OBRASCI KONTROLE TOKA BPMN
Informacioni sistemi 2

dr Miloš CVETANOVIĆ
dr Vladimir BLAGOJEVIĆ

Osnovni obrasci kontrole toka

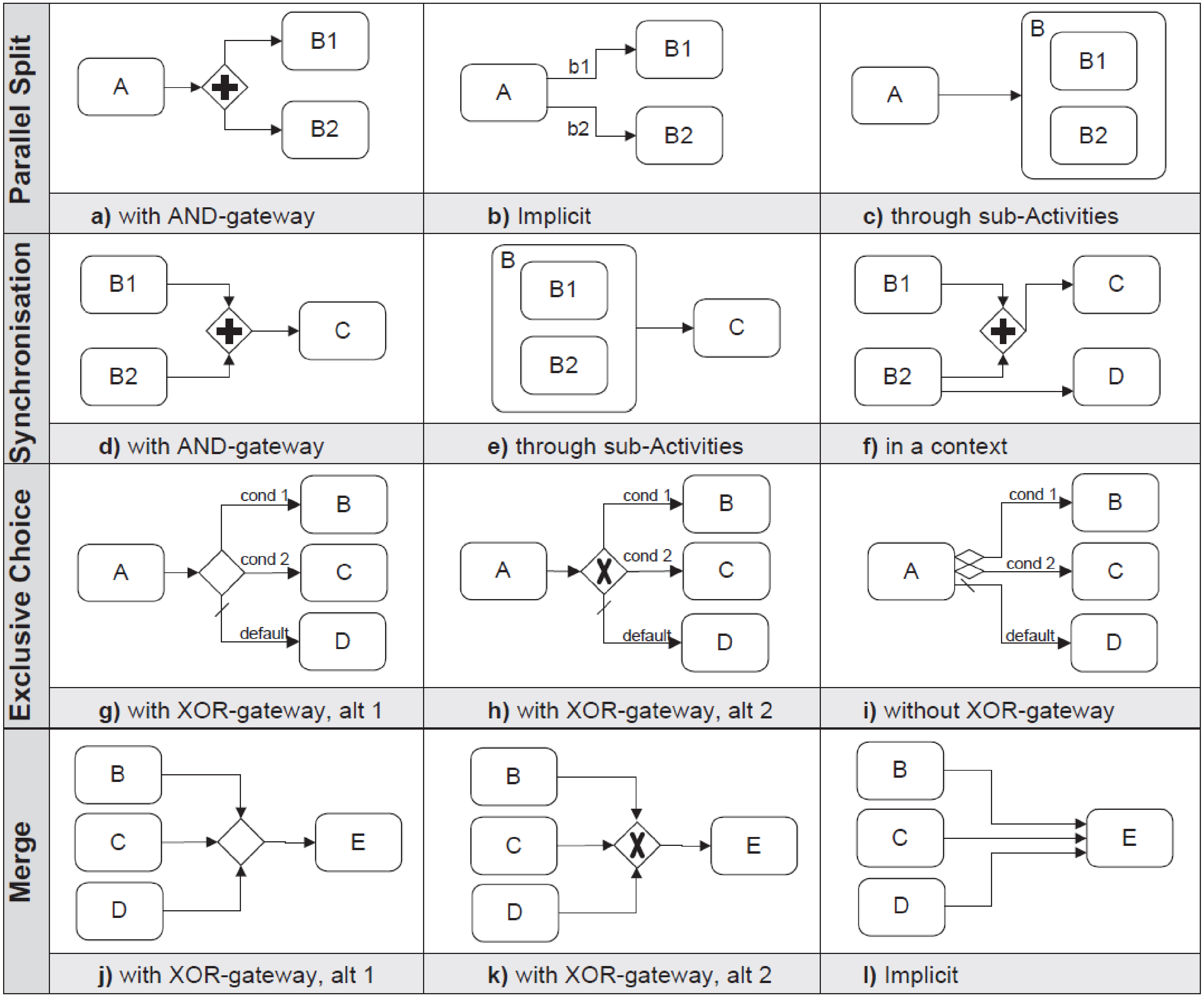
– CP1: Sequence – the ability to depict a sequence of activities;

