UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

Business Process Modelling with Continuous Validation

Stefan Kühne, Heiko Kern, Volker Gruhn, Ralf Laue
University of Leipzig, Germany
Department of Business Information Systems
Chair of Applied Telematics / e-Business

Continuous Validation

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Continuous Validation - What does it mean?
- The term has been borrowed from "continuous testing" / "continuous compilation"

Programming with Modern IDEs

Errors and warnings are generated immediately,

Compiler runs in background

Often, suggestions for fixing a problem automatically are generated.

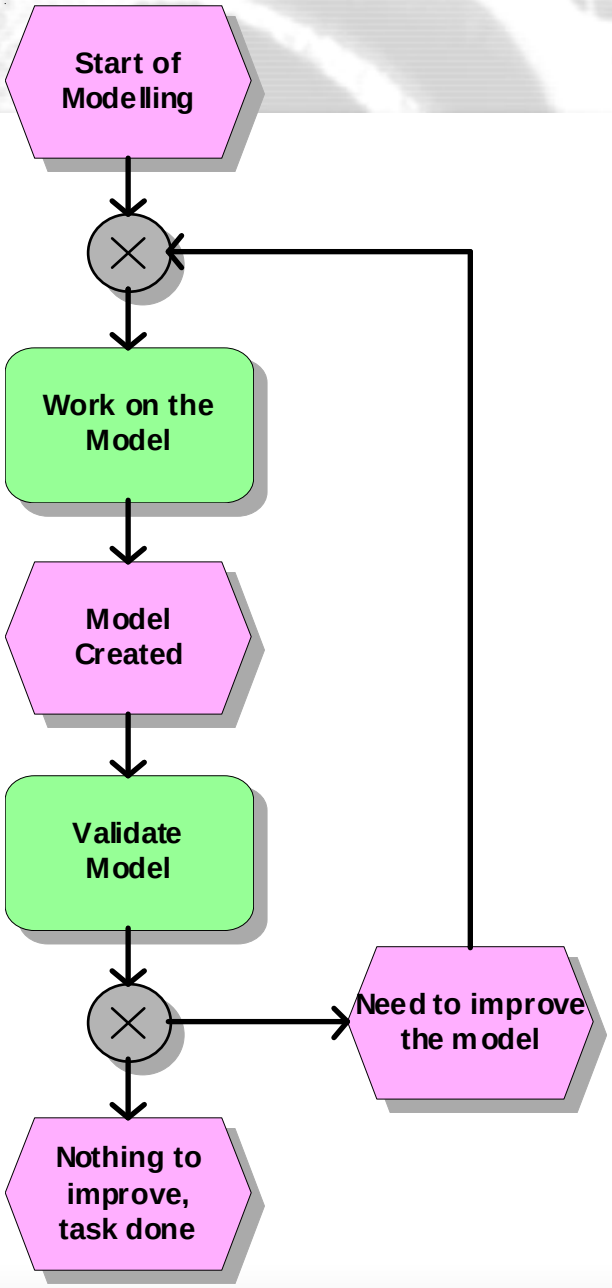
Goal

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

To build a prototype of a BPM editor with immediate feedback about

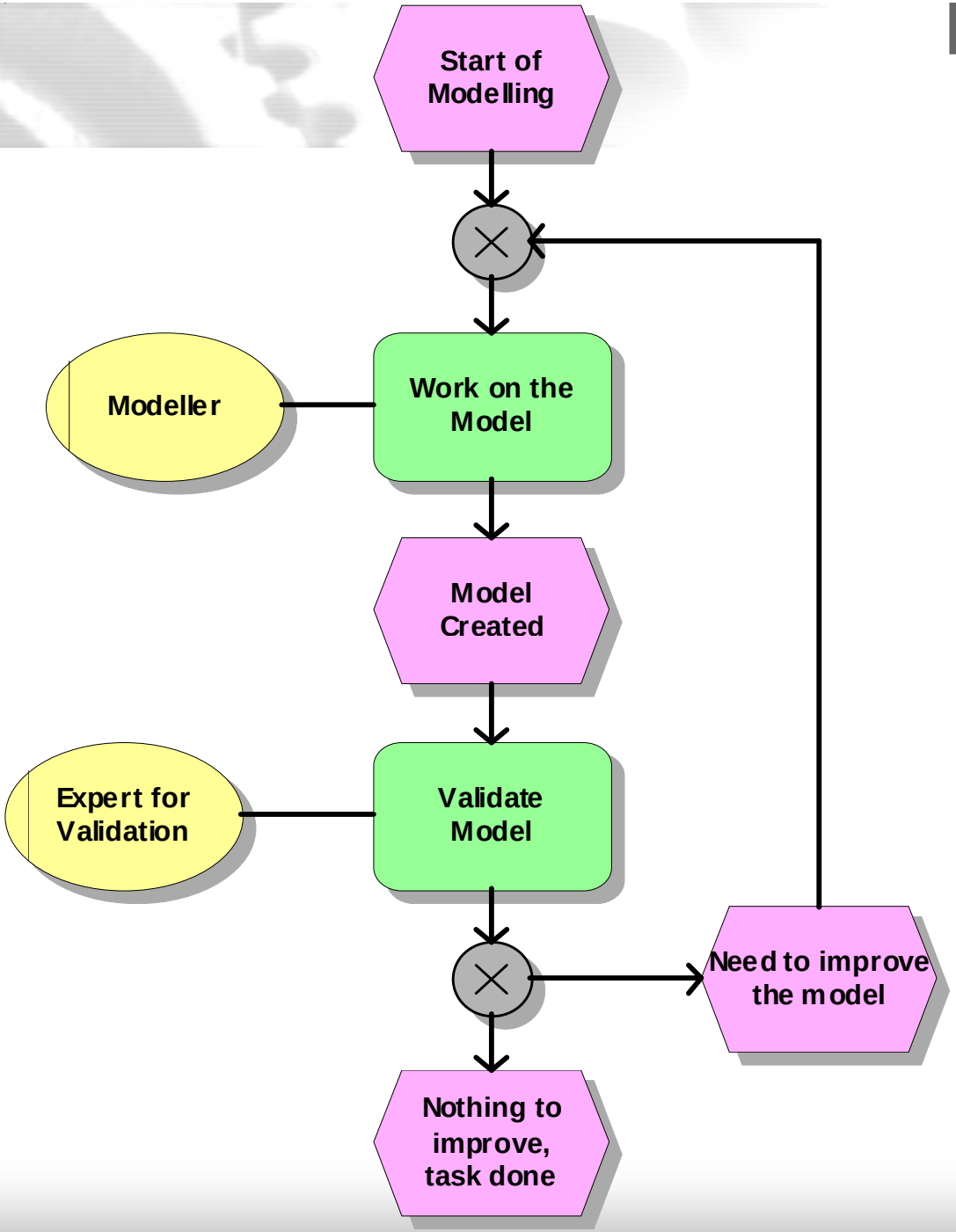
- syntactical errors
- execution problems (deadlocks, multiple termination)
- "bad style"

