Operator

From EPC Standard

Short Description

Operator is a subclass of process element. An operator determines the process's behavior in case its control-flow is split up into or joined from several execution branches. For each subtype exists a Split- and a Join-Operator. The split can either represent an exclusive or non exclusive choice between different alternative branches, or a parallel execution of several process branches. The join merges the split up control flow considering the logic used in the split. As a result, the join operator needs to be of the same type as the logical operator that initially split the control flow. [1]
Following that logic there are three different subtypes of the operator process element:

XOR Operator: Exclusive choice
OR Operator: Nonexclusive choice
AND Operator: Parallel Execution

Operators can also be altered by assigning 0 to n attributes to it.

Semantic Representation

In the following an operator is called connector.

A connector C is a finite and pairwise disjoint set. It is a part of an EPC = (E, F, P, V, l , A) and is represented in the mapping l: C→ {and, xor, or}.

A connector is also defined as a node, as it is part of the union $N = E \cup F \cup P \cup C$.[2]

Following subsets are defined:

- $J = \{c \in C \mid |\bullet c| > 1 \text{ and } |c\bullet| = 1\}$ as the set of join connectors,
  $S = \{c \in C \mid |\bullet c| = 1 \text{ and } |c\bullet| > 1\}$ as the set of split-connectors.[3][4]

Following requirements are made on connectors so an EPC can be called relaxed syntactically correct:

- There are no connector cycles, i.e. $\forall a, b \in C$: if $a \neq b$ and $a \rightarrow c \rightarrow b$, then $b \rightarrow c \rightarrow a$ does not exit.

- Connectors have one incoming and multiple outgoing arcs or multiple incoming and one outgoing arc. $\forall c \in C : (|\bullet c| = 1 \wedge |c\bullet| > 1) \vee (|\bullet c| > 1 \wedge |c\bullet| = 1)$.

- Connectors must have either functions, process interfaces, or fe-connectors in the preset and events or fe-connectors in the postset;
  or events or ef-connectors in the preset and functions, process interfaces, or ef-connectors in the postset:
  $\forall c \in C: (\bullet c \subseteq (F \cup P \cup C_{FE}))$

∧ c• ⊆ (E ∪ CFE)∨ (•c ⊆ (E ∪ CEF)
∧ (c• ⊆ (F ∪ P ∪ CEF)).[5]

- After an event, no xor/or connector is allowed.[6]

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] 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, p. 28.

- [*3] Van der Aalst, "Formalization and verification of event-driven process chains" Information and Software Technology 41, 1999, pp. 639-650
- [*4] Mendling: Event Driven Process Chains - Metrics for Process Models, Volume 6 of the series Lecture Notes in Business Information Processing, 2009, pp. 17-57.
- [*5] K. van Hee, O. Oanea, N. Sidorova, "Colored Petri Nets to Verify Extended Event-Driven Process Chains", OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", 2005, pp. 183-201.

- [*6] M. Nüttgens, F. J. Rump "Syntax und Semantik Ereignisgesteuerter Prozessketten", Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, Proceedings des GI-Workshops und Fachgruppentreffens, 2002, pp. 65-77.

Category:
Meta Model