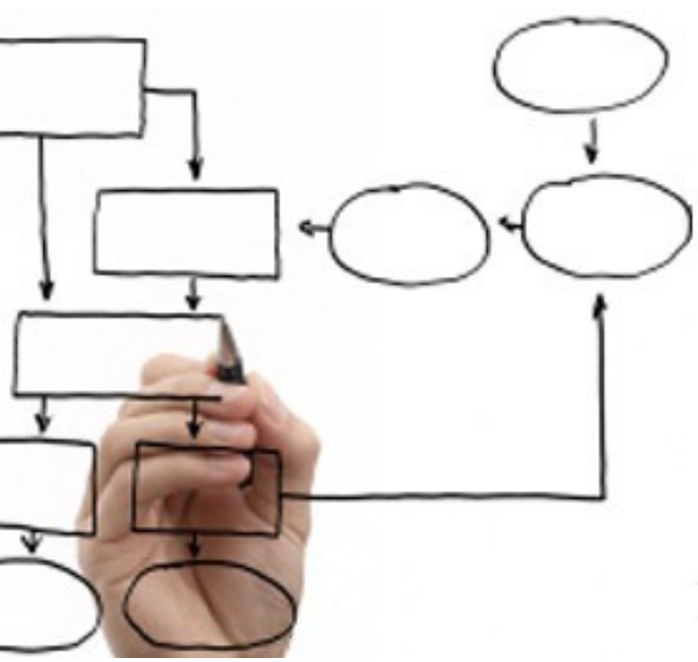
Methods for the specification and verification of business processes

MPB (6 cfu, 295AA)