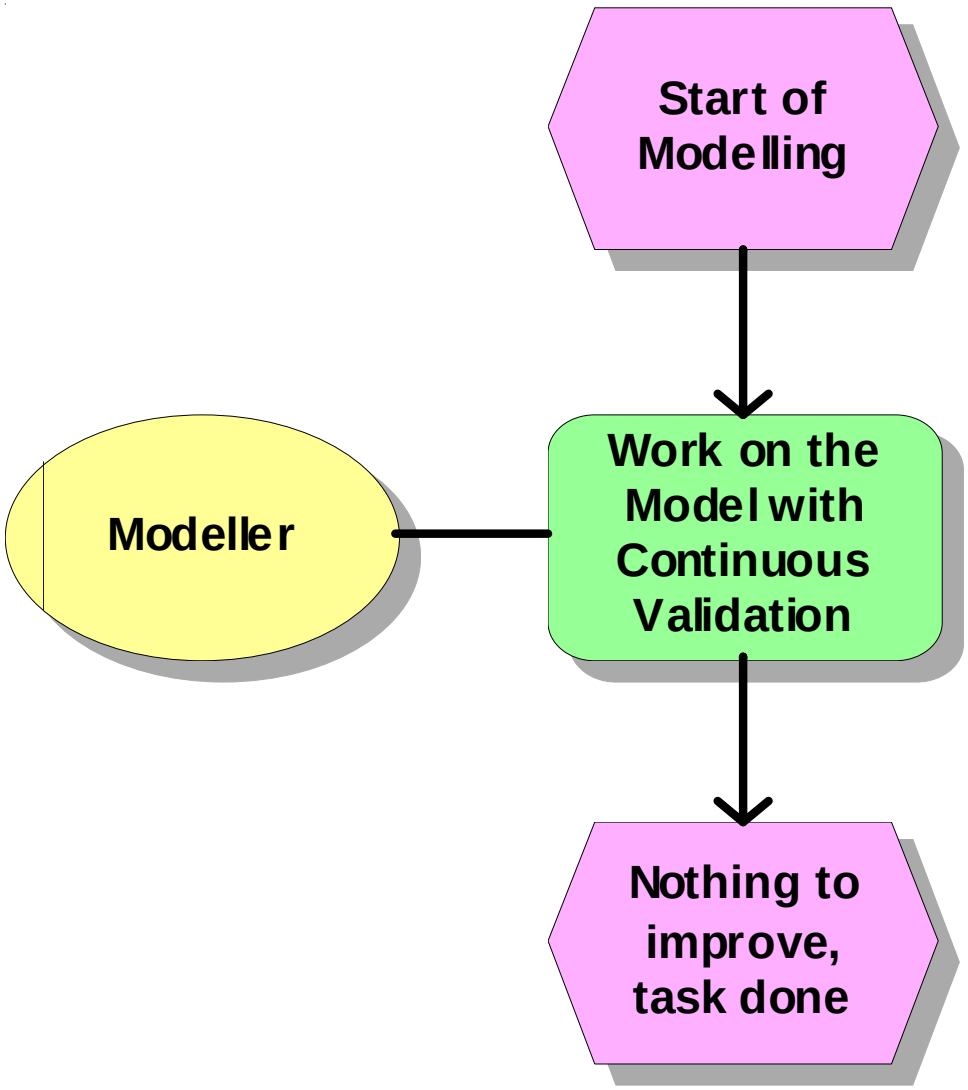


⊗ alternative execution (XOR-connectors)

∧ parallel execution (AND-connectors)

∨ OR-connectors: model parallel execution of one or more flows


Start of Modelling
Modeller
Work on the Model
Model Created
Expert for Validation
Validate Model
Need to improve the model
Nothing to improve, task done


Start of Modelling
Modeller
Work on the Model with Continuous Validation
Nothing to improve, task done

Known Validation Methods

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

(a) Give the model a formal semantics and explore the state space.

(b) Repeatedly reduce well-structured parts of the model
Model is correct, if it can be reduced to a single element

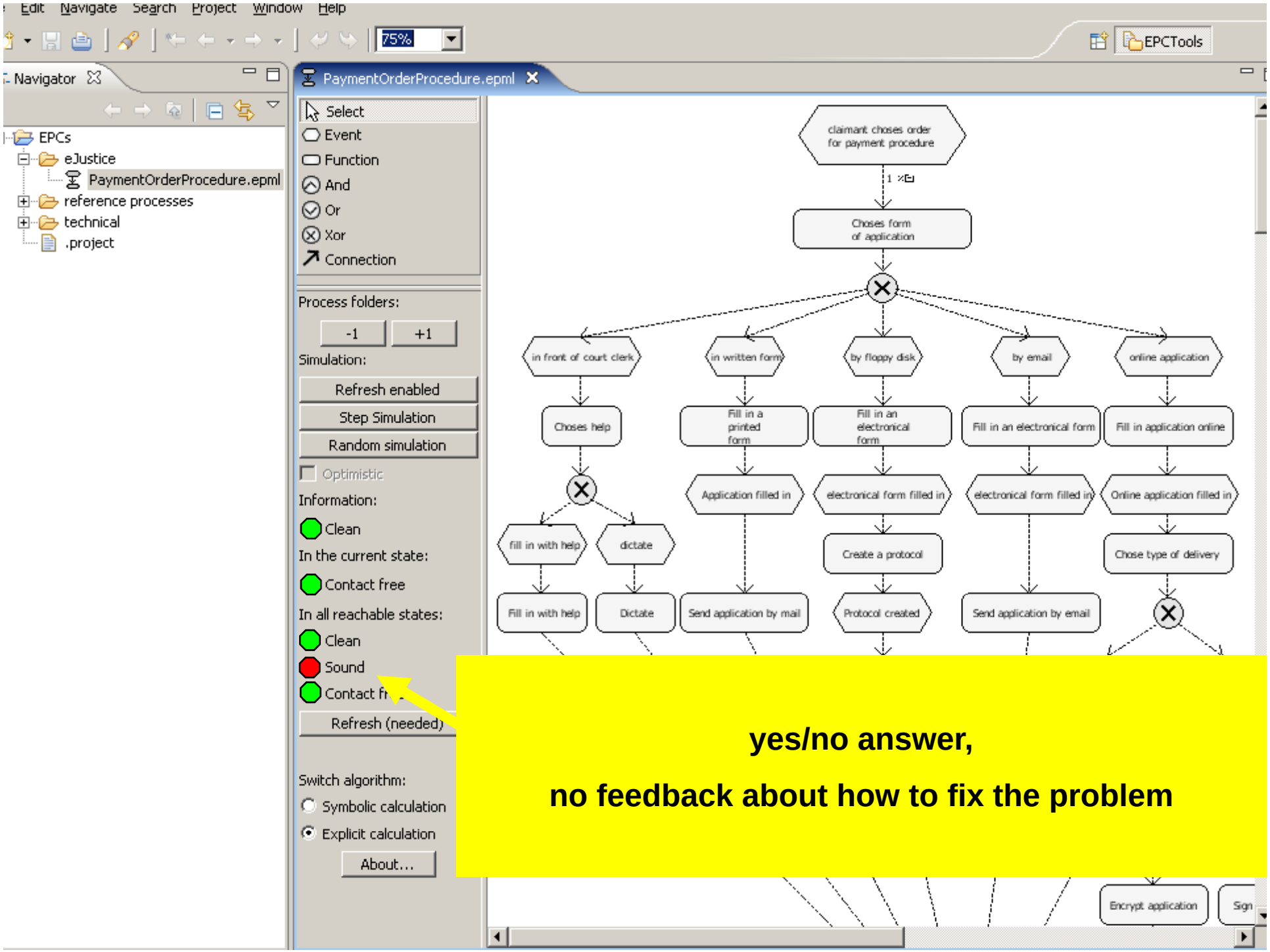
(c) Look for structural error patterns,
Heuristic approaches

(a) Explore the State Space
(b) Reduction Algorithms

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Problem 1:
  An error can be found but not its reason


Edit Navigate Search Project Window Help
75%
EPCTools
Navigator
EPCs
eJustice
PaymentOrderProcedure.epml
reference processes
technical
.project
PaymentOrderProcedure.epml
Select
Event
Function
And
Or
Xor
Connection
Process folders:
-1
+1
Simulation:
Refresh enabled
Step Simulation
Random simulation
Optimistic
Information:
Clean
In the current state:
Contact free
In all reachable states:
Clean
Sound
Refresh (needed)
Switch algorithm:
Symbolic calculation
Explicit calculation
About...
claimant choses order for payment procedure
Choses form of application
in front of court clerk
in written form
by floppy disk
by email
online application
Choses help
Fill in a printed form
Fill in an electronical form
Fill in an electronical form
Fill in application online
Application filled in
electronical form filled in
electronical form filled in
Online application filled in
fill in with help
dictate
Create a protocol
Chose type of delivery
Fill in with help
Dictate
Send application by mail
Protocol created
Send application by email
Encrypt application
Sign
yes/no answer,
no feedback about how to fix the problem

