

Renew 2.5 – Towards a Comprehensive Integrated Development Environment for Petri Net-Based Applications

Lawrence Cabac^(✉), Michael Haustermann^(✉), and David Mosteller^(✉)

Department of Informatics, University of Hamburg, Hamburg, Germany
{cabac,haustermann,mosteller}@informatik.uni-hamburg.de

Abstract. RENEW (The Reference Net Workshop) is an extensible Petri net IDE that supports the development and execution of high-level Petri nets and other modeling techniques. The Reference net formalism – the major formalism for RENEW – includes concepts such as net instances, synchronous channels and seamless Java integration. It combines the advantages of Petri nets and object-oriented programming for the development of concurrent and distributed software systems. Modeling support of RENEW focuses on convenience and ease for Petri net development. An outstanding feature is the support for multi-formalism simulation. The plugin architecture of RENEW enables the developers to extend the IDE for instance with additional formalisms. Alternatively to the inline mode – within the graphical user interface – the Simulator can also be run in a headless server fashion. Several configurations of RENEW are available, which are constituted through selections of plugins providing specialized functionality for multiple platforms. In this manner the RENEW family constitutes a product line architecture. RENEW is available free of charge including the Java source code. In this contribution we provide information about RENEW’s functionality and architecture as well as the development of the tool set over the last decade.

Keywords: High-level Petri nets · Nets-within-nets · Reference nets · Integrated Development Environment (IDE) · Java · Plugin architecture

1 Introduction

RENEW is a continuously developed extensible modeling and execution environment for Petri nets with various formalisms and other modeling techniques. The main formalism of RENEW is the Reference net formalism [10], which combines the concept of nets-within-nets [15] with a reference semantics and the expressive power of object-oriented programming in the form of Java. Reference nets in RENEW can handle Java objects as tokens and Java expressions in transition inscriptions to execute Java code during the simulation. With the nets-within-nets concept, it is possible to build dynamic hierarchies of arbitrary height. Multiple nets can communicate using synchronous channels, which enable the

bidirectional exchange of information. The formalism is, thus, well-suited for the implementation of concurrent software systems. RENEW is written in Java and is available for multiple platforms (including Windows, Linux and Mac). An experimental version of the simulator for Android exists. The current version 2.5 is available for download¹ free of charge including the source code [11].

This contribution presents the RENEW environment with its objectives and history (Sect. 2), a selection of features and improvements over the recent years of continuous development (Sect. 3) and a brief overview of the plugin architecture (Sect. 4). The paper is summarized in Sect. 5.

2 Objectives

Up to today the objectives of the RENEW tool and its various plugins have been widely extended. The following section gives a brief summary of the evolution of RENEW over the past recent years. This section is followed by some information about current research topics in the context of RENEW development.

2.1 History

The first official version of RENEW was released in 1999. Since then it has continuously been developed as a Petri net IDE by the TGI Group². Figure 1 shows the user interface of the first release of Renew, which offers the drawing of Petri net models and the starting and stopping of the simulation. Although other formalisms than Reference nets are available, the user interface offers no means of control to switch compilers.

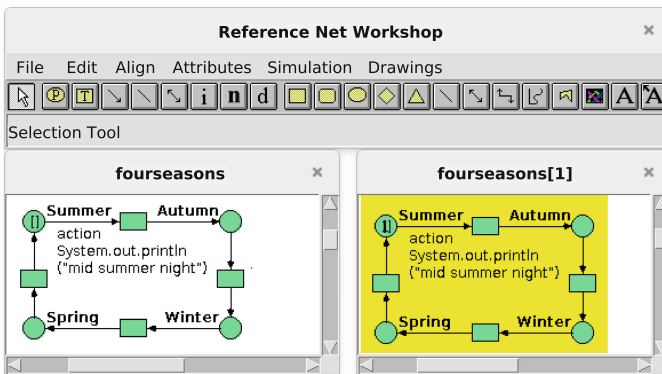


Fig. 1. RENEW 1.0 GUI, Debian 8/Java 7 together with a four seasons example (pattern and instance) featuring a simple Java inscription.

¹ RENEW 2.5 available at: <http://www.renew.de>.

² Theoretical Foundations Group, Department of Informatics, University of Hamburg.

