Tutorial on Business Process Management

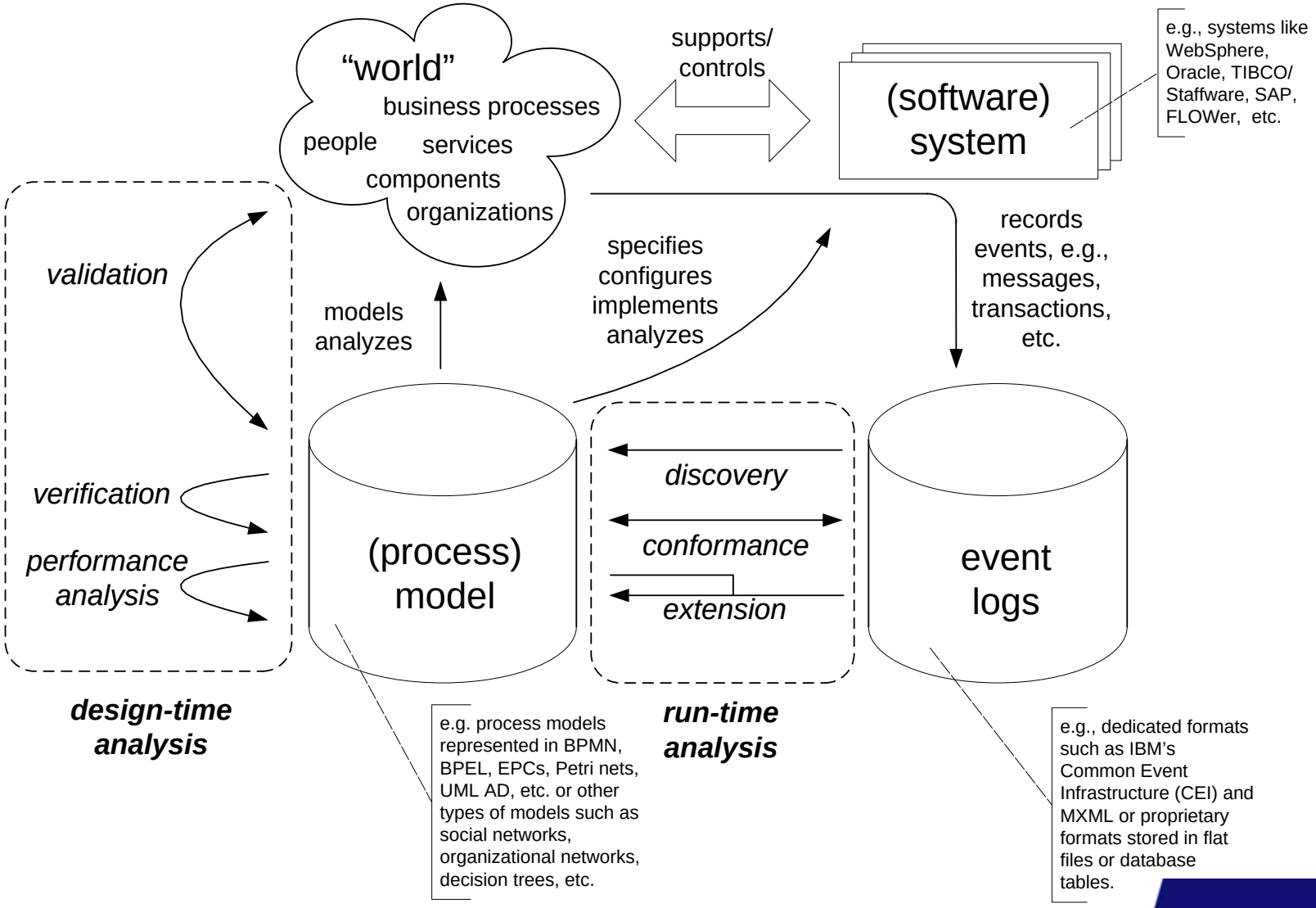
prof.dr.ir. Wil van der Aalst

vdaalst.com

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Overview

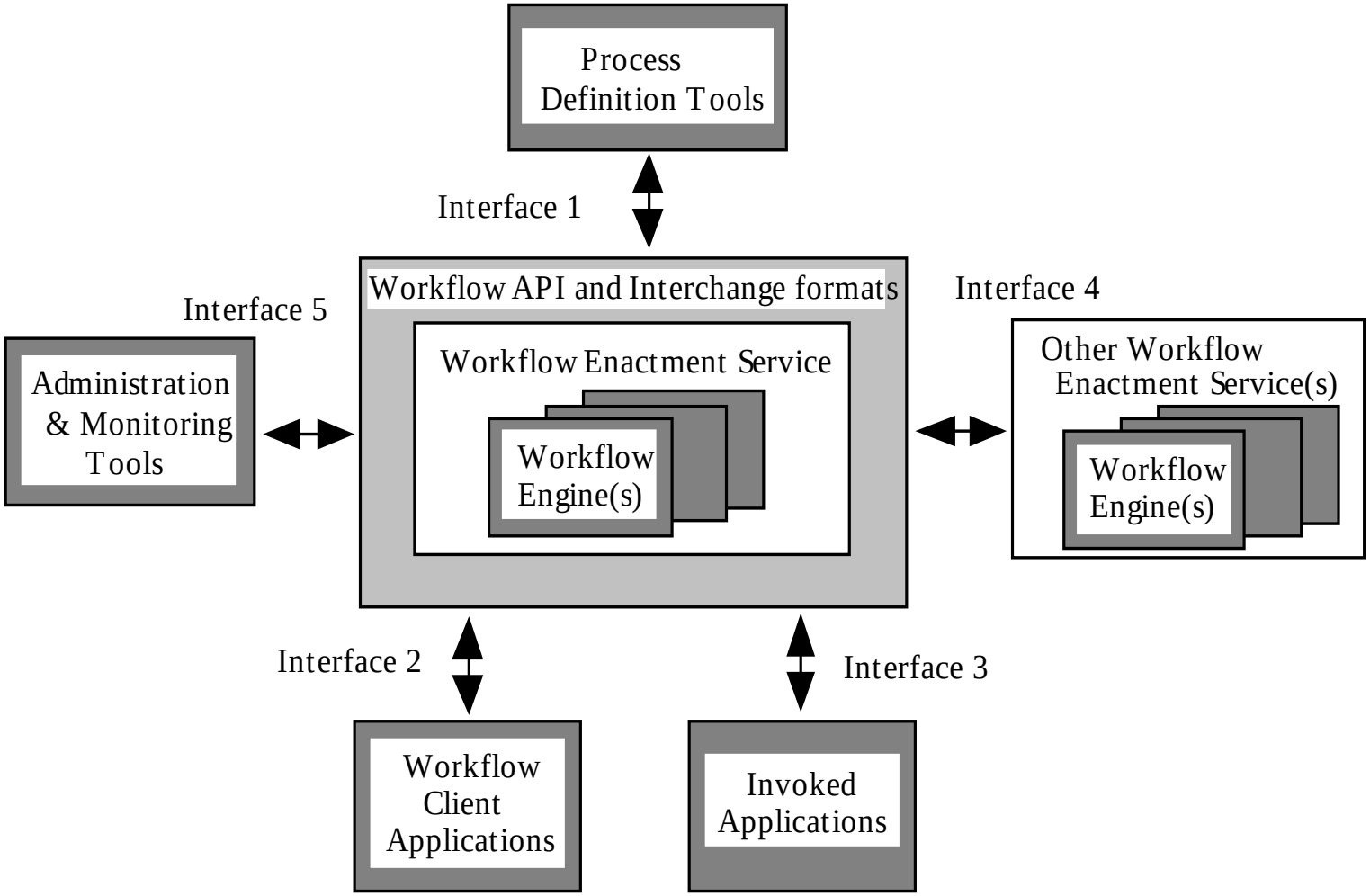

"world"
business processes
people
services
components
organizations
supports/
controls
(software)
system
e.g., systems like WebSphere, Oracle, TIBCO/ Staffware, SAP, FLOWer, etc.
validation
verification
performance
analysis
models
analyzes
specifies
configures
implements
analyzes
records
events, e.g.,
messages,
transactions,
etc.
(process)
model
discovery
conformance
extension
event
logs
design-time
analysis
run-time
analysis
e.g. process models represented in BPMN, BPEL, EPCs, Petri nets, UML AD, etc. or other types of models such as social networks, organizational networks, decision trees, etc.
e.g., dedicated formats such as IBM's Common Event Infrastructure (CEI) and MXML or proprietary formats stored in flat files or database tables.

BPM Systems

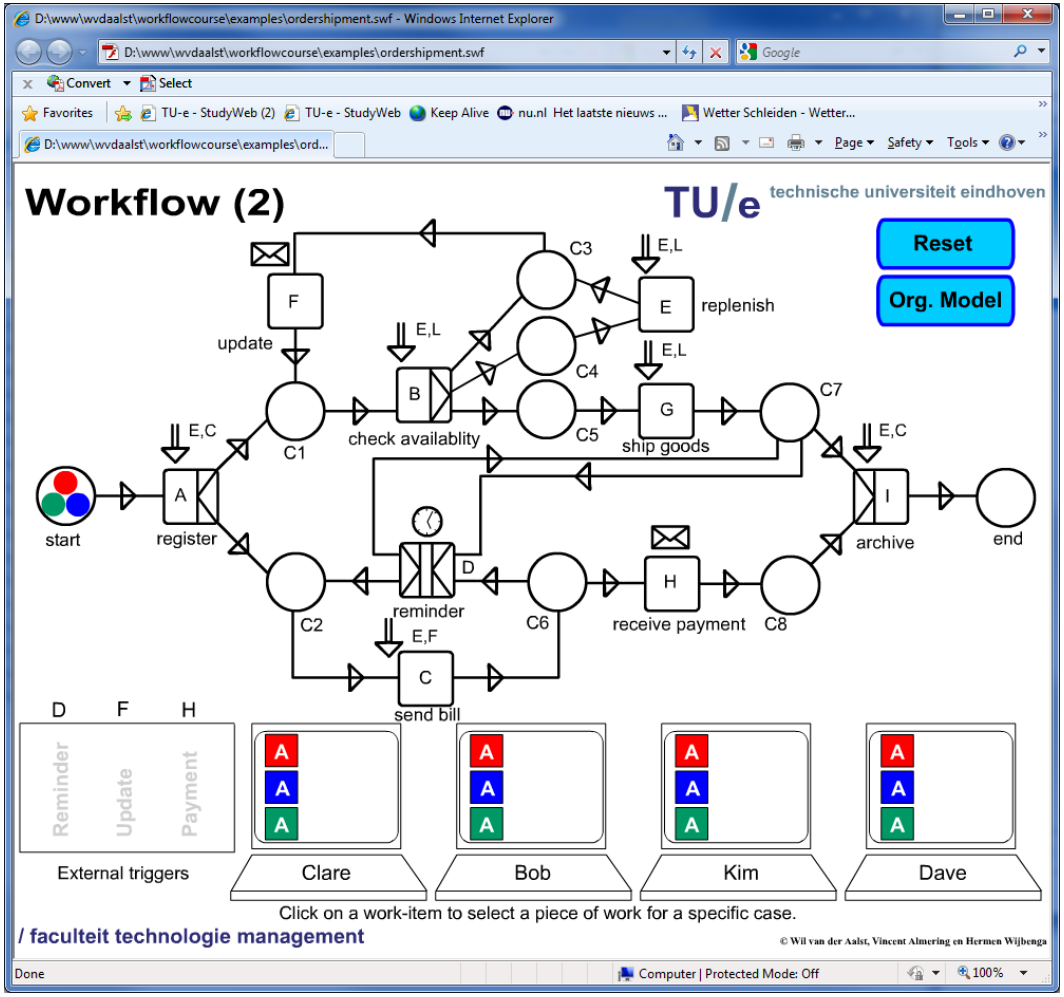
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

WfMC Reference model (early nineties)



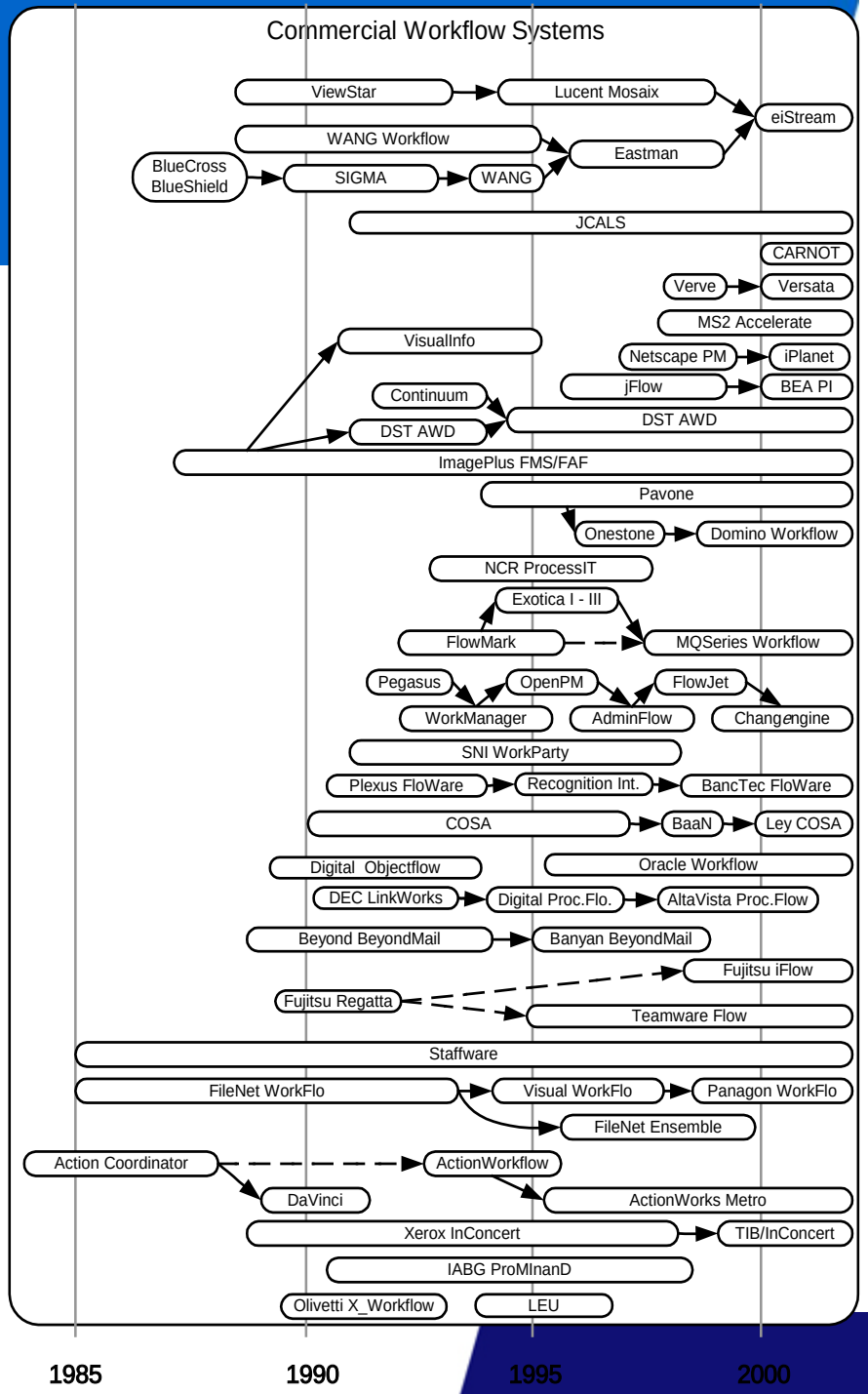
Basic Idea



D:\www\wvdaalst\workflowcourse\examples.htm
www.workflowcourse.com

Examples of workflow/BPM systems

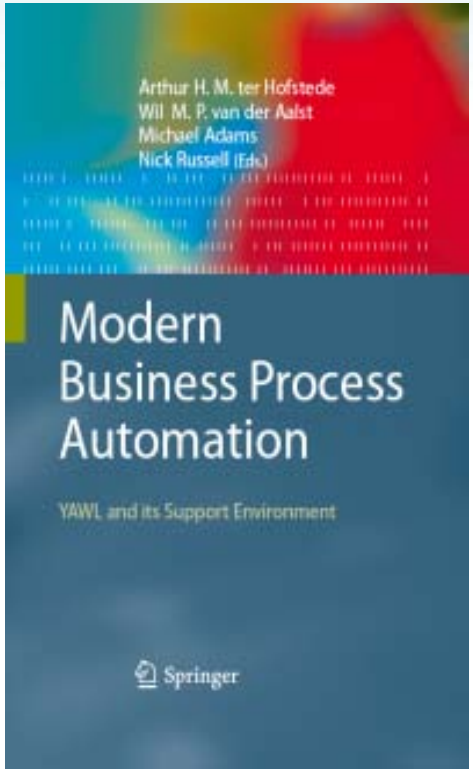
- BPM|one
- YAWL
- COSA
- FileNet
- Staffware
- Oracle BPEL
- SAP Workflow
- …



Focus on "classical" workflow management systems, but ...

1. Information systems with hard-coded workflows (process& organization specific).
2. Custom-made information systems with generic workflow support (organization specific).
3. Generic software with embedded workflow functionality (e.g., the workflow components of ERP, CRM, PDM, etc. systems).
4. Generic software focusing on workflow functionality (e.g., Staffware, MQSeries Workflow, FLOWer, COSA, Oracle BPEL, Filenet, etc.).

The power of expressiveness



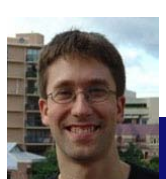




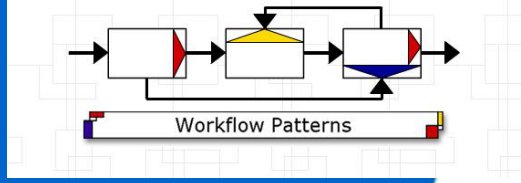








Workflow Patterns Initiative

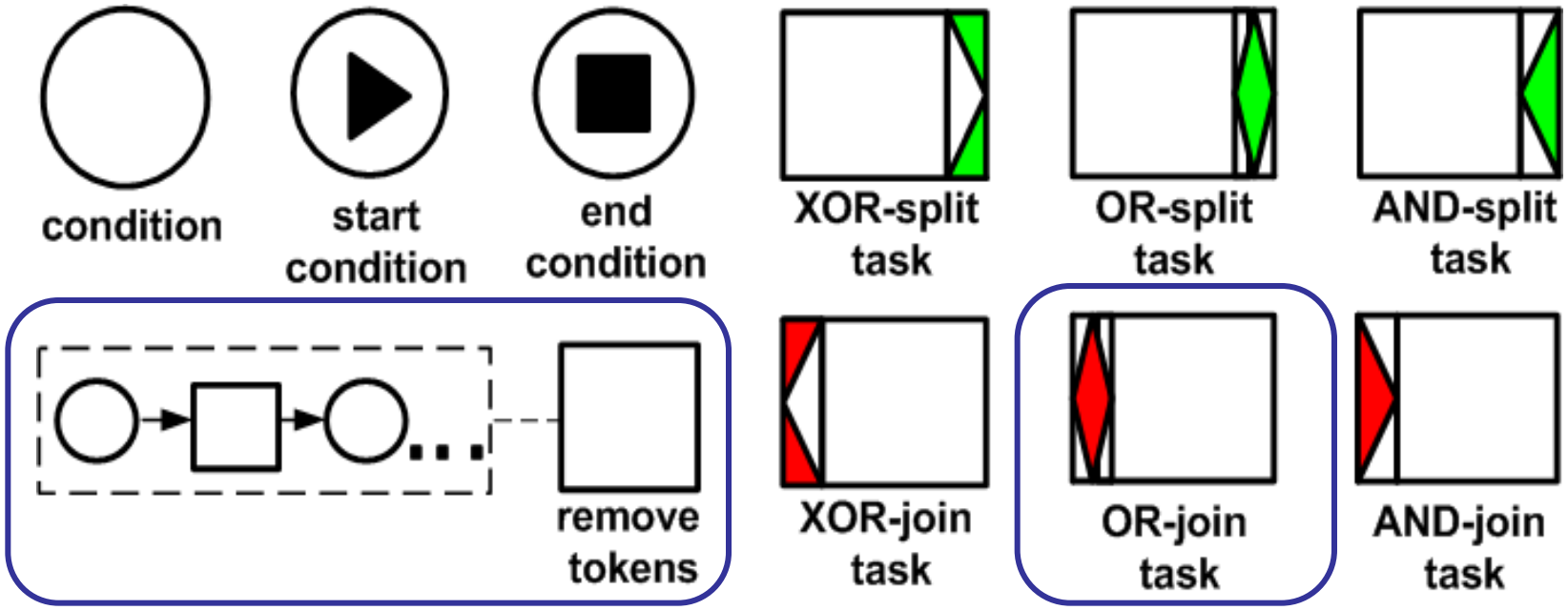


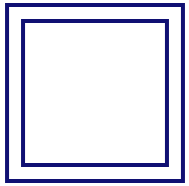
- **Started in 1999, joint work TU/e and QUT**
- **Objectives:**
  - **Identification of workflow modeling scenarios and solutions**
  - **Benchmarking**
    - **Workflow products (MQ/Series Workflow, Staffware, etc)**
    - **Proposed standards for web service composition (BPML, BPEL)**
    - **Process modelling languages (UML, BPMN)**
  - **Foundation for selecting workflow solutions**
- **Home Page: www.workflowpatterns.com**
- **Primary publication (most cited workflow paper ever):**
  - **W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, A.P. Barros, "Workflow Patterns", *Distributed and Parallel Databases* 14(3):5-51, 2003.**
- **Evaluations of commercial offerings, research prototypes, proposed standards for web service composition, etc**

YAWL vs Petri nets/simple notations

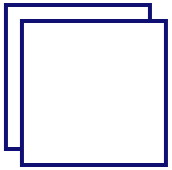
- **Petri nets have difficulties capturing:**
  - **The General Synchronizing Merge (OR-join)**
  - **Patterns involving Multiple Instances (order containing order lines)**
  - **Cancellation of a certain part of a process (reset arcs in PN terms)**

- **For the Control Flow Perspective, YAWL takes some concepts from Petri nets and adds constructs for:**
  - **OR-join to deal with General Synchronizing Merge**
  - **Multiple Instance (MI) tasks**
  - **Cancellation regions**
  - **“Syntactic Sugar” (various splits/joins, direct connections between tasks)**
  - **Some of this Syntactic Sugar was already introduced.**



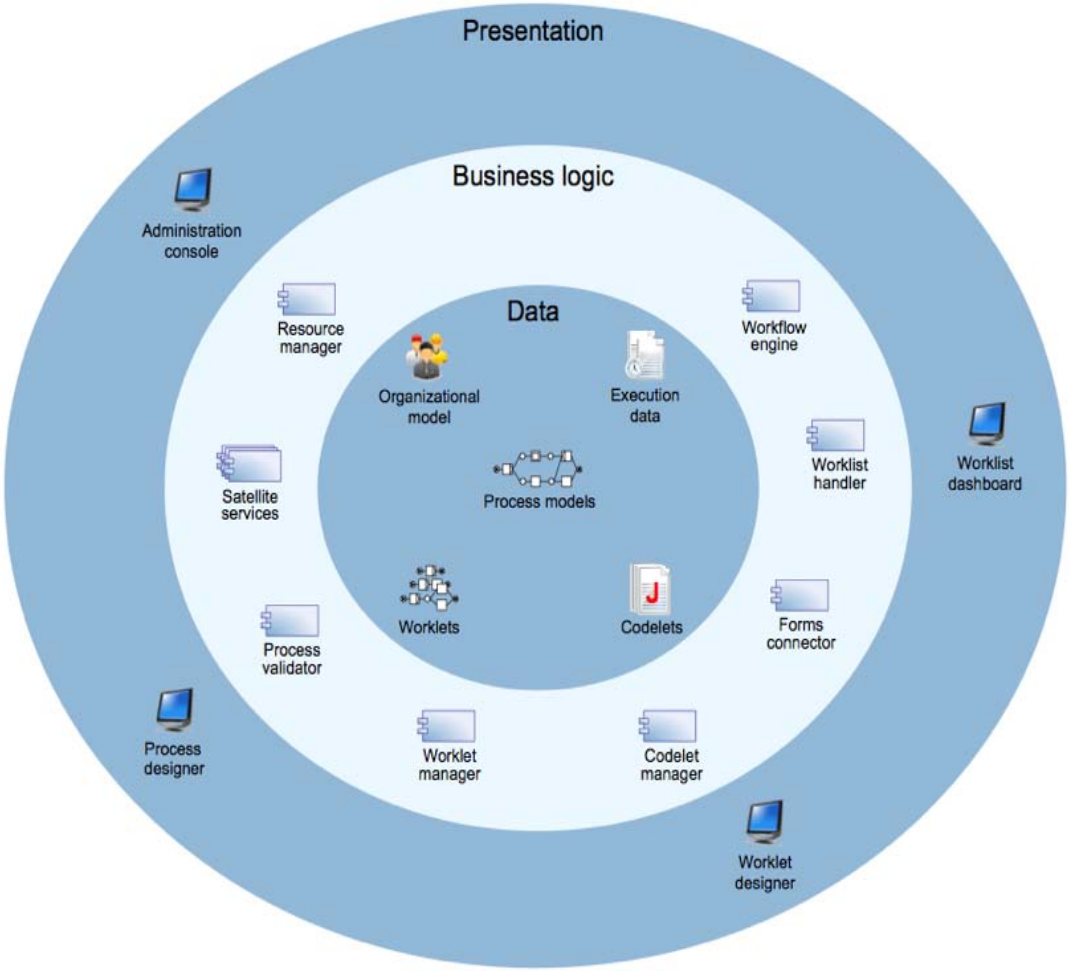


Composite task

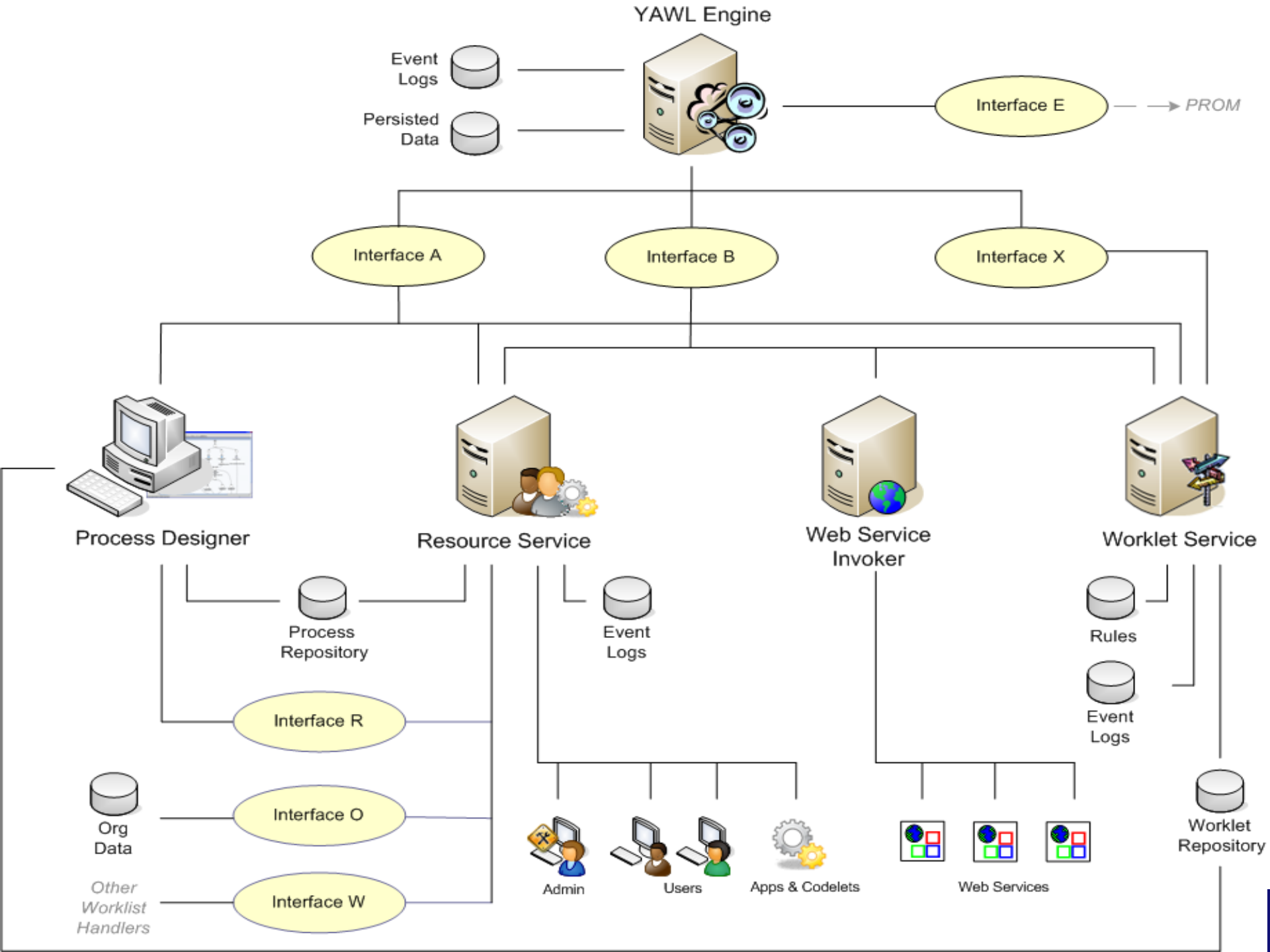


Multiple Instance task

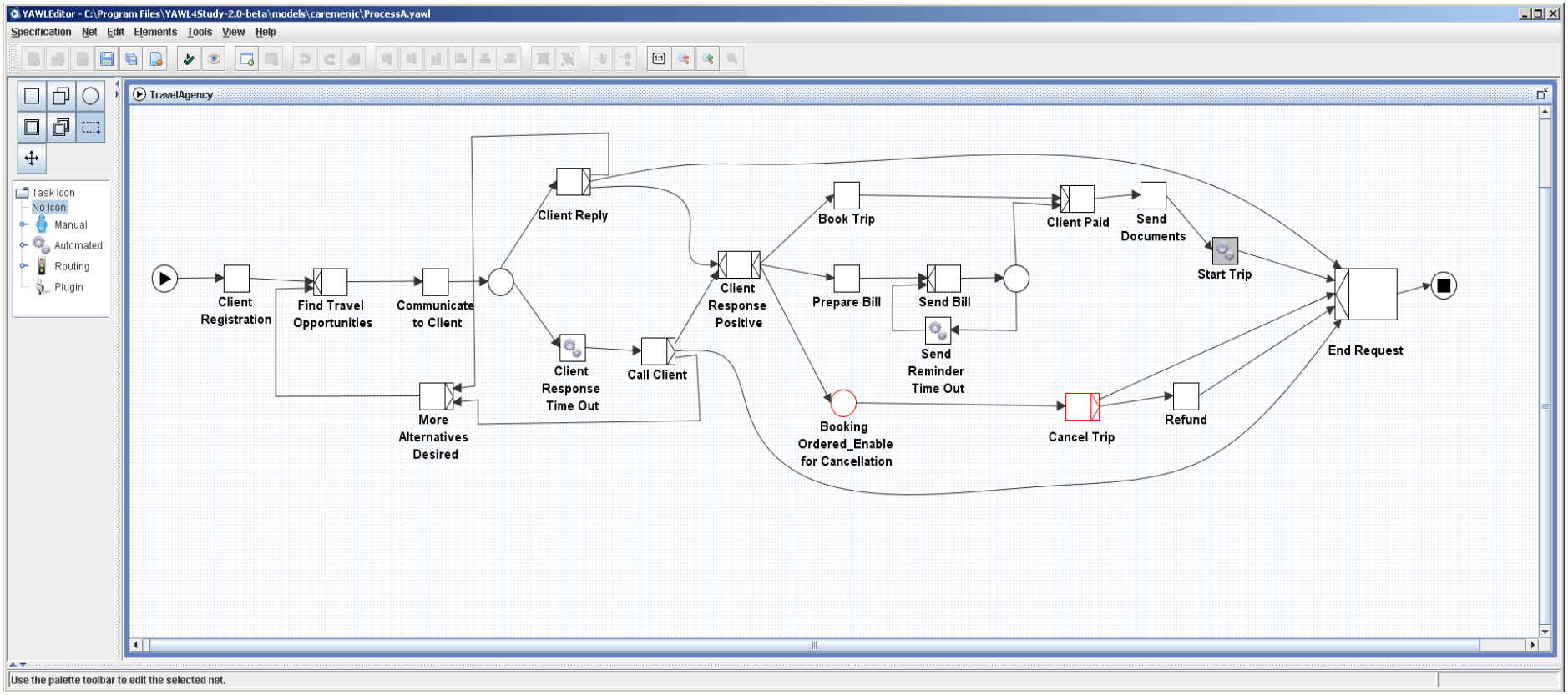
Architecture of YAWL



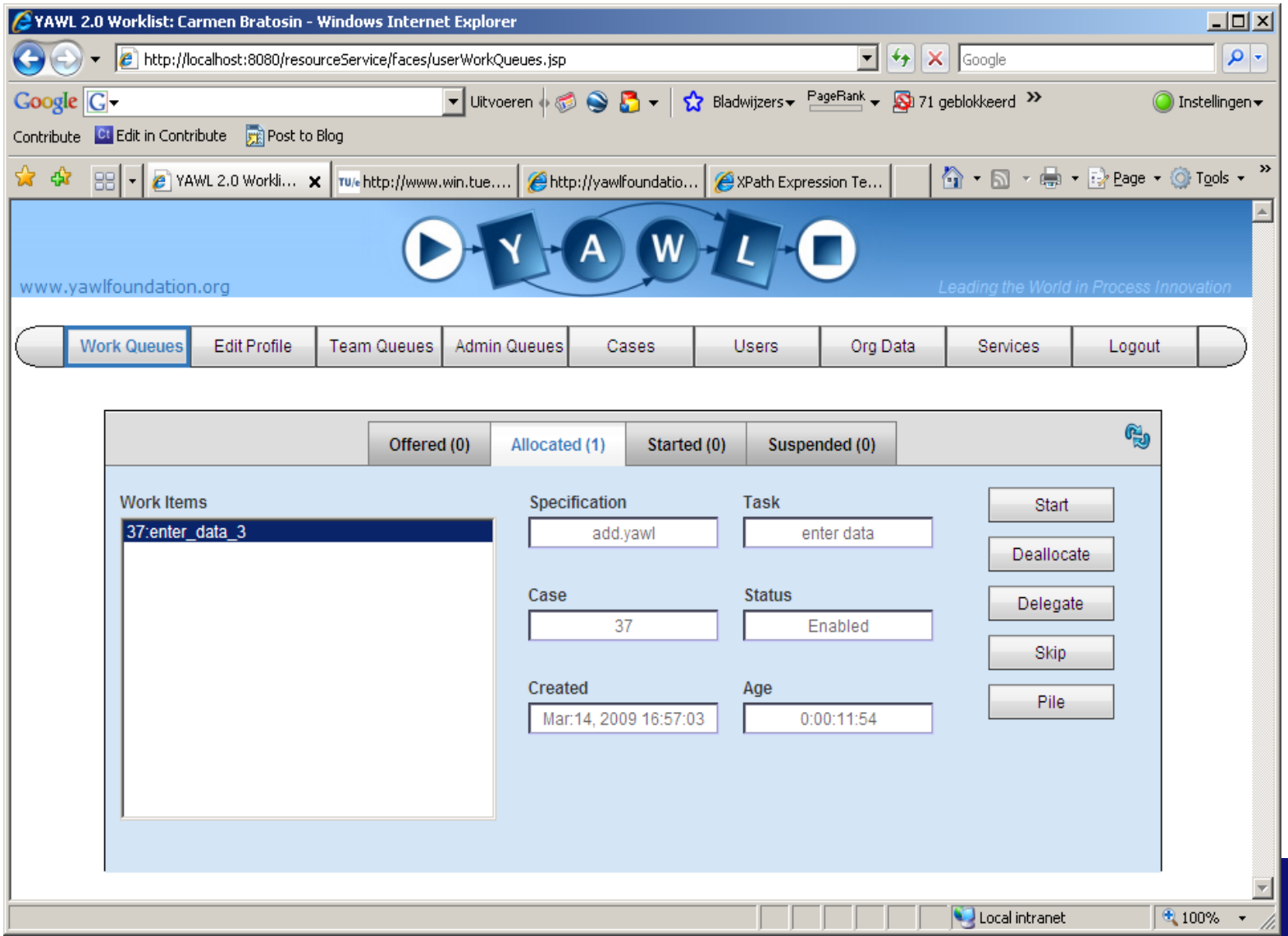
Taken from the Architecture Chapter of YAWL book by Michael Adams, Ma


YAWL Engine
Event Logs
Persisted Data
Interface E
PROM
Interface A
Interface B
Interface X
Process Designer
Resource Service
Web Service Invoker
Worklet Service
Process Repository
Event Logs
Rules
Event Logs
Interface R
Org Data
Interface O
Other Worklist Handlers
Interface W
Admin
Users
Apps & Codelets
Web Services
Worklet Repository

YAWL Editor

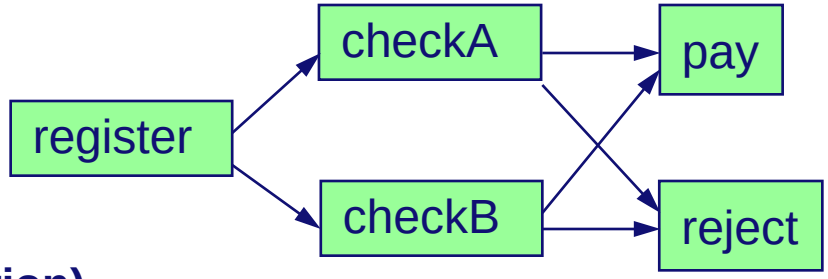

YAWLEditor - C:\Program Files\YAWL4Study-2.0-beta\models\caremenjc\ProcessA.yawl
Specification Net Edit Elements Tools View Help
TravelAgency
Task Icon
No Icon
Manual
Automated
Routing
Plugin
Client Registration
Find Travel Opportunities
Communicate to Client
Client Reply
Client Response Time Out
Call Client
More Alternatives Desired
Client Response Positive
Book Trip
Prepare Bill
Send Bill
Send Reminder Time Out
Booking Ordered_Enable for Cancellation
Client Paid
Send Documents
Start Trip
Cancel Trip
Refund
End Request
Use the palette toolbar to edit the selected net.

YAWL Control Centre



Example step-by-step

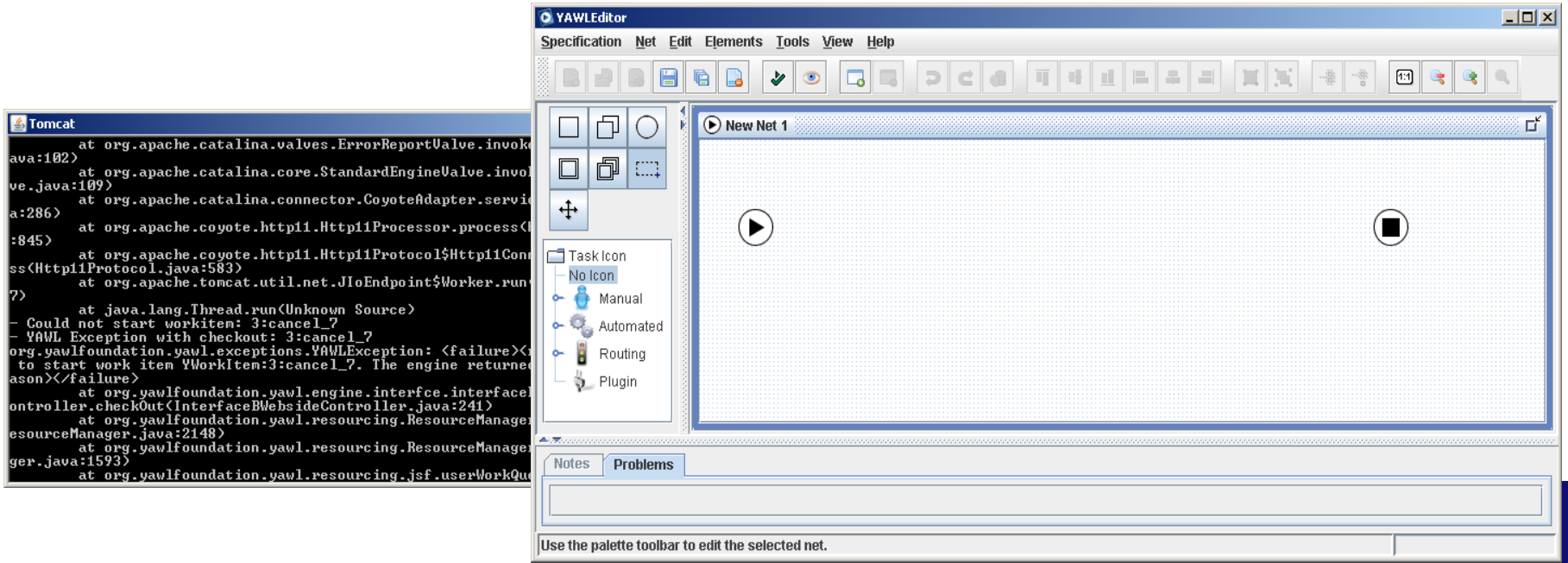
- Processing of insurance claims involving registration, two checks, and a payment of rejection
- Five tasks:
  - register (register insurance claim)
  - checkA (check insurance policy)
  - checkB (check damage reported)
  - pay (pay for the damage)
  - reject (inform customer about rejection)



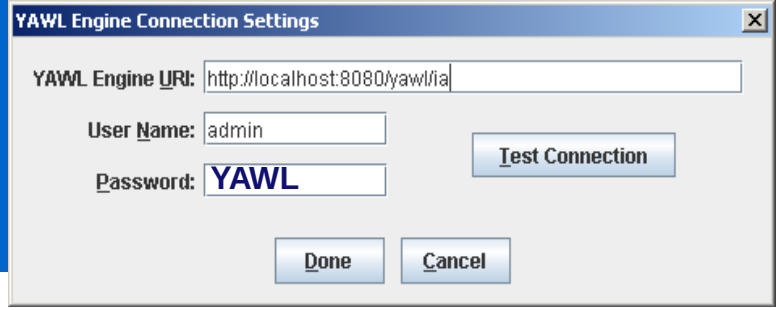
- Registration is followed by two checks which can be handled in parallel.
- Each of the checks results in “OK” or “not OK”.
- If both are OK, pay otherwise reject.
- No waste of work, i.e., stop after first "not OK".
- Three roles: register (for task register), checks (for both checks), and pay/reject (for final tasks).

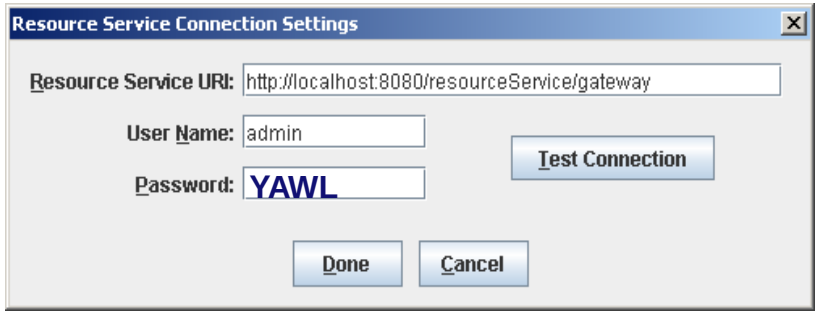
Starting YAWL

- Start Engine (Select "Start Engine" from start menu).
- Start Editor (Select "YAWL-Editor" from start menu).
- Create new specification (Select Create Specification in Editor's menu)

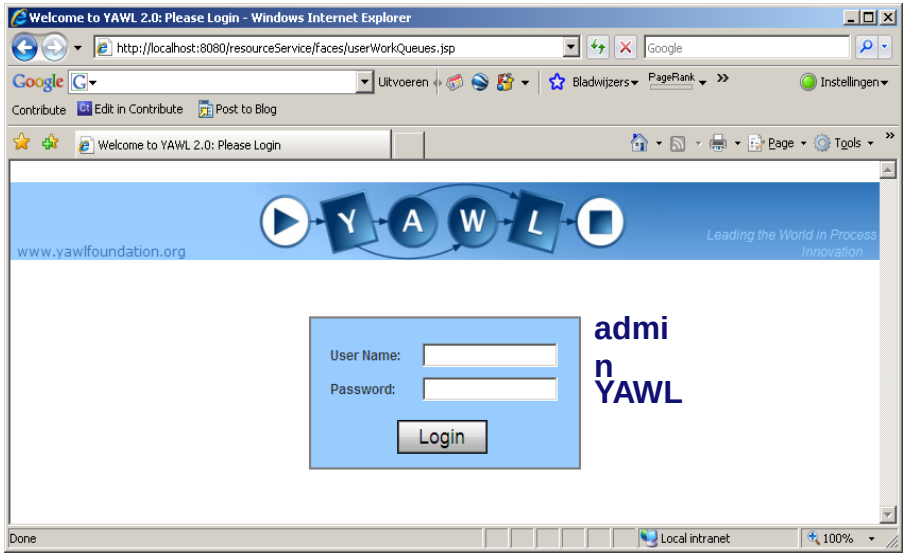


Connecting and logging in

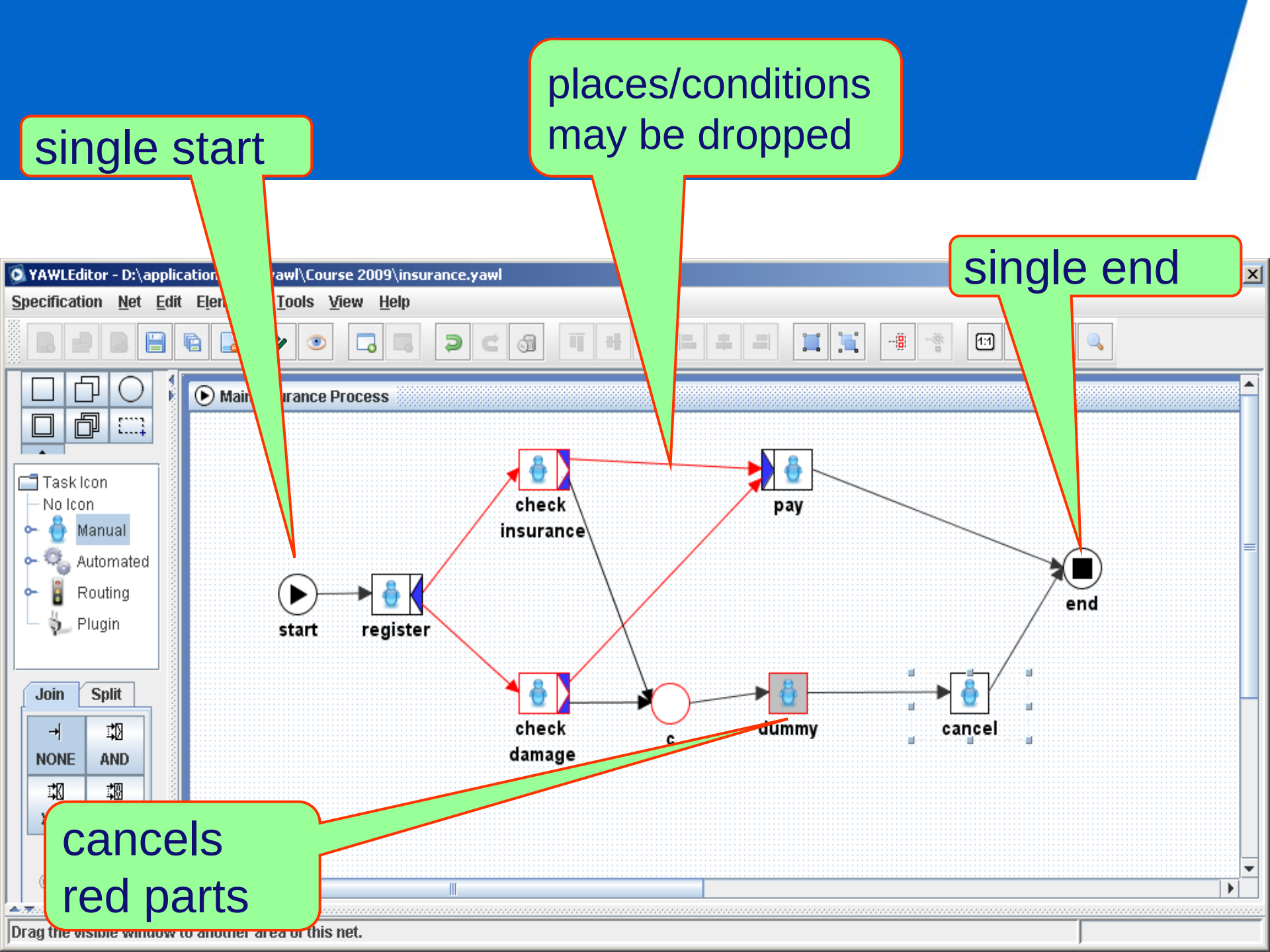




- In Editor connect to engine and resource service (under "Tools").
- The user name is always "admin" and the password is "YAWL".
- Also use this to access the run-time of YAWL (to administrate of execute work)



Control and data-flow in YAWL


single start
places/conditions may be dropped
single end
cancels red parts
YAWLEditor - D:\application
awl\Course 2009\insurance.yawl
Specification Net Edit Tools View Help
Main
rance Process
Task Icon
No Icon
Manual
Automated
Routing
Plugin
Join
Split
NONE
AND
start
register
check insurance
check damage
c
dummy
pay
cancel
end
Drag the visible window to another area of this net.