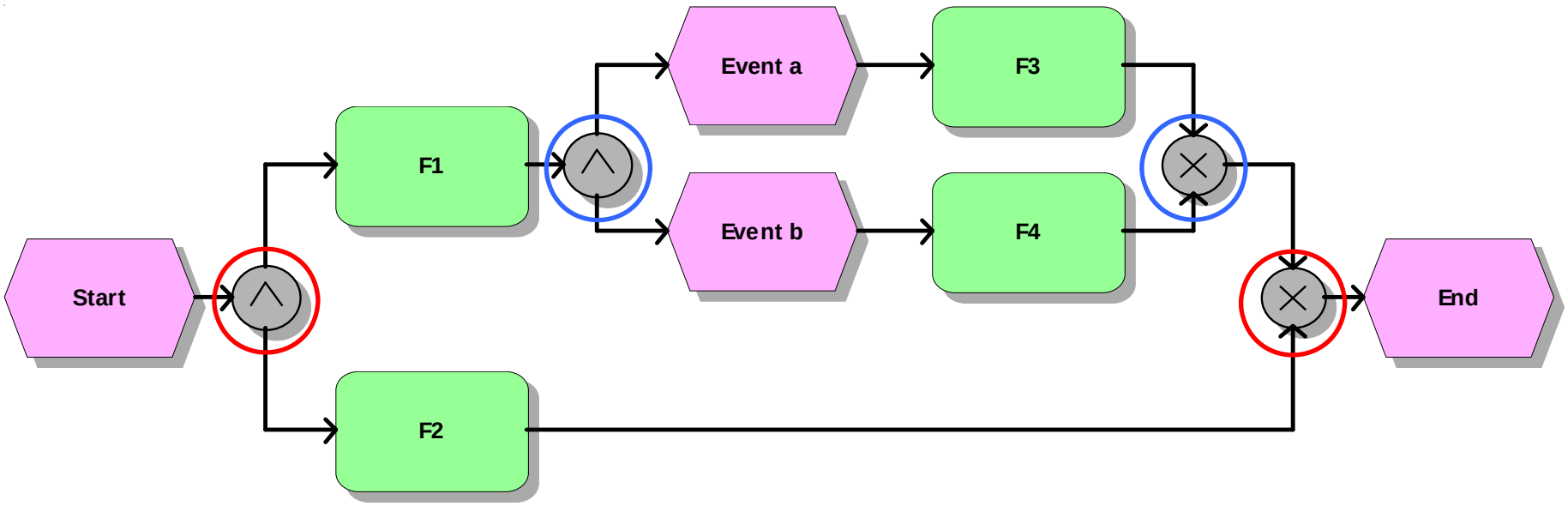
(a) Explore the State Space
(b) Reduction Algorithms

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Problem 1:
  An error can be found - but not its reason
- Problem 2:
  Even if the reason of the error can be identified, some errors remain undetected.

Example: One of two Problems?



(a) Explore the State Space
(b) Reduction Algorithms

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Problem 1:
  An error can be found - but not its reason
- Problem 2:
  Even if the reason of the error can be identified, some errors are ignored
- Problem 3:
  State Space Explosion

(a) Explore the State Space
(b) Reduction Algorithms

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

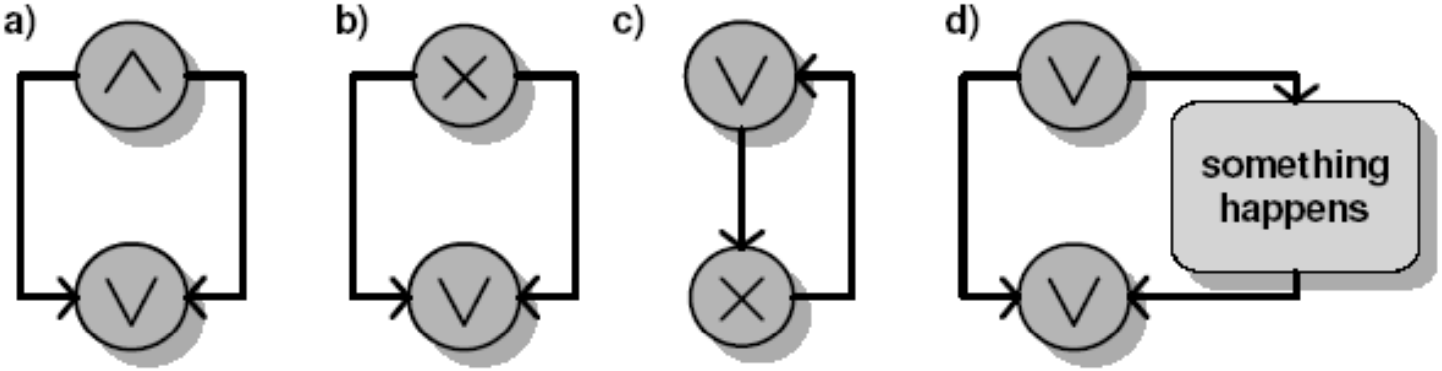
- Problem 1:
  An error can be found - but not its reason
- Problem 2:
  Even if the reason of the error can be identified, some errors are ignored
- Problem 3:
  State Space Explosion (for (a))
- Problem 4:
  Modelling style is out of the scope

Modelling Style

UNIVERSITÄT LEIPZIG

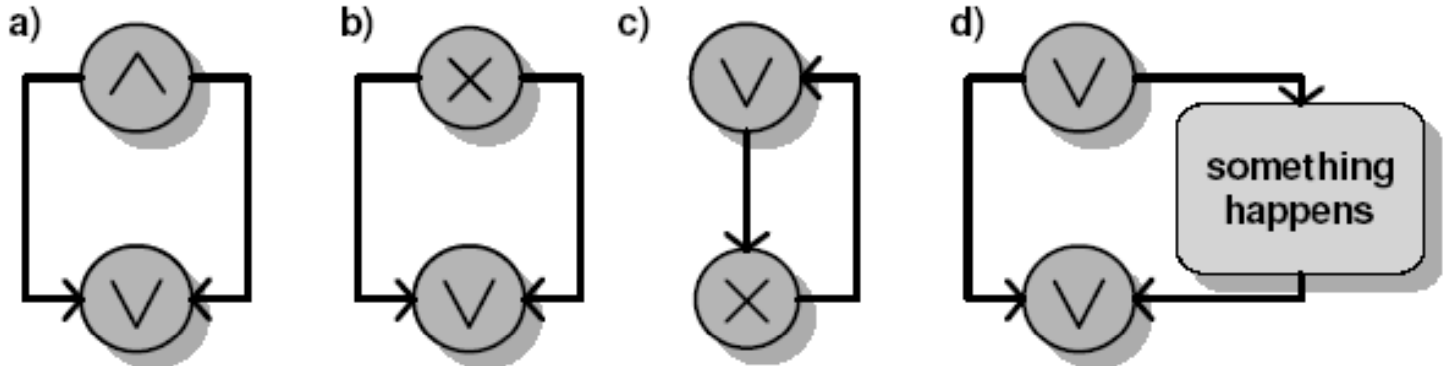
LPZ E-BUSINESS applied telematics

- All these model fragments are absolutely correct (from a theoretical point of view):

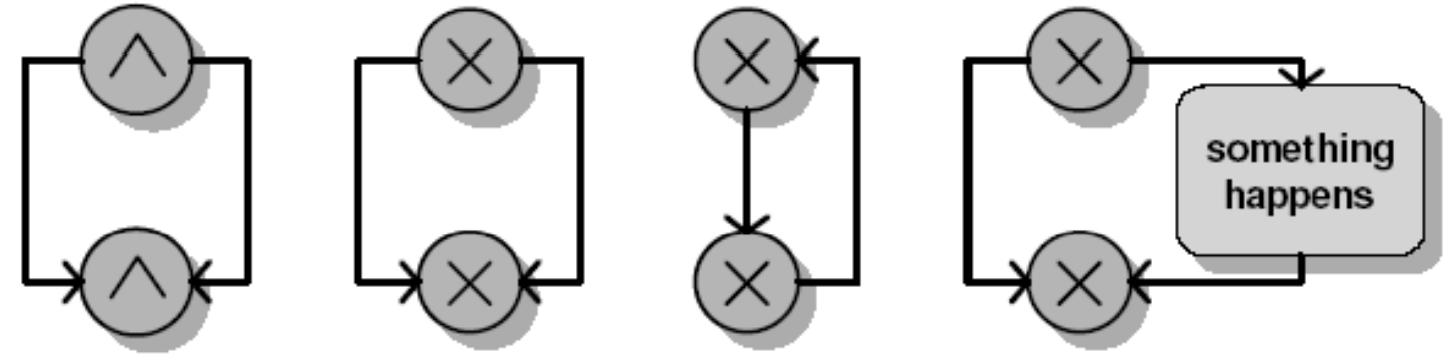


Modelling Style

- All these model fragments are absolutely correct (from a theoretical point of view):



but should be coorected anyway to:



