Workflow : *Patterns and Specifications*

Seminar Presentation
by

Ahana Pradhan

Under the guidance of

Prof. Rushikesh K. Joshi

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
April 30, 2012

Outline

- Workflow concepts
- Petri nets & workflow process
- Workflow patterns
- Petri nets & workflow patterns
- YAWL & workflow patterns
- BPMN & workflow patterns

Introduction

- **Business Process**
  Sequence of activities to serve a purpose.
  e.g. Process of railway ticket reservation.

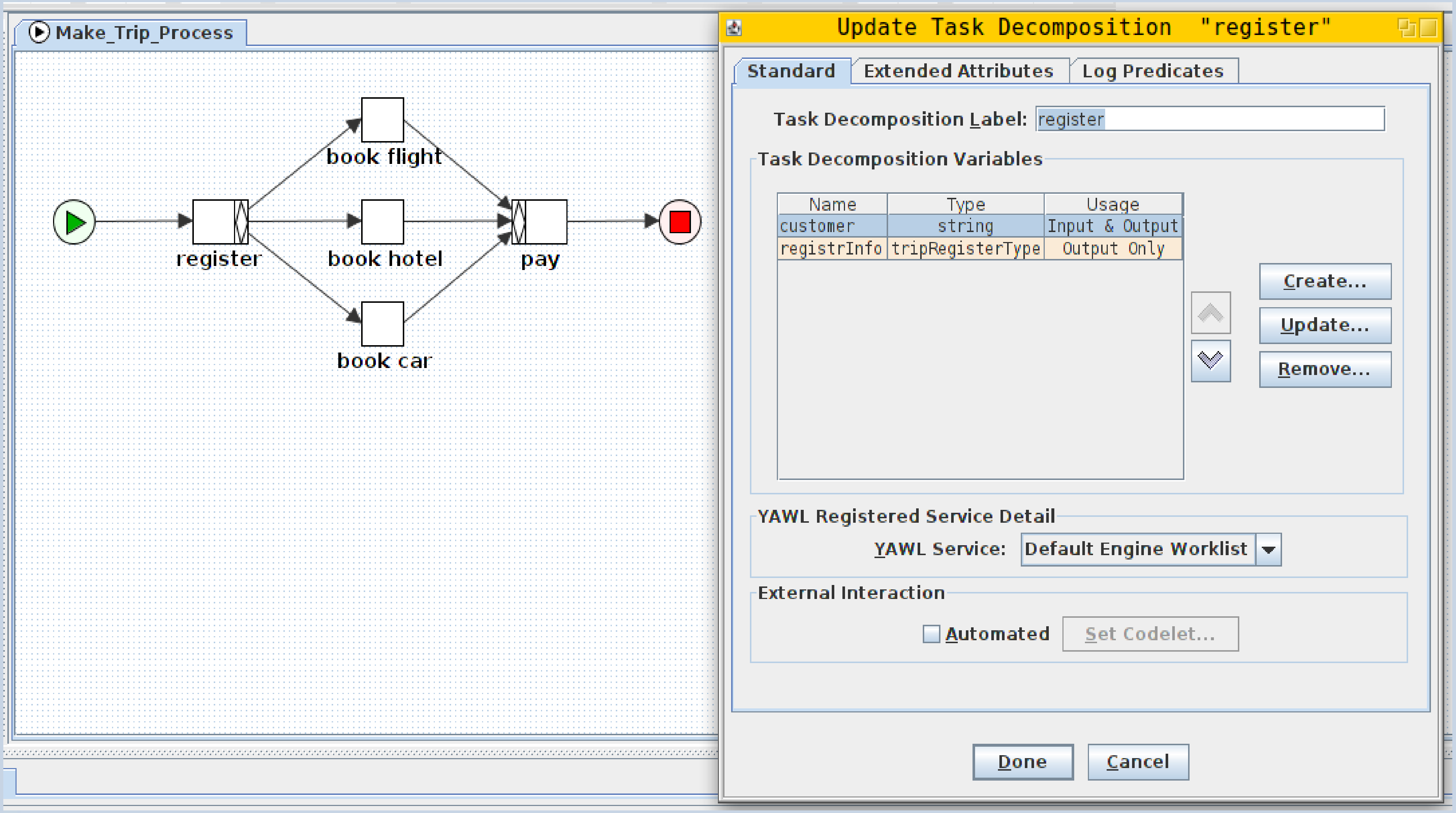
- **Workflow Process**
  Flow of work in a business process for a specific case.

- **Modeling and specifying processes**
  i.e. creating workflow definition.

  -- May be using graphical notation based languages.
  e.g. BPMN diagram.

  -- May be using XML based languages
  e.g. BPEL specification, YAWL specification.

Workflow definition example in YAWL



Task Decomposition Label: register

Task Decomposition Variables

Name	Type	Usage
customer	string	Input & Output
registrInfo	tripRegisterType	Output Only

YAWL Registered Service Detail

YAWL Service: Default Engine Worklist

External Interaction

☐ Automated

Workflow definition example in YAWL

```
...
<task id="register_3">
     <name>register</name>
     <flowsInto>
       <nextElementRef id="book_flight_8" />
       <predicate>/Make_Trip_Process/registrInfo/want_flight='true'</predicate>
     </flowsInto>
     <flowsInto>
       <nextElementRef id="book_car_10" />
       <predicate>/Make_Trip_Process/registrInfo/want_car='true'</predicate>
       <isDefaultFlow />
     </flowsInto>
     <flowsInto>
       <nextElementRef id="book_hotel_9" />
       <predicate>/Make_Trip_Process/registrInfo/want_hotel='true'</predicate>
     </flowsInto>
     <join code="xor" />
     <split code="or" />
     <startingMappings>
       <mapping>
         <expression query="&lt;customer&gt;{/Make_Trip_Process/customer/text()}&lt;/customer&gt;" />
         <mapsTo>customer</mapsTo>
       </mapping>
     </startingMappings>
     <completedMappings>
       <mapping>
         <expression query="&lt;registrInfo&gt;{/register/registrInfo/*}&lt;/registrInfo&gt;" />
         <mapsTo>registrInfo</mapsTo>
       </mapping>
       <mapping>
         <expression query="&lt;customer&gt;{/register/customer/text()}&lt;/customer&gt;" />
         <mapsTo>customer</mapsTo>
       </mapping>
     </completedMappings>
     <resourcing>
       <offer initiator="user" />
       <allocate initiator="user" />
       <start initiator="user" />
     </resourcing>
     <decomposesTo id="register" />
    </task>
    ...
```

Workflow Terminologies

- **Task**

  Each atomic work to be done in a workflow definition is task.

- **Work Item**

  When a task is assigned to some resource it is then a work item.

- **Activity**

  When a work item is being executed by a resource in a workflow process, it is an activity.

- **Trigger**

  It is some external environmental condition that makes an activity started.

Workflow languages

- **Workflow language** is *XML* based notations that is used to describe inter-task dependencies in a workflow process i.e. create workflow specifications.

- **Workflow language formalism** is for expressing the dependencies of tasks. It may be graphical representation.

- *Petri net* is a very well-known workflow language formalism because :

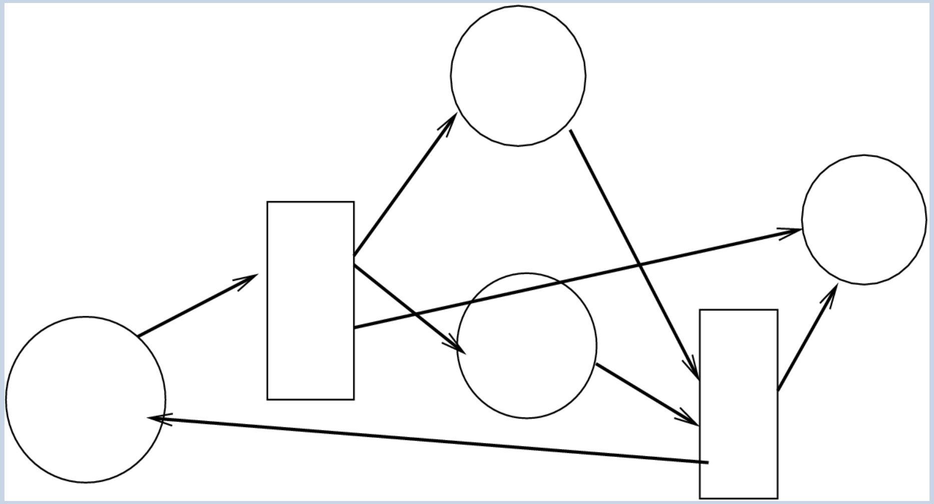
  -- Intuitive graphical representation

  -- Formal theory provides convenient base for analysis

  -- Abundance of analysis tenchniques

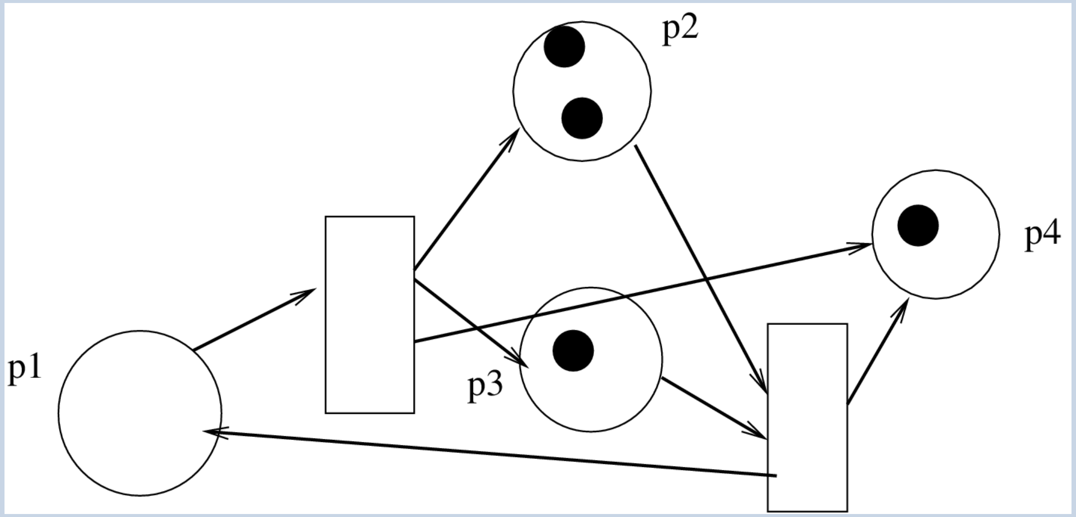
Petri Net

- A graph having circles and rectangles as nodes
- *Directed bipartite* graph
  - Edges are directed
  - Edge can be between either rectangle to circle or between circle to rectangle
- Circles are called ***place***
- Rectangles are called ***transition***
- Edges are called *arc/flow relation*

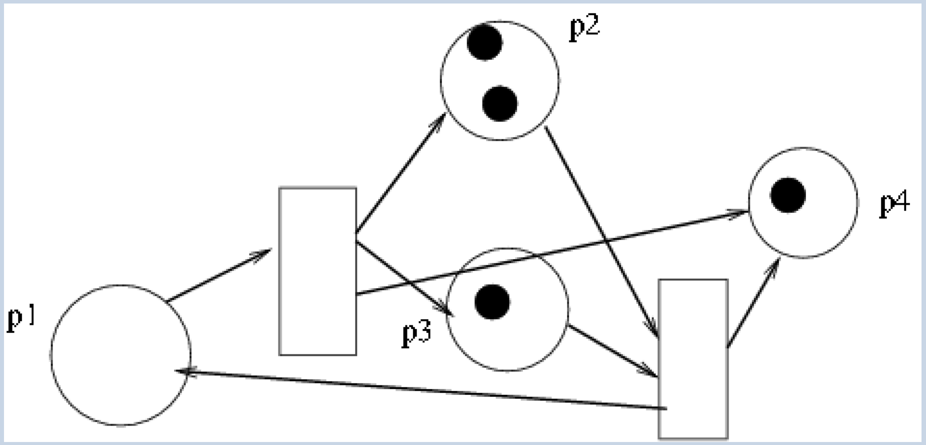


Token

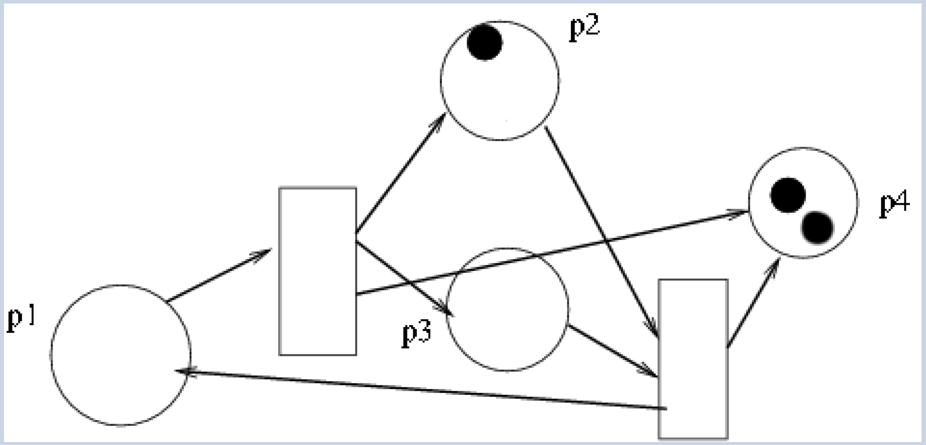
- *Dot* inside a *place*
- Theoritical concept
- Used to represent ***state*** of the net