YAWLEditor - D:\application_data\yawl\Course 2009\insurance.yawl

Specification Net Edit Elements Tools View Help

Task Icon
No Icon
Manual
Automated
Routing
Plugin

Join Split
NONE AND
XOR OR

Main Insurance Process

start
register
check insurance
check damage
c
pay
dummy
cancel
end

net variables

Update Net Decomposition "Main Insurance Process"

Net Decomposition Label: Main Insurance Process

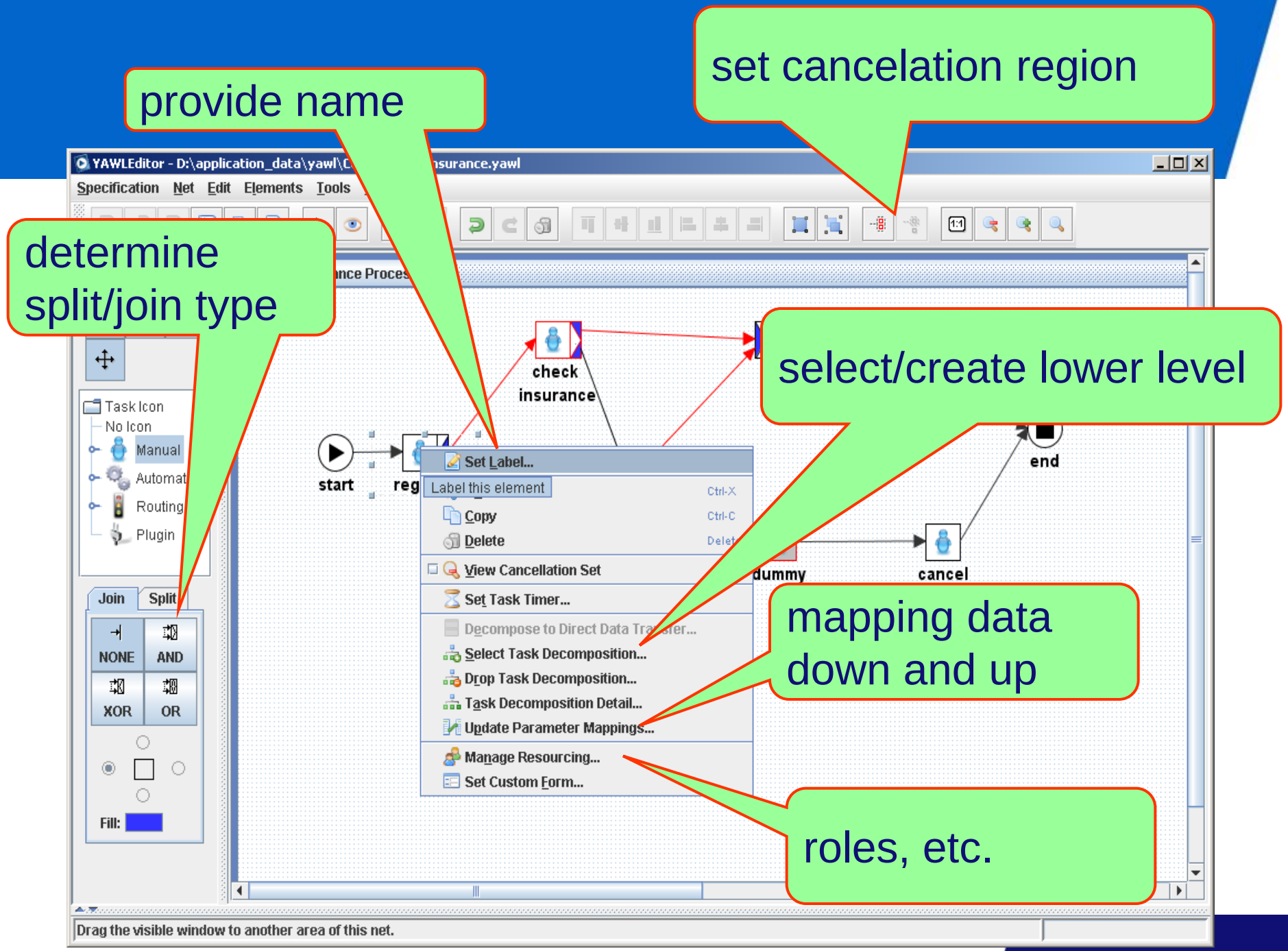
Net Decomposition Variables

Name	Type	Usage
Name	string	Local
InsuranceOK	boolean	Local
DamageOK	boolean	Local

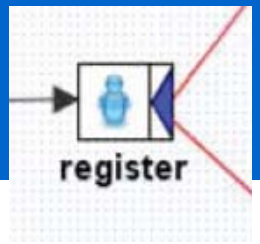
Create... Update... Remove...

Done Cancel

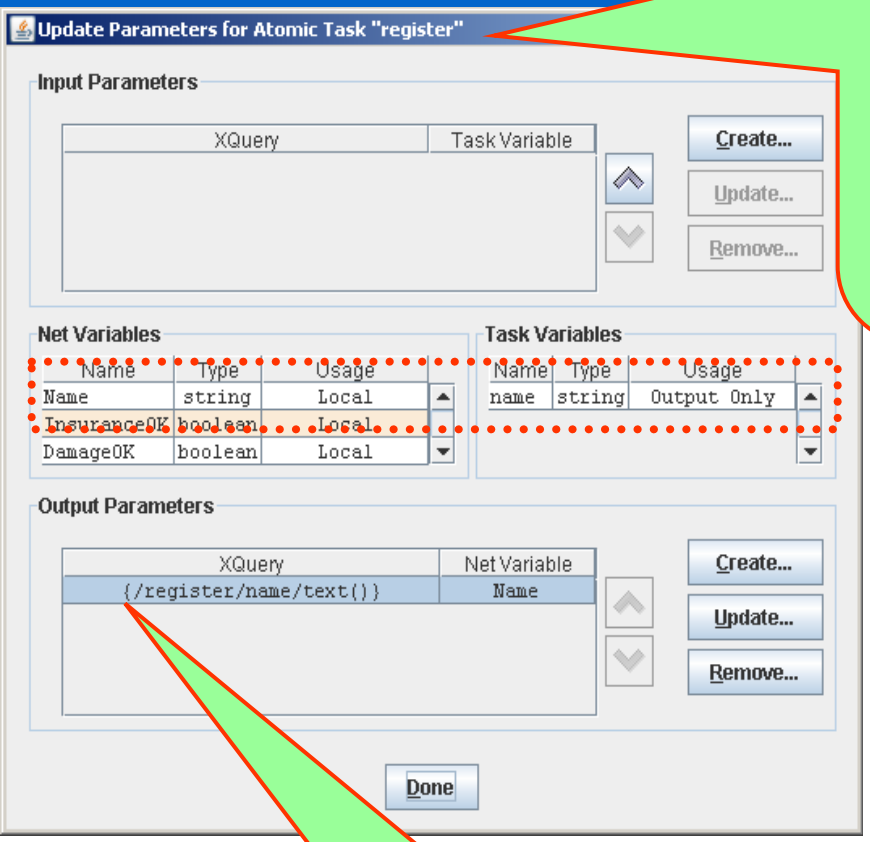
Drag the visible window to another area of this n

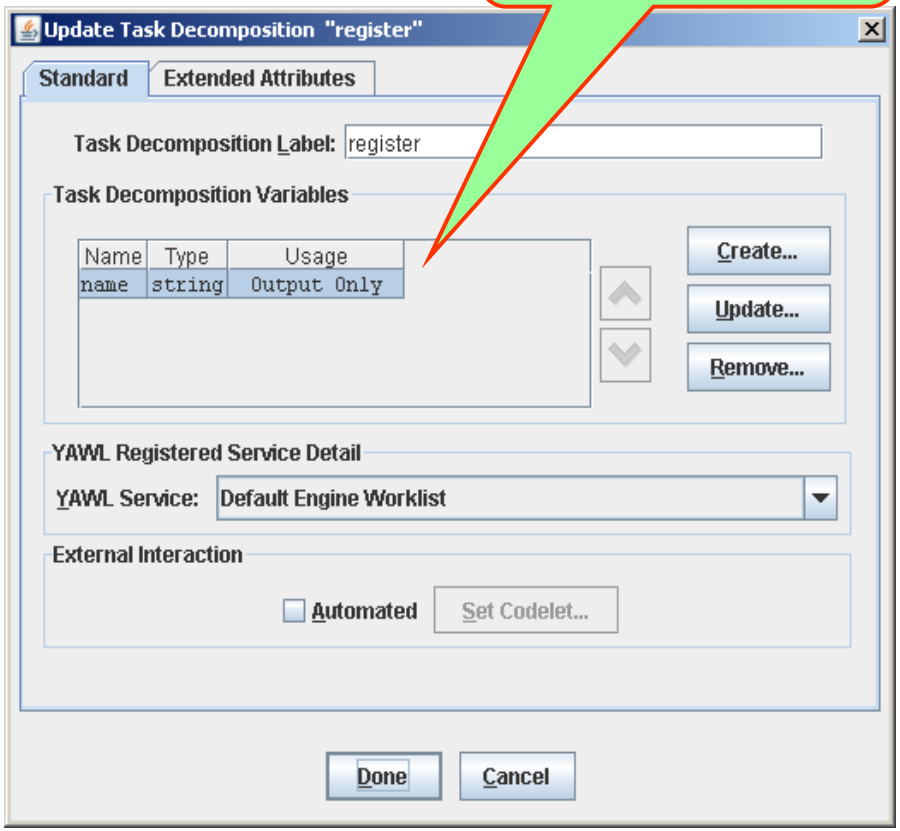

provide name
set cancelation region
determine split/join type
select/create lower level
mapping data down and up
roles, etc.
YAWLEditor - D:\application_data\yawl\C
nsurance.yawl
Specification
Net
Edit
Elements
Tools
Task Icon
No Icon
Manual
Automat
Routing
Plugin
Join
Split
NONE
AND
XOR
OR
Fill:
start
reg
check insurance
dummy
cancel
end
Set Label...
Label this element
Ctrl-X
Copy
Ctrl-C
Delete
View Cancellation Set
Set Task Timer...
Decompose to Direct Data Tra
Select Task Decomposition...
Drop Task Decomposition...
Task Decomposition Detail...
Update Parameter Mappings...
Manage Resourcing...
Set Custom Form...
Drag the visible window to another area of this net.

mapping from net level to task level and from task level to net level



task variable





only one mapping from task level to net level

mapping from net level to task level and from task level to net level

Update Parameters for Atomic Task "check insurance"

Input Parameters

XQuery	Task Variable
{/Main_Insurance_Process/Name/text()}	name

Create... Update... Remove...

down

Net Variables

Name	Type	Usage
Name	string	Local
InsuranceOK	boolean	Local
DamageOK	boolean	Local

Task Variables

Name	Type	Usage
name	string	Input Only
insuranceOK	boolean	Output Only

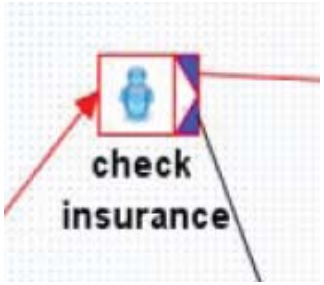
Output Parameters

XQuery	Net Variable
{/check_insurance/insuranceOK/text()}	InsuranceOK

Create... Update... Remove...

up

Done



Update Task Decomposition "check insurance"

Standard | Extended Attributes

Task Decomposition Label: check insurance

Task Decomposition Variables

Name	Type	Usage
name	string	Input Only
insuranceOK	boolean	Output Only

Create... Update... Remove...

YAWL Registered Service Detail

YAWL Service: Default Engine Worklist

External Interaction

Automated Set Codelet...

Done Cancel

YAWLEditor - D:\application_data\yawl\Course 2009\insurance.yawl

Specification Net Edit Elements Tools View Help

Main Insurance Process

start
register
check insurance
pay
check damage
c
dummy
cancel
end

Drag the visible window to another area of this net.

Update Net Decomposition "Main Insurance Process"

Net Decomposition Label: Main Insurance Process

Net Decomposition Variables

Name	Type	Usage
Name	string	Local
InsuranceOK	boolean	Local
DamageOK	boolean	Local

Create... Update... Remove...

Done Cancel

down

Update Task Decomposition "check insurance"

Standard Extended Attributes

Task Decomposition Label: check insurance

Task Decomposition Variables

Name	Type	Usage
name	string	Input Only
insuranceOK	boolean	Output Only

Update... Remove...

YAWL Registered Service Detail

YAWL Service: Default Engine Worklist

External Interaction

Automated Set Codelet...

Done Cancel

up

Update Parameters for Atomic Task "check insurance"

Input Parameters

XQuery	Task Variable
{/Main_Insurance_Process/Name/text()}	name

Create... Update... Remove...

Net Variables

Name	Type	Usage
Name	string	Local
InsuranceOK	boolean	Local
DamageOK	boolean	Local

Task Variables

Name	Type	Usage
name	string	Input Only
insuranceOK	boolean	Output Only

Output Parameters

XQuery	Net Variable
{/check_insurance/insuranceOK/text()}	InsuranceOK

Create... Update... Remove...

Done

YAWLEditor - D:\application_data\yawl\Course 2009\insurance.yawl

Specification Net Edit Elements Tools View Help

Task Icon
No Icon
Manual
Automated
Routing
Plugin

Join Split
NONE AND
XOR OR

Main Insurance Process

start
register
check insurance
check damage
pay
c
dummy
cancel
end

Drag the visible window to another area of this net.

Flow detail for Atomic Task "check insurance"

Target Task	Predicate
pay	/Main_Insurance_Process/...
c	/Main_Insurance_Process/...

Predicate...

The bottom-most flow will be used as the default.

Done

Update Flow Predicate

Net variable: Name XPath Expression

```
/Main_Insurance_Process/InsuranceOK/text()='true'
```

Update Flow Predicate

Net variable: Name XPath Expression

```
/Main_Insurance_Process/InsuranceOK/text()='false'
```

Done Cancel

YAWLEditor - D:\application_data\yawl\Course 2009\insurance.yawl

Specification Net Edit Elements Tools View Help

Task Icon
No Icon
Manual
Automated
Routing
Plugin

Join Split
NONE AND
XOR OR

Main Insurance Process

start
register
check insurance
check damage
pay
c
dummy
cancel
end

Drag the visible window to another area of this net.

Flow detail for Atomic Task "check damage"

Target Task	Predicate
pay	/Main_Insurance_Process/...
c	/Main_Insurance_Process/...

Predicate...

The bottom-most flow will be used as the default.

Done

Update Flow Predicate

Net variable: Name XPath Expression

```
/Main_Insurance_Process/DamageOK/text()='true'
```

Update Flow Predicate

Net variable: Name XPath Expression

```
/Main_Insurance_Process/DamageOK/text()='false'
```

Done Cancel

Update Task Decomposition "pay"

Standard | Extended Attributes

Task Decomposition Label: pay

Task Decomposition Variables

Name	Type	Usage
name	string	Input Only
insuranceOK	boolean	Input Only
damageOK	boolean	Input Only

Create... Update... Remove...

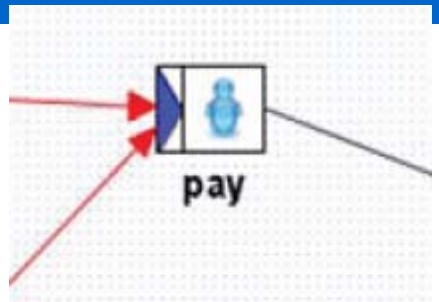
YAWL Registered Service Detail

YAWL Service: Default Engine Worklist

External Interaction

Automated Set Codelet...

Done Cancel



Update Parameters for Atomic Task "pay"

Input Parameters

XQuery	Task Variable
{/Main_Insurance_Process/Name/text()}	name
{/Main_Insurance_Process/InsuranceO...	insuranceOK
{/Main_Insurance_Process/DamageOK/t...	damageOK

Create... Update... Remove...

Net Variables

Name	Type	Usage
Name	string	Local
InsuranceOK	boolean	Local
DamageOK	boolean	Local

Task Variables

Name	Type	Usage
name	string	Input Only
insuranceOK	boolean	Input Only
damageOK	boolean	Input Only

Output Parameters

XQuery	Net Variable

Create... Update... Remove...

Done

Update Task Decomposition "cancel"

Standard | Extended Attributes

Task Decomposition Label: cancel

Task Decomposition Variables

Name	Type	Usage
name	string	Input Only

Create... Update... Remove...

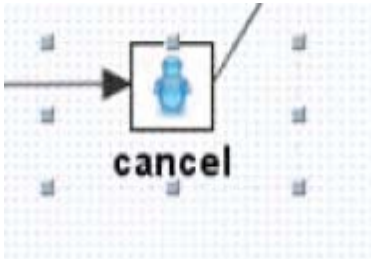
YAWL Registered Service Detail

YAWL Service: Default Engine Worklist

External Interaction

Automated Set Codelet...

Done Cancel



Update Parameters for Atomic Task "cancel"

Input Parameters

XQuery	Task Variable
{/Main_Insurance_Process/Nam...	name

Create... Update... Remove...

Net Variables

Name	Type	Usage
Name	string	Local
InsuranceOK	boolean	Local
DamageOK	boolean	Local

Task Variables

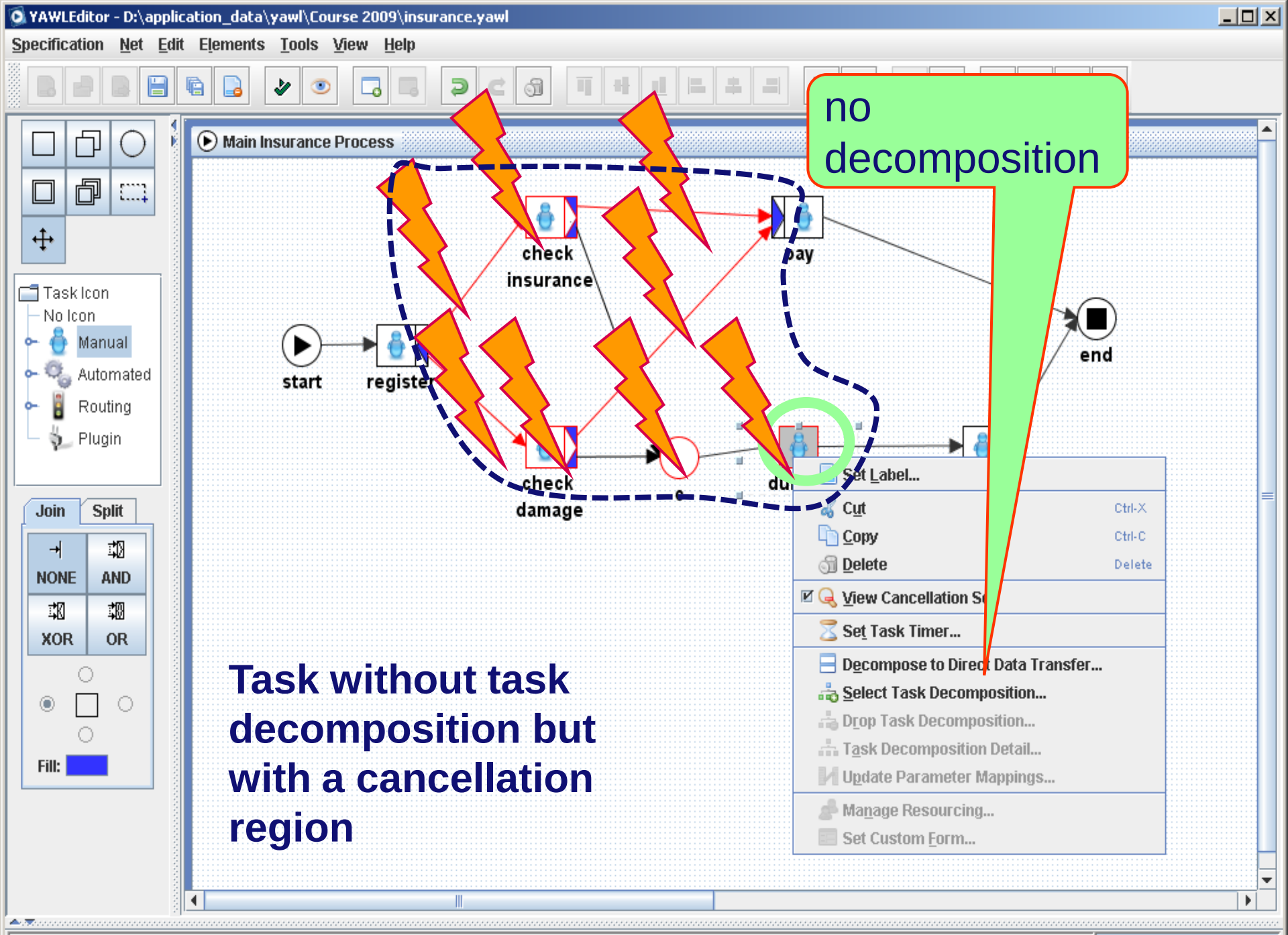
Name	Type	Usage
name	string	Input Only

Output Parameters

XQuery	Net Variable

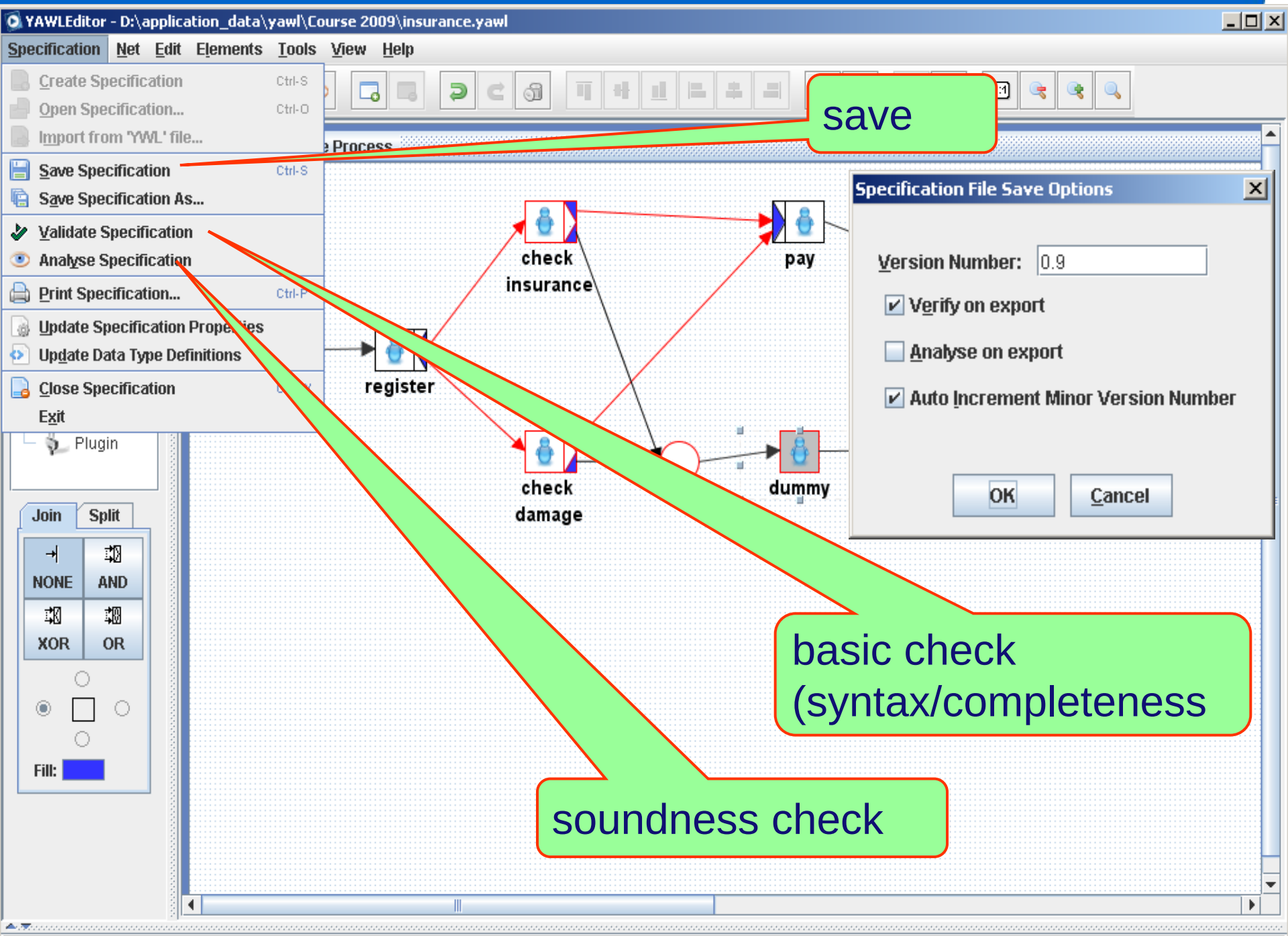
Create... Update... Remove...

Done



Task without task decomposition but with a cancellation region

Analysis in YAWL


YAWLEditor - D:\application_data\yawl\Course 2009\insurance.yawl
Specification
Net
Edit
Elements
Tools
View
Help
Create Specification
Ctrl-S
Open Specification...
Ctrl-O
Import from 'YWL' file...
Save Specification
Ctrl-S
Save Specification As...
Validate Specification
Analyse Specification
Print Specification...
Ctrl-P
Update Specification Properties
Update Data Type Definitions
Close Specification
Exit
save
Specification File Save Options
Version Number:
0.9
Verify on export
Analyse on export
Auto Increment Minor Version Number
OK
Cancel
check insurance
pay
register
check damage
dummy
Plugin
Join
Split
NONE
AND
XOR
OR
Fill:
basic check (syntax/completeness
soundness check
Drag the visible window to another area of this net.

YAWLEditor - D:\application_data\yawl\Course 2009\insurance.yawl

Specification Net Edit Elements Tools View Help

Main Insurance Process

start
register
check insurance
check damage
pay
c
dummy
cancel
end

Task Icon
No Icon
Manual
Automated
Routing
Plugin

Join Split
NONE AND
XOR OR
Fill:

Notes (dummy) Specification Analysis Problems

No problems were discovered in the analysis of this specification.

Drag the visible window to another area of this net.

YAWLEditor - D:\application_data\yawl\Course 2009\insurance_error1.yawl

Specification Net Edit Elements Tools View Help

Task Icon
No Icon
Manual
Automated
Routing
Plugin

Main Insurance Process

start
register
check insuranc
check damage
c
pay
dummy
cancel
end

Notes | Specification Analysis Problems

ResetNet Analysis Warning: The net Main_Insurance_Process does not have proper completion. A marking 2OutputCondition_2 larger than...
ResetNet Analysis Warning: The net Main_Insurance_Process does not satisfy the soundness property.

ResetNet Analysis Warning: The net Main_Insurance_Process does not have proper completion. A marking 2OutputCondition_2 larger than the final marking is reachable.The net Main_Insurance_Process can deadlock at
marking(s):1c{check_damage_5_pay_6}+1OutputCondition_2
1c{check_insurance_4_pay_6}+1OutputCondition_2

Drag the visible window to another area of this net.

YAWLEditor - D:\application_data\yawl\Course 2009\insurance_error2.yawl

Specification Net Edit Elements Tools View Help

Task Icon
No Icon
Manual
Automated
Routing
Plugin

Join Split
NONE AND
XOR OR
Fill:

Main Insurance Process

start
register
check insurance
check damage
pay
cancel
end

Notes (check damage) Specification Analysis Problems

ResetNet Analysis Warning: The net Main_Insurance_Process does not have proper completion. A marking 2OutputCondition_2 larger than...
ResetNet Analysis Warning: The net Main_Insurance_Process does n

ResetNet Analysis Warning: The net Main_Insurance_Process does not have proper completion. A marking 2OutputCondition_2 larger than the final marking is reachable.The net Main_Insurance_Process can deadlock at marking(s):1c{check_insurance_4_pay_6}+1OutputCondition_2 1c{check_damage_5_pay_6}+1OutputCondition_2

Drag the visible window to another area of this net.

YAWLEditor - D:\application_data\yawl\Course 2009\insurance-one-choice.yawl

Specification Net Edit Elements Tools View Help

Main Insurance Process

start
register
check insurance
check damage
decide
pay
cancel
end

Task Icon
No Icon
Manual
Automated
Routing
Plugin

Join Split
NONE AND
XOR OR
Fill:

Flow detail for Atomic Task "decide"

Target Task	Predicate
pay	(/Main_Insurance_Process...
cancel	(/Main_Insurance_Process...

Predicate...

The bottom-most flow will be used as the default.

Done

Update Flow Predicate

Net variable: Name XPath Expression

```
(/Main_Insurance_Process/InsuranceOK/text()='true')
and
(/Main_Insurance_Process/DamageOK/text()='true')
```

Done Cancel

Notes (decide) Specification Analysis Problems

No problems were discovered in the analysis of this specification.

Update Flow Predicate

Net variable: Name XPath Expression

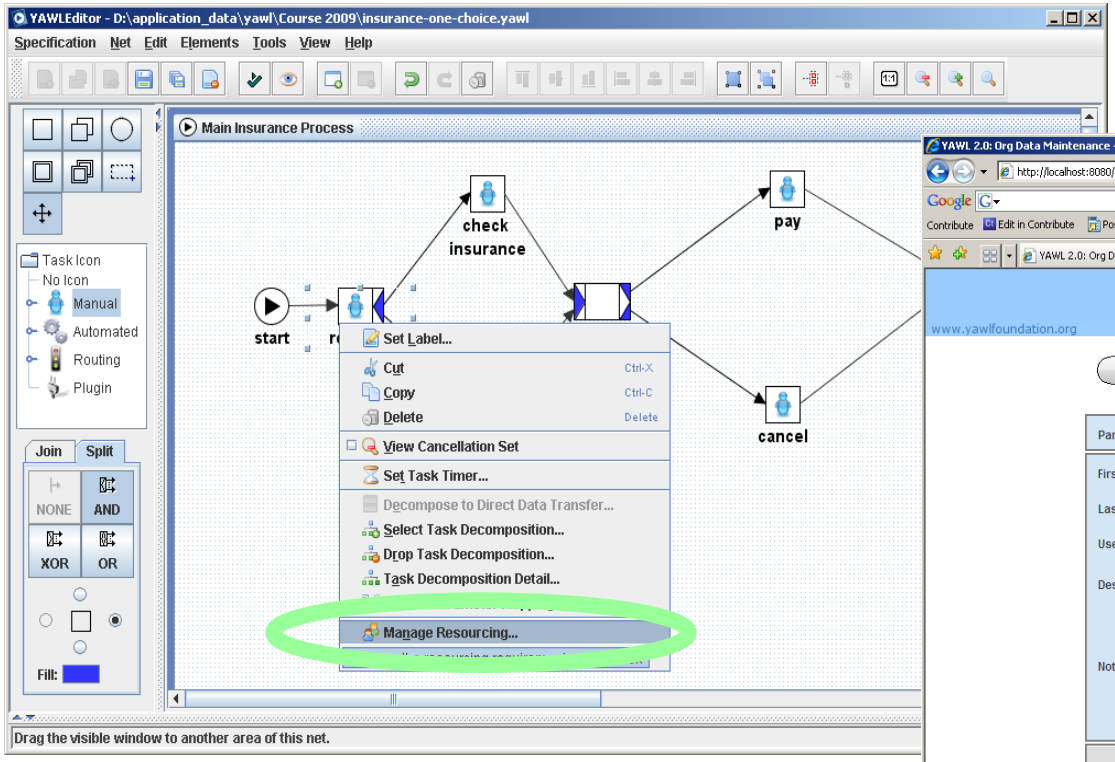
```
(/Main_Insurance_Process/InsuranceOK/text()='false')
or
(/Main_Insurance_Process/DamageOK/text()='false')
```

Done Cancel

Drag the visible window to another area of this net.

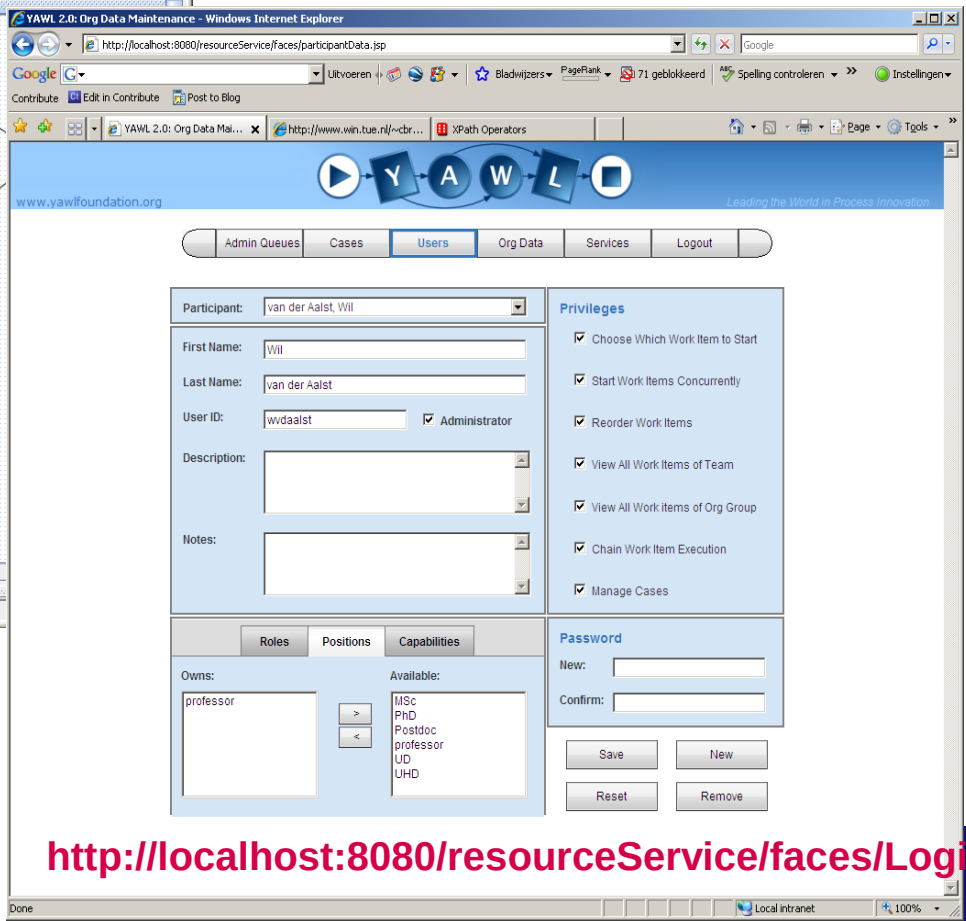
Resources in YAWL

Resource information resides at two places: Editor and YAWL Control Center

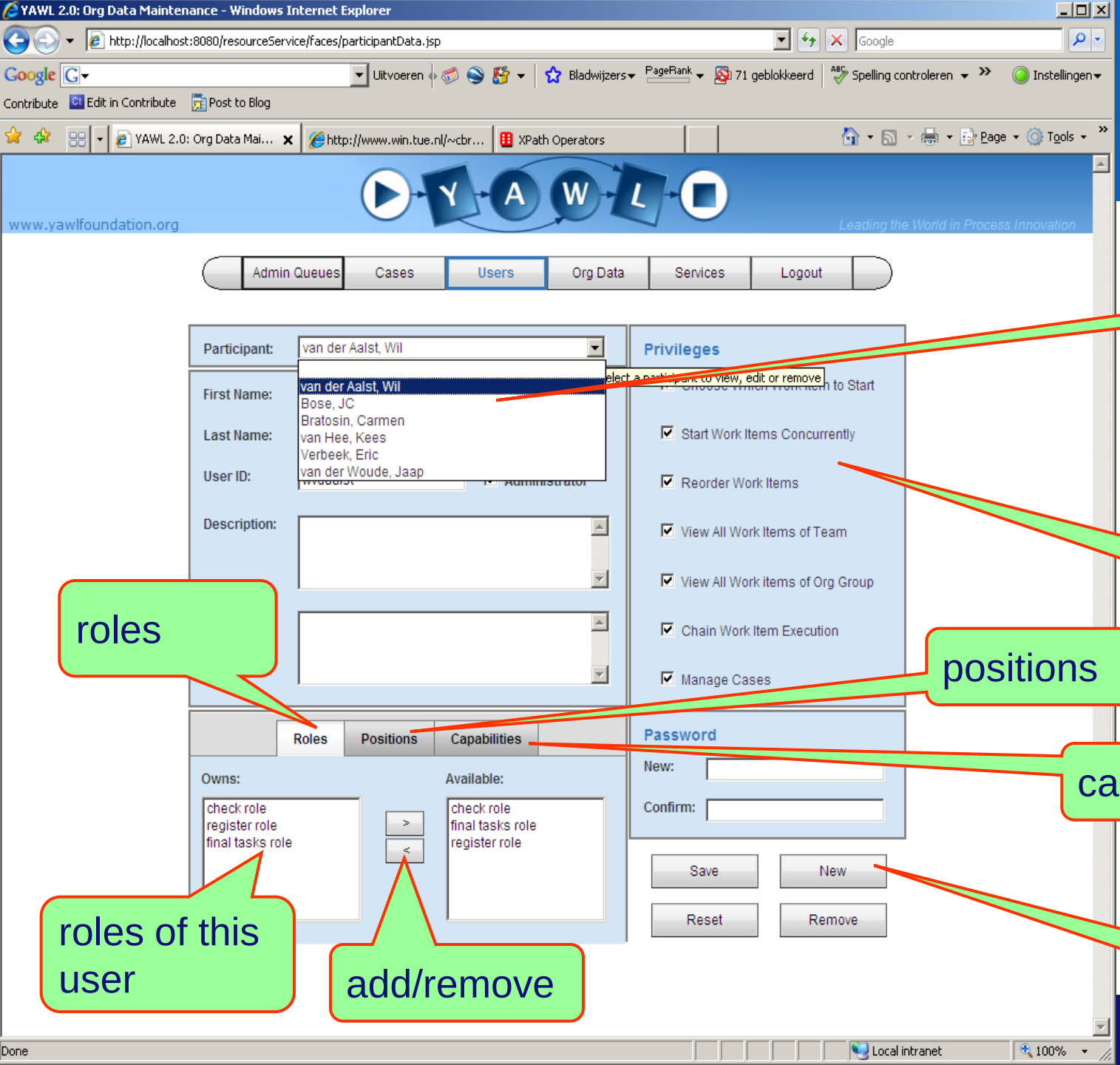


(1) Editor: specify roles, etc. per task

(2) Control Center: Users and Org Data



http://localhost:8080/resourceService/faces/Logi


YAWL 2.0: Org Data Maintenance - Windows Internet Explorer
http://localhost:8080/resourceService/faces/participantData.jsp
www.yawlfoundation.org
Leading the World in Process Innovation
Admin Queues
Cases
Users
Org Data
Services
Logout
Participant:
van der Aalst, Wil
Bose, JC
Bratosin, Carmen
van Hee, Kees
Verbeek, Eric
van der Woude, Jaap
First Name:
Last Name:
User ID:
Description:
Privileges
Start Work Items Concurrently
Reorder Work Items
View All Work Items of Team
View All Work items of Org Group
Chain Work Item Execution
Manage Cases
Roles
Positions
Capabilities
Owns:
check role
register role
final tasks role
Available:
check role
final tasks role
register role
Password
New:
Confirm:
Save
New
Reset
Remove
users
privileges
roles
positions
capabilities
roles of this user
add/remove
add new user

YAWL 2.0: Organisational Data Management - Windows Internet Explorer

http://localhost:8080/resourceService/faces/participantData.jsp

Google | Uitvoeren | Bladwijzers | PageRank | 71 geblokkeerd | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0: Organisationa... | http://www.win.tue.nl/~cbr... | XPath Operators | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Admin Queues | Cases | Users | Org Data | Services | Logout

Roles | Capabilities | Positions | OrgGroups

Role Names

- check role
- final tasks role
- register role

Description

Notes

Belongs To

nil

Save | New

Reset | Remove

Error on page.

Local intranet | 100%

YAWL 2.0: Organisational Data Management - Windows Internet Explorer

http://localhost:8080/resourceService/faces/orgDataMgt.jsp

Google | Uitvoeren | Bladwijzers | PageRank | 71 geblokkeerd | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0: Organisationa... | http://www.win.tue.nl/~cbr... | XPath Operators | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Admin Queues | Cases | Users | Org Data | Services | Logout

Roles | Capabilities | Positions | OrgGroups

Capability Names

- Fluent in Dutch
- Fluent in English

Description

Notes

Save | New

Reset | Remove

Done | Local intranet | 100%

YAWL 2.0: Organisational Data Management - Windows Internet Explorer

http://localhost:8080/resourceService/faces/orgDataMgt.jsp

Google | Uitvoeren | Bladwijzers | PageRank | 71 geblokkeerd | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0: Organisationa... | http://www.win.tue.nl/~cbr... | XPath Operators | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Admin Queues | Cases | Users | Org Data | Services | Logout

Roles | Capabilities | Positions | OrgGroups

Position Titles

- MSc
- PhD
- Postdoc
- professor
- UD
- UHD

Description

Notes

Reports To: nil

Org Group: TUE

Save | New | Reset | Remove

Done | Local intranet | 100%

YAWL 2.0: Organisational Data Management - Windows Internet Explorer

http://localhost:8080/resourceService/faces/orgDataMgt.jsp

Google | Uitvoeren | Bladwijzers | PageRank | 71 geblokkeerd | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0: Organisationa... | http://www.win.tue.nl/~cbr... | XPath Operators | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Admin Queues | Cases | Users | Org Data | Services | Logout

Roles | Capabilities | Positions | OrgGroups

Org Group Titles

- AIS
- DH
- TM
- TUE
- WIN

Description

Notes

Belongs To: WIN

Save | New | Reset | Remove

Done | Local intranet | 100%

Some hints

- **Make sure to complete the organizational model before referring to it in the Editor.**
- **The Editor needs to connect to the Engine and Resource service (default usercode: admin and password: YAWL).**
- **Remember the login data for added users!**
- **The runtime user interface (YAWL Control Center) depends on whether one is administrator or not!**

YAWL 2.0 Admin Worklist - Windows Internet Explorer

http://localhost:8080/resourceService/faces/adminQueues.jsp

Work Queues | Edit Profile | Team Queues | Admin Queues | Cases | Users | Org Data | Services | Logout

Unoffered (0) | Worklisted (1)

Work Items: 15:pay_6

Specification: insurance.yawl

Task: pay

Case: 15

Status: Enabled

Created: Mar:13, 2009 19:13:54

Age: 0:16:55:5

Assigned To (5): van der Aalst, Wil

Resource State: Offered

User with administrator rights

YAWL 2.0 Worklist: Eric Verbeek - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

www.yawlfoundation.org

Leading the World in Process Innovation

Work Queues | Edit Profile | Logout

Offered (1) | Allocated (0) | Started (0) | Suspended (0)

Work Items: 15:pay_6

Specification: insurance.yawl

Task: pay

Case: 15

Status: Enabled

Created: Mar:13, 2009 19:13:54

Age: 0:00:00:02

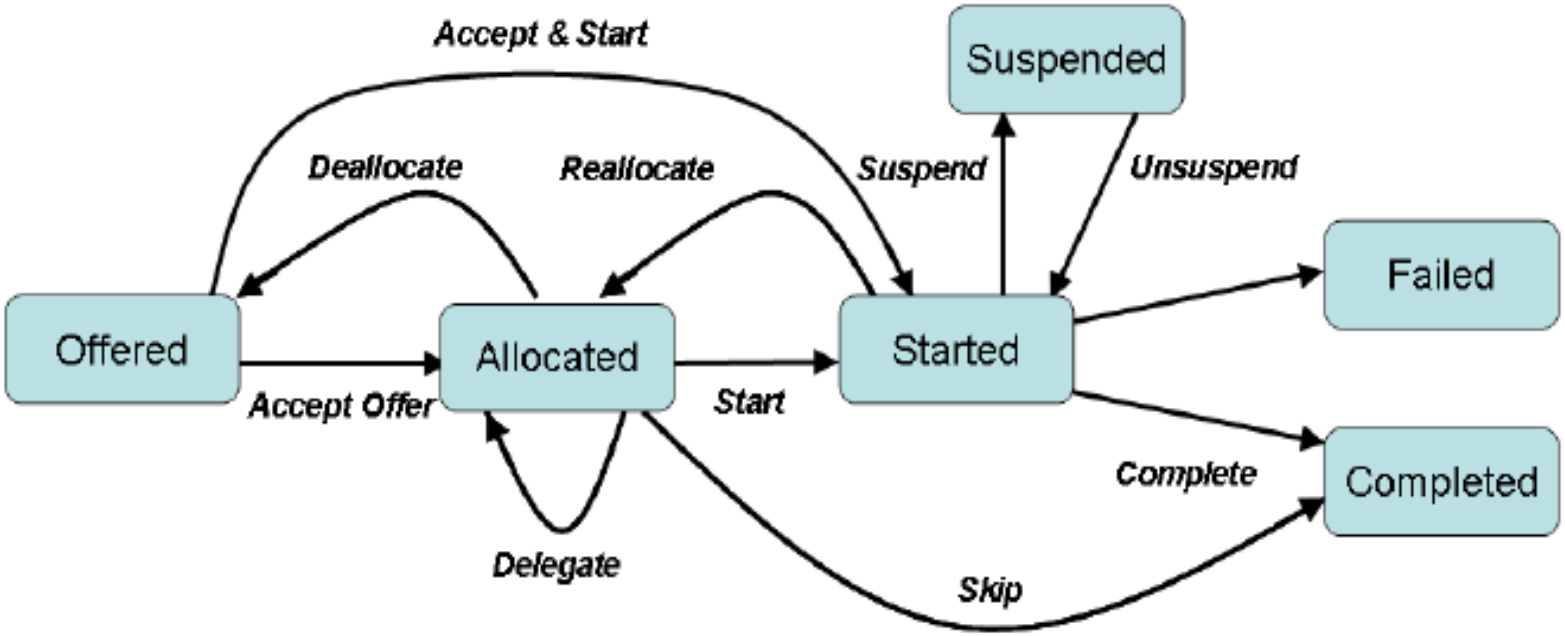
Accept Offer

Accept & Start

Chain

Normal user

Lifecycle of work-items in YAWL



Manage Resourcing Wizard for Atomic Task "register"

Step 1 : Choose Behaviour At Interaction Points

There are three key decision points for managing the resourcing of work items spawned from a task. At each of these *interaction points*, you may choose to have the system dynamic make a decision on resourcing at each point, or alternately, allow a user to manually make each decision. Each interaction point is briefly described below:

- Offer: The point at which it is decided that a number of participants *could* undertake the work item.
- Allocation: The point at which one of the participants offered the work item is *nominated to do* that work item.
- Start: The point at which the participant allocated a work item *begins* working on it.