– CP2: Parallel split – a split of a single thread of control into multiple threads of

– CP3: Synchronization – a convergence of multiple parallel sub-processes/activities into a single thread of control thus synchronizing multiple threads;

– CP4: Exclusive choice – a decision point in a workflow process where one of several branches is chosen;

– CP5: Simple merge – a point in the workflow process where two or more alternative branches come together without synchronization.



a) with AND-gateway

b) Implicit

c) through sub-Activities

d) with AND-gateway

e) through sub-Activities

f) in a context

g) with XOR-gateway, alt 1

h) with XOR-gateway, alt 2

i) without XOR-gateway

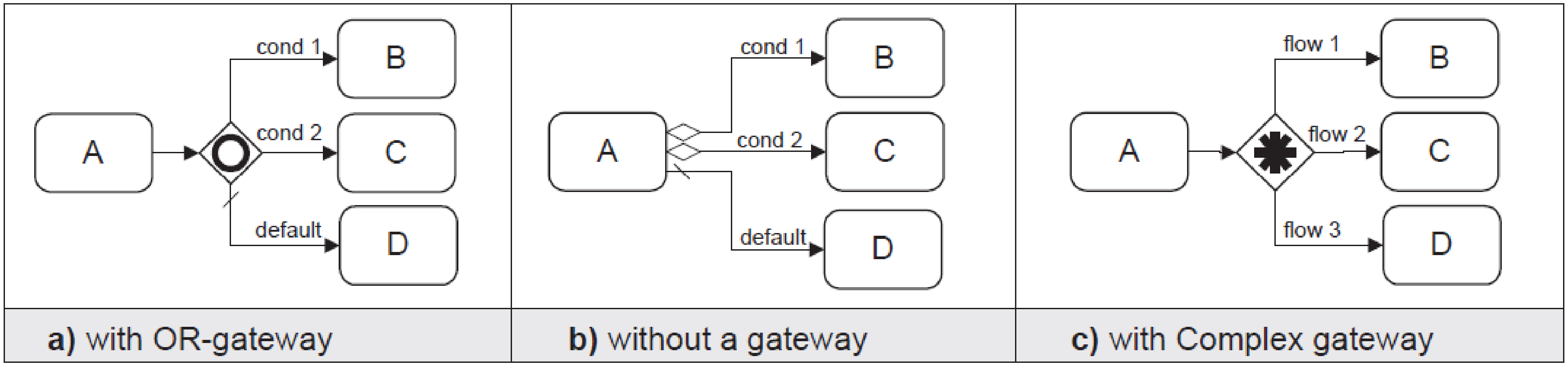
j) with XOR-gateway, alt 1

k) with XOR-gateway, alt 2

l) Implicit

Napredni obrasci grananja i sinhronizacije

– CP6: Multiple Choice – the ability to represent a divergence of the thread of control into several parallel branches on a selective basis;