State transition by token firing



State transition by token firing

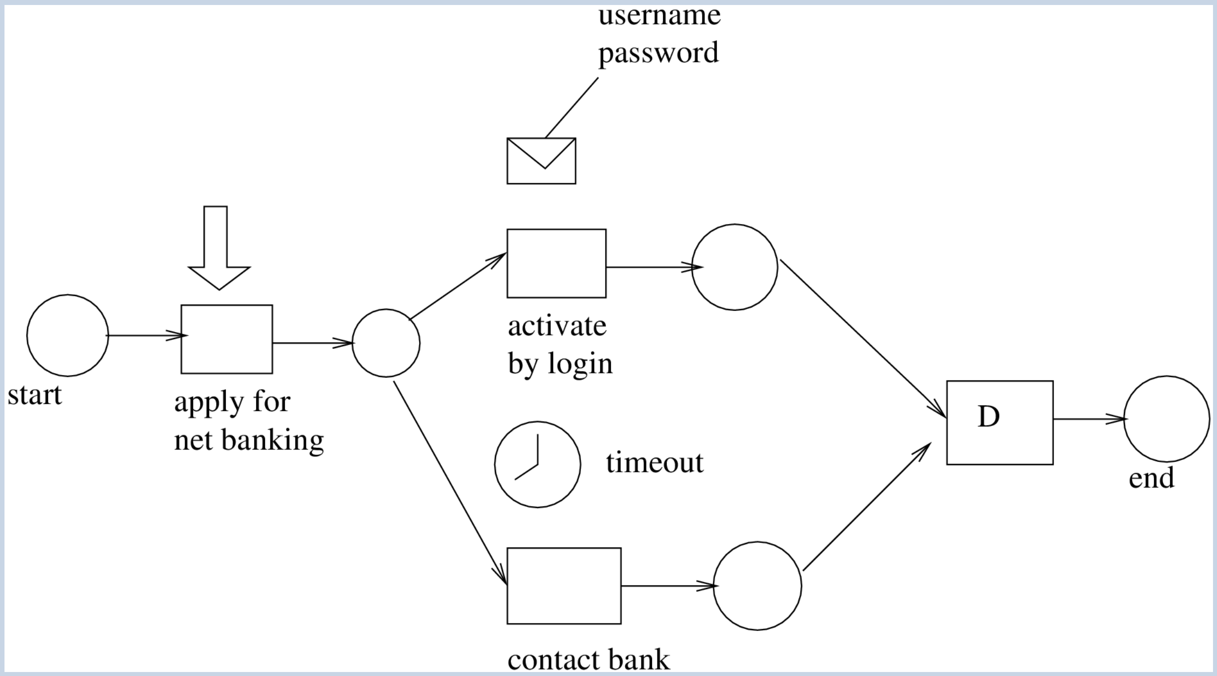


Petri net as workflow language

- ***Place*** is ***condition***
- ***Transition*** is ***task***
- ***Token*** corresponds to ***case***
- A petri net is a valid *workflow net* (***WF-net***) if and only if the following criterias are satisfied:
  1. The net has one source place i.e. No transitions have it as output place.
  2. The net has one sink place i.e. No transitions have it as input place.
  3. If a transition is added to the net from sink place to source place, the net will be strongly connected.

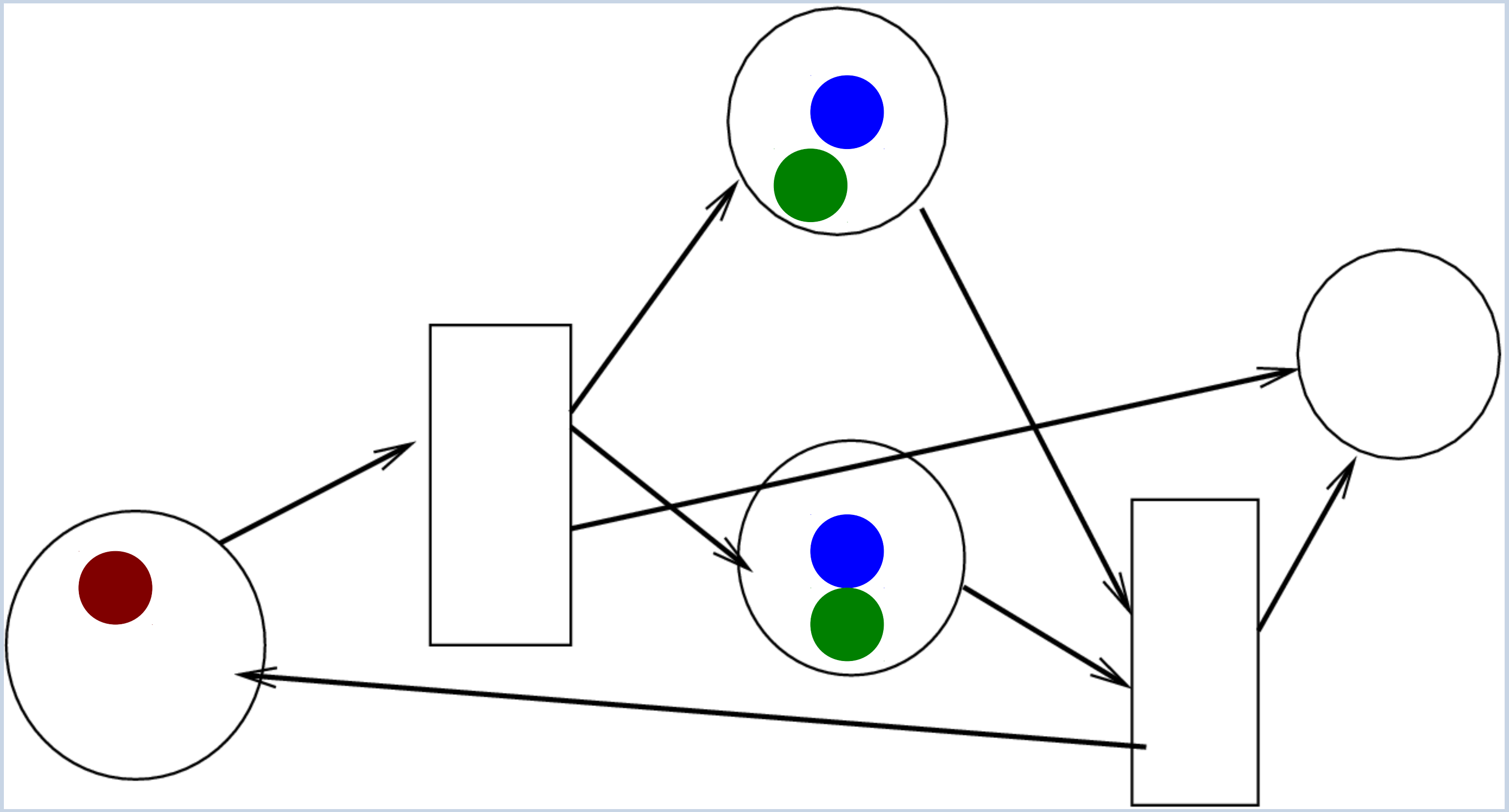
Petri net as workflow language

- Petri nets do not have any abstraction that can map to triggers of workflow task. However, additional icons of triggers can be used to specify trigger



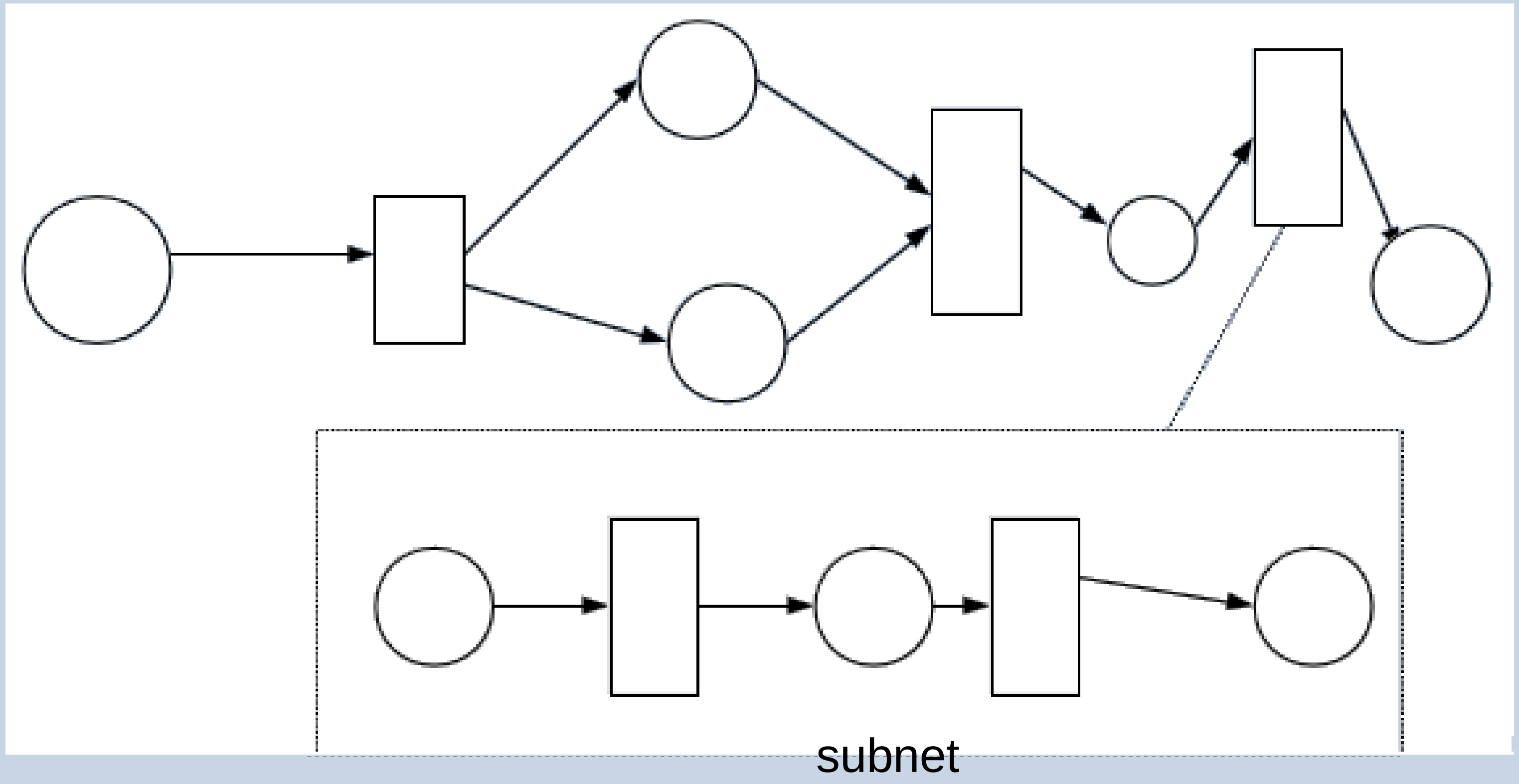
Extension of petri nets and use of it in workflow specification

Colored petri net



Extension of petri nets and use of it in workflow specification

Hierarchical petri net



Workflow Patterns

- Some frequently observed routings sequences in the order of tasks in most of the workflow processes.
- Provides basis for assessing relative strength and weakness of workflow description languages.

Basic patterns

- Sequence
- Parallel split (AND-split)
- Synchronization (AND-join)
- Conditional split (XOR-split)
- Simple merge (XOR-join)

Advanced Branch and synchronization patterns

- Multi-choice
- Synchronizing merge
- Multi-merge
- Discrimination

Structural patterns

- Arbitrary cycle
- Implicit termination

Multiple instance patterns

- Multiple instance without synchronization
- Multiple instance with a priori design time knowledge
- Multiple instance with a priori runtime knowledge
- Multiple instance without priori runtime knowledge

State based patterns

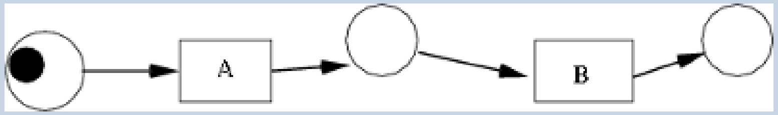
- Deferred choice
- Interleaved parallel routing
- Milestone

Cancellation patterns

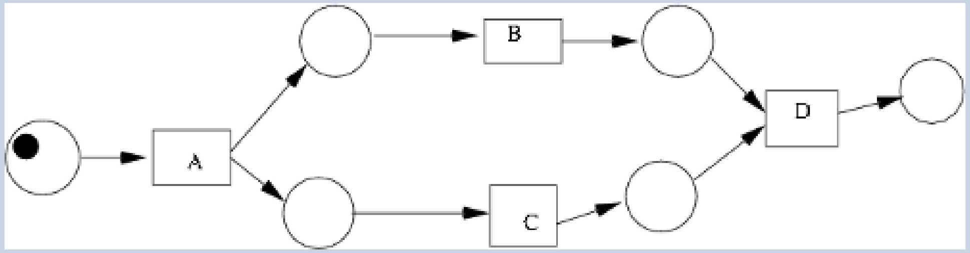
- Cancel activity
- Cancel case

Workflow patterns & petri nets

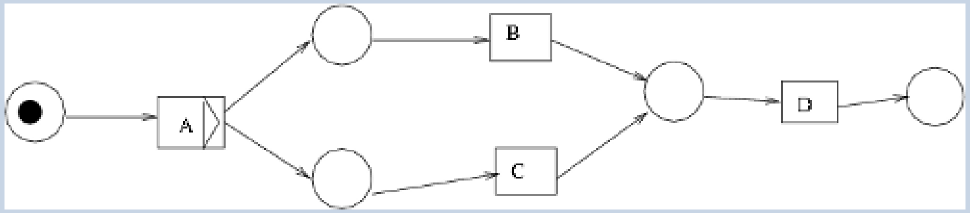
(Basic patterns)



