



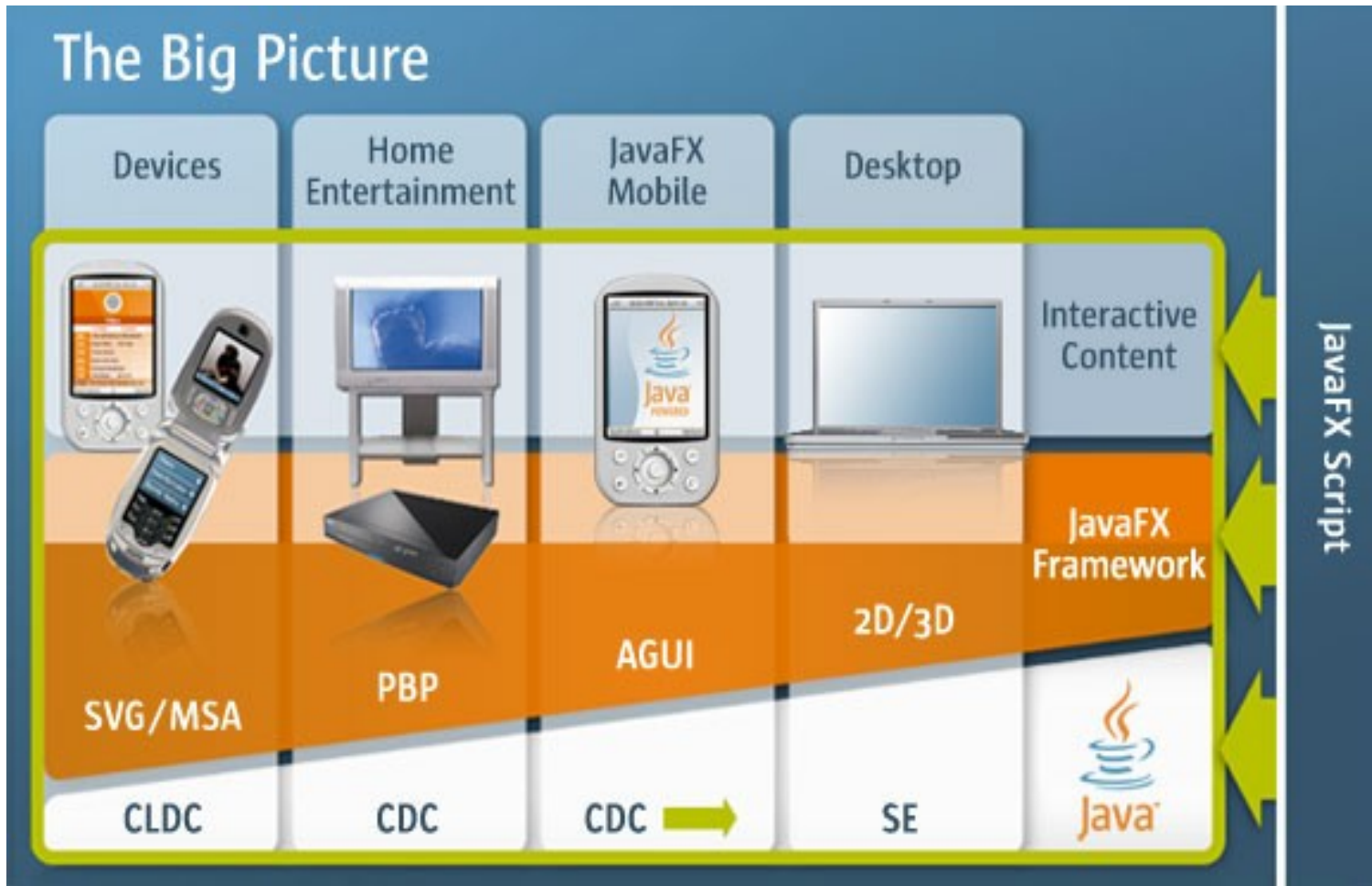
JavaFX

Simon Ritter

Technology Evangelist

Sun Microsystems

JavaFX



JavaFX: Design Questions

- Why does it take a long time to write GUI programs?
- How can we avoid the “Ugly Java GUI” stereotype?
- Why do Flash programs look different than Java programs?
- Why does it seem easier to write web-apps than Swing programs?
- And how can I avoid having an enormous, writhing mass of listener patterns?

JavaFX Basics

- Programming Language for the Java Platform
- Object-oriented
- Declarative Syntax
- Statically-typed with type-inference
- Automatic data binding
- Extensive Widget library encompassing Swing and Java 2D™ API
- Development tools including NetBeans™ and Eclipse IDE plugins

The “Ugly Java GUI” Stereotype

- AWT/Swing Container/Component Hierarchy
 - > A tree of rectangular (mostly grey) boxes
 - > If all you do is compose Swing components together, the result is typically “the Ugly Java GUI”
 - > Same problem exists with other toolkits, e.g., GTK, VB
- UI Designers and Swing programmers are using different building blocks
 - > UI Designers compose designs in tools like Photoshop and Illustrator
 - > The building blocks they use have direct analogs in Java 2D, but not always directly in Swing

