

SonarEditor: A Tool for Multi-Agent-Organizations Modelling

Jan Bolte

Department of Informatics, TGI, University of Hamburg
8bolte@informatik.uni-hamburg.de

Abstract. This paper presents the SonarEditor, which supports the creation of SONAR models. It provides pre-build net components, a well-formedness check and a wizard that aims at the support for possible enhancements of the model.

SonarEditor [1] is a prototypical implementation that enables multi-agent system developers to model organizational models following the SONAR formalism. It is implemented as plugin for RENEW [3]. The SonarEditor consists of three parts: pre-build net components [2], which help to create models, a well-formedness check that can check the well-formedness of the model and a wizard that can be used to enhance the model. Well-formedness of a model is defined by the definition of an organization and that the model is acyclic [4, Section 3.1].

SONAR is an approach to model organizations based on Petri nets. Such a model is composed of a delegation net and a set of distributed workflow nets (DWFs). The DWFs model the real workflows and the delegation net all possible courses of actions in an organization. The SonarEditor focuses on the delegation net, which is a Petri net (P, T, F) with P a set of Tasks, T a set of Implementations and F a set of arcs between P and T . Every Task and Implementation has to be assigned to a Position. This Position models a position in the organization. A Task models a task, which has to be executed. The Implementations define how every Position can implement each task. There are four different types of Implementations allowed by formal definition of the delegation net. The four types are named *execute*, *delegate*, *split*, *refine* and the pre-build net components for them are presented in the bottom of Figure 1. The top of Figure 1 displays the toolbar of the SONAR net components. There are from left to right *Position*, *Initial Task*, *Task*, *execute*, *delegate*, *split*, *refine*, *refine+split*. These are the net components for the delegation net. *refine+split* is the combination of the Implementations *refine* and *split*. The next components *Declaration Node*, *Role* and *DWFAction* are the components for creating DWFs. The last three buttons trigger the well-formedness check and the wizard: the first two trigger the well-formedness check with and without DWFs, the last triggers the wizard.

The SonarEditor shows an error frame if errors occur while checking the well-formedness. This frame contains a list of these errors and three buttons: *Select*, *Select All* and *Cancel*. If an error item is selected in the error frame the button *Select* can be pushed to select the corresponding element in the delegation net. With the button *Select All* all elements causing errors will be selected.

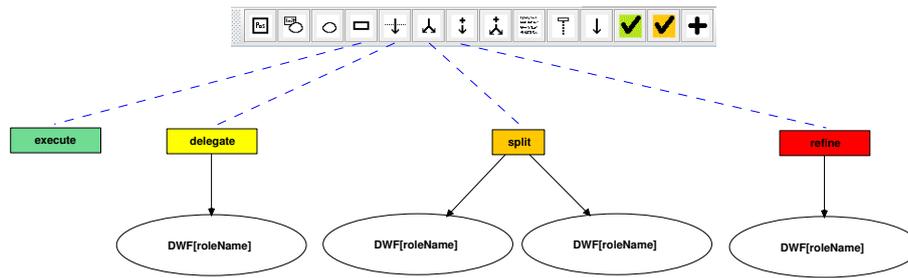


Fig. 1. Toolbar and SONAR net components

The last button of the toolbar starts the wizard, which supports the user by proposing possible enhancements based on the existing DWFs and delegation net. The wizard pages hold their previous and their following page and manage the remaining GUI of the wizard. There are five different wizard pages in this implementation:

ErrorPage is the first page, if a Task is selected, which causes an error.

InitOrPosPage is the first page, if nothing is selected and you can create a Position or start the creation of an initial Task with this page.

TypePage is the first page, if the selected Task does not cause an error. On this page you can define the type of the new Implementation. In the cases of *execute* and *split* this page is the last page.

PosPage is the page, where one can select the Position of the initial Task or the output Task of a new *delegate* Implementation.

DWFPage is the page, where you can select the DWF of the initial Task or the refining DWF in a *refine* or *refine+split* Implementation.

The SonarEditor supports the user with net components, a well-formedness check and a wizard. By these means, it supports the creation of well-formed SONAR models. A possible extension to this tool would be the consideration of the DWF well-formedness in the well-formedness check.

References

1. Jan Bolte. Werkzeug-Unterstützung für organisationsorientierte Modellierung in SONAR, 2012. Bachelor thesis.
2. Lawrence Cabac. Net components: Concepts, tool, praxis. In Daniel Moldt, editor, *Petri Nets and Software Engineering, International Workshop, PNSE'09. Proceedings*, Technical Reports Université Paris 13, pages 17–33, 99, avenue Jean-Baptiste Clément, 93 430 Villetaneuse, June 2009. Université Paris 13.
3. Olaf Kummer, Frank Wienberg, Michael Duvigneau, and Lawrence Cabac. Renew – the Reference Net Workshop. <http://www.renew.de/>, 2012. Release 2.3.
4. Michael Köhler-Bußmeier and Matthias Wester-Ebbinghaus. Analysing Model Transformations in SONAR. 2012.