

A Motivation for Multiple Activity Instantiation in BPEL4WS Processes

Jan Mendling, Mark Strembeck, and Gustaf Neumann

Department of Information Systems, New Media Lab
Vienna University of Economics and BA
{jan.mendling|mark.strembeck|gustaf.neumann}@wu-wien.ac.at

Abstract We suggest to extend BPEL4WS with structured activities for multiple instantiation since this is a crucial feature of a business process modelling language. In particular, we propose to extend BPEL4WS with a `collect` and a `broadcast` activity to model multiple instances as well as `list` data structures to handle similar messages of multiple parties that act according to the same role.

1 Introduction

The Business Process Execution Language for Web Services (BPEL4WS or BPEL) [1] is the de facto standard for XML-based business process modelling. Although it provides a rich set of primitives to specify Web Service compositions, it does not support multiple instantiation (see also [2]). However, there is a need to model certain activities that are executed multiple times within the same process instance without knowing the number of parallel executions a priori. This is especially the case for interorganizational business processes that often include $1 : n$ interactions. Typically, they can be divided into two parts as the example of an auction process illustrates:

1. A set of potential partners is created (see e.g. [3]). In an auction process each bidder can be regarded as a potential business partner. The bidder with the best offer is chosen as a partner for further interaction.
2. The offerer and the auction winner continue the process in a bilateral way. The winner receives a bill, and the offerer initiates the shipment.

Auctions are only one case of such $1 : n$ situations in business processes. The interaction between a teacher and multiple students or request for quotes are further examples. The Business Process Modeling Notation (BPMN) [4] provides a dedicated control flow element for multiple instantiation in order to allow simple modelling of such interactions. However, BPEL does not support corresponding language constructs. Although corresponding work-arounds exist [2], they are too complicated in the general case for interorganizational business process modelling. Nevertheless, the second phase where the multi-party process converges to a bilateral interaction can be modelled with BPEL in a straight forward manner.

2 BPEL4WS and Multiple Instantiation

In order to allow for multiple instance modeling in BPEL, at least the following issues have to be addressed:

- *Structured Activities for Multiple Instantiation:* From our experiences two kinds of structured multiple instance activities are needed to extend BPEL:
 1. An activity to model the receipt of multiple messages of different parties acting as the same `partnerRole`; as for example in an auction where multiple parties act as bidders and send bid messages. We propose a new BPEL activity called `collect` to address this need.
 2. Multiple messages need to be sent to a set of external parties who were identified via a previous `collect`. In the auction example each bidder receives a notification after the auction. For this purpose, we propose to define a new BPEL element called `broadcast`.
For both activities, synchronization conditions need to be modelled.
- *List of Messages:* Furthermore, we need to address data handling for message exchanges with multiple parties. For this purpose, we propose to extend BPEL with `lists` and list related operations. For example different messages could be appended to a list via an `add()` operation. In a `broadcast` operation a `next()` operation will be helpful to pick up the next message for processing.
- *List of External Parties:* In a multiple instantiation activity different partners may participate in the same role. This implies the need for some correlation mechanism to retrieve messages of individual parties and vice versa.

3 Conclusion and Future Work

In this position paper, we motivated the need for a native support of multiple instantiation in BPEL. Furthermore, we proposed to define two additional language elements, the `collect` and the `broadcast` activity. In our opinion, such modeling constructs are a prerequisite to provide for a simpler alignment of business processes and their BPEL representation. One important goal in our future work is to implement a BPEL process engine including the multiple instantiation extensions outlined in this paper.

References

1. Andrews, T., Curbera, F., Dholakia, H., Golland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services, Version 1.1. Specification, BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems (2003)
2. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M.: Analysis of Web Service Composition Languages: The Case of BPEL4WS. In: Proceedings of Conceptual Modeling - ER 2003. Volume LNCS 2813. (2003) 200–215
3. Bakos, Y.: The emerging role of electronic marketplaces on the internet. Communications of the ACM **41** (1998) 35–42
4. White, S.A.: Business Process Modeling Notation. Working Draft, BPML.org (2003)