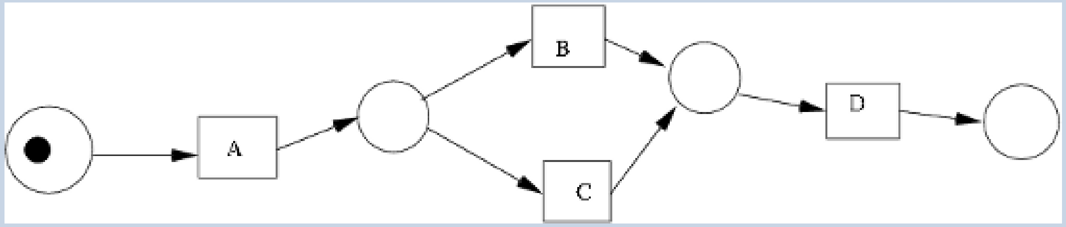
sequence



AND-split/join

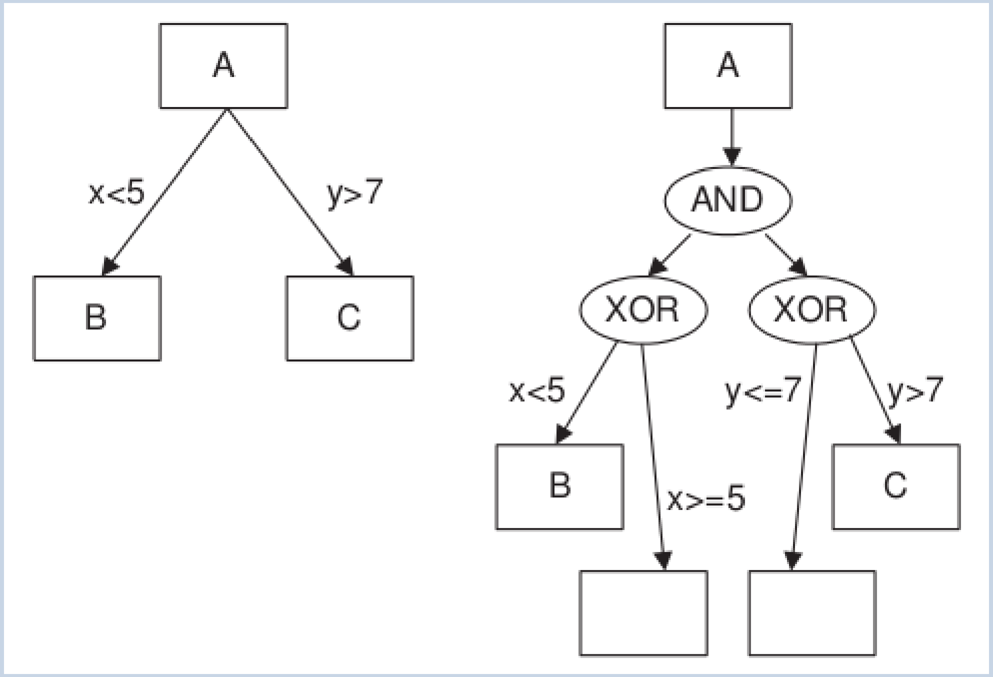


XOR-split/join (explicit)



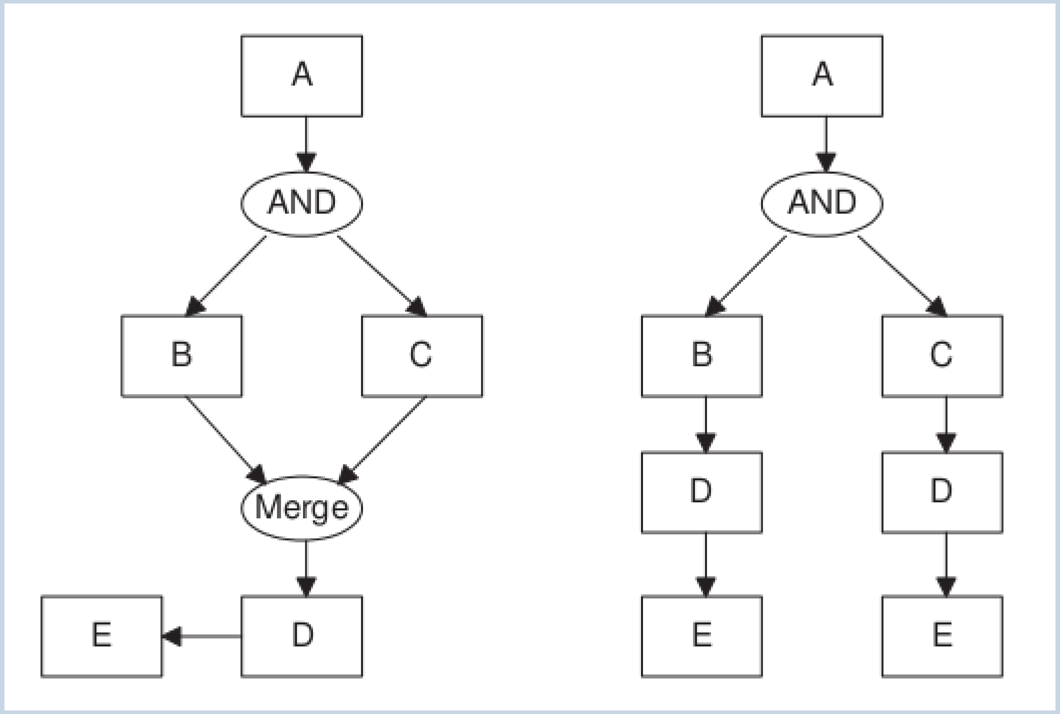
XOR-split/join (implicit)

