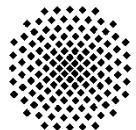


Process Views to Support Compliance Management in Business Processes

11th International Conference on
Electronic Commerce and Web Technologies
September 02, 2010

David Schumm, Frank Leymann, Alexander Streule
Institute of Architecture of Application Systems



Universität Stuttgart
Germany

{lastname}@iaas.uni-stuttgart.de

Overview

- Introduction and Background
 - Compliance
 - Compliance fragments
 - Process views
- Managing Compliance Fragments with Process Views
 - Extraction
 - Highlighting
 - Hiding
- Implementation Aspects: Extracting and Hiding
 - Architecture
 - Implementation
- Conclusion and Outlook

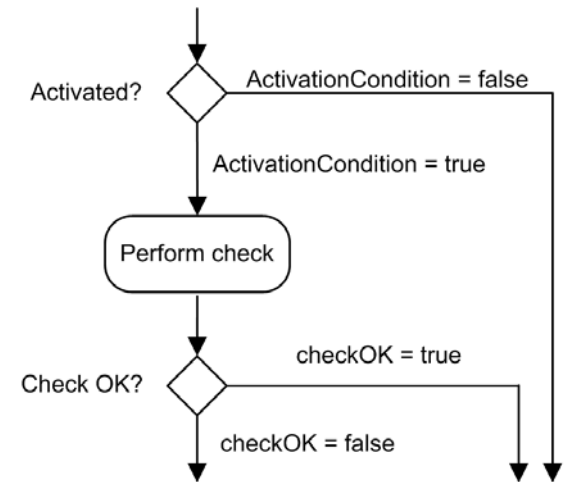
Introduction and Background

Compliance

- Organizations have to cope with an increasing number of complex laws, regulations etc.
- Compliance denotes if an organization adheres to the requirements derived therefrom
- Compliance management ensures that business processes are in accordance with a set of prescribed requirements
- Compliance has a profound impact on the business process management life cycle

Process Fragments for Compliance

- A process fragment is a connected, possibly incomplete process graph
- Not necessarily directly executable
- A process fragment is made up of
 - Activities
 - Activity placeholders (so-called regions)
 - Control edges defining control dependencies
 - Multiple incoming and outgoing control edges
- A *compliance fragment* is a process fragment which realizes compliance requirements regarding a process



Challenges of Managing Compliance Fragments

- **Extraction**
 - Goal: Create reusable compliance fragments
 - Solution: Extract them from a process in which they are contained
- **Integration**
 - Goal: Make a given process compliant with requirements
 - Solution: Integrate a corresponding compliance fragment
- **Highlighting**
 - Goal: Proof compliance to an auditor
 - Solution: Highlight compliance fragments contained in a process
- **Hiding**
 - Goal: Show a clear view on an “unpolluted” process
 - Solution: Hide compliance fragments contained in a process
- *Process views* provide a means for extraction, highlighting and hiding

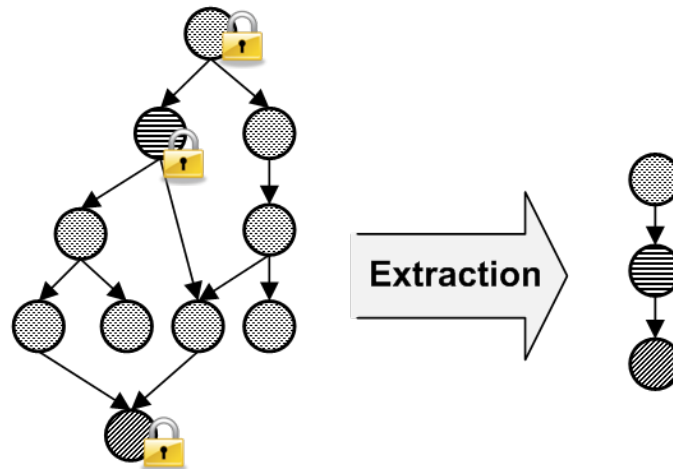
Process Views

- What is a process view...?
 - The presentation of the result from specific transformations applied to a process model
- What is the purpose of a process view...?
 - Reduction of complexity
 - Abstraction from undesired details
 - Separation of concerns
 - Providing a user-specific perspective
- What are the basic view transformations...?
 - Omission of structures
 - Aggregation of structures
 - Change of the visualization of structures
- And: views can be used to manage compliance fragments

