Drools & Predictive Analytics
Follow Your Rules and Listen to Your Data

Alex Guazzelli – VP Analytics
Kostas Stathatos – VP Engineering
Zementis, Inc.
Outline

1) Rules + Predictive Analytics - Why?

2) How? ADAPA Decision Engine

3) Drools Extensions

4) Sample Solution & Demo
**Predictive Analytics + Rules**

**Enhanced Decisioning**

**Rules** = *Expert Knowledge*

Logic used by experts to solve problems can be represented as business rules:

*When body temperature more than X AND blood pressure more than Y, then ...*

**Predictive Analytics** = *Data-Driven Knowledge*

Based on the ability to automatically recognize patterns in data not obvious to the expert eye.

*Learn from past behavior present in historical data to predict the future.*
Enhanced Decisioning 1
Predictive Analytics + Rules
Enhanced Decisioning 2
Predictive Analytics + Rules
Enhanced Decisioning 3
Predictive Analytics + Rules
Outline

1) Rules + Predictive Analytics  - Why?

2) How? ADAPA Decision Engine

3) Drools Extensions

4) Sample Solution & Demo
ADAPA: Bringing together the power of predictive analytics and rules

Business Rules

Predictive Analytics

PMML
ADAPA & Amazon Cloud

- **ADAPA by Zementis**
  - Predictive Decisioning Platform
  - Standards-based Predictive Analytics
    - PMML (www.dmg.org)
    - Generated by SAS, IBM/SPSS, KXEN, R, KNIME, ...
  - Drools to integrate business logic
  - Accessed through web services
  - Scalable execution platform

- **ADAPA on Amazon Elastic Computing Cloud**
  - Software as a service
  - Up/Down scaling as needed
  - Pay-as-you-go, no up-front capital investment
    - Amazon Payments
  - Amazon experience & reliability
Invoking Predictive Models from Rules

```java
global com.zementis.adapa.models.runtime.ModelInvoker ModelInvoker;

rule "Score Neural Network for ARM Products"
    when
        product:Product(processingStage="Model Selection", rateType = "ARM")
        borrower:Borrower()
        property:Property()
    then
        Map record = createRecord(borrower, property, ...);
        product.setAdapaScore( (Double) ModelInvoker.apply("NeuralNetwork_for_ARM", "risk", record));
        ... update(product);
end

function Map createRecord(Borrower borrower, Property property, ...) {
    Map record = new HashMap();
    record.put("firstTimeBuyer", borrower.getFirstTimeBuyer());
    record.put("employmentIndicator", borrower.getEmploymentType());
    record.put("borrowersState", property.getState());
    ...
    return record;
}
```
Deployment Options

(Predictive Analytics + Rules)  
On the Cloud

(Predictive Analytics On the Cloud)  
+ (Rules On Site)
Outline

1) Rules + Predictive Analytics - Why?

2) How? ADAPA Decision Engine

3) Drools Extensions

4) Sample Solution & Demo
Drools Extensions

ADAPA

Predictive Models
Business Rules

DAV - Decision, Acceptance, and Validation Framework

Lookup Tables

Multisheet Decision Tables
## Drools Extensions

### DAV Framework

<table>
<thead>
<tr>
<th>rulesets</th>
<th>ZementisGuideline</th>
<th>Loan Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>usage</td>
<td>given</td>
<td>check</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test Description</th>
<th>Borrower</th>
<th>LTV</th>
<th># of Objects</th>
<th>Loan Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Good Borrower</td>
<td>72</td>
<td>1</td>
<td>900000</td>
</tr>
<tr>
<td>Case 2</td>
<td>Good Borrower</td>
<td>80</td>
<td>1</td>
<td>850000</td>
</tr>
<tr>
<td>Case 3</td>
<td>Average Borrower</td>
<td>72</td>
<td>1</td>
<td>800000</td>
</tr>
</tbody>
</table>
Outline

1) Rules + Predictive Analytics - Why?
2) How? ADAPA Decision Engine
3) Drools Extensions
4) Sample Solution & Demo
Sample Solution:
Zementis Mortgage Company offers …
ADAPA Solution – Demo
Predictive Models

ZEMENTIS

![Image of ZEMENTIS dashboard]

**New Model Upload**

- **PMML File**
  - **Browse**
- **Upload Model**

**Note:** ADAPA attempts to upgrade and/or correct uploaded PMML files to version 3.2 as necessary.

**Available Models**

<table>
<thead>
<tr>
<th>Name</th>
<th>Actions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdapaDemoModelARM</td>
<td>![ ]</td>
<td>This is the ADAPA demo model. It computes risk for loans with ARM rate. The risk is hypothetical and is generated here only for demo purposes.</td>
</tr>
<tr>
<td>AdapaDemoModelFixed</td>
<td>![ ]</td>
<td>This is the ADAPA demo model. It computes risk for loans with fixed rate. The risk is hypothetical and is generated here only for demo purposes.</td>
</tr>
<tr>
<td>Iris_CT</td>
<td>![ ]</td>
<td>Classification Tree using the Iris dataset</td>
</tr>
<tr>
<td>Iris_MLR</td>
<td>![ ]</td>
<td>Multinomial Logistic Regression using the Iris dataset</td>
</tr>
<tr>
<td>Iris_NN</td>
<td>![ ]</td>
<td>Neural Network for multi-class classification using the Iris dataset</td>
</tr>
</tbody>
</table>

**Score/Classify Data**

- **RPC Web Service Description (WSDL)**
ADAPA Solution – Demo
Rule Sets

[Image of a webpage showing rule sets management with a table listing rule set names, actions, and creation dates.]
ADAPA Solution – Demo
Resources

- AdapaDemo.xsd: Domain Model
- borrowersStateMappingTable.xls: Lookup Tables

Available Resources:

- File Name: AdapaDemo.xsd
- Actions: Upload
- Content: Domain Model
- Resources: Package Names: com.zementis.adapa.demo

- File Name: borrowersStateMappingTable.xls
- Actions: Upload
- Content: Lookup Tables
- Resources: Table Names: State
ADAPA = EDM
Integrating across all systems and processes
ADAPA Solution - Demo

Predictive Models
Business Rules
Sample Solution:
Invoking models from rules

global com.zementis.adapa.models.runtime.ModelInvoker ModelInvoker;

rule "Score Neural Network for ARM Products"
  when
    product:Product(processingStage=="Model Selection", rateType == "ARM")
    borrower:Borrower()
    property:Property()
    ...
  then
    Map record = createRecord(borrower, property, ...);
    product.setAdapaScore( (Double) ModelInvoker.apply("NeuralNetwork_for_ARM", "risk", record));
    ...
    update(product);
end

function Map createRecord(Borrower borrower, Property property, ...) {
    Map record = new HashMap();
    record.put("firstTimeBuyer", borrower.getFirstTimeBuyer());
    record.put("employmentIndicator",borrower.getEmploymentType());
    record.put("borrowersState", property.getState());
    ...
    return record;