Syntactical Correct UML

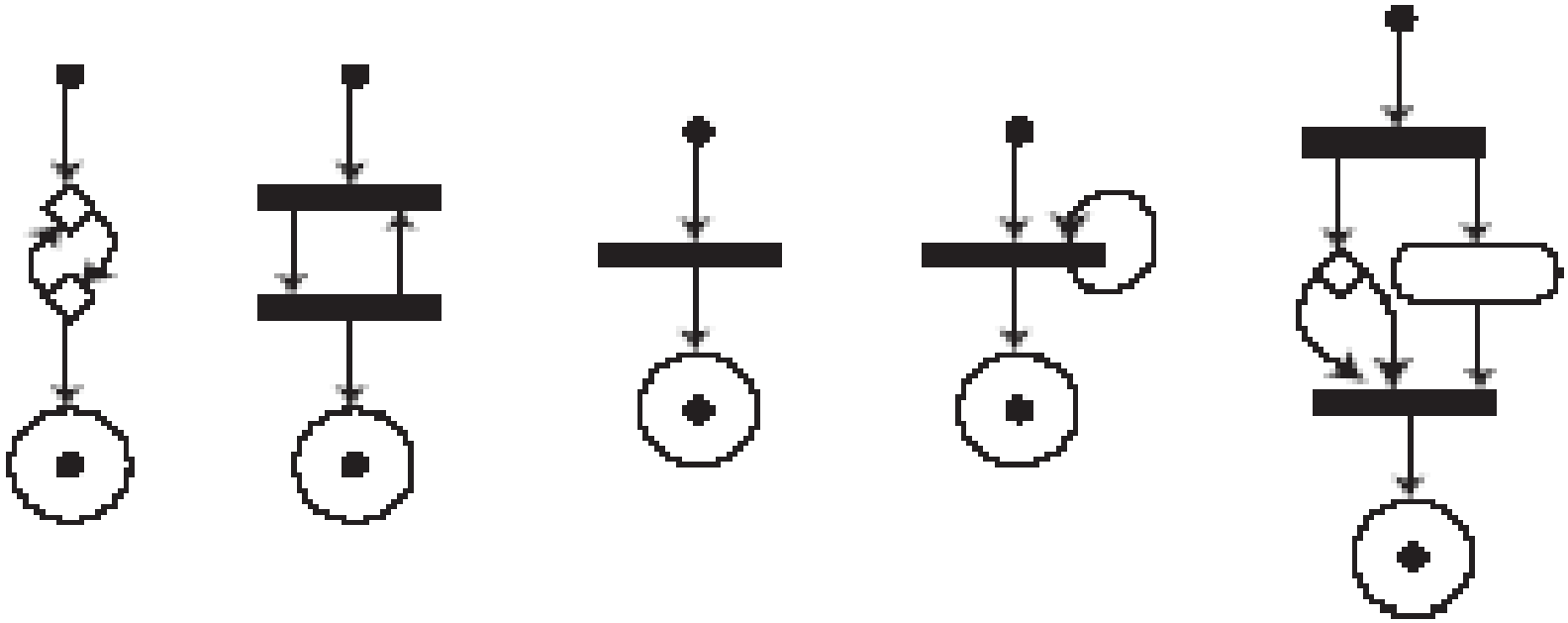
UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

An UML AD example: syntactically correct...

but absolutely useless

→should be detected by an editor



source: Harald Störrle. Semantics of UML 2.0 Activities

(a) Explore the State Space
(b) Reduction Algorithms

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Problem 1:
  An error can be found - but not its reason
- Problem 2:
  Even if the reason of the error can be identified, some errors are ignored
- Problem 3:
  State Space Explosion (for (a))
- Problem 4:
  Modelling style is out of the scope
- Problem 5:
  Most approaches cannot be used for incomplete models

Our Tests

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Syntactical Correctness
  (beyond the metamodell)
- Error patterns for deadlocks and multiple termination
- Modelling style

Heuristic Approach

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Our choice:
- Heuristic Approach: We try to find common error patterns.

- Validation needed:
  How good does this approach work, compared with the formal methods mentioned before?

Validation

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Reference:
- Jan Mendling: Detection and Prediction of Errors in EPC Business Process Models (PhD thesis)

- uses reduction rules and Petri-net analysis for locating errors in EPC models
- 604 models from the SAP R/3 Reference Model:
- Reduction rules found 178 error patterns in 90 models
- 57 models could not be reduced completely
  Petri-net analyzer:
  36 unsound
  21 sound

Results

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Reduction rules found 178 error patterns in 90 models

  Our approach found 176 of them.

  The remaining two models are sound when another formal semantics is assumed.

Validation

UNIVERSITÄT LEIPZIG

LPZ E-BUSINESS applied telematics

- Reduction rules found 178 error patterns in 90 models

  Our approach found 176 of them.

  The remaining two models are sound when another formal semantics is assumed.

- 57 models could not be reduced completely
  Petri-net analyzer:
  36 unsound
  We found at least one error in all these models

Validation

UNIVERSITÄT LEIPZIG

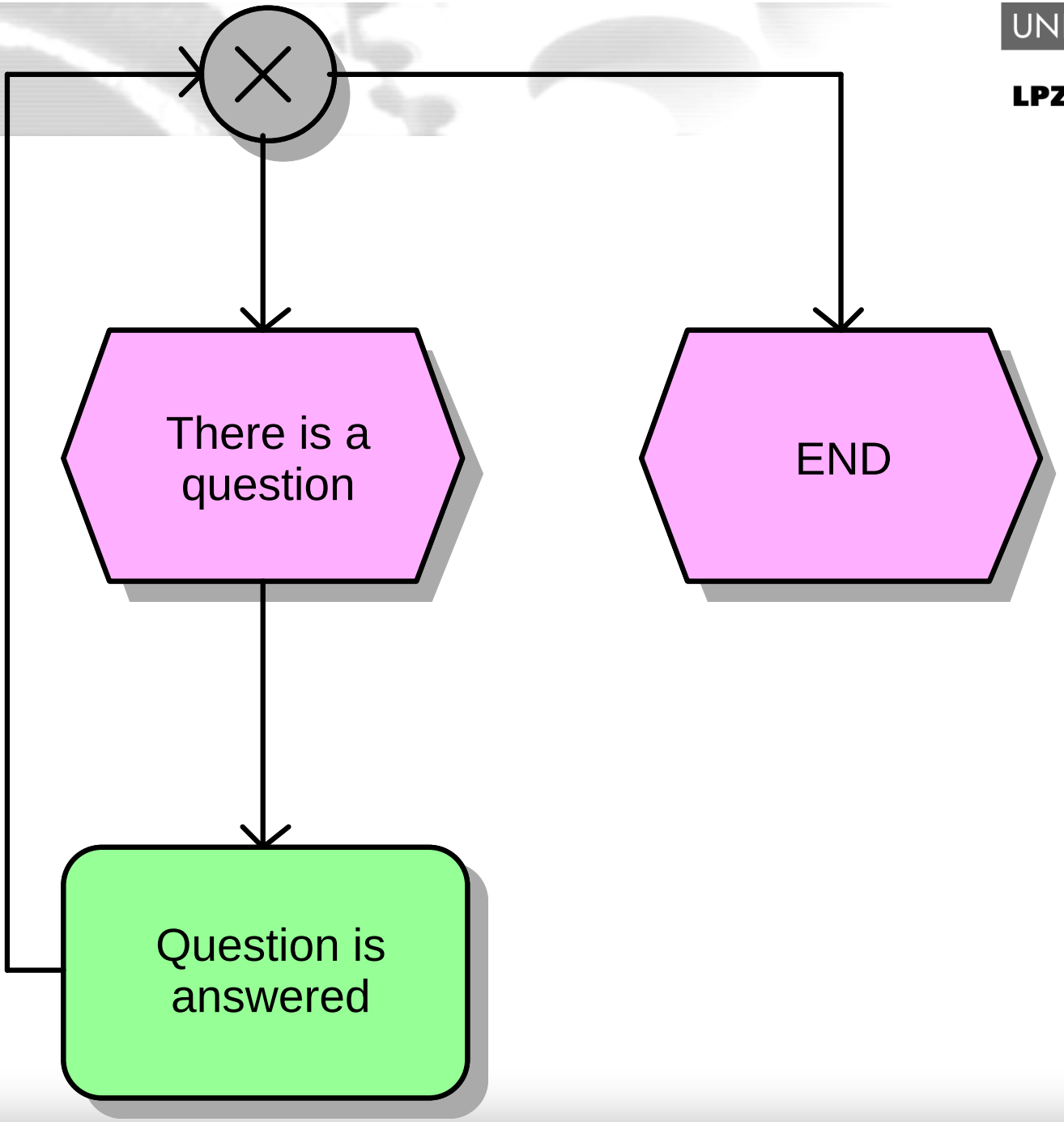
LPZ E-BUSINESS applied telematics

- Reduction rules found 178 error patterns in 90 models

  Our approach found 176 of them.