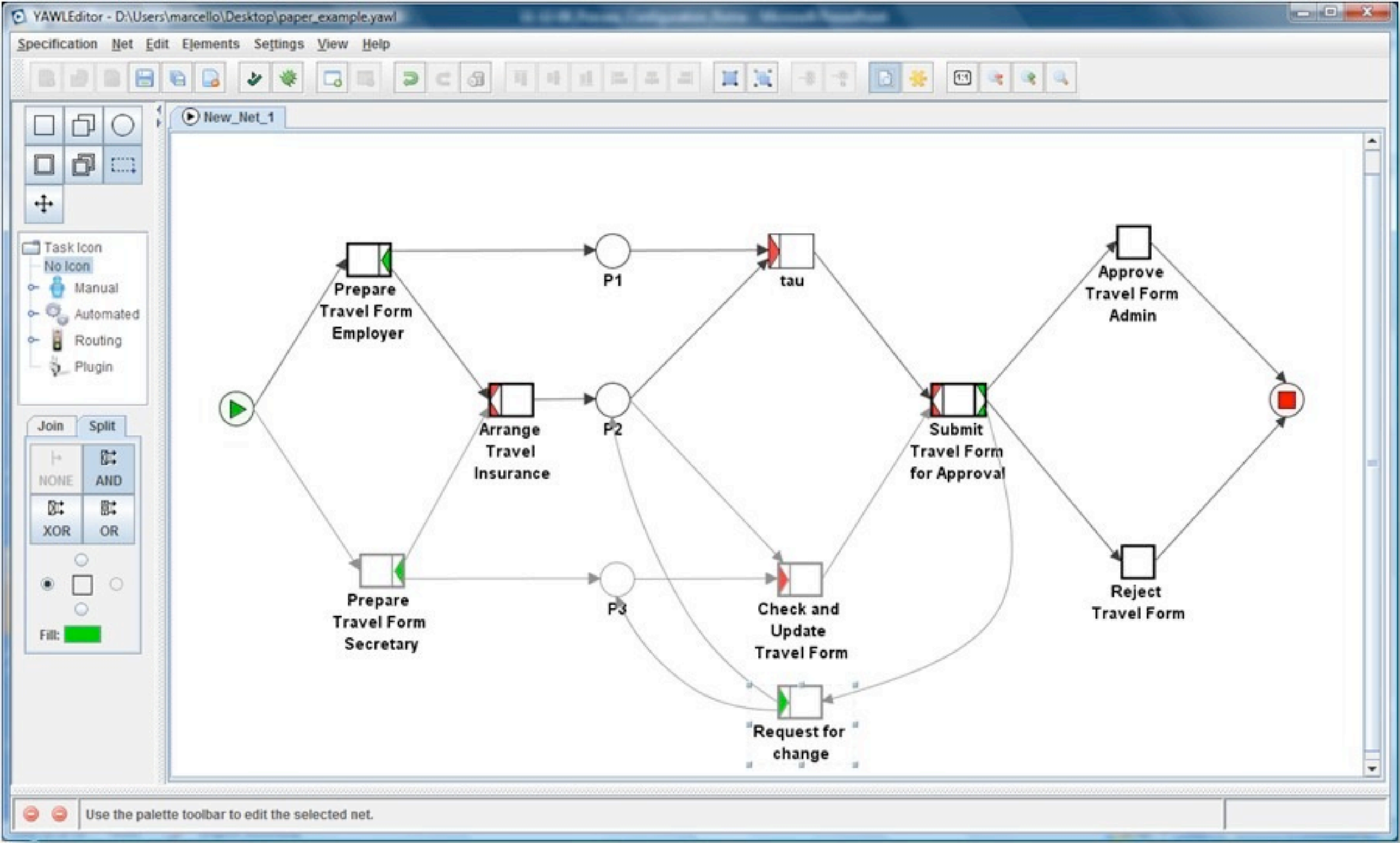
Roberto Bruni

http://www.di.unipi.it/~bruni

23 - Yet another workflow language

Object

We overview the key features of YAWL and the main differences with Petri nets


YAWLEditor - D:\Users\marcello\Desktop\paper_example.yawl
Specification Net Edit Elements Settings View Help
New_Net_1
Task Icon
No Icon
Manual
Automated
Routing
Plugin
Join
Split
NONE
AND
XOR
OR
Fill:
Prepare Travel Form Employer
P1
tau
Approve Travel Form Admin
Arrange Travel Insurance
P2
Submit Travel Form for Approval
Prepare Travel Form Secretary
P3
Check and Update Travel Form
Reject Travel Form
Request for change
Use the palette toolbar to edit the selected net.

Expressiveness...
not in the formal sense
(w.r.t. WF patterns)

A wish list

Graphical language

Formal syntax

Rigourous theory

Direct support for (all?) workflow patterns

Tool support

Basic Control Flow Patterns

- Pattern 1 (Sequence)
- Pattern 2 (Parallel Split)
- Pattern 3 (Synchronization)
- Pattern 4 (Exclusive Choice)
- Pattern 5 (Simple Merge)

Advanced Branching and Synchronization Patterns

- Pattern 6 (Multi-choice)
- Pattern 7 (Synchronizing Merge)
- Pattern 8 (Multi-merge)
- Pattern 9 (Discriminator)

Structural Patterns

- Pattern 10 (Arbitrary Cycles)
- Pattern 11 (Implicit Termination)

Cancellation Patterns

- Pattern 19 (Cancel Activity)
- Pattern 20 (Cancel Case)

State-based Patterns

- Pattern 16 (Deferred Choice)
- Pattern 17 (Interleaved Parallel Routing)
- Pattern 18 (Milestone)

Patterns involving Multiple Instances

- Pattern 12 (Multiple Instances Without Synchronization)
- Pattern 13 (Multiple Instances With a Priori Design Time Knowledge)
- Pattern 14 (Multiple Instances With a Priori Runtime Knowledge)
- Pattern 15 (Multiple Instances Without a Priori Runtime Knowledge)

pattern	product							
	Staffware	COSA	InConcert	Eastman	FLOWer	Domino	Meteor	Mobile
1 (seq)	+	+	+	+	+	+	+	+
2 (par-spl)	+	+	+	+	+	+	+	+
3 (synch)	+	+	+	+	+	+	+	+
4 (ex-ch)	+	+	+/-	+	+	+	+	+
5 (simple-m)	+	+	+/-	+	+	+	+	+
6 (m-choice)	-	+	+/-	+/-	-	+	+	+
7 (sync-m)	-	+/-	+	+	-	+	-	-
8 (multi-m)	-	-	-	+	+/-	+/-	+	-
9 (disc)	-	-	-	+	+/-	-	+/-	+
10 (arb-c)	+	+	-	+	-	+	+	-
11 (impl-t)	+	-	+	+	-	+	-	-
12 (mi-no-s)	-	+/-	-	+	+	+/-	+	-
13 (mi-dt)	+	+	+	+	+	+	+	+
14 (mi-rt)	-	-	-	-	+	-	-	-
15 (mi-no)	-	-	-	-	+	-	-	-
16 (def-c)	-	+	-	-	+/-	-	-	-
17 (int-par)	-	+	-	-	+/-	-	-	+
18 (milest)	-	+	-	-	+/-	-	-	-
19 (can-a)	+	+	-	-	+/-	-	-	-
20 (can-c)	-	-	-	-	+/-	+	-	-