The continuous development resulted in many improvements and bugfixes as well as feature enhancements. One major task has been the decoupling of editor and simulator, which started with the introduction of the separation layer in RENEW 1.6. The plugin system, introduced with the major release 2.0 in 2004 [12], enabled the extension of RENEW into various directions. RENEW was extensively applied and extended to perform agent-oriented software engineering (see following sections) and was furthermore utilized to provide a workflow management engine and clients. Besides using RENEW primarily for modeling Petri nets, plugins provide support for various modeling techniques, e.g. diagrams from UML or BPMN. Over the years RENEW evolved increasingly more into an IDE for software engineering with high-level Petri nets.

2.2 Implementing Petri Net-Based Applications

One main focus of RENEW is the execution/simulation of Reference nets. With the full support of that formalism, RENEW serves as an execution engine for net-based Java applications. Figure 2 shows a simple example net that uses Java objects, which could be part of a graphical application. In the net a frame and a button are created concurrently. After that the button is added to the frame and the frame is resized (again concurrently). When these two steps are finished the frame is displayed and in a next step disposed. This simple example gives a first impression on how to implement Petri net-based applications. With the possibility to structure multiple nets using the nets-within-nets concept, it is possible to implement even complex software systems (e.g. with the PAOSE approach, see Sect. 2.3).

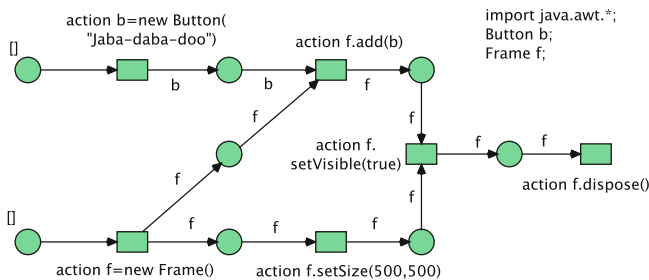


Fig. 2. Example net using Java objects.

2.3 Petri Net-Based, Agent-Oriented Software Engineering

The Petri Net-Based agent-oriented software engineering (PAOSE, [2]) is an approach to software development with an emphasis on distribution and concurrency. With the MULAN framework (multi-agent nets, [14]) Reference nets are

applied as implementation artifacts for the development of multi-agent applications (MAA). To this end RENEW serves as an IDE for the development of MAA by providing the editing, debugging and simulation facilities. With its support for abstract modeling languages and the use of generative techniques to translate them into Reference nets RENEW supports the PAOSE development process in every single step from requirements engineering to system execution. RENEW is applied in the context of PAOSE as a tool for teaching and research in the field of agent-oriented software engineering.