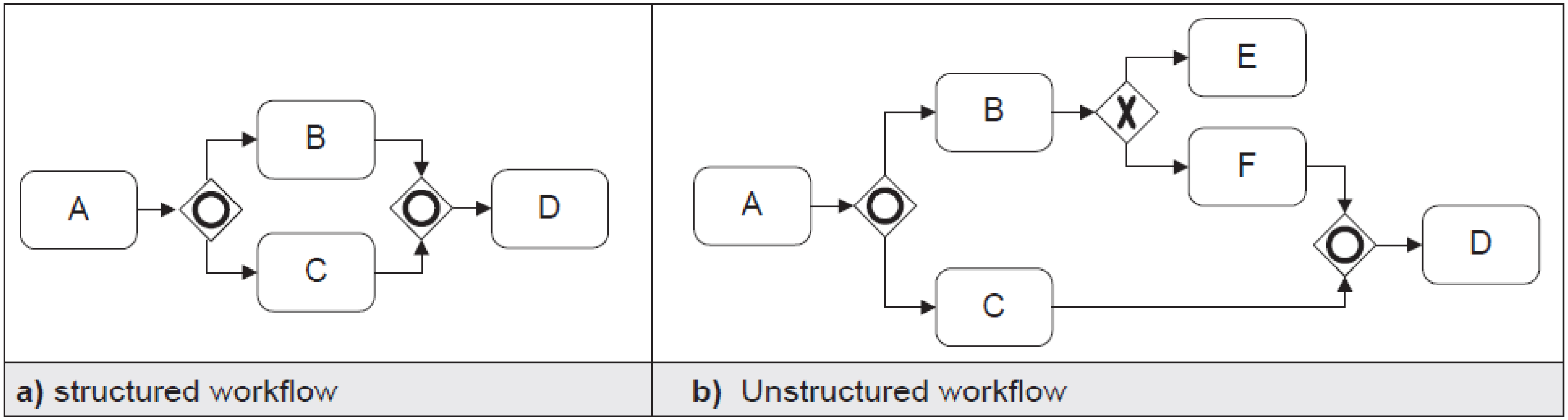
a) with OR-gateway

b) without a gateway

c) with Complex gateway

Napredni obrasci grananja i sinhronizacije

– CP7: Synchronizing merge– the ability to depict the synchronized convergence of two or more alternative branches;

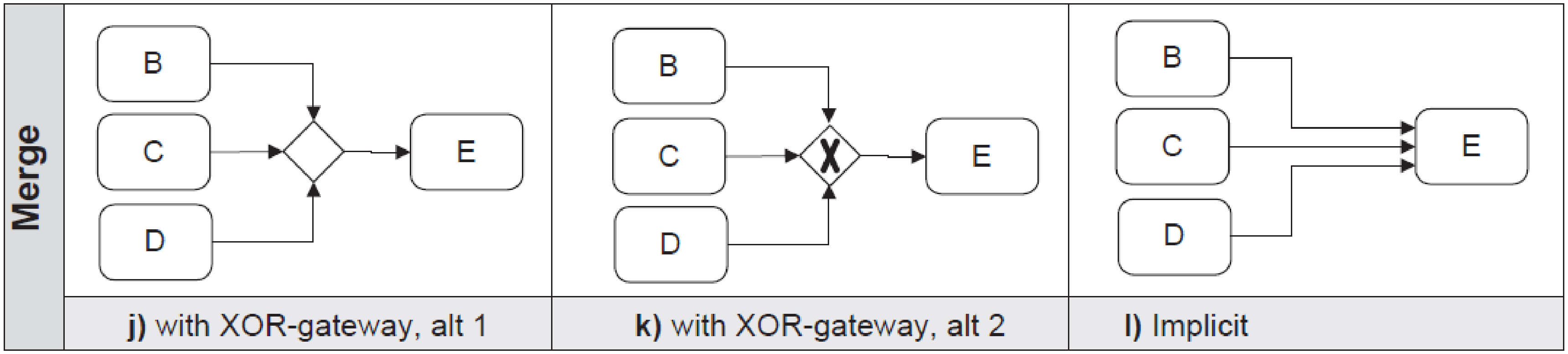


a) structured workflow

b) Unstructured workflow

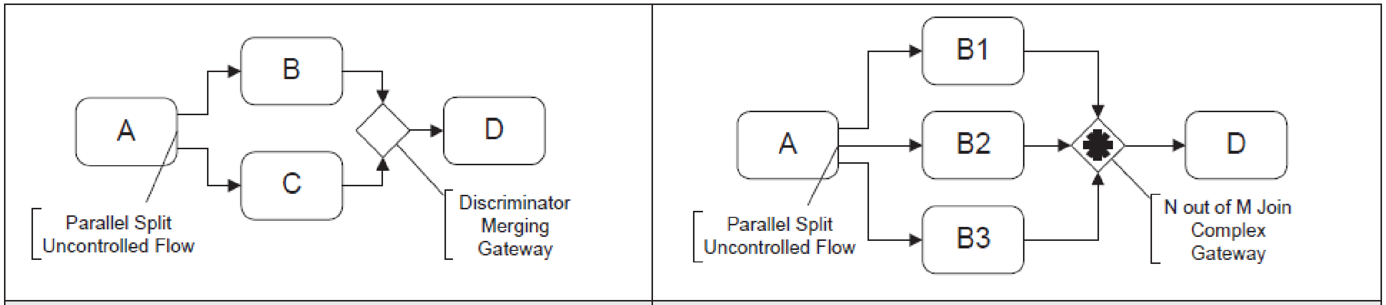
Napredni obrasci grananja i sinhronizacije