pattern	product						
	MQSeries	Forté	Verve	Vis. WF	Changeng.	I-Flow	SAP/R3
1 (seq)	+	+	+	+	+	+	+
2 (par-spl)	+	+	+	+	+	+	+
3 (synch)	+	+	+	+	+	+	+
4 (ex-ch)	+	+	+	+	+	+	+
5 (simple-m)	+	+	+	+	+	+	+
6 (m-choice)	+	+	+	+	+	+	+
7 (sync-m)	+	-	-	-	-	-	-
8 (multi-m)	-	+	+	-	-	-	-
9 (disc)	-	+	+	-	+	-	+
10 (arb-c)	-	+	+	+/-	+	+	-
11 (impl-t)	+	-	-	-	-	-	-
12 (mi-no-s)	-	+	+	+	-	+	-
13 (mi-dt)	+	+	+	+	+	+	+
14 (mi-rt)	-	-	-	-	-	-	+/-
15 (mi-no)	-	-	-	-	-	-	-
16 (def-c)	-	-	-	-	-	-	-
17 (int-par)	-	-	-	-	-	-	-
18 (milest)	-	-	-	-	-	-	-
19 (can-a)	-	-	-	-	-	-	+
20 (can-c)	-	+	+	-	+	-	+

pattern	*standard*				
	BPEL	XLANG	WSFL	BPML	WSCI
Sequence	+	+	+	+	+
Parallel Split	+	+	+	+	+
Synchronization	+	+	+	+	+
Exclusive Choice	+	+	+	+	+
Simple Merge	+	+	+	+	+
Multi Choice	+	−	+	−	−
Synchronizing Merge	+	−	+	−	−
Multi Merge	−	−	−	+/−	+/−
Discriminator	−	−	−	−	−
Arbitrary Cycles	−	−	−	−	−
Implicit Termination	+	−	+	+	+
MI without Synchronization	+	+	+	+	+
MI with a Priori Design Time Knowledge	+	+	+	+	+
MI with a Priori Runtime Knowledge	−	−	−	−	−
MI without a Priori Runtime Knowledge	−	−	−	−	−
Deferred Choice	+	+	−	+	+
Interleaved Parallel Routing	+/−	−	−	−	−
Milestone	−	−	−	−	−
Cancel Activity	+	+	+	+	+
Cancel Case	+	+	+	+	+

Some deficiencies

About **synchronization patterns**

Petri nets have difficulties in dealing with or-join patterns
especially, synchronization merge, discriminator

Examples:
optional booking of flight, train, hotel, rental car, show tickets

number of paper submissions to a conference

Some deficiencies

About **multiple instances**

Petri nets do not provide adequate means to easily describe multiple instances tasks

Examples:
witness statements in processing insurance claim

number of paper submissions to a conference

Some deficiencies

About **nonlocal firing behaviour**

Petri nets do not provide adequate means
to model cancellation patterns
(the firing of a transition should atomically trigger the
cancellation of other tokens in some places,
if there are some tokens there)

Examples:
abort of a commercial transaction

YAWL

Extended workflow net

See [Weske] Def. 4.9 on page 183

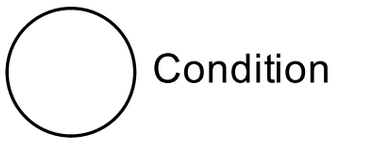
Some transitions are labelled as AND/XOR/OR split

Some transitions are labelled as AND/XOR/OR join

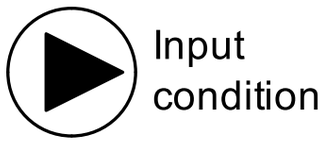
Some transitions are assigned a set of places
to vacuum-clean when firing

Some transitions are assigned a (min, max, threshold)
number of multiple instances to be spawn
(statically or dynamically)