Offering a work item for this task to a number of participants is to be done by:	○ User	● System
Allocating a work item for this task to one of the offered participants is to be done by:	○ User	● System
Starting an allocated work item of this task is to be done by:	○ User	● System

Default setting is (System, User, User)

< Back | > Next | Finish

Manage Resourcing Wizard for Atomic Task "register"

Step 2 : Specify System Behaviour when Offering a Work Item

The offer process involves choosing which participants should be informed of the existence of the work item, one of whom will eventually do this work. As you have specified the system manage the offer process, you must now choose who the work item should be offered to. Begin by specifying a set of participants and/or to distribute offers of work to. You may also specify a net parameter which at runtime will contain a participant's userid or the name of a role.

Participants

- Wil van der Aalst (wvdaalst)
- JC Bose (jbose)
- Carmen Bratosin (cbratosin)
- Kees van Hee (kvhee)
- Eric Verbeek (everbeek)
- Jaap van der Woude (jvdwoude)

Unselect All

Roles

- check role
- final tasks role
- register role

Unselect All

Net Parameters

Name	Refers to

< Back > Next Finish

Manage Resourcing Wizard for Atomic Task "register"

Step 3 : Specify Distribution Set Filter(s)

Filters

Select the filters to be applied to the distribution set. Set parameter values for the selected filter as required.

	Filter
☑	Filter by Capability
☑	Filter by Organisational Data

Parameter	Value
OrgGroup	AIS
Position	professor

Runtime Constraints

☐ Allow this task to be piled to a single participant.

☐ Choose participant(s) who completed previous task:

☐ Do not choose participant(s) who completed previous task:

< Back > Next Finish

Manage Resourcing Wizard for Atomic Task "register"

Step 4 : Specify System Behaviour when Allocating a Work Item

The allocation process involves choosing a single participant, from those who are offered a work item, to actually undertake that work. As you have specified that the system dynamically do this, you must now select the strategy for doing so. Choose from one of the strategies below.

Choose the runtime allocation strategy: Random Choice

- Shortest Queue
- Round Robin (by frequency)
- Random Choice
- Round Robin (by time)

< Back > Next Finish

Manage Resourcing Wizard for Atomic Task "register"

Step 5 : Establish Default User Runtime Privileges for this Task

Can a participant suspend a started work item of this task?	◉ No	○ Yes
Can a participant reallocate a work item of this task to another participant, resetting state?	◉ No	○ Yes
Can a participant reallocate a work item of this task to another participant, retaining state?	◉ No	○ Yes
Can a participant deallocate themselves from a work item of this task?	◉ No	○ Yes
Can a participant delegate a work item of this task to another participant?	◉ No	○ Yes
Can a participant skip a work item of this task?	◉ No	○ Yes

< Back > Next Finish

Alternative scenarios

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Manage Resourcing Wizard for Atomic Task "register"

Step 1 : Interaction Points

Each task passes through three decisions, or *interaction* points before a participant begins working on it. For each of the interaction points below, please specify whether the task is to be handled by the *System* (dynamically, based on the settings chosen later in this wizard), or by the *User* (manually, by a participant or an administrator), when the task is executed.

Offer: The task is made available to a number of participants:	◉ User	○ System
Allocate: The task is assigned to a single participant:	◉ User	○ System
Start: Work begins on the task:	◉ User	○ System

Manage Resourcing Wizard for Atomic Task "register"

Step 5 : Establish Default User Runtime Privileges for this Task

Can a participant suspend a started work item of this task?	○ No	◉ Yes
Can a participant reallocate a work item of this task to another participant, resetting state?	◉ No	○ Yes
Can a participant reallocate a work item of this task to another participant, retaining state?	○ No	◉ Yes
Can a participant deallocate themselves from a work item of this task?	○ No	◉ Yes
Can a participant delegate a work item of this task to another participant?	◉ No	○ Yes
Can a participant skip a work item of this task?	○ No	◉ Yes

< Back | > Next | Finish

Manage Resourcing Wizard for Atomic Task "register"

Step 1 : Interaction Points

Manage Resourcing Wizard for Atomic Task "register"

Step 2 : Specify System Behaviour when Offering a Work Item

The offer process involves choosing which participants should be informed of the existence of the work item, one of whom will eventually do this wo
be offered to. Begin b
which at runtime will

Participants

Wil van der Aalst (
Boudewijn van Do
Kees van Hee (kvh
Eric Verbeek (eve

Un

Manage Resourcing Wizard for Atomic Task "register"

Step 3 : Specify Distribution Set Filter(s)

Filters

Select the filters to b

☐	Filter by Cap
☑	Filter by Org

Runtime Constraints

Manage Resourcing Wizard for Atomic Task "register"

Step 5 : Establish Default User Runtime Privileges for this Task

Can a participant suspend a started work item of this task?	○ No	● Yes
Can a participant reallocate a work item of this task to another participant, resetting state?	● No	○ Yes
Can a participant reallocate a work item of this task to another participant, retaining state?	○ No	● Yes
Can a participant deallocate themselves from a work item of this task?	○ No	● Yes
Can a participant delegate a work item of this task to another participant?	● No	○ Yes
Can a participant skip a work item of this task?	○ No	● Yes

< Back > Next Finish

Manage Resourcing Wizard for Atomic Task "register"

Step 1 : Interaction Points

Each task passes through three decisions, or *interaction* points before a participant

Manage Resourcing Wizard for Atomic Task "register"

Step 4 : Specify System Behaviour when Allocating a Work Item

The allocation process involves c
work. As you have specified that
the strategies below.

Choos

Manage Resourcing Wizard for Atomic Task "register"

Step 5 : Establish Default User Runtime Privileges for this Task

Can a participant suspend a started work item of this task?	○ No	● Yes
Can a participant reallocate a work item of this task to another participant, resetting state?	● No	○ Yes
Can a participant reallocate a work item of this task to another participant, retaining state?	○ No	● Yes
Can a participant deallocate themselves from a work item of this task?	○ No	● Yes
Can a participant delegate a work item of this task to another participant?	● No	○ Yes
Can a participant skip a work item of this task?	○ No	● Yes

< Back > Next Finish

Executing work in YAWL

YAWL 2.0 Case Management - Windows Internet Explorer

http://localhost:8080/resourceService/faces/

Google

Uitvoeren

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Ca... | main_page [YA...

www.yawlfoundation.org

Work Queues | Edit Profile | Team Queues | Admin Queues | Cas

Upload Specification

D:\application_data\yawl\Course 2009\insurance.yawl | Browse...

Upload File

Loaded Specifications

insurance.yawl	0.9	No description has been giv

Launch Case | Unload Spec

Done | Local intranet | 100%

YAWL 2.0 Case Management - Windows Internet Explorer

http://localhost:8080/resourceService/faces/

Google

Uitvoeren | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Ca... | main_page [YA...

Page | Tools

Browse...

Upload File

Loaded Specifications

insurance.yawl	0.9	No description has been given.

Launch Case | Unload Spec

Running Cases

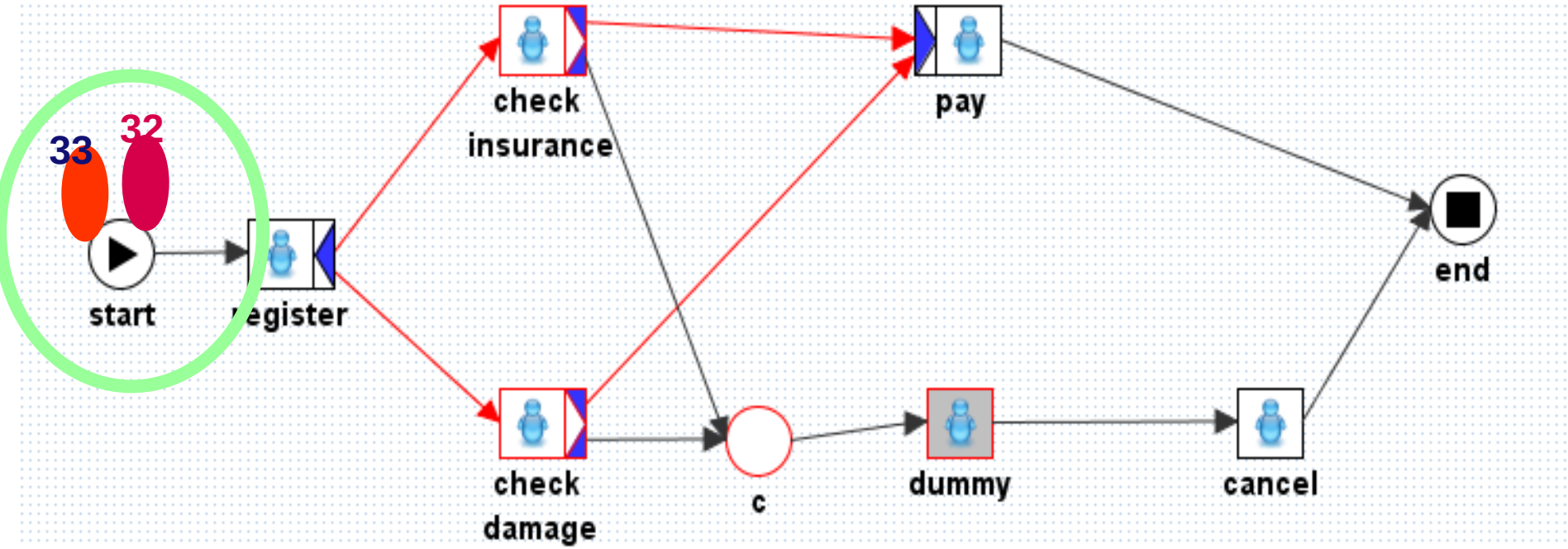
32: insurance.yawl (0.9)
33: insurance.yawl (0.9)

Cancel Case

Done | Local intranet | 100%

Loading processes and creating

Specification ID: insurance.yawl, Net ID: Main_Insurance_Process



YAWL 2.0 Worklist: Eric Verbeek - Windows Internet Explorer

http://localhost:8080/resourceService/faces/Login.jsp

Google | Uitvoeren | Bladwijzers | PageRank | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Worklist: Eric V... | main_page [YAWL Book] | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Work Queues | Edit Profile | Logout

Offered (2) | Allocated (0) | Started (0) | Suspended (0)

Work Items

32:register_3
33:register_3

Specification: insurance.yawl

Task: register

Case: 32

Status: Enabled

Created: Mar:14, 2009 15:10:55

Age: 0:00:05:43

Accept Offer

Accept & Start

Chain

Done | Local intranet | 100%

YAWL 2.0 Worklist: Eric Verbeek - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google | Uitvoeren | Bladwijzers | PageRank | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Worklist: Eric V... | main_page [YAWL Book] | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Work Queues | Edit Profile | Logout

Offered (1) | Allocated (0) | Started (1) | Suspended (0)

Work Items

32.1:register_3

Specification: insurance.yawl

Task: register

Case: 32.1

Status: Executing

Created: Mar:14, 2009 15:10:55

Age: 0:00:11:54

View/Edit

Suspend

Reallocate s/l

Reallocate s/f

New Instance

Complete

Local intranet | 100%

YAWL 2.0 - Edit Work Item - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google Uitvoeren Bladwijzers PageRank Instellingen

Contribute Edit in Contribute Post to Blog

YAWL 2.0 - Edit Work Item | main_page [YAWL Book] | Page Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Edit Work Item: 32.1

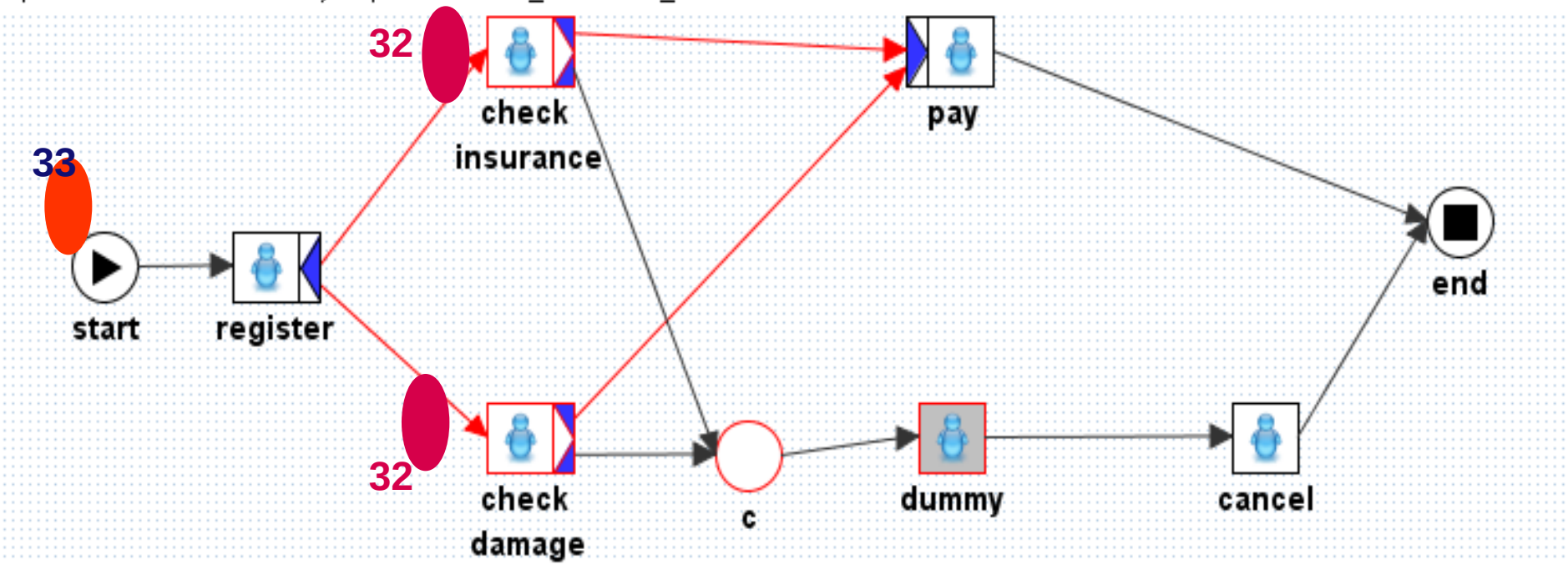
register

name: Case Jansen

Cancel Save Complete

Done Local intranet 100%

Specification ID: insurance.yawl, Net ID: Main_Insurance_Process



YAWL 2.0 Worklist: Eric Verbeek - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google | Uitvoeren | Bladwijzers | PageRank | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Worklist: Eric V... | main_page [YAWL Book] | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Work Queues | Edit Profile | Logout

Offered (3) | Allocated (0) | Started (0) | Suspended (0)

Work Items

- 33:register_3
- 32:check_insurance_4
- 32:check_damage_5

Specification: insurance.yawl

Task: check insurance

Case: 32

Status: Enabled

Created: Mar:14, 2009 15:25:08

Age: 0:00:01:59

Accept Offer

Accept & Start

Chain

http://localhost:8080/resourceService/faces/userWorkQueues.jsp# | Local intranet | 100%

YAWL 2.0 Worklist: Eric Verbeek - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google | Uitvoeren | Bladwijzers | PageRank | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Worklist: Eric V... | main_page [YAWL Book] | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Work Queues | Edit Profile | Logout

Offered ( ) | Allocated (1) | Started (0) | Suspended (0)

Work Items

32:check_insurance_4

Specification: insurance.yawl

Task: check insurance

Case: 32

Status: Enabled

Created: Mar:14, 2009 15:25:08

Age: 0:00:02:50

Start

Deallocate

Delegate

Skip

Pile

Done | Local intranet | 100%

YAWL 2.0 Worklist: Eric Verbeek - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google | Uitvoeren | Bladwijzers | PageRank | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Worklist: Eric V... | main_page [YAWL Book] | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Work Queues | Edit Profile | Logout

Offered (2) | Allocated (0) | Started (1) | Suspended (0)

Work Items

32.2:check_insurance_4

Specification: insurance.yawl

Task: check insurance

Case: 32.2

Status: Executing

Created: Mar:14, 2009 15:25:08

Age: 0:00:04:05

View/Edit

Suspend

Reallocate s/l

Reallocate s/f

New Instance

Complete

Local intranet | 100%

YAWL 2.0 - Edit Work Item - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google Uitvoeren Bladwijzers PageRank Instellingen

Contribute Edit in Contribute Post to Blog

YAWL 2.0 - Edit Work Item | main_page [YAWL Book] | Page Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Edit Work Item: 32.2

check insurance

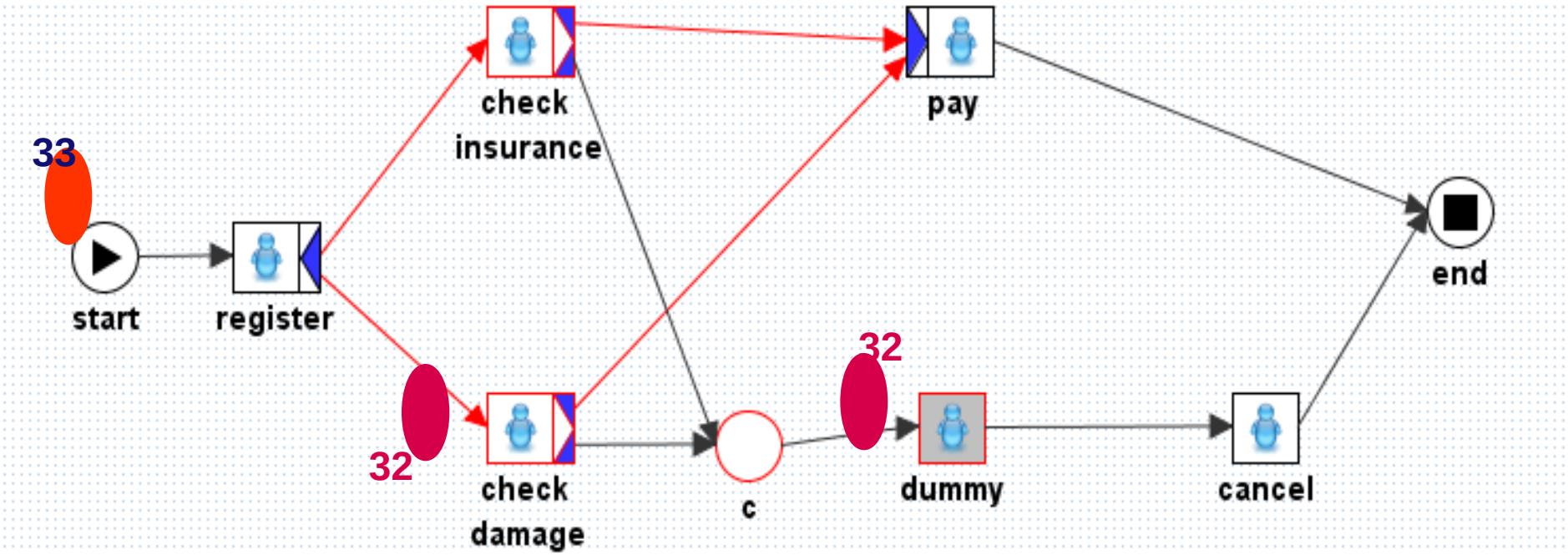
insuranceOK: ☐

name: Case Jansen

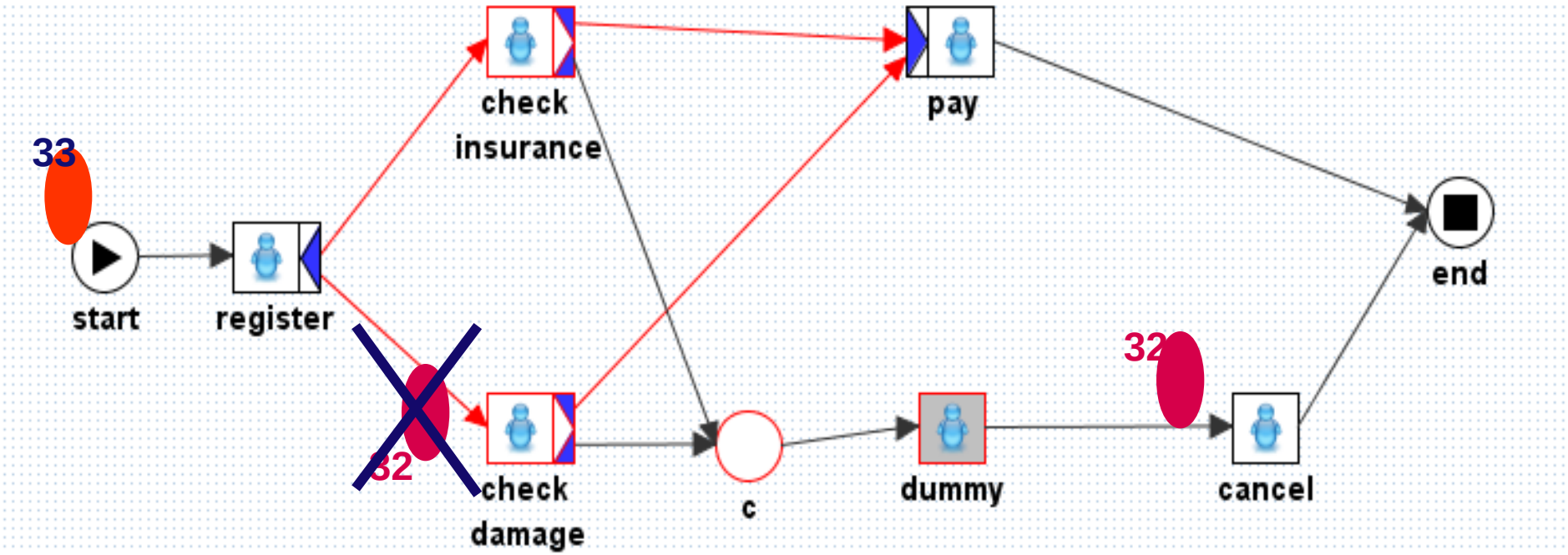
Cancel Save Complete

Done Local intranet 100%

Specification ID: insurance.yawl, Net ID: Main_Insurance_Process



Specification ID: insurance.yawl, Net ID: Main_Insurance_Process



YAWL 2.0 Worklist: Eric Verbeek - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google | Uitvoeren | Bladwijzers | PageRank | Instellingen

Contribute | Edit in Contribute | Post to Blog

YAWL 2.0 Worklist: Eric V... | main_page [YAWL Book] | Page | Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Work Queues | Edit Profile | Logout

Offered (2) | Allocated (0) | Started (0) | Suspended (0)

Work Items

33:register_3
32:cancel_7

Specification: insurance.yawl

Task: cancel

Case: 32

Status: Enabled

Created: Mar:14, 2009 15:30:42

Age: 0:00:00:05

Accept Offer

Accept & Start

Chain

Done | Local intranet | 100%

YAWL 2.0 - Edit Work Item - Windows Internet Explorer

http://localhost:8080/resourceService/faces/userWorkQueues.jsp

Google Uitvoeren Bladwijzers PageRank Instellingen

Contribute Edit in Contribute Post to Blog

YAWL 2.0 - Edit Work Item | main_page [YAWL Book] | Page Tools

www.yawlfoundation.org

Leading the World in Process Innovation

Edit Work Item: 32.4

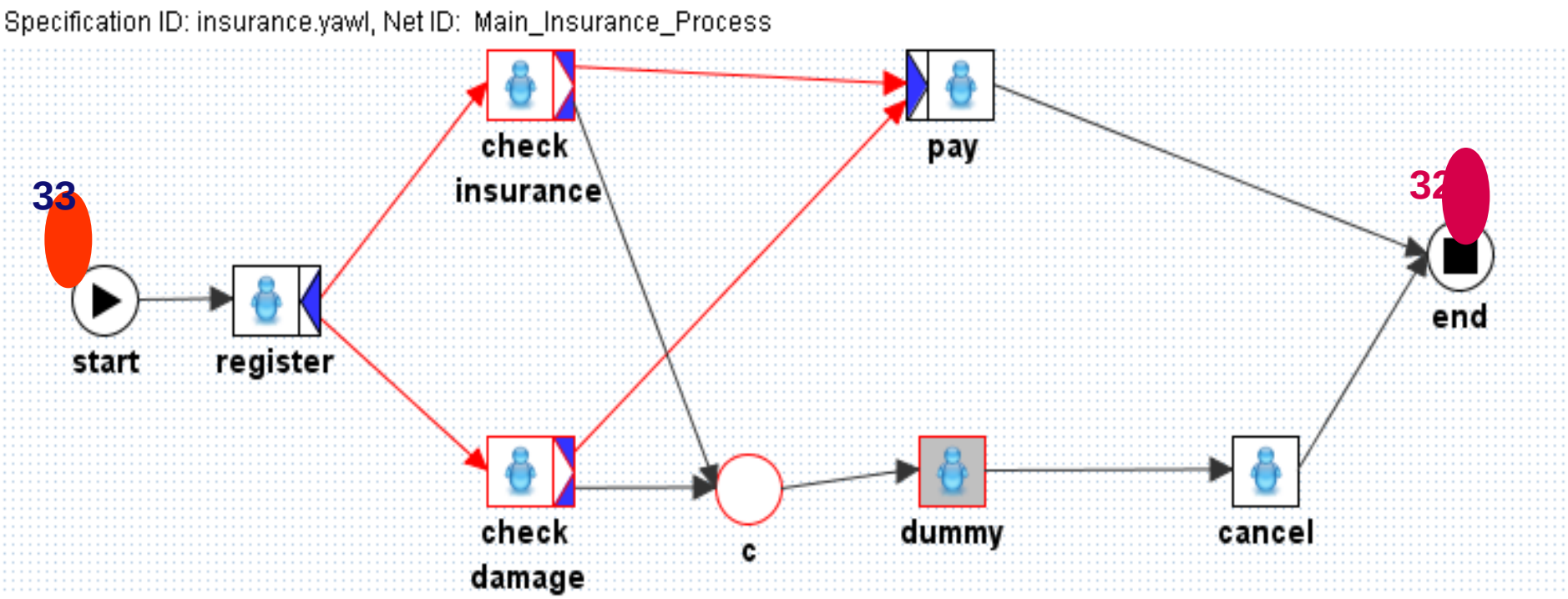
cancel

name: Case Jansen

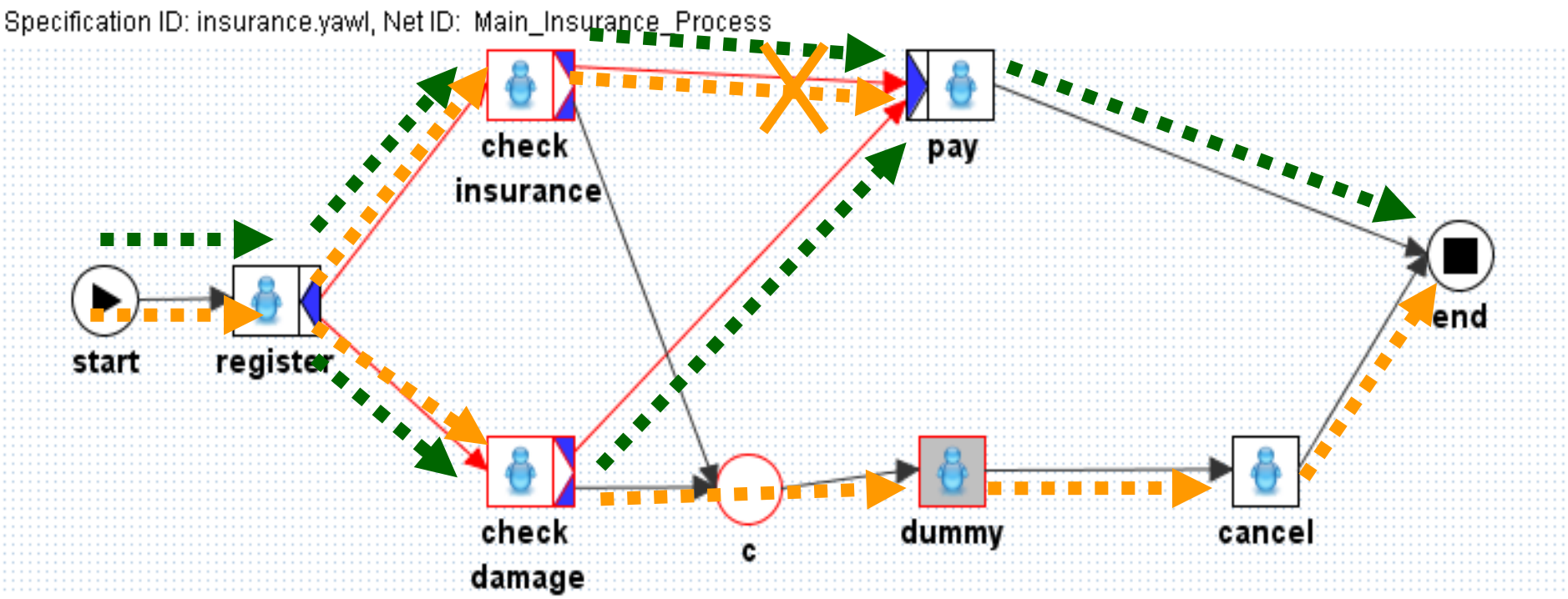
Cancel Save Complete

Done Local intranet 100%

Completed!


Specification ID: insurance.yawl, Net ID: Main_Insurance_Process
33
start
register
check
insurance
check
damage
c
dummy
cancel
pay
32
end

Alternative paths

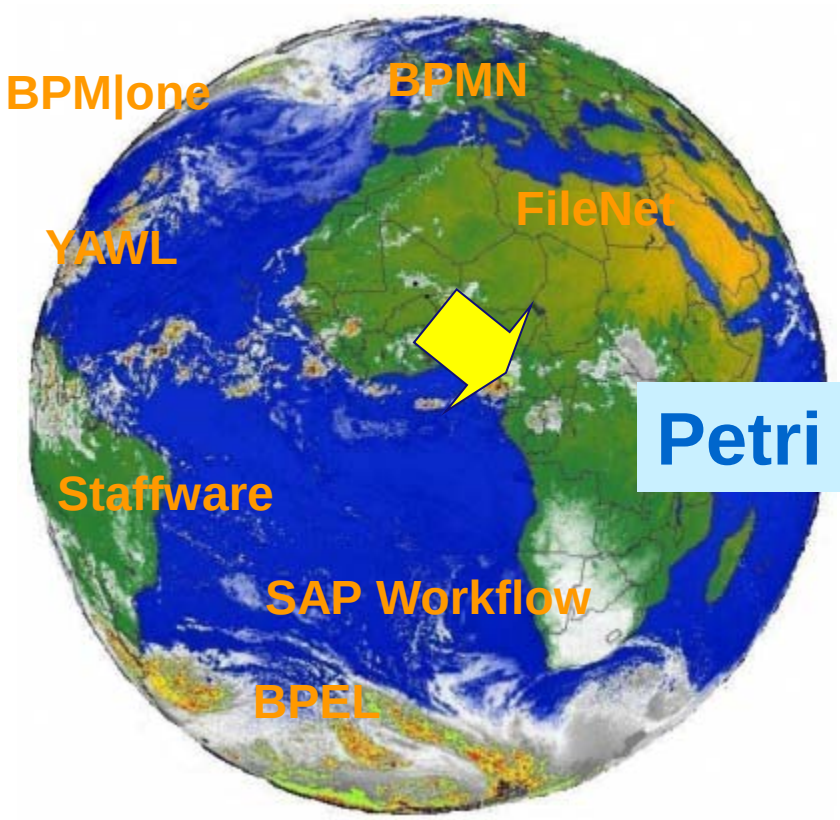


Verification

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Design-time analysis of processes



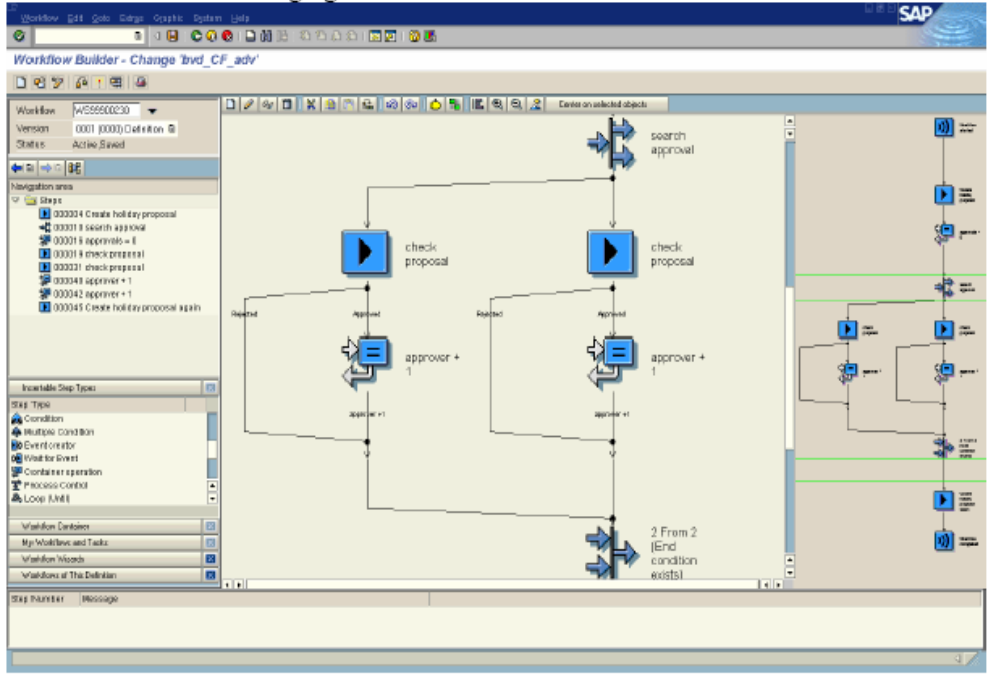
Petri net

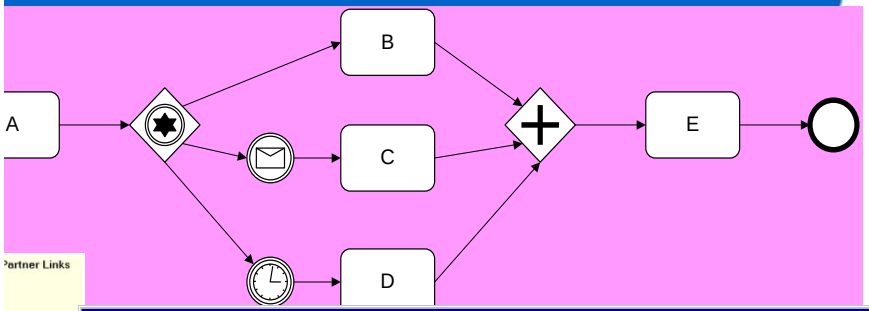
- **linear algebraic analysis techniques**
- **Markov chain analysis techniques**
- **state-space analysis techniques**

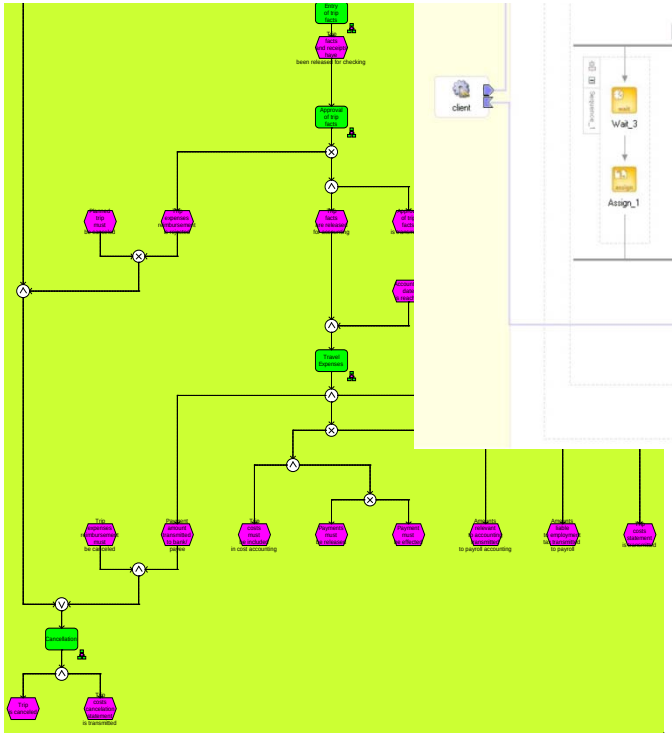
simulation

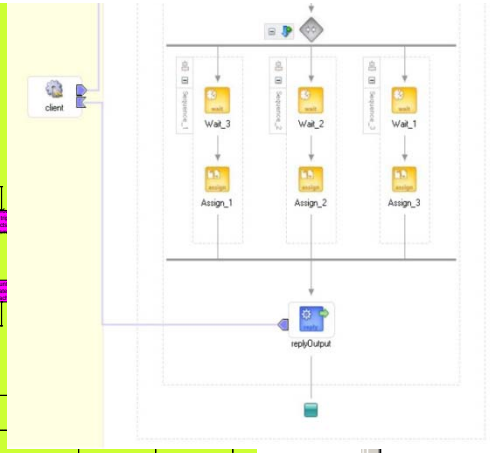
....

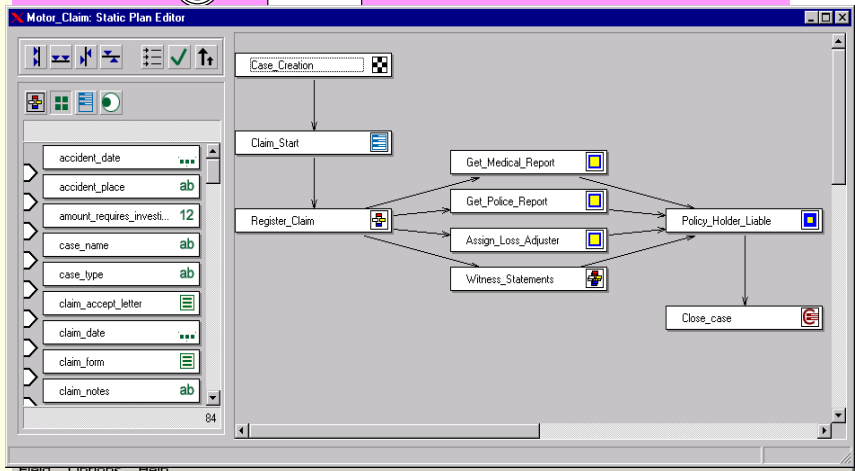
Correct?

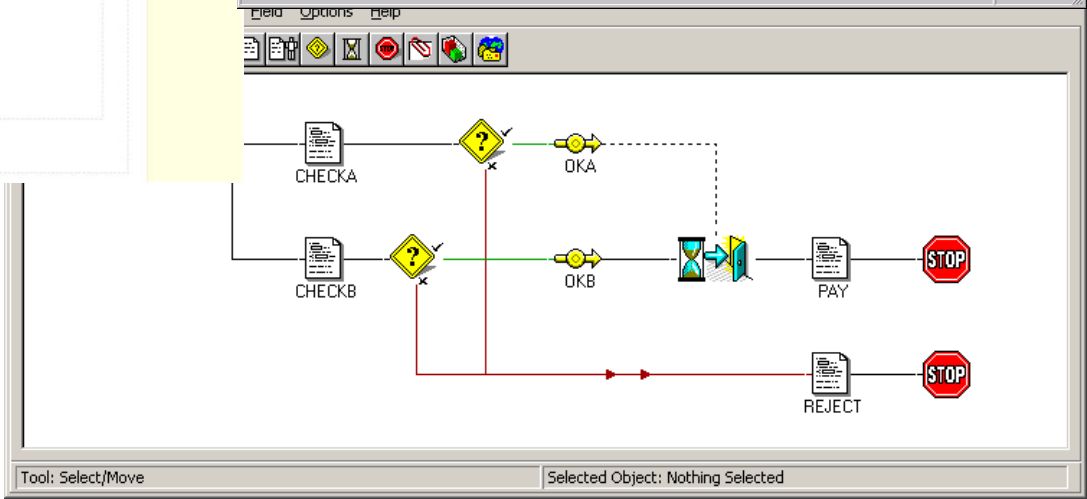

Workflow Builder - Change 'brvd_CF_adv'
search approval
check proposal
check proposal
approver + 1
approver + 1
2 From 2 (End condition exists)


A
B
C
D
E




client
Wait_3
Wait_2
Wait_1
Assign_1
Assign_2
Assign_3
replyOutput

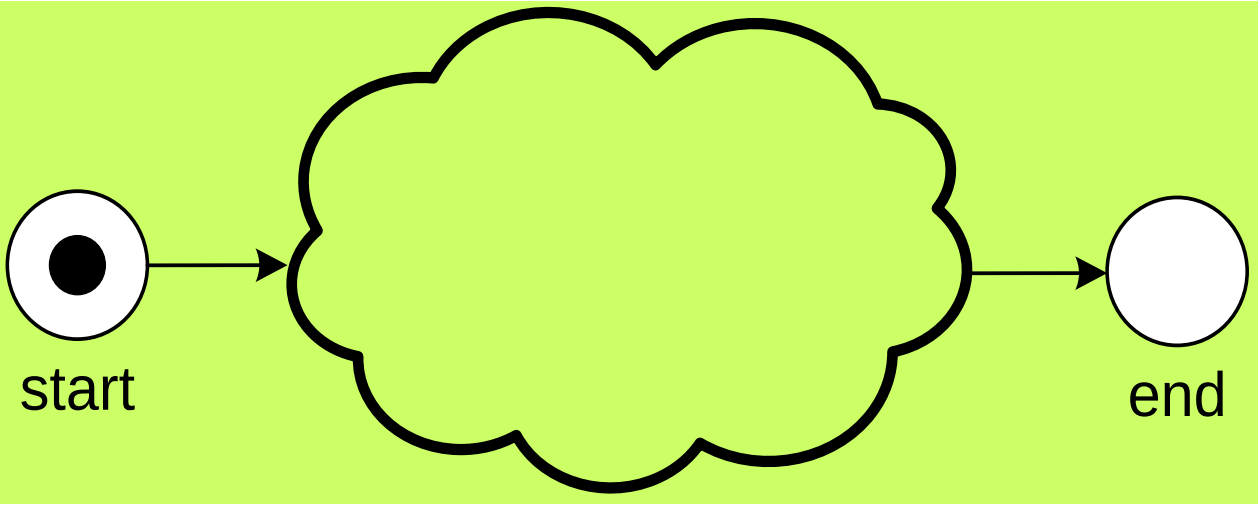

Motor_Claim: Static Plan Editor
Case_Creation
Claim_Start
Get_Medical_Report
Get_Police_Report
Register_Claim
Policy_Holder_Liable
Assign_Loss_Adjuster
Witness_Statements
Close_case
accident_date
accident_place
amount_requires_investi...
case_name
case_type
claim_accept_letter
claim_date
claim_form
claim_notes

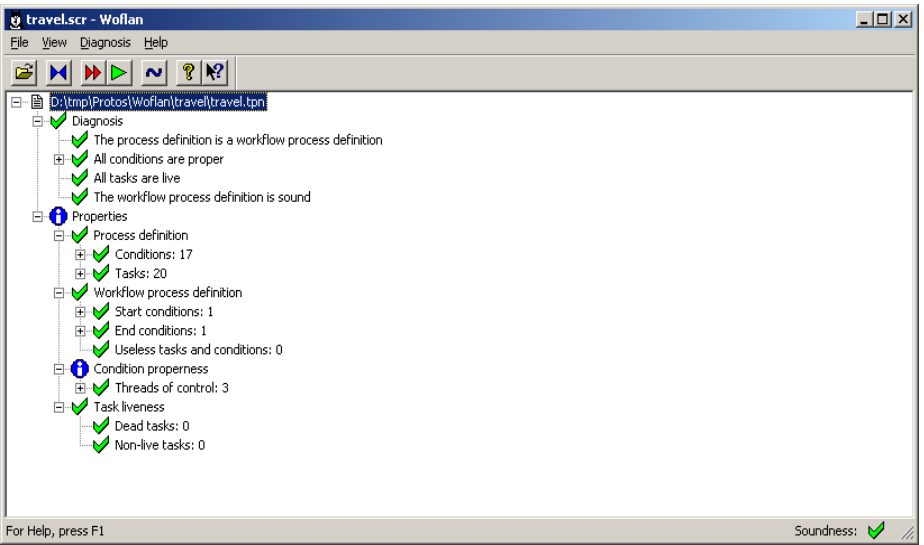

CHECKA
OKA
CHECKB
OKB
PAY
STOP
REJECT
STOP
Tool: Select/Move
Selected Object: Nothing Selected

Questions

- Can the workflow deadlock?
- Is it always possible to terminate?
- Is a livelock possible?
- Are there "dead" activities?
- Is A always followed by B?
- Is it possible to terminate without doing X?
- Can Y be executed more than 4 times?
- Can C and D be executed in parallel?
- ...

Context independent notion: Soundness



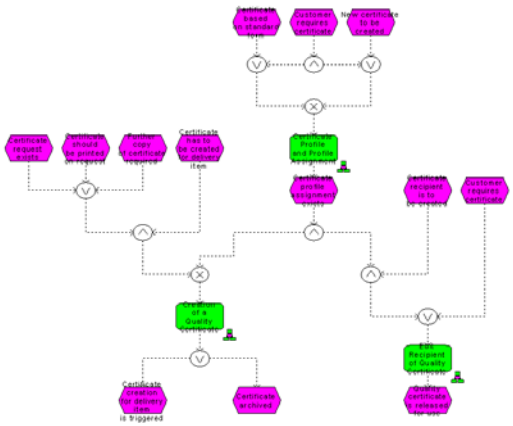


Hot topic!

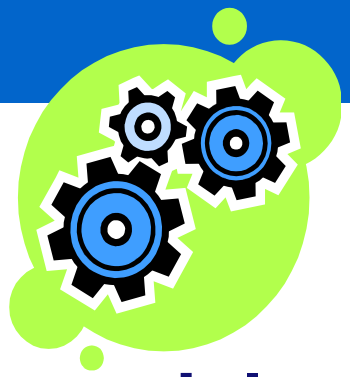
Many approaches/results!

