Horizontal and Vertical Business Process Model Integration*

(Abstract)

Klaus-Dieter Schewe[1,2]

[1] Software Competence Center Hagenberg, Austria
kd.schewe@scch.at
[2] Johannes-Kepler-University Linz, Austria
kd.schewe@cdcc.faw.jku.at

Modelling information systems in general is a complex endeavour, as systems comprise many different aspects such as the data, functionality, interaction, distribution, context, etc., which all require different models. In addition, models are usually built on different levels of abstraction and the switch from one of these levels to another one may cause mismatches. Horizontal model integration refers to the creation of system models by successive enlargement, whereas vertical model integration refers to the systematic, seamless refinement process of high-level abstract (conceptual) models down to running systems. Our research on horizontal and vertical model integration has concentated on business process models. The results will be reported in the monograph [5].

With respect to horizontal model integration several submodels have to be defined and integrated. The common model to start with addresses the *control flow model*, i.e. a business process is decsribed in an abstract way by a set of activities and gateways, the latter ones for splitting and synchronisation, plus start and termination events. Depending on whether one, all or an arbitrary selection of (outgoing) paths are enabled in splitting gateways, we adopt the common distinction between XOR-, AND- and OR-gateways with an analogous distinction for the synchronisation gateways. However, this terminology is in a sense misleading, as there need not be a well-nested structure, in which a splitting-gateway corresponds to exactly one synchronisation gateway. This is one of the reasons, why we formalise the semantics of each of the constructs by means of Abstract State Machines (ASMs, [2]). As a state-based rigorous method, ASMs support the unambiguous capture of the semantics of OR-synchronisation [1]. Furthermore, on grounds of ASMs necessary subtle distinctions and extensions to the control flow model such as counters, priorities, freezing, etc. can be easily integrated in a smooth way. All constructs found in a control flow model are supposed to be exceuted in parallel for all process instances.

The control flow model is then extended by a *message model* and an *event model*. For this refinement in ASMs – mainly conservative extensions – are

* The research reported in this paper was supported by the European Fund for Regional Development as well as the State of Upper Austria for the project *Vertical Model Integration* within the program “Regionale Wettbewerbsfähigkeit Oberösterreich 2007-2013”.

H. Decker et al. (Eds.): DEXA 2013, Part I, LNCS 8055, pp. 1–3, 2013.

exploited. In particular, the ground specification of firing conditions that depend on the state of the control flow, data, events and resources and actions that update this state [3] requires that only conditions and actions are refined. While messages are easily captured by means of specifications of sender and receiver, it becomes more subtle to define details such as synchronised vs. asynchronised messaging, delivery failure, rejection, message box overflows, etc. In H-BPM the ASM-based specification of messaging from S-BPM [4] has been adopted. For the event model it is necessary and sufficient to specify what kind of events are to be observed, which can be captured on the grounds of monitored locations in ASMs, and which event conditions are to be integrated into the model.

The next horizontal extensions concern the *actor model*, i.e. the specification of responsibilities for the execution of activities (roles), as well as rues governing rights and obligations. This leads to the integration of deontic constraints [6], some of which can be exploited to simplify the control flow [7]. In this way subtle distinctions regarding decision-making responsibilities in BPM can be captured. Horizontal model integration through refinement is then extended towards an *interaction model* and a *data model*. For this, an abstract dialogue model is adopted [8] capturing interaction by means of operations on views that are defined on top of a database schema. In this way the data model results from view integration, but global consistency has to be addressed, as a global database infers dependencies between activities that are not visible on the control flow level.

Finally, an *exception handling model* has to be integrated to complete the horizontal integration picture. This is still in a preliminary state in H-BPM. Overall, the general idea is that an exception is a disruptive event that requires partial rollback and depending on the state the continuation with a different subprocess.

Vertical integration is achieved by further refining the involved ASMs in a development process that is targeting the executable specification of a workflow engine that is enriched with components for data and dialogue handling and exception processing. Throughout the process rigorous quality assurance methods have to be applied.

References

1. Börger, E., Sörensen, O., Thalheim, B.: On defining the behavior of OR-joins in business process models. Journal of Universal Computer Science 15(1), 3–32 (2009)
2. Börger, E., Stärk, R.: Abstract State Machines. Springer, Heidelberg (2003)
3. Börger, E., Thalheim, B.: Modeling workflows, interaction patterns, web services and business processes: The ASM-based approach. In: Börger, E., Butler, M., Bowen, J.P., Boca, P. (eds.) ABZ 2008. LNCS, vol. 5238, pp. 24–38. Springer, Heidelberg (2008)
4. Fleischmann, A., et al.: Subject-Oriented Business Process Management. Springer, Heidelberg (2012)
5. Kossak, F., et al.: The Hagenberg Business Process Modelling Method H-BPM (2014) (forthcoming)

6. Natschläger, C., Kossak, F., Schewe, K.D.: BPMN to Deontic BPMN: A trusted model transformation. Journal of Software and Systems Modelling (to appear, 2013)
7. Natschläger, C., Schewe, K.D.: A flattening approach for attributed type graphs with inheritance in algebraic graph transformation. Electronic Communications of the EASST 47, 160–173 (2012)
8. Schewe, K.D., Schewe, B.: Integrating database and dialogue design. Knowledge and Information Systems 2(1), 1–12 (2000)