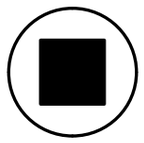
Notational elements of YAWL



Condition



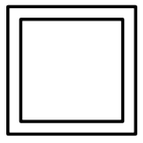
Input condition



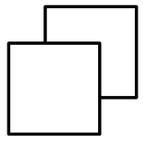
Output condition

Arc

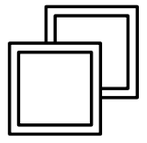
Atomic task



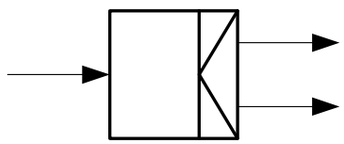
Composite task



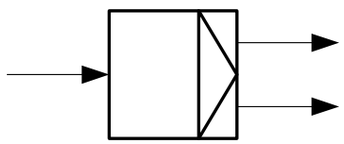
Multiple instances of an atomic task



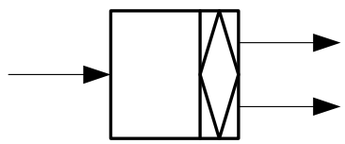
Multiple instances of a composite task



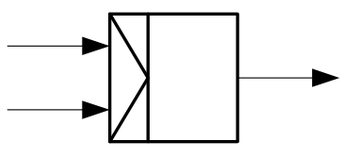
AND-split task



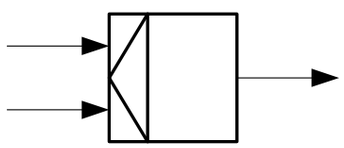
XOR-split task



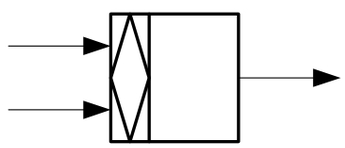
OR-split task



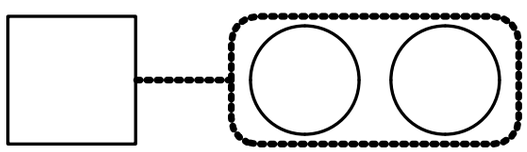