Managing Compliance Fragments with Process Views

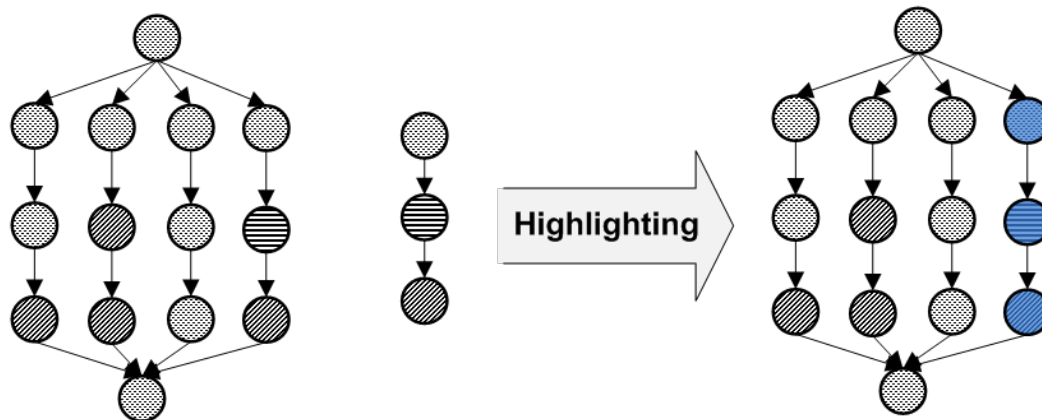
Compliance Fragment Extraction

- Some parts of a process related to compliance are *reusable* and therefore shall be extracted
- Example: “checking up on appointment conflicts”
- Proposed solution
 - Creation of a view which only contains the process structures which shall be extracted
 - The structures which shall be extracted need to be tagged
 - In a next step all other activities are omitted
 - The resulting compliance fragment can be stored in a repository
 - Problem: dealing with ambiguities



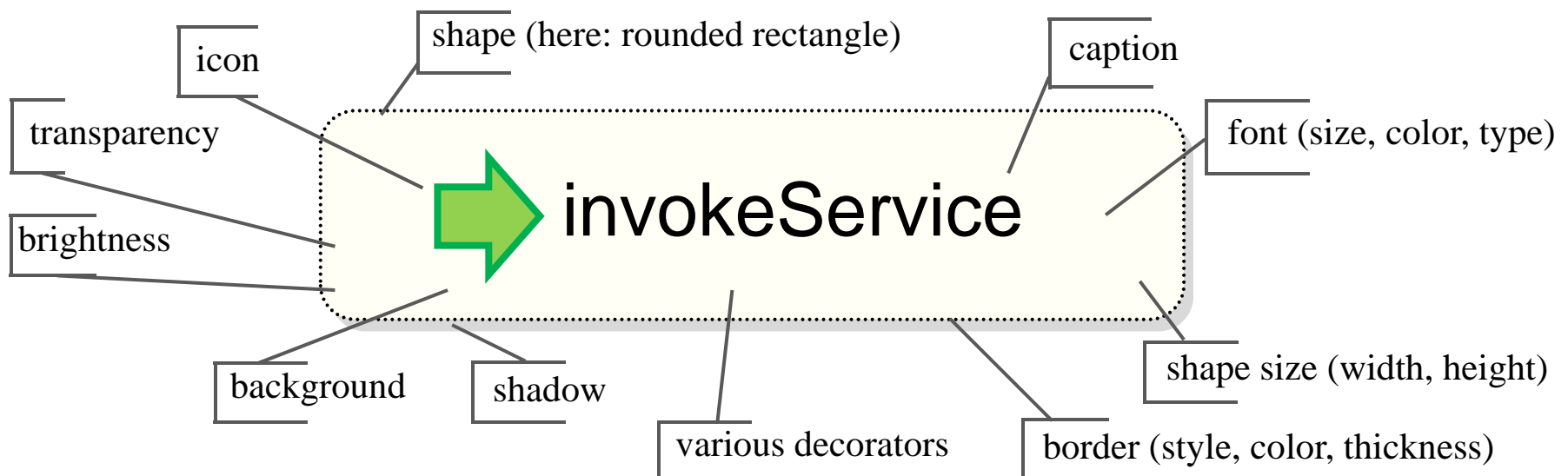
Compliance Fragment Highlighting

- For an *audit* only particular structures are of interest and need to be highlighted
- Example: How is “checking up on appointment conflicts” realized in this process?
- Proposed solution
 - Recognition of contained compliance fragments
 - Binding of the visualization to the result of the fragment recognition