Case study

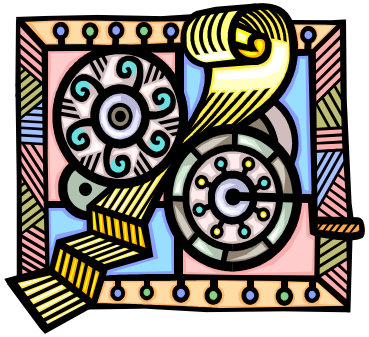
Approach



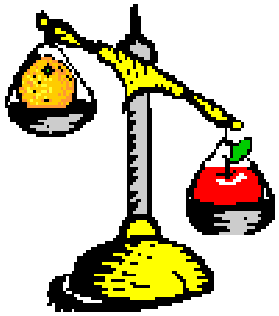
2003 non-trivial process models



model analysis



collect characteristics

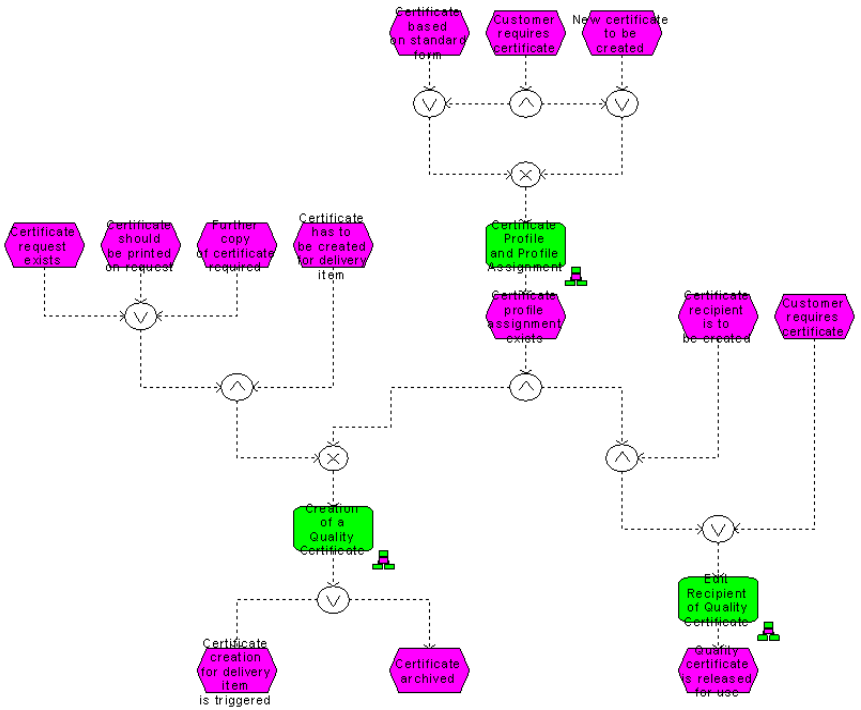


compare

Thanks to Jan Mendling!

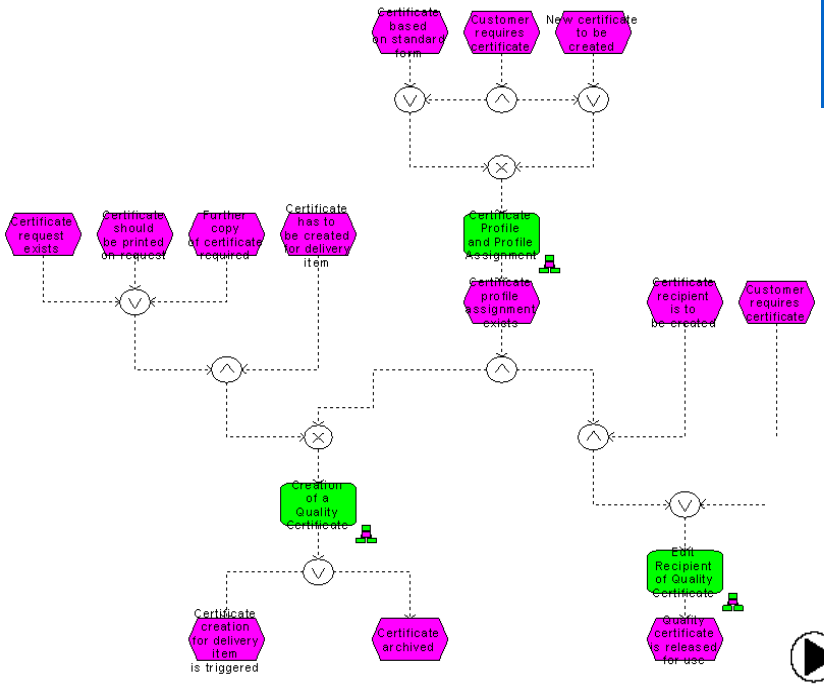
•Jan Mendling, H. M. W. Verbeek, Boudewijn F. van Dongen, Wil M. P. van der Aalst, Gustaf Neumann: Detection and prediction of errors in EPCs of the SAP reference model. Data Knowl. Eng. 64(1): 312-329 (2008).
•Jan Mendling, Michael Moser, Gustaf Neumann, H. M. W. Verbeek, Boudewijn F. van Dongen, Wil M. P. van der Aalst: Faulty EPCs in the SAP Reference Model. Business Process Management 2006: 451-457.

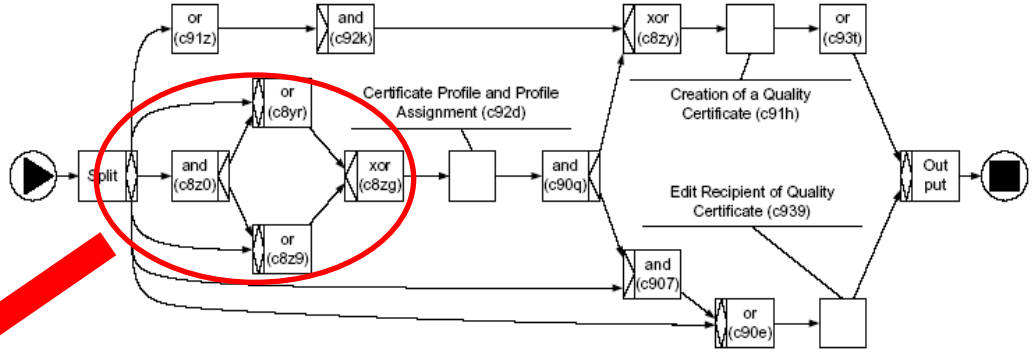
- **The SAP reference model contains more than 600 non-trivial process models expressed in terms of Event-driven Process Chains (EPCs).**

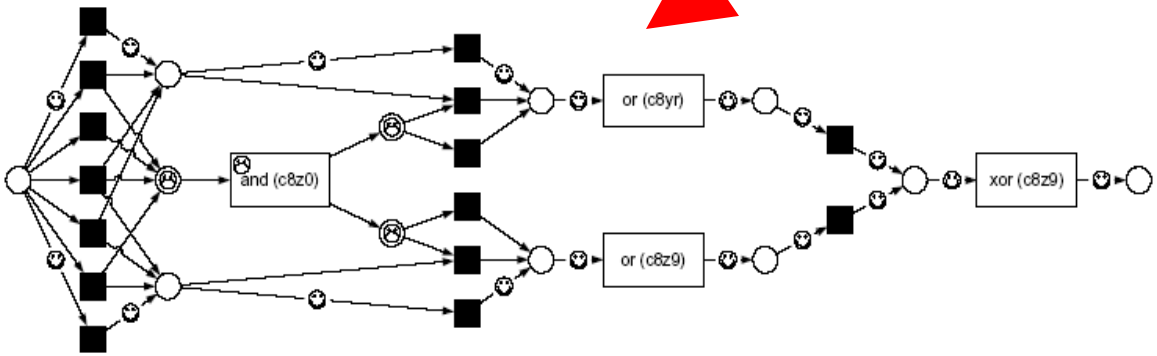


Hierarchy Level	Models	eEPC	Function Allocation Diagram	Process Selection Diagram	Role Activity Diagram	EPC	Error
1	1	1	0	0	0	0	0
2	58	29	0	29	0	0	0
3	175	73	0	0	0	102	15
4	1226	724	0	0	0	502	19
5	8384	3035	3035	0	2014	0	0
All Levels	9844	3862	3035	29	2014	604	34

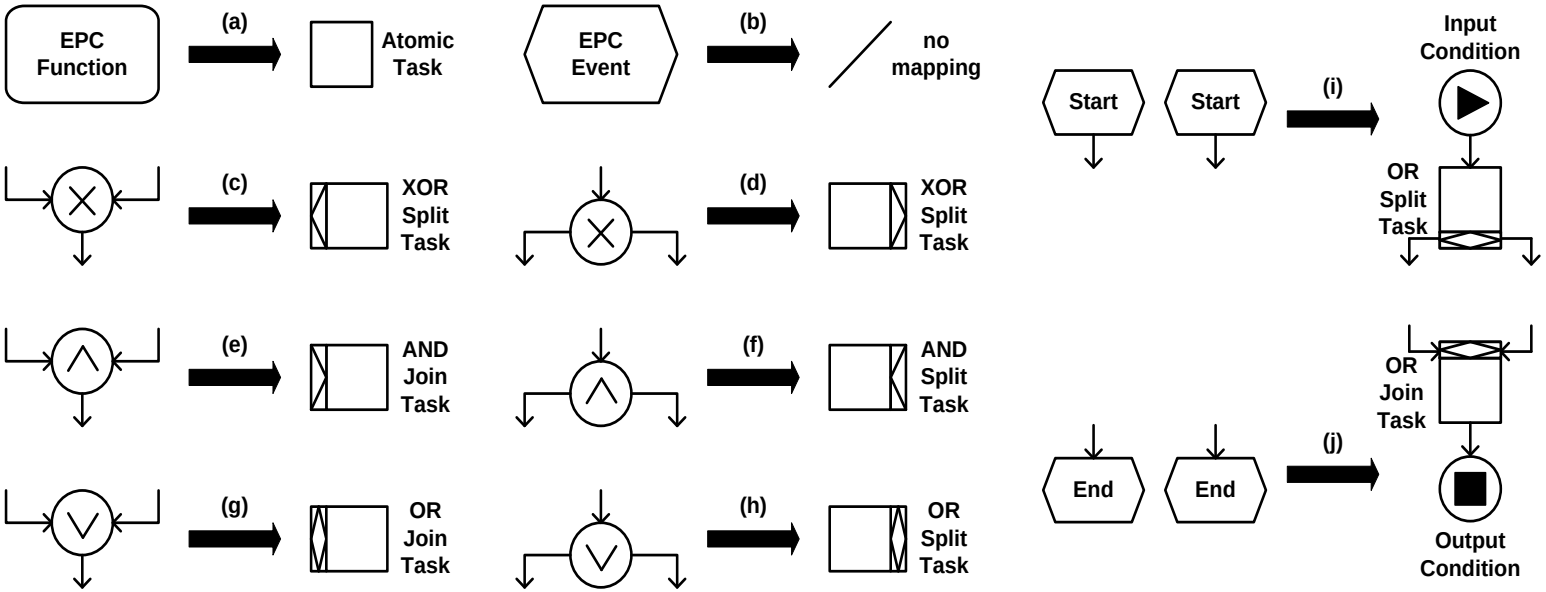
Simplistic approach: YAWL + Petri net invariants







Simplistic approach: YAWL + invariants



Analysis using transition invariants, i.e., only lower bound! ProM allows for more precise analysis

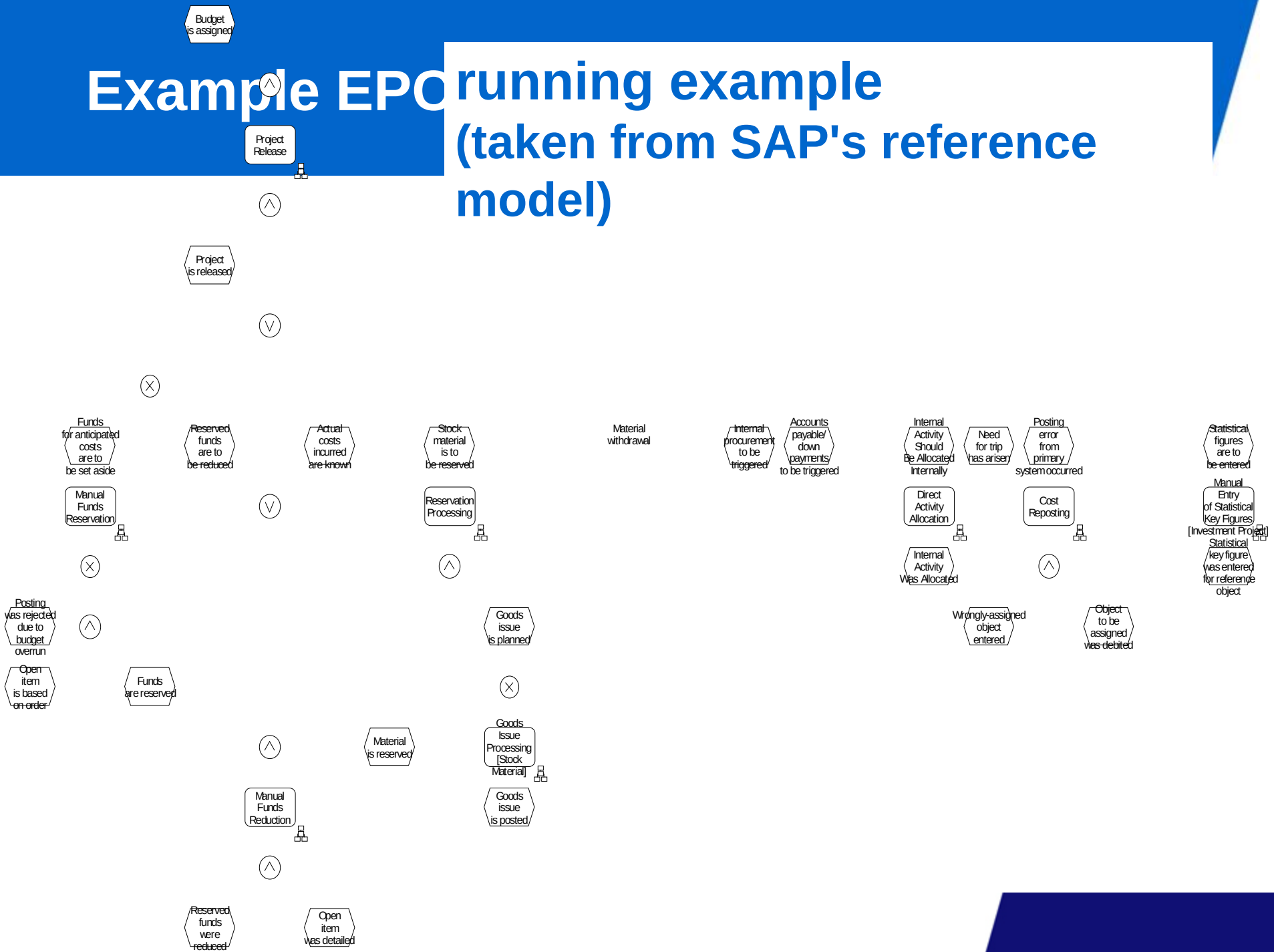
Branch	Model	%	EPC	%	$E_{av.}$	$F_{av.}$	$C_{av.}$	$A_{av.}$	Cycle	Error	%
Asset Accounting	461	4.7%	43	7.1%	13.9	4.0	5.2	23.3	0	7	16.3%
Benefits Administration	50	0.5%	6	1.0%	9.5	3.3	5.8	19.7	3	0	0.0%
Compensation Management	122	1.2%	18	3.0%	7.6	3.4	3.3	13.7	3	1	5.6%
Customer Service	402	4.1%	41	6.8%	16.5	3.6	9.0	29.5	3	1	2.4%
Enterprise Co						0.1	6.1	32.1	0	3	13.6%
Environment,						2.7	1.2	7.0	0	0	0.0%
Financial Acc						4.0	5.1	21.8	0	3	5.6%
Position Mana						0.0	0.0	0.0	0	0	n.a.
Inventory Management	184	1.9%	3	0.5%	15.0	7.0	6.0	28.0	2	0	0.0%
Organizational Management	37	0.4%	5	0.8%	12.0	3.0	6.6	24.0	3	0	0.0%
Payroll	541	5.5%	7	1.2%	5.7	3.1	2.1	11.4	0	1	14.3%
Personnel Administration	15	0.2%	4	0.7%	7.3	1.5	4.0	12.3	0	0	0.0%
Personnel Development	60	0.6%	10	1.7%	8.7	2.5	4.4	15.6	3	1	10.0%
Personnel Time Management	87	0.9%	1				3	19.5	1	2	16.7%
Plant Maintenance	399	4.1%	3				4	37.8	9	1	2.9%
Procurement	444	4.5%	3				7	12.4	0	2	5.4%
Product Data Management	366	3.7%	20				2.2	13.7	0	0	0.0%
Production	296	3.0%	17	2.8%	8.8	3.0	2.9	13.7	0	1	5.9%
Production Planning	194	2.0%	17	2.8%	5.7	2.9	3.0	11.5	0	0	0.0%
Project Management	347	3.5%	36	6.0%	8.5	3.8	2.2	14.0	0	0	0.0%
Quality Management	209	2.1%	20	3.3%	20.5	3.8	11.7	37.8	1	1	5.0%
Real Estate Management	169	1.7%	6	1.0%	12.7	6.5	7.3	27.0	1	1	16.7%
Recruitment	56	0.6%	9	1.5%	7.4	2.6	4.1	13.8	3	0	0.0%
Retail	842	8.6%	1	0.2%	7.0	5.0	2.0	11.0	0	0	0.0%
Revenue & Cost Controlling	568	5.8%	19	3.1%	16.5	10.2	7.9	36.0	1	1	5.3%
Sales & Distribution	703	7.1%	76	12.6%	10.6	3.1	4.3	16.6	0	1	1.3%
Training & Event Management	95	1.0%	12	2.0%	13.0	2.7	6.2	22.2	0	1	8.3%
Travel Management	116	1.2%	1	0.2%	24.0	7.0	16.0	48.0	0	0	0.0%
Treasury	1761	17.9%	48	7.9%	10.5	3.5	4.5	18.1	0	6	12.5%
All 29 Branches	9844	100%	604	100%	11.5	4.0	5.2	20.8	33	34	5.6%

Lowerbound!

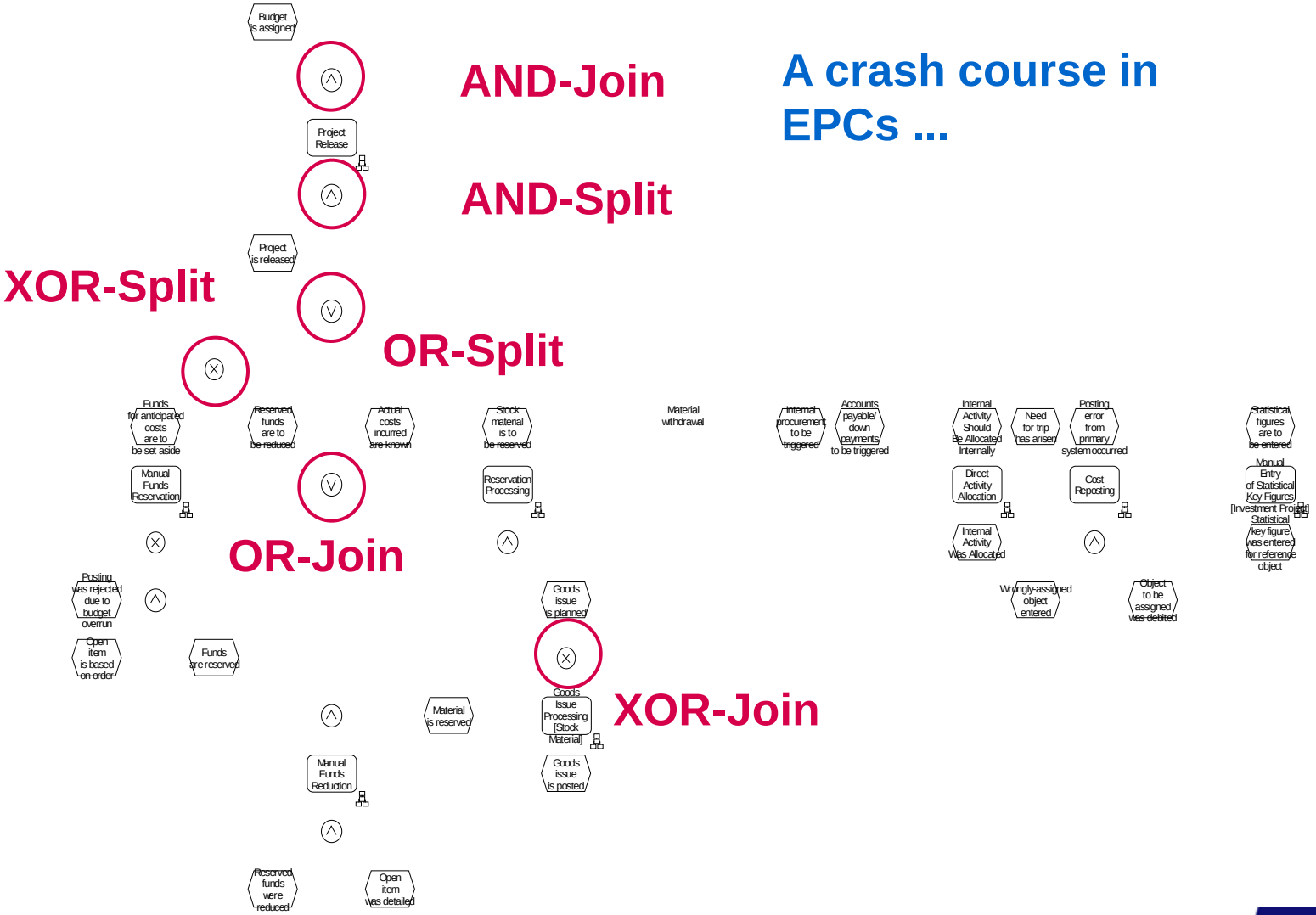
Why?

5.6%

Example EPC running example (taken from SAP's reference model)

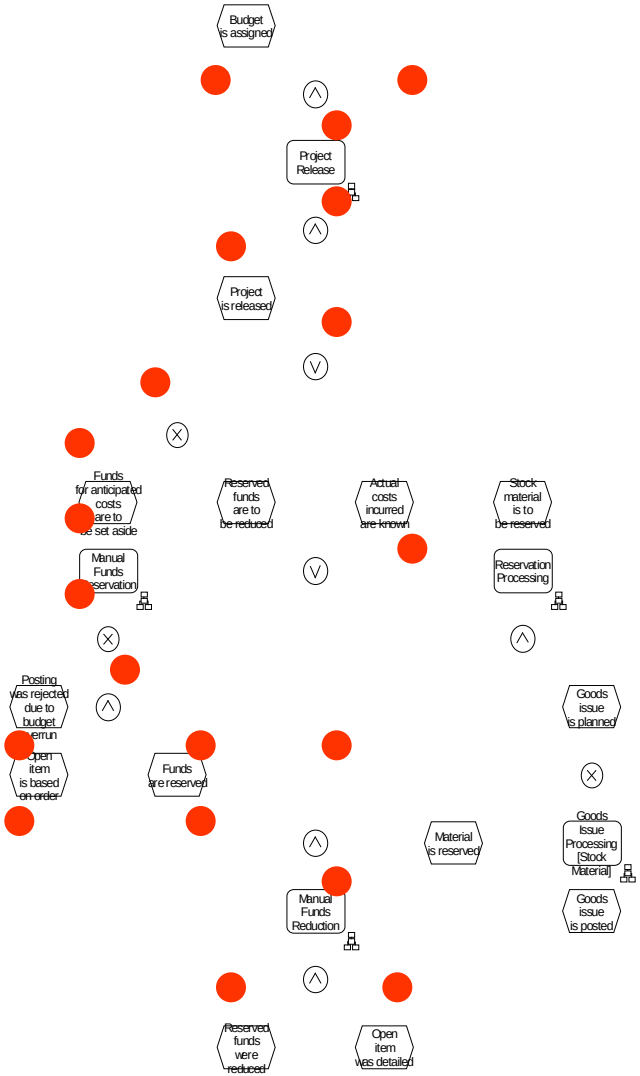


EPC Connectors

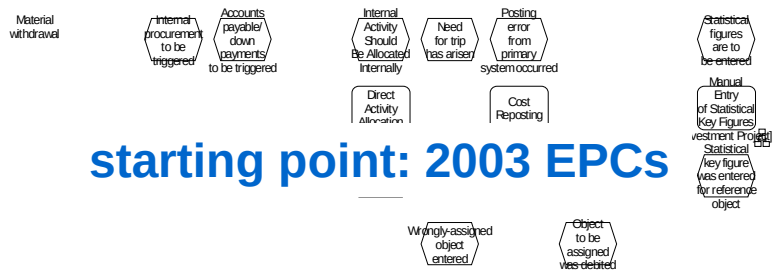


A crash course in EPCs ...

EPC Behavior



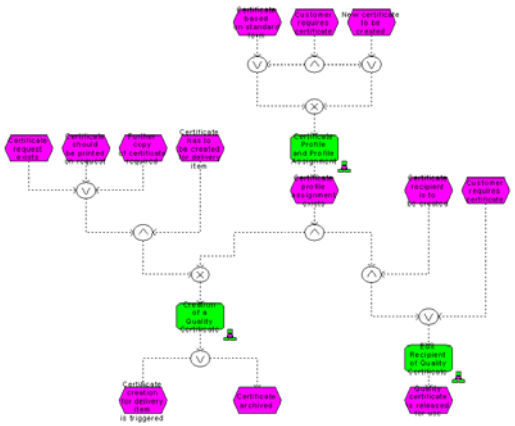
example path



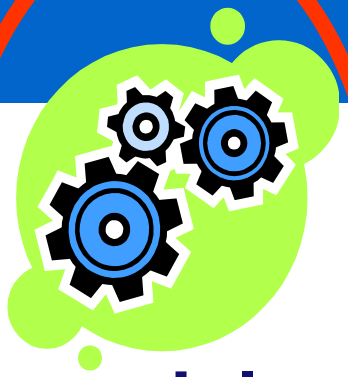
starting point: 2003 EPCs

1. *SAP Reference Model:* The first collection of EPCs is the SAP Reference Model. The development of the SAP reference model started in 1992 and first models were presented at CEBIT'93 [19, p.VII]. We use the SAP reference model in its version from 2000 that includes 604 non-trivial EPCs.
2. *Service Model:* The second collection of EPCs stems from a German process reengineering project in the service sector. The project was carried out in the late 1990s. The models of this project include 381 non-trivial EPCs.
3. *Finance Model:* The third model collection contains EPCs of a process documentation project in the Austrian financial industry. It includes 935 EPCs.
4. *Consulting Model:* The fourth collection covers in total 83 EPCs from three different consulting companies.

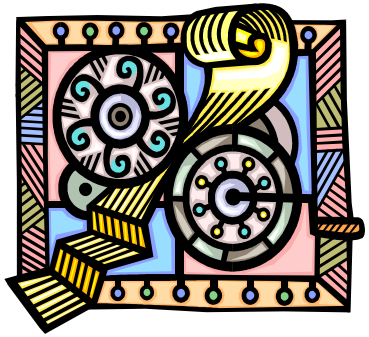
•Jan Mendling, Gustaf Neumann, Wil M. P. van der Aalst: Understanding the Occurrence of Errors in Process Models Based on Metrics. OTM Conferences (1) 2007: 113-130.
•J. Mendling: Detection and Prediction of Errors in EPC Business Process Models. Doctoral Thesis, Vienna University of Economics and Business Administration. Vienna, Austria, May 2007.



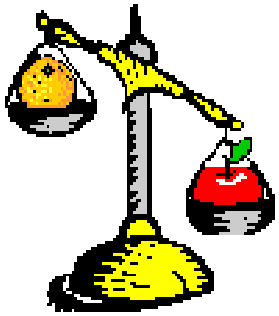
2003 non-trivial process models



model analysis



collect characteristics

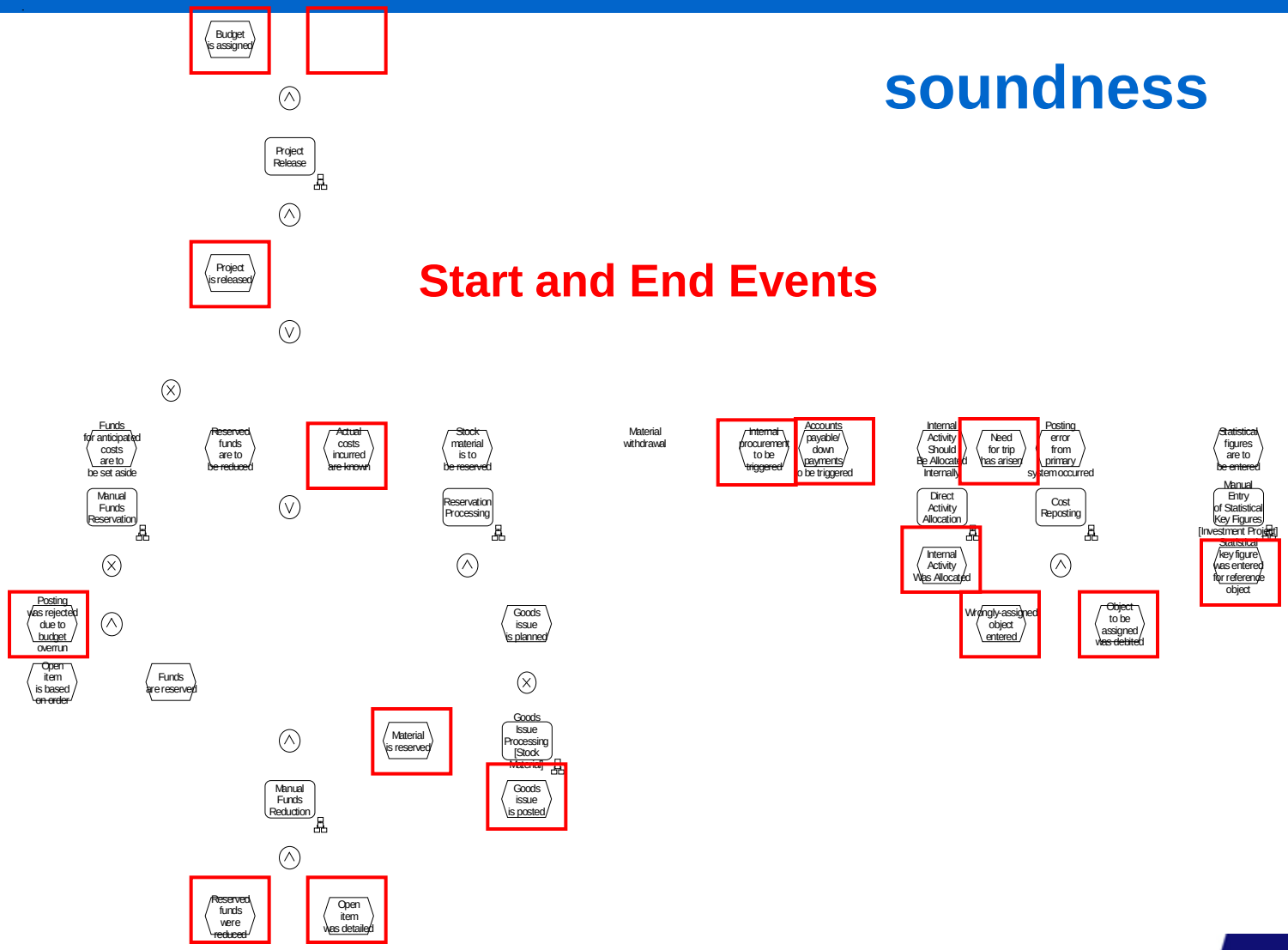


compare

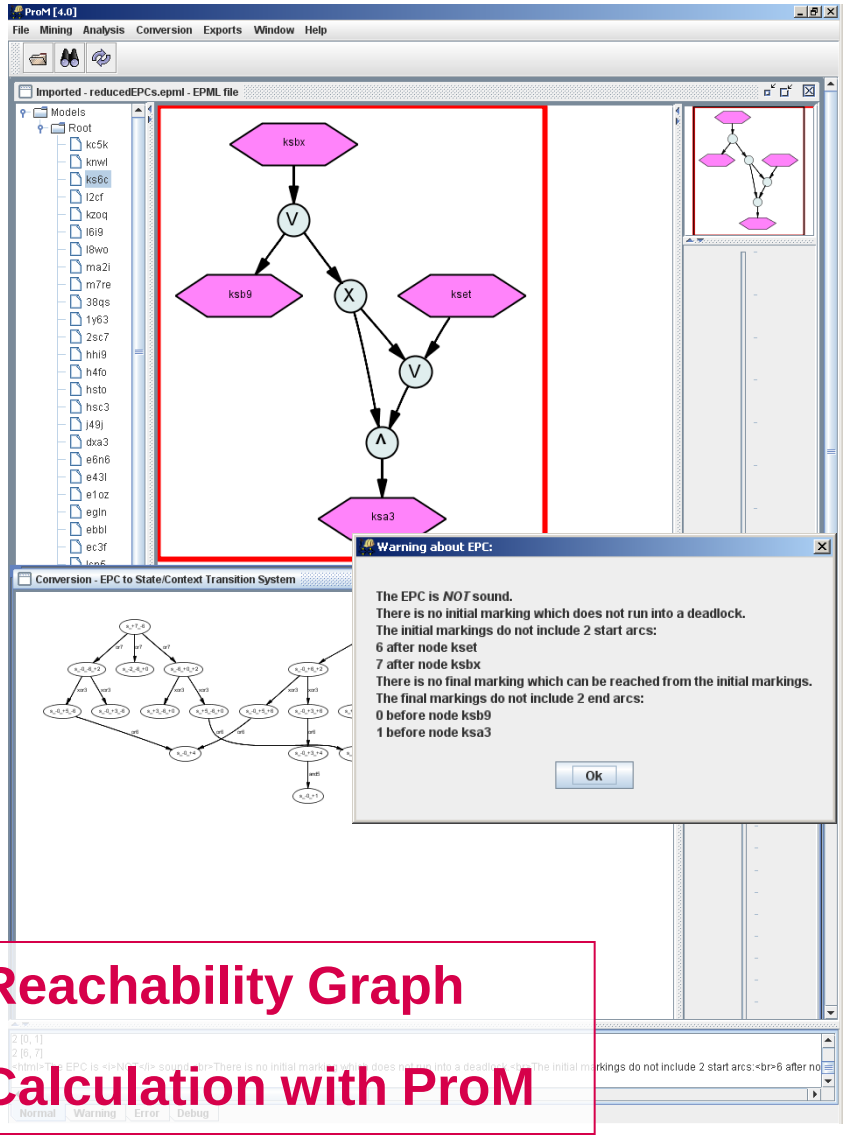
EPC Soundness

soundness

Start and End Events



analyzing soundness

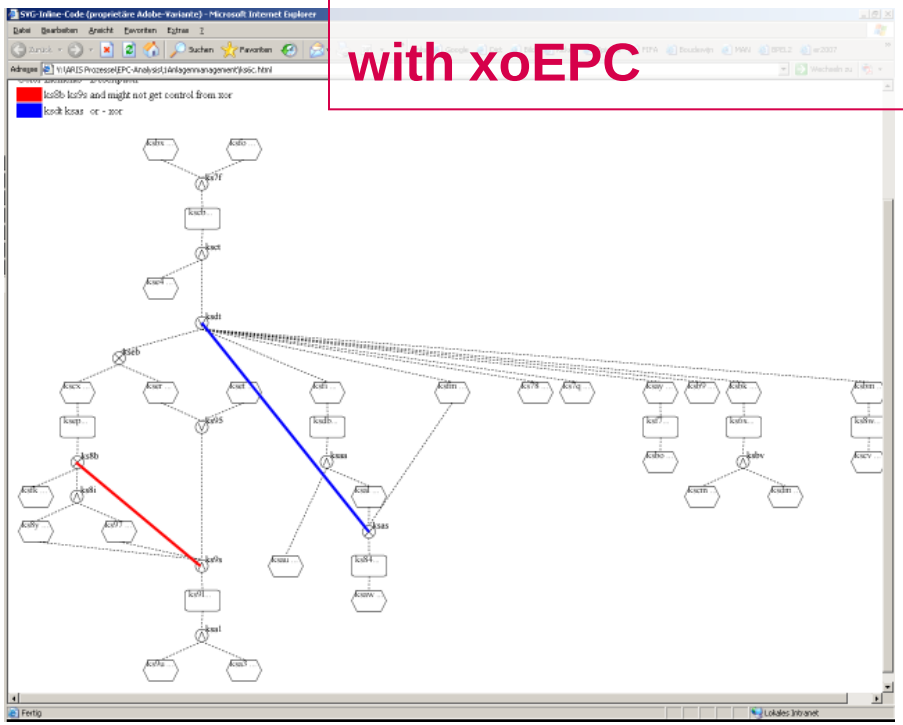


Reachability Graph

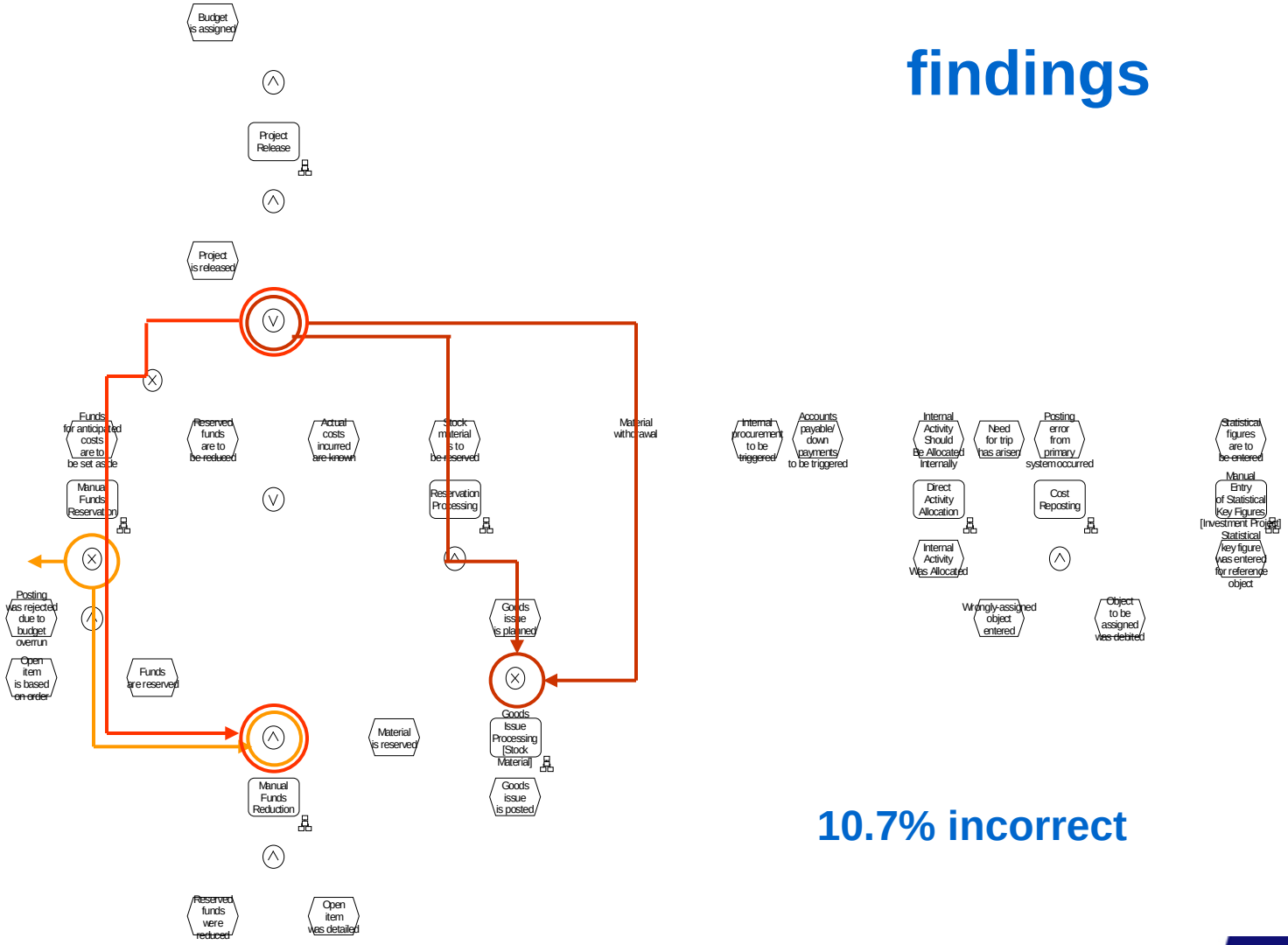
Calculation with ProM

Reduction Rules

with xoEPC



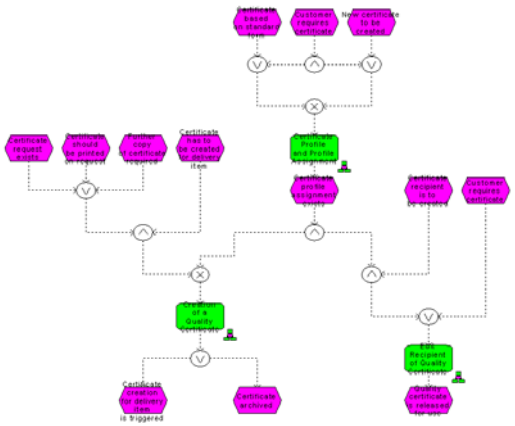
findings



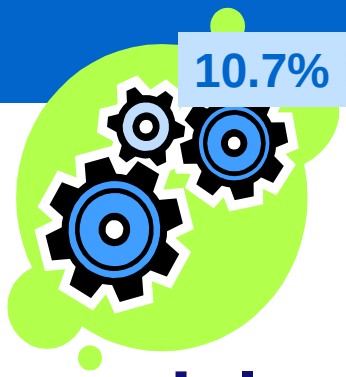
10.7% incorrect

Parameter Parameter	Complete Sample	SAP Ref. Model	Services Model	Finance Model	Consulting Models
xoEPC errors	154	90	28	26	10
Unreduced EPCs	156	103	18	17	18
ProM error EPCs	115	75	16	7	17
EPCs with errors	215	126	37	31	21
EPCs in total	2003	604	381	935	83
Error ratio	10.7%	20.9%	9.7%	3.3%	25.3%

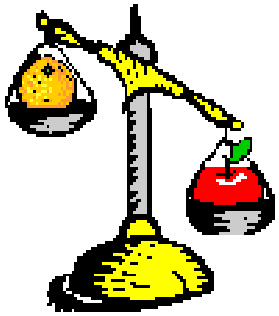
10.7% incorrect



2003 non-trivial process models



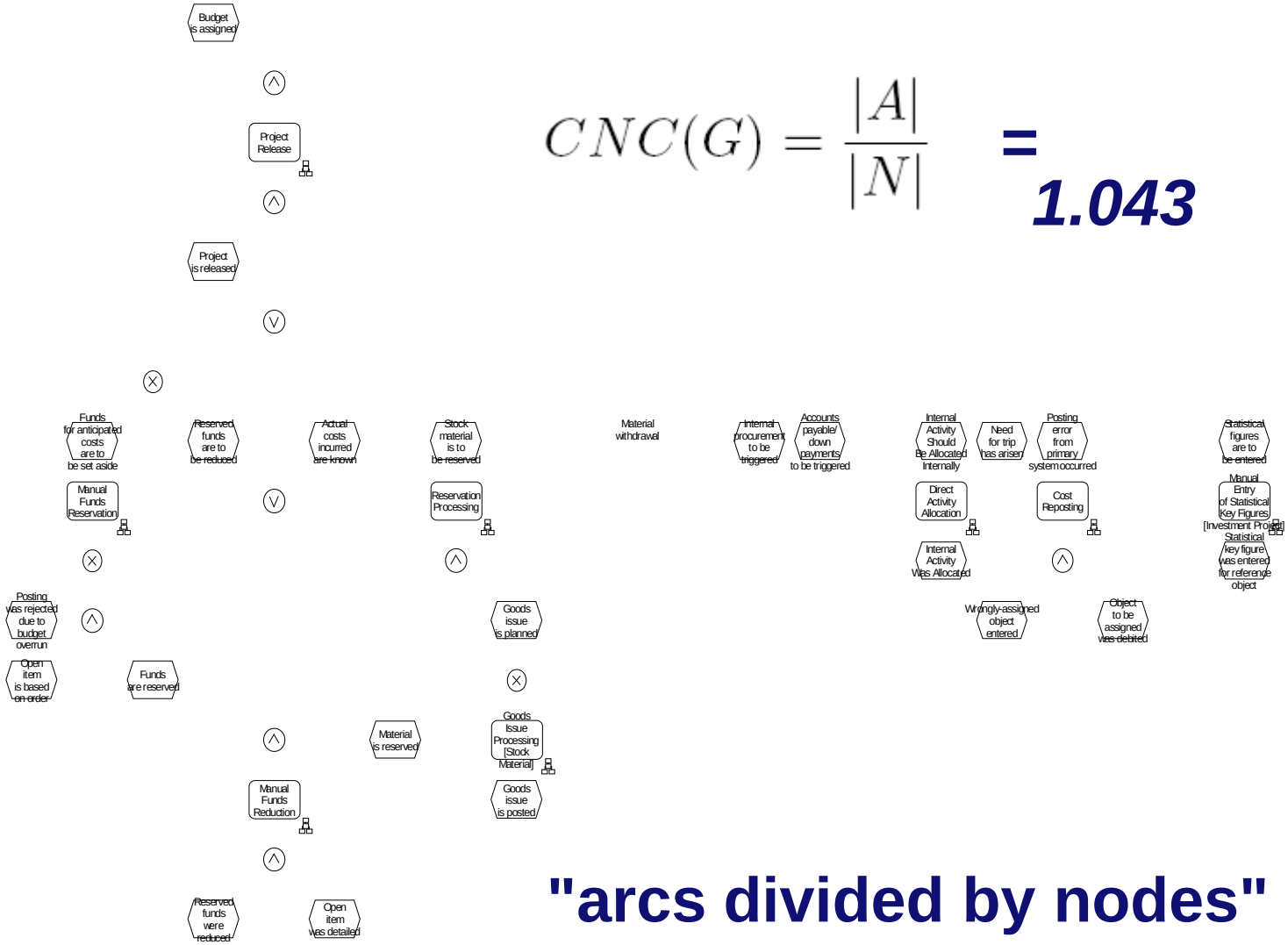
model analysis



collect characteristics

compare

Model Metrics:
Coefficient of Network Connectivity (CNC)

