

MMC-BPM: A Domain-Specific Language for Business Processes Analysis

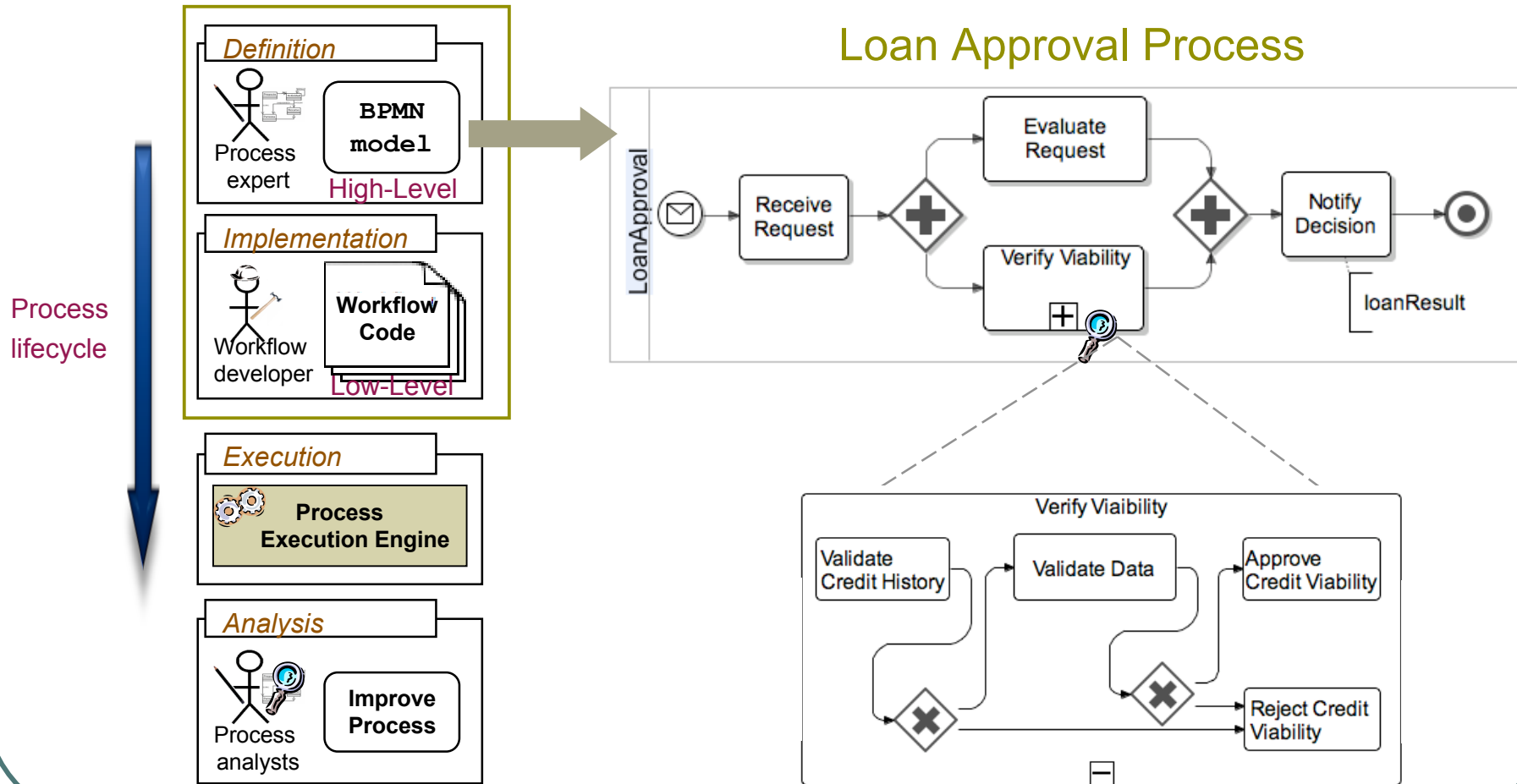
Oscar González, Rubby Casallas, and Dirk Deridder



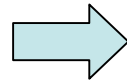
Agenda

- Business Process Analysis Scenario
- Problematic Aspects
- Our Approach
- Conclusions and Ongoing Work

Business Process Scenario



Process Improvement



Maximize the number of approved loan requests

R1. Obtain the state of the loan request



A1. Intercept workflow interactions
A2. Gather Information
A3. Access process information

R2. Obtain the rejected requests rate



A4. Construct custom measures
A5. Store Information

R3. Trigger a notification if the rejected requests rate is higher than 50%



A6. Define conditionals
A7. Apply control actions

How to support these requirements ?