– CP8: Multiple Merge – the ability to represent the unsynchronized convergence of two or more distinct branches. If more than one branch is active, the activity following the merge is started for every activation of every incoming branch;

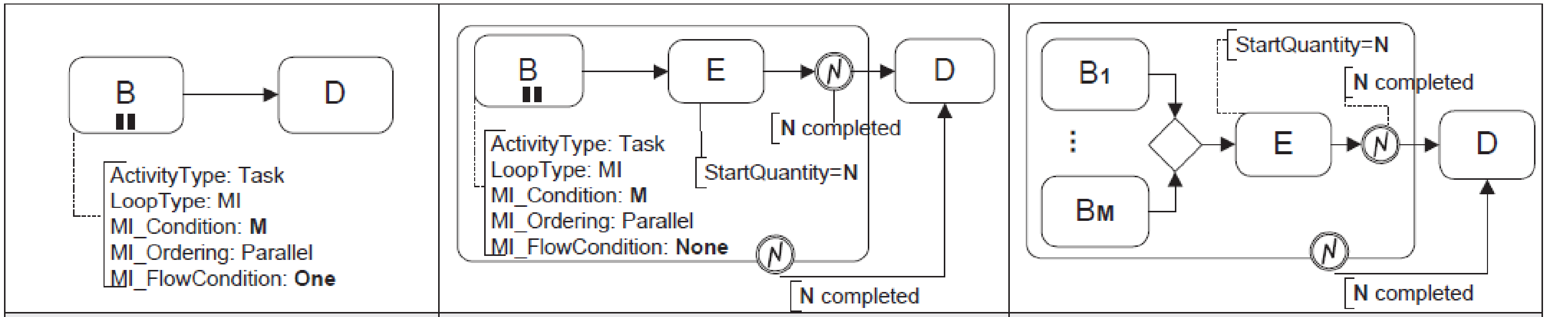


Napredni obrasci grananja i sinhronizacije

– CP9: Discriminator– the ability to depict the convergence of two or more branches such that the first activation of an incoming branch results in the subsequent activity being triggered and subsequent activations of remaining incoming branches are ignored. The discriminator is a special case of the N-out-of-M Join where N=1.



a) copy of Figure 76 from the BPMN specification **b)** copy of Figure 78 from the BPMN specification



a) Discriminator for a MI task **b)** N-out-of-M Join for a MI task **c)** N-out-of-M Join

Strukturni obrasci

- – CP10: Arbitrary cycles – the support for multiple ways of entering end exiting the areas with repetitive activities;

- – CP11: Implicit termination – the ability to depict the notion that a given sub-process should be terminated when there are no remaining activities to be completed.

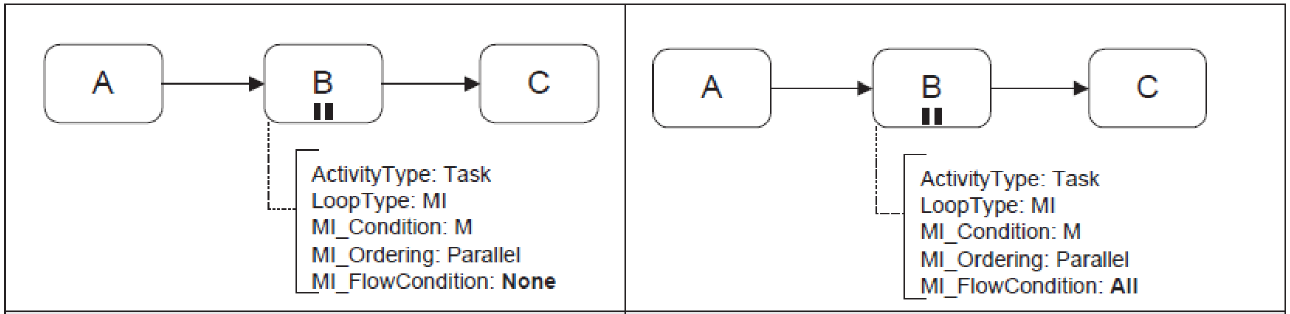
Obrasci sa višestrukim instancama

– CP12: MI without Synchronization – the ability for one case to initiate multiple instances of an activity;