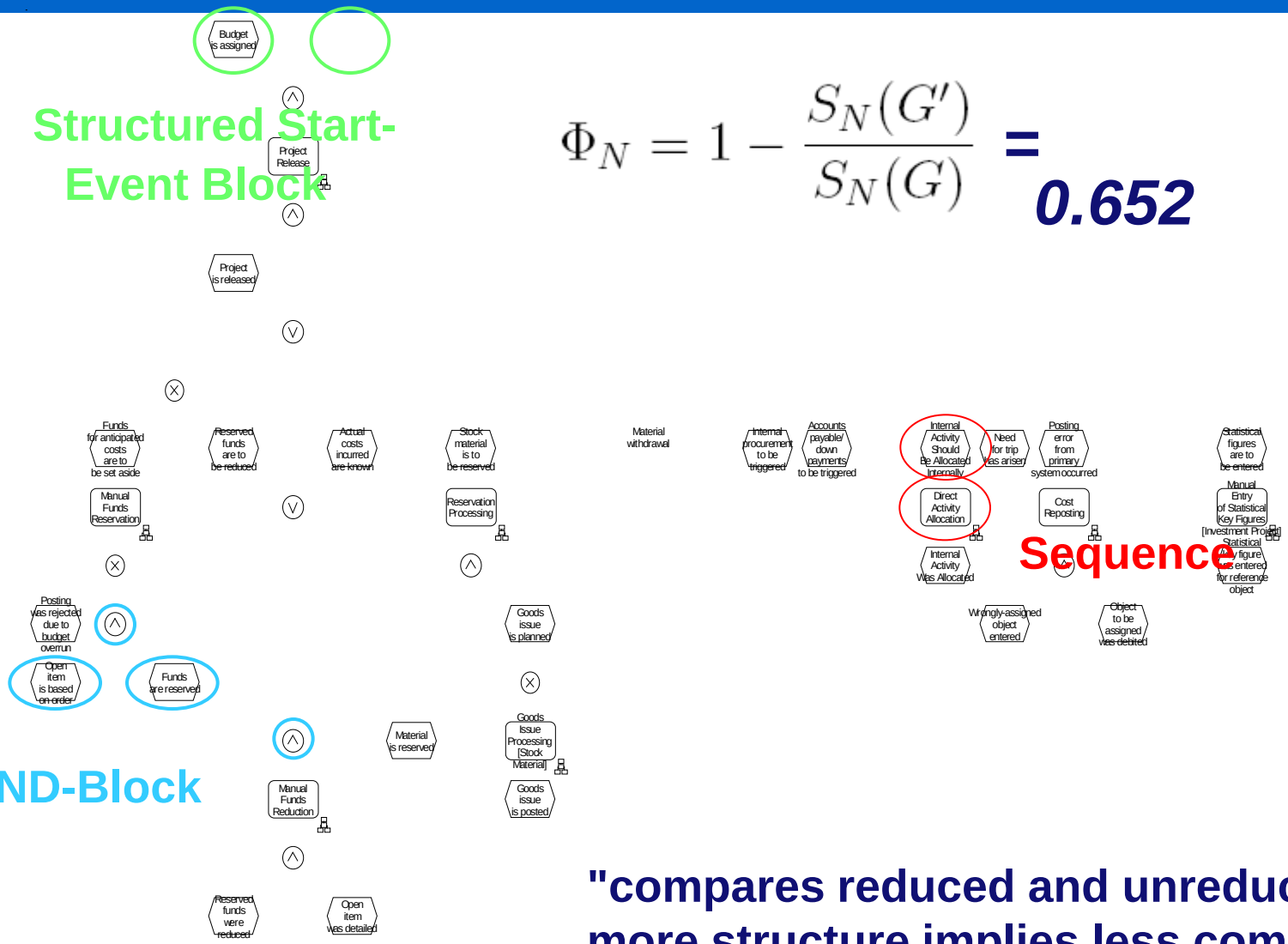


$$CNC(G) = \frac{|A|}{|N|} = 1.043$$

"arcs divided by nodes"

more arcs implies more complexity

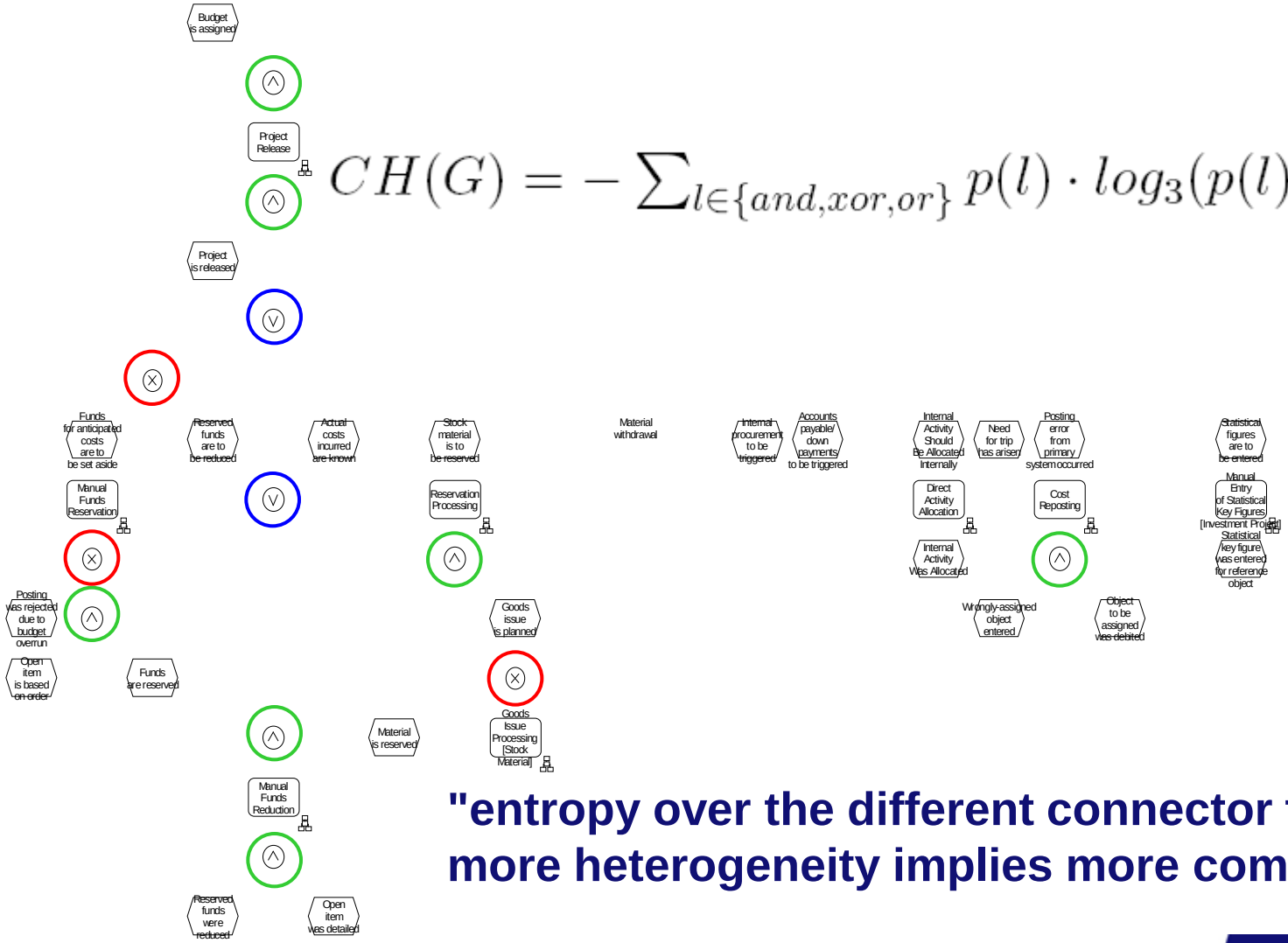
Model Metrics: Structuredness (Phi)



$$\Phi_N = 1 - \frac{S_N(G')}{S_N(G)} = 0.652$$

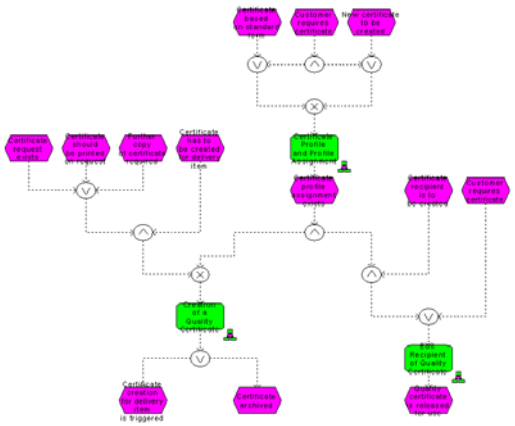
"compares reduced and unreduced graph"
more structure implies less complexity

Model Metrics: Connector Heterogeneity (CH)

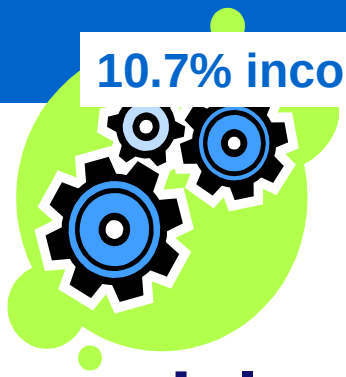


$$CH(G) = -\sum_{l \in \{and, xor, or\}} p(l) \cdot log_3(p(l)) = 0.873$$

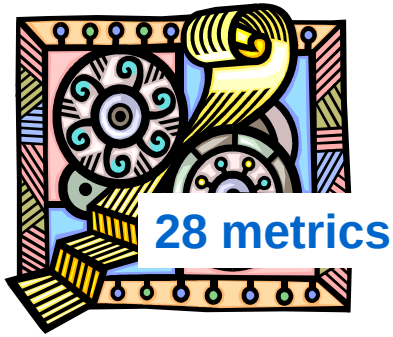
"entropy over the different connector types"
more heterogeneity implies more complexity



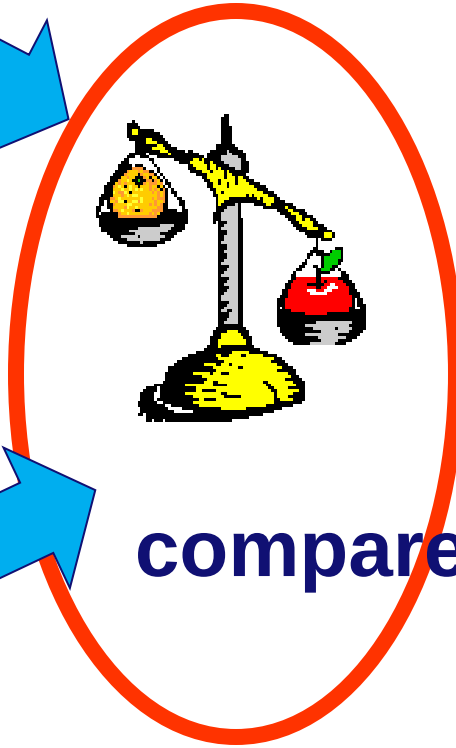
2003 non-trivial process models



model analysis



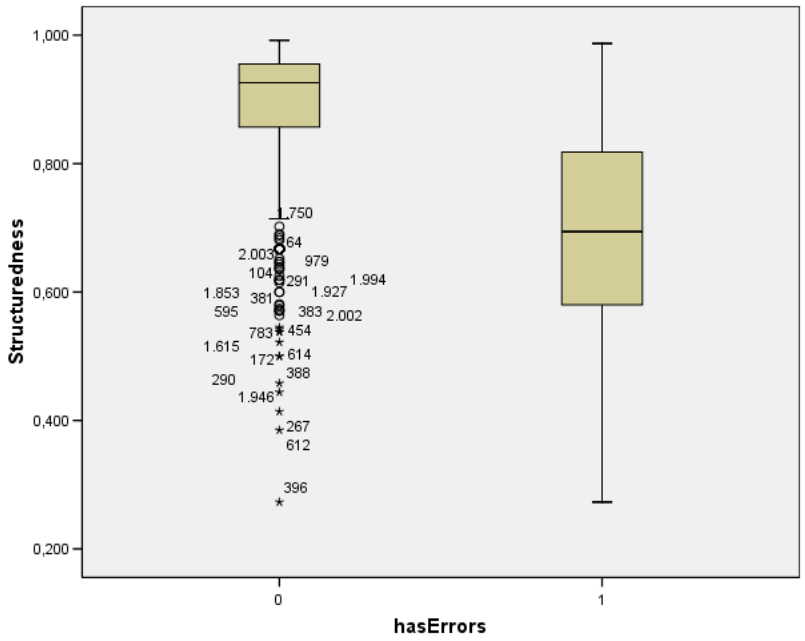
collect characteristics

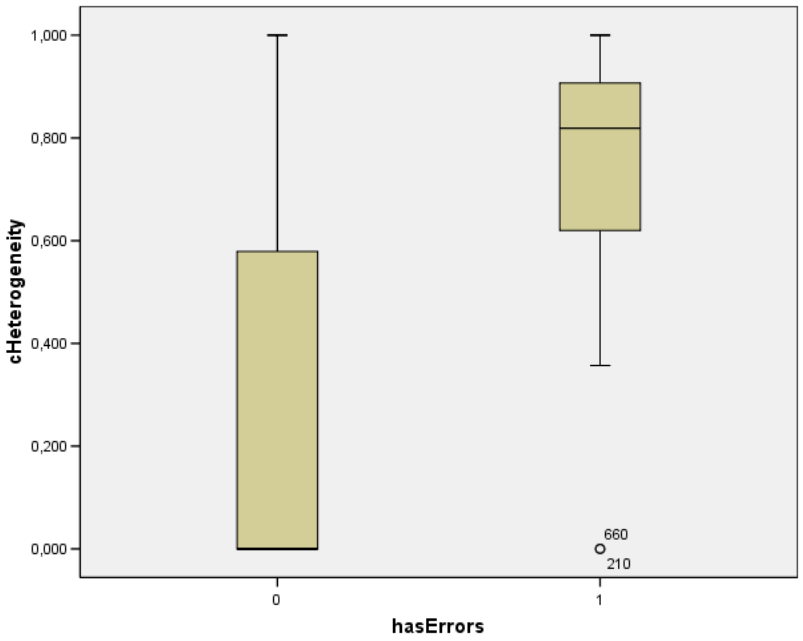


compare

Correlation of Metrics and Errors

Sample of 2003 EPCs from practice





$r_{Phi,hasError} = -0,36$

$r_{CH,hasError} = 0,46$

Logistic Regression Model

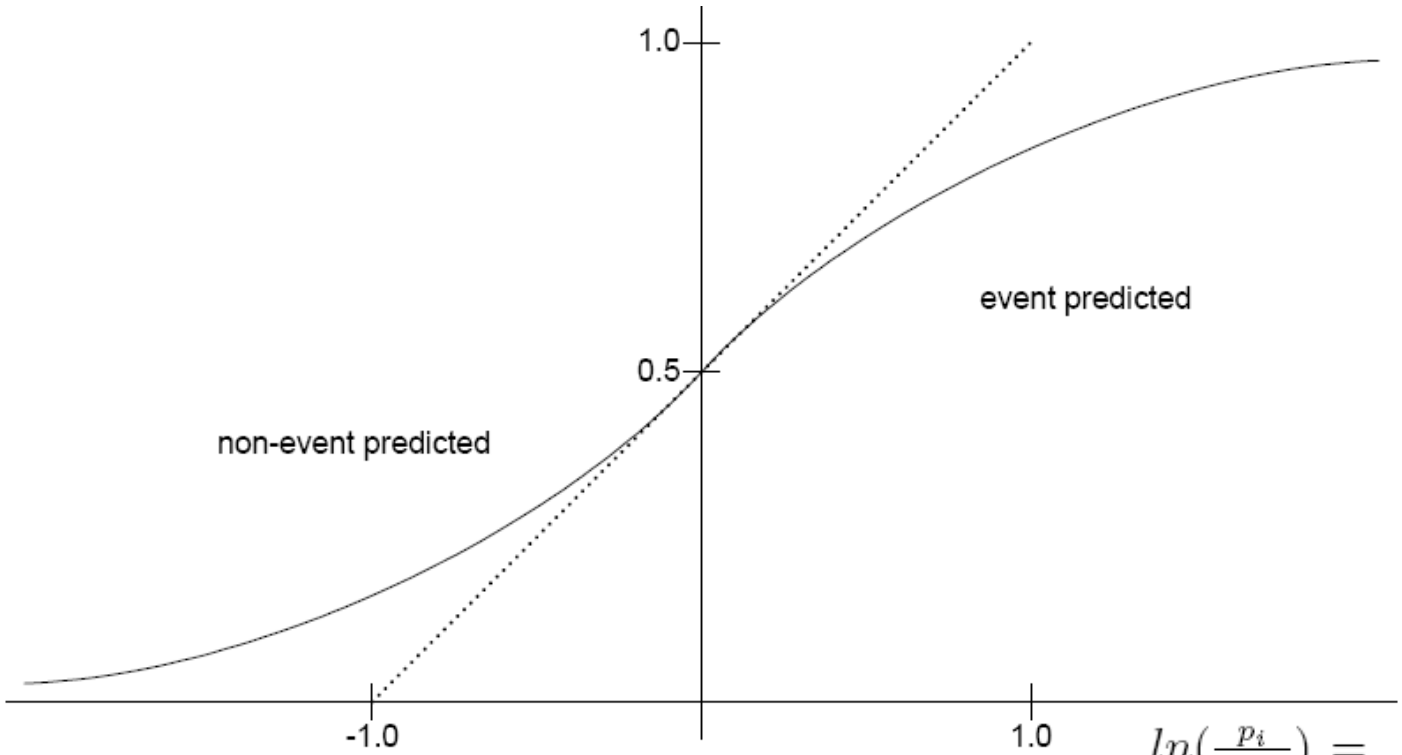
$$p(EPC) = \frac{e^{logit(EPC)}}{1 + e^{logit(EPC)}}$$

$$\begin{aligned} logit(EPC) = \quad & +4.008\, CNC \\ & +0.094\, MM \\ & +3.409\, CYC \\ & -2.338\, \Pi \\ & -9.957\, \Phi \\ & +3.003\, CH \\ & +0.064\, diam \end{aligned}$$

Nagelkerke R^2 has an excellent value of 0.901

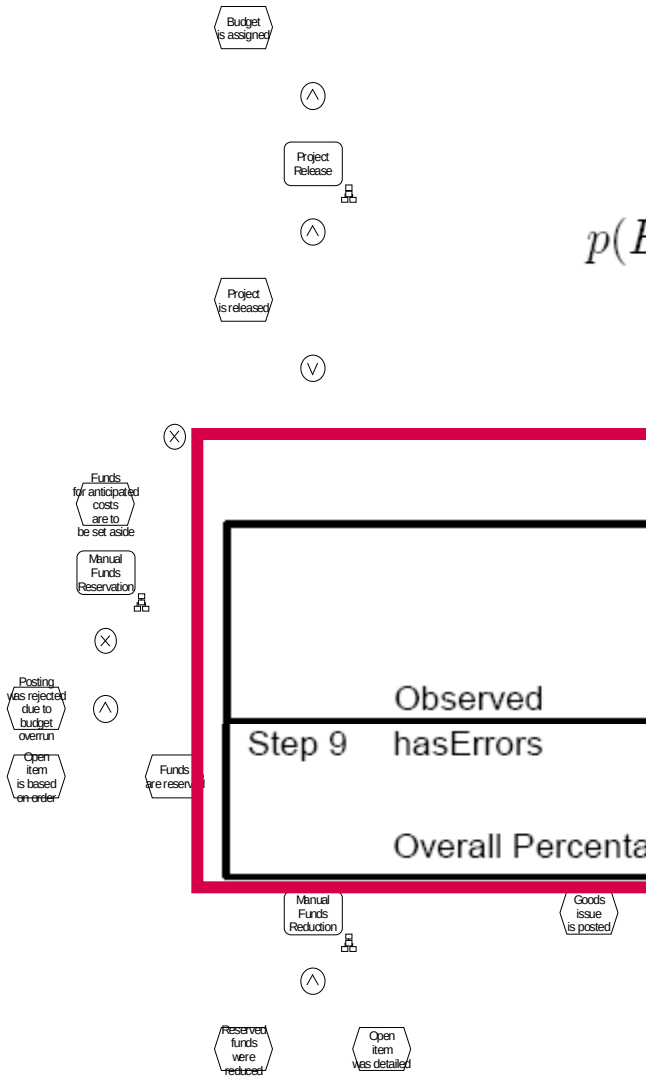
Logistic Regression Results

$$p_i = \frac{e^{\beta_0+\beta_1 x_{1,i}+\cdots+\beta_k x_{k,i}}}{1+e^{\beta_0+\beta_1 x_{1,i}+\cdots+\beta_k x_{k,i}}}$$



$$ln(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}$$

Error Prediction



$$p(EPC) = \frac{e^{logit(EPC)}}{1 + e^{logit(EPC)}}$$ **= 0.811 > 0.5**

Classification Table[a]

Observed			Predicted		
			hasErrors		Percentage Correct
			0	1	
Step 9	hasErrors	0	1724	37	97,9
		1	58	155	72,8
	Overall Percentage				95,2

Holdout Sample Results

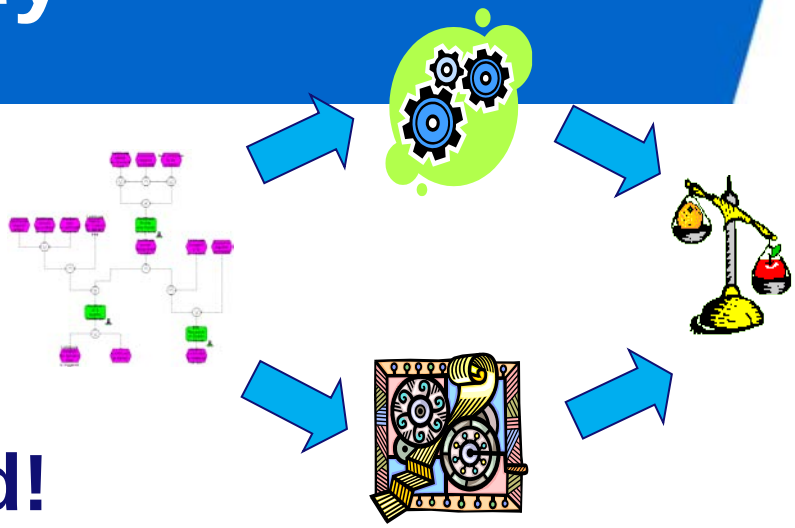
- 113 EPCs from books by
  - Scheer,
  - Becker & Schütte,
  - Staud, and
  - Seidlmeier

Classification Table

Observed		Predicted hasErrors 0	Predicted hasErrors 1	Percentage Correct
hasErrors	0	86	2	97,73%
	1	9	16	64,00%
Overall Percentage				90,27%

- Accuracy interval for prediction function
  - 81.15% to 96.77%
  - with 95% confidence

Conclusion: case study



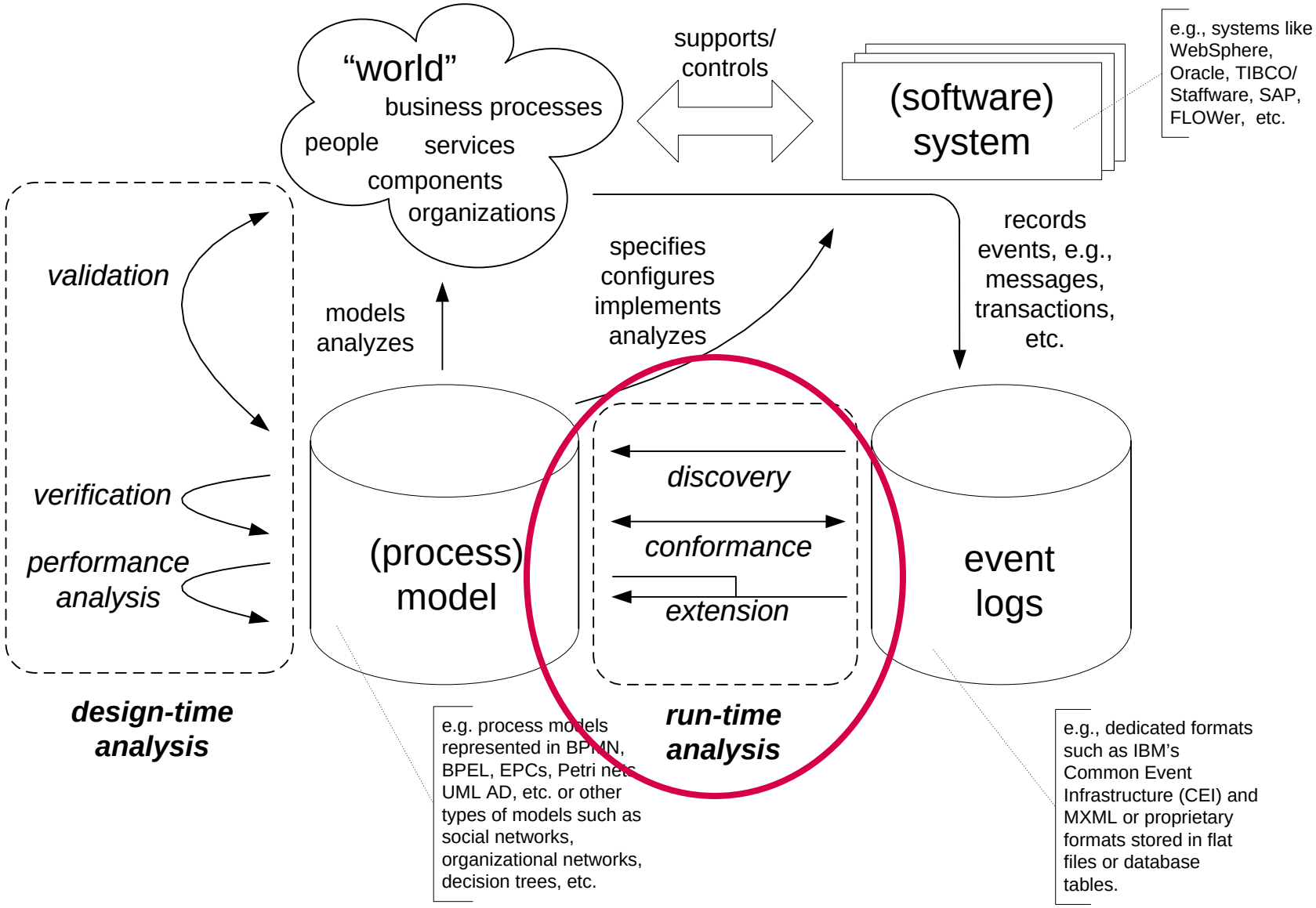
- Verification is mature!
- Models are not!
- Errors can be predicted!
- How to improve?
  - Take models seriously, otherwise do not bother.
  - Improve teaching.
  - Build smart editors, e.g., based on patterns and anti-patterns

Process Mining

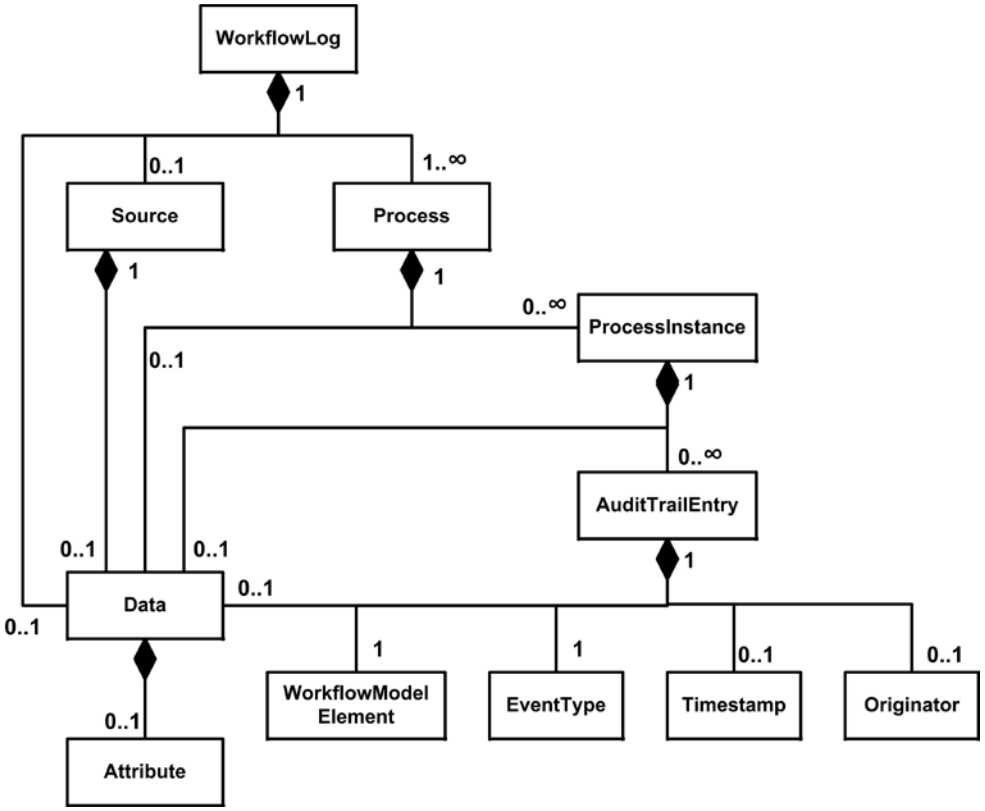
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Design-time analysis vs run-time analysis



Starting point: event logs



event logs, audit trails, databases, message logs, etc

unified event log (MXML)

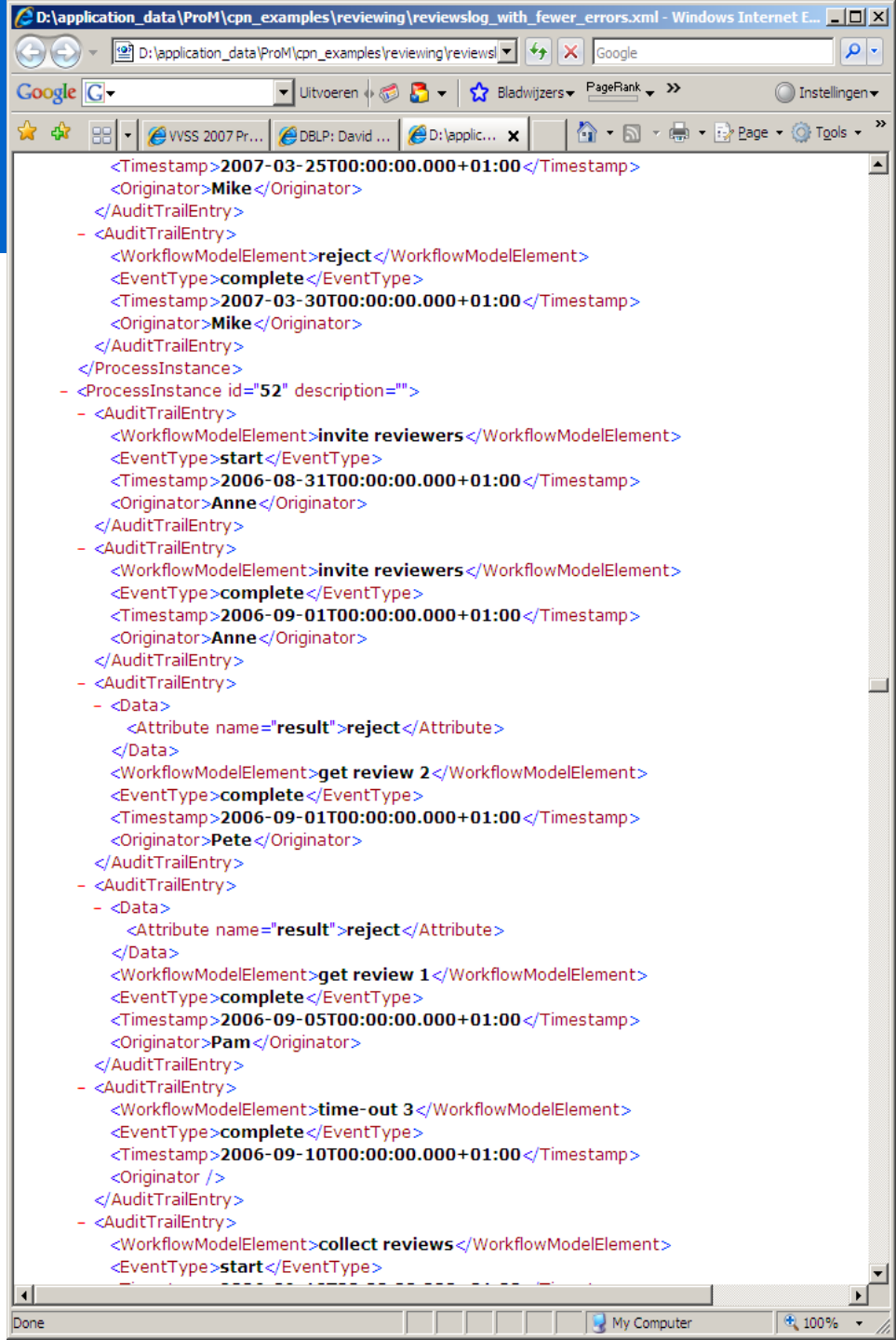
MXML

Event log:

- processes
  - process instances
    - events

Per event:

- activity name
- (event type)
- (originator)
- (timestamp)
- (data)



```
</ProcessInstance>
- <ProcessInstance id="51" description="">
  - <AuditTrailEntry>
      <WorkflowModelElement>invite reviewers</WorkflowModelElem
      <EventType>start</EventType>
      <Timestamp>2006-08-28T00:00:00.000+01:00</Timestamp>
      <Originator>Mike</Originator>
    </AuditTrailEntry>
  - <AuditTrailEntry>
      <WorkflowModelElement>invite reviewers</WorkflowModelEle
      <EventType>complete</EventType>
      <Timestamp>2006-08-31T00:00:00.000+01:00</Timestamp
      <Originator>Mike</Originator>
    </AuditTrailEntry>
  - <AuditTrailEntry>
    - <Data>
        <Attribute name="result">reject</Attribute>
      </Data>
      <WorkflowModelElement>get review 3</WorkflowModelEleme
      <EventType>complete</EventType>
      <Timestamp>2006-09-02T00:00:00.000+01:00</Timestamp
      <Originator>Mary</Originator>
    </AuditTrailEntry>
  - <AuditTrailEntry>
      <WorkflowModelElement>time-out 1</WorkflowModelElement>
      <EventType>complete</EventType>
      <Timestamp>2006-09-03T00:00:00.000+01:00</Timestamp>
      <Originator />
    </AuditTrailEntry>
  - <AuditTrailEntry>
```

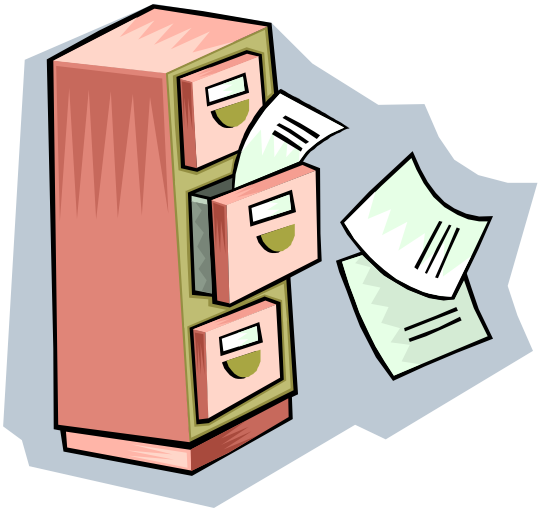

attributes of an event

end of activity

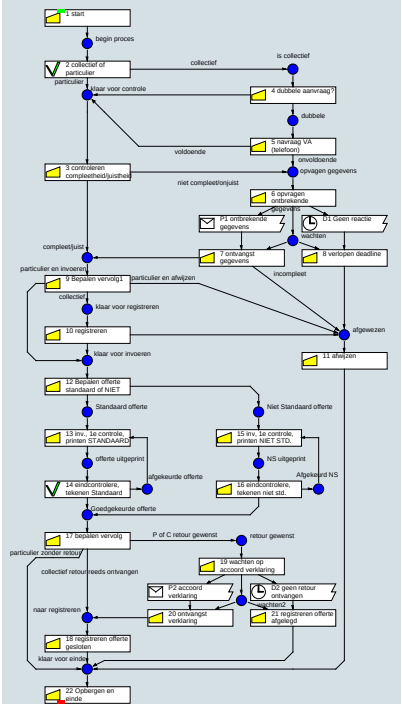
activity

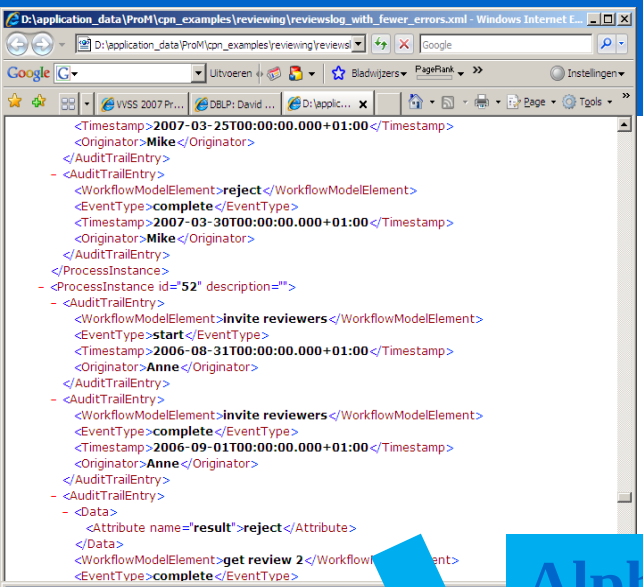
instance

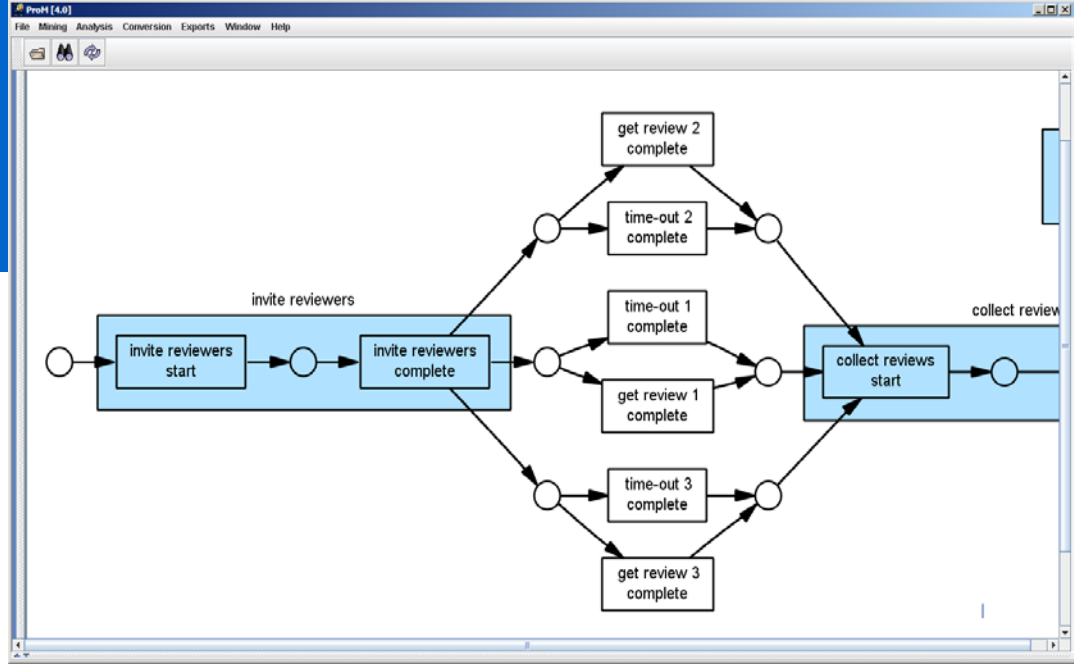
Process Mining: The alpha algorithm



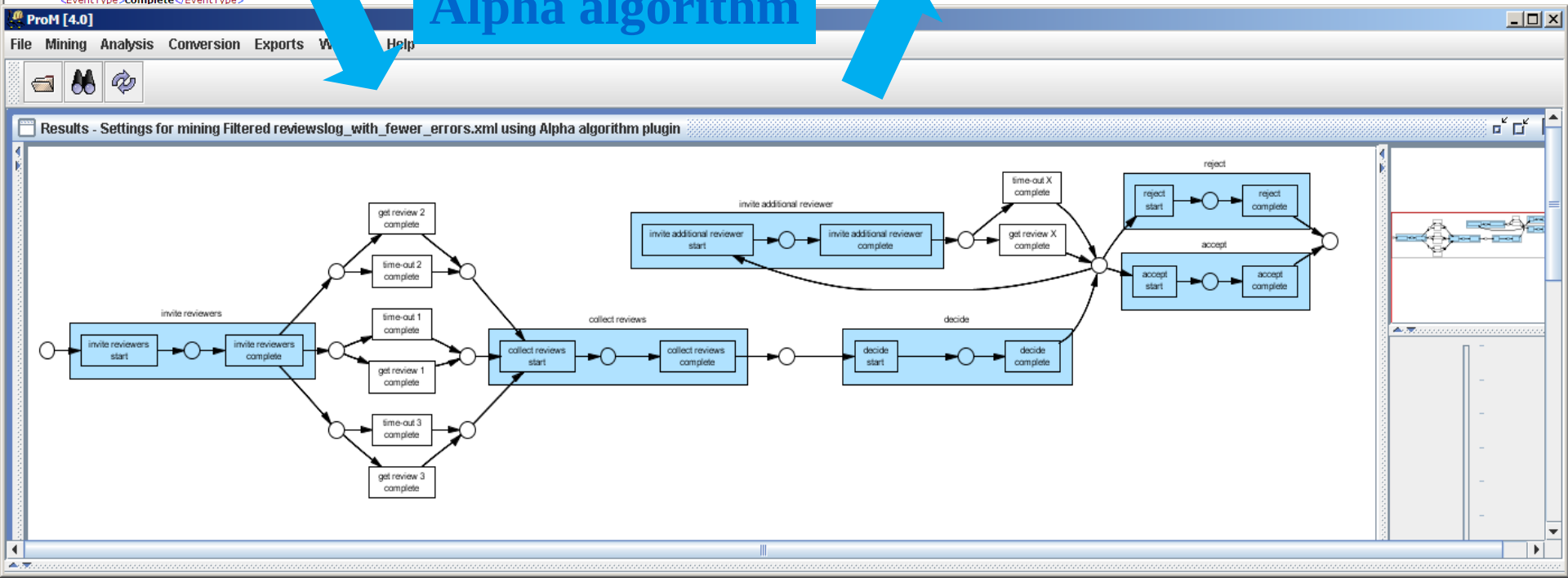
α
algorithm



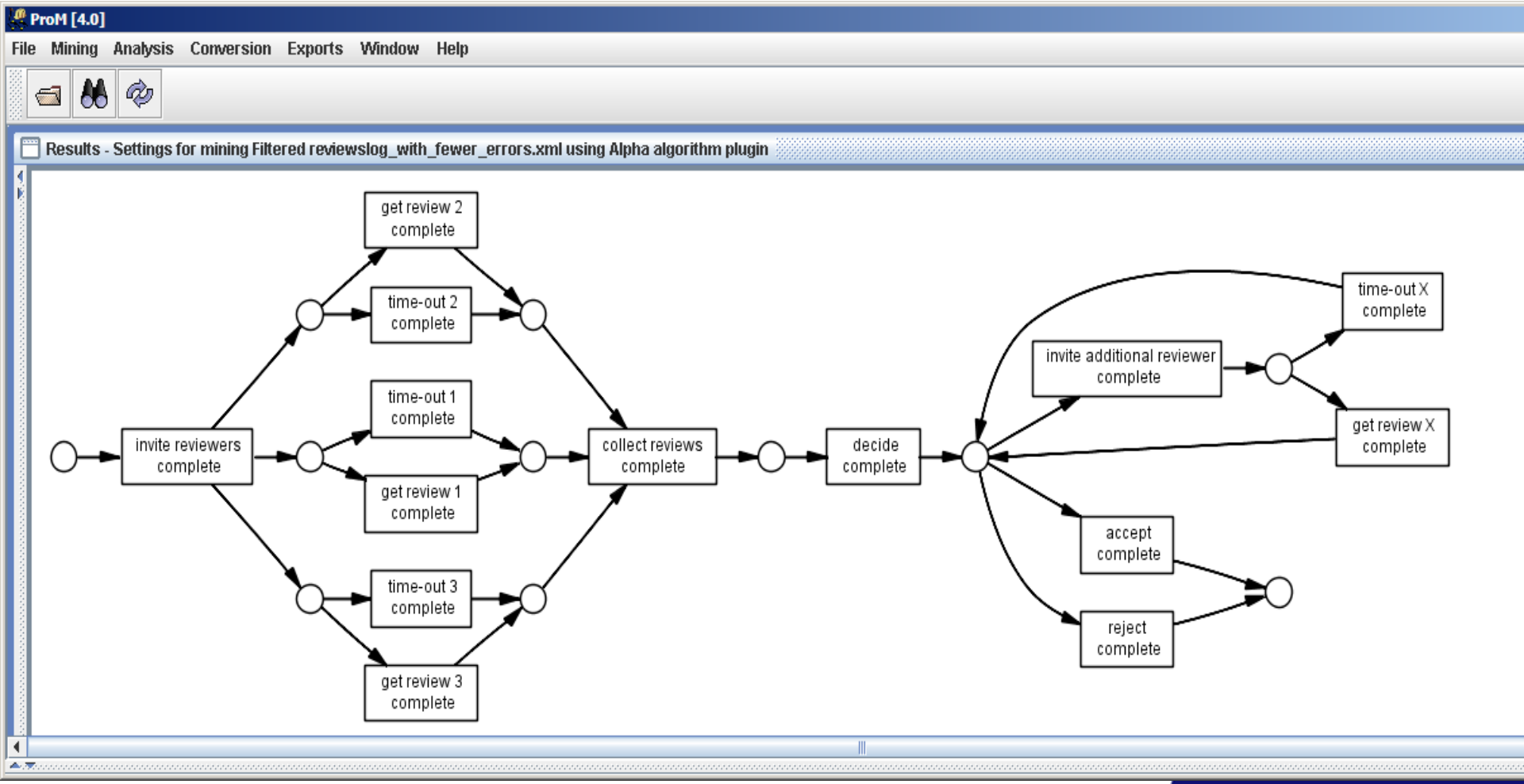




Alpha algorithm



Without transactional information (just completes)



Example log

- Minimal information in log: case id's and task id's.
- Additional information: event type, time, resources, and data.
- Sequences:
  - 1: ABCD
  - 2: ACBD
  - 3: ABCD
  - 4: ACBD
  - 5: EF
- So this log there are three possible sequences:
  - ABCD
  - ACBD
  - EF

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

>,→,||,# relations

- Direct succession: **x>y iff for some case x is directly followed by y.**
- Causality: **x→y iff x>y and not y>x.**
- Parallel: **x||y iff x>y and y>x**
- Choice: **x#y iff not x>y and not y>x.**

