Introduction to a Groovy based DSL for Apache OFBiz™

Jacopo Cappellato

CORINTHIA HOTEL
BUDAPEST, HUNGARY
NOVEMBER 17-21, 2014

“How to write better business logic code with the new OFBiz DSL”

Presented by Jacopo Cappellato

OFBiz committer since before the project joined the ASF (2006)
ASF member since 2007
OFBiz PMC Chair since 2010
VP Technology at HotWax Media www.hotwaxmedia.com
Email: jacopo.cappellato@hotwaxmedia.com or jacopoc@apache.org

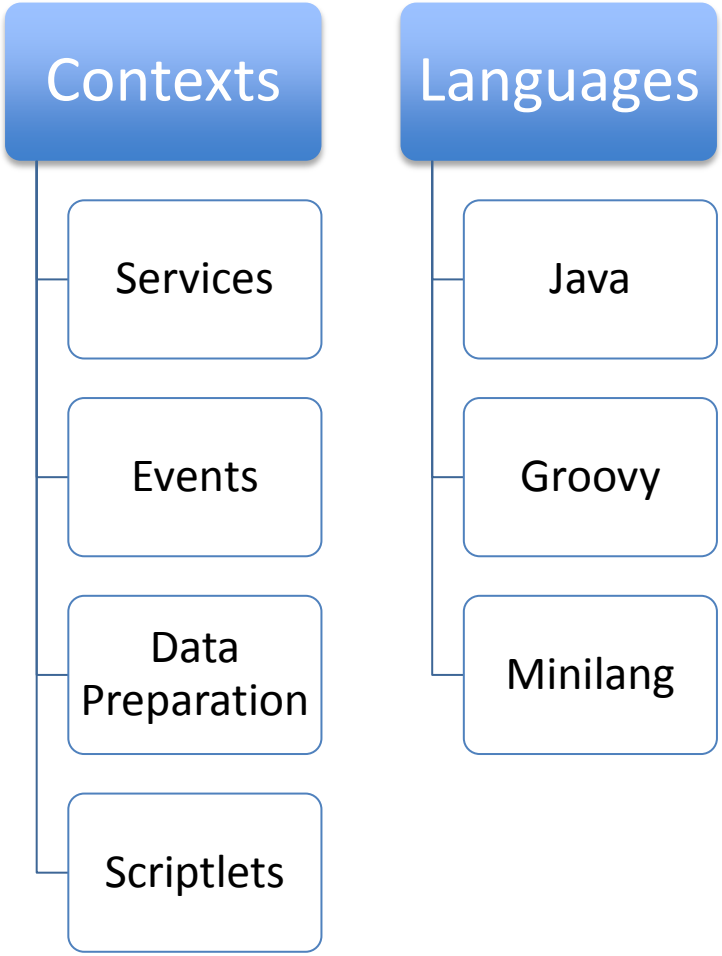
Outline

- Introduction to the business logic in OFBiz
  - Business logic contexts
  - Languages currently adopted
- Introduction to the OFBiz DSL
- Language comparison by examples
- Summary and references

Introduction to the business logic in OFBiz

Business logic contexts in OFBiz and languages currently adopted

Business logic contexts in OFBiz and languages currently adopted



From different languages for different contexts…

Minilang	Java	Groovy
• Services • Events • Data preparation	• Services • Events	• Services • Events • Data preparation • Scriptlets

… to one language for all the contexts.