– CP13: MI with a Priori Design Time Knowledge– the ability for one case to initiate multiple instances of an activity and when completed to synchronize them. The number of instances is known at design time;

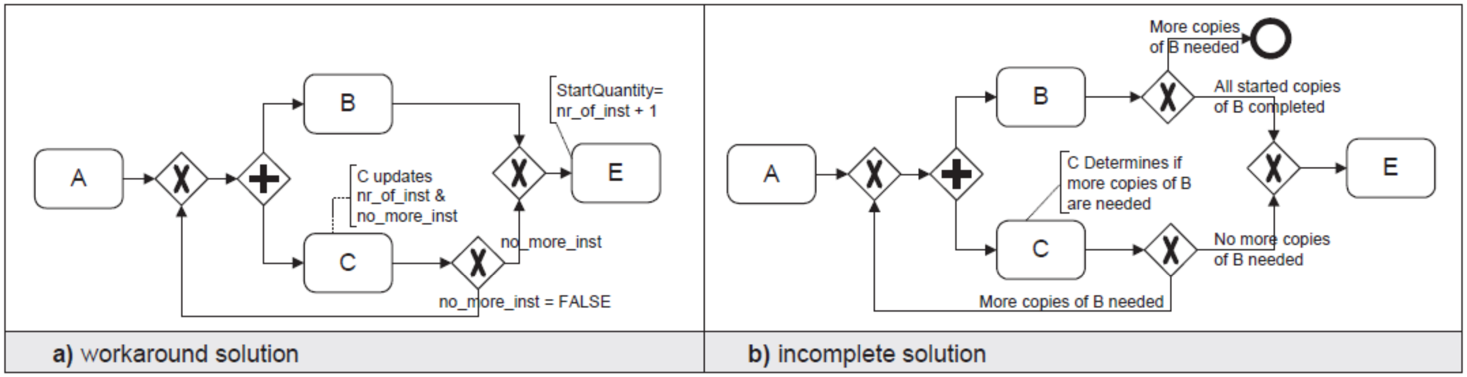
– CP14: MI with a Priori Runtime Knowledge– As the previous pattern, but the number of instances to be created is first known at runtime before the instances must be created;

– CP15: MI without a Priori Runtime Knowledge– As the previous pattern with the extension that the number of instances to be created is not known a priori and new instances can be created even while other instances are executing or have already completed.