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

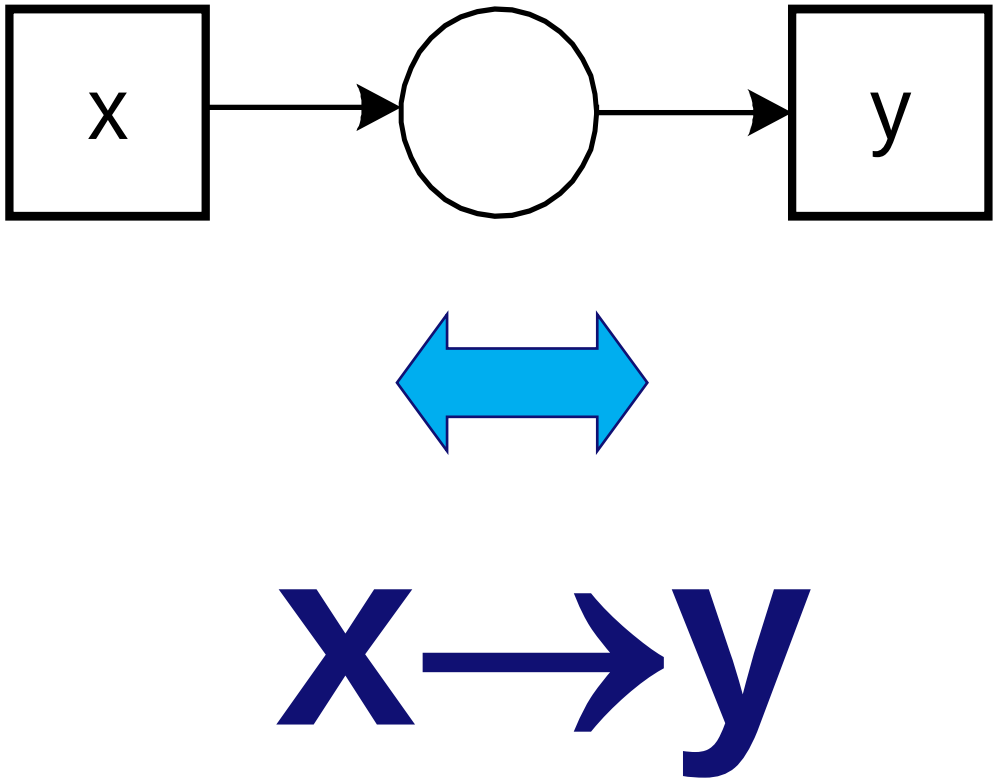
ABCD
ACBD
EF

A>B
A>C
B>C
B>D
C>B
C>D
E>F

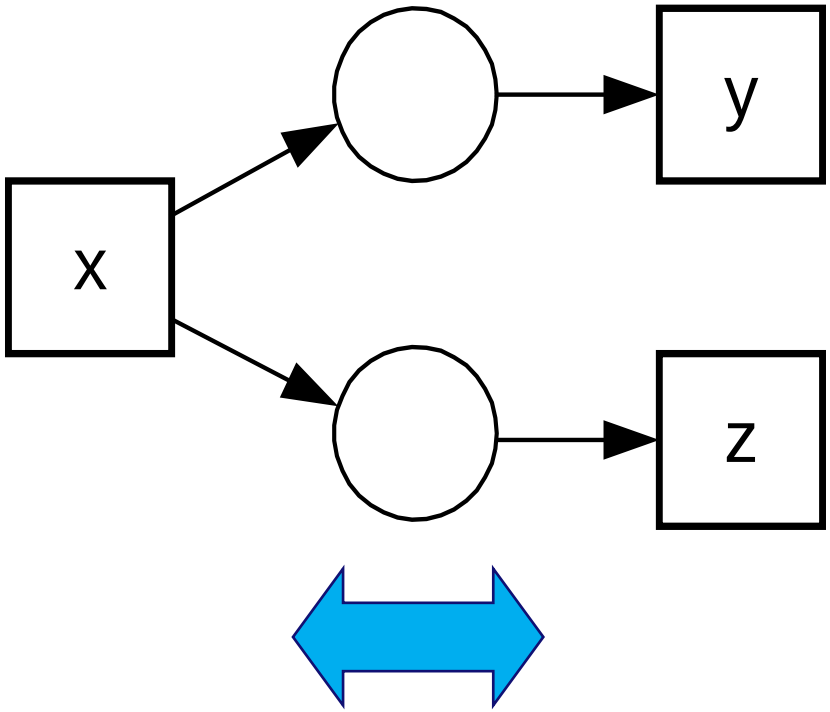
A→B
A→C
B→D
C→D
E→F

B||C
C||B

Basic idea (1)

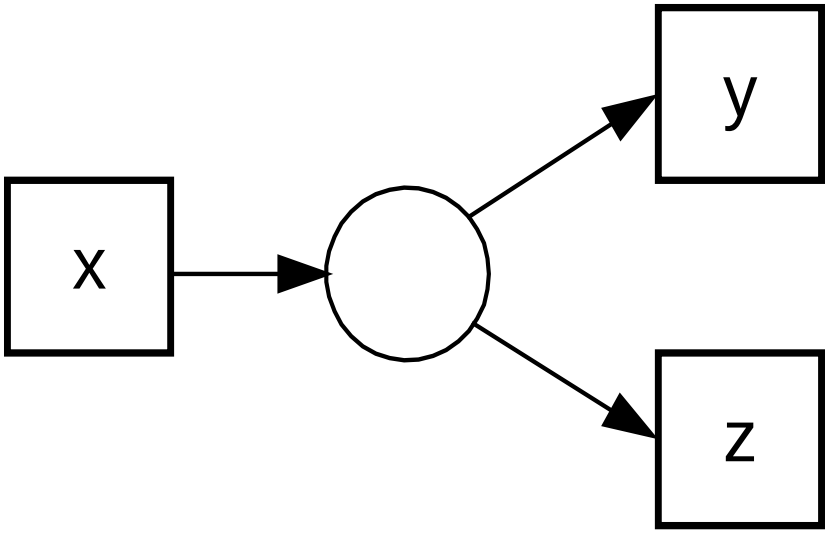


Basic idea (2)



$x \rightarrow y$, $x \rightarrow z$, and $y \| z$

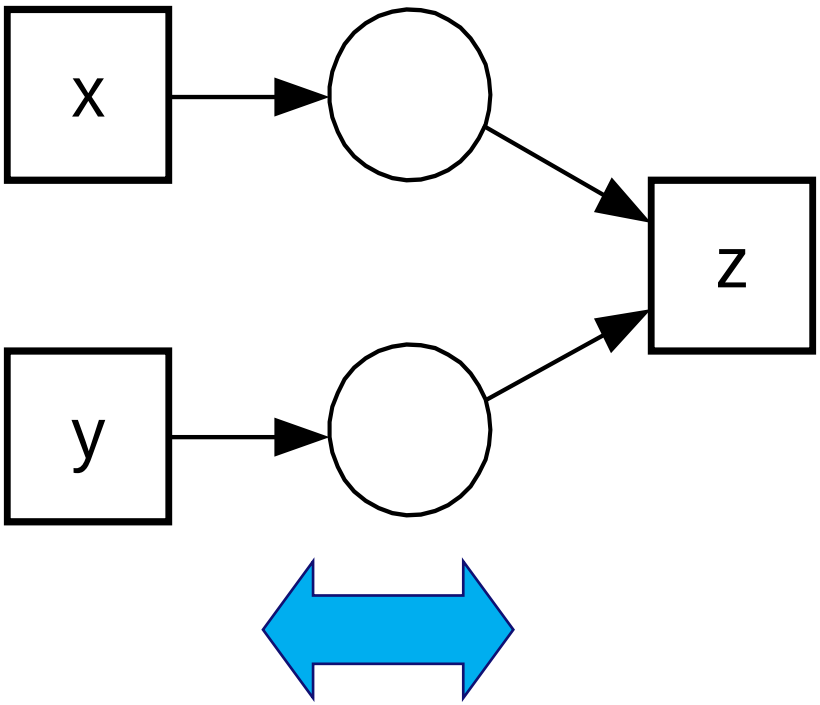
Basic idea (3)



$\Leftrightarrow$

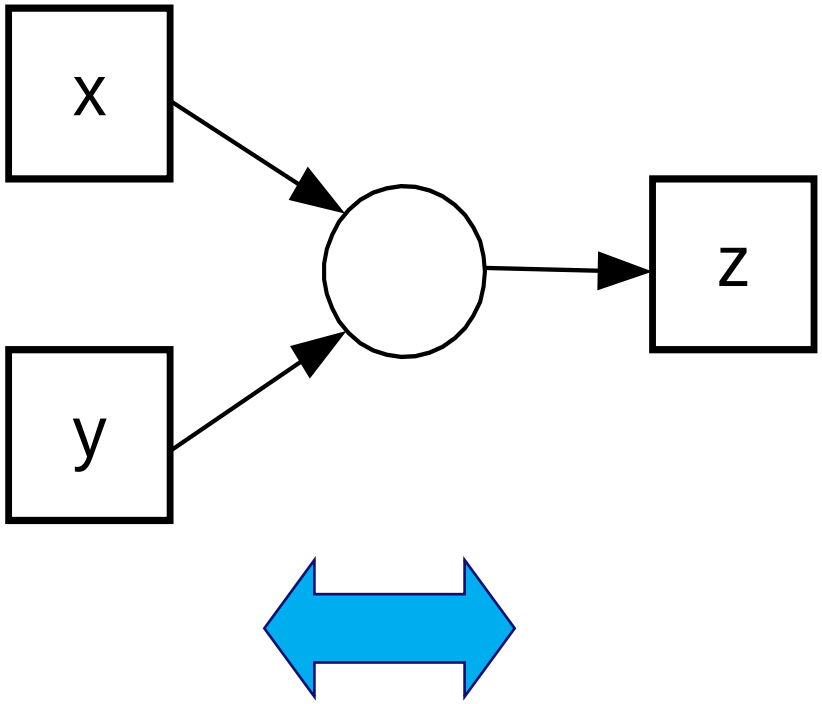
$x \rightarrow y$, $x \rightarrow z$, and $y \# z$

Basic idea (4)



$x \rightarrow z$, $y \rightarrow z$, and $x \| y$

Basic idea (5)



$x \rightarrow z$, $y \rightarrow z$, and $x \# y$

It is not that simple!
Basic alpha algorithm

Let W be a workflow log over T. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge$
   $\forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \;\wedge\; \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \;\wedge\; \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in$
   $Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

Example revisited

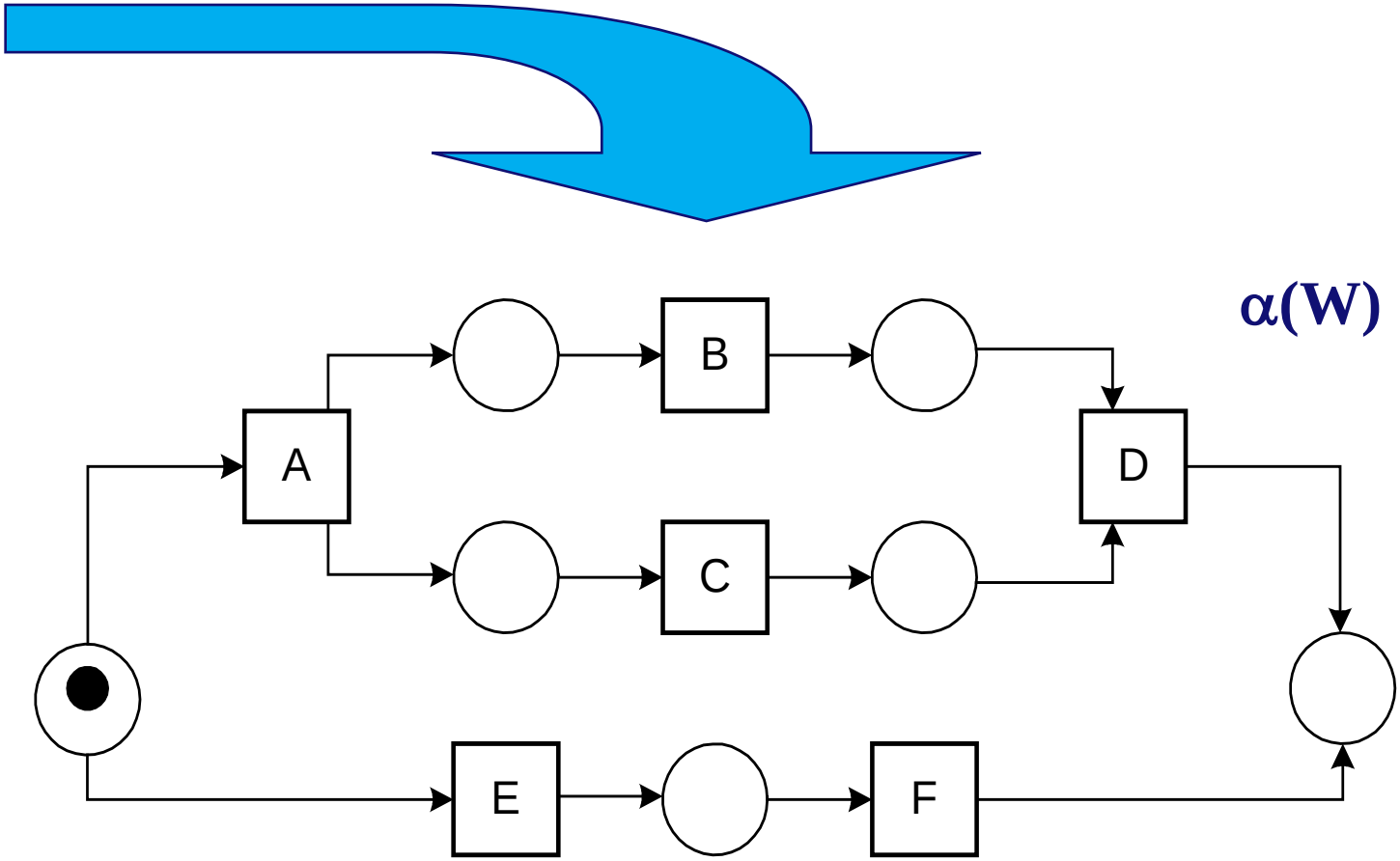
W:

```
case 1 : task A
case 2 : task A
case 3 : task A
case 3 : task B
case 1 : task B
case 1 : task C
case 2 : task C
case 4 : task A
case 2 : task B
case 2 : task D
case 5 : task E
case 4 : task C
case 1 : task D
case 3 : task C
case 3 : task D
case 4 : task B
case 5 : task F
case 4 : task D
```

A>B
A>C
B>C
B>D
C>B
C>D
E>F

A→B
A→C
B→D
C→D
E→F

B||C
C||B



Exercise (1)

- **What does the alpha algorithm produce for a log consisting only of the following traces?**
  - **ABCD**
  - **ACBD**
  - **AED**

Let W be a workflow log over T. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

- **Direct succession: x>y iff for some case x is directly followed by y.**
- **Causality: $x \rightarrow y$ iff x>y and not y>x.**
- **Parallel: x||y iff x>y and y>x**
- **Choice: x#y iff not x>y and not y>x.**

Another example taken step-by-step ...

Definition 3.1 (Workflow trace, Workflow log). *Let T be a set of tasks. $\sigma \in T^*$ is a* workflow trace *and $W \in \mathcal{P}(T^*)$ is a* workflow log.[2]

The workflow trace of case 1 in Table 1 is $ABCD$. The workflow log corresponding to Table 1 is

$$\{ABCD, ACBD, AED\}.$$

ase identifier	task identifier
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task A
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task E
case 5	task D
case 4	task D

Definition 3.2 (Log-based ordering relations). *Let W be a workflow log over T, i.e., $W \in \mathcal{P}(T^*)$. Let $a, b \in T$:*

- $a >_W b$ *iff there is a trace* $\sigma = t_1 t_2 t_3 \ldots t_{n-1}$ *and* $i \in \{1, \ldots, n-2\}$ *such that* $\sigma \in W$ *and* $t_i = a$ *and* $t_{i+1} = b$,
- $a \rightarrow_W b$ *iff* $a >_W b$ *and* $b \not>_W a$,
- $a \#_W b$ *iff* $a \not>_W b$ *and* $b \not>_W a$, *and*
- $a \|_W b$ *iff* $a >_W b$ *and* $b >_W a$.

Consider the workflow log $W = \{ABCD, ACBD, AED\}$ (i.e., the log shown in Table 1). Relation $>_W$ describes which tasks appeared in sequence (one directly following the other). Clearly, $A >_W B$, $A >_W C$, $A >_W E$, $B >_W C$, $B >_W D$, $C >_W B$, $C >_W D$, and $E >_W D$. Relation $\rightarrow_W$ can be computed from $>_W$ and is referred to as the *(direct) causal relation* derived from workflow log W. $A \rightarrow_W B$, $A \rightarrow_W C$, $A \rightarrow_W E$, $B \rightarrow_W D$, $C \rightarrow_W D$, and $E \rightarrow_W D$. Note that $B \not\rightarrow_W C$ because $C >_W B$. Relation $\|_W$ suggests potential parallelism. For log W, tasks B and C seem to be in parallel, i.e., $B \|_W C$ and $C \|_W B$. If two tasks can follow each other directly in any order, then all possible interleavings are present and,

A>B
A>C
A>E
B>C
D>D
C>B
C>D
E>D

A→B
A→C
A→E
B→D
C→D
E→D

B||C
C||B

Definition 4.3 (Mining algorithm α). *Let W be a workflow log over T. $\alpha(W)$ is defined as follows:*

$$W = \{ABCD, ACBD, AED\}$$

1. $T_W = \{t \in T \mid \exists_{\sigma \in W} t \in \sigma\}$,
2. $T_I = \{t \in T \mid \exists_{\sigma \in W} t = first(\sigma)\}$,
3. $T_O = \{t \in T \mid \exists_{\sigma \in W} t = last(\sigma)\}$,
4. $$X_W = \{(A, B) \mid \wedge\ \forall_{a \in A} \forall \ldots \wedge\ \forall_{b_1, b_2 \in} \ldots$$
5. $$Y_W = \{(A, B \ldots \wedge B \subseteq B' \Longrightarrow (A, B) = (A', B')\},$$
6. $P_W = \{p_{(A,B)} \mid (A, B) \in Y_W\} \cup \{i_W, o_W\}$,
7. $$F_W = \{(a, p_{(A,B)}) \mid (A, B) \in Y_W \ \wedge\ a \in A\} \cup \{(p_{(A,B)}, b) \mid (A, B) \in Y_W \ \wedge\ b \in B\} \cup \{(i_W, t) \mid t \in T_I\} \cup \{(t, o_W) \mid t \in T_O\},$$

   and
8. $\alpha(W) = (P_W, T_W, F_W)$.

1. $T_W = \{A, B, C, D, E\}$,

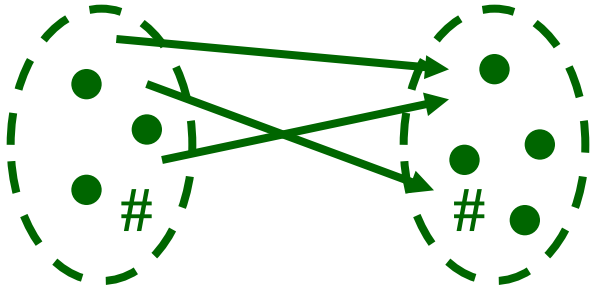
2. $T_I = \{A\}$,

3. $T_O = \{D\}$,

Definition 4.3 (Mining algorithm α). *Let W be a workflow log over T. $\alpha(W)$ is defined as follows:*

$$W = \{ABCD, ACBD, AED\}$$

1. $T_W = \{t \in T \mid \exists_{\sigma \in W} t \in \sigma\}$,
2. $T_I = \{t \in T \mid \exists_{\sigma \in W} t = first(\sigma)\}$,
3. $T_O = \{t \in T \mid \exists_{\sigma \in W} t = last(\sigma)\}$,
4. $$X_W = \{(A, B) \mid A \subseteq T_W \wedge B \subseteq T_W \wedge \forall_{a \in A} \forall_{b \in B} a \rightarrow_W b \wedge \forall_{a_1, a_2 \in A} a_1 \#_W a_2 \wedge \forall_{b_1, b_2 \in B} b_1 \#_W b_2\},$$



A and B need to be non-empty.

5.

4.

$$X_W = \{(\{A\}, \{B\}), (\{A\}, \{C\}), (\{A\}, \{E\}), (\{B\}, \{D\}), (\{C\}, \{D\}), (\{E\}, \{D\}), (\{A\}, \{B, E\}), (\{A\}, \{C, E\}), (\{B, E\}, \{D\}), (\{C, E\}, \{D\})\},$$

8. $\alpha(W) = (P_W, T_W, F_W)$.

4.

$$X_W = \{(\{A\},\{B\}),(\{A\},\{C\}),(\{A\},\{E\}),$$
$$(\{B\},\{D\}),(\{C\},\{D\}),(\{E\},\{D\}),$$
$$(\{A\},\{B,E\}),(\{A\},\{C,E\}),(\{B,E\},\{D\}),$$
$$(\{C,E\},\{D\})\},$$

✓ 5.

$$Y_W = \{(A,B) \in X_W \mid \forall_{(A',B')\in X_W} A \subseteq A' \wedge B \subseteq B' \Longrightarrow (A,B) = (A',B')\},$$

$$\boxed{W = \{ABCD, ACBD, AED\}}$$

5.

$$Y_W = \{(\{A\},\{B,E\}),(\{A\},\{C,E\}),(\{B,E\},\{D\}),(\{C,E\},\{D\})\},$$

and

8. $\alpha(W) = (P_W, T_W, F_W)$.

5.

$$Y_W = \{(\{A\},\{B,E\}),(\{A\},\{C,E\}),(\{B,E\},\{D\}),(\{C,E\},\{D\})\},$$

$\wedge\ \forall_{a\in A}\forall_{b\in B} a \rightarrow_W b\ \wedge\ \forall_{a_1,a_2\in A} a_1 \#_W a_2$
$\wedge\ \forall_{b_1,b_2\in B} b_1 \#_W b_2\},$

$$W = \{ABCD, ACBD, AED\}$$

$$P_W = \{i_W, o_W, p_{(\{A\},\{B,E\})}, p_{\{A\},\{C,E\})}, p_{(\{B,E\},\{D\})}, p_{(\{C,E\},\{D\})}\},$$

$$F_W = \{(i_W, A), (A, p_{(\{A\},\{B,E\})}), (p_{(\{A\},\{B,E\})}, B) \ldots (D, o_W)\},$$

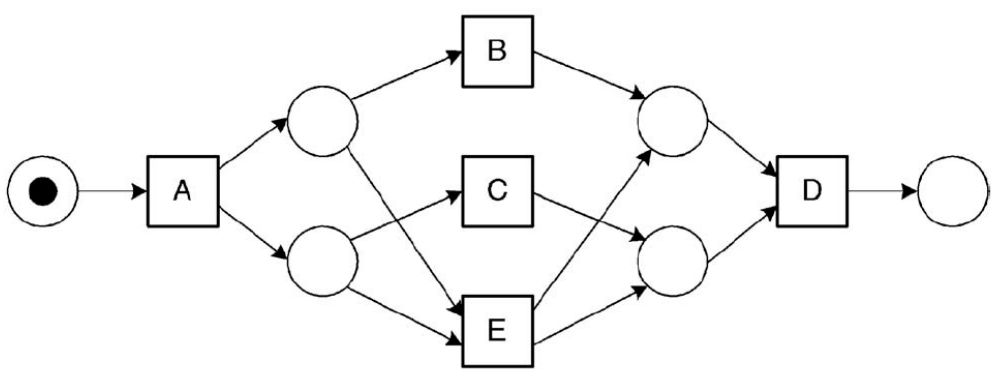
✓ 6. $P_W = \{p_{(A,B)} \mid (A,B) \in Y_W\} \cup$

✓ 7.

$F_W = \{(a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge$
$\cup\ \{(p_{(A,B)}, b) \mid (A,B) \in Y_W$
$\cup \{(i_W, t) \mid t \in T_I\}\ \cup \{(t, o$

and

✓ 8. $\alpha(W) = (P_W, T_W, F_W).$



Exercise (2)

- **What does the alpha algorithm produce for a log consisting only of the following traces?**
  - **ACD**
  - **BCE**

Let W be a workflow log over T. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

- **Direct succession: x>y iff for some case x is directly followed by y.**
- **Causality: $x \rightarrow y$ iff x>y and not y>x.**
- **Parallel: x||y iff x>y and y>x**
- **Choice: x#y iff not x>y and not y>x.**

Exercise (3)

- **What does the alpha algorithm produce for a log consisting only of the following traces?**
  - **ACEG**
  - **AECG**
  - **BDFG**
  - **BFDG**

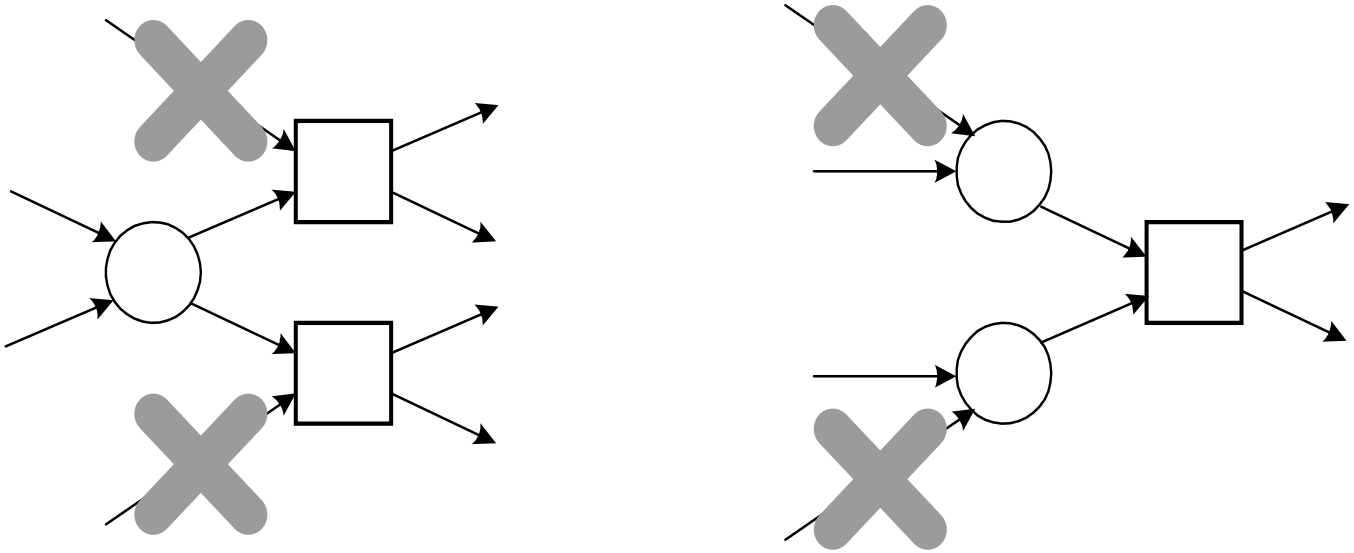
Let W be a workflow log over T. $\alpha(W)$ is defined as follows.

1. $T_W = \{ t \in T \mid \exists_{\sigma \in W}\, t \in \sigma \}$,
2. $T_I = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{first}(\sigma) \}$,
3. $T_O = \{ t \in T \mid \exists_{\sigma \in W}\, t = \mathit{last}(\sigma) \}$,
4. $X_W = \{ (A,B) \mid A \subseteq T_W \wedge A \neq \emptyset \wedge B \subseteq T_W \wedge B \neq \emptyset \wedge \forall_{a \in A} \forall_{b \in B}\, a \rightarrow_W b \wedge \forall_{a1,a2 \in A}\, a_1 \#_W a_2 \wedge \forall_{b1,b2 \in B}\, b_1 \#_W b_2 \}$,
5. $Y_W = \{ (A,B) \in X \mid \forall_{(A',B') \in X}\, A \subseteq A' \wedge B \subseteq B' \Rightarrow (A,B) = (A',B') \}$,
6. $P_W = \{ p_{(A,B)} \mid (A,B) \in Y_W \} \cup \{i_W, o_W\}$,
7. $F_W = \{ (a, p_{(A,B)}) \mid (A,B) \in Y_W \wedge a \in A \} \cup \{ (p_{(A,B)}, b) \mid (A,B) \in Y_W \wedge b \in B \} \cup \{ (i_W, t) \mid t \in T_I \} \cup \{ (t, o_W) \mid t \in T_O \}$, and
8. $\alpha(W) = (P_W, T_W, F_W)$.

- **Direct succession: x>y iff for some case x is directly followed by y.**
- **Causality: $x \rightarrow y$ iff x>y and not y>x.**
- **Parallel: x||y iff x>y and y>x**
- **Choice: x#y iff not x>y and not y>x.**

Properties of the Alpha algorithm

- **If log is complete with respect to relation >, it can be used to mine any SWF-net!**
- ***Structured Workflow Nets* (SWF-nets) have no implicit places and the following two constructs cannot be used:**

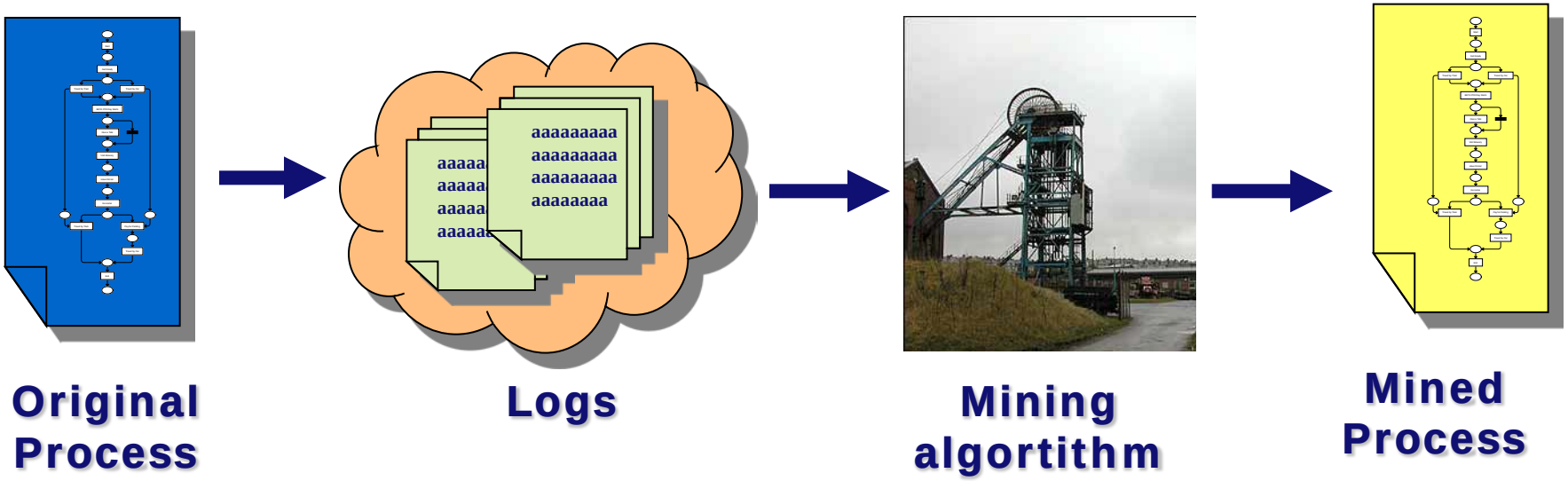


(Short loops require some refinement but not a problem.)

Alpha algorithm

- Mainly of theoretical interest!
- Too simple to be applicable to real-life logs.
- Does not address issues such as noise, etc.
- Should NOT be taken as a benchmark.
- However, the algorithm reveals:
  - basic process mining ideas and concepts in 8 lines,
  - theoretical limits of process mining.

Basic test for any mining algorithm: Rediscovery



Can the mined process generate all the behavior in the log?

How close is the behavior of the mined process to the original one?

Controlled choices cannot be rediscovered (and in may cases this is good!)

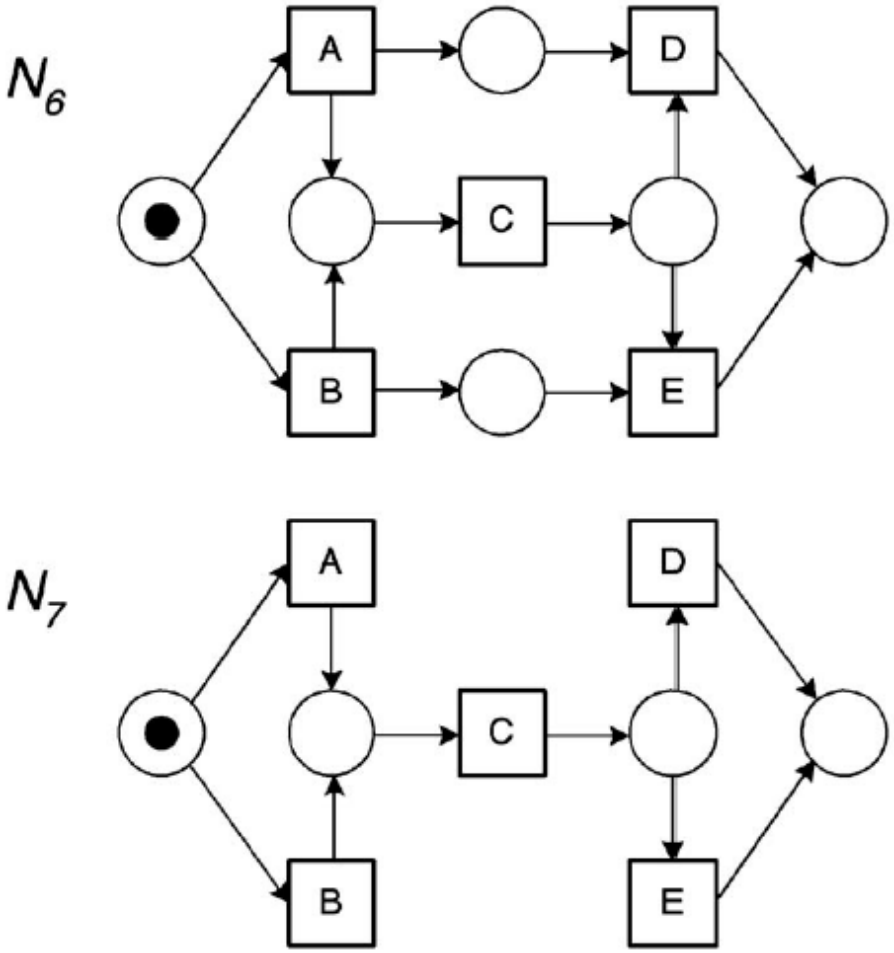


Fig. 7. The nonfree-choice WF-net N_6 cannot be rediscovered by the α algorithm.

Log only contains information about behavior and not structure

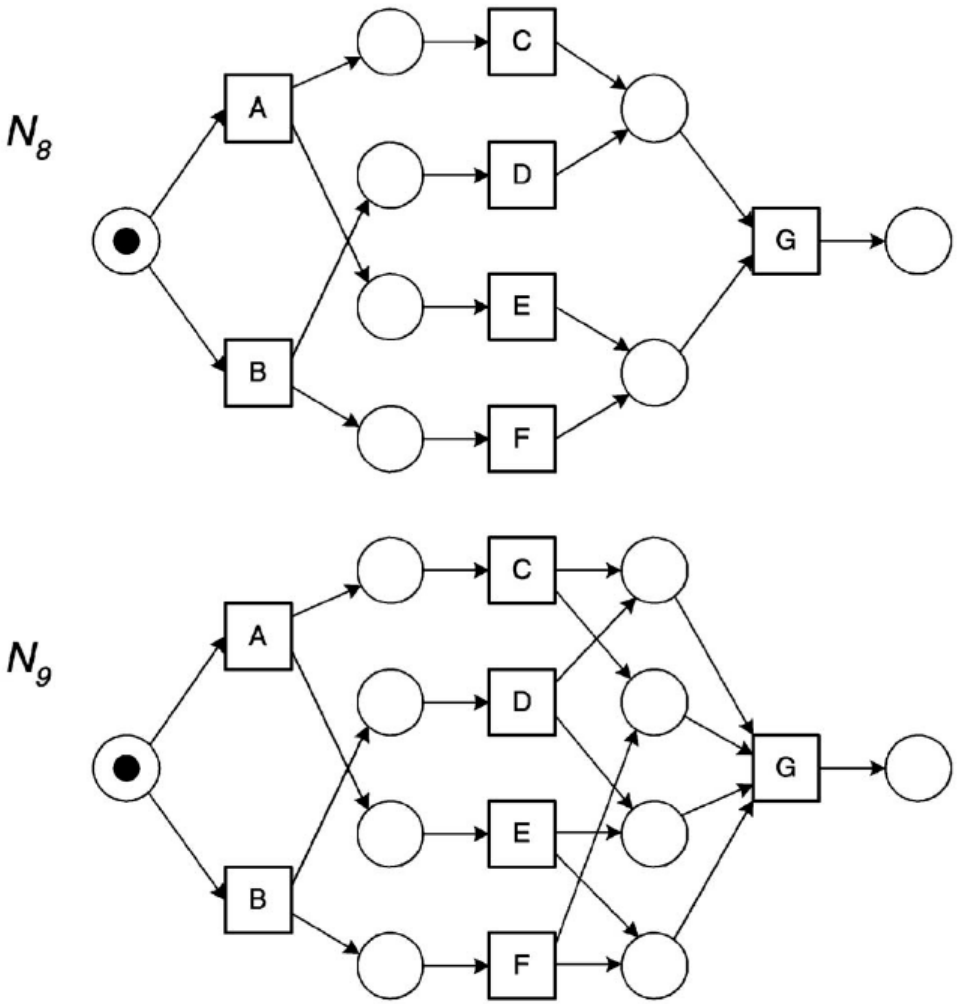


Fig. 8. WF-net N_8 cannot be rediscovered by the α algorithm. Nevertheless, α returns a WF-net which is behavioral equivalent.

Completeness notion may be too crude in some cases

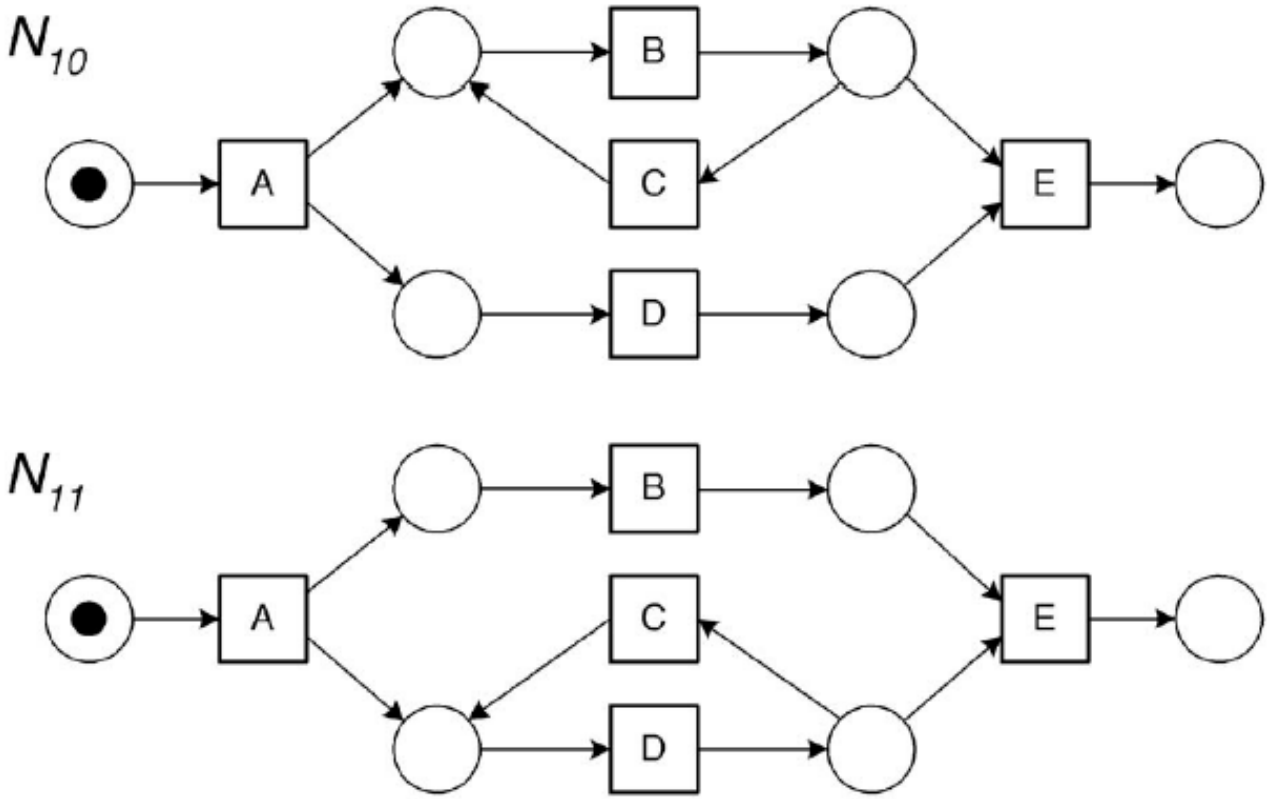


Fig. 9. Although both WF-nets are not behavioral equivalent they are identical with respect to $>$.

Another example of behaviorally equivalent SWF-nets

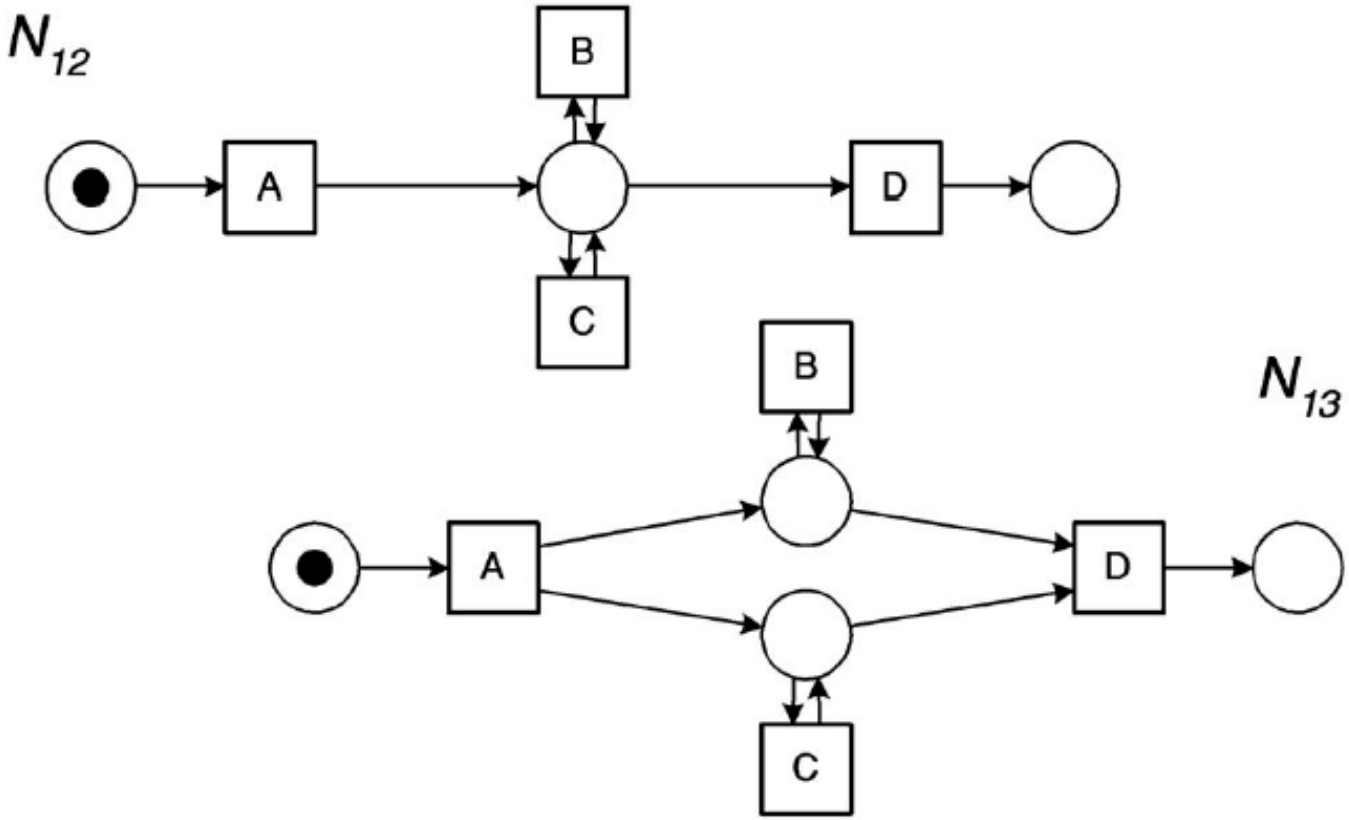
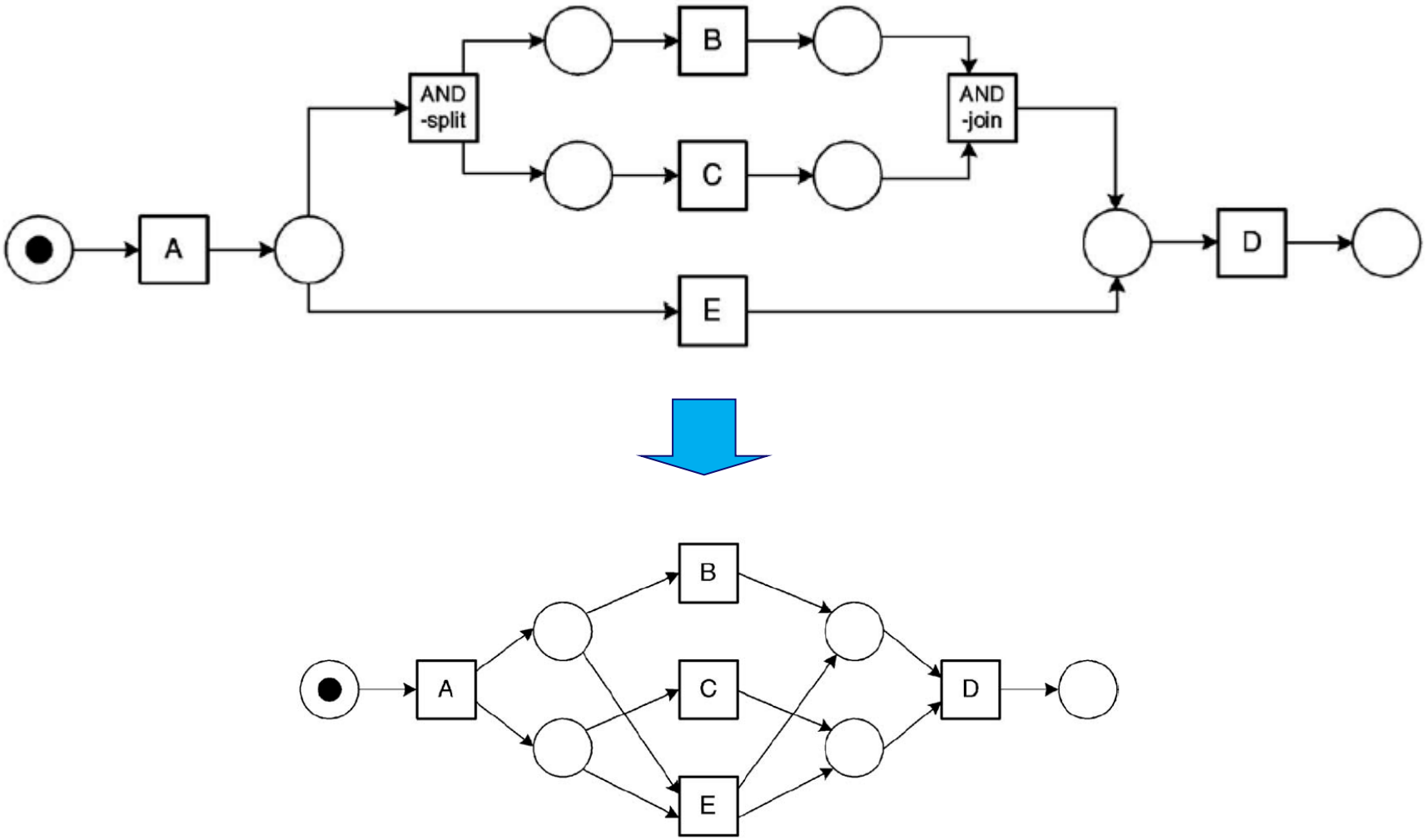
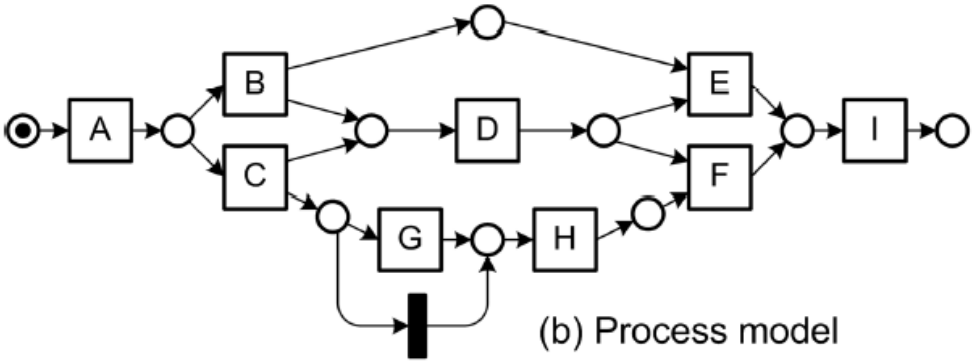


Fig. 10. Both SWF-nets are behavioral equivalent and, therefore, any algorithm will be unable to distinguish N_{12} from N_{13} (assuming a notion of completeness based on $>$).