  The remaining two models are sound when another formal semantics is assumed.

- 57 models could not be reduced completely
  Petri-net analyzer:
  36 unsound
  We found at least one error in all these models
  21 sound
  Our tool produced one false alert.


There is a question
END
Question is answered

Business Process Modelling with Continuous Validation

Stefan Kühne[1] Heiko Kern[1] Volker Gruhn[2] Ralf Laue[2]

[1]University of Leipzig, Department of Computer Science, Business Information Systems

[2]University of Leipzig, Department of Computer Science, Applied Telematics/e-Business

1st International Workshop on Model-Driven Engineering for Business Process Management (MDE4BPM 2008)

Outline

Overview

bflow* toolbox

- open source project (`http://www.bflow.org/`)
- initiated by Frank Rump and Markus Nüttgens
- EMF-, GMF-based implementation

Overview

bflow* toolbox

- open source project (`http://www.bflow.org/`)
- initiated by Frank Rump and Markus Nüttgens
- EMF-, GMF-based implementation

MDD tools

- model validation: OCL, Check, EVL
- model-to-model transformation: QVT, ATL, XTend, Viatra2
- model-to-text transformation: MOFScript, XPand, JET
- frameworks: oAW, Epsilon

Overview

bflow* toolbox

- open source project (`http://www.bflow.org/`)
- initiated by Frank Rump and Markus Nüttgens
- EMF-, GMF-based implementation

MDD tools

- model validation: OCL, Check, EVL
- model-to-model transformation: QVT, ATL, XTend, Viatra2
- model-to-text transformation: MOFScript, XPand, JET
- frameworks: oAW, Epsilon

Overview

bflow* toolbox

- open source project (`http://www.bflow.org/`)
- initiated by Frank Rump and Markus Nüttgens
- EMF-, GMF-based implementation

MDD tools

- model validation: OCL, Check, EVL
- model-to-model transformation: QVT, ATL, XTend, Viatra2
- model-to-text transformation: MOFScript, XPand, JET
- frameworks: oAW, Epsilon

oAW Check (example)

Check

```
context epc::Element
    if (this.isConnector())
    WARNING 'RS006: Connectors should have either multiple incoming or
        multiple outgoing arcs' :
    !(this.isJoin() && this.isSplit());
```

oAW Check (example)

Check

```
context epc::Element
    if (this.isConnector())
    WARNING 'RS006: Connectors should have either multiple incoming or
        multiple outgoing arcs' :
    !(this.isJoin() && this.isSplit());
```

XTend

```
Boolean isJoin(epc::Element element) :
    element.isConnector() && element.incomingControlFlows().size > 1;
```

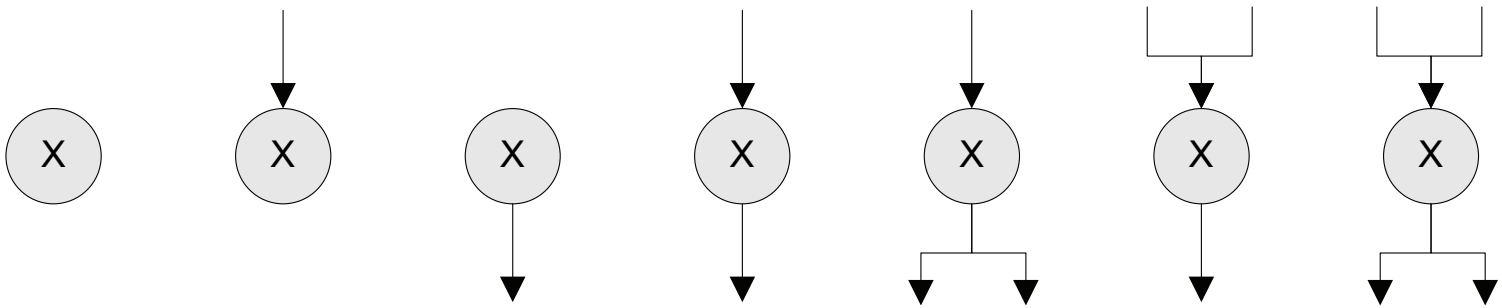
oAW Check (example)

Check

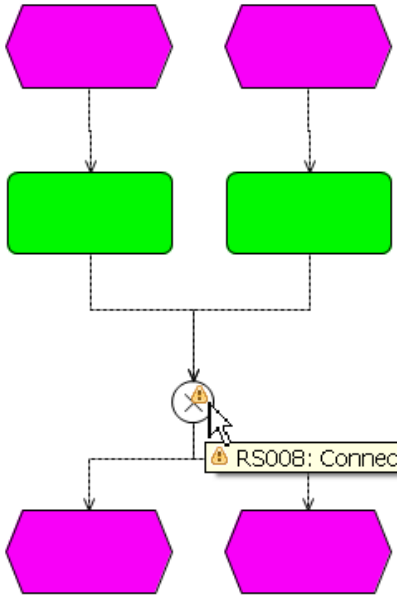
```
context epc::Element
    if (this.isConnector())
    WARNING 'RS006: Connectors should have either multiple incoming or
        multiple outgoing arcs' :
    !(this.isJoin() && this.isSplit());
```

XTend

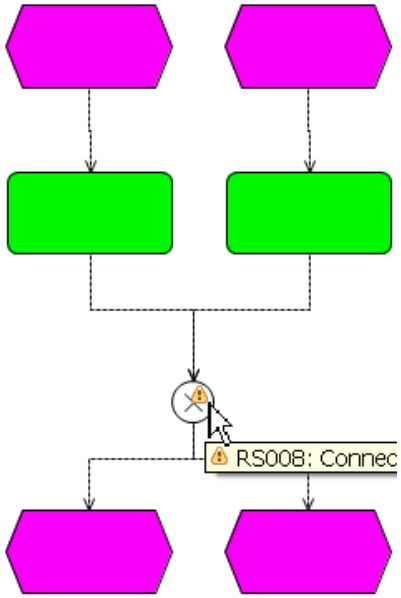
```
Boolean isJoin(epc::Element element) :
    element.isConnector() && element.incomingControlFlows().size > 1;
```

