

Web-based Editor for YAWL

Felix Mannhardt

Bonn-Rhein-Sieg University of Applied Sciences, Germany*
felix.mannhardt@smail.wir.h-brs.de

Abstract. This paper presents a web-based editor that offers YAWL editing capabilities and comprehensive support for the XML format of YAWL. The open-source project Signavio Core Components is extended with a graphical user interface (GUI) for parts of the YAWL Language, and an import-/export component that converts between YAWL and the internal format of Signavio Core Components. This conversion, between the web-based editor and the official YAWL Editor, is lossless so both tools may be used together. Compared to the official YAWL Editor, the web-based editor is missing some features, but could still facilitate the usage of the YAWL system in use cases that are not supported by a desktop application.

1 Introduction

Software that is available only as a traditional desktop application is unlikely to receive substantial adoption nowadays, in a time when these applications are more and more replaced by web applications, and mobile apps. Major companies from the industry even work on operating systems solely based on a web browser (e.g. Google Inc. with Chrome OS and Mozilla Corp. with Firefox OS). The YAWL System provides a complete tool-chain for Workflow Management: the YAWL Editor for process design, the YAWL Control Center for configuration and management of organizational resource, and the YAWL Engine for process execution. Excluding the YAWL Editor all components of the YAWL system are web applications designed with a service-oriented architecture. The YAWL Editor, instead, is only available as a rather monolithic desktop application, which is built with Java using Swing and the JGraph library. Additionally, new features cannot be added to the YAWL Editor in a pluggable way, due to the software architecture. However, it is possible to create a new YAWL editor that mitigates these shortcomings and retains compatibility to the official YAWL Editor by using the well-defined YAWL XML format.

This new editor for YAWL is based on Signavio Core Components (SCC) and called web-based YAWL Editor hereafter. SCC is an open-source project started by Signavio GmbH that is based on the discontinued Oryx project [2] and the commercial, web-based process modeling tool Signavio Editor. The SCC editor provides basic process modeling functionality, a file-based repository and is

* Parts of the work done while visiting Queensland University of Technology, Australia

extensible with new process modeling languages through plug-ins. Unfortunately, also the SCC project was discontinued in 2011, shortly after the first release. Nevertheless, the source-code is still available¹, and efforts are made to continue the development as part of the Apromore² project [7]. The Apromore project also includes the YAWL editing capabilities that are presented in this paper. Most of the SCC editor is implemented with JavaScript that manipulates Scalable Vector Graphics (SVG) to show and edit process models in the browser window. For each supported process modeling language of SCC a so called Stencil Set [2] needs to be defined.

The new web-based YAWL Editor consists of SCC together with three added components: the YAWL Stencil Set, a client-side component and a server-side component. The YAWL Stencil Set includes SVG graphics of all YAWL elements (e.g. Tasks, Conditions, Edges), all their properties (e.g. description, routing conditions, task variables) and basic rules regarding their interconnection. Additional rules that cannot be expressed within the Stencil Set are added through the client-side component with JavaScript. The server-side component, which is realized with Java, is responsible to convert between the YAWL XML format and the JSON format of SCC. This conversion is lossless, therefore workflows created with the original YAWL Editor can be imported, modified with the web-based YAWL Editor and exported back to the original YAWL Editor.

In Section 2 the functionality of the new web-based YAWL Editor is presented, then limitations of the current implementations are discussed in Section 3, and before concluding this paper related work is described in Section 4.

2 Web-based YAWL Editor

Figure 1 shows the web-based YAWL Editor displaying the top level net of the *Orderfulfilment* process [5] that was imported using the XML file that is distributed as an example together with YAWL. On the left side of the browser window, there is the *Shape Repository* containing all available tasks (Atomic, Composite, Atomic Multiple Task, Composite Multiple Task) and conditions (Normal, Input, Output) of YAWL. A new instance of a shape can be added to the diagram area in the middle either by using drag and drop from the *Shape Repository* or by clicking on one of the context menu icons showing the follow-up element. The context menu is visible in Figure 1 next to the focussed *Ordering* task. The web-based YAWL Editor enforces the connection rules of YAWL, for example that tasks without attached split gateway must not have more than one outgoing edges or that an Input Condition must not have an incoming edge. As a result without the XOR-split on the *Ordering* task, there would no follow-up task or follow-up condition be displayed in the context menu.