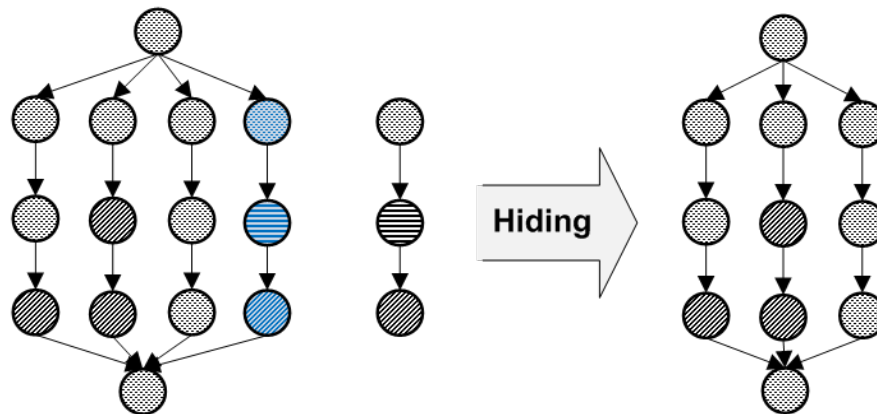
Changing the Visualization for Highlighting

- Predefined paint methods of the process constructs can be modified
- Several display properties of activities can be modified to provide the highlighting



Compliance Fragment Hiding

- A large number of compliance fragments possibly *pollute* a process model
- Example: checking up on appointment conflicts, informing the supervisor, checking for ...
- Proposed solution
 - Omission of process structures matching a particular predicate P
 - P can be based on an identifier, a tag, the activity name etc.
 - Compliance fragments are used to generate rules for their omission