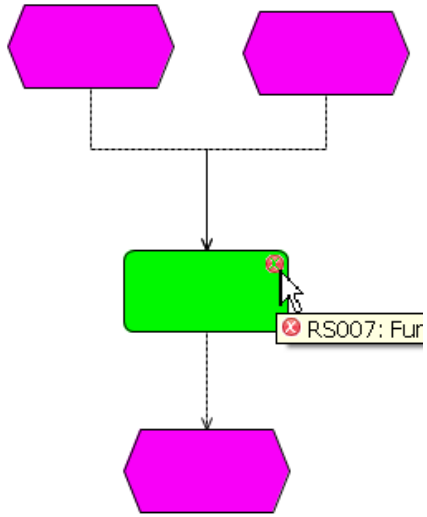


Syntax errors

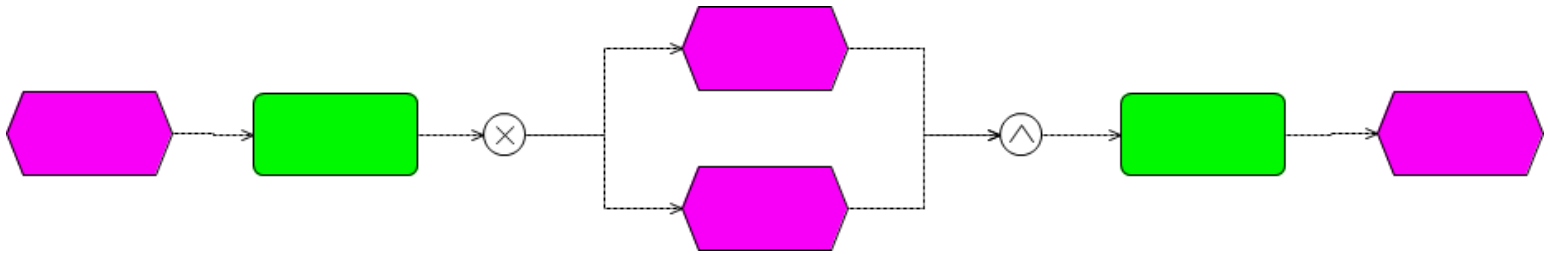


Syntax errors

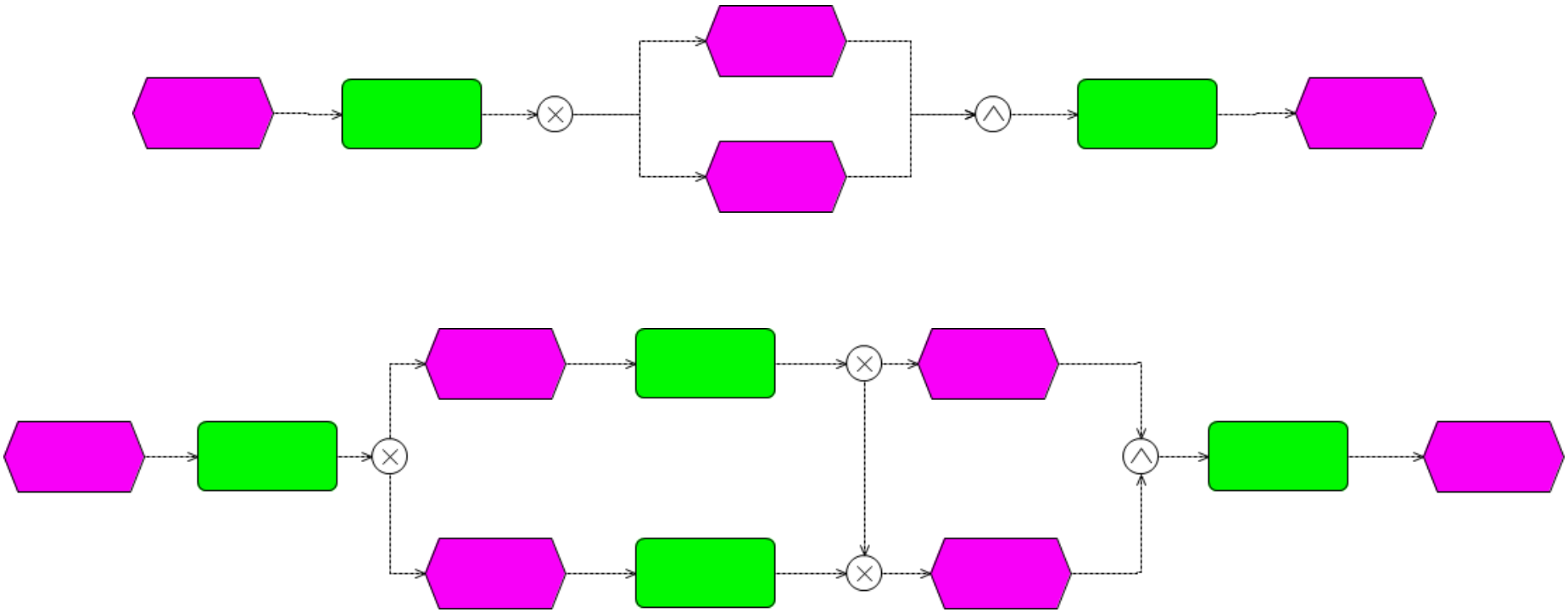




Mismatched connectors



Mismatched connectors



Match and seseMatch

Definition (match(s,j))

A split s is matched by a join j (symbol: $match(s, j)$) iff there exist two directed paths from s to j which only common elements are s and j.

Match and seseMatch

Definition (match(s,j))

A split s is matched by a join j (symbol: $match(s, j)$) iff there exist two directed paths from s to j which only common elements are s and j.

Definition (seseMatch(s,j))

Let s be a split and j be a join with $match(s, j)$. We say that there are no entries and no exits between s and j, $seseMatch(s, j)$, if the following conditions hold:

1. Every path from s to an end event must contain j.
2. Every path from a start event to j must contain s.
3. Every path from s to s must contain j.
4. Every path from j to j must contain s.

Match computation with XTend

XTend: match

```
Boolean match(epc::Element split, epc::Element join) :
    split.children().select(c|c == join || c.isConnectedTo(join)).size > 1
```

Match computation with XTend

XTend: match

```
Boolean match(epc::Element split, epc::Element join) :
    split.children().select(c|c == join || c.isConnectedTo(join)).size > 1
    && split.joinOutsBetween(join).notExists(a| a.separates(split, join));
```

Match computation with XTend

XTend: match

```
Boolean match(epc::Element split, epc::Element join) :
    split.children().select(c|c == join || c.isConnectedTo(join)).size > 1
    && split.joinOutsBetween(join).notExists(a| a.separates(split, join));
```

XTend: separates

```
Boolean separates(epc::Arc arc, epc::Element source, epc::Element target)
    :
    source.arcsBetween(target)
        .remove(arc)
        .without(arc.from.preceedingArcs(source))
        .without(arc.to.succeedingArcs(target))
        .isEmpty;
```

Sese computation with XTend

XTend: succeedingArcs

```
Set[epc::Arc] succeedingArcs(epc::Element element, Set[epc::Element]
      processed) :
     element.outArcs().union(
          element.children().without(processed)
               .succeedingArcs(processed.union(element.children())));
```

Sese computation with XTend

XTend: succeedingArcs

```
Set[epc::Arc] succeedingArcs(epc::Element element, Set[epc::Element]
      processed) :
     element.outArcs().union(
          element.children().without(processed)
               .succeedingArcs(processed.union(element.children())));
```

XTend: sese

```
Boolean sese(epc::Element s, epc::Element t) :
    s.succeedingArcs({t}.toSet()) == t.preceedingArcs({s}.toSet());
```

Sese computation with XTend

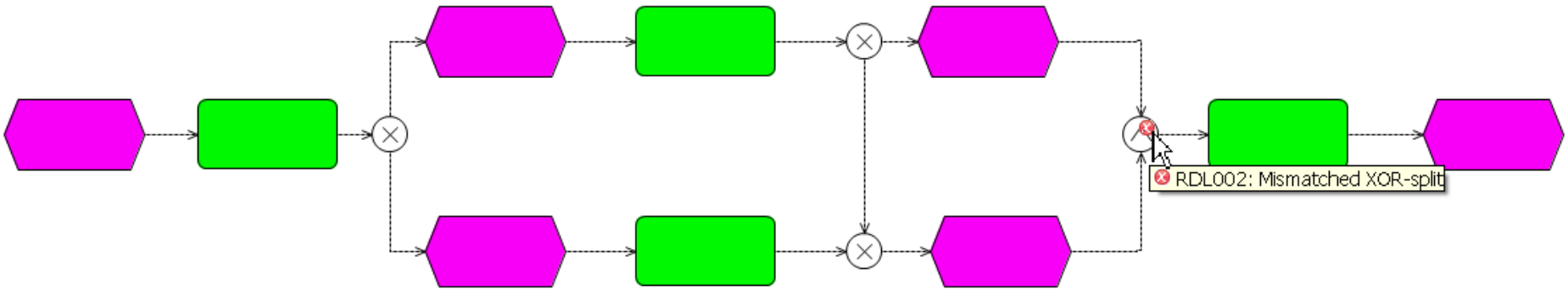
Check

```
context epc::Element if this.isAndJoin()
    ERROR "Mismatched XOR-split" :
    this.precessors().notExists(s| s.isXorSplit()
        && s.seseMatches(this));
```