Silent steps (and duplicate steps) cannot be discovered





(b) Process model

(c) Process model

fitness +
precision +
generalization +
structure +

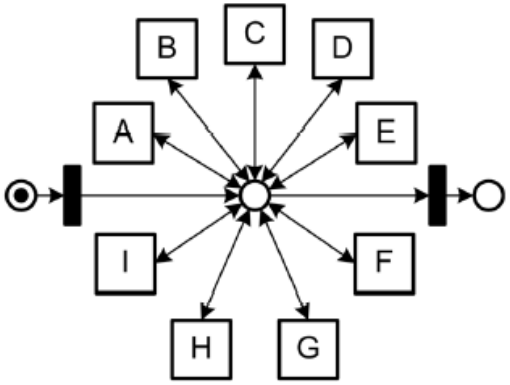
fitness +
precision -
generalization +
structure +

No. of Instances	Log Traces
1207	ABDEI
145	ACDGHFI
56	ACGDHFI
23	ACHDFI
28	ACDHFI

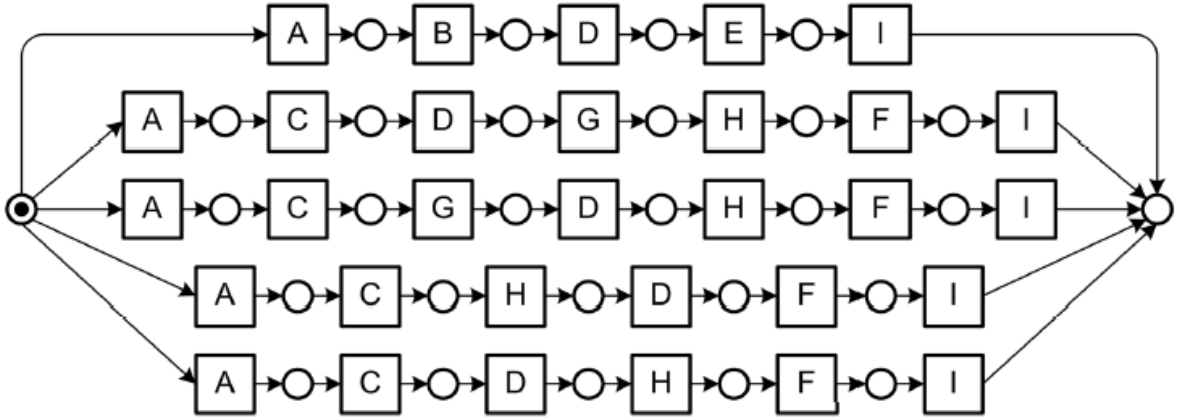
(a) Event Log

fitness -
precision +
generalization -
structure +

fitness +
precision +
generalization -
structure -



(d) Process model



(e) Process model

Simple process mining algorithms tend to:

- Have problems with complex control-flow constructs. **For example, many process mining algorithms are unable to deal with non-free-choice constructs and complex nested loops.**
- Not allow for duplicates. **In the event log it is not possible to distinguish between activities that are logged in a similar way, i.e., there are multiple activities that have the same "footprint" in the log. As a result, most algorithms map these different activities onto a simple activity thus making the model incorrect or counter-intuitive.**
- Silent steps. **Things that are not recorded cannot be discovered.**
- Underfit (i.e., overgeneralize) or overfit. **Many algorithms have a tendency to overgeneralize, i.e., the discovered model allows for much more behavior than actually recorded in the log. In some circumstances this may be desirable. However, there seems to be a need to flexibly balance between "overfitting" and "underfitting".**
- Yield inconsistent models. **For more complicated processes many algorithms have a tendency to produce models that may have deadlocks and/or livelocks. It seems vital that the generated models satisfy some soundness requirements (e.g., the soundness property).**

Other Process Discovery Techniques

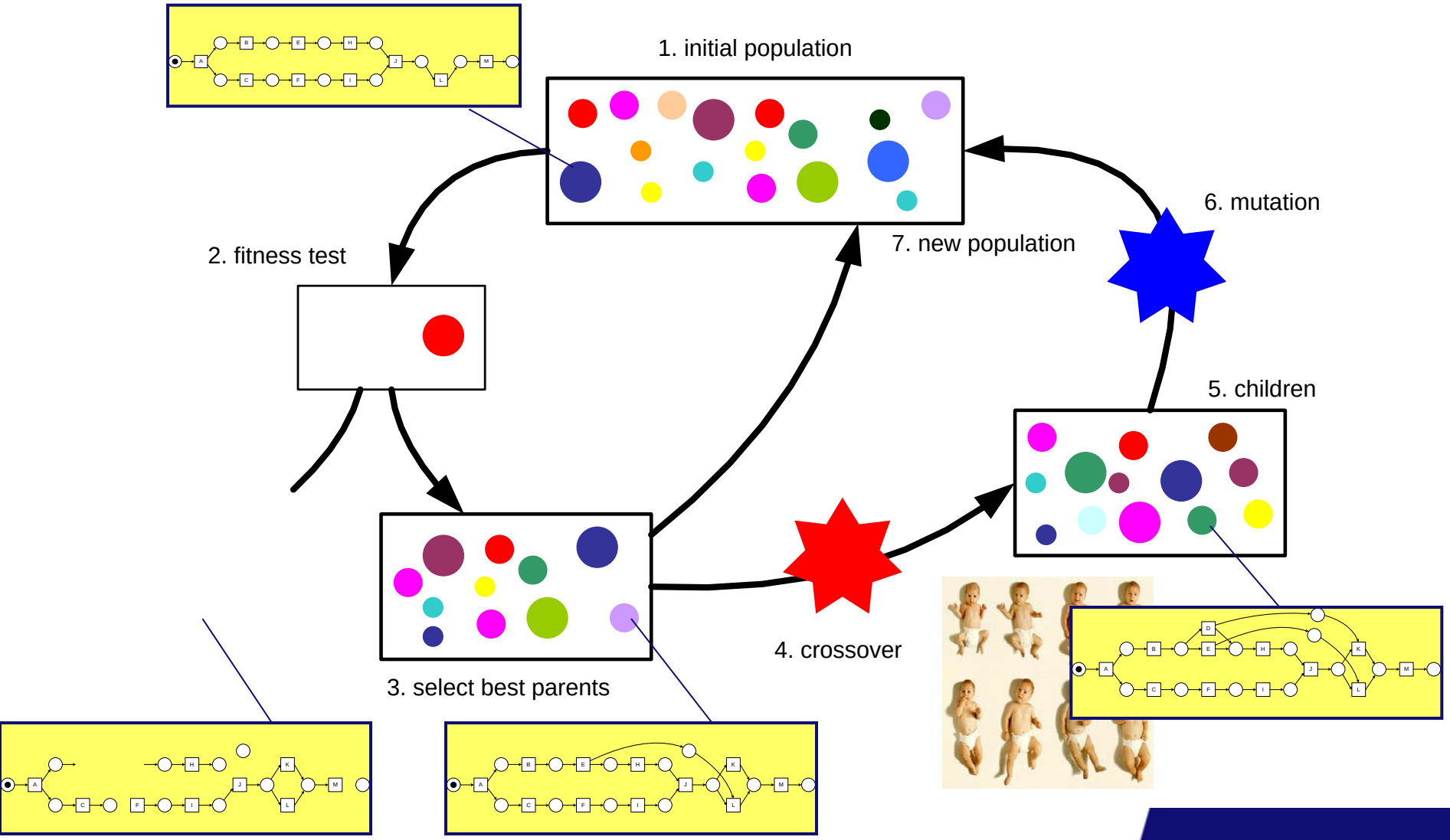
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

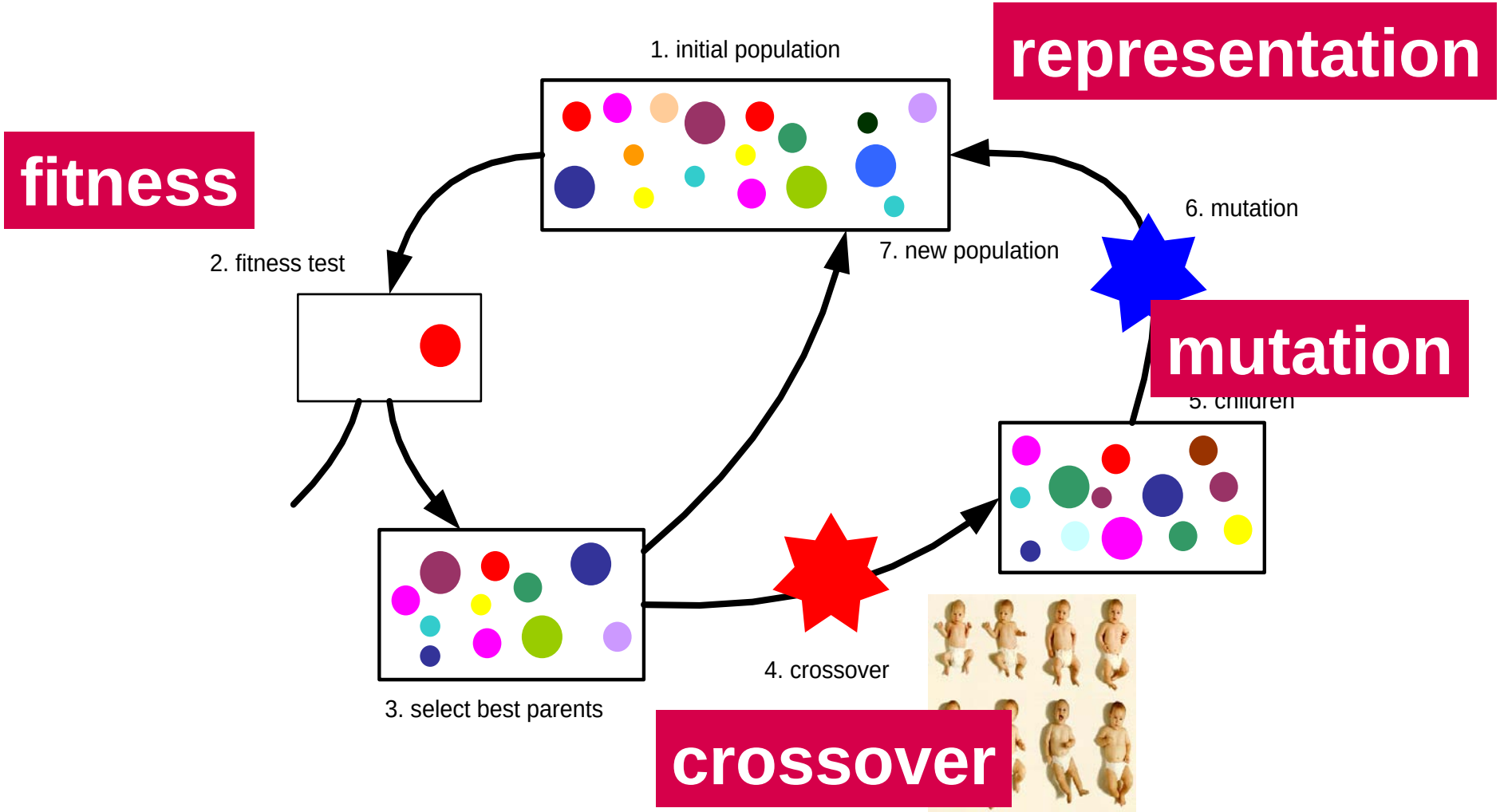
Overview of process discovery techniques

- Classical techniques (e.g., learning state machines and the theory of regions): cannot handle concurrency and/or do not generalize (i.e., if it did not happen, it cannot happen).
- Algorithmic techniques
  - Alpha miner
  - Alpha+, Alpha++, Alpha#
  - Heuristic miner
  - Multi phase miner
  - ...
- Genetic process mining
- Region-based process mining
  - State-based regions
  - Language based regions

Genetic Mining
(Ana Karla Alves de Medeiros et al.)

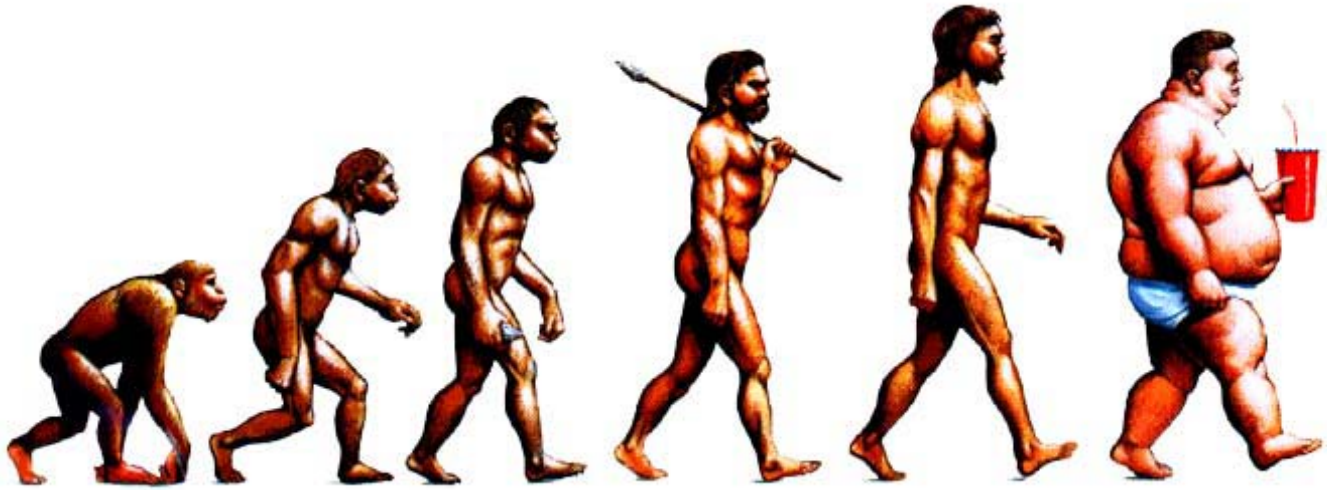


Design choices



Properties of Genetic Mining

- **Requires a lot of computing power.**
- **Can deal with noise, infrequent behavior, duplicate tasks, invisible tasks, etc.**
- **Allows for incremental improvement and combinations with other approaches (heuristics post-optimization, etc.).**



Balancing Between Overfitting and Underfitting

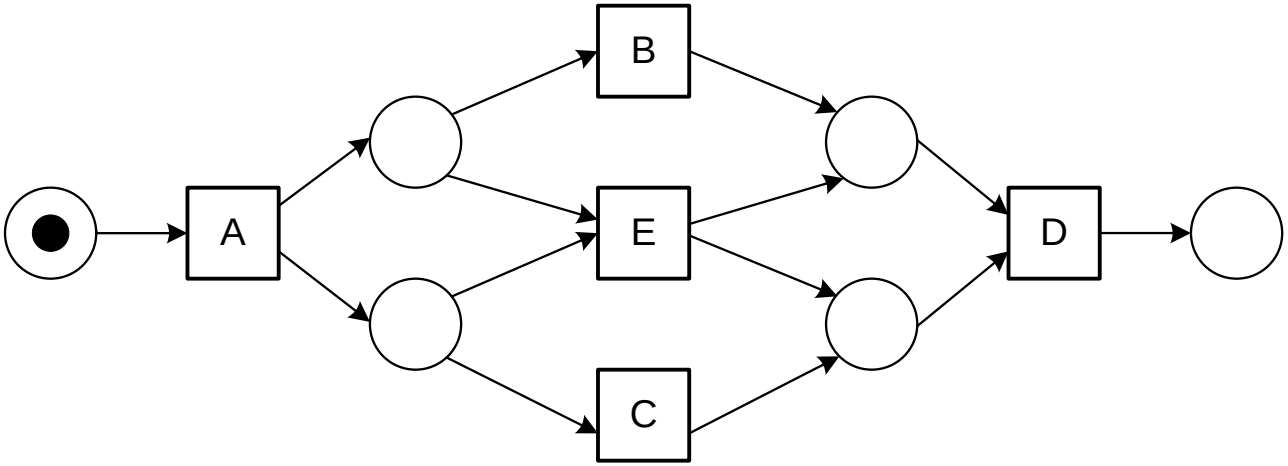
Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Challenge: Balancing Between Underfitting and Overfitting

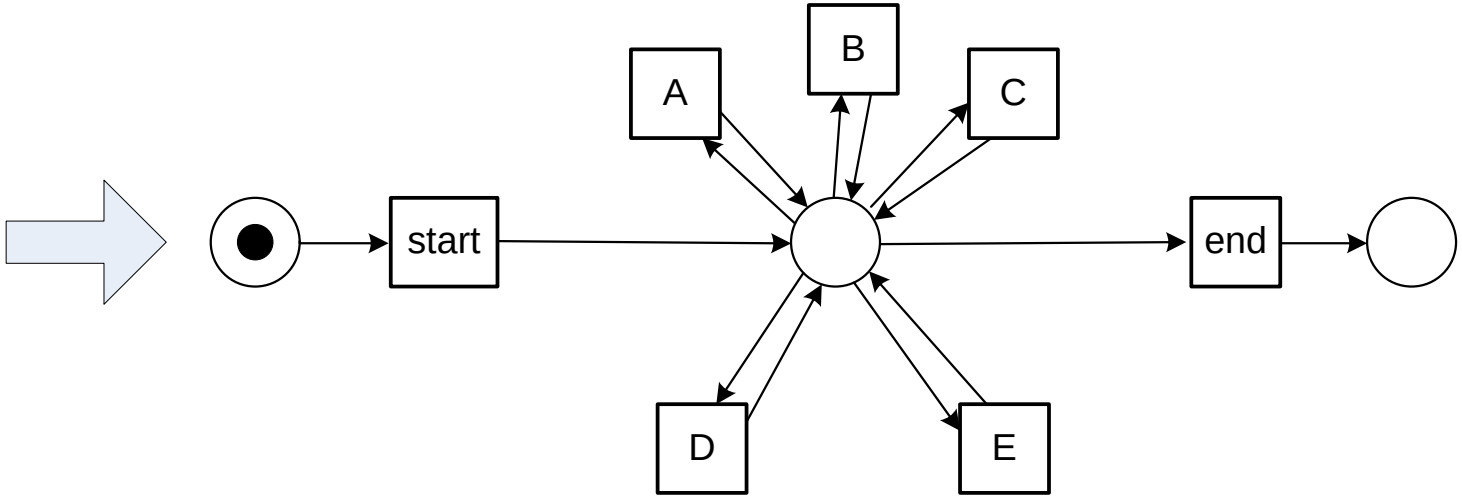
The essence

ABCD
ACBD
AED
ABCD
ABCD
AED
ACBD
...

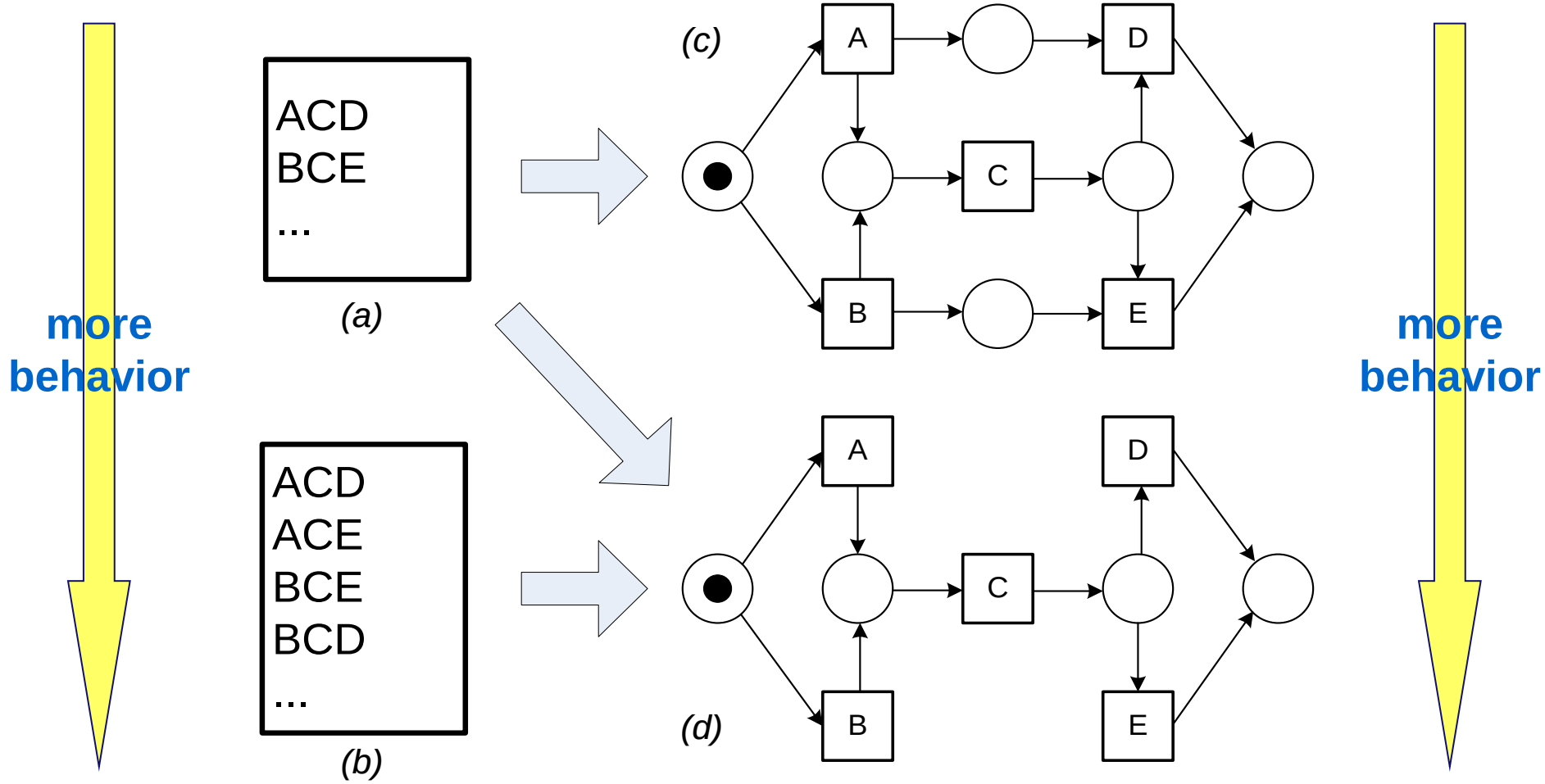


But ...

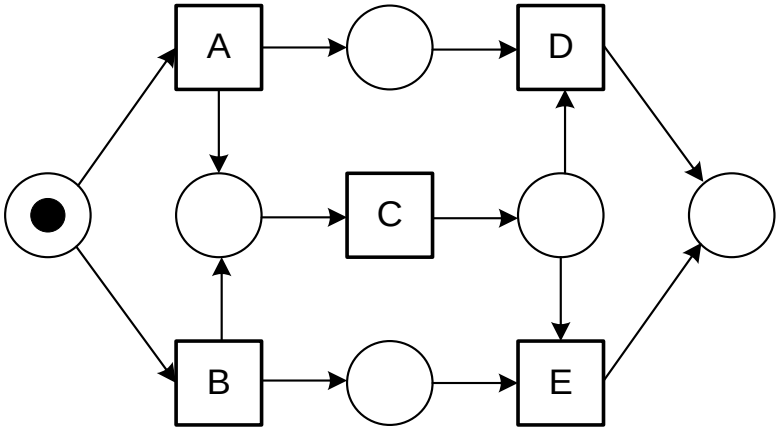
Any log containg activities A, B, C, D, and E.

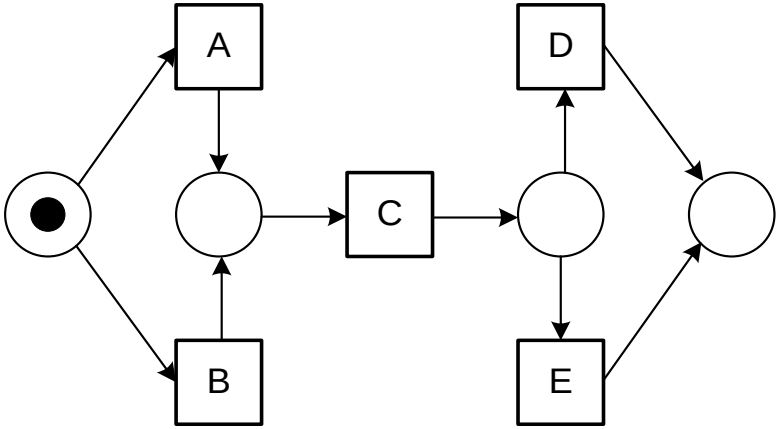


Finding a balance

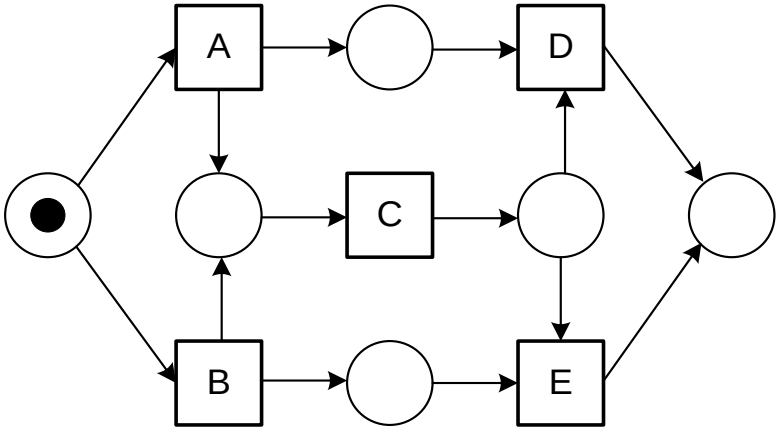


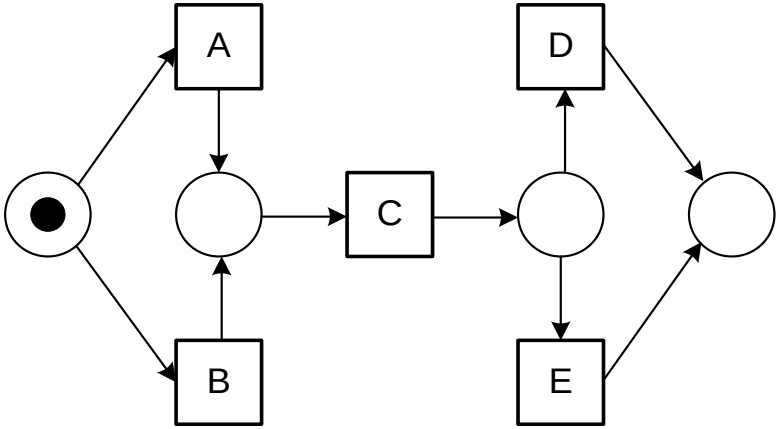
ACD	**99**
ACE	**0**
BCE	**85**
BCD	**0**



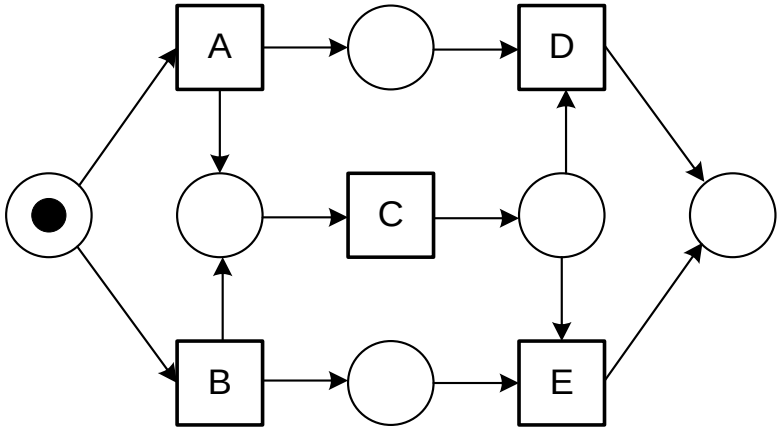


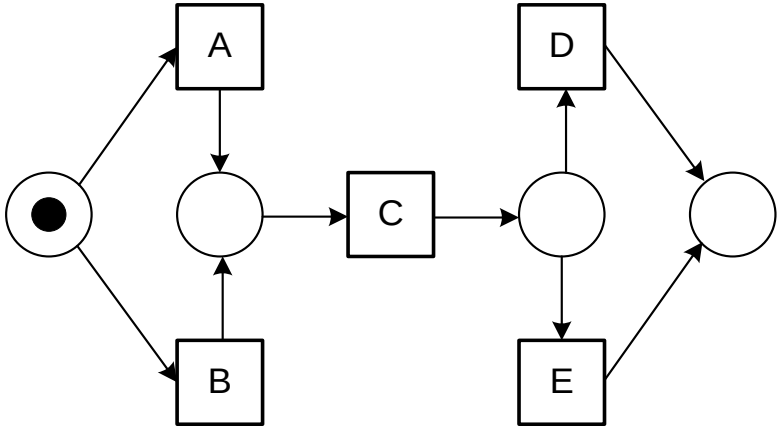
ACD	**99**
ACE	**88**
BCE	**85**
BCD	**78**





ACD	**99**
ACE	**2**
BCE	**85**
BCD	**3**

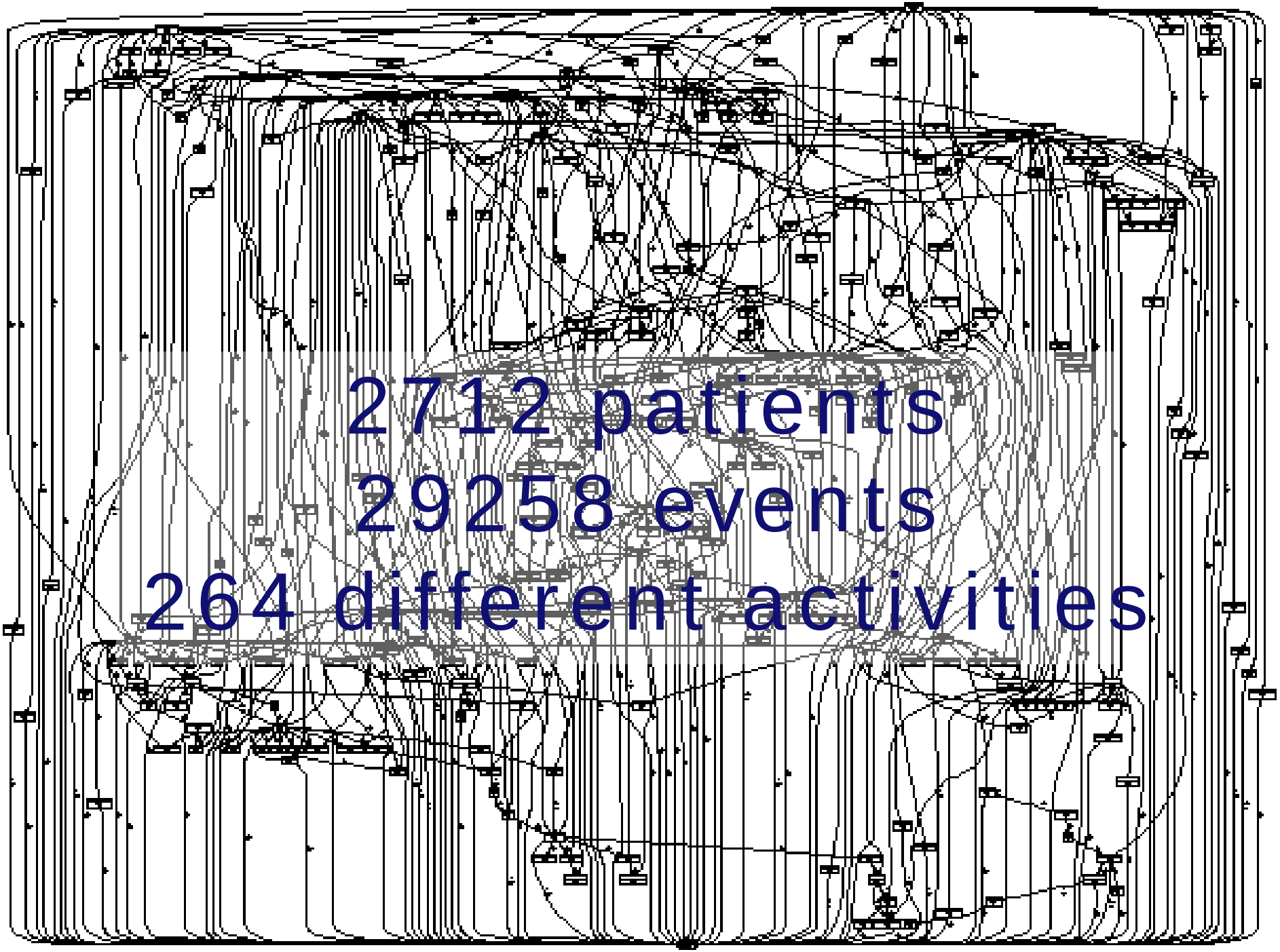


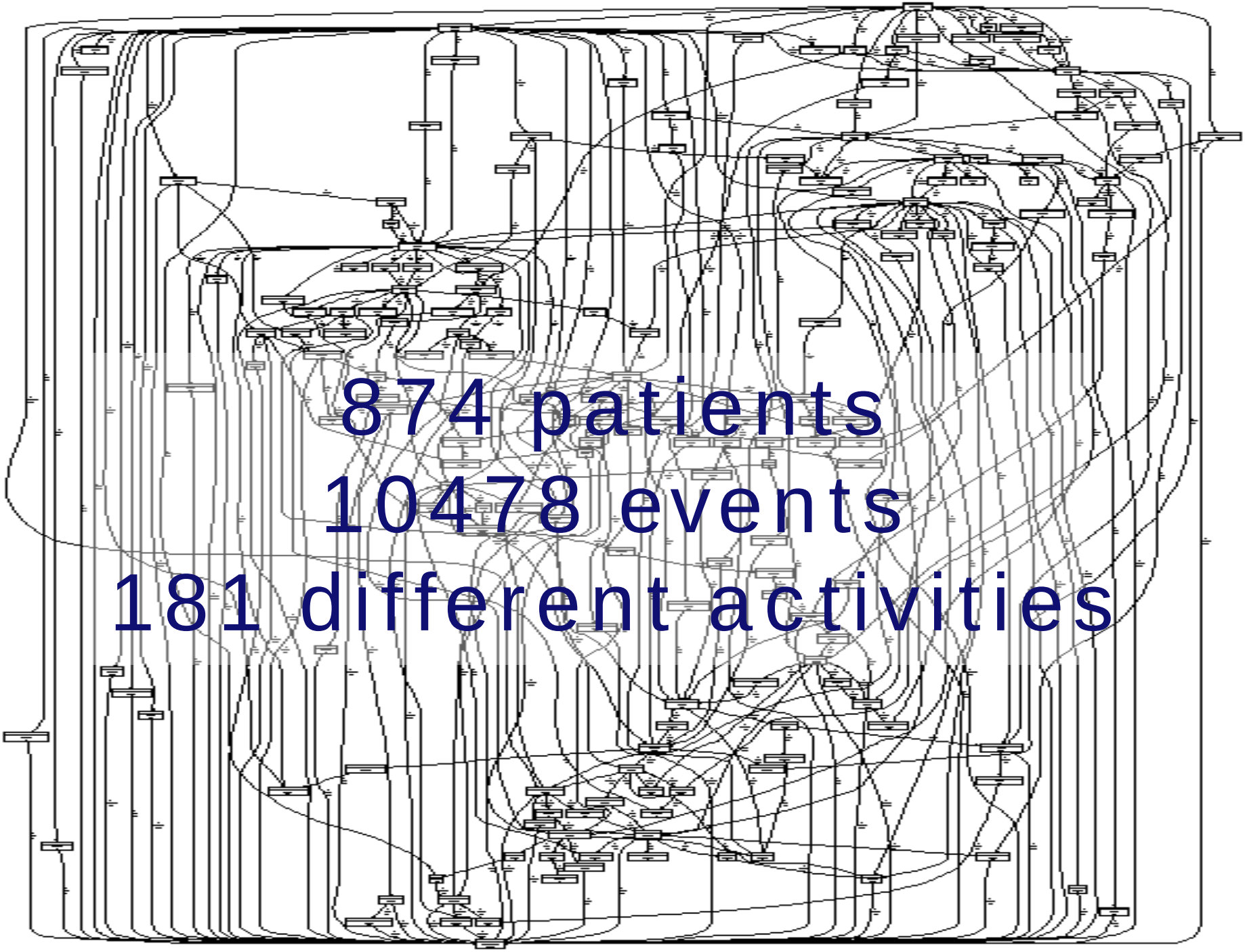


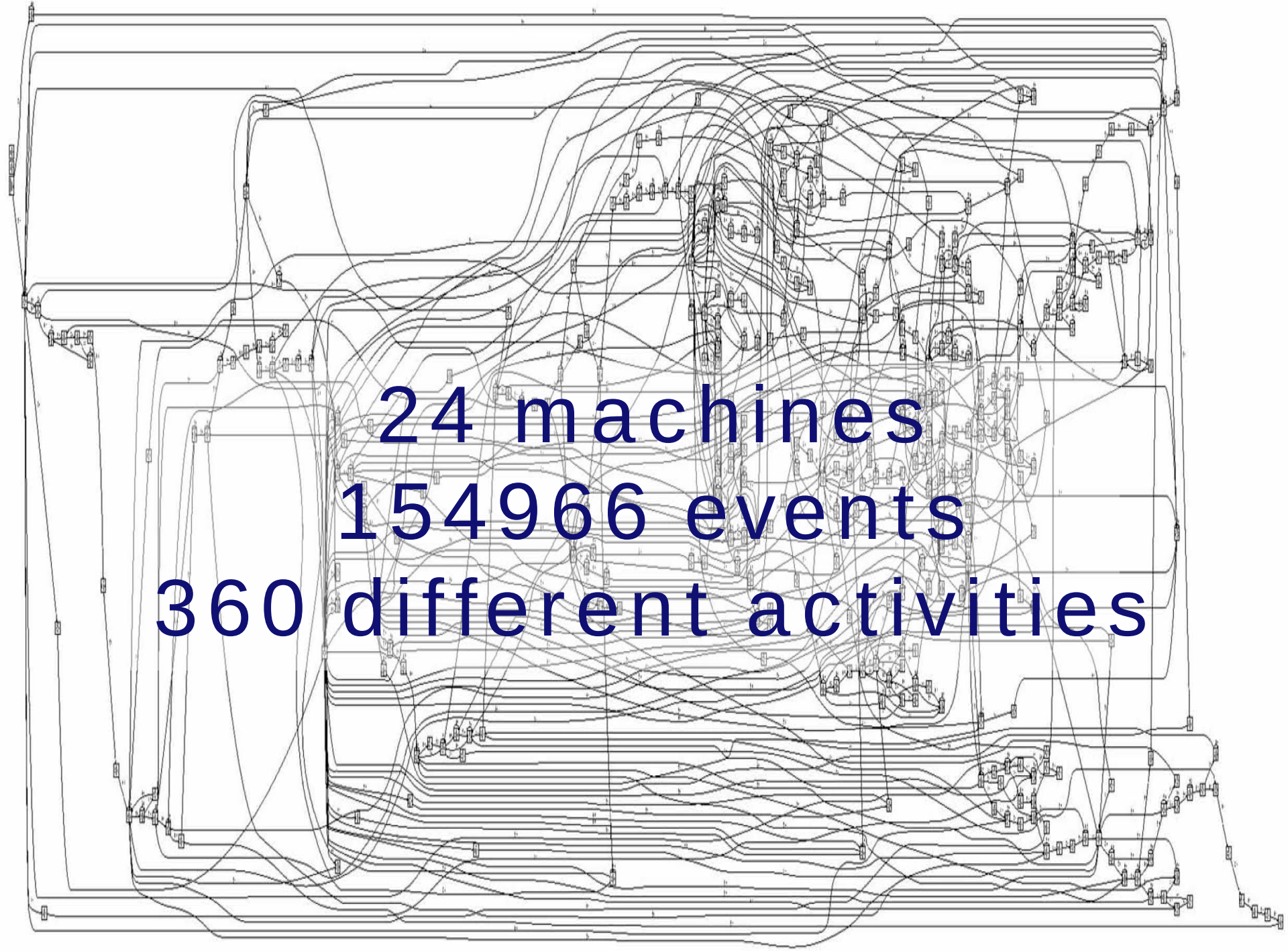
Important observations

- **Frequencies matter!**
- **Adding a place equals restricting behavior!**
- **"The model" does not exist!**

Relevance


2712 patients
29258 events
264 different activities

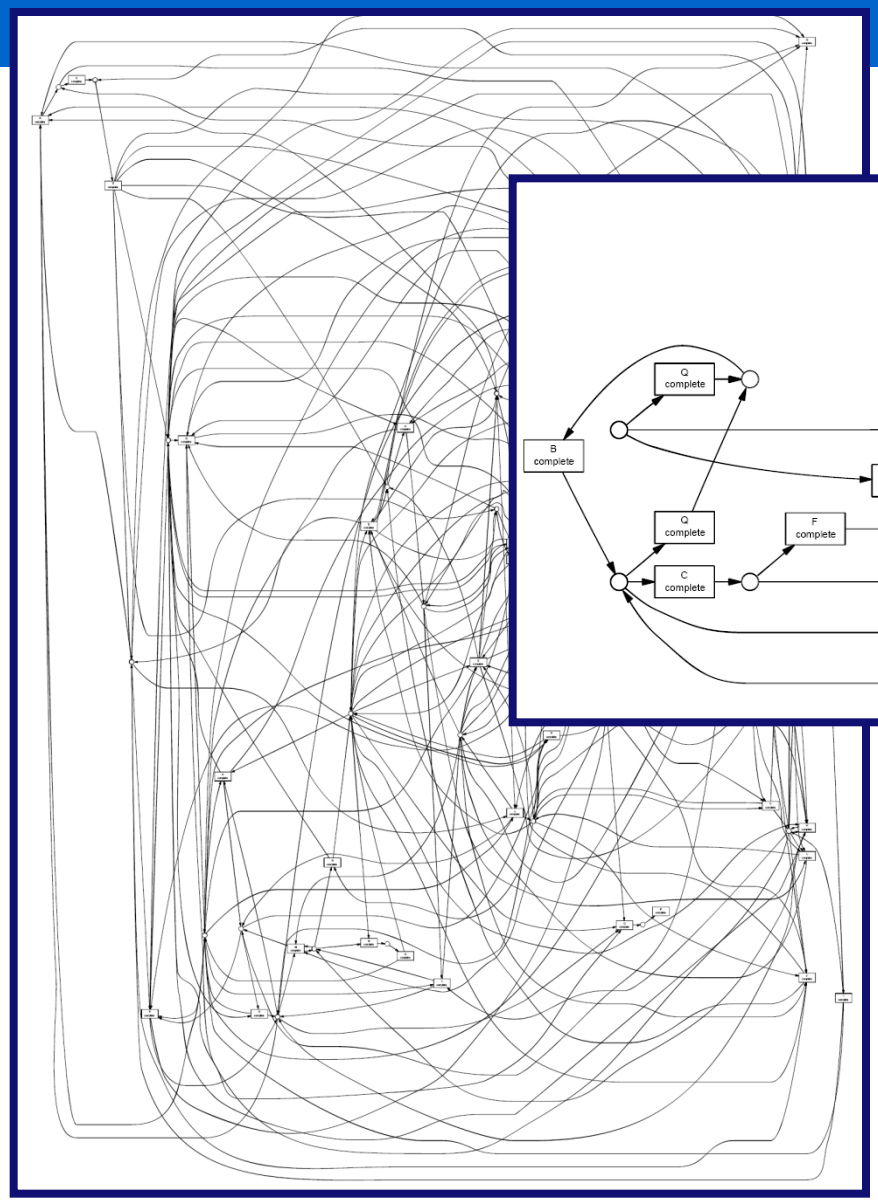

874 patients
10478 events
181 different activities

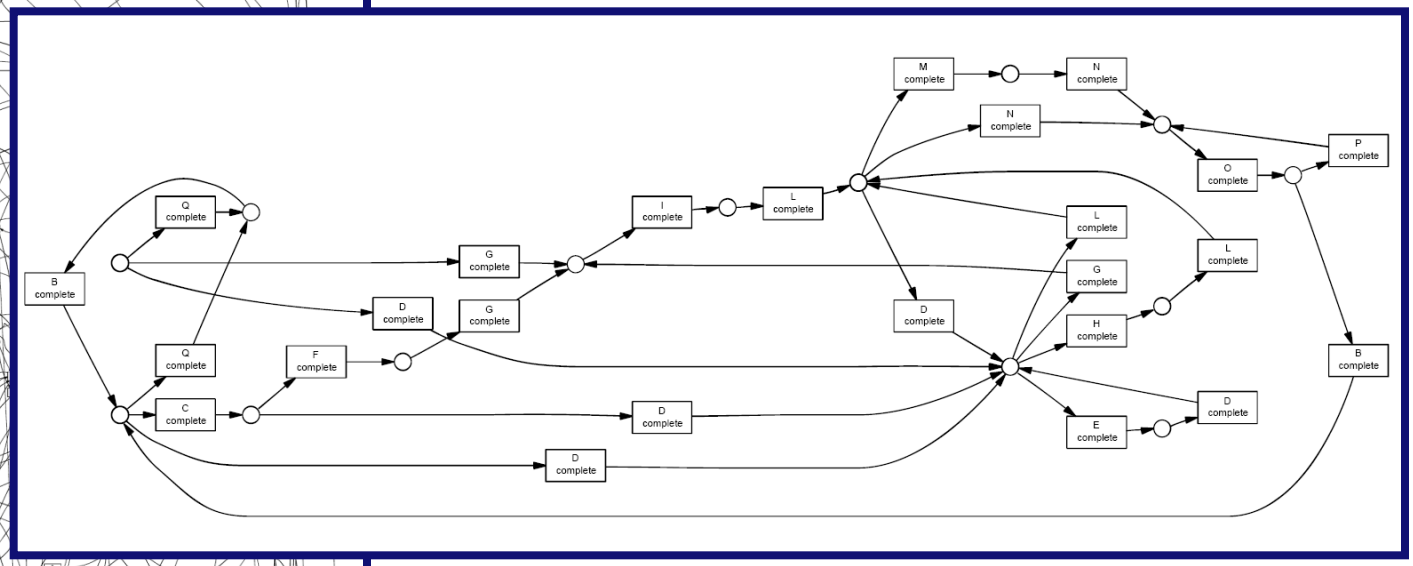

24 machines
154966 events
360 different activities

Problems

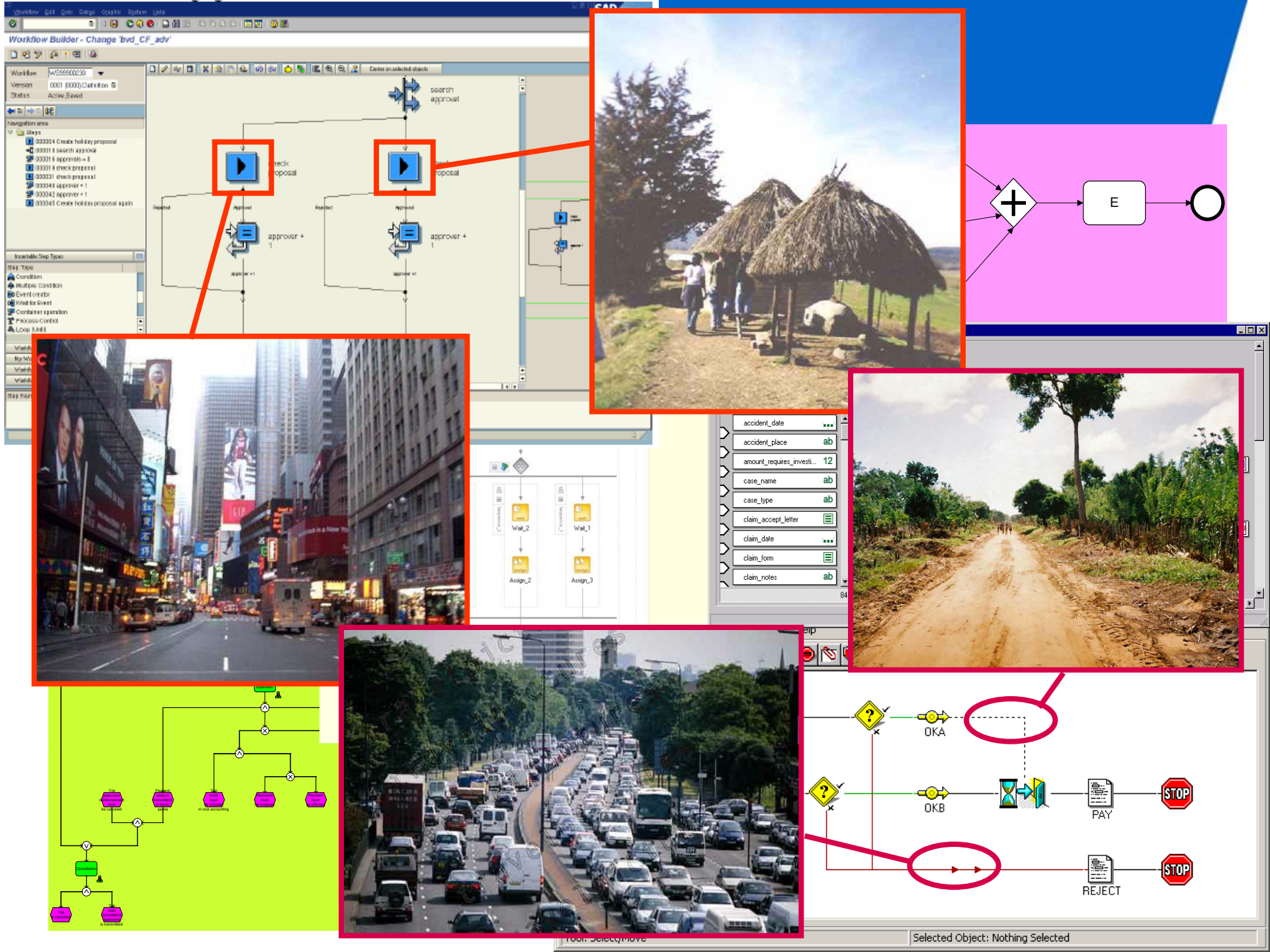
- **Representational bias (i.e., generalization is driven by representation rather than log or preferences).**
- **Inability of dealing with or detecting noise.**
- **Wrong abstraction level.**
- **Limitations of current process modeling (visualization) techniques.**