AND-join task



XOR-join task

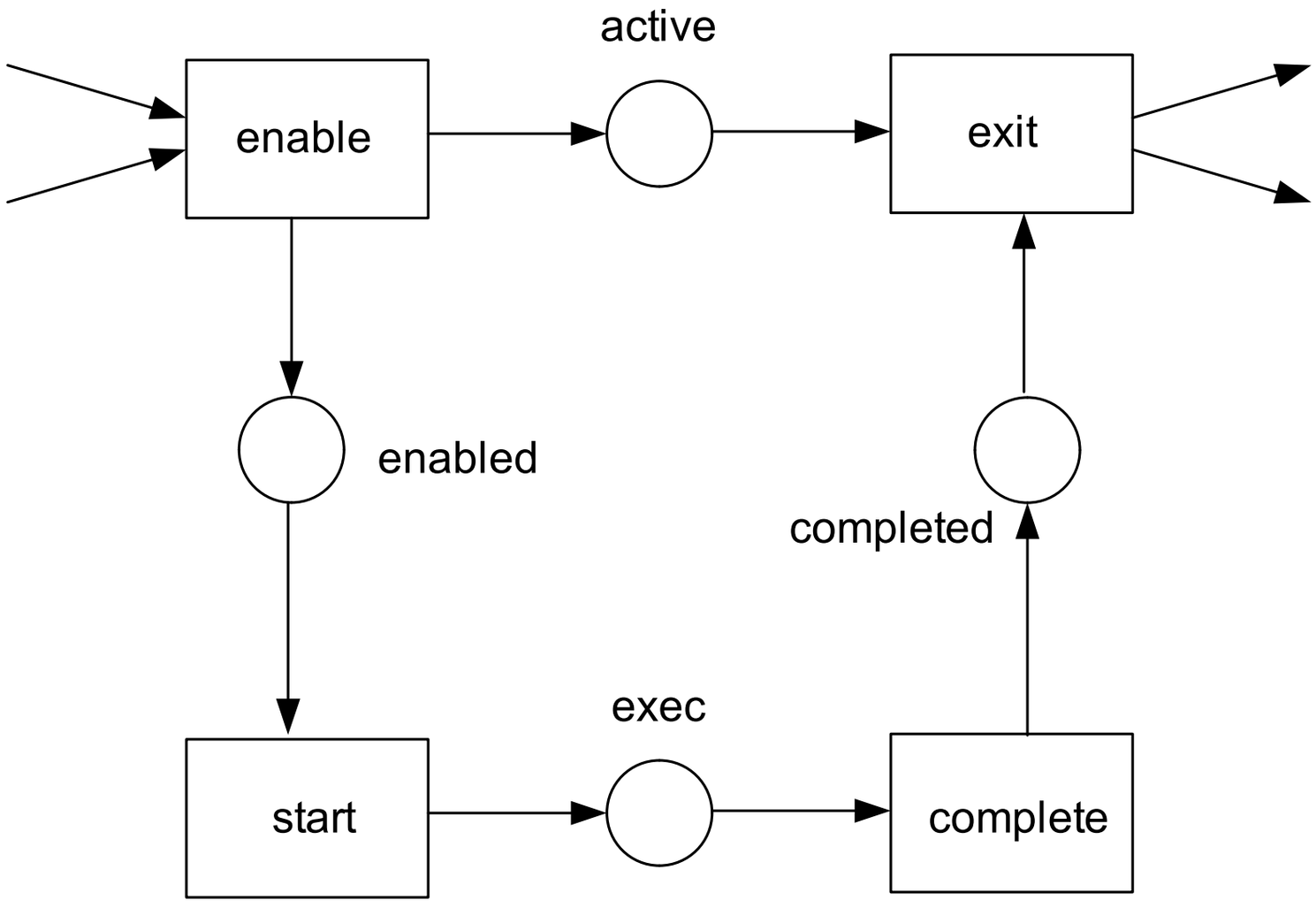


OR-join task

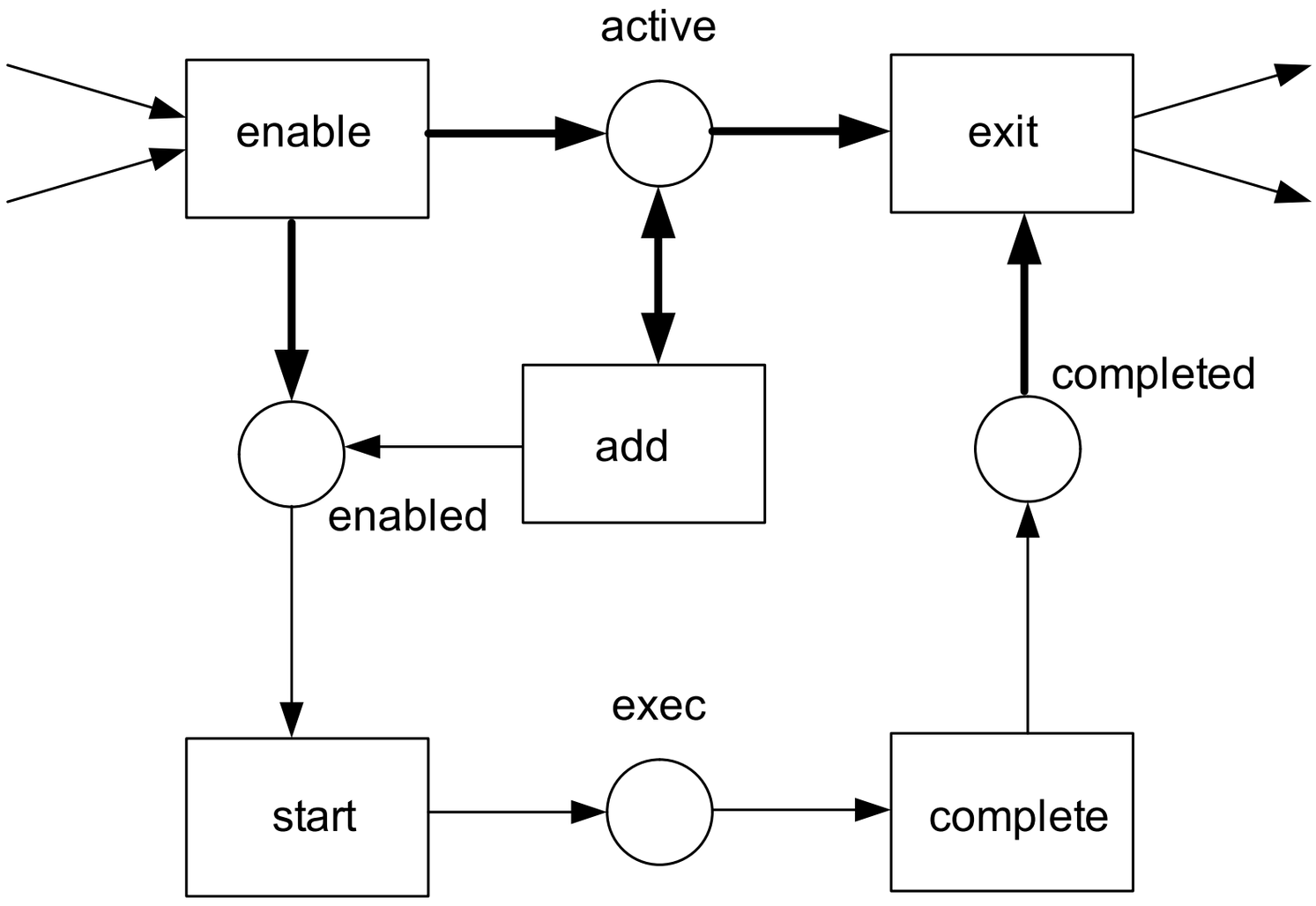


Remove tokens

Phases for single instance tasks

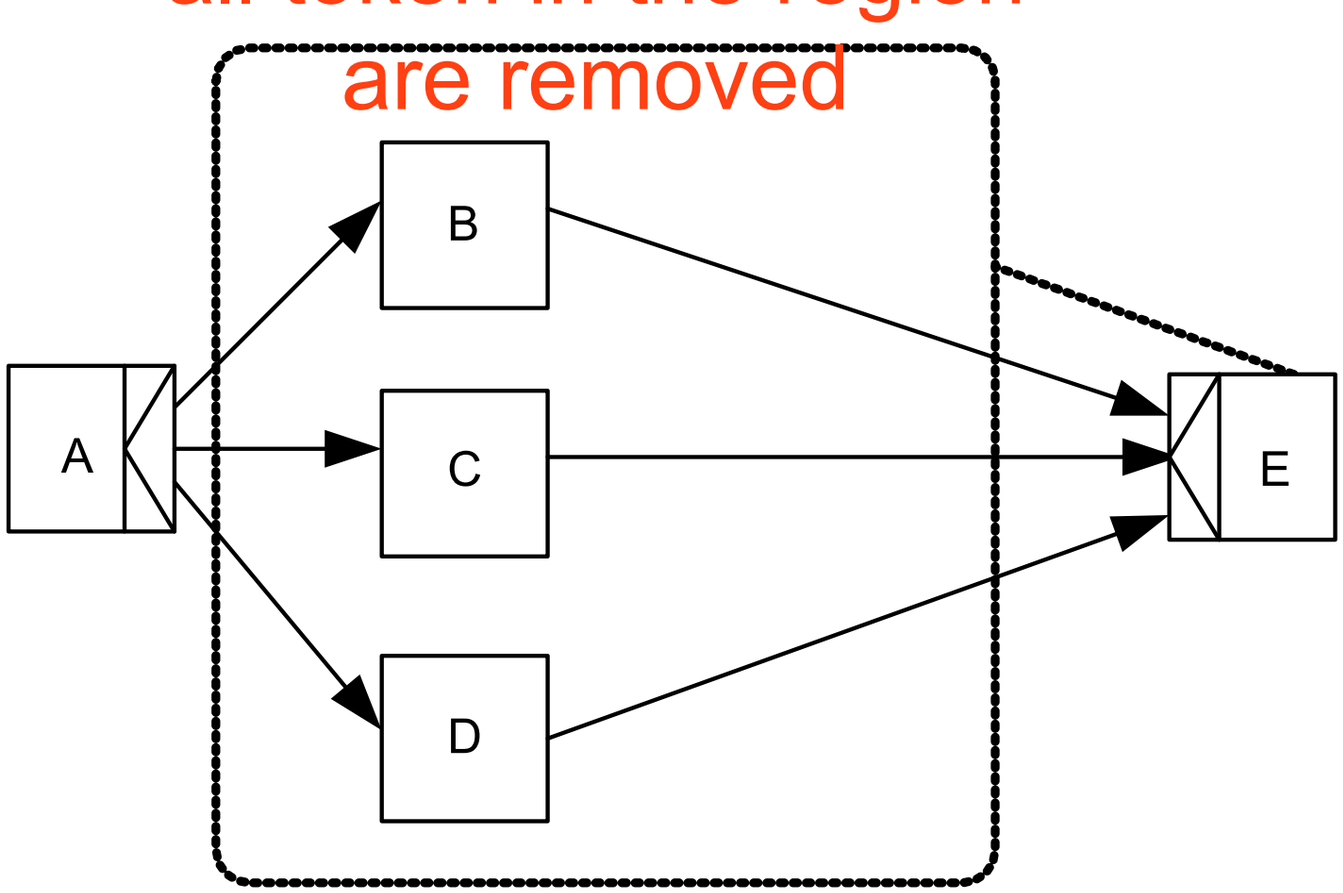


Phases for multiple instance tasks



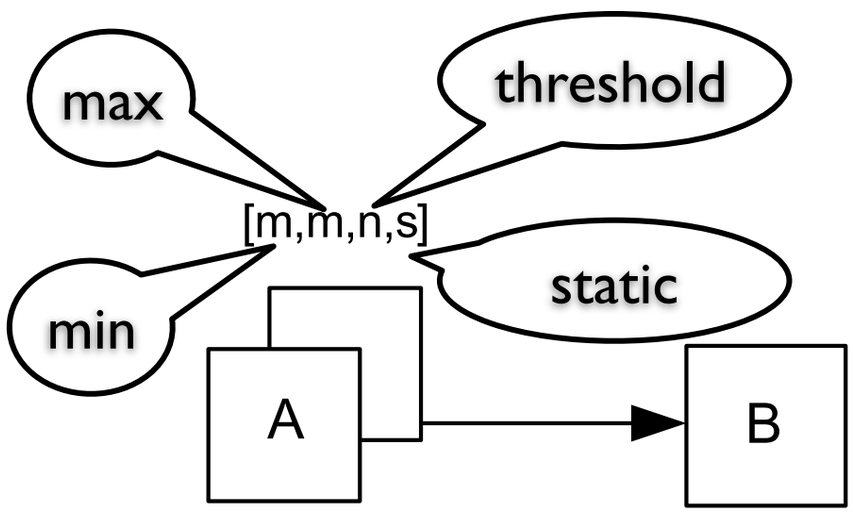
Discriminator (via cancellation region)