Workflow patterns & petri nets



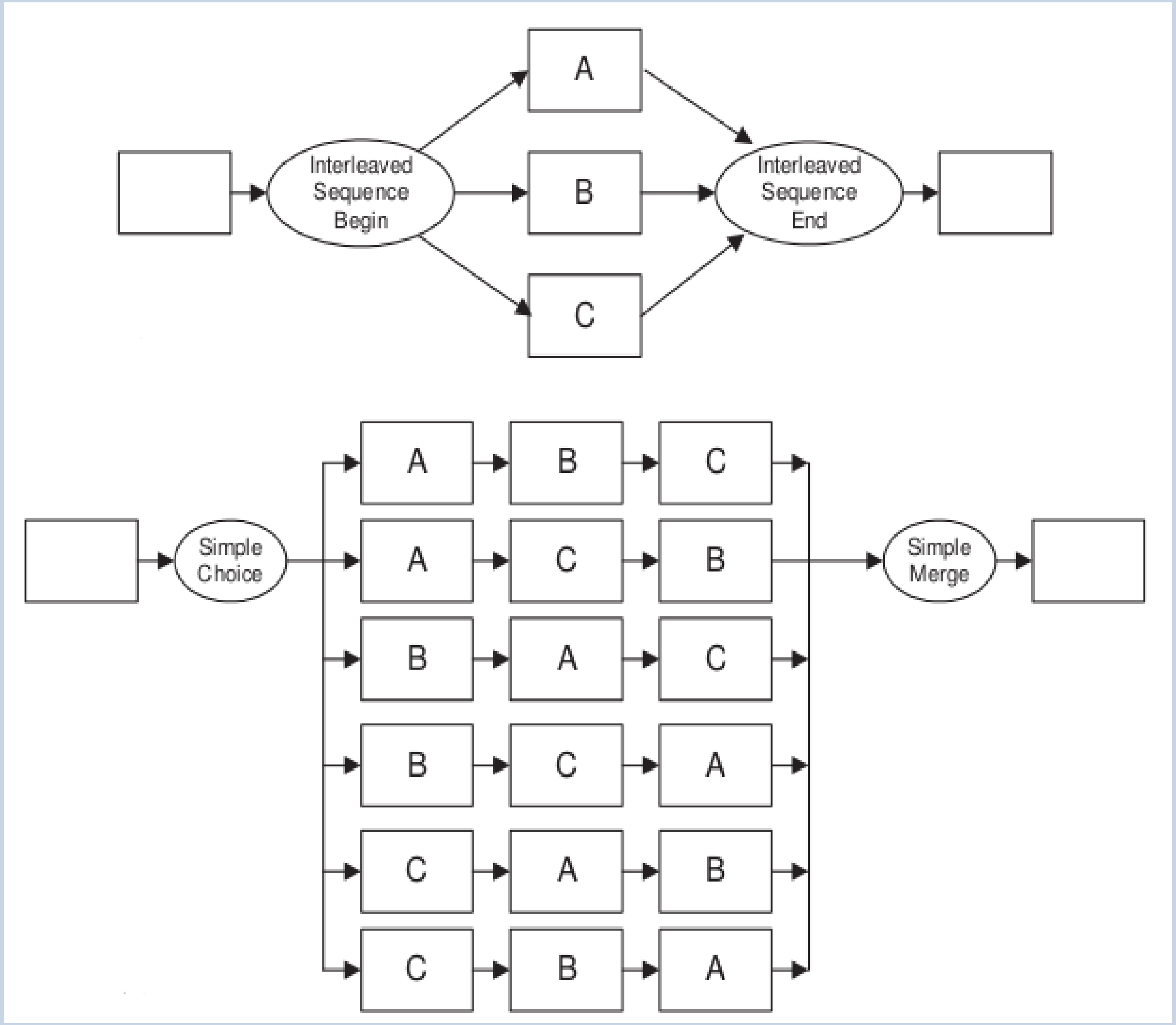
Possible implementation of OR-split

Workflow patterns & petri nets



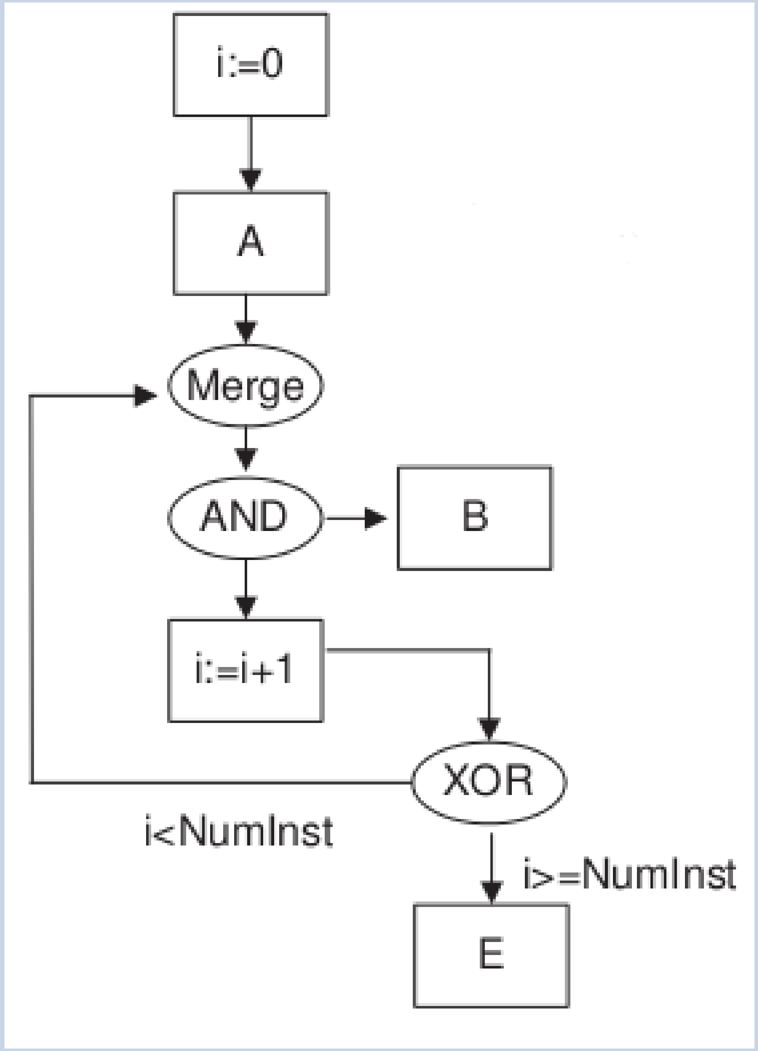
Implementation of multi-merge

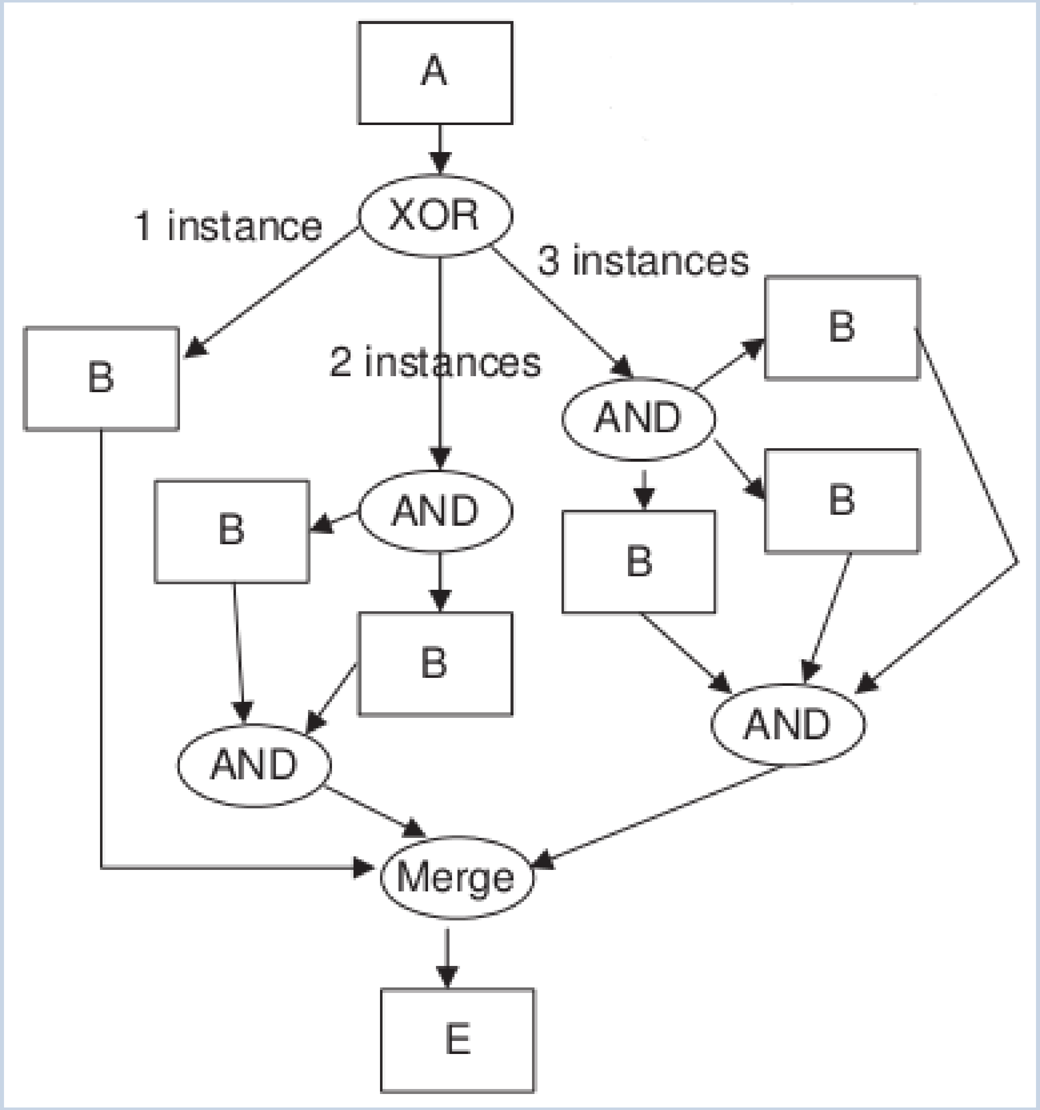
Workflow patterns & petri nets



Implementation of parallel interleaved

Workflow patterns & petri nets





Patterns involving multiple instances

Workflow patterns & petri nets

- **Other than basic patterns are not trivially supported.**
- **Too much designer effort and result is very complex diagrams.**
- **Advanced branching & Synchronization patterns :**

  -- OR split/marge sometimes behave like AND, sometimes like XOR, sometimes like n-out-of-m.
- **Multiple-instance patterns :**

  -- Burden of keeping track of active instances for synchronization.
- **Cancellation patterns:**

  -- Solution is not easy because of local nature of petri nets.

YAWL : yet another workflow language

- Similar (not the exact same) notations like petri nets
- Independent theory and semantics
- Can handle multi-instance and cancellation patterns effectively

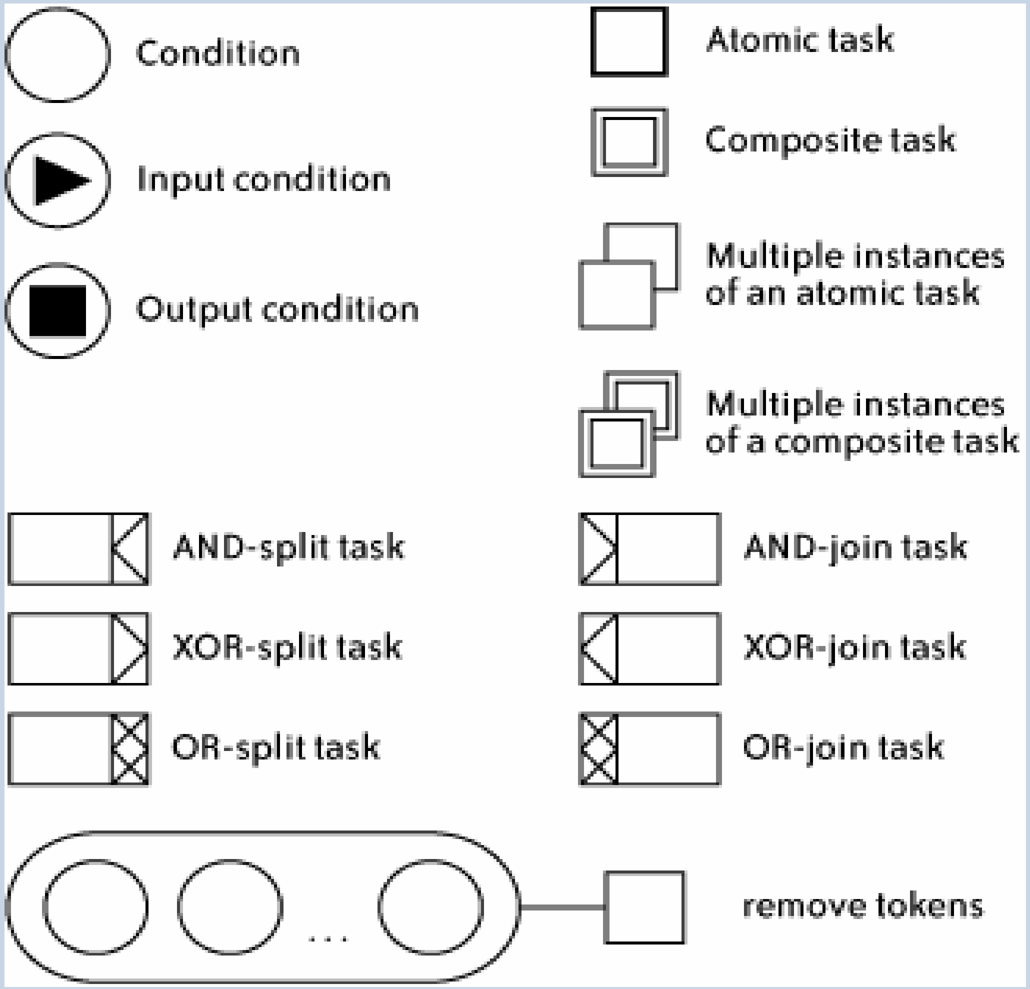
Similarity with petri nets (WF-net)

- Conditions (includes start condition and end condition)
- Arcs
- Task
- Token
- Composite task (high-level petri net)
- Support for AND and XOR split/join

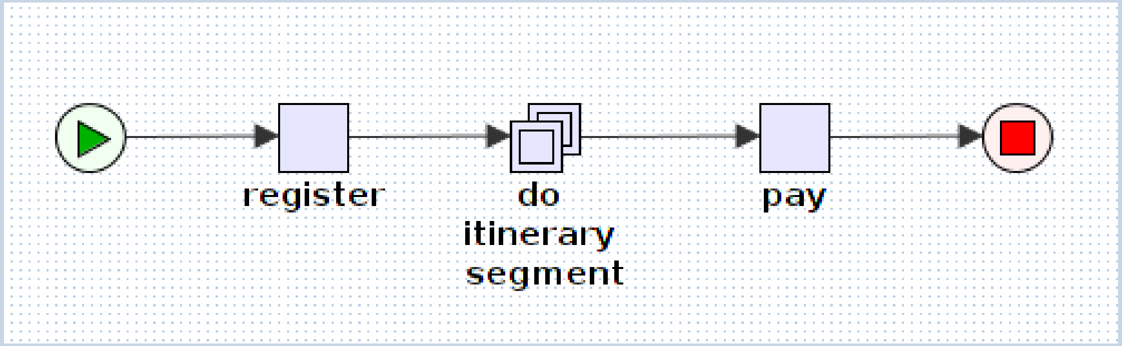
Extension over petri nets

- Multiple instances of task
- Directly connected tasks
- Cancellation set
- OR-split and Join attributes of task

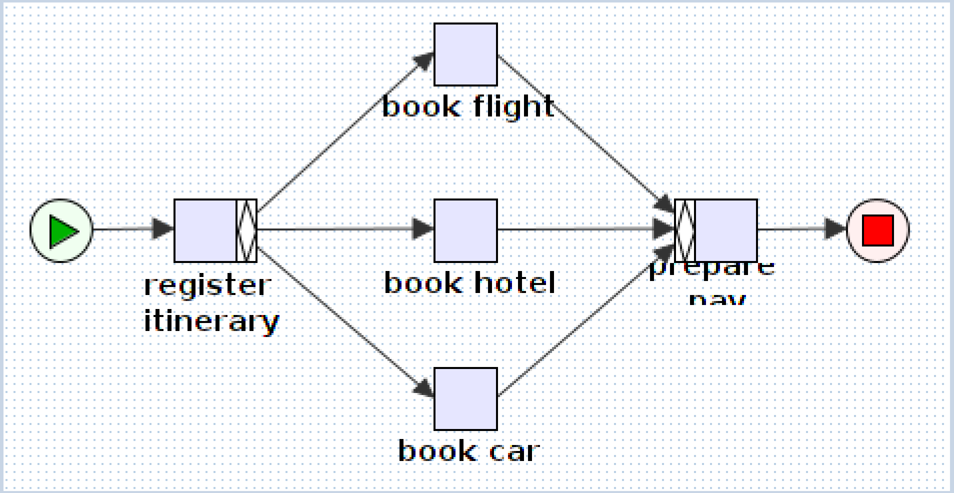
Symbols


Condition
Input condition
Output condition
Atomic task
Composite task
Multiple instances of an atomic task
Multiple instances of a composite task
AND-split task
XOR-split task
OR-split task
AND-join task
XOR-join task
OR-join task
...
remove tokens

Example of workflow in YAWL

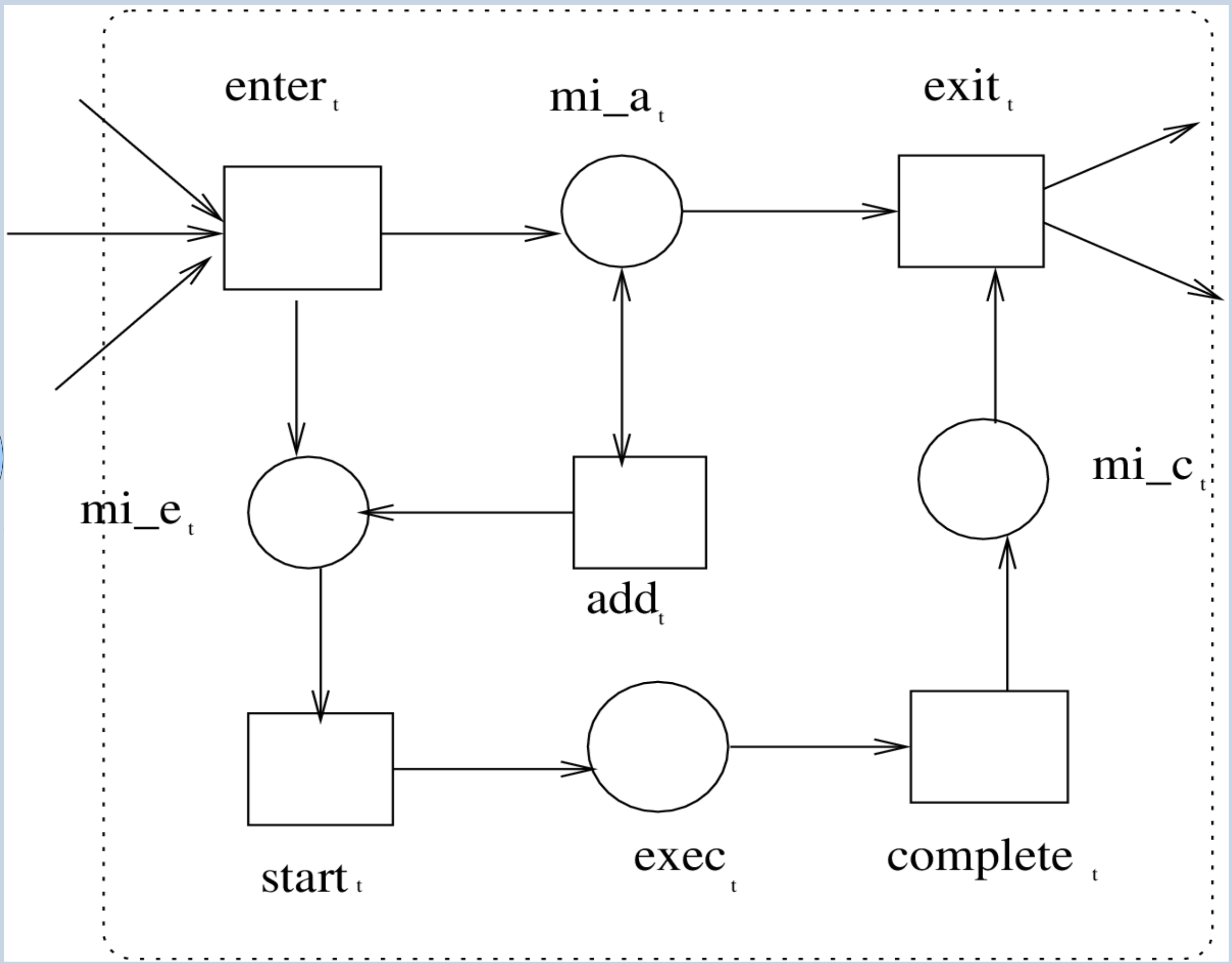


Maketrip process (starting net)