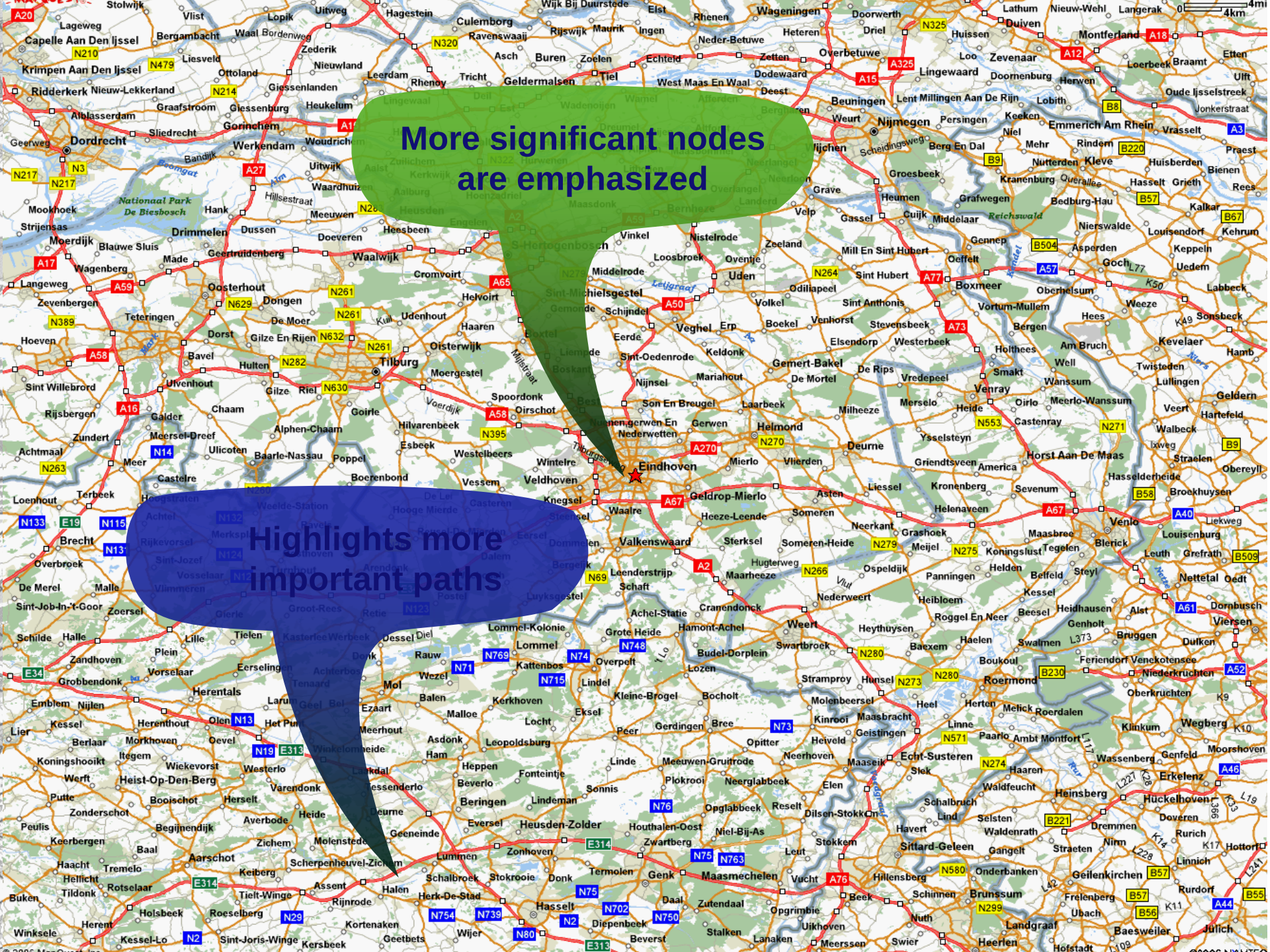
Example: Heusden





State horizon set to last activity.


Workflow Builder - Change 'brvd_CF_adv'
search approval
approver + 1
E
accident_date
accident_place
amount_requires_investi...
case_name
case_type
claim_accept_letter
claim_date
claim_form
claim_notes
Wait_2
Wait_1
Assign_2
Assign_3
OKA
OKB
PAY
REJECT
STOP
STOP
Selected Object: Nothing Selected


More significant nodes are emphasized
Highlights more important paths
Eindhoven
Tilburg
Nijmegen
Dordrecht
Hasselt
Genk
Maasmechelen
Roermond
Venlo
Helmond
Geldrop-Mierlo
Valkenswaard
Veldhoven
Waalwijk
Oosterhout
Breda
Den Bosch
Uden
Veghel
Weert
Sittard-Geleen
Heerlen
Aachen
Antwerpen
Leuven
Tienen
Mechelen
Brussel
Gent
Zwolle
Amersfoort
Utrecht
Arnhem
Nationaal Park De Biesbosch
Reichswald

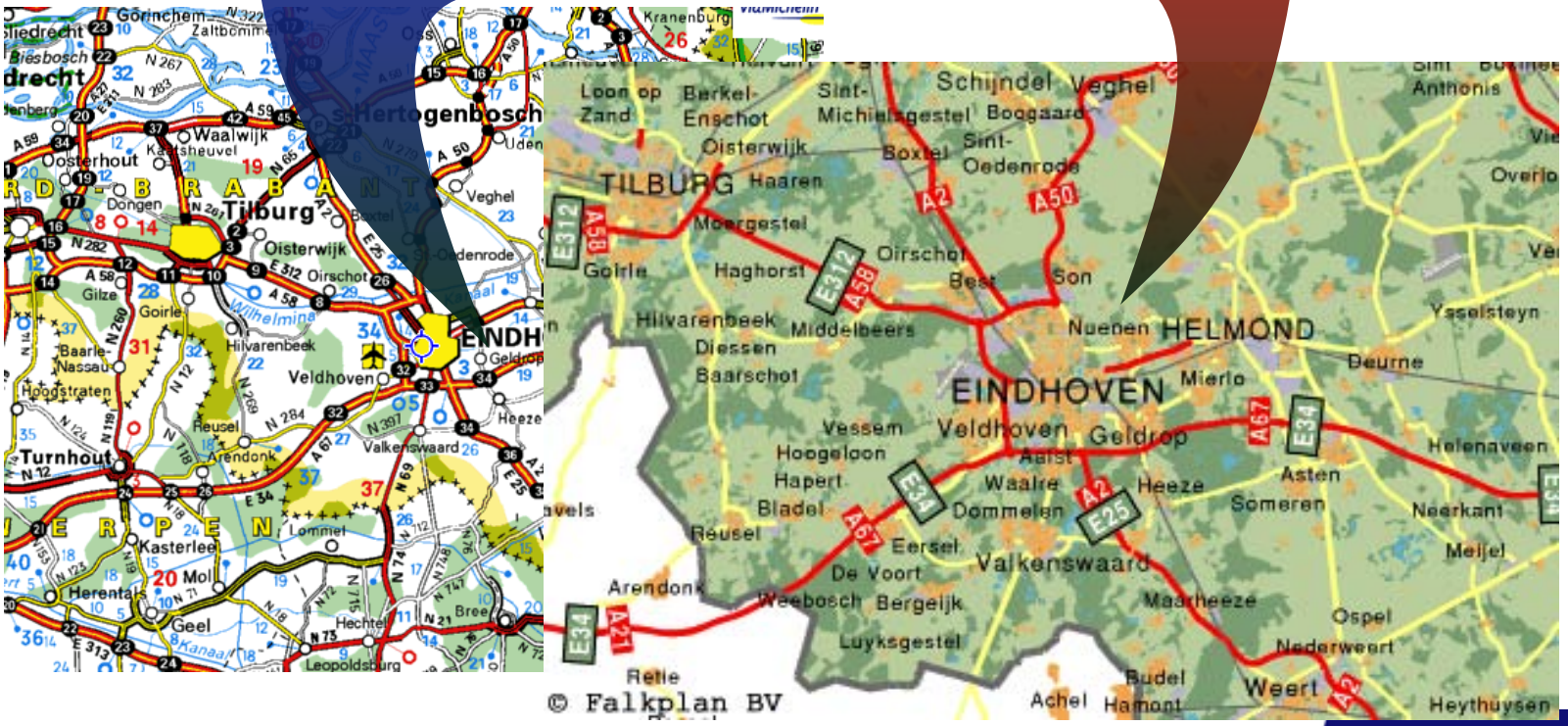
More to learn from maps...

Aggregation

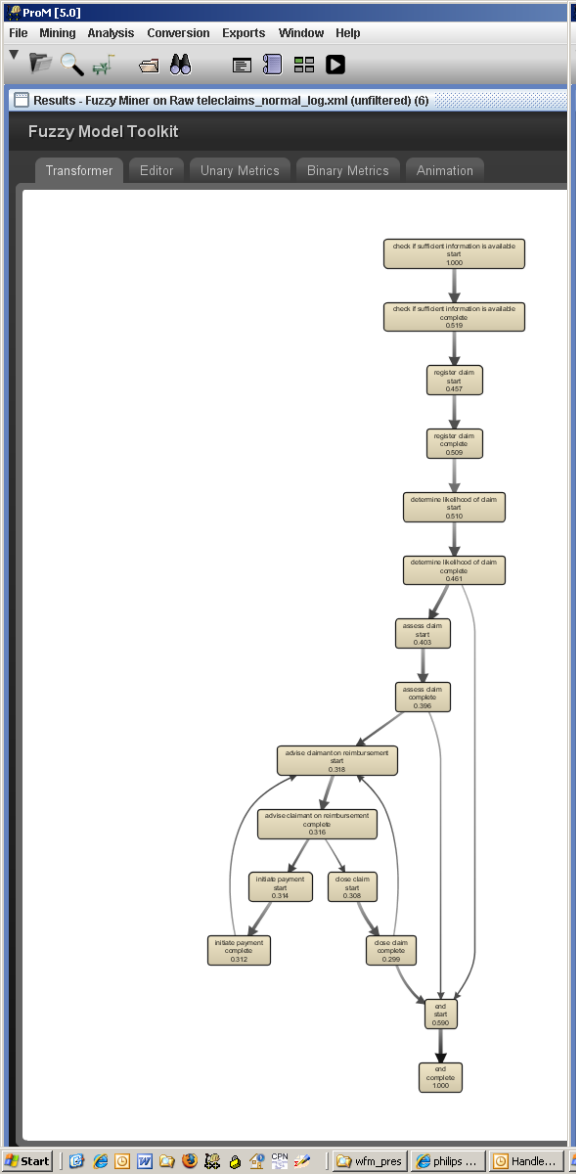
Clustering of coherent, less significant structures

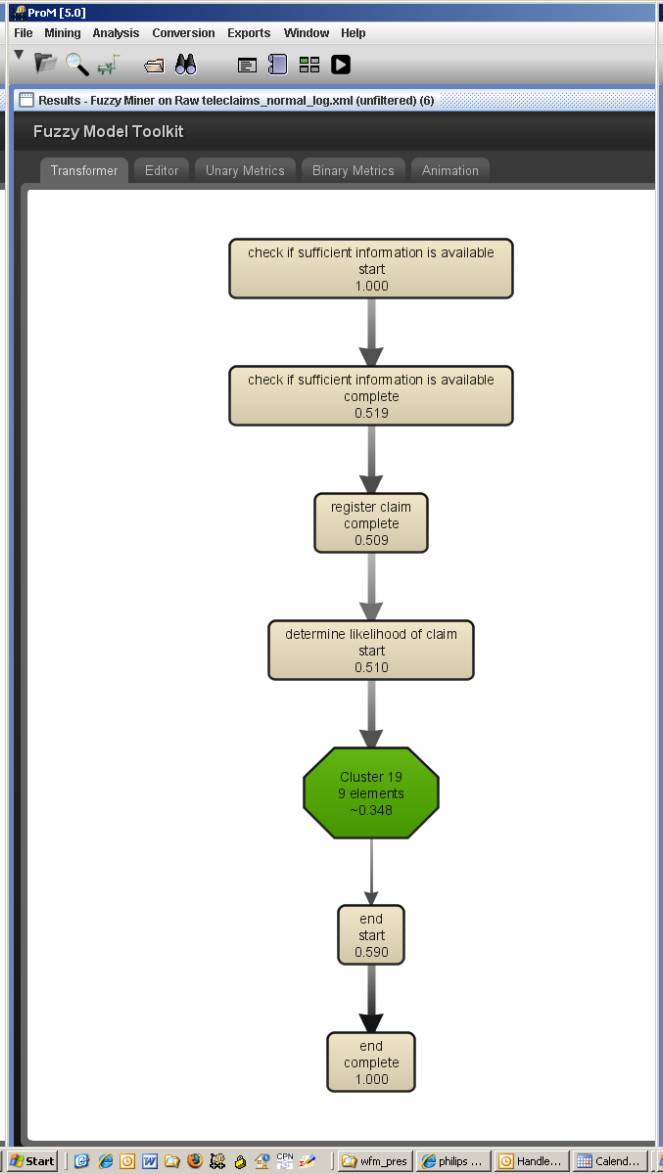
Abstraction

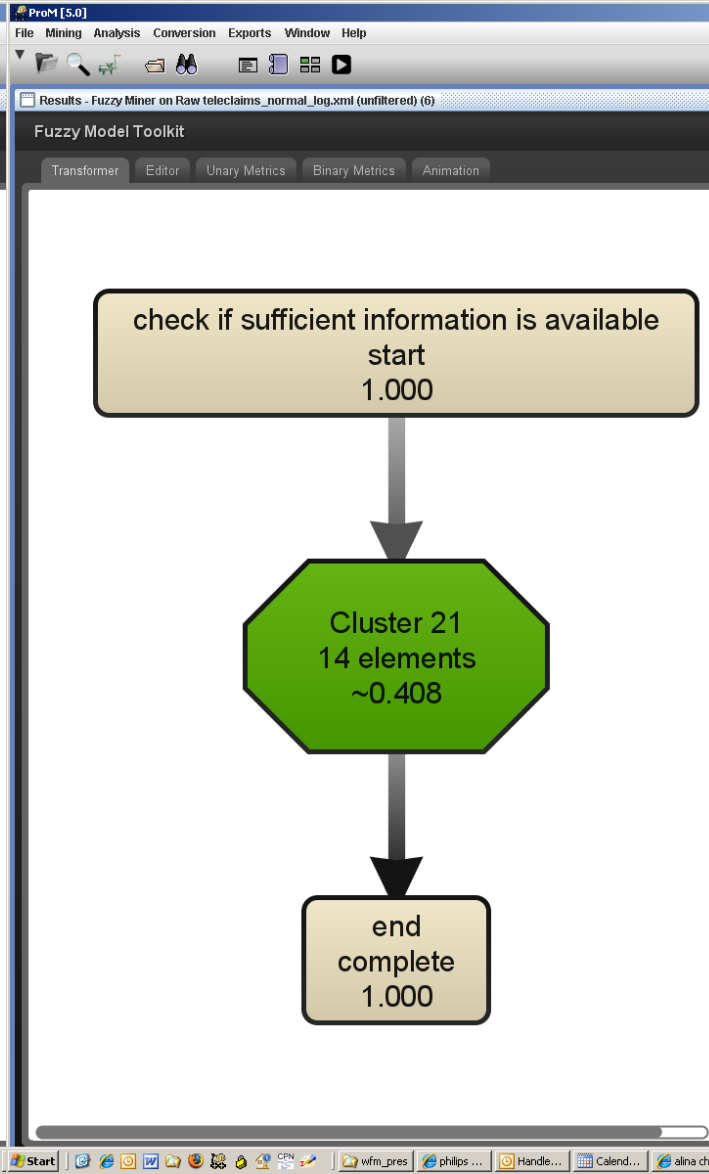
Removing isolated, less significant structures



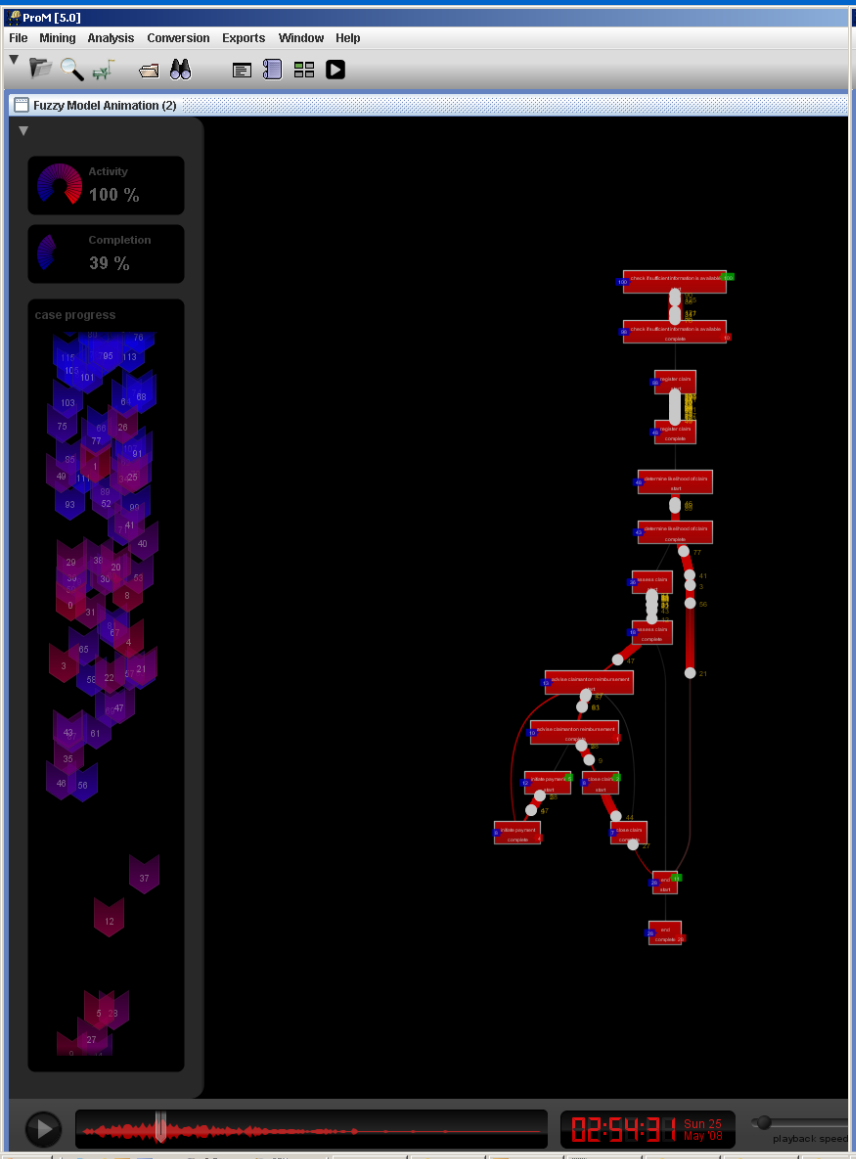
Fuzzy miner

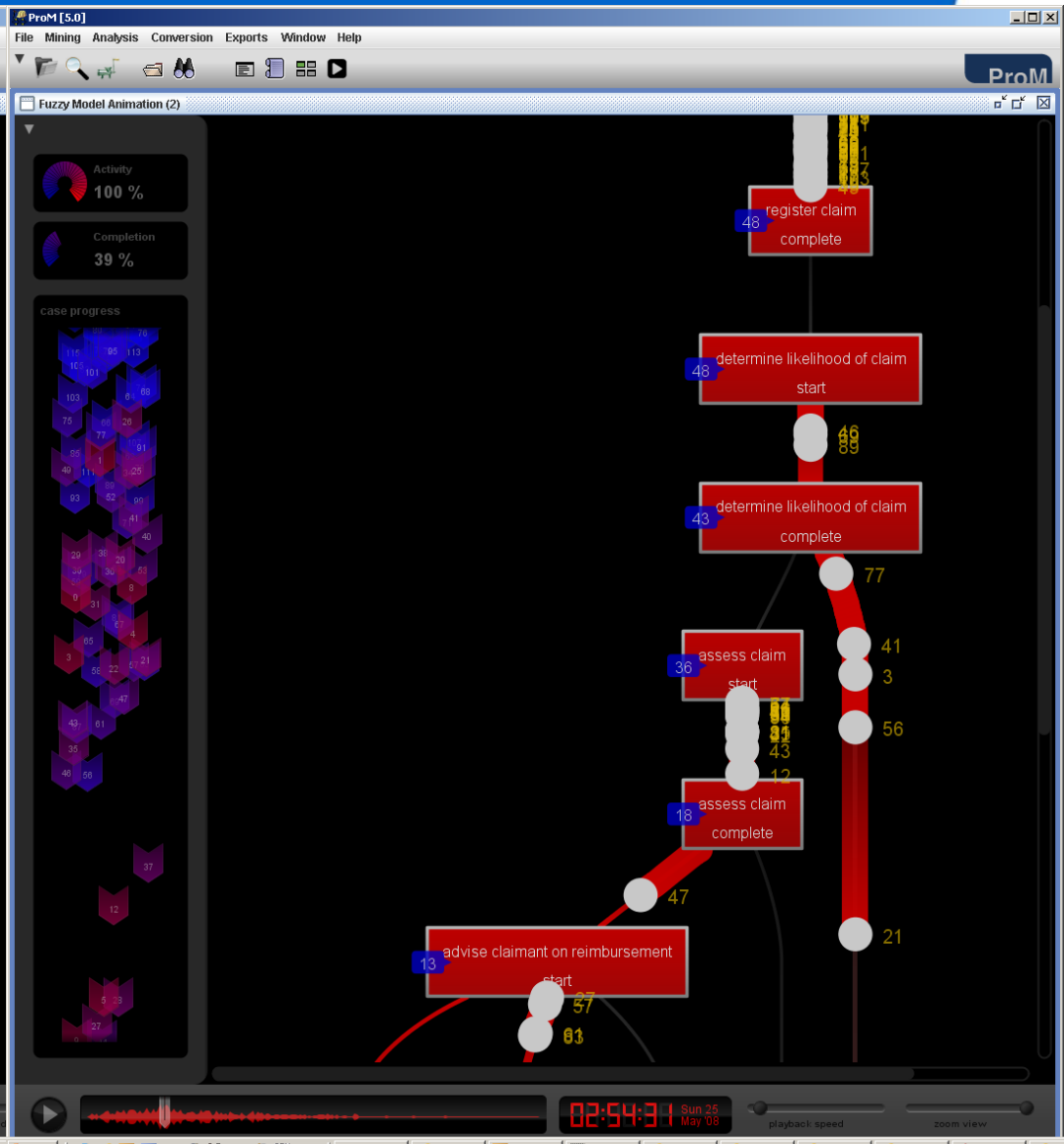

ProM [5.0]
File Mining Analysis Conversion Exports Window Help
Results - Fuzzy Miner on Raw teleclaims_normal_log.xml (unfiltered) (6)
Fuzzy Model Toolkit
Transformer Editor Unary Metrics Binary Metrics Animation


ProM [5.0]
File Mining Analysis Conversion Exports Window Help
Results - Fuzzy Miner on Raw teleclaims_normal_log.xml (unfiltered) (6)
Fuzzy Model Toolkit
Transformer Editor Unary Metrics Binary Metrics Animation
check if sufficient information is available
start
1.000
check if sufficient information is available
complete
0.519
register claim
complete
0.509
determine likelihood of claim
start
0.510
Cluster 19
9 elements
~0.348
end
start
0.590
end
complete
1.000


ProM [5.0]
File Mining Analysis Conversion Exports Window Help
Results - Fuzzy Miner on Raw teleclaims_normal_log.xml (unfiltered) (6)
Fuzzy Model Toolkit
Transformer Editor Unary Metrics Binary Metrics Animation
check if sufficient information is available
start
1.000
Cluster 21
14 elements
~0.408
end
complete
1.000

Showing reality





Relevant WWW sites

- http://www.processmining.org
- **http:// promimport.sourceforge.net**
- **http://prom.sourceforge.net**
- **http://www.workflowpatterns.com**
- **http://www.workflowcourse.com**
- **http://www.vdaalst.com**

Exercises Process Discovery

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Exercise A1

- **From some transactional system the following event log consisting of six traces is extracted:**
  - ***a b e f***
  - ***a b e c d b f***
  - ***a b c e d b f***
  - ***a b c d e b f***
  - ***a e b c d b f***
- **Use the 8 steps of the alpha-algorithm (included on last page) to construct the corresponding Petri-net and draw the Petri-net (delivering all of the intermediate results is not necessary, only the resulting Petri-net is required).**
- **Give a trace possible according to the discovered model but not (yet) observed in the log.**

Exercise A2

- **From some transactional system the following event log consisting of six traces is extracted:**
  - ***a b c d f***
  - ***a c b d f***
  - ***a b d c f***
  - ***a c d b f***
  - ***a d e f***
  - ***a e d f***
- **Use the 8 steps of the alpha-algorithm (included on last page) to construct the corresponding Petri-net and draw the Petri-net (delivering all of the intermediate results is not necessary, only the resulting Petri-net is required).**
- **Give a trace possible according to the discovered model but not (yet) observed in the log.**

Exercise A3

- **From some transactional system the following event log consisting of six traces is extracted:**
  - ***a d e f h***
  - ***a e d f h***
  - ***g h***
  - ***a b c d f h***
  - ***a c b d f h***
  - ***a b d c f h***
  - ***a c d b f h***
- **Use the 8 steps of the alpha-algorithm (included on last page) to construct the corresponding Petri-net and draw the Petri-net (delivering all of the intermediate results is not necessary, only the resulting Petri-net is required).**
- **Give a trace possible according to the discovered model but not (yet) observed in the log.**

Hands-on with Prom

Technische Universiteit
Eindhoven
University of Technology

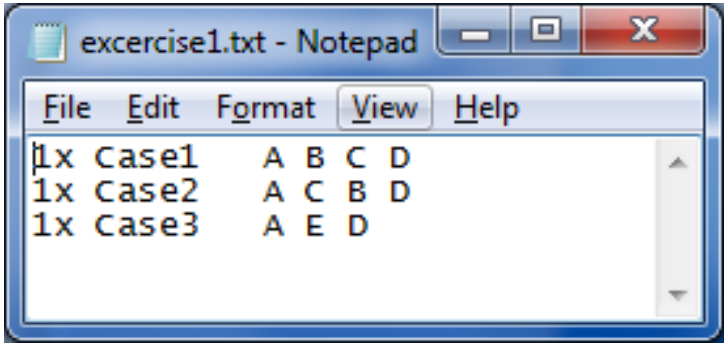
Where innovation starts

Download/install ProM 5.2

The software can be downloaded from http://prom.win.tue.nl/tools/prom/.
(More information can be found at www.processmining.org and in case of problems one can try http://prom.win.tue.nl/tools/prom/nightly5/ instead.)
To make the exercises use the files in http://wwwis.win.tue.nl/~wvdaalst/ProM-examples.zip

Get files

- **Load file ProM-examples.zip and unzip.**
- **Note the files txt and mxml files.**




```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!-- MXML version 1.1 -->
<!-- Created by ProM Import Framework, Version 5.0 (Attaboy) -->
<!-- via MXMLib Version 1.9 (http://promimport.sf.net/) -->
<!-- (c) 2004-2007 C.W. Guenther (christian@deckfour.org); Eindhoven Technical Universit
<!-- This event log is formatted in MXML, for use by BPI and Process Mining Tools. -->
<!-- You can load this file e.g. in the ProM Framework for Process Mining. -->
<!-- More information about MXML, Process Mining, and ProM: http://www.processmining.org
<WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchema
    <Data>
        <Attribute name="app.name">ProM Import Framework</Attribute>
        <Attribute name="app.version">5.0 (Attaboy)</Attribute>
        <Attribute name="java.vendor">Sun Microsystems Inc.</Attribute>
        <Attribute name="java.version">1.6.0_07</Attribute>
        <Attribute name="mxml.creator">MXMLib (http://promimport.sf.net/)</Attribute>
        <Attribute name="mxml.version">1.1</Attribute>
        <Attribute name="os.arch">x86</Attribute>
        <Attribute name="os.name">Windows XP</Attribute>
        <Attribute name="os.version">5.1</Attribute>
        <Attribute name="user.name">wvdaalst</Attribute>
    </Data>
    <Source program="Rapid Synthesizer"/>
    <Process id="example-alpha-1">
        <ProcessInstance id="Case3.0">
            <AuditTrailEntry>
                <WorkflowModelElement>A</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2008-12-09T08:20:01.527+01:00</Timestamp>
                <Originator>UNDEFINED</Originator>
            </AuditTrailEntry>
            <AuditTrailEntry>
                <WorkflowModelElement>E</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2008-12-09T08:21:01.527+01:00</Timestamp>
                <Originator>UNDEFINED</Originator>
            </AuditTrailEntry>
            <AuditTrailEntry>
                <WorkflowModelElement>D</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2008-12-09T08:22:01.527+01:00</Timestamp>
                <Originator>UNDEFINED</Originator>
            </AuditTrailEntry>
        </ProcessInstance>
        <ProcessInstance id="Case2.0">
            <AuditTrailEntry>
                <WorkflowModelElement>A</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2008-12-09T08:20:01.527+01:00</Timestamp>
                <Originator>UNDEFINED</Originator>
            </AuditTrailEntry>
            <AuditTrailEntry>
                <WorkflowModelElement>C</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2008-12-09T08:21:01.527+01:00</Timestamp>
                <Originator>UNDEFINED</Originator>
            </AuditTrailEntry>
            <AuditTrailEntry>
                <WorkflowModelElement>B</WorkflowModelElement>
                <EventType>complete</EventType>
                <Timestamp>2008-12-09T08:22:01.527+01:00</Timestamp>
                <Originator>UNDEFINED</Originator>
            </AuditTrailEntry>
            <AuditTrailEntry>
```

Redo exercises 1-3 (see slides)

- **Inspect the files**
  - **exercise1.txt/exercise1.mxml,**
  - **exercise2.txt/exercise2.mxml, and**
  - **exercise3.txt/exercise3.mxml.**
- **Redo exercises using ProM 5.2.**

Exercise (4)

- **Consider the following log:**
  - **a b c d f**
  - **a c b d f**
  - **a b d c f**
  - **a c d b f**
  - **a d e f**
  - **a e d f**
- **Use the Alpha algorithm/ProM to discover the corresponding process model (Filename: exercise4.mxml)**

Exercise (5)

- **Consider the following log:**
  - **a b c d e f b d c e g**
  - **a b d c e g**
  - **a b c d e f b c d e f b d c e g**
- **Use the Alpha algorithm/ProM to discover the corresponding process model (Filename: exercise5.mxml)**

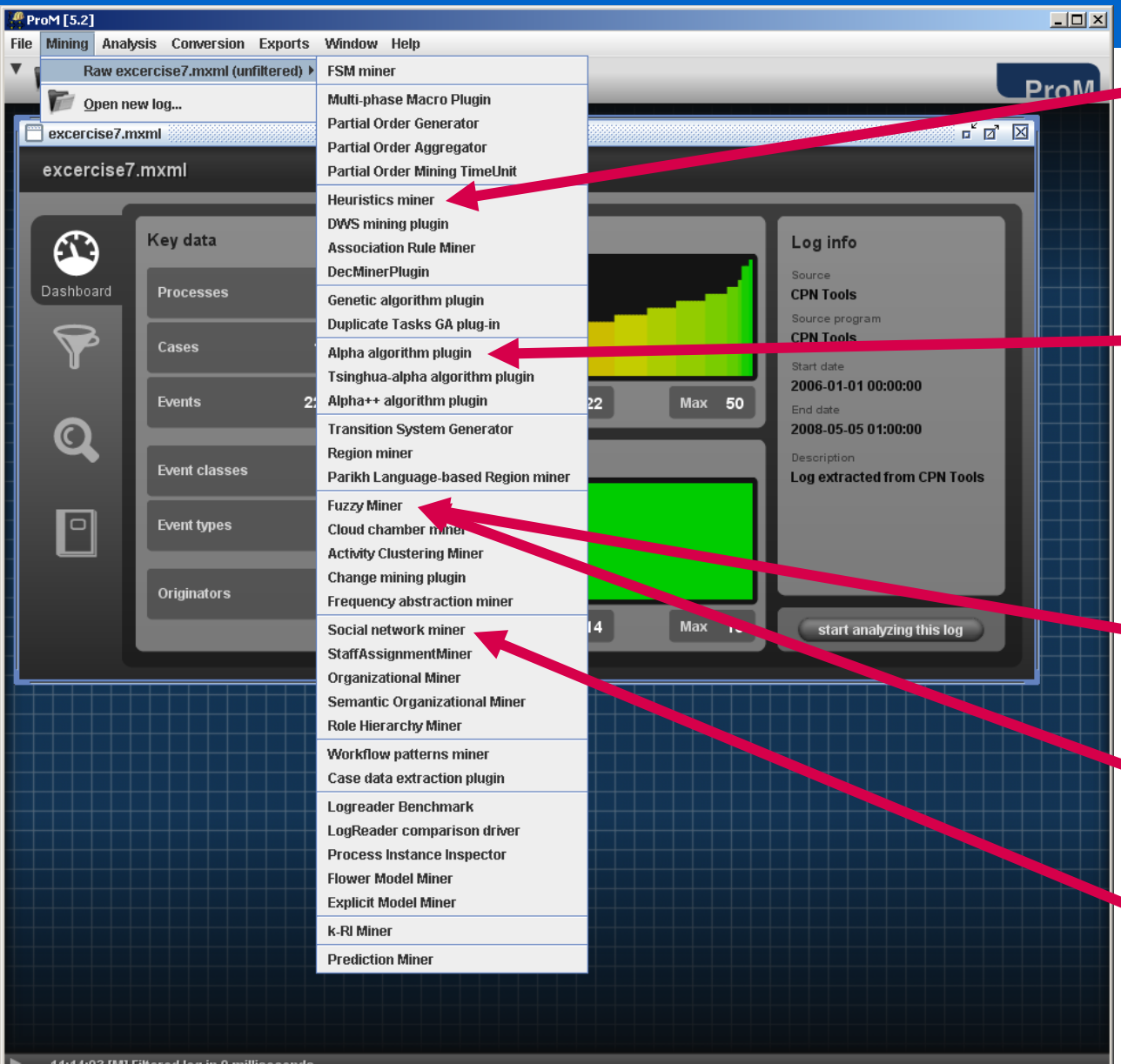
Exercise (6)

- **Consider the following log:**
  - **a b e f**
  - **a b e c d b f**
  - **a b c e d b f**
  - **a b c d e b f**
  - **a e b c d b f**
- **Use the Alpha algorithm/ProM to discover the corresponding process model (Filename: exercise6.mxml)**

Exercise (7)

- **Load exercise7.mxml**
- **Inspect the log (e.g. using the internet explorer)**
- **Use the Alpha algorithm to discover the corresponding process model.**

Explore ProM using exercise7.mxml



[Heuristics miner]
discover process model and play with settings

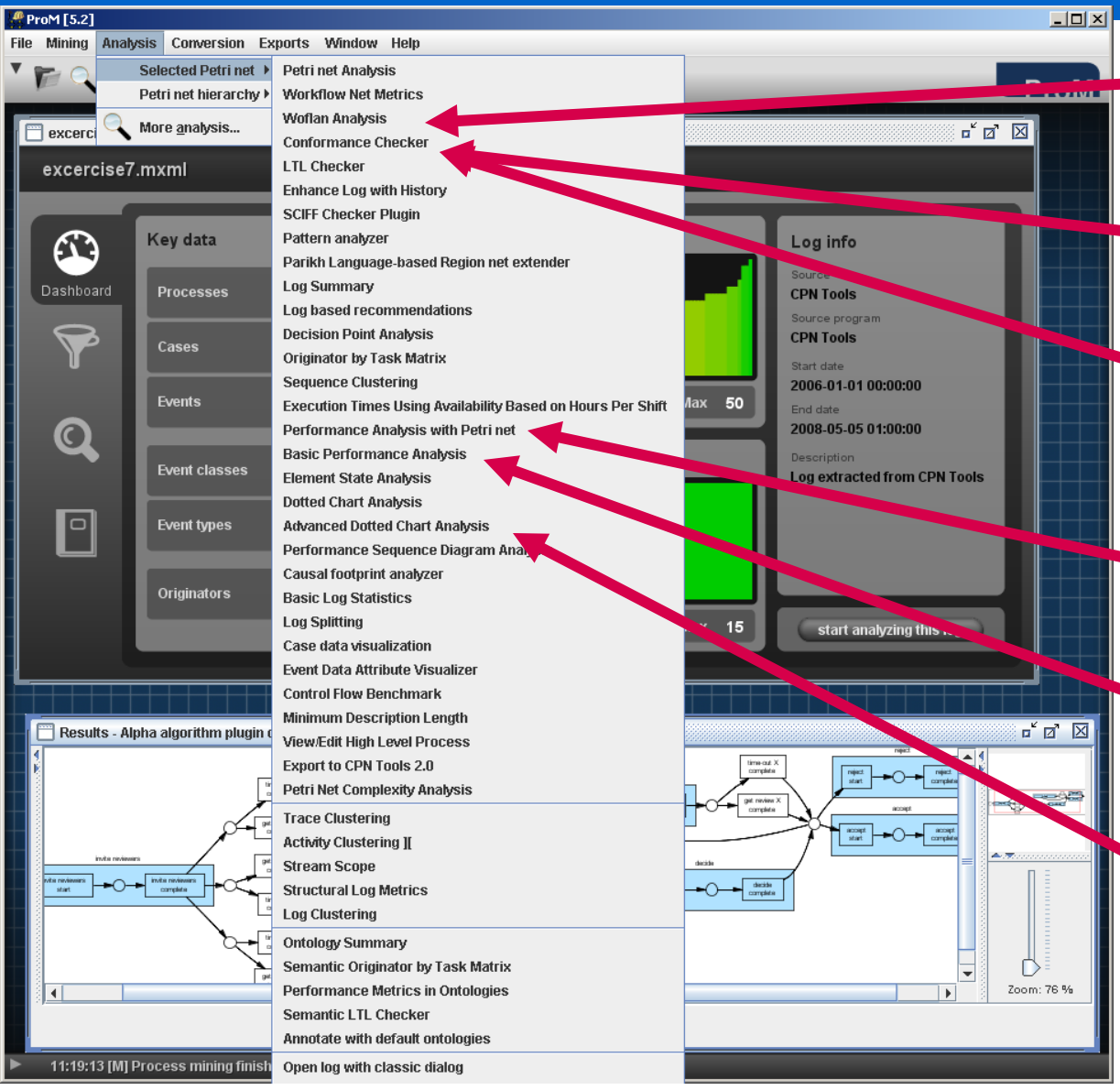
[Alpha algorithm plugin]
discover process model and play with settings

[Fuzzy Miner]
discover process model and explore the various views

[Fuzzy Miner]
animate the model

[Social Network Miner]
discover the social network

Explore ProM using exercise7.mxml (2)



[Woflan Analysis]
verify correctness of model

[Conformance Checker] check the conformance of the mined model

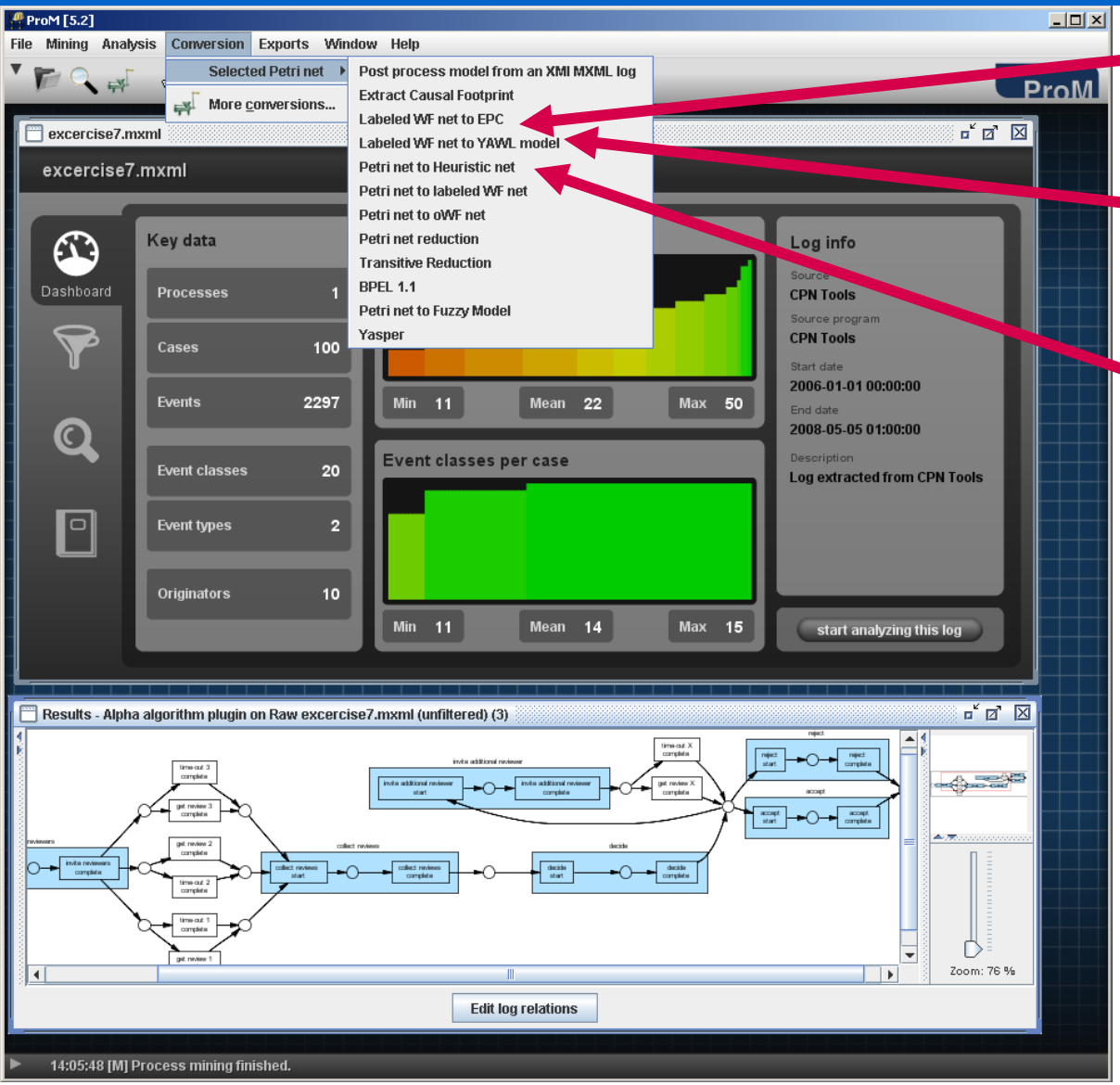
[Conformance Checker] modify the log (delete and insert events) and the check the conformance

[Performance Analysis with Petri net]
analyze the performance (where are the bottlenecks)

[Basic Performance Analysis] analyze the performance (explore all options)

[(Advanced) Dotted Chart Analysis]
construct dotted charts and explore all options

Explore ProM using exercise7.mxml (3)



convert discovered Petri net into EPC model

convert discovered Petri net into YAWL model

convert discovered Petri net into heuristic net

If time is left, ...

- **Repeat the process using exercise8.mxml**
- **Repeat the process using repairexample.zip and repairexamplesample2.zip**
- **Use the above two files to follow the steps described in the ProM Framework Tutorial**