Adding Analysis Activities Manually

Process Definition 449 xml LOCs

```

<bpel:process name="LoanApproval" ... > ...
<bpel:assign name="NotifyDecision">
  <bpel:copy>
    <bpel:from>$mensajeRespuestaWSApprove.return</bpel:from>
    <bpel:to>$mensajeRespuesta.body/tns:requestState</bpel:to>
  </bpel:copy> ... </bpel:assign>
  <bpel:copy> ... </bpel:assign>
  <bpel:from>$mensajeRespuesta.body/tns:requestState</bpel:from>
  <bpel:to>$request</bpel:to>
  </bpel:copy></bpel:assign> ...
</bpel:if>
<bpel:condition>false() = $request</bpel:condition> ...
<bpel:assign>
  <bpel:copy>
    <bpel:from>$RR div $STR</bpel:from>
    <bpel:to>$RRR</bpel:to>
  </bpel:copy></bpel:assign>
  <bpel:copy> ... </bpel:copy> ...
  <bpel:from>$code.pid</bpel:from>
  <bpel:to>$registerMonitoringVariableRequestMsg.arg2</bpel:to>
  </bpel:copy> ... </bpel:assign>
  <bpel:invoke partnerLink="WSMonitoringVariablePlkVar" .../>
</bpel:if>
<bpel:condition>$RRR * 100 &gt; 50</bpel:condition>
<bpel:assign>
  <bpel:copy>
    <bpel:from>"o-gonzal@uniandes.edu.co"</bpel:from>
    <bpel:to>$notifyActionRequestMsg.arg0</bpel:to>
  </bpel:copy> ... </bpel:assign>
  <bpel:invoke partnerLink="WSControlActionPlkVar" .../>
  <bpel:reply partnerLink="ClientAndRequestLoanProcessPlkVar" .../>
</bpel:assign>
<bpel:copy>
  <bpel:from>$mensajeRespuestaWSApprove.return</bpel:from>
  <bpel:to>$mensajeRespuesta.body/tns:requestState</bpel:to>
  </bpel:copy> ... </bpel:assign>
  <bpel:from>$mensajeRespuesta.body/tns:requestState</bpel:from>
  <bpel:to>$request</bpel:to>
  </bpel:copy></bpel:assign> ...
</bpel:if>
<bpel:condition>false() = $request</bpel:condition> ...
<bpel:assign>
  <bpel:copy>
    <bpel:from>$RR div $STR</bpel:from>
    <bpel:to>$RRR</bpel:to>
  </bpel:copy></bpel:assign>
  <bpel:copy> ... </bpel:copy> ...
  <bpel:from>$code.pid</bpel:from>
  <bpel:to>$registerMonitoringVariableRequestMsg.arg2</bpel:to>
  </bpel:copy> ... </bpel:assign>
  <bpel:invoke partnerLink="WSMonitoringVariablePlkVar" .../>
</bpel:if>
<bpel:condition>$RRR * 100 &gt; 50</bpel:condition>
<bpel:assign>
  <bpel:copy>
    <bpel:from>"o-gonzal@uniandes.edu.co"</bpel:from>
    <bpel:to>$notifyActionRequestMsg.arg0</bpel:to>
  </bpel:copy> ... </bpel:assign>
  <bpel:invoke partnerLink="WSControlActionPlkVar" .../>
  <bpel:reply partnerLink="ClientAndRequestLoanProcessPlkVar" .../>
</bpel:assign>
</bpel:process>
  