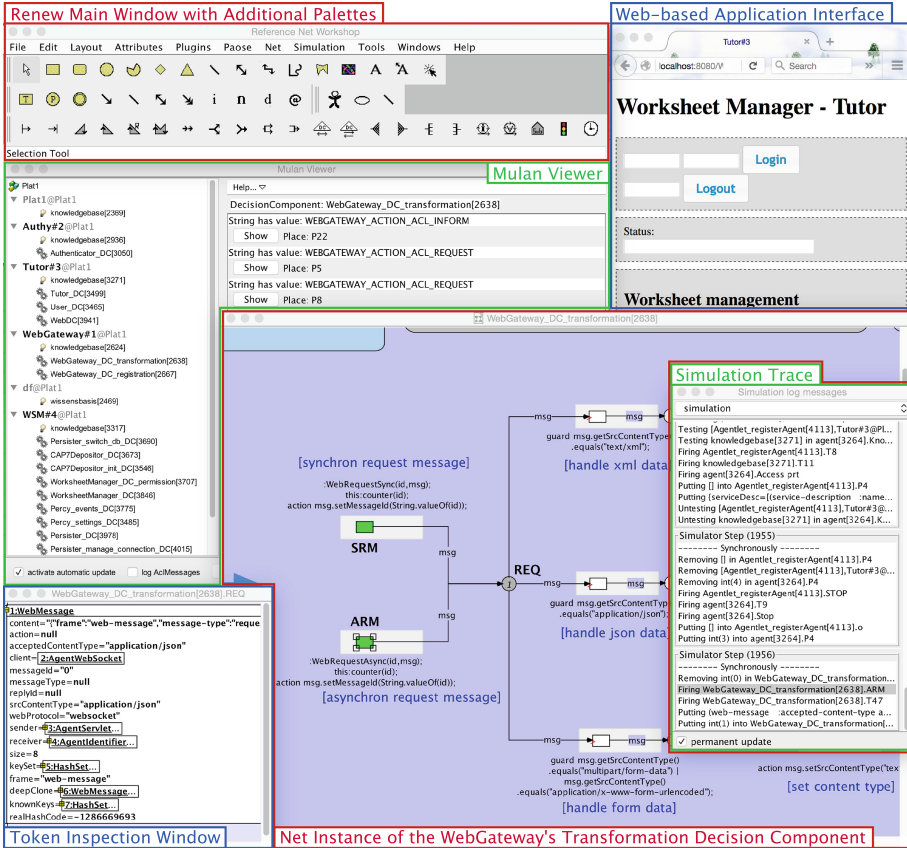


Fig. 3. Renew 2.5 execution of a multi-agent system.

Figure 3 shows the current version of the RENEW GUI together with several views on the system provided by various tools. The whole structure of the simulated PAOSE system can be inspected and navigated with the Mulan Viewer [3], displayed beneath the main GUI. In the structure tree the developers may inspect the agents with their sub-components. In the image the transformation decision

component (DC) of the WebGateway agent is selected and details about this artifact can be inspected in the detail view (to the right of the structure tree, partly hidden). In the center/lower part a fraction of the artifact itself – the transformation DC – is displayed featuring one selected transition (*ARM* – asynchronous request message) and a marked place (*REQ*). The simulation has been halted for inspection after the transition *ARM* has fired through a preset local simulation breakpoint. This transition was involved in the firing step 1956 as can be inspected in the simulation log window depicted in the right. The selection of the transition in the Petri net has been done from the simulation log by using the context menu, which has been activated again for the presentation. The single token depicted as 1 on the place *REQ* can be inspected either by changing the representation directly in the net instance to a token bag view showing the string representation of the object or through a UML-like deep inspection of the object shown in the window located in the lower left corner of the screenshot. The token is a *WebMessage*, which contains the login data of a user that has clicked on the *login* button in the Web interface shown in the upper right corner of the screenshot. The deep inspection window is provided by the Feature Structure plugin.

Several implementations based on MULAN are available on the PAOSE Web page³. These include some demos, a distributed implementation of a multi-player game, and an export and diffing Web service for RENEW-compatible diagrams. The latter is presented through a stand-alone Web interface as well as integrated into the Redmine project management environment.

2.4 PAOSE Meta-Modeling

During its evolution over the years RENEW’s capabilities for providing modeling support were extended beyond the initial focus on Petri nets. A great number of the additional modeling techniques are applied within the PAOSE approach. As mentioned in the previous section RENEW supports each step of the PAOSE development process from requirements engineering, over system design and specification to implementation and execution with appropriate modeling techniques [2]: we use Concept Diagrams to model agent ontologies, Use-Case Diagrams to capture the overall structure and a variant of Sequence Diagrams to model agent behavior in conversations. The latter are used to generate Reference nets, which can be executed as protocols in agent interactions. With the plugin from Haustermann [8], BPMN models can also be applied for this purpose. In our current research we are working on the RENEW Modeling and Transformation framework (RMT [13]), a framework for model-driven development of domain specific modeling languages and tools based on RENEW.

3 Highlights and Improvements

RENEW offers a wide range of features for the creation, the editing and manipulation of models as well as for execution, debugging and deployment of the

³ PAOSE Web page: <http://www.paose.net>.

designed systems. We present a selection that – to our opinion – describes best the nature of the tool set. Although many of the presented features can not be put into one category we present the highlights as functional feature, as usability feature or as IDE related feature.