A Basic Java GUI: Not Very Pretty



Java 2D API

- To match the designs of UI designers requires using Java 2D API
- Java 2D API doesn't have compositional behavior
 - > Makes it too complex for many programmers to use efficiently
- In addition to Swing Components, JavaFX includes SVG-like interfaces to Java 2D API as first-class elements which can be composed together into higher-level components
- FX allows declarative expression of this composition

Benefits of Declarative Syntax

- You can see it in Web applications
- For example, ease of composing styled text
 - > HTML versus JTextPane
- HTML Table using JSTL versus JTable
- JavaFX brings that same ease of use to Swing and Java 2D programming

HTML Table With JSTL

```
<table>
  <tr>
    <th>Description</th>
    <th>Price</th>
  </tr>
  <forEach var="item" items="{cart.items}">
    <tr>
      <td>${item.description}</td>
      <td>${item.price}</td>
    </tr>
  </forEach>
</table>
```

JavaFX Widget Set

- Maps to Swing and Java 2D components
- Can be used easily in scripts
 - > Define attributes quickly and easily
 - > Code functionality around this
- Lets look at the scripting syntax

Variable Declaration

- `var name : type [?, +, *] = initializer;`
 - > ? = optional
 - > + = one or more
 - > * = zero or more
- `var digits : Number* = [1, 2, 3];`
- `var digits = [1, 2, 3];`
- `var name = 'foo'; // or = "foo";`
- `{ }` for variable name in strings
- `<< >>` to use reserved word for variable name
- Newlines can be placed directly in strings

Functions

- Pure functional subset of Java language
- Only contains var declarations and return statement

```
function percent(a, b) {  
    var i = a * 100;  
    return i / b;  
}
```

Operations (Procedures)

```
operation substring(s:String, n:Number) :  
  String {  
    try {  
      return s.substring(n) ;  
    catch (e : StringOutOfBoundsException) {  
      return "Index out of bounds";  
    }  
  }  
}
```

Array Definitions

- Enclosed in []
- Separated by commas
- Do not nest
- Use .. to indicate arithmetic range
 - > `var oneToTen = [1 .. 10];`
- Can contain expressions
 - > `var greaterThanFive = oneToTen[. > 5];`
- indexof function
 - > `list[indexof . > 0];`
// all but first element

Inserting Into Arrays

- insert expr [as first | as last] into expr2;
- insert expr before expr2;
- insert expr after expr2;

```
var x = [1,2];  
insert 12 into x; // [1,2,12]  
insert 10 as first into x; // [10,1,2,12]  
insert 11 after x[. == 2]; //  
    [10,1,2,11,12]
```

Deleting From Arrays

- delete var;
- delete expr.attribute;
- delete variable[predicate];
- delete expr.variable[predicate];

```
var x = [1,2,3,4,5];  
delete x[. == 2]; // [1,3,4,5]  
delete x[. > 3]; // [1,3]  
delete x; // []
```

Querying Arrays (List Comprehension)

```
var titleTracks =  
    select indexof track + 1 from  
        album in albums,  
        track in album.tracks  
    where track == album.title;  
  
var squares = select n*n from n in [1..10];
```

Formatting

- expr format as << directive >>
- directive can be:
 - > java.text.DecimalFormat
 - > java.text.SimpleData
 - > java.util.Formatter (always starts with %)

```
100.896 format as <<%f>> // 100.896000
```

```
31.intValue() format as <<%02X>> // 1F
```

Expressions

- if, while, try – Same syntax as Java
- `for (i in [0..10]) ...`
- `for (i in [0..10] where i%2 == 0)`
...
- `for (i in [0..10], j in [0..10]) ...`

Avoiding the Event Dispatch Thread

```
do {  
    // block of code executes in  
    // separate thread  
}
```

```
do later {  
    // block of code using  
    // java.awt.EventQueue.invokeLater  
}
```

Classes

```
class Person {
    attribute name: String;
    attribute parent: Person inverse
        Person.children;
    attribute children: Person* inverse
        Person.parent;
    function getNumberOfChildren(): Number;
}
function Person.getNumberOfChildren() {
    return sizeof this.children;
}
```

Attributes

```
class Point {  
    attribute x: Number;  
    attribute y: Number;  
    attribute z: Number;  
}
```

```
attribute Point.x = 10;  
attribute Point.y = bind x + 10;  
attribute Point.z = bind lazy y + 10;
```

```
var p = new Point(); // x=10,y=20,z=0;  
p.x = 5 // x=5,y=15,z=0  
System.out.println("z="+p.z); // "z=30" 22
```

Triggers

```
class X {
    attribute nums: Number*;
}

trigger on new X { // Creation trigger
    insert [1,2] into this.nums;
}

trigger on insert num into X.nums {
    System.out.println("{num} added to X");
}

trigger on delete num from X.nums {
    System.out.println("{num} deleted from X");
}

trigger on X.nums[oVal] = nVal {
    System.out.println("{nVal} replaced {oVal} in X");
}
```

Summary

- JavaFX is a family of products and technologies aimed at content creators
- JavaFX script simplifies GUI programming
 - > Let the graphic artists do the hard work
- More coming, watch this space
 - > Ease of deployment
 - > Consumer, modular JRE

Further Information

<http://www.sun.com/javafx>

<http://openjfx.org>

JavaFX

Simon Ritter

simon.ritter@sun.com

<http://blogs.sun.com/simonri>

Sun Microsystems