" ) {
  if ( (poem \ "year").text.trim.toInt > 1960 ) {
    var poet = (poem \ "poet").text
    var title = (poem \ "title").text
    println("\\""+title+"\\" by "+poet)
  }
}
```

Truths

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Truths

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- EETimes Europe
(<http://www.electronics-eetimes.com>): "As Moore's Law runs out of steam and computing goes mobile, technologists are searching for ways to make the leap to new parallel programming frameworks that can leverage low-power multicore architectures."

Truths

- EETimes Europe (<http://www.electronics-eetimes.com>): "As Moore's Law runs out of steam and computing goes mobile, technologists are searching for ways to make the leap to new parallel programming frameworks that can leverage low-power multicore architectures."

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Truths

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- EETimes Europe
(<http://www.electronics-eetimes.com>): "As Moore's Law runs out of steam and computing goes mobile, technologists are searching for ways to make the leap to new parallel programming frameworks that can leverage low-power multicore architectures."
- Concurrent programming is not only the next logical step in software development, but the next necessary step.

Truths

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- EETimes Europe
(<http://www.electronics-eetimes.com>): "As Moore's Law runs out of steam and computing goes mobile, technologists are searching for ways to make the leap to new parallel programming frameworks that can leverage low-power multicore architectures."
- Concurrent programming is not only the next logical step in software development, but the next necessary step.

Truths

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

- EETimes Europe (<http://www.electronics-eetimes.com>): "As Moore's Law runs out of steam and computing goes mobile, technologists are searching for ways to make the leap to new parallel programming frameworks that can leverage low-power multicore architectures."
- Concurrent programming is not only the next logical step in software development, but the next necessary step.
- **All modern programming languages must provide constructs and libraries that will ease the construction of concurrent programs**

Scala and Concurrency

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Scala programmers can code concurrent structures
with...

Scala and Concurrency

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Scala programmers can code concurrent structures with...

- threads (i.e., Java's legacy to Scala)

Scala and Concurrency

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Scala programmers can code concurrent structures with...

- threads (i.e., Java's legacy to Scala)

Scala and Concurrency

Scala programmers can code concurrent structures with...

- threads (i.e., Java's legacy to Scala)
- **mailboxes (something like actors)**

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Scala and Concurrency

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Scala programmers can code concurrent structures with...

- threads (i.e., Java's legacy to Scala)
- mailboxes (something like actors)

Scala and Concurrency

Scala programmers can code concurrent structures with...

- threads (i.e., Java's legacy to Scala)
- mailboxes (something like actors)
- **actors**

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Creating threaded applications by extending class Thread

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Creating threaded applications by extending class Thread

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

```
class PrintProgressMark(val mark: Char,  
                        val delay :Int) extends Thread {  
  private var i = 0  
  override def run(): Unit =  
    try {  
      while (i <= 100) {  
        print(mark); i += 1; Thread.sleep(delay);  
      }  
    } catch { case ex : InterruptedException => return }  
}  
  
object threadExample {  
  def main(args: Array[String]) {  
    new PrintProgressMark('+', 40).start  
    new PrintProgressMark('*', 100).start  
  }  
}
```

Creating threaded applications by using trait Runnable

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Creating threaded applications by using trait Runnable

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

```
class PrintProgressMark2(val mark: Char,  
                          val delay :Int) extends Runnable {  
  private var i = 0  
  override def run(): Unit =  
    try {  
      while (i <= 100) {  
        print(mark); i += 1; Thread.sleep(delay_)  
      }  
    } catch { case ex : InterruptedException => return }  
}  
  
object threadExample2 {  
  def main(args: Array[String]) {  
    var plus = new PrintProgressMark2('+', 40)  
    var ast = new PrintProgressMark2('*', 100)  
    new Thread(plus).start(); new Thread(ast).start()  
  } }  
}
```

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- **a finite set of new actors created.**

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- a finite set of new actors created.

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- a finite set of new actors created.

Thus,

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- a finite set of new actors created.

Thus,

- behaviors are *history sensitive*;

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- a finite set of new actors created.

Thus,

- behaviors are *history sensitive*;

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- a finite set of new actors created.

Thus,

- behaviors are *history sensitive*;
- there is *no presumed sequentiality* in the actions an actor performs; and,

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- a finite set of new actors created.

Thus,

- behaviors are *history sensitive*;
- there is *no presumed sequentiality* in the actions an actor performs; and,

What is an actor?

An actor is a computational agent which maps each incoming communication to a triple consisting of...

- a finite set of communications sent to other actors;
- a new behavior that will govern the response to the next communication processed; and,
- a finite set of new actors created.

Thus,

- behaviors are *history sensitive*;
- there is *no presumed sequentiality* in the actions an actor performs; and,
- ***actor creation is an integral of the computational model.***

A first example in actor programming

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

A first example in actor programming

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

```
import scala.actors.Actor._
object actorsExample {
  def main(args: Array[String]) {
    var PrintProgressPlus = actor {
      var i = 0
      while (i <= max) {
        print('+'); i += 1; Thread.sleep(40)
      }
    }
    var PrintProgressTimes = actor {
      var i = 0
      while (i <= max) {
        print('*'); i += 1; Thread.sleep(100)
      }
    }
    1 // <--- block must end in result expression,
  } //      not in definition
}
```

Defining a class that subclasses trait Actor

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Defining a class that subclasses trait Actor