3.1 Functional Features

Multiple Formalisms. The simulator is capable of handling different formalisms. The main formalism is the Java Reference net formalism, for which different extensions exist, such as inhibitor, reset and timed arcs. The workflow net formalism, provided by an optional plugin, adds a task transition, which can be canceled during execution, so that its effect on the net can be reverted. Other formalisms provide simulation of P/T nets, feature structure nets and bool nets. Simulation is available in different modes. In the interactive simulation mode the user may control the simulation by choosing the transitions to fire and inspect each single step. The automatic simulation mode is usable for system execution and can be run with and without graphical feedback (operation as server).

Net Loading and Class Loading. RENEW features dynamic loading of nets on demand from the netpath or from the GUI. Nets may be available in the RENEW editor’s format (rnw), which permits direct inspection, or as pre-compiled net system without graphical information (shadow net system), which allows a silent execution. The configurable *class reinit mode* allows to quickly develop models that combine nets and Java classes. Classes that are recompiled are reloaded at simulation start, allowing nets and Java classes to be quickly developed, tested and debugged without any restart of the environment.

Logging and Remote Monitoring. Logging in RENEW is provided through log4j. It is configurable and allows to log tool behavior and simulation events. Several plugins, which provide general or special purpose monitoring facilities may also be used for the inspection of local or remote simulations.

Graphical Editor. RENEW provides an easy to use graphical editor for Petri net models and other types of models and a simulation engine, which is seamlessly integrated into this editor. It has a plugin architecture, which makes it easily extensible. The core plugins are provided as part of the RENEW distribution. Many advanced features are supplied as optional plugins.

FreeHep Graphical Export. Since version 2.2 the graphical export relies on the FreeHep⁴ libraries, which provide a wider range of formats including SVG and a high quality of output.

⁴ The FreeHep libraries are available at <https://github.com/freehep/>.

3.2 Usability Features

Drag & Drop. Drag and drop support for RENEW and the navigator eases the usability by providing means to opening files or project folders directly within the editor’s GUI. Additionally, images can now be added to a drawing by dropping them on the canvas.

Interaction Enhancements. The quick-draw mode has been improved to even faster draw Petri net models. Also configurable shortcuts for drawing tools and layout manipulation are available in order to combine tool selection by keyboard and positioning of net elements by the mouse.

Several drawing elements react to modifier keys during direct manipulation, so that ratios of height and width can be preserved or unified. This is for example useful when re-sizing an image.

Target Text Tool. The target text tool allows to add hyperlinks to graphical elements of a model. By this means models may be inter-linked, in order to offer simplified navigation between related models. Additionally, the usual schemes are supported through the desktop integration as well, allowing to link to Web pages (e.g. documentation/wikis) or other source code.

3.3 IDE Features

The editor has been improved over the last years and received many small usability enhancements and has evolved into an integrated development environment (IDE) for net-based software development. It contains a syntax check during editing and debugging tools, such as breakpoints or manual transitions. Furthermore the editor features desktop integration, a file navigator and image export to various formats.

Net/Diagram Diff. The net diff feature – for the first time presented in 2008 – has been added as ImageNetDiff [5,6] plugin to the optional plugins in version 2.3 and has been further enhanced for effectiveness and efficiency. It provides the functionality of diffing Petri net models and other diagrams directly within RENEW or as a Web service⁵, which is used and integrated to offer this for the integrated project-management environment Redmine [1].

Quick-Fix. Syntax error notification dialogues are now enhanced to suggest possible solutions to the syntax errors, such as available methods, fields or constructors. The user can interactively choose from the provided solutions and apply the changes to the model in a quick-fix manner. Also undeclared variables are semi-automatically added to the declaration node. In Fig. 4 the variable *net*

⁵ Available at <http://paose.informatik.uni-hamburg.de/export/>.

(compare with the declaration node) is not yet declared, which results in a syntax exception. The exception presents several solutions, of which the first one is currently selected. By clicking on the *apply* button (or double-clicking on the proposal) the variable will be added to the declaration node.

Refactoring. The renaming of variables in a Petri net drawing and changing of synchronous channels in net systems is now supported through the Refactoring plugin. However, these features are still under development and are still experimental.

Project Navigator. The Navigator – first introduced in version 2.3 – has been further improved. Due to a complete redesign, it provides a quicker update strategy, persistence and a filter functionality. The Navigator GUI is now extendable by plugins. On the right hand side of Fig. 4 the new Navigator GUI is displayed. The top level of buttons provides control over the content of the Navigator’s tree view. The second row provides a filter for the elements in the view. Extensions may add functionality as additional buttons or as context menus. The SVN/Diff Navigator extension provides the diff of two selected Petri nets (or other models) via button and diffs against the document base via context menu. Other SVN related functionality is provided as well.