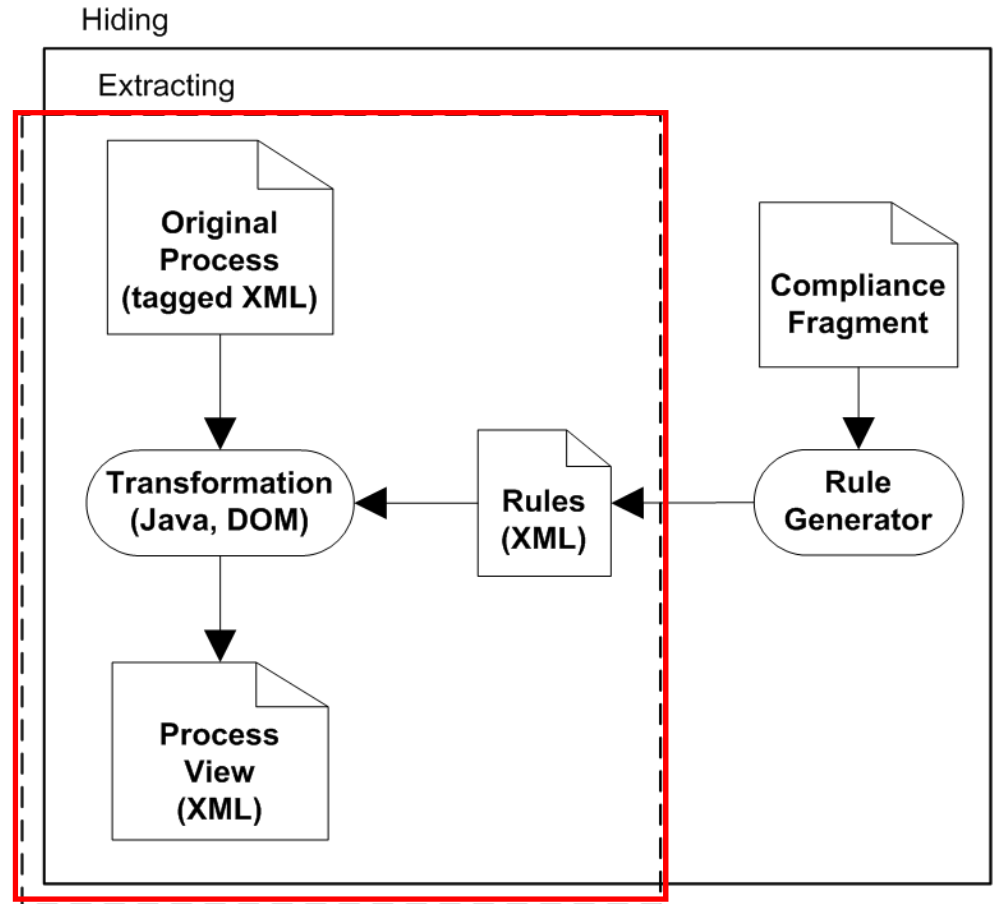


Implementation Aspects: Extracting and Hiding

Principle of the Viewing Application

■ Extracting

- Structures that should be extracted have to be tagged for preservation
- Transformation rules have to be specified
- For extraction, these rules are always the same