OFBiz DSL

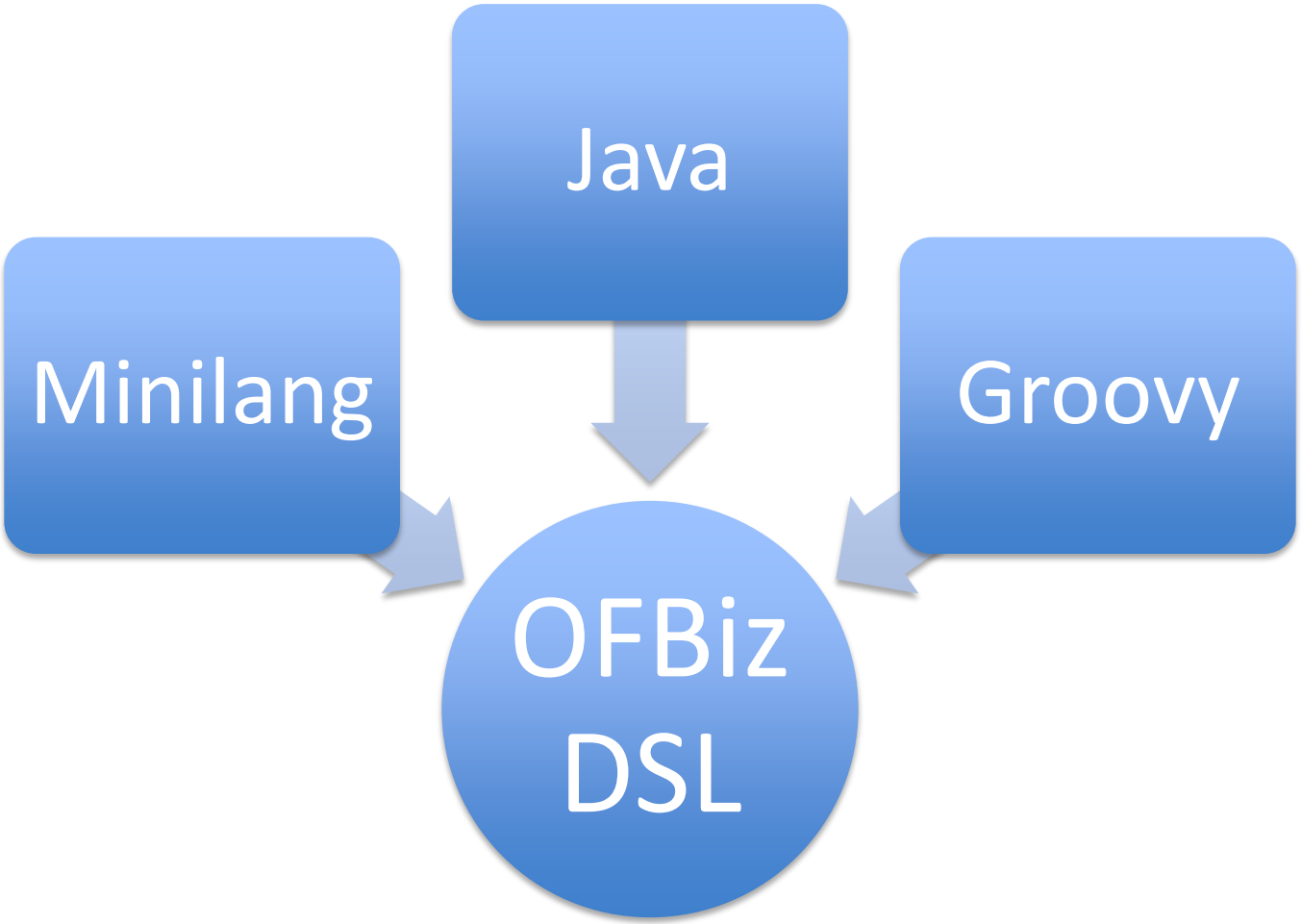
- Services
- Events
- Data preparation
- Scriptlets


Introduction to the OFBiz DSL

Definitions

- Domain Specific Language
  - “A Domain Specific Language (DSL) is a programming language designed specifically to express solutions to problems in a specific domain”
- OFBiz DSL
  - A simple Groovy-based DSL for the implementation of business logic in OFBiz (events, services, data preparation scripts, scriptlets)

The OFBiz DSL as an evolution of the existing languages used in OFBiz



The OFBiz DSL brings together into one unified language all the powerful features of Java, Groovy and Minilang

How to use the OFBiz DSL

- Full support available in the OFBiz trunk
- All Groovy code in OFBiz is already enabled to use the DSL
  - Services
    - engine="groovy"
  - Events
    - type="groovy"
  - Data preparation scripts
    - *.groovy
  - Scriptlets
    - ${groovy:

How to implement OFBiz DSL code

- DSL methods currently available
  - Service operations
    - runService
  - Entity operations
    - findOne, findList, makeValue, select, from
  - Results
    - success, failure, error
  - Logging
    - logInfo, logWarning, logError
- In evolution

Language comparison by examples

Service method definitions

```
public static Map<String, Object> myService(DispatchContext dctx,
                                            Map<String, Object> context) {
    return ServiceUtil.returnSuccess("myService executed successfully");
}
```

Service method definitions

```java
public static Map<String, Object> myService(DispatchContext dctx,
                                            Map<String, Object> context) {
    return ServiceUtil.returnSuccess("myService executed successfully");
}
```

```xml
<simple-method method-name="myService">
    <return/>
</simple-method>
```

Service method definitions

```java
public static Map<String, Object> myService(DispatchContext dctx,
                                            Map<String, Object> context) {
    return ServiceUtil.returnSuccess("myService executed successfully");
}
```

```xml
<simple-method method-name="myService">
    <return/>
</simple-method>
```

```groovy
def myService() {
    success 'myService executed successfully'
}
```

Logging

```
Debug.logInfo("Product " + productId + " found", module);
```

Logging

```
Debug.logInfo("Product " + productId + " found", module);
```

```
<log level="info" message="Product ${productId} found"/>
```

Logging

```
Debug.logInfo("Product " + productId + " found", module);
```

```
<log level="info" message="Product ${productId} found"/>
```

```
logInfo "Product ${productId} found"
```