Console Plugin. The new Console plugin, which utilizes JLine⁶ as library, provides many convenience enhancements for the RENEW command line in compar-

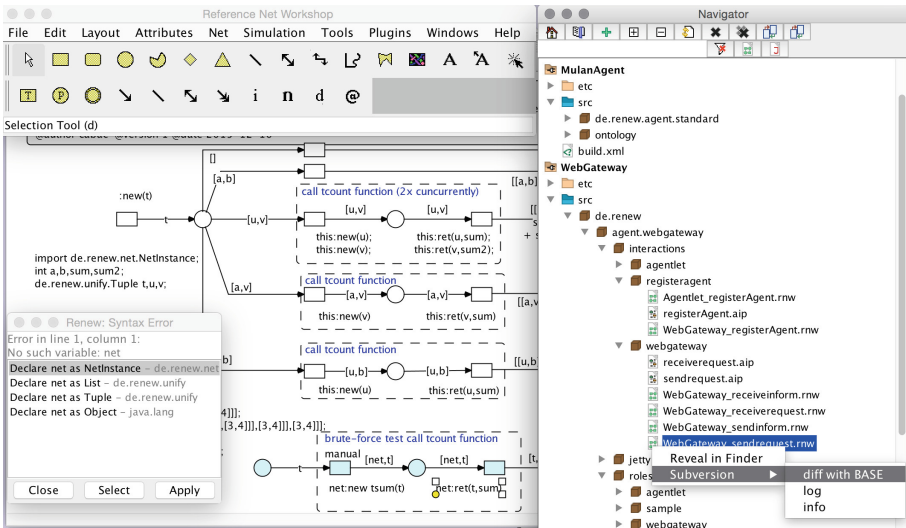


Fig. 4. RENEW Editor GUI featuring the navigator and the quick-fix functionality.

⁶ JLine: <https://github.com/jline/jline2/>.

ison to the replaced Prompt plugin. The main advantages are editable command lines, tab-based command and argument completions as well as searchable and persistent command history.

Analysis. With the integration of the LoLA⁷ verification tool [9] RENEW is also suited for verification tasks during modeling. So far only verification of P/T net models is provided through LoLA.

Net Components. Net components can be provided – and shared among developing teams – in order to ease recurrent modeling tasks and provide a conventionalized modeling style, which improves the readability and maintainability of net models.

Availability and OS-Integration. RENEW is available for most common platforms including Linux (and other Unices), Windows and Mac OS X. For all Platforms there exists a GUI integration into the respective Desktop environment. For Linux/X11 we provide an integration for the FreeDesktop standard and experimental debian/ubuntu packages. For Mac OS X we provide a specialized plugin for the desktop integration and an application bundle as well as a specialized installation package (DMG, disk image file).

4 Architecture

The plugin architecture – introduced in release 2.0 of RENEW – has already been presented [4, 7, 12]. Figure 5 shows the main parts, such as *simulator*, *graphical framework* (JHotDraw⁸), the formalism management and the Petri net IDE core. The latter extends the graphical framework as well as the simulator. On top of these core elements other plugins can be included, such as editor plugins, formalism plugins, tools that extend these formalisms and also applications, which may use Java code and Petri net code in combination.

Over the last years many plugins in various stages of maturity have been created by the maintainers or by other developers (e.g. students). Some of these extend the RENEW environment, while others extend the MULAN environment or provide multi-agent applications.

A boost for the development of plugins have been the plugins that allow the creation of plugin source folders. There exists a rather simple version of a *Plugin Development Plugin* for Renew plugins and a more elaborated version that allows the creation of multi-agent application folders – the *Use Case Components Plugin*⁹. With these plugins it is possible to create a new RENEW plugin simply by providing the name and the location of the source folder. The latter must be a valid source code repository in order to be able to compile the

⁷ LoLA – A Low Level Petri Net Analyzer: <http://www.service-technology.org>.

