Compliance Monitoring on Process Event Streams from Multiple Sources

Who are we? Patrik König, Jürgen Mangler, Stefanie Rinderle-Ma
Where are we from? University of Vienna

Contents of this presentation?

- Motivation.
- Resulting approach.
- Evaluation.
Motivation

• Business processes have to comply to a vast amount of regulations.
• These regulation can have multiple sources:
  • Internal: business rules, internal policies.
  • External: law, quality standards, other companies.
• The execution of one business process may span across multiple information systems.
• All involved information systems produce massive amounts of events.
• Every event may have a different data structure, based on the source.
• Checking of regulations must either be integrated in all information systems, OR operate on events from all information systems.
The need for separate compliance modeling

Process Model

Compliance Rule
Our Approach

Based on a collection of example compliance rules* we assumed:

• PAIS emit relatively small set of highly structured events.
• Information is structured based on tasks.
• Matching tasks to rules is based on static event properties.
• Conditions compare static information of different task.
• Actions are simple, and trigger no cascading rules.

We thus conclude, that we need:

(1) a simple tree-based data structure to gather all event data,
(2) to design an easy to understand and maintain** rule format.

---


** rule bases become hard to understand maintain as they grow.
Process Activity Life-Cycle

For every activity

For the whole process
Data Structure

- Instance UUID
- Activity ID
- Source System
- Event Topic
- Event Name
- Event Data

- fc4fd4b6-e106-4dac-bc42-bdbd82996286:
  - :ai:
  - :engine:
  - :activity:
    - :calling:
      - endpoint: http://coms.wst.univie.ac.at:9300
        parameters:
          label: "OK OR NOT OK"
          method: post
          arguments:
            orgmodel: organisation1
            domain: Virtual Business 1
            form: http://coms.wst.univie.ac.at/form/form-f.html
            time: 744055584018421247/500000000
  - :done:
    - endpoint: http://coms.wst.univie.ac.at:9300
      time: 744055591203937709/500000000
    - :dataelements:
      - :change:
        - endpoint: http://coms.wst.univie.ac.at:9300
          changed:
            comment: "some comment"
            time: 186013897800974363/125000000
    - :worklist:
      - :user:
        - :fake:
          - user: weiss9
            callback_id: fbae8679569a23586996439c4a75f3c6
            domain: Virtual Business 1
            cpee_label: "OK OR NOT OK (63)"
            orgmodel: http://cpee.org/~demo/orgviz/organisation_informatik.xml
            organisation:
              Workflow Systems and Technology:
              - Regular
              - Staff
            wl_instance: http://coms.wst.univie.ac.at:9300/Virtual Business 1
Rules - Overview

<table>
<thead>
<tr>
<th>match:</th>
<th>Definition relevant process activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>match block</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>condition:</th>
<th>Conditions which apply to the activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>condition block</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>if:</th>
<th>Actions which are taken if the defined conditions evaluate as true or false</th>
</tr>
</thead>
<tbody>
<tr>
<td>[]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ifnot:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[cpee_stop]</td>
<td></td>
</tr>
</tbody>
</table>
Rules - Matching

A task has multiple attributes. These attributes determine the behavior of the task as well and may alter the controlflow. Keys describe the possible input, values the outcome.

A Web service is called to automatically check the provided information. E.g. check consistency of provided information.

The information is then reviewed manually.
Rules - Matching

match:
  - taskA :
    - [ "engine > activity > calling > endpoint", 
      "==", 
      "https://someservice.com"],
    - [ "engine > activity > receiving > parameters", 
      "include?", 
      "checked"]

  - taskB :
    - [ "engine > activity > calling > endpoint", 
      "==", 
      "https://someotherservice.com"],
    - [ "engine > activity > receiving > parameters > form", 
      "==", 
      "https://someotherservice.com/forms/form-f"]
Actions

- Actions are defined separately to the rules.
- They can be used to remediate compliance violations or to notify other services.
- Possible remediations:
  - Stop the executions, giveback a user task, assign a user task, restart an instance, ...
- Values can be shared between actions (blackboard approach).
- Example:

  - :cpee_stop
  - :method : 'put'
  - :url : ["engine > activity > calling > instance", "/properties/values/state"]
  - :data :
    - value : "stopping"
Actions - Blackboard Approach

```python
condition:
[  
    - [  
        "taskB > engine > activity > receiving > received > damage", "exits?"
    ]
]  
if:      [blackboardize_damage_value]
ifnot:   []

def blackboardize_power_value(ctx):
    value = ctx.engine > activity > receiving > received > damage
    ctx.blackboard.values.last_value = value
end
```
Evaluation - Real World Use-Case

Process spanning processing monitoring:

- Different parts are produced, each with its own process model.
- Different sensors in an EMCO “MaxxMill 500” milling machine are monitored through process instances running in parallel.

Purpose - predict and notify about tool changes:

- Monitoring deviations in the power-consumption during the milling.
- Detect when a milling tool breaks or the edge of a milling tool rapidly deteriorates.

Use-Case & Results:

- 1624 events.
- 2 simple rules allow to reliably detect breaks and deterioration.
- Process models can remain unchanged, the semantics of monitoring for tool failure can be kept out of the process models.

GitHub: https://github.com/pakoe/coms
Possible Performance Gains

<table>
<thead>
<tr>
<th>Prolog</th>
<th>Drools</th>
<th>Jessy</th>
<th>COMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tabling</td>
<td>Search &amp; Backtrack</td>
<td>Rete</td>
<td>Lazy Rete</td>
</tr>
<tr>
<td>W*C</td>
<td>W^C</td>
<td>W^C</td>
<td>&lt; W^C</td>
</tr>
</tbody>
</table>

The matchset is denoted as productions C
Event data is denoted as working memory W

Given the assumption that PAIS requirements can be satisfied by simple events and rules, a simpler rule engine with better performance, is sufficient.

GitHub: https://github.com/pakoe/coms
Conclusion

- Rule-based solutions fail because they are hard to maintain.
- It is important to focus on the usability of rule-bases.
- Highly structured event data + simple rules are sufficient to cover a wide array of compliance rules.
- Performance gains are possible.

Who are We? Patrik König, Jürgen Mangler, Stefanie Rinderle-Ma
Organisation? University of Vienna
What’s the title of the paper? Compliance Monitoring on Process Event Streams from Multiple Sources.