```
import scala.actors._
class PrintProgressMark(val mark: Char,
                        val delay :Int) extends Actor {
  private var i = 0
  def act: Unit = {
    while (i <= 100) {
      print(mark); i += 1; Thread.sleep(delay_)
    }
  }
}

object actorsExample {
  def main(args: Array[String]) {
    new PrintProgressMark('+', 40).start
    new PrintProgressMark('*', 100).start
  }
}
```

Computing Factorials

Carl Hewitt in “Viewing Control Structures as Patterns of Passing Messages” (*Artificial Intelligence* 8, 3 (1977), 323–364)

Scala: An OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional Features

Parser Builders

XML Processing

XML Processing

Concurrent Programming

Computing Factorials

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Carl Hewitt in “Viewing Control Structures as Patterns of Passing Messages” (*Artificial Intelligence* 8, 3 (1977), 323–364)

- “present[s] an approach to modelling intelligence in terms of a society of communicating knowledge-based problem solving experts”;

Computing Factorials

Carl Hewitt in “Viewing Control Structures as Patterns of Passing Messages” (*Artificial Intelligence* 8, 3 (1977), 323–364)

- “present[s] an approach to modelling intelligence in terms of a society of communicating knowledge-based problem solving experts”;

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Computing Factorials

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Carl Hewitt in “Viewing Control Structures as Patterns of Passing Messages” (*Artificial Intelligence* 8, 3 (1977), 323–364)

- “present[s] an approach to modelling intelligence in terms of a society of communicating knowledge-based problem solving experts”;
- “each of the experts can be viewed as a society that can be further decomposed in the same way until the primitive actors of the system are reached”;

Computing Factorials

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Carl Hewitt in “Viewing Control Structures as Patterns of Passing Messages” (*Artificial Intelligence* 8, 3 (1977), 323–364)

- “present[s] an approach to modelling intelligence in terms of a society of communicating knowledge-based problem solving experts”;
- “each of the experts can be viewed as a society that can be further decomposed in the same way until the primitive actors of the system are reached”;

Computing Factorials

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Carl Hewitt in “Viewing Control Structures as Patterns of Passing Messages” (*Artificial Intelligence 8*, 3 (1977), 323–364)

- “present[s] an approach to modelling intelligence in terms of a society of communicating knowledge-based problem solving experts”;
- “each of the experts can be viewed as a society that can be further decomposed in the same way until the primitive actors of the system are reached”;
- he was “investigating the nature of the communication mechanisms needed for effective problem-solving by a society of experts and the conventions of civilized discourse that make this possible.”

Computing Factorials

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Computing Factorials

- Hewitt demonstrates these “principles” by showing how to compute the factorial function with actors:

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Computing Factorials

- Hewitt demonstrates these “principles” by showing how to compute the factorial function with actors:

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Computing Factorials

- Hewitt demonstrates these “principles” by showing how to compute the factorial function with actors:
 - 1 using a *recursive* procedure, that is, a procedure that uses *replication* and

Scala: An OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

Computing Factorials

- Hewitt demonstrates these “principles” by showing how to compute the factorial function with actors:
 - 1 using a *recursive* procedure, that is, a procedure that uses *replication* and

Scala: An OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional Features

Parser Builders

XML Processing

XML Processing

Concurrent Programming

Computing Factorials

- Hewitt demonstrates these “principles” by showing how to compute the factorial function with actors:
 - 1 using a *recursive* procedure, that is, a procedure that uses *replication* and
 - 2 an *iterative* procedure, that is, a procedure that relies on message passing to deliver a result.

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

Actor Replication

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Actor Replication

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

```
case class Val(a: Actor, n: Int)
case class Res(n: Int)
class Fact extends Actor {
  def act =
    react {
      case Val(a,0) => a ! Res(1)
      case Val(a,n) =>
        var p = new Fact
        p.start
        p ! Val(self,n-1)
        react { case Val(_, m) => a ! Val(a, n*m)
                case Res(m)      => a ! Val(a, n*m)}
    }
}
```

Using Class Fact

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Using Class Fact

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

```
object actorsExample {
  def main(args: Array[String]) {
    var factorial = actor {
      react {
        case Val(a,n) =>
          var q = new Fact
          q.start
          q ! Val(self,n)
          react { case Val(_,f) => a ! Val(a,f) }
      }
    }
    factorial ! Val(self,6)
    react { case Val(_, n) => println(n) }
  }
}
```

Iterative Actor

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Iterative Actor

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional

Features

Parser

Builders

XML

Processing

XML

Processing

Concurrent

Program-

ming

```
case class InitVal(a: Actor, n: Int)
case class Val(a: Actor, n: Int, m: Int)
object actorsExample {
  def main(args: Array[String]) {
    val loopActor = actor {
      var ResActor : Actor = null
      loop{
        react {
          case Val(_,acc,1)      =>
            ResActor ! acc; exit("done")
          case Val(a,1,count)   =>
            ResActor = a
            self ! Val(self,count,count-1)
          case Val(a,acc,count) =>
            self ! Val(self,acc*count,count-1)
        }
      }
    }
  }
}
```

Using Class Fact

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Using Class Fact

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?
Object Orientation
Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

```
var factorial = actor {
  react {
    case InitVal(a,0) => a ! InitVal(a, 1)
    case InitVal(a,1) => a ! InitVal(a, 1)
    case InitVal(a,n) =>
      loopActor ! Val(self,1,n)
      react { case f: Int => a ! InitVal(a,f) }
  }
}
factorial ! InitVal(self,6)
receive { case InitVal(_, n) => println(n) }
}
```

The end!

Scala: An
OO Surprise

Syropoulos

Introduction

What is Scala?

Object Orientation

Scalability

Functional
Features

Parser
Builders

XML
Processing

XML
Processing

Concurrent
Program-
ming

Thank you so much for your attention!