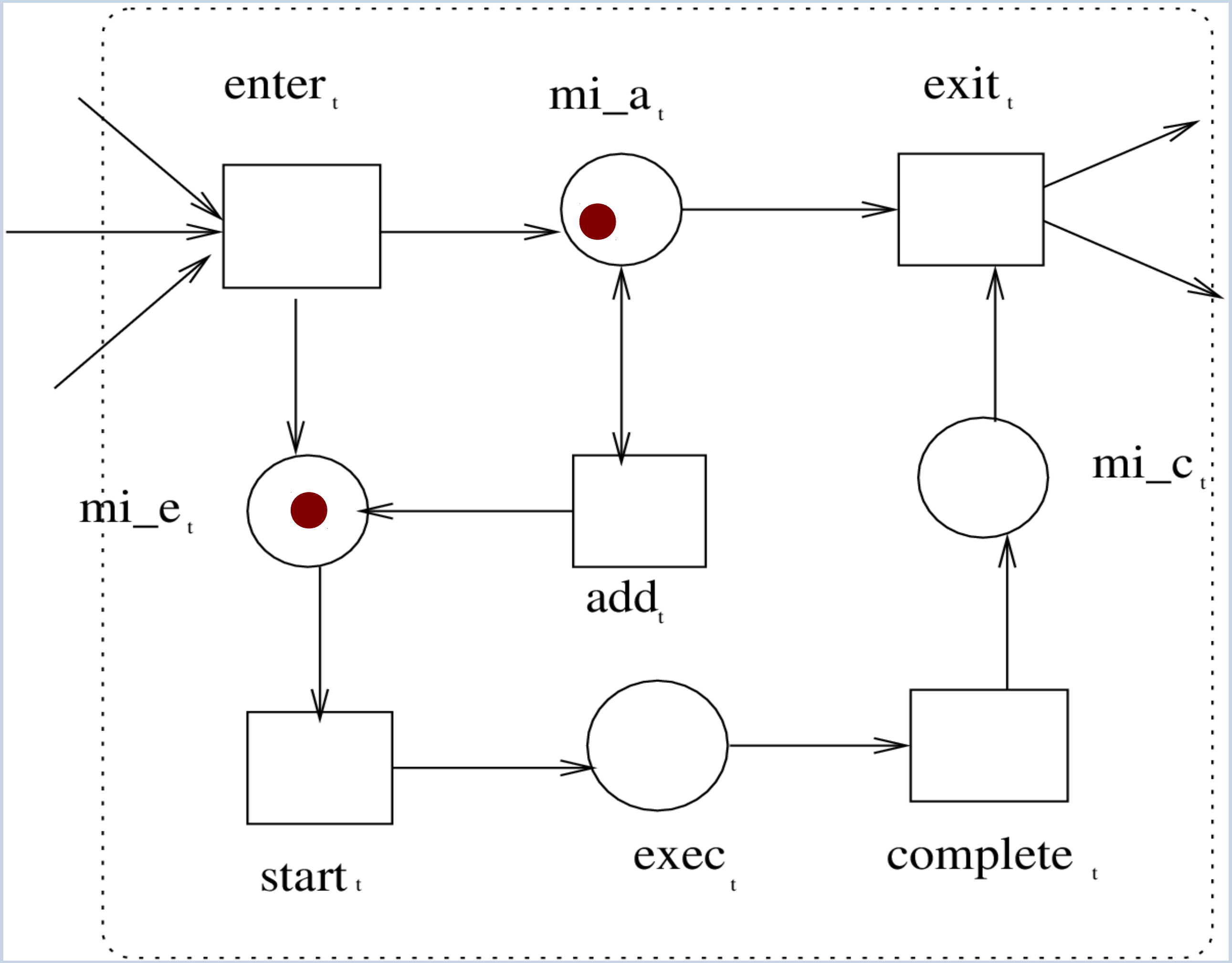


Do itinerary segment task (subnet)