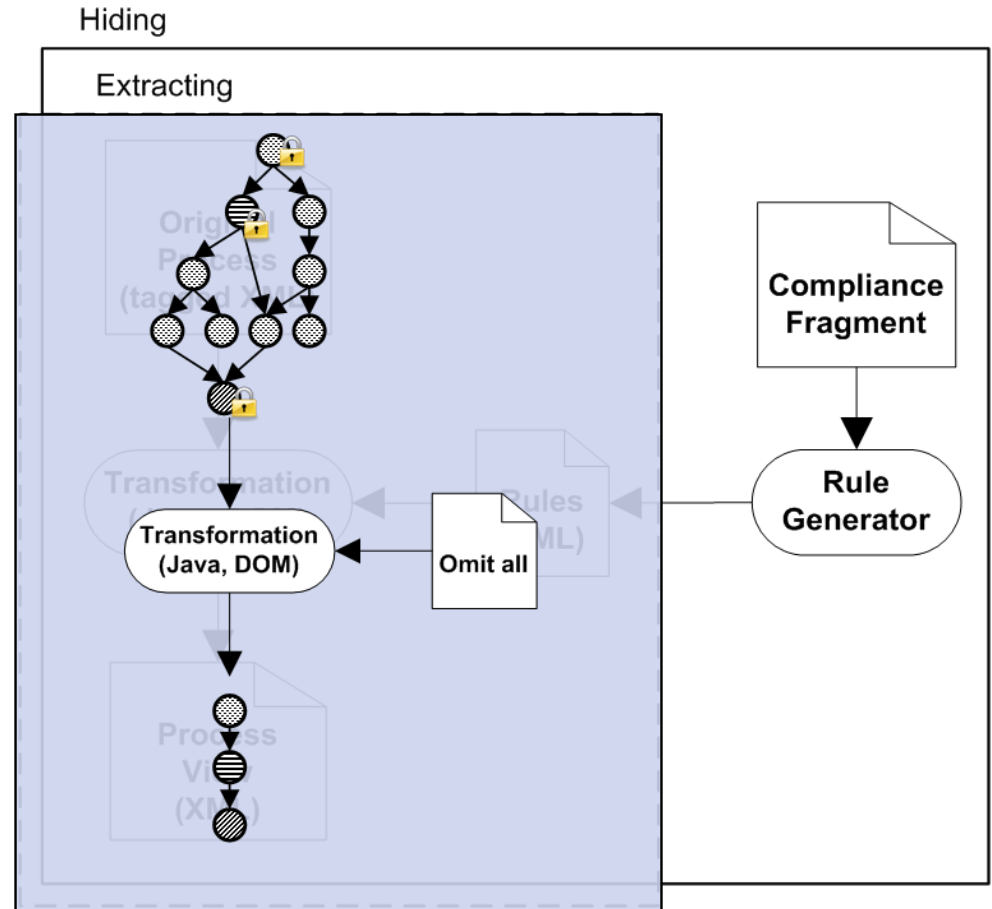
Principle of the Viewing Application

■ Extracting

- Structures that should be extracted have to be tagged for preservation
- Transformation rules have to be specified
- For extraction, these rules are always the same