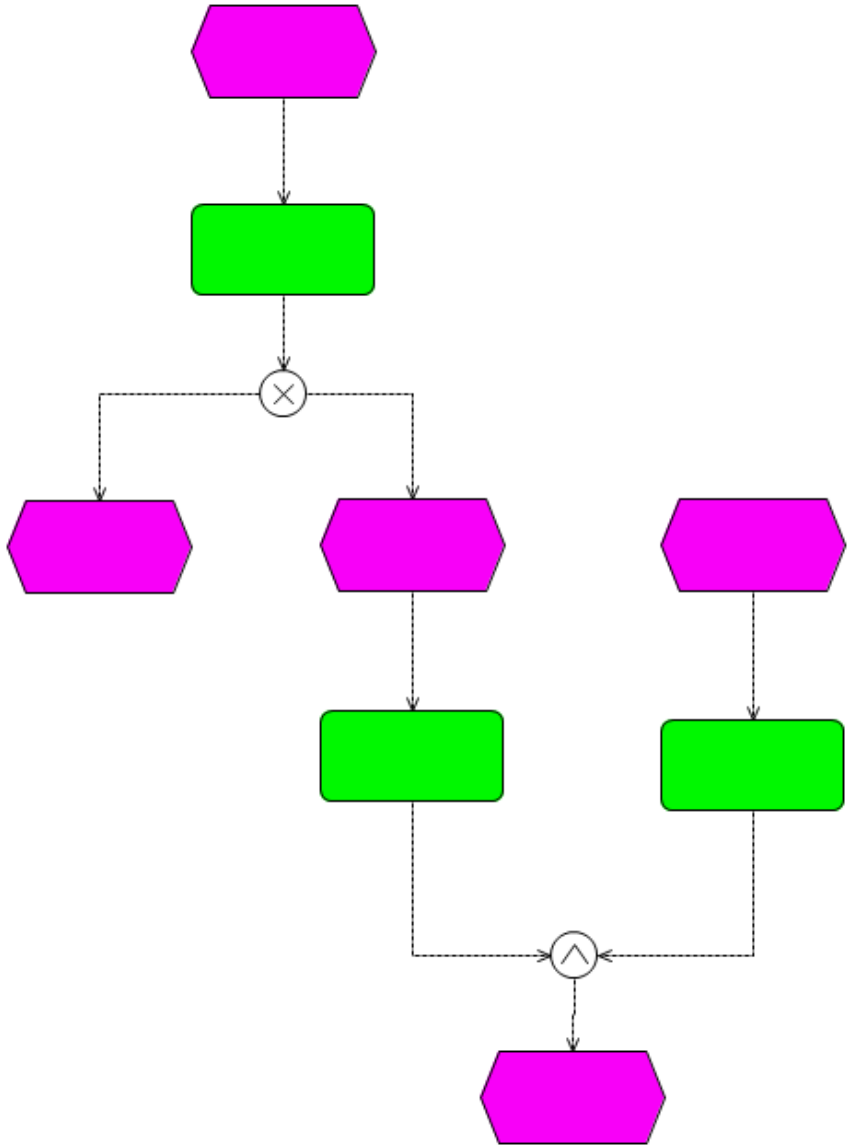
Sese computation with XTend

Check

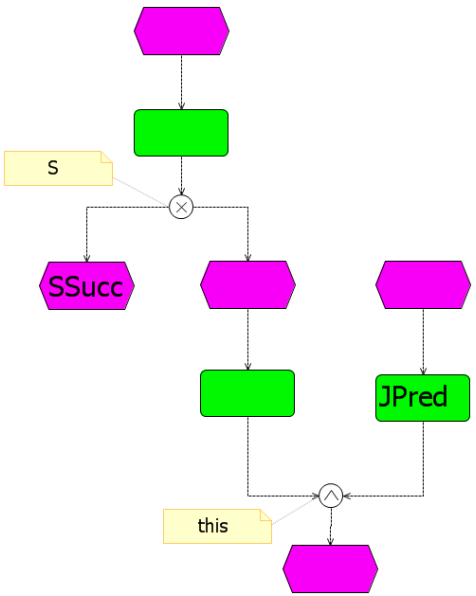
```
context epc::Element if this.isAndJoin()
    ERROR "Mismatched XOR-split" :
    this.precessors().notExists(s| s.isXorSplit()
        && s.seseMatches(this));
```



Synchronisation Problem in AND-join



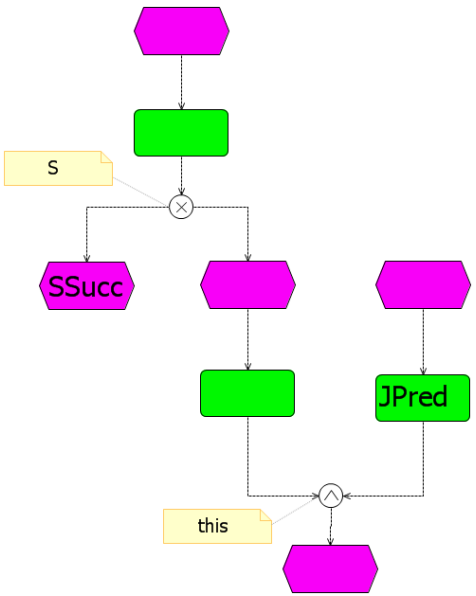
Synchronisation Problem in AND-join



Check

```
context epc::Element
```

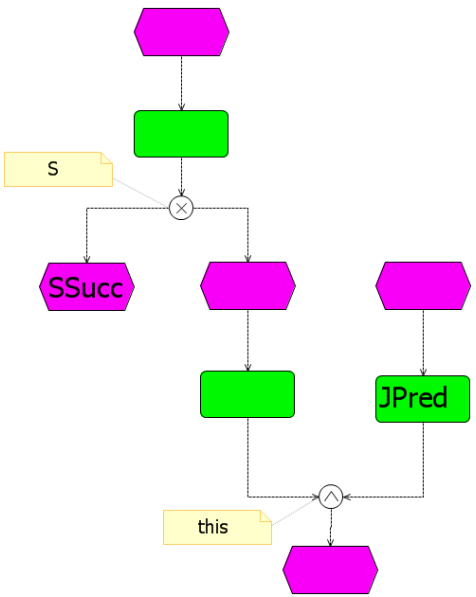
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
```

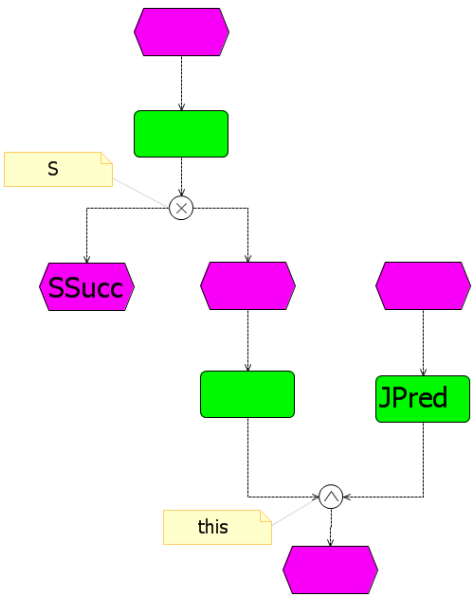
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
    ERROR "AND−Join might not get control" :
```

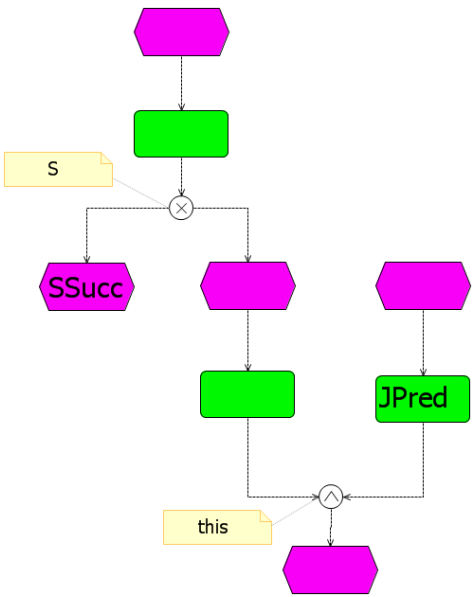
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
    ERROR "AND−Join might not get control" :
    this.precessors().notExists(S|
```

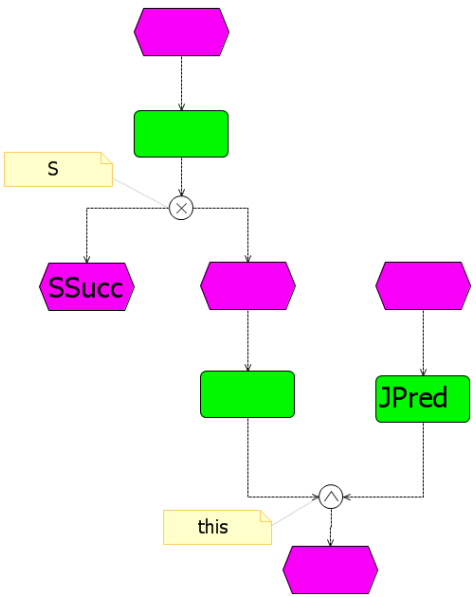
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
    ERROR "AND−Join might not get control" :
    this.precessors().notExists(S|
        (S.isXorSplit() || S.isOrSplit())
```