a) without Synchronisation

b) with a Priory Design/Runtime Knowledge

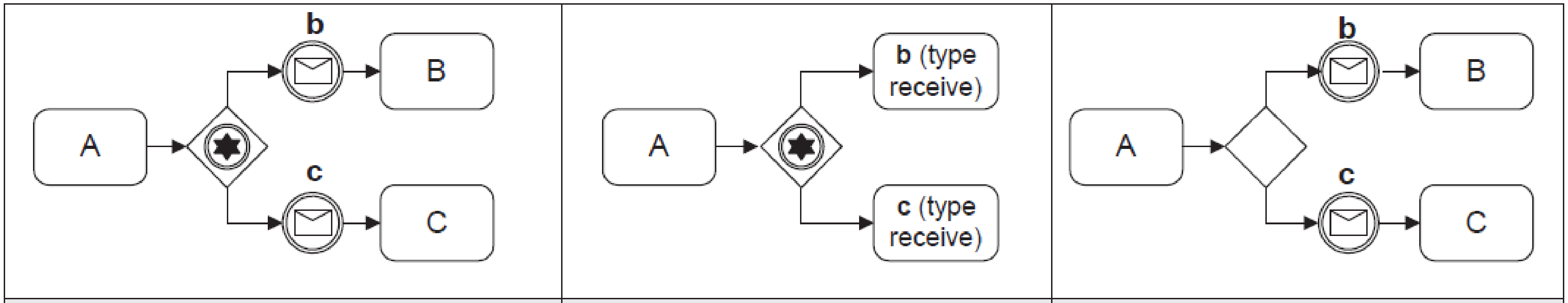


a) workaround solution

b) incomplete solution

Obrasci zasnovani na stanju

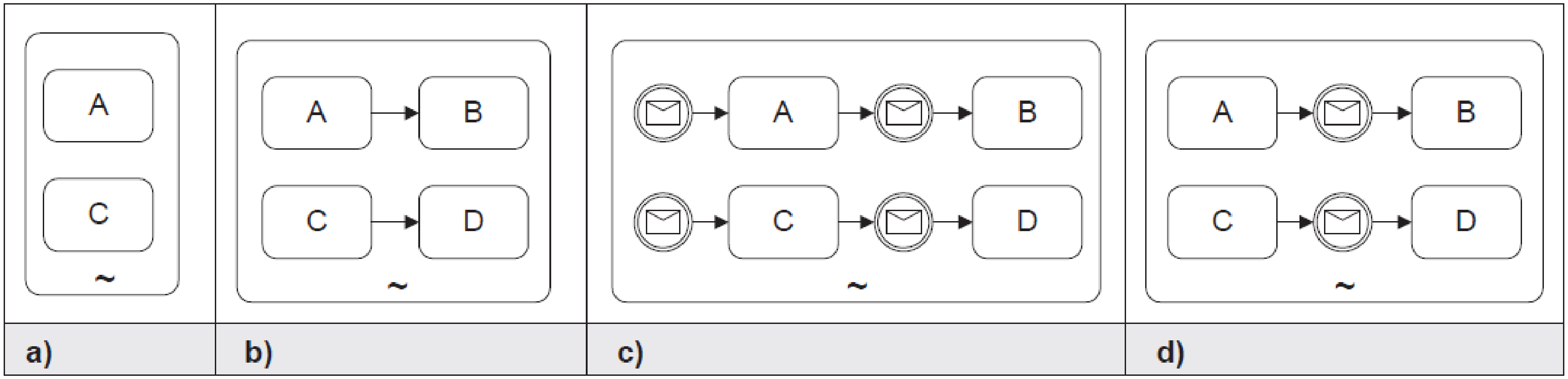
– CP16: Deferred Choice – the ability to depict a divergence point in a process where one of several possible branches should be activated. The actual decision on which branch is activated is made by the environment and is deferred to the latest possible moment;



a) with Event-Based Exclusive Gateway **b)** with Receive actities **c)** incorrect solutions