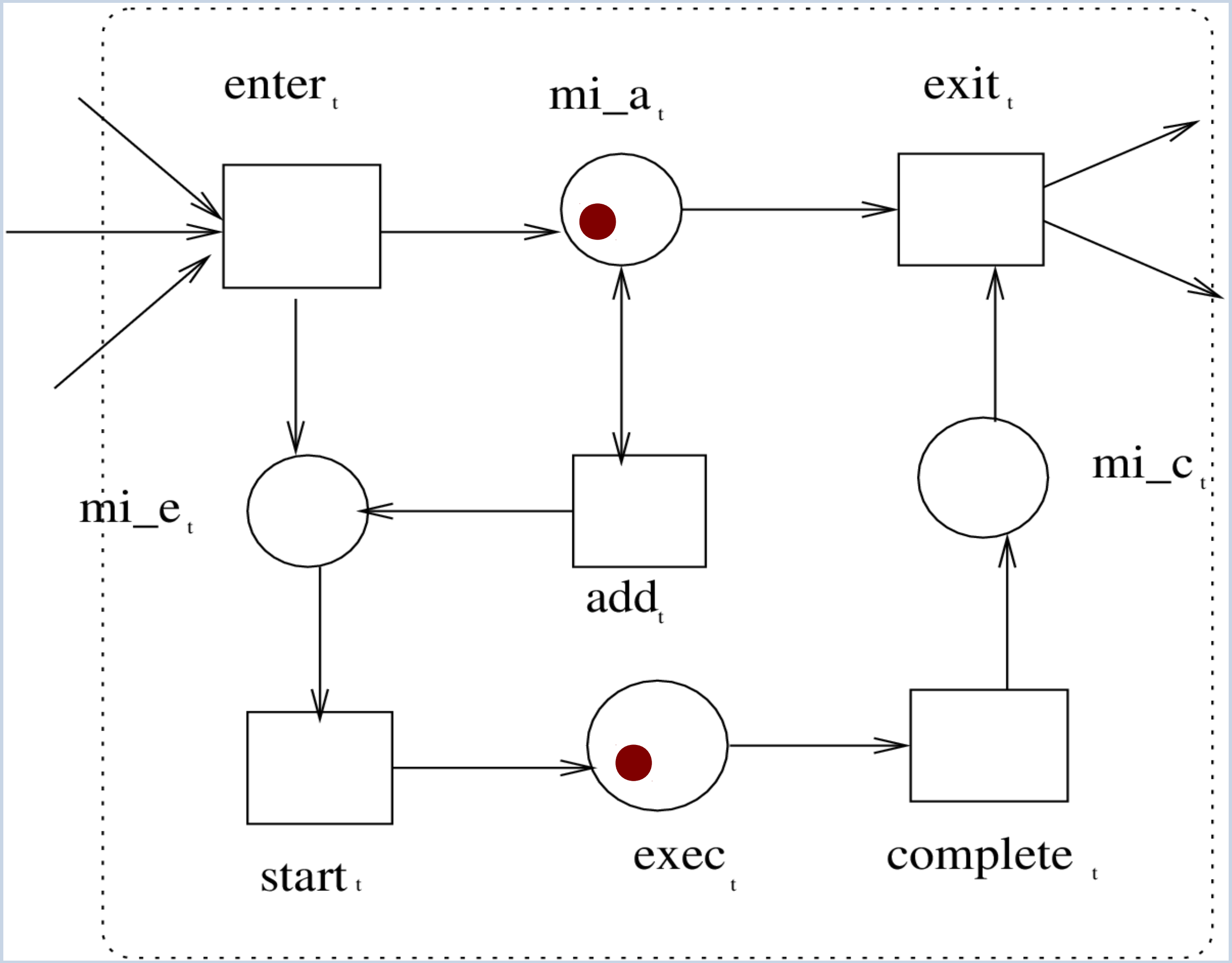
Task in YAWL



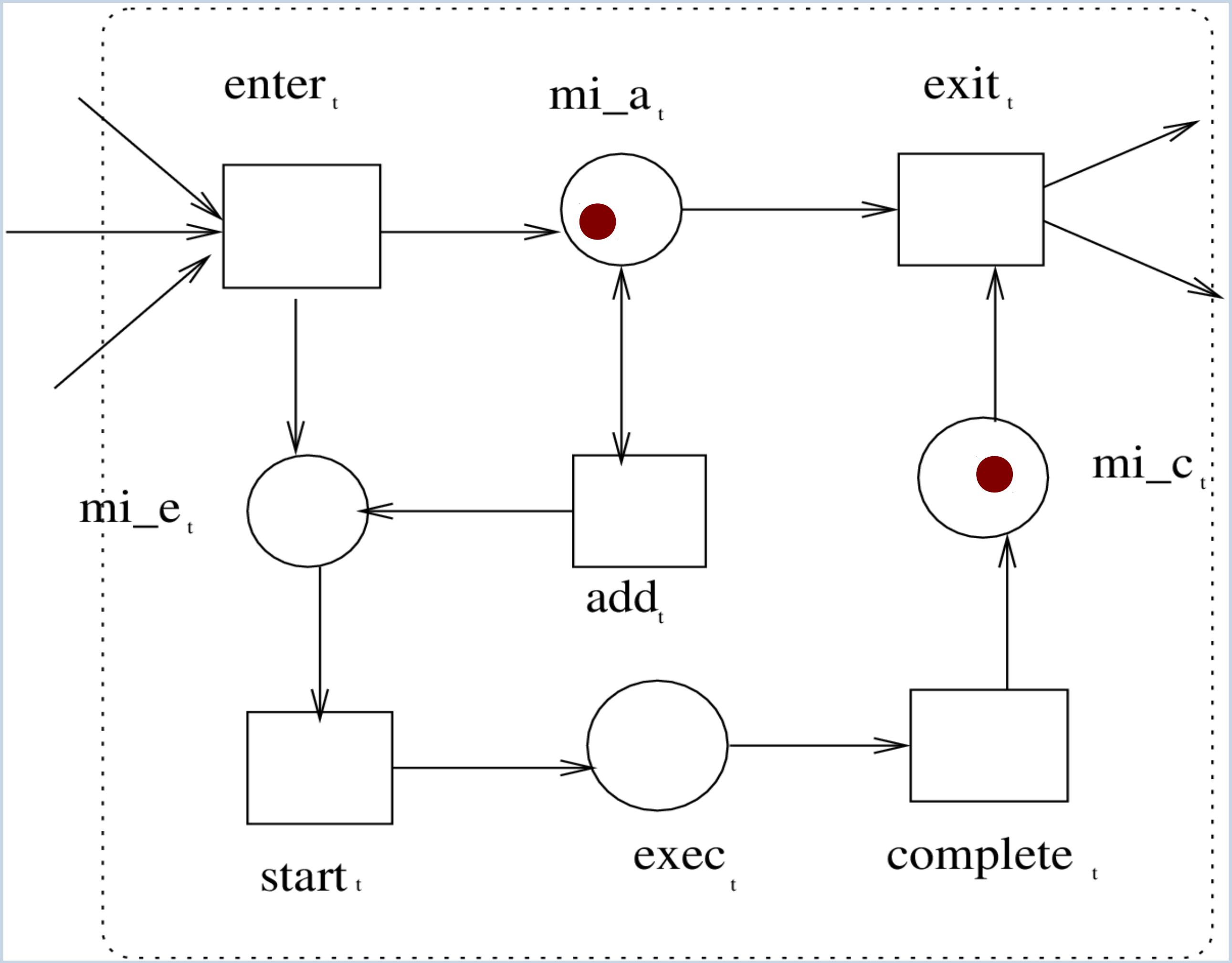
Task in YAWL



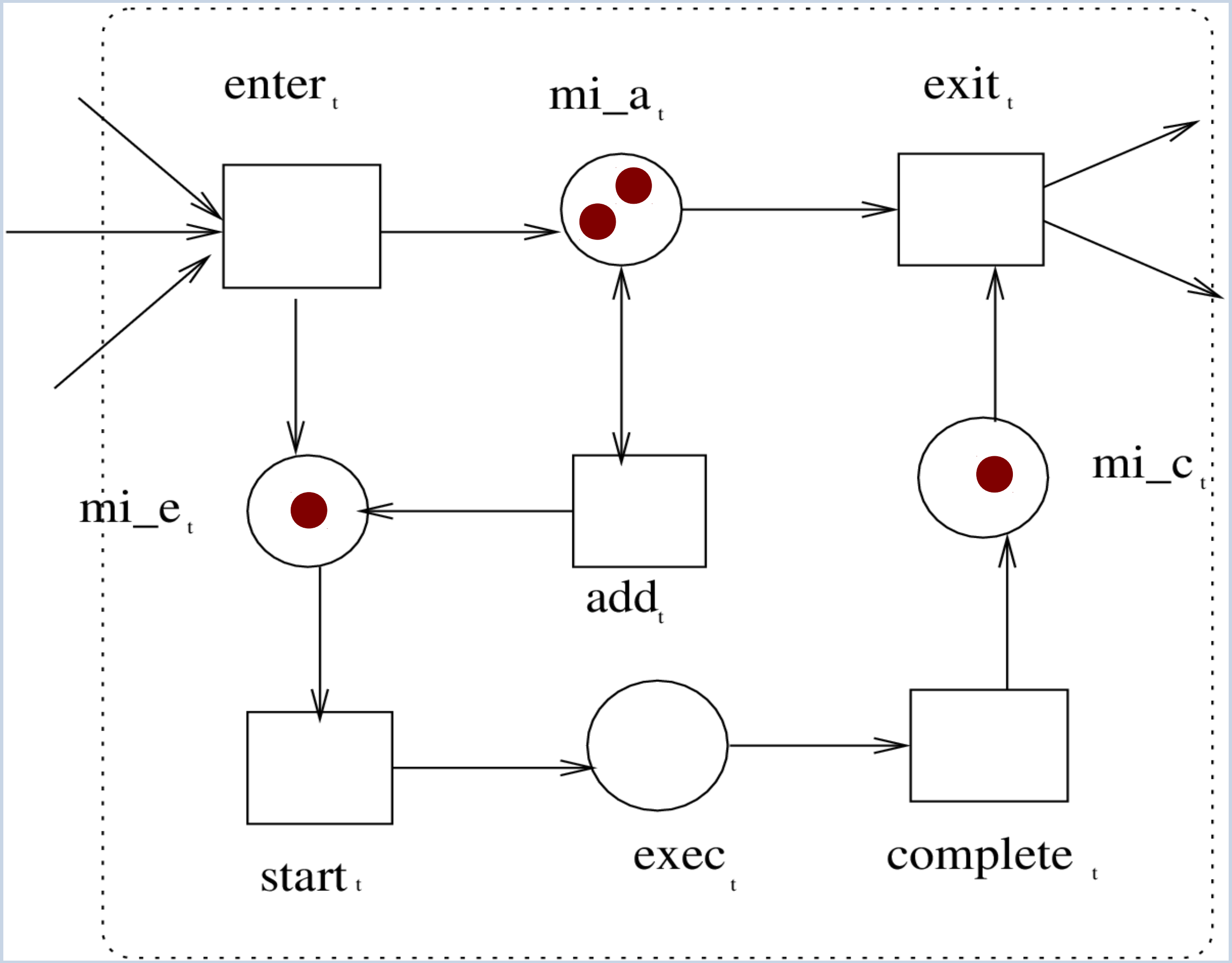
Task in YAWL



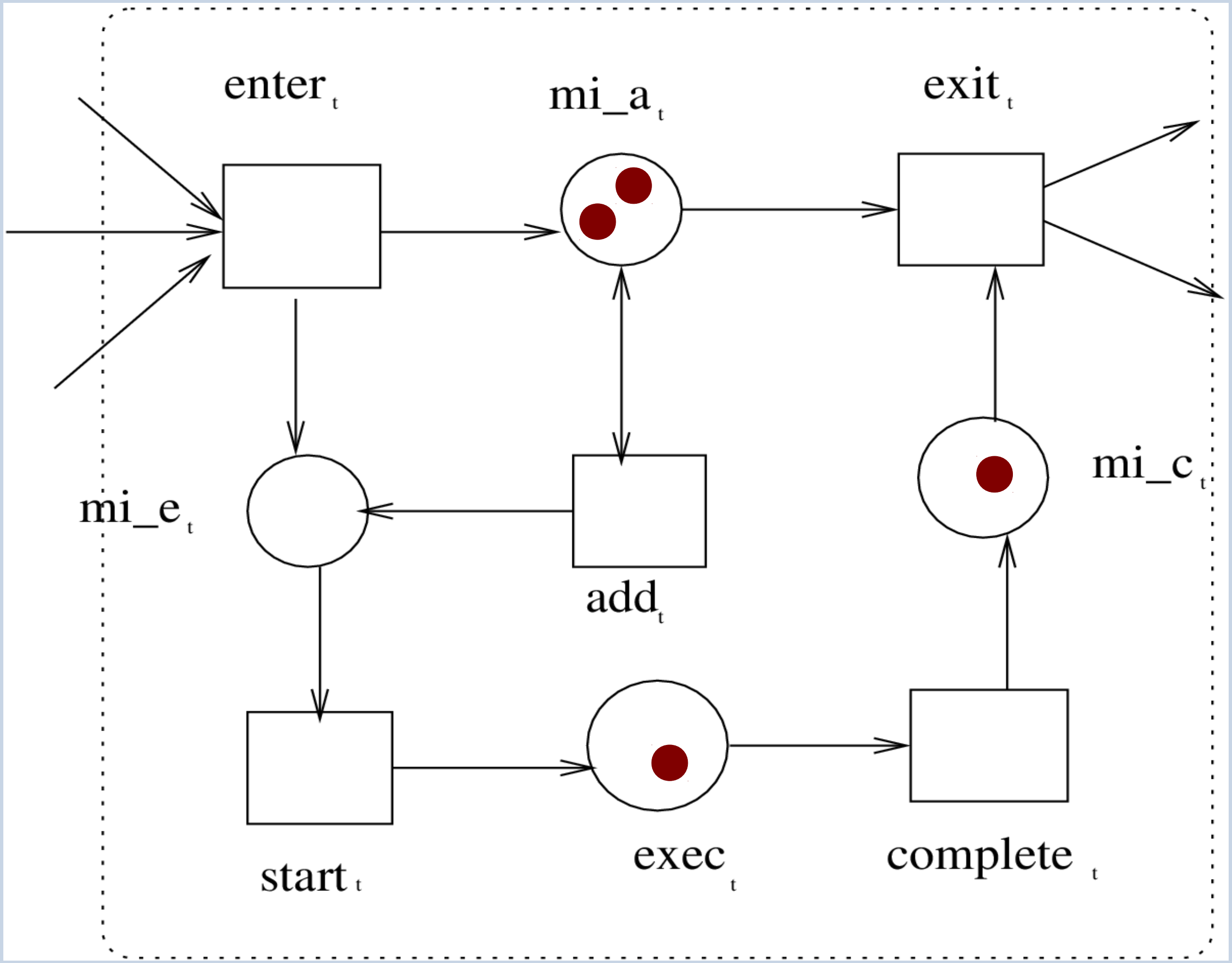
Task in YAWL



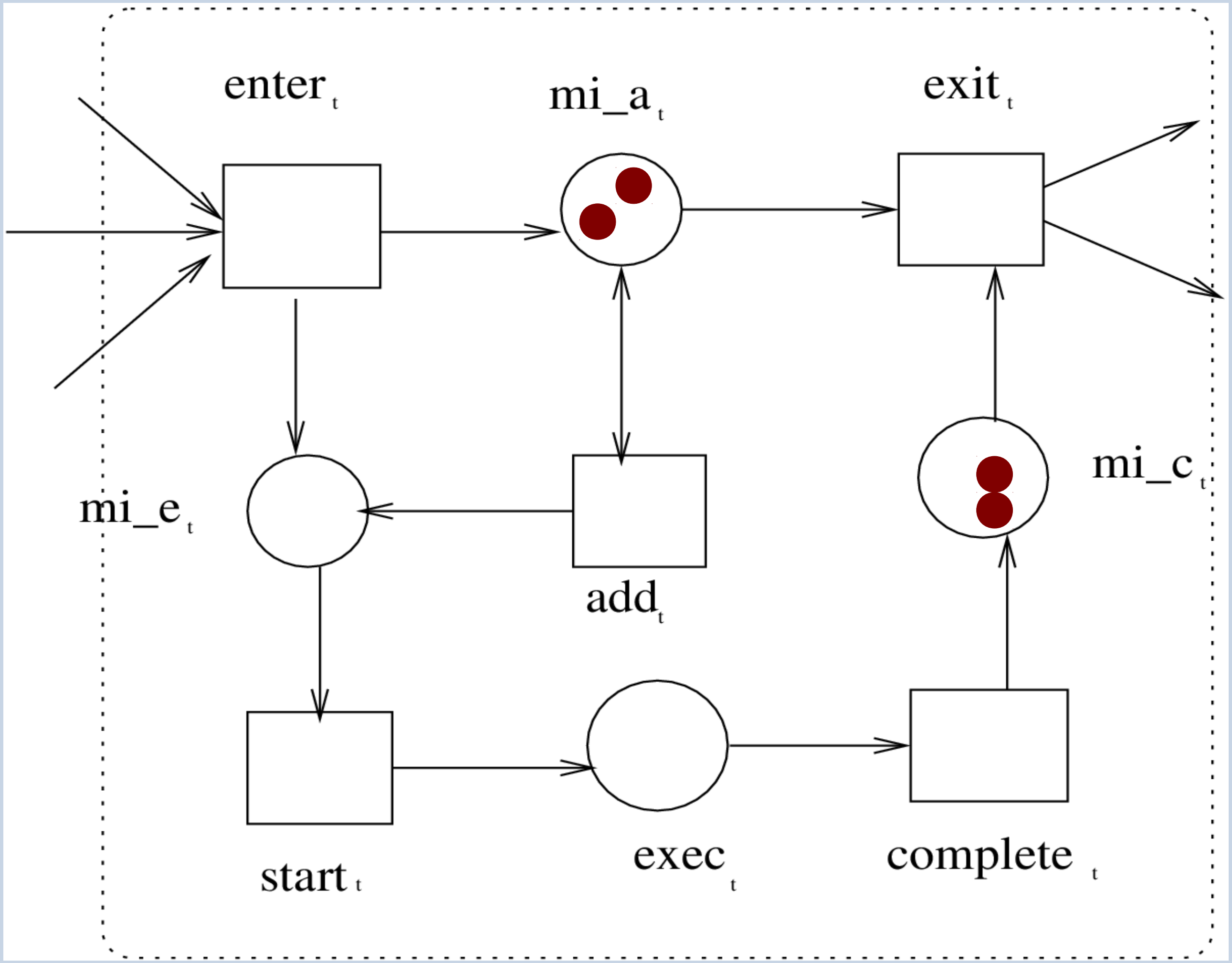
Task in YAWL