⁸ JHotDraw: <http://www.jhotdraw.org>.

⁹ The palette of the *Use Case Components Plugin* is activated in Fig. 3.

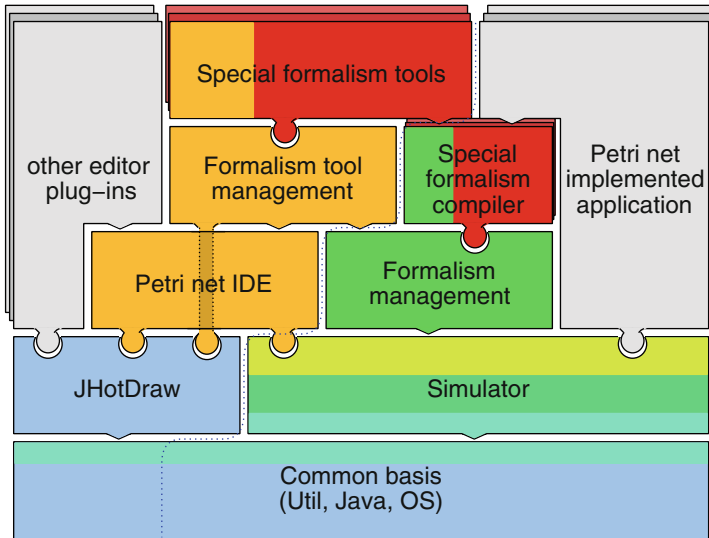


Fig. 5. RENEW's abstract architecture [7] – components as plugins.

plugin. A newly created plugin folder features already all necessary artifacts for the compilation of the plugin. These are the *Ant* build script, the plugin configuration file (*plugin.cfg*) as well as the source code folder. Additionally, a valid plugin facade class is generated. The Use Case Components Plugin also generates artifacts for the creation of MULAN applications according to a provided Coarse Design Diagram (CDD, a form of Use Case Diagram). These artifacts comprise the skeletons for agent role declarations, interaction definitions and ontology specification.

Java classes, RENEW nets and other artifacts – for instance *JavaCC* grammars or *JavaScript* files – can be added to the source code folder. The plugins may be accompanied by third party libraries and test cases.

An application that has been used heavily in the last couple of years is the above mentioned export and diff Web service. The *Mulan Export* plugin provides an agent (the *ExportAgent*) that offers the export and diff services to other agents and through the WebGateway also as a Web service to other clients. The backend of the export and diff functionality is provided by the respective RENEW plugins (FreeHep Export and ImageNetDiff). Over the last couple of years the service running as a MULAN system in an instance of RENEW has been constantly available and used, with up-times of several months. These long lasting running instances of Reference net systems and the high availability of the services show the feasibility and efficiency of the approach. Now several instances¹⁰ of the export and diff service are running on several servers.

¹⁰ A publicly available demo can be reached from the PAOSE home page: <http://www.paose.net>.

5 Conclusion

RENEW's development has been ongoing for almost two decades. It has developed from a simulator and editor for a single high-level Petri net formalism to an integrated development environment for Java- and Petri net-based applications. The Reference net formalism and RENEW's simulator have proven themselves in various implementations to provide powerful and efficient means to develop and execute systems that include a strong focus on concurrency. Within the PAOSE context distributed, concurrency-aware systems have been implemented with the MULAN framework extension. In this environment many objectives, such as support for monitoring or specialized modeling techniques, have been provided on the grounds of the RENEW framework and were only possible through RENEW's plugin architecture.

Today RENEW exists in various configurations. Since the possibilities are growing constantly we are aiming at a product line architecture in order to ease configuration-intensive work and endorse reliable, feature-complete tool sets for multiple application domains.

Acknowledgment. We thank all developers that participated in the development of RENEW, especially Prof. Dr. Rüdiger Valk, Dr. Daniel Moldt and all of the TGI group of the Department of Informatics, University of Hamburg.