■ Walkthrough

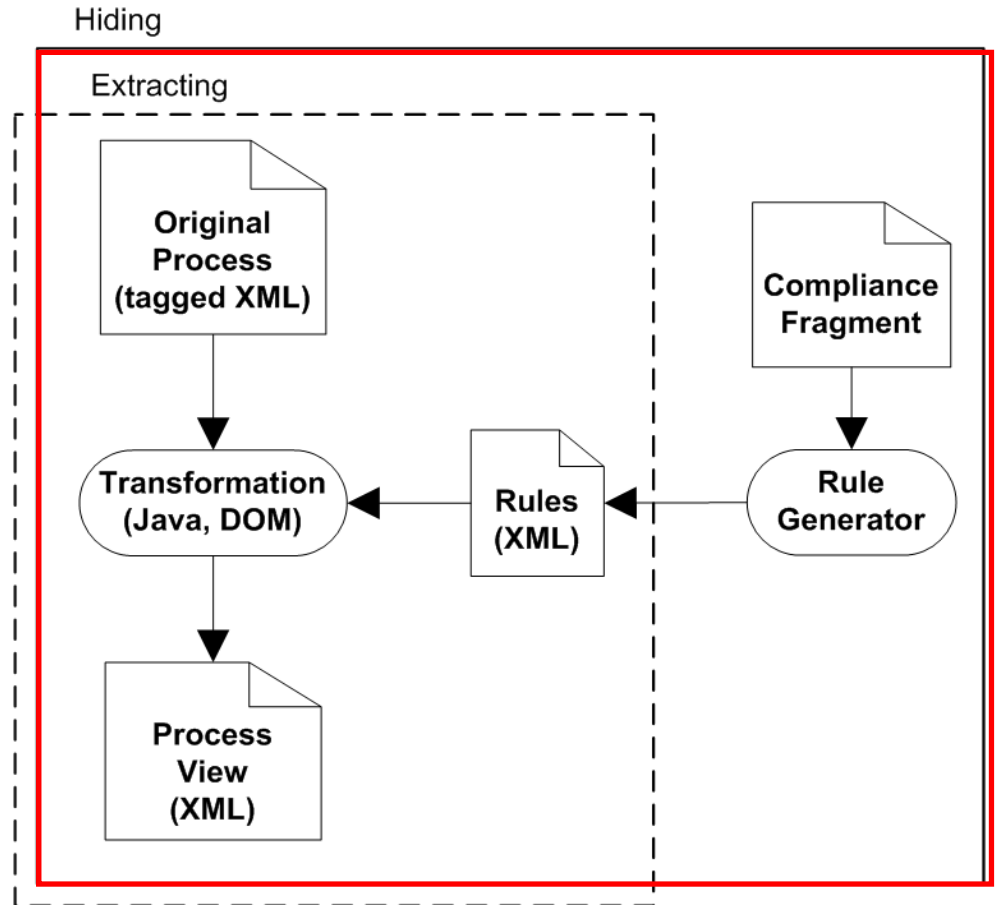
- Checking up on appointment conflicts



Principle of the Viewing Application

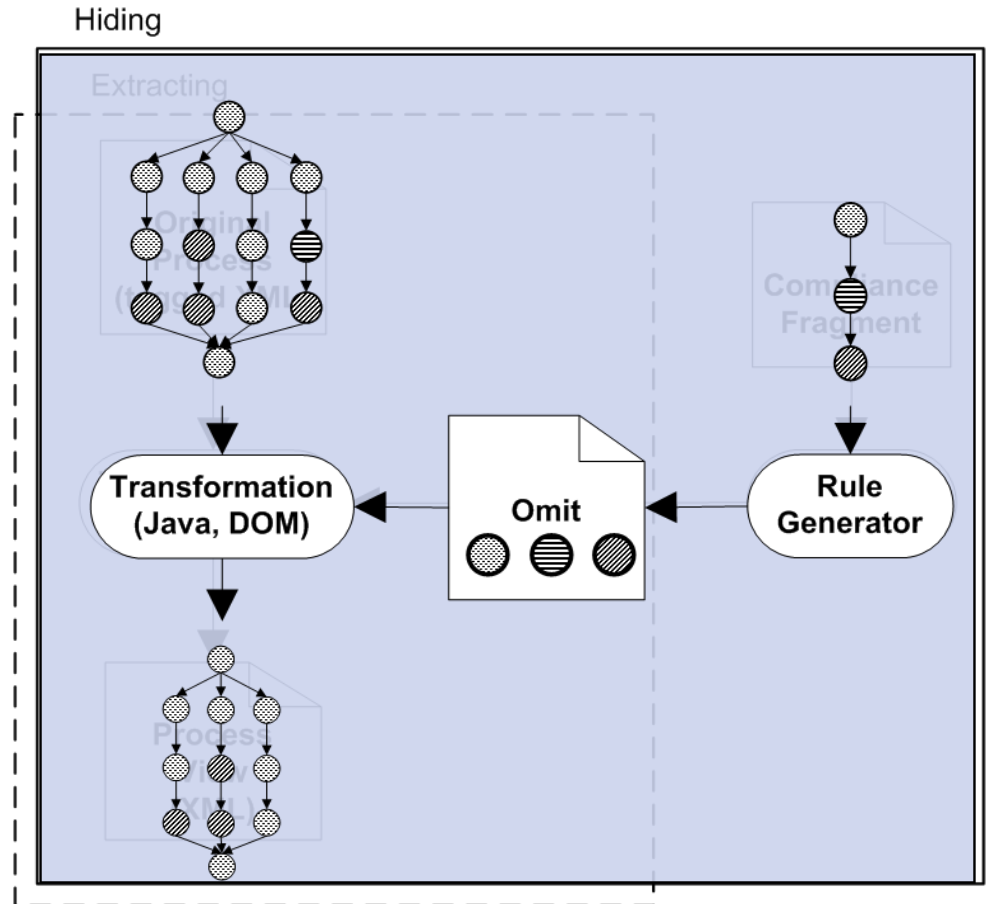
■ Hiding

- The rule generator takes a fragment as input
- Rules for the omission are generated automatically



