Corrigendum to “BPMN 2.0 OR-Join Semantics: Global and local characterisation” [Information Systems 105 (2022), 101934]

Asvin Goel[a]

[a] *Kühne Logistics University, Großer Grasbrook 17, Hamburg, 20457, Hamburg, Germany*

The original article[1] studies the activation behavior of OR-joins according to the following specification of BPMN 2.0:

The **Inclusive Gateway** *is activated if*

- *At least one incoming* **Sequence Flow** *has at least one token and*
- *For every directed path formed by sequence flows that*
  - *(i) starts with a* **Sequence Flow** *f of the diagram that has a token,*
  - *(ii) ends with an incoming* **Sequence Flow** *of the inclusive gateway that has no token, and*
  - *(iii) does not visit the* **Inclusive Gateway**.
- *There is also a directed path formed by* **Sequence Flow** *that*
  - *(iv) starts with f,*
  - *(v) ends with an incoming* **Sequence Flow** *of the inclusive gateway that has a token , and*
  - *(vi) does not visit the* **Inclusive Gateway**.

In this specification of the OR-join behavior, the only reasonable interpretation of the term *visit* implies that a path ending at the node is not automatically considered to be visiting the node because such a “visit” would immediately lead to a contradiction of conditions (v) and (vi).

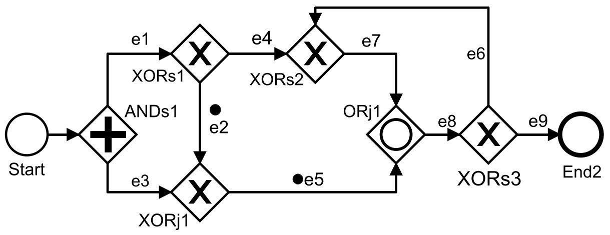

Figure 1: OR-join activation

Figure 4 of the original article provides several examples illustrating the activation behavior of OR-joins according to the BPMN specification. The last example is shown in Figure 1. For this example, the following explanation is given in the article:

> *In this case, the activation of $ORj1$ depends on the token in e_2. It is simultaneously in a path ending with a non-empty edge incoming in ORj1 (i.e., (e_2, e_5)), and in a path ending with a non-empty edge incoming in $ORj1$ (i.e., $(e_2, e_5, e_8, e_6, e_7)$). Despite there exists a path ending with an incoming sequence edge of the inclusive gateway that has no token, it visits the gateway. Thus, condition (iii) is violated, and the $ORj1$ results inactive also in this case.*

The correct assessment of the activation of $ORj1$ is that:

- there is an incoming sequence flow with a token, i.e., e_5,
- the set of directed paths satisfying condition (i) is the set of all paths starting with e_2 or with e_5,
- the set of directed paths satisfying condition (ii) is the set of all paths ending with e_7,
- all directed paths starting either with e_2 or e_5 and ending with e_7 must include (e_8, e_6, e_7) at least once.
- condition (iii) is violated for every directed path satisfying conditions (i) and (ii),
- the set of directed paths satisfying conditions (i) to (iii) is empty and, thus, conditions (iv) to (vi) are not applicable.

Concludingly, $ORj1$ is activated.

It must be noted that the OR-join behaviour of the BPMN specification is based on the semantics for safe workflow graphs proposed by Völzer (2010). In such graphs, all tokens are associated to arcs, and unlike in the discussed example, race conditions can not occur. The BPMN token flow logic also allows tokens to reside at nodes, i.e., at activities or catching events. Strictly following the OR-join behaviour given by the specification would not make sense, because condition (i) only considers paths starting with sequence flows having a token and disregards paths starting at a node with a token. Simple work arounds would be to either also consider paths starting at a node with a token, or to work on an auxiliary graph in which each node that may hold a token is replaced by an auxiliary sequence flow and tokens residing at such nodes are mapped to the respective sequence flows.

[1] DOI of original article: https://doi.org/10.1016/j.is.2021.101934

Acknowledgements

Thanks to Hagen Völzer and the authors of the original article for validating the corrected assessment. Many thanks to the editors for their support in preparing this corrigendum.

References

Völzer, H. (2010). A New Semantics for the Inclusive Converging Gateway in Safe Processes. In: Hull, R., Mendling, J., Tai, S. (eds) Business Process Management. BPM 2010. Lecture Notes in Computer Science, vol 6336. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15618-2_21