On the right side of the browser window, all properties of the currently selected node can be edited. Please note that the original identifier of the YAWL element is preserved in the property *Internal YAWL ID*, and will be re-used upon

¹ <http://code.google.com/p/signavio-core-components/>

² Apromore is a process model repository, available at: <http://apromore.qut.edu.au/>

export of the workflow back to the YAWL XML format. The properties *Split* and *Join* determine whether a task has split or join routing behavior. These properties provide selectable options for all combinations of routing behavior (AND, XOR, OR) and placement of the visible counterpart on the task node (Top, Left, Right, Bottom).

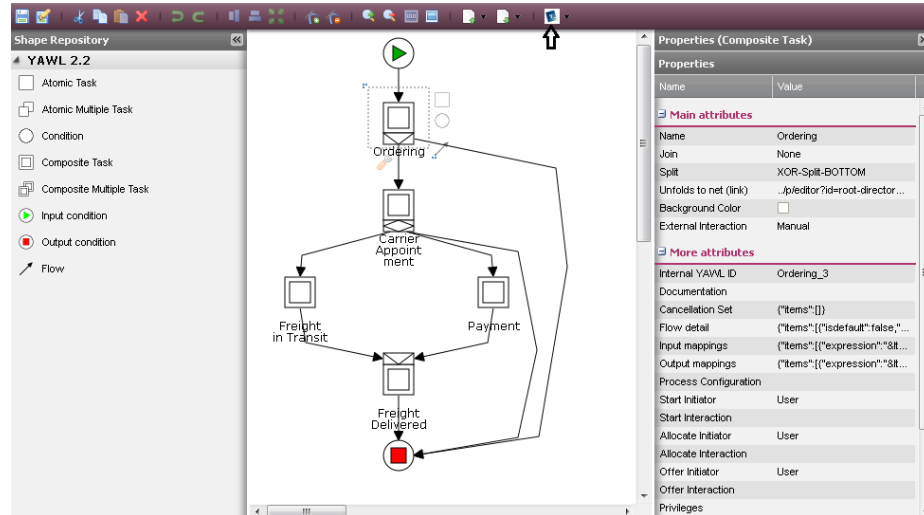


Fig. 1. Web-based YAWL Editor showing the Orderfulfilment example process

In contrast to the original YAWL Editor, which stores subnets of a workflow embedded in the workflow specification, upon import of the *Orderfulfilment* process all subnets have been stored separately in the repository. All composite tasks, like the task *Ordering* in Figure 1, just store a reference to the target subnet, as visible in the property *Unfolds to net (link)*. The reference is stored in form of an URI that points to the subnet in the SCC repository. When using another repository (e.g. Apromore), then this reference will be an URI that points to the process model in the external repository. This way, the same subnet can be used in multiple YAWL workflow specifications that are stored in the web-based YAWL Editor. A subnet that is created in the web-based YAWL Editor may also be exported as the rootnet of a YAWL workflow specification. This allows for more flexibility in the creation of workflows, for example the *Payment* subnet could be exported as a stand-alone workflow. Of course, during generation of the YAWL XML format all referenced subnets are embedded again in order to generate valid YAWL XML.

The features *Highlight Cancellation Region* and *Unfold to Subnet* have been implemented to show that the web-based YAWL Editor can be extended with more advanced features of the official YAWL Editor. *Highlight Cancellation Region* emphasizes the nodes and edges that are in the Cancellation Region of a

selected task. YAWL specific features, like *Highlight Cancellation Region*, can be activated under the YAWL symbol in the menu as indicated by the arrow shown in Figure 1. The second feature, *Unfold to Subnet*, supports the user with the handling of subnets. Upon selecting a Composite Task a context menu appears, as shown next to the task *pay* in Figure 2. The user can either open the already linked subnet, create a new subnet, link the task to an existing subnet that was created before, or unlink the currently assigned subnet. The selection of an existing subnet is supported by showing a list of all available YAWL workflows in the repository of the web-based YAWL Editor.