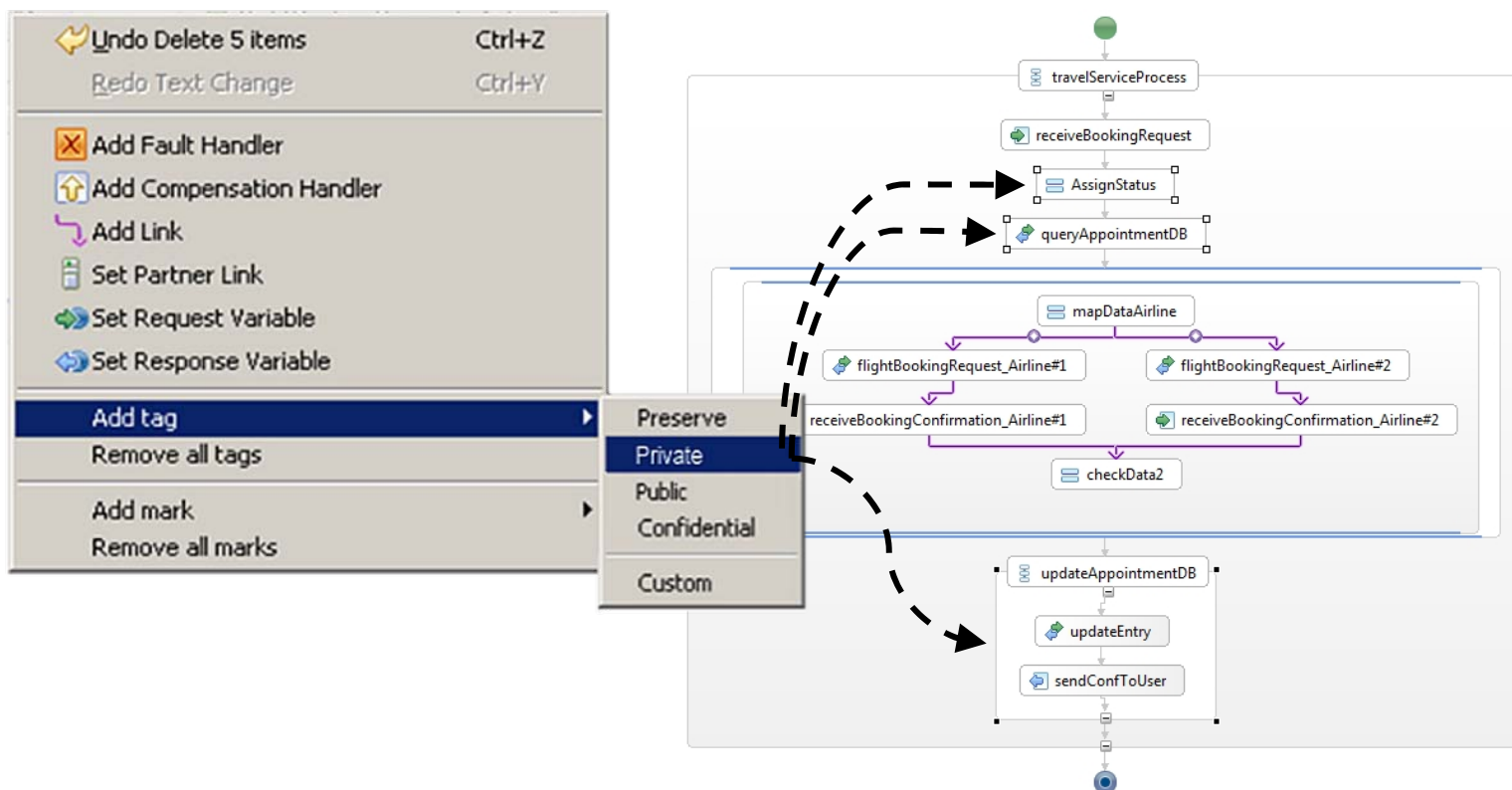
Principle of the Viewing Application

- **Hiding**
 - The rule generator takes a fragment as input
 - Rules for the omission are generated automatically
- **Walkthrough**
 - Checking up on appointment conflicts




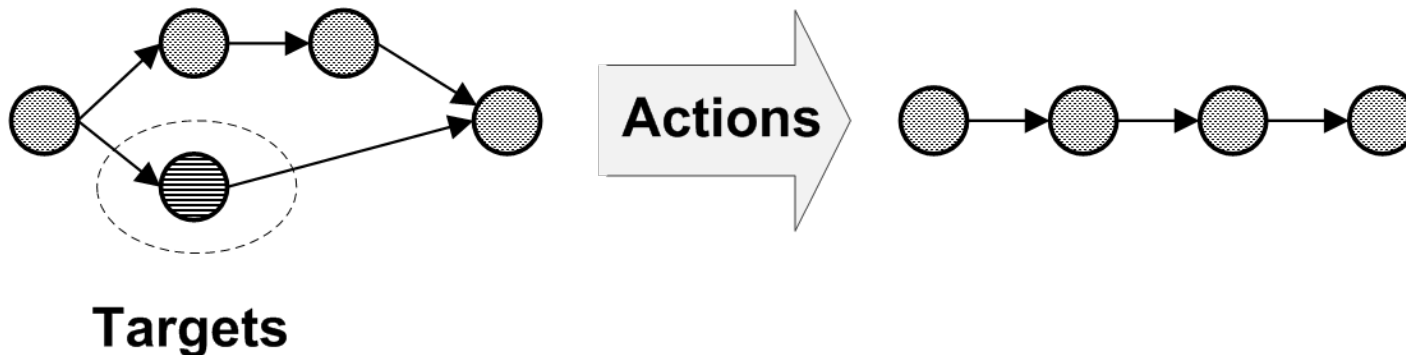
Annotation of the Process Model

- Tagging eases the definition of transformation rules
- Rules can be based on the annotations
- If no annotations are available, then construct attributes like name, portType etc. have to be used to select *targets*