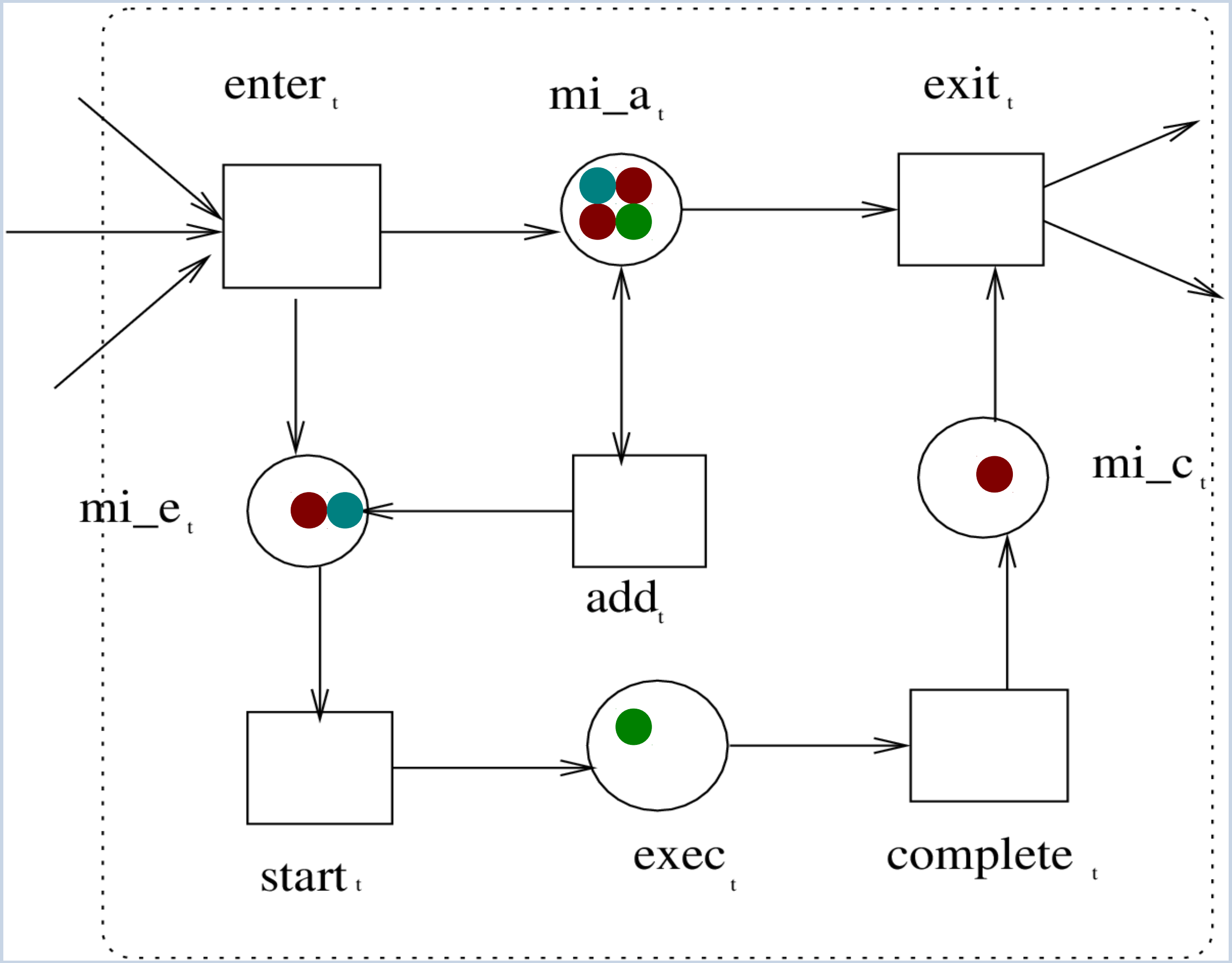
Task in YAWL



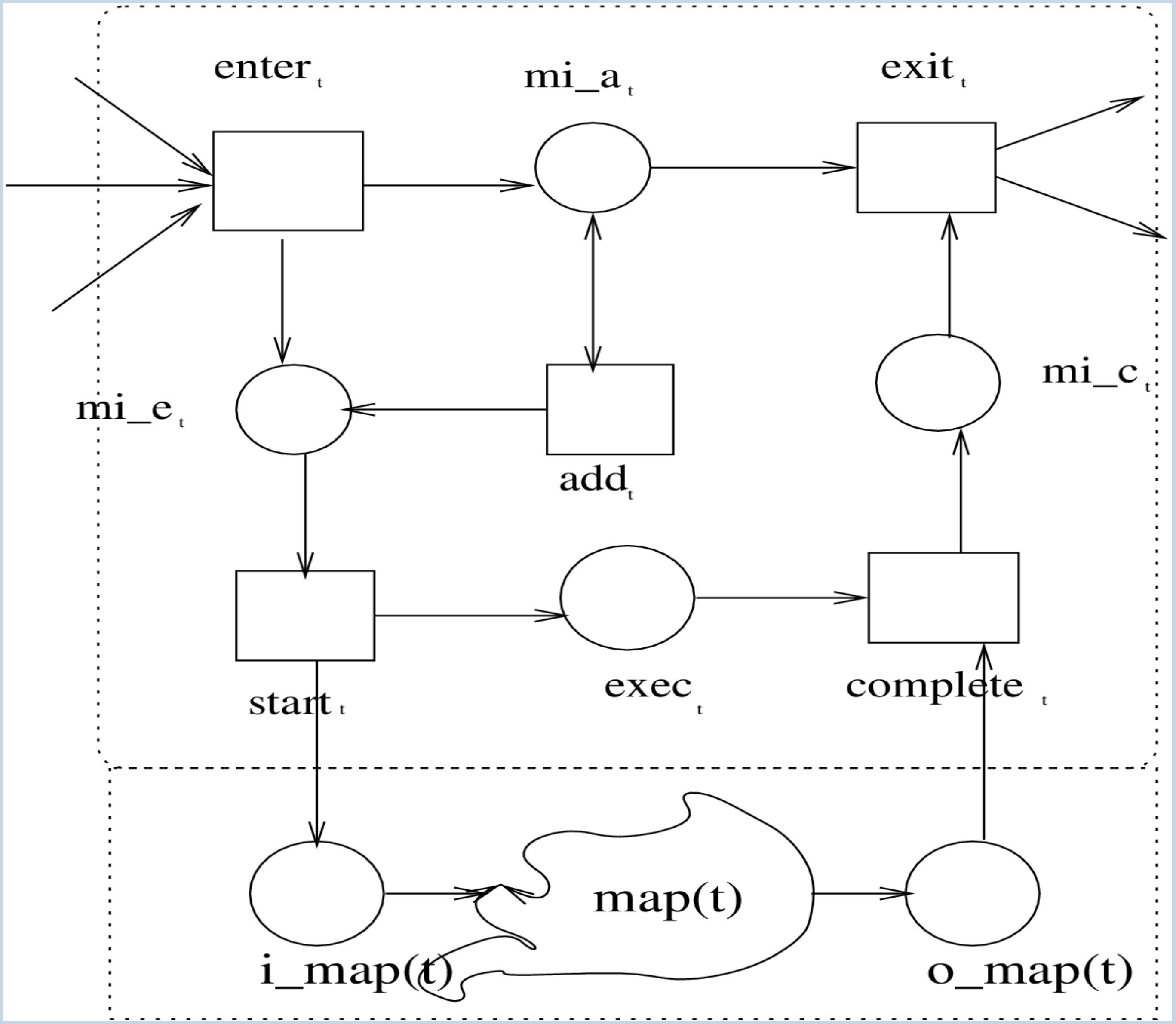
Task in YAWL



Task in YAWL

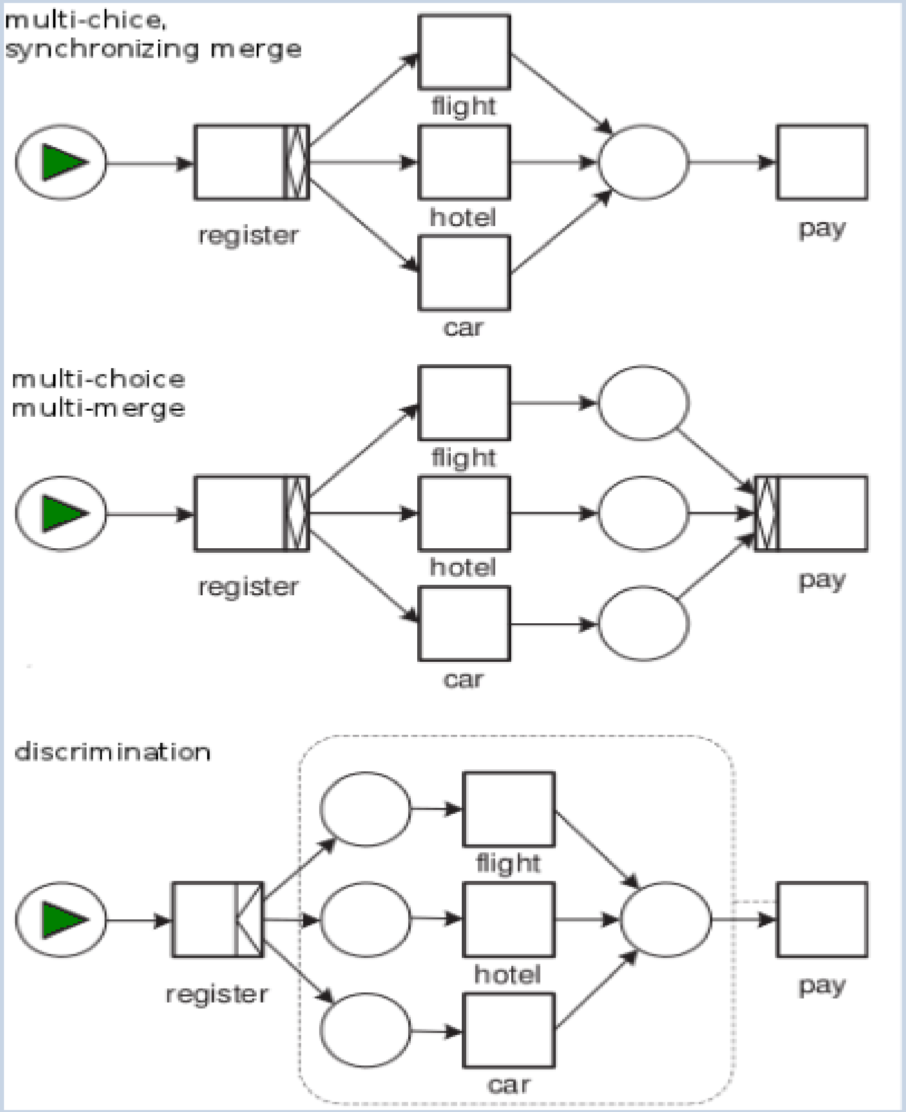


Composite task in YAWL



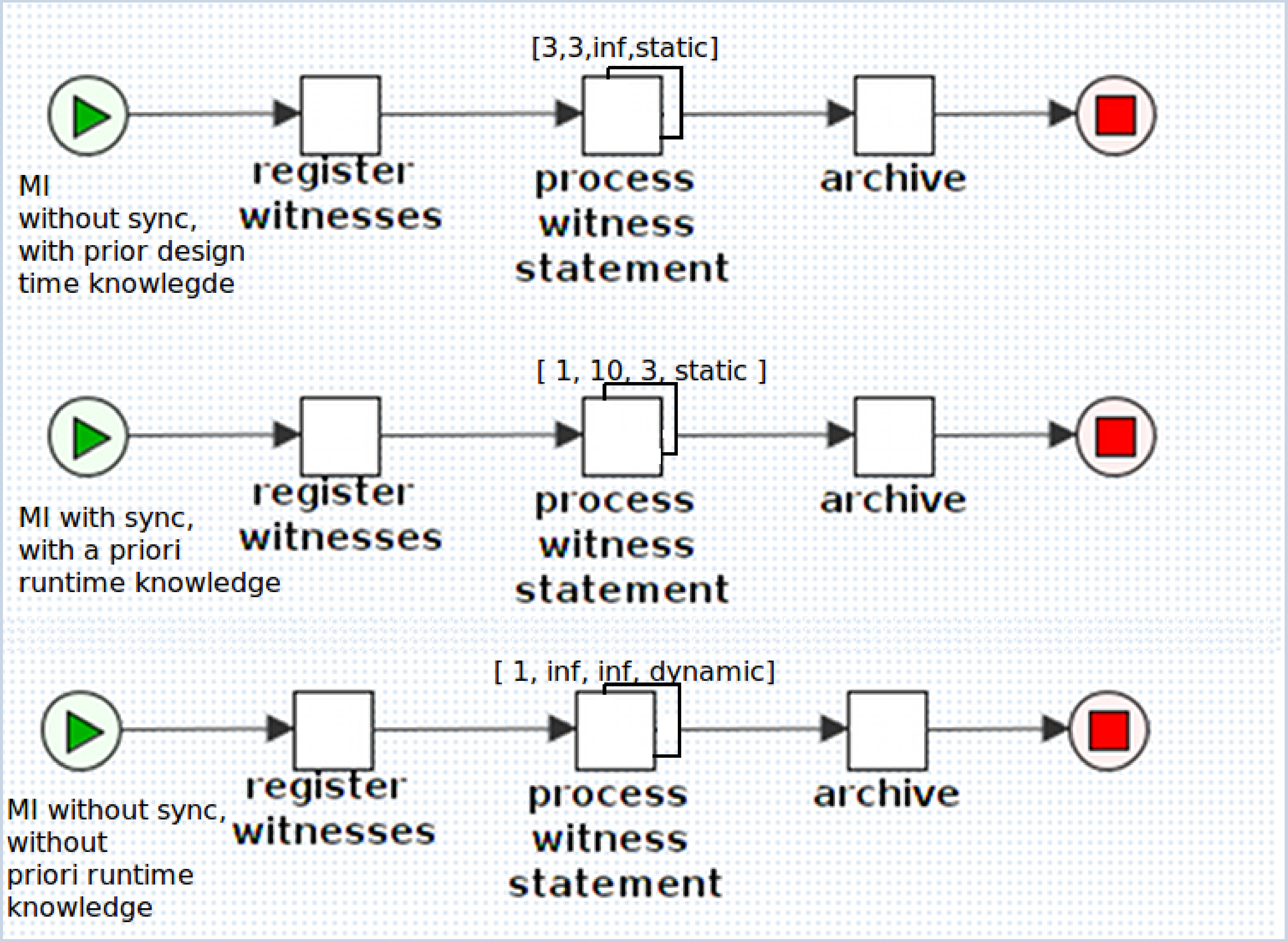
Workflow patterns using YAWL

(Advanced branching and synchronization)