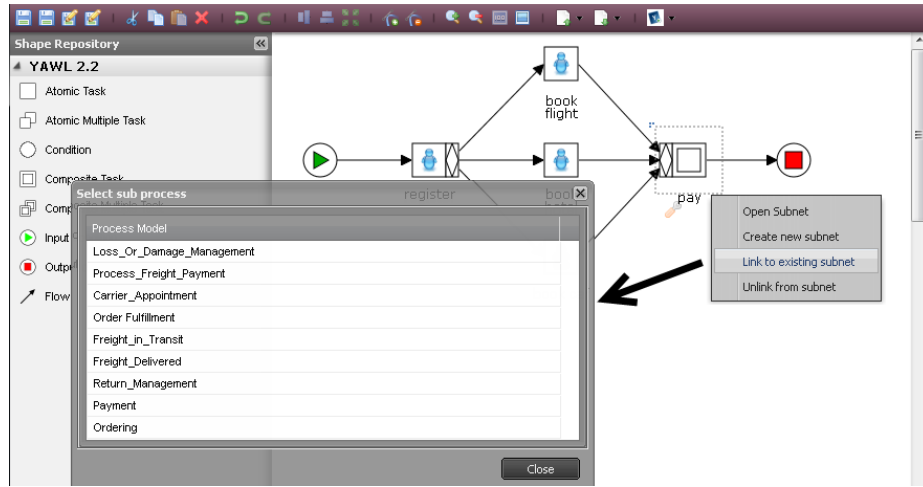


Fig. 2. Handling of YAWL subnets within SCC

The source code of the web-based YAWL Editor is currently available as part of the Apromore Editor: <http://code.google.com/p/apromore/>. Additionally, more information about future development and a demo version is available on the website: <http://www.yaug.org/WebYAWLEditor>.

3 Limitations

The current implementation of the web-based YAWL Editor provides fewer features than the official YAWL Editor. The most significant limitations are in the area of the resource and data perspective of YAWL. At this stage of the development, the web-based YAWL Editor does not connect to the Resource Service of YAWL to look up organizational resources (e.g. Roles, Participants, Secondary Resources), but the information about the resource perspective of an imported YAWL workflow is preserved using fragments of the original YAWL XML format. In a future version, the web-based YAWL Editor could connect to the Interface

R of YAWL to look up organizational resources and manipulate the stored XML fragment according to the selection. Regarding the data perspective, there is some support through structured editors, but the web-based YAWL Editor is missing *look up* and *validation* features. Figure 3 shows the structured editor for the routing predicates. New routing predicates can be added in a structured way, but the editor in Figure 3 still provides no means of automatically picking the target task, or checking the XSD expression for validity. The editor for task variables has similar limitations.

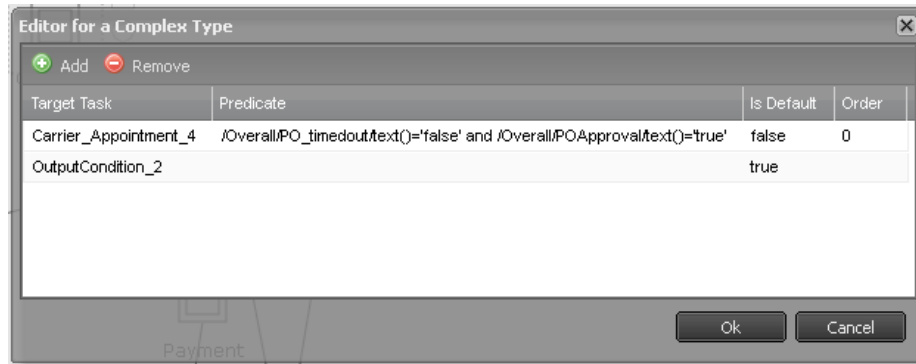


Fig. 3. Property editor for the routing predicates

Furthermore, there are still some minor limitations regarding the visual appearance of the YAWL workflows in the web-based Editor. SCC does not support bended edges with bezier curves or splines and it is missing support for exact positioning of labels. As a result, in the web-based YAWL Editor all labels are positioned directly under the corresponding YAWL elements and edges using bezier curves or splines are shown as straight lines.