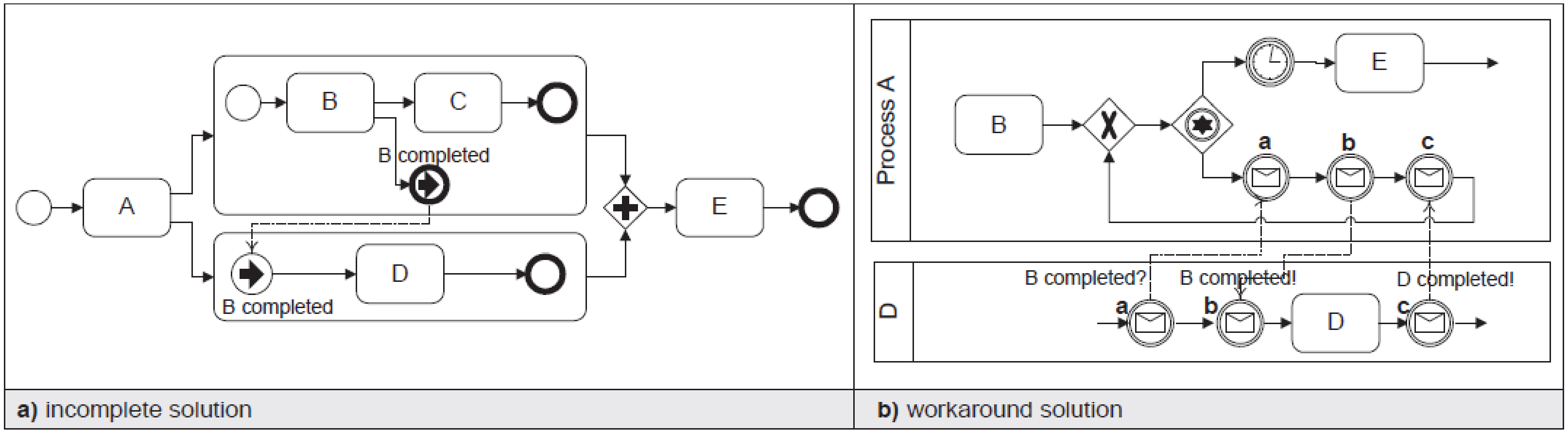
Obrasci zasnovani na stanju

– CP17: Interleaved Parallel Routing – the ability to depict a set of activities that can be executed in arbitrary order;



Obrasci zasnovani na stanju

– CP18: Milestone – the ability to depict that a specified activity cannot be commenced until some nominated state is reached.



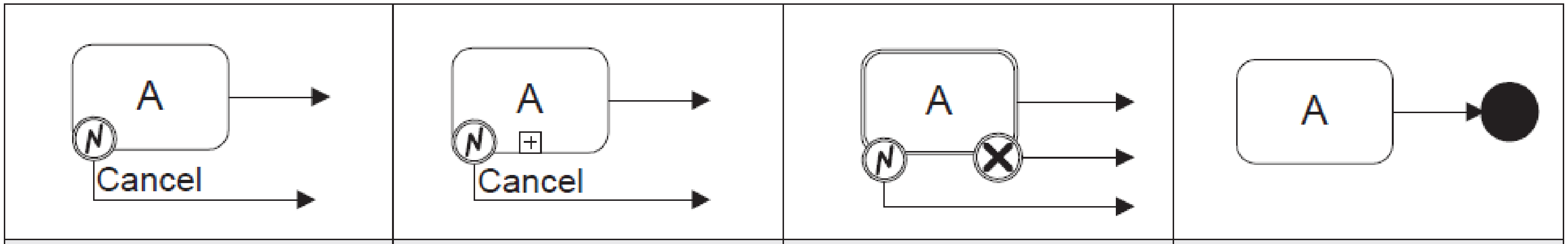
a) incomplete solution

b) workaround solution

Obrasci otkazivanja

– CP19: Cancel activity – the ability to depict that an enabled activity should be disabled in some nominated circumstance;

– CP20: Cancel case – the ability to represent the cancellation of an entire process instance (i.e. all activities relating to the process instance) in some nominated circumstance.