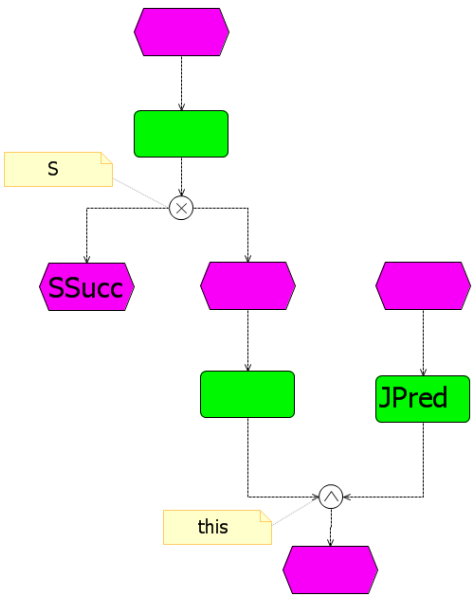
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
    ERROR "AND−Join might not get control" :
    this.precessors().notExists(S|
        (S.isXorSplit() || S.isOrSplit())
        && S.children().exists(SSucc|
```

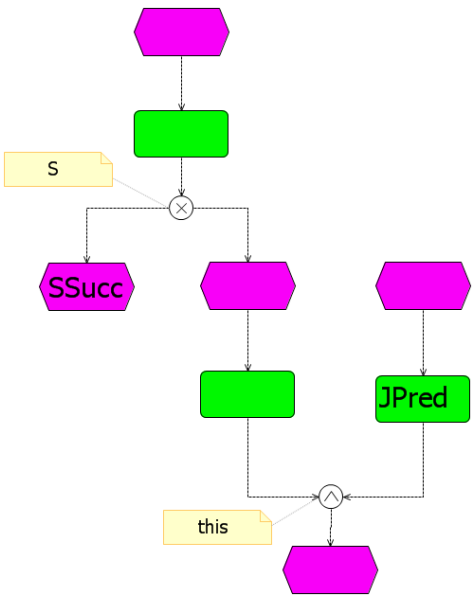
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
    ERROR "AND−Join might not get control" :
    this.precessors().notExists(S|
        (S.isXorSplit() || S.isOrSplit())
        && S.children().exists(SSucc|
            SSucc != this && SSucc.
                isNotConnectedTo(this))
```

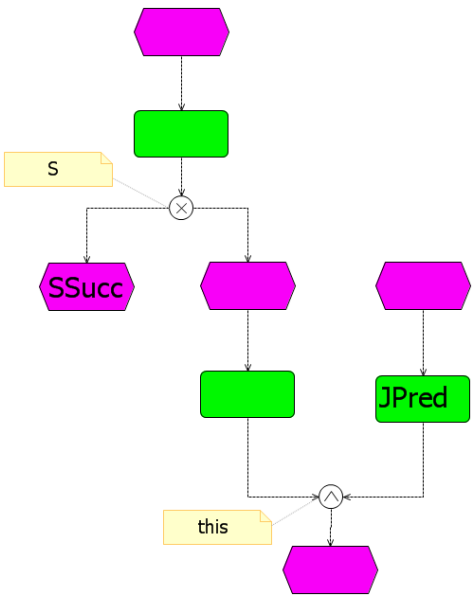
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
    ERROR "AND−Join might not get control" :
    this.precessors().notExists(S|
        (S.isXorSplit() || S.isOrSplit())
        && S.children().exists(SSucc|
            SSucc != this && SSucc.
                isNotConnectedTo(this))
        && this.precessors().exists(JPred|
```

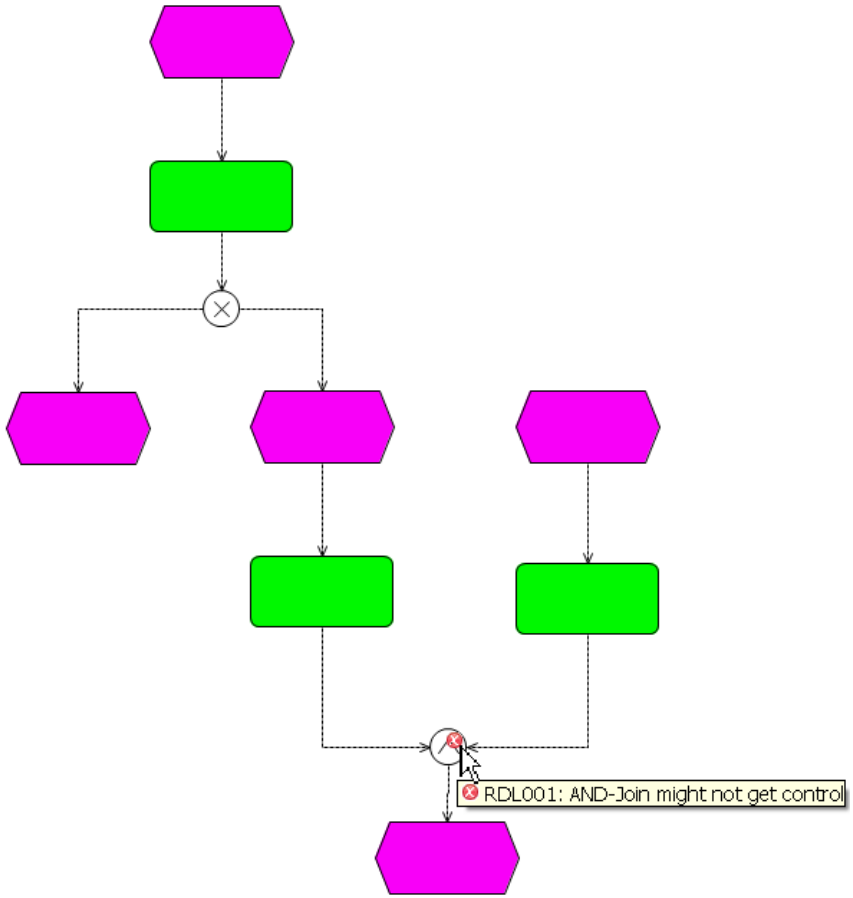
Synchronisation Problem in AND-join



Check

```
context epc::Element if this.isAndJoin()
    ERROR "AND-Join might not get control" :
    this.precessors().notExists(S|
        (S.isXorSplit() || S.isOrSplit())
        && S.children().exists(SSucc|
            SSucc != this && SSucc.
                isNotConnectedTo(this))
        && this.precessors().exists(JPred|
            JPred != S && S.isNotConnectedTo(
                JPred))
    );
```

Synchronisation Problem in AND-join



Batch processing

Modifications

- Iteration over all models
- Cached XTend functions
- Relation of error messages to EPC context
- oAW Workflow

Batch processing

Modifications

- Iteration over all models
- Cached XTend functions
- Relation of error messages to EPC context
- oAW Workflow

Syntax checks in SAP reference model

```
...
14422 ERROR WorkflowRunner – RS009: /Main group/SAP/Production/Discrete Production/Discrete Production: Connector
      has no incoming arcs. [org.eclipse.emf.ecore.impl.DynamicEObjectImpl@a76a1f (eClass: org.eclipse.emf.ecore.impl.
      EClassImpl@4e79f1 (name: OR) (instanceClassName: null) (abstract: false, interface: false))]
14422 ERROR WorkflowRunner – RS009: /Main group/SAP/Quality Management/Test Equipment Management/Maintenance
      Order/Maintenance Order: Connector has no incoming arcs. [org.eclipse.emf.ecore.impl.DynamicEObjectImpl@f0a6e8 (
      eClass: org.eclipse.emf.ecore.impl.EClassImpl@11a64ed (name: AND) (instanceClassName: null) (abstract: false,
      interface: false))]
14422 ERROR WorkflowRunner – RS010: /Main group/SAP/Production/Discrete Production/Discrete Production: Connector
      has no outgoing arcs. [org.eclipse.emf.ecore.impl.DynamicEObjectImpl@a76a1f (eClass: org.eclipse.emf.ecore.impl.
      EClassImpl@4e79f1 (name: OR) (instanceClassName: null) (abstract: false, interface: false))]
14422 ERROR WorkflowRunner – RS010: /Main group/SAP/Quality Management/Test Equipment Management/Maintenance
      Order/Maintenance Order: Connector has no outgoing arcs. [org.eclipse.emf.ecore.impl.DynamicEObjectImpl@f0a6e8 (
      eClass: org.eclipse.emf.ecore.impl.EClassImpl@11a64ed (name: AND) (instanceClassName: null) (abstract: false,
      interface: false))]
```