4 Related Work

There have already been a few attempts bringing support for new process modeling languages to the Oryx Editor (predecessor of SCC) similar to this approach. For example, support for Coloured Petri Nets and UML class diagrams³. The web-based jBPM Designer⁴, which is based on the Oryx Editor, is developed as part of the jBPM project. As it is solely dedicated to BPMN 2.0 it was not considered as starting point for the development of the web-based YAWL Editor. Regarding to YAWL and SCC/Oryx the implementation of an automatic export of BPMN models to YAWL in the Oryx Editor⁵ using the approach described

³ <http://bpt.hpi.uni-potsdam.de/Oryx/OryxScreencasts>

⁴ <http://www.jboss.org/jbpm/components/designer>

⁵ <http://bpt.hpi.uni-potsdam.de/Oryx/BpmnToYawl>

in [1] could be seen as related. The approach is limited to a subset of BPMN 1.2 and can not be used to edit existing YAWL workflows on the web. There is a YAWL Android Client⁶ that supports the execution of a YAWL workflow on mobile devices, but this project does not provide support for the design time of a workflow. To the best of my knowledge there is currently no other web-based YAWL Editor available.

5 Conclusion

With the web-based YAWL Editor, it is possible to create new, and modify existing YAWL workflow specifications using solely a web browser. The web-based YAWL Editor is fully compatible with the official YAWL Editor. It has been tested with the publicly available workflow specifications, which are distributed together with YAWL. The web-based YAWL Editor can be offered as a service without the need of a local installation, and it can be integrated in existing web applications. In comparison to the original YAWL Editor some use cases are simplified, like the usage of YAWL in a classroom during an exercise, and new use cases are made possible, like the interactive presentation of YAWL workflow specifications on a web site.

The web-based YAWL Editor is, furthermore, already integrated into the process model repository Apromore. In the future, using Apromore and the web-based YAWL Editor features like, for example, version control [4] and process merging [6] may be available to the YAWL community. Still, there are some shortcomings with the current implementation of the web-based YAWL Editor that need to be mitigated. Most of all, better support for the resource and data perspective of YAWL is needed. Overall with some additional development effort, the web-based YAWL Editor could evolve to a beneficial addition to the YAWL ecosystem.

References

1. Gero Decker, Remco M. Dijkman, Marlon Dumas, and Luciano García-Bañuelos. Transforming bpmn diagrams into yawl nets. In Dumas et al. [3], pages 386–389.
2. Gero Decker, Hagen Overdick, and Mathias Weske. Oryx - an open modeling platform for the bpm community. In Dumas et al. [3], pages 382–385.
3. Marlon Dumas, Manfred Reichert, and Ming-Chien Shan, editors. *Business Process Management, 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings*, volume 5240 of *Lecture Notes in Computer Science*. Springer, 2008.
4. Chathura C. Ekanayake, Marcello La Rosa, Arthur H. M. ter Hofstede, and Marie-Christine Fauvet. Fragment-based version management for repositories of business process models. In Robert Meersman, Tharam S. Dillon, Pilar Herrero, Akhil Kumar, Manfred Reichert, Li Qing, Beng Chin Ooi, Ernesto Damiani, Douglas C.

⁶ <http://www.yaug.org/tryOutYAWLAndroidClient>

- Schmidt, Jules White, Manfred Hauswirth, Pascal Hitzler, and Mukesh K. Mohania, editors, *OTM Conferences (1)*, volume 7044 of *Lecture Notes in Computer Science*, pages 20–37. Springer, 2011.
5. Marcello La Rosa, Stephan Clemens, Arthur Hofstede, and Nick Russell. Appendix A The Order Fulfillment Process Model. In Arthur H M ter Hofstede, Wil M P van der Aalst, Michael Adams, and Nick Russell, editors, *Modern Business Process Automation*, pages 599–616. Springer Berlin Heidelberg, 2010.
 6. Marcello La Rosa, Marlon Dumas, Reina Uba, and Remco M. Dijkman. Merging business process models. In Robert Meersman, Tharam S. Dillon, and Pilar Herrero, editors, *OTM Conferences (1)*, volume 6426 of *Lecture Notes in Computer Science*, pages 96–113. Springer, 2010.
 7. Marcello La Rosa, Hajo A. Reijers, Wil M.P. van der Aalst, Remco M. Dijkman, Jan Mendling, Marlon Dumas, and Luciano Garca-Baueles. Apromore: An advanced process model repository. *Expert Systems with Applications*, 38(6):7029 – 7040, 2011.