a) Cancel Activity **b)** Cancel Case **c)** Cancel Event **d)** Terminate Event

Pattern	Score	Motivation
1 (seq)	+	Directly supported by linking activities with sequence flow arcs.
2 (par-spl)	+	Supported by AND-split gateway.
3 (synch)	+	Supported by AND-join gateway.
4 (ex-ch)	+	Supported by XOR-split gateway.
5 (simple-m)	+	Supported by XOR-join gateway.
6 (m-choice)	+	Supported in three distinct ways: via an implicit split with conditions on the arcs, an OR-split or a complex gateway.
7 (s-sync-m)	+	Supported through the OR-join gateway.
8 (multi-m)	+	Supported by XOR-join gateway.
9 (s-disc)	+/–	Although support for this pattern is referred to in the BPMN 1.0 specification, it is unclear how the IncomingCondition expression on the COMPLEX-join gateway is specified.
10 (arb-c)	+	Unstructured repetition can be directly supported.
11 (impl-t)	+	Supported by ending every thread with an End Event. When the last token generated by the Start Event is consumed, the process instance terminates.
12 (mi-no-s)	+	Supported via multiple instance task with MI Flow_Condition attribute set to none.
13 (mi-dt)	+	Supported via multiple instance task with MI Flow_Condition attribute set to all.
14 (mi-rt)	+	Supported via multiple instance task with MI_Condition attribute set at runtime to the actual number of instances required.
15 (mi-no)	–	Not supported. There is no means of adding further instances to a multiple instance task once started.
16 (def-c)	+	Supported via an event-based exclusive gateway followed by either intermediate events using message-based triggers or receive tasks.
17 (int-par)	–	Supported for simple tasks via an ad-hoc process but no support for interleaving groups or sequences of tasks.
18 (milest)	–	Not supported. No support for states.
19 (can-a)	+	Supported via an error type intermediate event trigger attached to the boundary of the activity to be cancelled.
20 (can-c)	+	Directly supported by including the entire process in a transaction. Triggering the cancel end event associated with the transaction will effectively terminate all activities associated with a process instance.
21 (str-l)	+	Both while and repeat loops are directly supported by activity looping.
22 (recur)	–	Not supported. No means of specifying recursive composition with a process model.
23 (t-trig)	–	Not supported. Triggers are supported through durable messages.
24 (p-trig)	+	Supported through the use of message events.
25 (can-r)	+/–	Partially supported by enclosing the cancellation region in a sub-process and associating an error event with the sub-process to enable cancellation, however the cancellation region is restricted to a connected subgraph of the overall process model.
26 (can-mi)	+	Achievable via a MI task which has an error type intermediate event trigger at the boundary. When the MI activity is to be withdrawn, a cancel event is triggered to terminate any remaining MI activities.
27 (comp-mi)	–	Not supported. No means of cancelling remaining MI activity instances.
28 (b-disc)	+/–	Although support for this pattern is referred to in the BPMN 1.0 specification, it is unclear how the IncomingCondition expression on the COMPLEX-join gateway is specified.
29 (c-disc)	+	Supported by including the incoming branches and the OR-join in a subprocess that passes control to the following activity once the first branch has completed as well as cancelling the remaining activities in the subprocess using an error type intermediate event.
30 (s-pjoin)	+/–	Although support for this pattern is referred to in the BPMN 1.0 specification, it is unclear how the IncomingCondition expression on the COMPLEX-join gateway is specified.
31 (b-pjoin)	+/–	Although support for this pattern is referred to in the BPMN 1.0 specification, it is unclear how the IncomingCondition expression on the COMPLEX-join gateway is specified.
32 (c-pjoin)	+/–	Although support for this pattern is referred to in the BPMN 1.0 specification, it is unclear how the IncomingCondition expression on the COMPLEX-join gateway is specified.
33 (g-and-join)	+	Supported by the AND-join gateway.
34 (st-pjoin-mi)	+/–	Notionally supported via multiple instance task with MI Flow_Condition attribute set to complex where ComplexMI_FlowCondition is an expression that evaluates to true when exactly M instances have completed and passes on a single token to the following activity. However, it is unclear exactly how the ComplexMI_FlowCondition should be specified.
35 (c-pjoin-mi)	+/–	Notionally achievable via a MI task (as for WCP-34) which has an error type intermediate event trigger at the boundary. Immediately after the MI task is an activity that issues a cancel event to terminate any remaining MI activities. However the same issue arises as for WCP-34 in that it is unclear how the ComplexMI_FlowCondition should be specified.
36 (dyn-pjoin-mi)	–	There is no ability to dynamically add instances to a multiple instance activity.
37 (a-sync-m)	–	The OR-join gateway assumes it will be used in a structured context.
38 (g-sync-m)	–	Not supported. No means of assessing whether an OR-join gateway should fire based on a complete state analysis of the process instance.
39 (crit-sec)	–	Not supported. No means of limiting concurrent execution of a set of activities.
40 (int-rout)	+/–	Indirectly supported via an ad-hoc process containing all of the activities to be interleaved with AdHocOrdering set to sequential however it is unclear what the required AdHocCompletionCondition should be to ensure each activity is executed precisely once.
41 (tm)	+	Directly supported by setting the StartQuantity attribute on activities immediately following the MI activity.
42 (ts)	+	Directly supported by setting the Quantity attribute on the outgoing sequence flow from an activity.
43 (exp-t)	+	Supported via a terminate end event.