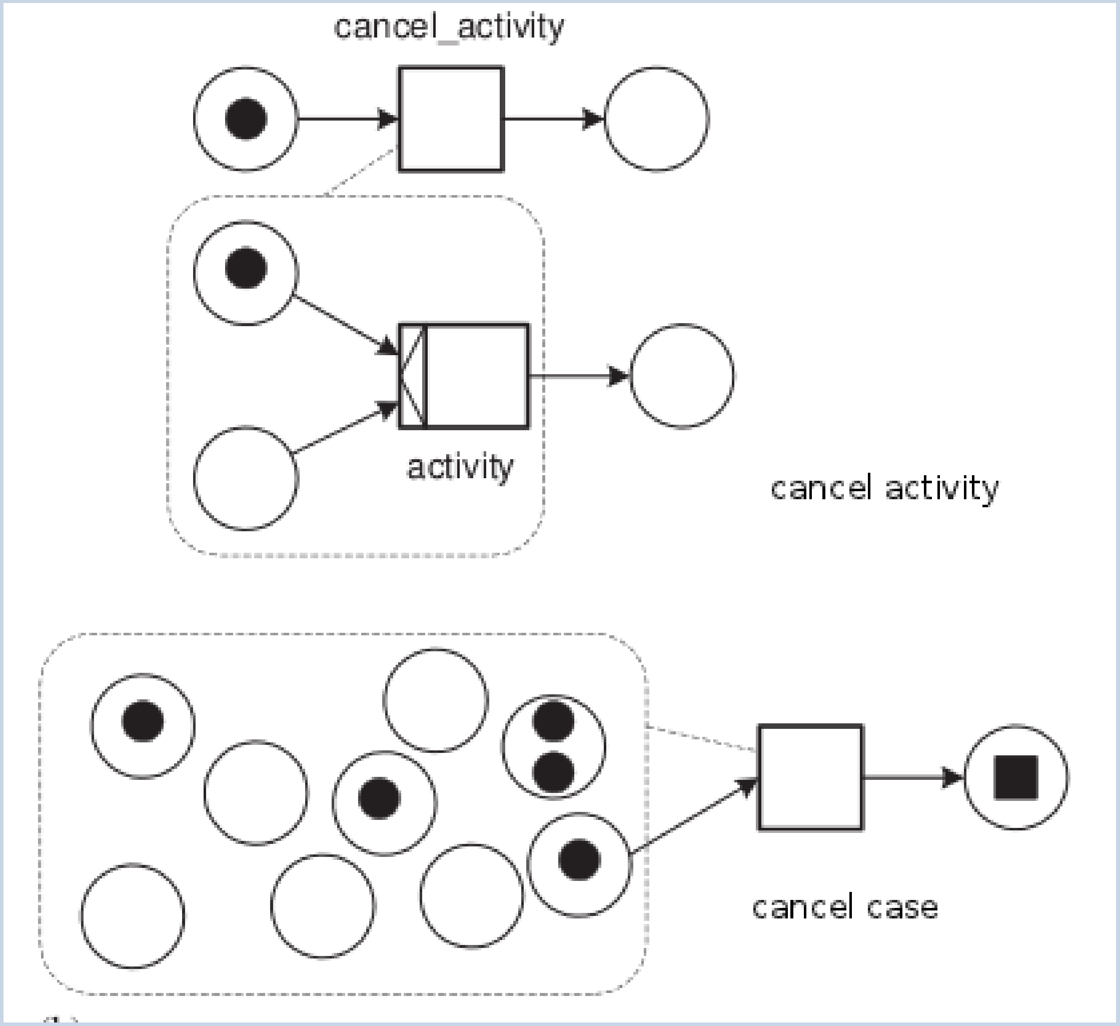
Workflow patterns using YAWL

(Multi-instance patterns)


[3,3,inf,static]
register witnesses
process witness statement
archive
MI without sync, with prior design time knowlegde
[1, 10, 3, static]
register witnesses
process witness statement
archive
MI with sync, with a priori runtime knowledge
[1, inf, inf, dynamic]
register witnesses
process witness statement
archive
MI without sync, without priori runtime knowledge

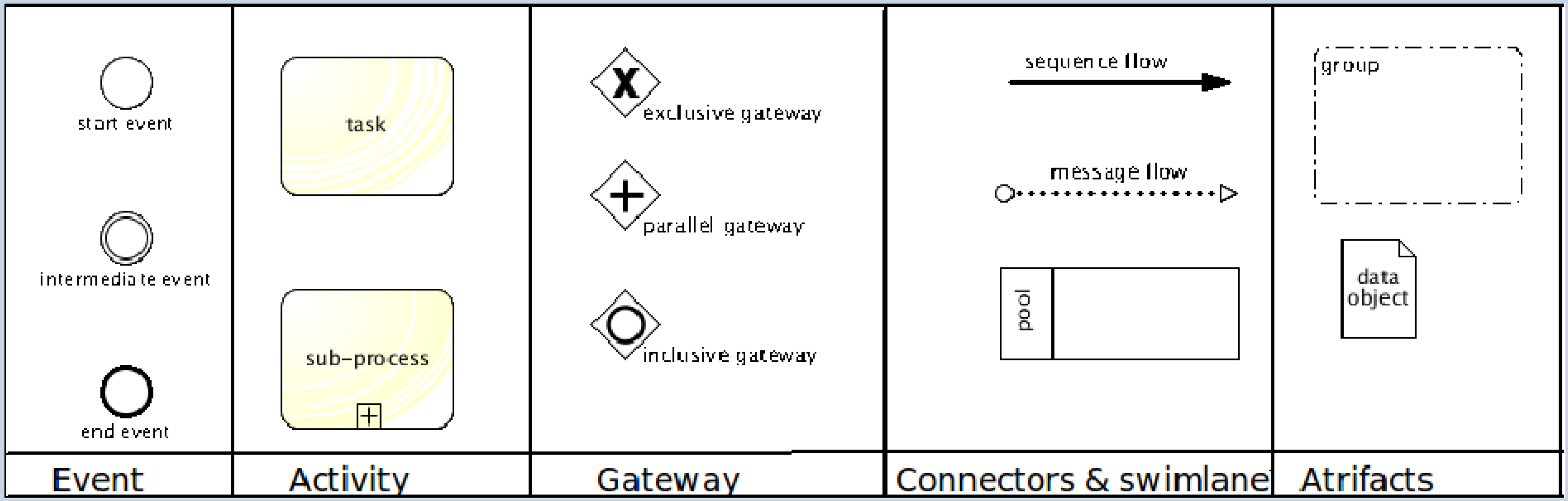
Workflow patterns using YAWL

(Cancellation patterns)



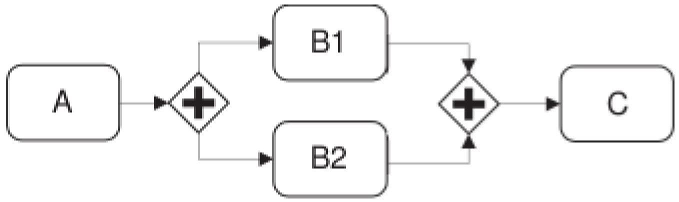
BPMN: Business Process Model & Notation

- Modeling notation for process specification by OMG.
- Execution semantics
- Have mapping to execution languages like BPEL, YAWL.

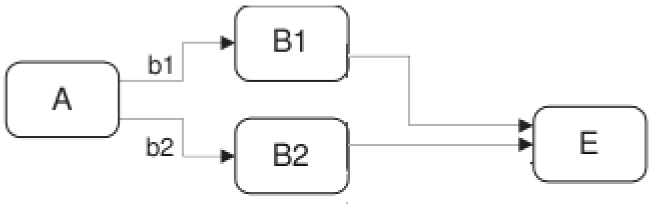


BPMN: expressing workflow patterns

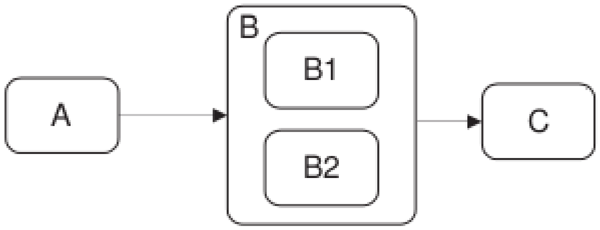
(Basic patterns)



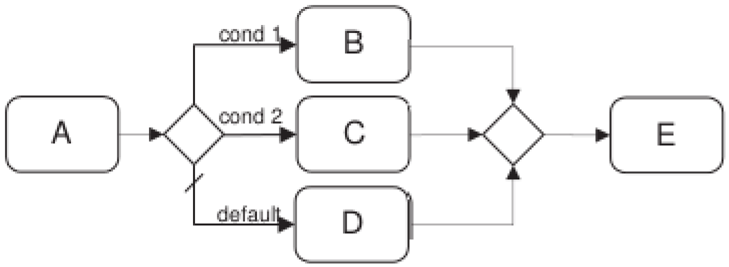
Parallel split/synchronization with parallel gateway



Implicit parallel split/merge



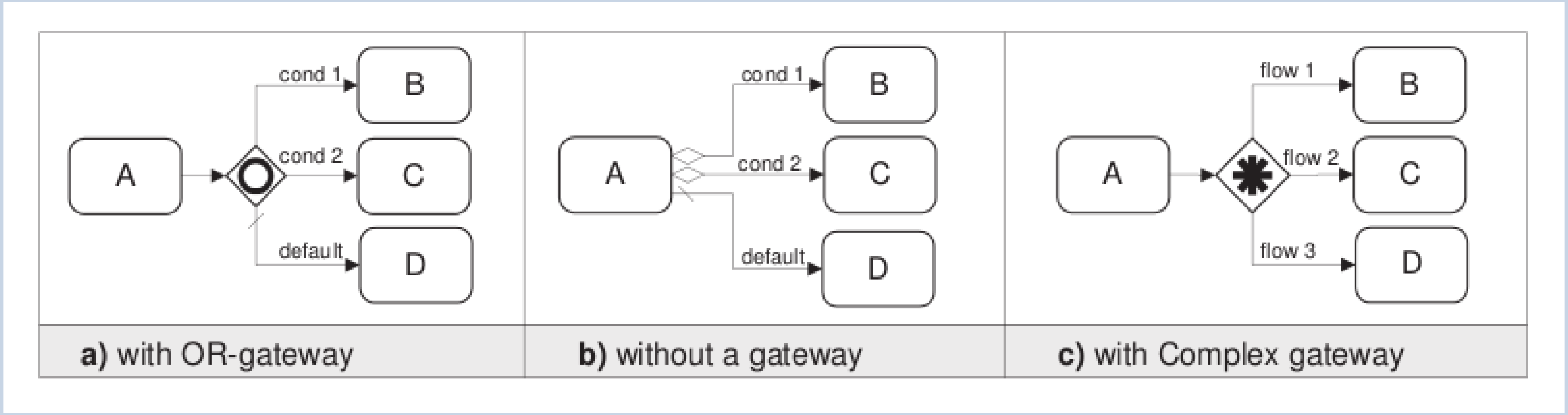
Parallel split/synchronization by using sub-process



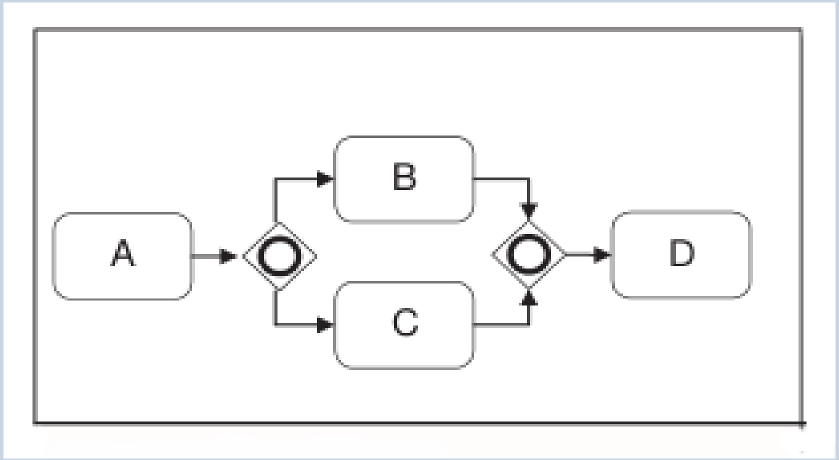
Conditional choice and simple merge using conditional gateway

BPMN: expressing workflow patterns

(Multiple choice)

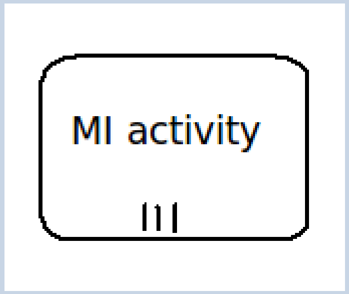


(Synchronizing merge)

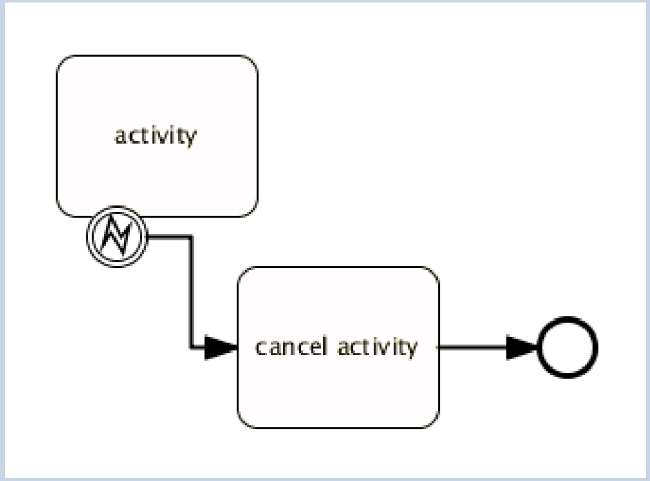


BPMN: expressing workflow patterns

(Multiple instance patterns)



(Cancellation patterns)



References

*[1] **The Application of Petri Nets to Workflow Management**, W.M.P. van der Aalst, The Journal of Circuits, Systems and Computers, 8(1):21-66, 1998*

*[2] **YAWL: yet another workflow language**, W.M.P. van der Aalst and A.H.M. ter Hofstede, Information Systems, 30(4):245-275, 2005*

*[3] **Business Process Model and Notation (BPMN)**, Version 2.0, OMG Document Number:formal/2011-01-03, Standard document URL: http://www.omg.org/spec/BPMN/2.0*

*[4] **workflow patterns**, W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P.Barros, BETA Working Paper Series, WP 47, Eindhoven University of Technology , Eindhoven, 2000*

*[5] **Pattern-based Analysis of BPMN - an extensive evaluation of the Control-flow,the Data and the Resource Perspectives**, P. Wohed, W.M.P. van der Aalst, M. Dumas, A.H.M. ter Hofstede, N. Russell, BPM Center Report BPM-06-17, BPMcenter.org, 2006*

[6] www.workflowpatterns.com

[7] www.gridworkflow.org