when E fires,
all token in the region
are removed



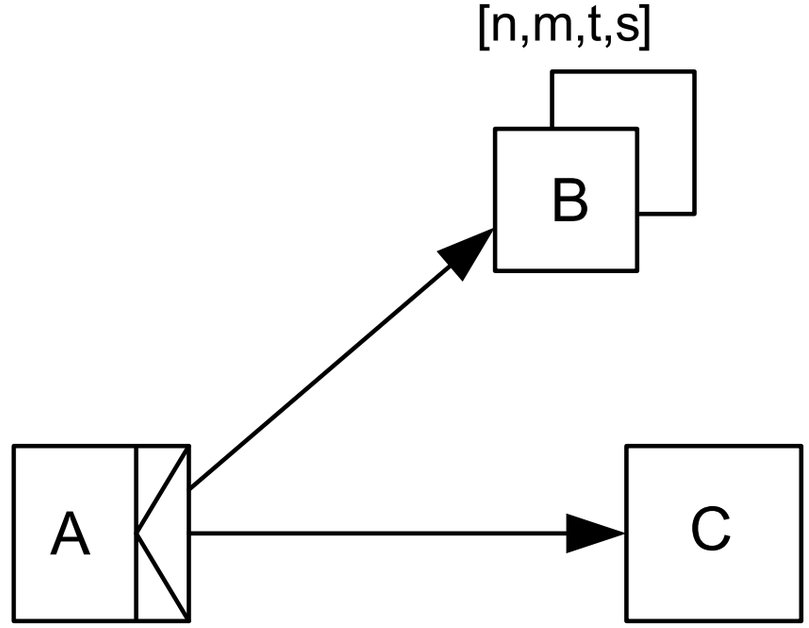
N-out-of-M
(via multiple instances)

Not fully realized
(N-out-of-M of the same activity)

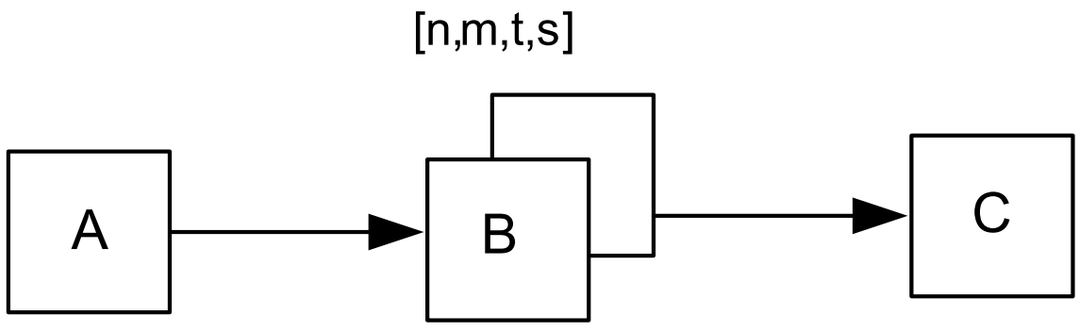


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

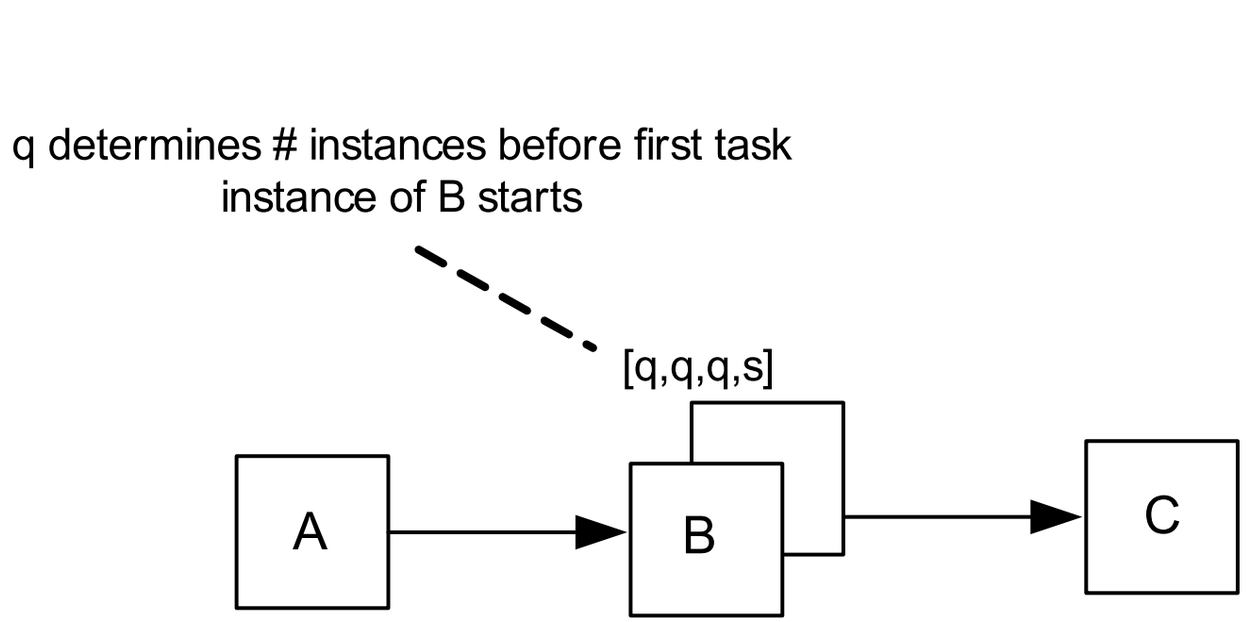
Multiples instances without synchronization