```

254 java LOCs

Underlying Implementation

Analysis Implementation

123 xml LOCs
210 java LOCs

```

<bpel:if>
  <bpel:condition>$RejectedRate * 100 &gt; 50</bpel:condition>
  <bpel:sequence>
    <bpel:assign>
      <bpel:copy>
        <bpel:from>"o-gonzal@uniandes.edu.co"</bpel:from>
        <bpel:to>$notifyActionRequestMsg.arg0</bpel:to>
      </bpel:copy><bpel:copy>
        <bpel:from>"review the log."</bpel:from>
        <bpel:to>$notifyActionRequestMsg.arg1</bpel:to>
      </bpel:copy><bpel:copy>
        <bpel:from>"Rejected rate is too high"</bpel:from>
        <bpel:to>$notifyActionRequestMsg.arg2</bpel:to>
      </bpel:copy>
    </bpel:assign>
    <bpel:invoke inputVariable="notifyActionRequestMsg"
      operation="notifyAction"
      outputVariable="notifyActionResponseMsg"
      partnerLink="WSControlActionPlkVar"
      portType="ControlAction:IControlAction"/>
  </bpel:sequence>
</bpel:if>
  
```

Problems

P1. Limited expressiveness

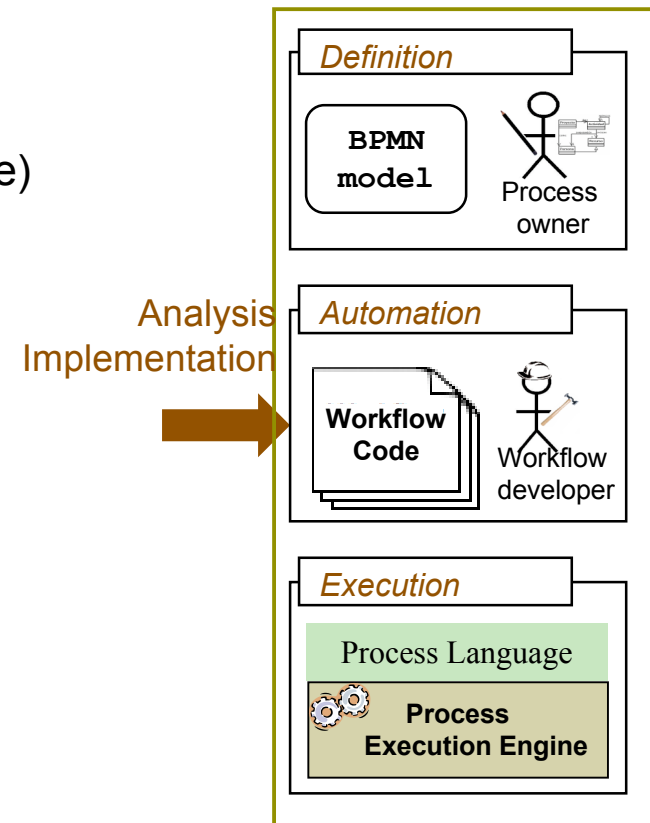
- ✗ Lack of abstractions
- ✗ Generic Monitoring (time running, current state)
- ✗ Process data is not explicit

P2. Tied to particular workflow platforms

- ✗ Usage of proprietary languages and engines

P3. Low-level implementation

- ✗ Encoded in the workflow implementation
- ✗ Crosscutting and entangled code
- ✗ Manual Composition