Transformation Language

- **Targets** indicate the constructs that should be transformed
 - tag: Targeting based on annotations (annotated with )
 - attribute: Targeting based on name-value pairs of attributes
 - type: Targeting based on the XML element type
 - Logical connectors to formulate complex selection statements
- **Actions** denote which transformations have to be applied
 - actionOmit: Omission of an arbitrary construct
 - actionOpaque: Transformation of an activity into an opaque activity
 - actionSetAttributeTo: Changing an arbitrary attribute value



Transformation Specification

■ Extraction

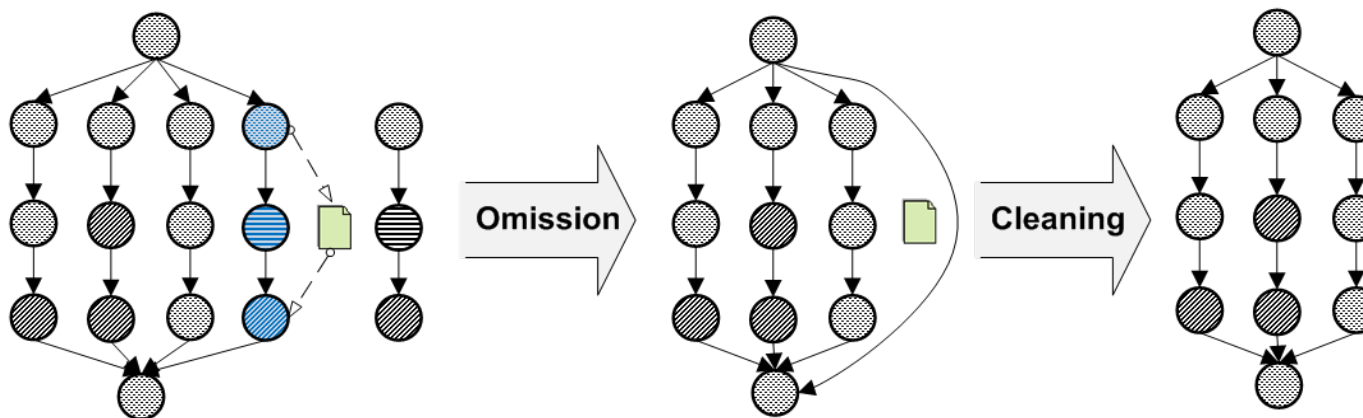
```
<rules>
  <rule name="extractFragment" apply="true">
    <actions>
      <actionOmit preserveChildren="true"/>
    </actions>
    <targets>
      <not>
        <tag tagName="preserve" />
      </not>
    </targets>
  </rule>
</rules>
```

■ Hiding

```
<rule name="omit%CONSTRUCT-NAME%" apply="true">
  <actions>
    <actionOmit preserveChildren="true"
      preserveTransitionConditions="false"/>
  </actions>
  <targets>
    <attribute attributeName="name"
      value="%CONSTRUCT-NAME%" />
  </targets>
</rule>
```

Transformation Execution

- For each rule
 - Evaluate the set of targets
 - Apply actions to this set
- Transformation actions
 - setOpaque and setAttributeTo are straightforward
 - In actionOmit amiguities arise
- After transformation
 - Removing unused artifacts
 - Removing redundant control structures



Conclusion and Outlook

Limitations of the Approach to Manage Compliance

- Compliance of business processes
 - Compliance fragments are capable of addressing compliance requirements related to control flow and activities *within* a process
 - A compliance fragment cannot ensure that people who are involved in the process will behave compliant
 - A compliance fragment cannot control the applications and services which it orchestrates
- Compliance requirements have an impact on *all* components in the IT infrastructure
- Compliance also has an impact on the business processes which run *outside* of the IT systems
- Compliance fragments are just one aspect of control in an overall solution to compliance management

Conclusion and Outlook

- Compliance is an important topic in research and in industry
- We proposed techniques for extraction, highlighting and hiding of structures to support compliance in BPM
- Ongoing work regarding compliance fragments
 - Development of a wizard for integration of compliance fragments into processes
 - BPEL extensions for compliance fragments
 - Development of a view designer to ease view specification
- Final results of the COMPAS research project
 - Available at the beginning of 2011
 - Downloads at www.compas-ict.eu

End of Document