Arithmetic operations

```
<calculate field="estimatedTaskTime">
    <calcop field="totalEstimatedTaskTime" operator="subtract">
        <calcop field="setupTime" operator="get"/>
    </calcop>
</calculate>
```

```
<set field="estimatedTaskTime"
     value="${totalEstimatedTaskTime - setupTime}"
     type="BigDecimal"/>
```

Arithmetic operations

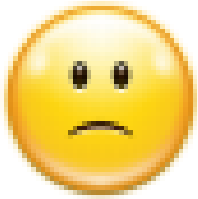
```
<calculate field="estimatedTaskTime">
    <calcop field="totalEstimatedTaskTime" operator="subtract">
        <calcop field="setupTime" operator="get"/>
    </calcop>
</calculate>
```

```
<set field="estimatedTaskTime"
     value="${totalEstimatedTaskTime - setupTime}"
     type="BigDecimal"/>
```

```
estimatedTaskTime = totalEstimatedTaskTime - setupTime;
```

Service calls

```
LocalDispatcher dispatcher = dctx.getDispatcher();
GenericValue userLogin = (GenericValue) context.get("userLogin");
String productId = (String) context.get("productId");
Map<String, Object> inputMap = new HashMap<String, Object>();
Map<String, Object> outputMap;
inputMap.put("productId", productId);
inputMap.put("facilityId", "WebStoreWarehouse");
inputMap.put("userLogin", userLogin);
try {
    outputMap = dispatcher.runSync("getInventoryAvailableByFacility",
                                   inputMap);
    BigDecimal qoh = (BigDecimal) outputMap.get("quantityOnHandTotal");
    if (qoh.equals(BigDecimal.ZERO)) {
        Debug.logWarning("QOH is zero for " + productId, module);
    }
} catch(GenericServiceException gse) {}
```



Service calls

```
<set field="inputMap.productId" from-field="parameters.productId"/>
<set field="inputMap.facilityId" value="WebStoreWarehouse"/>
<call-service service-name="getInventoryAvailableByFacility"
              in-map-name="inputMap">
    <result-to-field result-name="qoh"/>
</call-service>
<if-compare field="qoh" operator="equals" value="0" type="BigDecimal">
    <log level="warning" message="QOH is zero for ${parameters.productId}"/>
</if-compare>
```

Service calls

```
countResult = runService('getInventoryAvailableByFacility',
                         [productId: parameters.productId,
                          facilityId: 'WebStoreWarehouse'])
if (countResult.qoh == 0) {
    logWarning "QOH is zero for ${parameters.productId}"
}
```

Service calls

```
result = runService('getReturnableItems', [orderId : orderId])
```

Service calls

```
result = runService('getReturnableItems', [orderId : orderId])
```

```
result = run service: 'getReturnableItems', with: [orderId : orderId]
```

Database operations

```
product = select().from('Product').where('productId', 'WG-1111').queryOne()

product = from('Product').where('productId', 'WG-1111').queryOne()
```

Database operations

```
product = select().from('Product').where('productId', 'WG-1111').queryOne()
```

```
product = from('Product').where('productId', 'WG-1111').queryOne()
```

```
product = select('productId', 'internalName').from('Product').where('productId', 'WG-1111').queryOne()
```

Database operations

```
product = select().from('Product').where('productId', 'WG-1111').queryOne()
```

```
product = from('Product').where('productId', 'WG-1111').queryOne()
```

```
product = select('productId', 'internalName').from('Product').where('productId', 'WG-1111').queryOne()
```

```
products = select().from('Product').queryList()
products.each { product ->
    logInfo "Product ${product.productId}"
}
```

Summary and references

Highlights of OFBiz DSL features

- Available in all the contexts (events, services, data preparation, scriptlets)
- Clean and simple API (in evolution)
- Powerful programming language (Groovy)
- Programming by exception pattern:
  - “Unless specified differently, the framework will apply the default rules”
- Automatic context detection
  - return success()
- Automatic parameter detection (e.g. userLogin)
  - runSync(‘createProduct’, [productId:’TEST-PRODUCT’])
- Transparent error and db transaction handling

Next steps

- Migrating Java/Minilang/Groovy code
- Testing and bug fixing
- Enhancing the OFBiz DSL
- IDE integration (Eclipse, IntelliJ available)

References

- Further information (and code samples):
  - https://cwiki.apache.org/confluence/x/_M_oAQ
- Apache OFBiz website:
  - http://ofbiz.apache.org
- Emails:
  - user@ofbiz.apache.org
  - jacopoc@apache.org
- Trademarks
  - Apache OFBiz, Apache are trademarks of The Apache Software Foundation
  - Java is a registered trademark of Oracle and/or its affiliates
  - Groovy is a registered trademark of The Codehaus