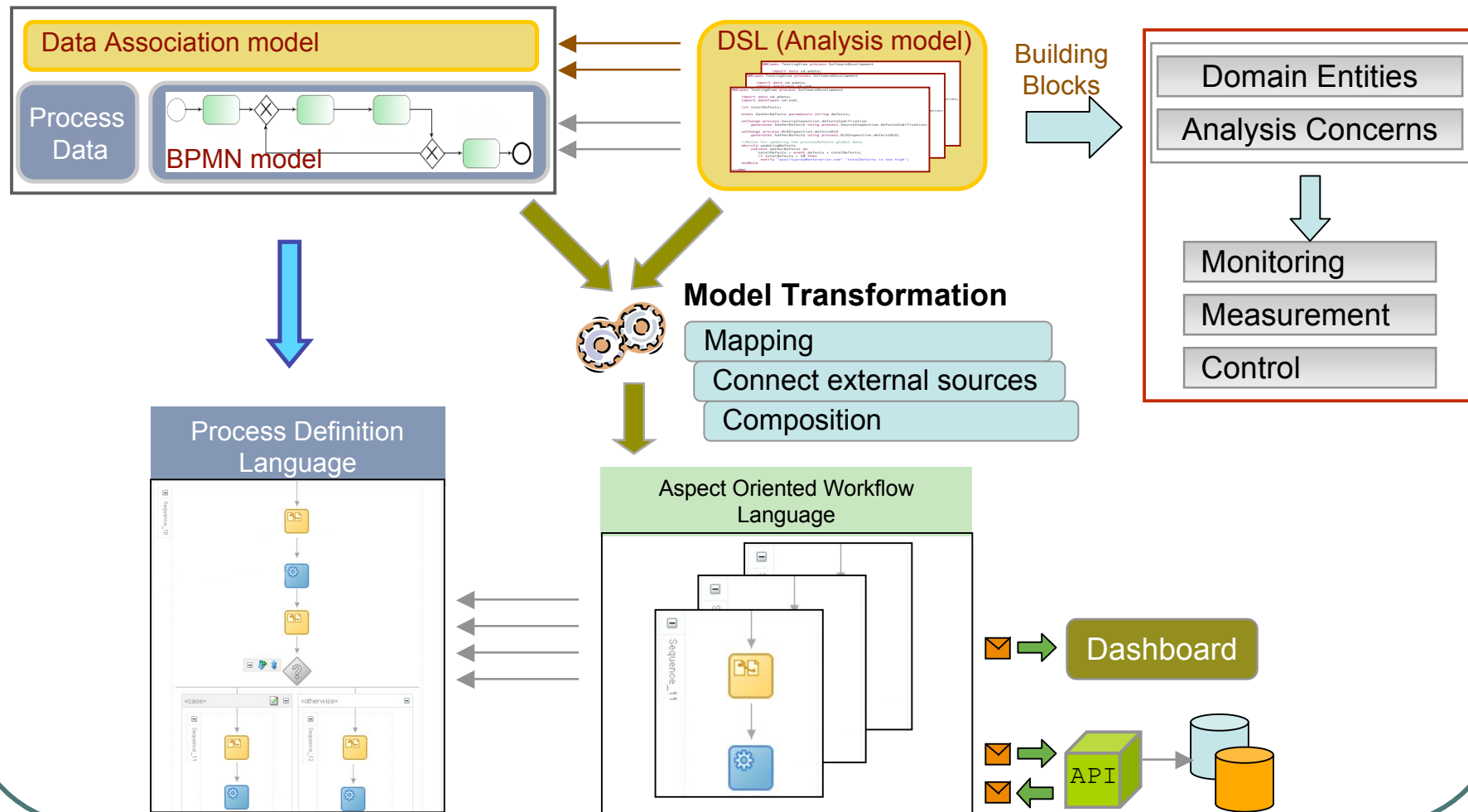


Adding Analysis Activities

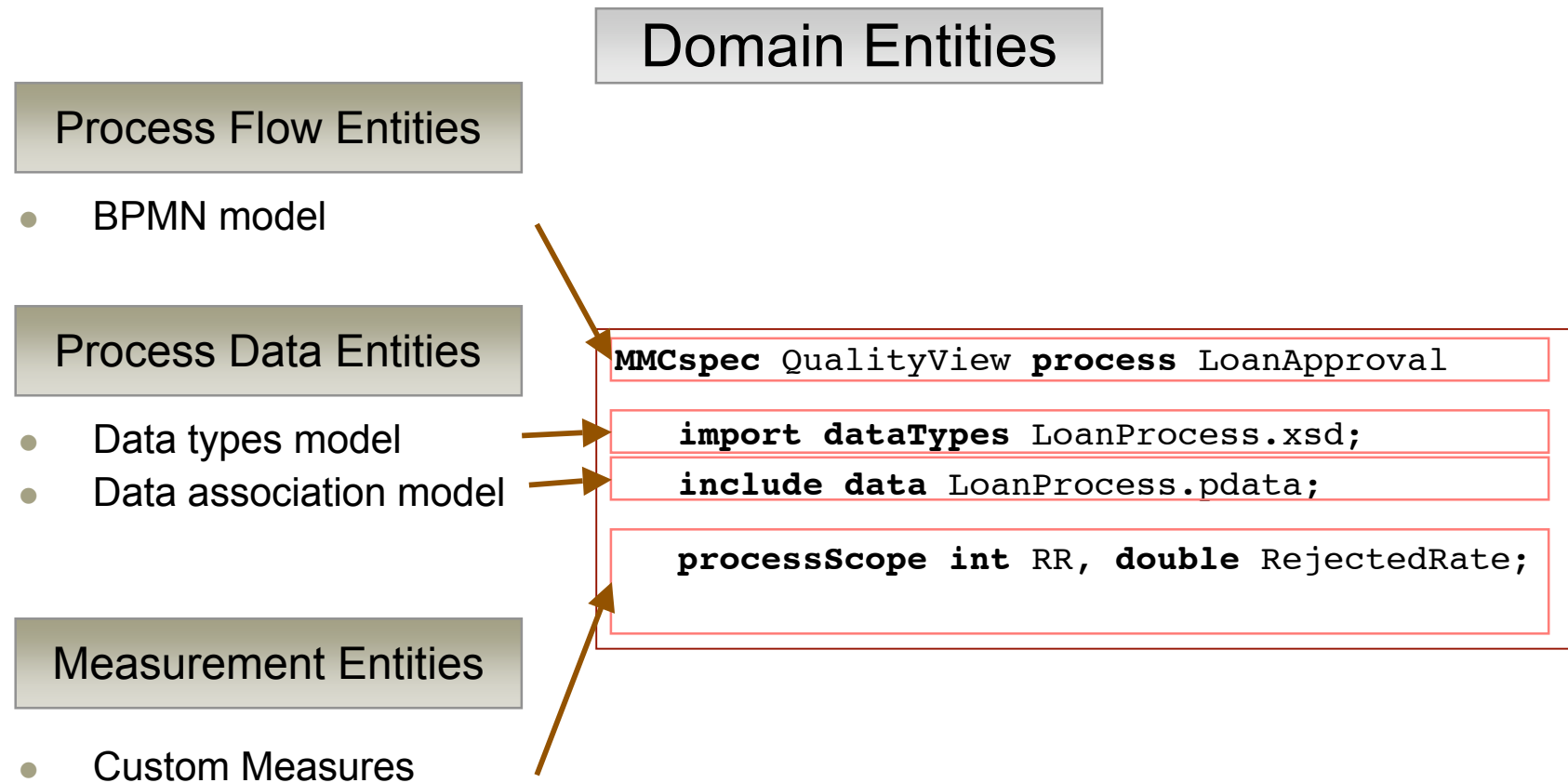
Desirable Features

- Process platform independence
- Domain-specific analysis
- Modularized Implementation
- Automatic Composition

