AND Operator

From EPC Standard

AND Operator

Contents

Short Description

The AND Operator is a subtype of the operator process element. On the one hand it can split the control flow in at least two different branches, which are executed parallel (AND-Split). On the other hand, it merges an AND-Split control flow, when the tokens of the previously activated branches arrive at it (AND-Join).

Syntax

Similar to the OR and XOR Operator, two subtypes of operators exist for the AND Operator: an AND-Split and a AND-Join operator. The split operator splits up the control flow in at least two branches, while the join operator reunites the split control flow. The AND-Split has exactly one ingoing and at least two or more outgoing arcs, while the AND-Join has multiple incoming arcs and just one outgoing arc.[2]–[7] An AND-Split can be triggered by a function or an event as it does not represent a decision but a parallel execution. The AND-Join merges alternative branches, which were generated due to the split of the control flow arc by the AND-Split. It can also be followed by a function or an event, depending on which type of process elements were merged together by the join-operator.[8]

Semantics

Operators express that a function can be started by one or more events, or a function can generate one or more events as a result.

Semantic Representation

Generally, Operators can be either split or join operators:

There are specific AND-Split and AND-Join operators. Following definitions exist:

- Cand = {c ∈ C | l(c) = and} as the set of AND-connectors. [8]
- Cas = {c ∈ C | l(c) = and ∧ |nin| = 1} as the set of AND-split operators.
  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.
- Caj = {c ∈ C | l(c) = and ∧ |nout| = 1} as the set of AND-join operators.
  An AND-join waits to get the control flow on all its incoming arcs before allowing the control flow to continue on its outgoing arc.[9] [10][11]

XML Representation

Edit the XML Code

<source lang="xml">

<xs:complexType name="typeAND"> <xs:sequence> <xs:element name="documentation" type="xs:anyType" minOccurs="0"/> <xs:element name="toolInfo" type="xs:anyType" minOccurs="0"/> <xs:element name="name" type="xs:string" minOccurs="0"/> <xs:element name="description" type="xs:string" minOccurs="0"/> <xs:choice minOccurs="0"> <xs:element name="graphics" type="epml:typeGraphics"/> </xs:choice> <xs:choice minOccurs="0"> <xs:element name="syntaxInfo"> <xs:complexType> <xs:attribute name="implicitType"> <xs:simpleType> <xs:restriction base="xs:string"> <xs:enumeration value="andFunctionEventSplit"/> <xs:enumeration value="andFunctionEventJoin"/> <xs:enumeration value="andEventFunctionSplit"/> <xs:enumeration value="andEventFunctionJoin"/> </xs:restriction> </xs:simpleType> </xs:attribute> </xs:complexType> </xs:element> </xs:choice> <xs:choice minOccurs="0" maxOccurs="unbounded"> <xs:element name="attribute" type="epml:typeAttribute"/> </xs:choice> </xs:sequence> <xs:attribute name="id" type="xs:positiveInteger" use="required"/> <xs:attribute name="defRef" type="xs:positiveInteger" use="optional"/> </xs:complexType> </source>

References

- [*1] M. Fellmann, S. Bittmann, A. Karhof, C. Stolze, and O. Thomas, “Do We Need a Standard for EPC Modelling? The State of Syntactic, Semantic and Pragmatic Quality,” in 5th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA), 2013, pp. 103–116.
- [*2] N. Cuntz and E. Kindler, “On the Semantics of EPCs: Efficient Calculation and Simulation,” Bus. Process Manag., pp. 398–403, 2005.
- [*3] B. F. van Dongen, R. M. Dijkman, and J. Mendling, “Measuring Similarity between Business Process Models,” Adv. Inf. Syst. Eng., vol. 5074, pp. 450–464, 2008.
- [*4] J. Dehnert, J. Dehnert, P. Rittgen, and P. Rittgen, “Relaxed soundness of business processes,” Lect. notes Comput. Sci., pp. 157–170, 2001.
- [*5] E. Kindler, “On the semantics of EPCs: Resolving the vicious circle,” Data Knowl. Eng., vol. 56, no. 1, pp. 23–40, 2006.
- [*6] M. La Rosa, M. Dumas, A. H. M. ter Hofstede, and J. Mendling, Configurable multi-perspective business process models, vol. 36, no. 2. 2011.

- [*7] J. Mendling, M. La Rosa, and a H. M. Hofstede, “(2008) Soundness of EPC Process Models with Objects and Roles . Copyright 2008 (The authors) Soundness of EPC Process Models with Objects and Roles,” vol. 2008, 2008.
- [*8] Mendling: Event Driven Process Chains - Metrics for Process Models, Volume 6 of the series Lecture Notes in Business Information Processing, 2009, pp. 17-57.
- [*9] V. Gruhn and R. Laue, “What business process modelers can learn from programmers,” Sci. Comput. Program., vol. 65, no. 1, pp. 4–13, 2007.
- [*10] 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.
- [*11] 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.

Categories:
Meta Model