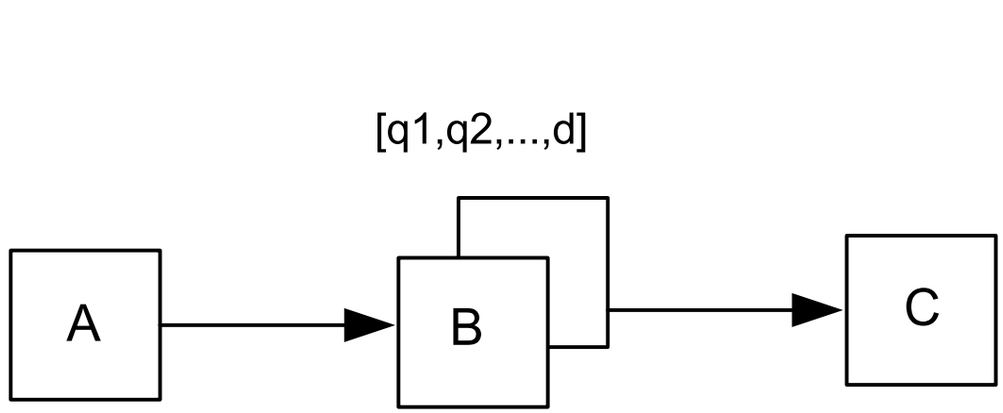
Multiples instances with a priori design time knowledge



Multiples instances with a priori runtime knowledge



Multiples instances without a priori runtime time knowledge



pattern	YAWL
1 (seq)	+
2 (par-spl)	+
3 (synch)	+
4 (ex-ch)	+
5 (simple-m)	+
6 (m-choice)	+
7 (sync-m)	+
8 (multi-m)	+
9 (disc)	+
10 (arb-c)	+
11 (impl-t)	–(iv)
12 (mi-no-s)	+
13 (mi-dt)	+
14 (mi-rt)	+
15 (mi-no)	+
16 (def-c)	+
17 (int-par)	+
18 (milest)	+
19 (can-a)	+
20 (can-c)	+

(i) The synchronizing merge is not supported because the designer has to keep track of the number of parallel threads and decide to merge or synchronize flows (cf. Section 3.2).

(ii) The discriminator is not supported because the designer needs to keep track of the number of threads running and the number of threads completed and has to reset the construct explicitly by removing all tokens corresponding to the iteration (cf. Section 3.2).

(iii) Implicit termination is not supported because the designer has to keep track of running threads to decide whether the case is completed.(about YAWL)

(iv) Implicit termination is not supported because the designer is forced to identify one unique final node. Any model with multiple end nodes can be transformed into a net with a unique end node (simply use a synchronizing merge). This has not been added to YAWL to force the designer to think about successful completion of the case. This requirement allows for the detection of unsuccessful completion (e.g., deadlocks).

(v) Multiple instances with synchronization are not supported by high-level Petri nets (cf. Section 3.1).

(vi) Also not supported, cf. Section 3.1.

(vii) Cancel activity is only partially supported since one can remove tokens from the input place of a transition but additional bookkeeping is required if there are multiple input places and these places may be empty (cf. Section 3.3).

(viii) Cancel activity is not supported because one needs to model a vacuum clearer to remove tokens which may of may not reside in specific places (cf. Section 3.3).

YAWL discussion

Advantages and drawbacks

Graphical presentation closely related to WF nets
Formal semantics (state- transitions model)
Support for Multiple Instance patterns
Support for cancellation region

Unfortunately most soundness properties are undecidable when reset arcs are used

But you can still play with YAWL by downloading it from http://www.yawlfoundation.org/