

AND-Split

From EPC Standard

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

AND-Split	
Graphical Notation	
There is no image yet, do you want to upload one?	
IsSubClassOf	IsSubClassOf::AND Operator
Successors	hasSuccessor::Function hasPredecessor::Function,
Predecessors	hasPredecessor::Event, hasPredecessor::Operator, hasPredecessor::Process interface
HasIncomingControlFlow	hasIncomingControlFlow::1
HasOutgoingControlFlow	hasOutgoingControlFlow::2, hasOutgoingControlFlow::n
HasResource	hasResource::0
HasAttribute	hasAttribute::0
Edit the Properties	

Brief Information

This is an autogenerated section!

You are not able to edit this information by hand, but by edit the Form (and therefore the properties) of this page. Please refer to the Edit the properties link at the bottom of the info box. {{#show: AND-Split | ?Is a | Intro=The AND-Split is a }}. {{#show: AND-Split | ?contains | Intro=It contains }}. {{#show: AND-Split | ?hasSuccessor | Intro=Possible succeeding element(s) is/are }}. {{#show: AND-Split | ?hasPredecessor | Intro=Previous element(s) can be }}. {{#show: AND-Split | ?hasIncomingControlFlow | Intro=The cardinalities are | Outro= (incoming)}} {{#show: AND-Split | ?hasOutgoingControlFlow | Intro=and | Outro= (outgoing) respectively }}. {{#show: AND-Split | ?refersTo | Intro=The AND-Split refers to }}. {{#show: AND-Split | ?attachedTo | Intro=The AND-Split is attached to a }}.

Short Description

An AND-Split Operator is a subtype of an [AND Operator](#).

It is responsible for splitting the control flow in at least two different branches, which are executed parallelly.[1]

That is why AND-Split has just one incoming arc and multiple outgoing arcs:

$C_{as} = \{c \in C \mid l(c) = \text{and} \wedge |cin| = 1\}$. [2] The AND-split represents a parallel execution. It waits to get the control flow on its incoming arc before allowing the control flow to continue on all its outgoing arcs.[3]

The XOR-Split represents an exclusive choice between one of several alternative branches within the process. As a result, an XOR-Split triggers exactly one of several possible following events. Which of the possible branches is activated depends on the process conditions.[4][5][6]

References

- [*1] N. Cuntz and E. Kindler, "On the Semantics of EPCs: Efficient Calculation and Simulation," *Bus. Process Manag.*, pp. 398–403, 2005.
- [*2] Mendling: *Event Driven Process Chains - Metrics for Process Models*, Volume 6 of the series *Lecture Notes in Business Information Processing*, 2009, pp. 17-57.
- [*3] E. Kindler "On the semantics of EPCs: resolving the vicious circle", *Data & Knowledge Engineering - Special issue: Business process management archive Volume 56 Issue 1*, 2006, pp.23-40.
- [*4] V. Gruhn and R. Laue, "What business process modelers can learn from programmers," *Sci. Comput. Program.*, vol. 65, no. 1, pp. 4–13, 2007.
- [*5] R. Dijkman, "Diagnosing differences between business process models," *Lect. Notes Comput. Sci.* (including *Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics*), vol. 5240 LNCS, pp. 261–277, 2008.
- [*6] Ekkart Kindler "On the semantics of EPCs: resolving the vicious circle", *Data & Knowledge Engineering - Special issue: Business process management archive Volume 56 Issue 1*, 2006, pp.23-40.

[Category:](#)

[Meta Model](#)

This page was last edited on 15 January 2021, at 14:30.

Content is available under [Creative Commons „Zero“ \(Gemeinfreiheit\)](#) unless otherwise noted.

[Privacy policy](#)

[About EPC Standard](#)

[Disclaimers](#)

[Powered by MediaWiki](#)