Visualization of Complex BPEL Models

Benjamin Albrecht[1], Philip Effinger[1], Markus Held[2], Michael Kaufmann[1], and Stephan Kottler[1,2]

[1] Parallel Computing, Universität Tübingen, Germany
[2] Symbolic Computation Group, Universität Tübingen, Germany
{albrecht,effinger,mheld,mk, kottlers}@informatik.uni-tuebingen.de

1 Introduction

In this work, we present our approach for producing layouts of complex workflows given in the *Business Process Execution Language* (BPEL) [1]. BPEL is a verbose, hierarchical workflow language containing nested, alternative and concurrent execution paths. Our approach enhances the Sugiyama algorithm [2] by introducing special paths, which are constrained to be drawn in parallel, and hence, orthogonally to the layers in the Sugiyama model. To prove the feasibility of our approach, we have developed an extension to the collaborative BPEL development system HOBBES [3] [4]. Collaboration enhances the need for visualizations of complex workflow models, as team members have to coordinate their activities.

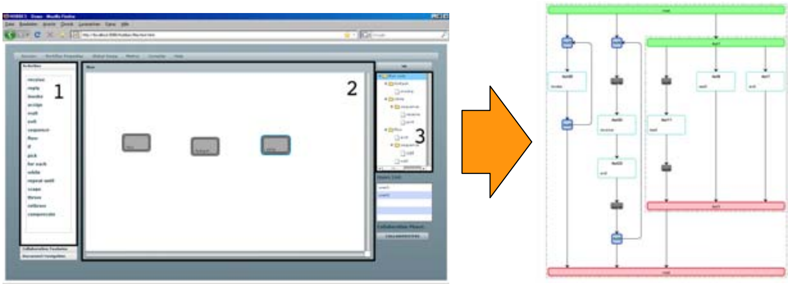
2 Collaborative Workflow Development with HOBBES

The Business Process Execution Language The XML-based *Business Process Execution Language.* (BPEL) has become the *de-facto* standard for business workflows. It is a key element of the *Service Oriented Architecture* (SOA) [1]. BPEL control flow is a mixture of block-oriented and graph-oriented elements. Atomic tasks like service invocations or waiting commands are called *basic activities*. Control structures are expressed as *structured activities* (e.g. Sequence, If, While), which can contain child activities. Concurrency can be modeled using the structured activities Flow and ForEach. Flow allows the definition of Directed Acyclic Graphs of activities while ForEach loops may be marked as "parallel". Expressions, given in the XPath language, are used for conditionals, for triggering links between activities and for assignments.

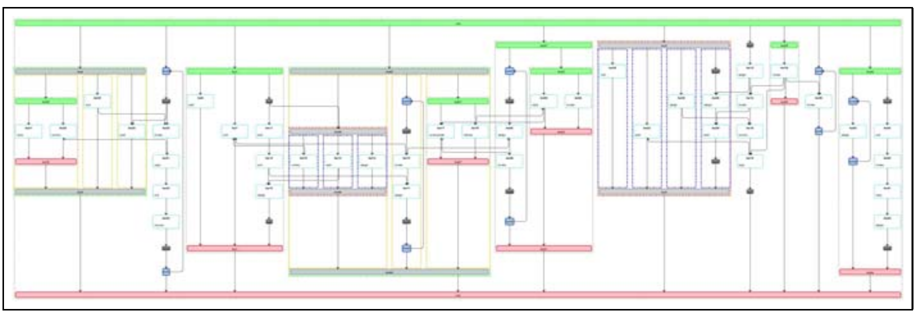
The HOBBES *system.* HOBBES is a web-based BPEL development system, which enables synchronous collaboration sessions [3, 4]. A team leader may grant privileged access to workflow parts or assign tasks to team members, as well as inspect the workflow using BPEL-specific software metrics (*workflow metrics*). Team members have different views on the BPEL model edited in a HOBBES session, to enable parallel development activities. For communication purposes and to enable a better understanding of the entire model, the need for graphical visualizations arises. These have to present the control flow paths in a concise way and preserve the hierarchies of the process' structure.

D. Eppstein and E.R. Gansner (Eds.): GD 2009, LNCS 5849, pp. 421–423, 2010.

HOBBES has been implemented using the Adobe Flex framework, which enables rich user interfaces as well as server-to-client notification. Communication between clients is relayed via a Java-based server, which governs access to a central object model. A demo version of HOBBES can be accessed at:
`http://www-sr.informatik.uni-tuebingen.de/workflows`



(a) Hobbes' user interface (left): The modeling frame (2) shows a single hierarchical level of a BPEL model, the tree of the whole current process is provided (3). Rectangle (1) depicts the palette of available BPEL elements. The result of our layout approach for this process is shown on the right: The hierarchies of the BPEL model are visualized unfolded. Start and Exit of a hierarchical element are depicted by green and red boxes, and surrounded by colored rectangles.



(b) Example of a layout for a complex BPEL-Process.

3 Realizing Layout Capabilities in Hobbes

A visualization of BPEL processes suggests a layered drawing technique. Thus, our layout approach is based on the Sugiyama algorithm. Since there is no unique method to derive paths from a BPEL model, we consider the number of descendant BPEL activites of structured activities for path construction and embedding. In order to draw each path in parallel, orthogonally to the layers in the Sugiyama model, three steps of the standard algorithm (Cycle Removal, Layer Assignment and Computation of the Horizontal Coordinates) have to be modified.

Each path consists of so-called *path-edges* and should preferably be drawn from top to bottom. Regarding cycles in the graph, we have to consider the special case of a cycle that consists only of path-edges. In this case, we perform a division of at least one of the paths contributing at least one edge to the cycle. Furthermore, additional constraints are added to the layer assignment phase in order to ensure that the source node of each path-edge is placed in a layer above the target node.

To increase the readability of the main activities in a workflow, paths should be drawn in a straight vertical manner. Thus, for the computation of the horizontal coordinates, the standard algorithm is applied as a first step followed by a postprocessing step: For each path p, the barycenter b_p of the x-coordinates of all nodes in p is computed. The horizontal coordinates of nodes contained in exactly one path p are set to b_p. For nodes contained in paths $p_1, \ldots, p_k$ with $k \geq 2$, the x-coordinate is set to the barycenter of $b_{p_1}, \ldots, b_{p_k}$.

Finally, in order to avoid overlapping nodes, non-path nodes are moved to the left and to the right until all nodes of the graph adhere to the minimal node distance. Final modifications are applied to the graphical representation of the elements of the graph: Each node is assigned its own shape according to its representative activity. All edges are drawn in a way that they respect the flow from the top to the bottom. To further improve the readability of the workflow, all hierarchical activities containing several subordinate activities are surrounded by colored rectangles.

References

[1] Alves, A., Arkin, A., Askary, S., Baretto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guizar, A., Kartha, N., Liu, C., Khalaf, R., König, D., Marin, M., Mehta, V., Thatte, S., van der Rijn, D., Yendluri, P., Yiu, A.: Web Services Business Process Execution Language Version 2.0. OASIS standard (April 2007)

[2] Sugiyama, K., Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical system structures. IEEE Transactions on Systems, Man, and Cybernetics SMC-11(2), 109–125 (1981)

[3] Held, M., Blochinger, W.: Collaborative BPEL Design in a Rich Internet Application. In: CCGRID 2008: 8th International Symposium on Cluster Computing and the Grid. IEEE Computer Society Press, Los Alamitos (2008)

[4] Held, M., Blochinger, W.: Structured collaborative workflow design. Future Generation Computer Systems FCGS 25(6), 638–653 (2009)