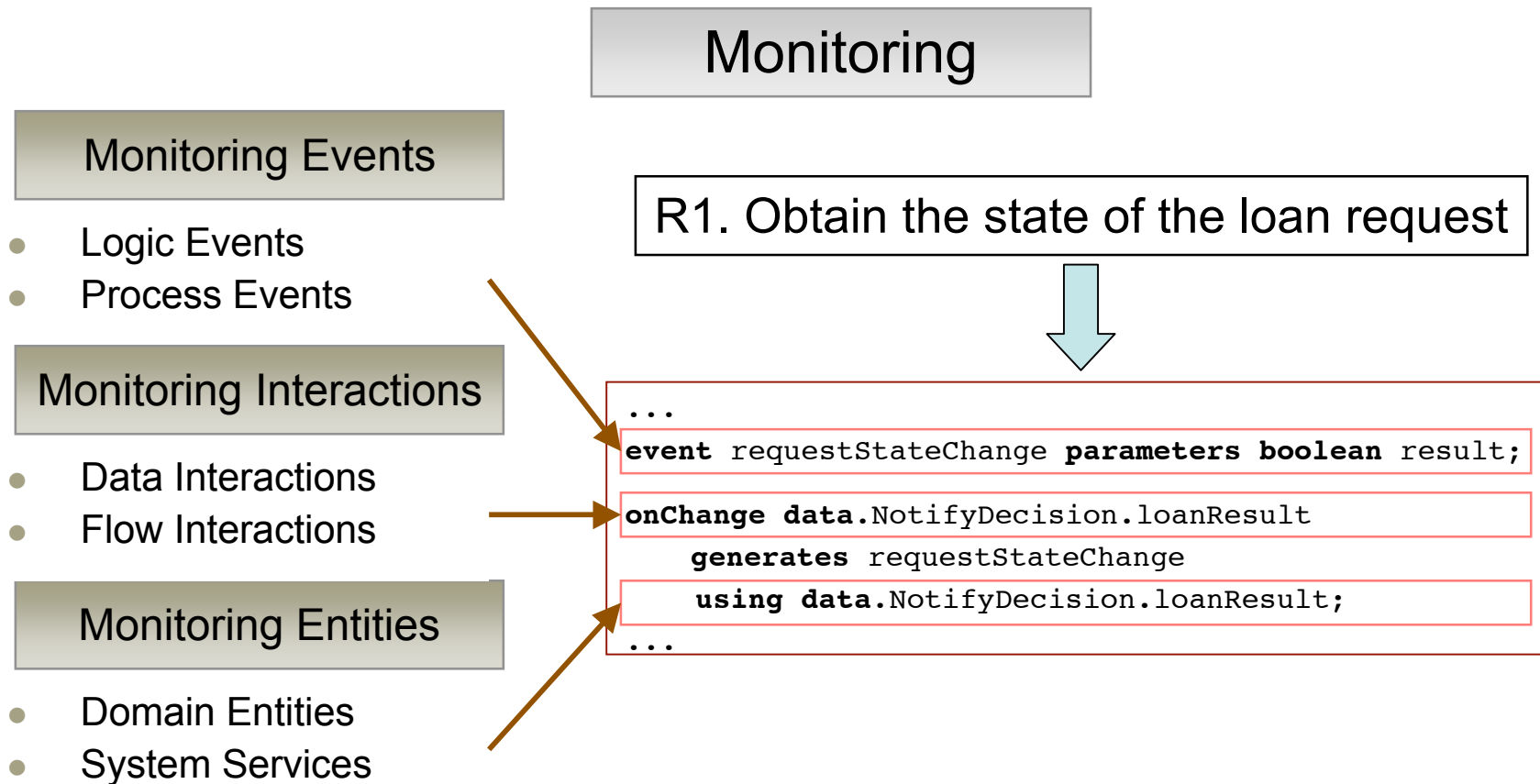
Overall Approach (Bppbification)



Domain Entities Specification



Analysis Concerns Specification



Analysis Concerns Specification

Measurement and Control

Measurement Actions

- Access information
- Manage measures

Evaluation Rules

- Condition-action

Control Actions

- Notify
- Alarm
- Trace

R2. Obtain the rejected requests rate

```
mmcrule UpdateRequestState onEvent requestStateChange do  
  increase(TR);  
  if !(event.result) then  
    increase(RR); RejectedRate = RR/engine.instances; endif  
  if RejectedRate*100 > 50 then  
    notify 'o-gonzal@uniandes.edu.co' ...; endif  
endRule  
endMMC
```

R3. Trigger a notification if the rejected requests rate is higher than 50%

Summary

Language Advantages

- Process platform independence
- Domain-specific analysis
- Modularized Implementation
- Automatic Composition
- Less specification effort
 - Analysis specification = 21 LOCS

Current and Ongoing Work

Domain-specific language and transformations

- <http://qualdev.uniandes.edu.co/>



Work in progress

- X Encoded in the workflow implementation
- ? Implementation of data interactions
- ? Verification of interferences

More Information

Oscar González

Email: o-gonza1@uniandes.edu.co