References

1. Betz, T., Cabac, L., Güttler, M.: Improving the development tool chain in the context of Petri net-based software development. In: Duvigneau, M., Moldt, D., Hiraishi, K. (eds.) Petri Nets and Software Engineering. International Workshop PNSE 2011, Newcastle upon Tyne, UK, June 2011. CEUR Workshop Proceedings, vol. 723, pp. 167–178. CEUR-WS.org, June 2011. <http://CEUR-WS.org/Vol-723>
2. Cabac, L.: Modeling Petri Net-Based Multi-Agent Applications, Agent Technology - Theory and Applications, vol. 5. Logos Verlag, Berlin (2010). <http://www.logos-verlag.de/cgi-bin/engbuchmid?isbn=2673&lng=eng&id=>
3. Cabac, L., Döriges, T., Rölke, H.: A monitoring toolset for PAOSE. In: van Hee, K.M., Valk, R. (eds.) PETRI NETS 2008. LNCS, vol. 5062, pp. 399–408. Springer, Heidelberg (2008)
4. Cabac, L., Duvigneau, M., Moldt, D., Rölke, H.: Modeling dynamic architectures using nets-within-nets. In: Ciardo, G., Darondeau, P. (eds.) ICATPN 2005. LNCS, vol. 3536, pp. 148–167. Springer, Heidelberg (2005)
5. Cabac, L., Markwardt, K., Schlüter, J.: ImageNetDiff: finding differences in models. In: Moldt, D., Ultes-Nitsche, U., Augusto, J.C. (eds.) In: Proceedings of the 7th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems - MSVVEIS 2009. Conjunction with ICEIS 2009. Milan, Italy, May 2009, pp. 156–161. INSTICC PRESS, Portugal (2009)
6. Cabac, L., Schlüter, J.: ImageNetDiff: a visual aid to support the discovery of differences in Petri nets. In: 15. Workshop Algorithmen und Werkzeuge für Petrinetze, AWPN 2008. CEUR Workshop Proceedings, vol. 380, pp. 93–98. Universität Rostock, September 2008. <http://CEUR-WS.org/Vol-380/paper15.pdf>

7. Duvigneau, M.: Konzeptionelle Modellierung von Plugin-Systemen mit Petrinetzen, *Agent Technology - Theory and Applications*, vol. 4. Logos Verlag, Berlin (2010)
8. Haustermann, M.: BPMN-Modelle für petrinetzbasierte agentenorientierte Softwaresysteme auf Basis von Mulan/Capa. Master thesis, Department of Informatics, University of Hamburg, Vogt-Kölln Str. 30, D-22527 Hamburg, September 2014
9. Hewelt, M., Wagner, T., Cabac, L.: Integrating verification into the PAOSE approach. In: Duvigneau, M., Moldt, D., Hiraishi, K. (eds.) *Petri Nets and Software Engineering. International Workshop PNSE 2011*, Newcastle upon Tyne, UK, June 2011. CEUR Workshop Proceedings, vol. 723, pp. 124–135. CEUR-WS.org, June 2011. <http://CEUR-WS.org/Vol-723>
10. Kummer, O.: Referenznetze. p. 456. Logos Verlag, Berlin (2002). <http://www.logos-verlag.de/cgi-bin/engbuchmid?isbn=0035&lng=eng&id=>
11. Kummer, O., Wienberg, F., Duvigneau, M., Cabac, L.: Renew - user guide (Release 2.4.2). In: Faculty of Informatics, Theoretical Foundations Group, University of Hamburg, Hamburg, January 2015. <http://www.renew.de/>
12. Kummer, O., Wienberg, F., Duvigneau, M., Schumacher, J., Köhler, M., Moldt, D., Rölke, H., Valk, R.: An extensible editor and simulation engine for Petri nets: RENEW. In: Cortadella, J., Reisig, W. (eds.) *ICATPN 2004*. LNCS, vol. 3099, pp. 484–493. Springer, Heidelberg (2004)
13. Mosteller, D., Cabac, L., Haustermann, M.: Providing petri net-based semantics in model driven-development for the renew meta-modeling framework. In: Moldt, D., Rölke, H., Störrle, H. (eds.) *Petri Nets and Software Engineering. International Workshop, PNSE 2015*, Brussels, Belgium, June 22–23, 2015. CEUR Workshop Proceedings, vol. 1372, pp. 99–114. CEUR-WS.org (2015). <http://CEUR-WS.org/Vol-1372>
14. Rölke, H.: Modellierung von Agenten und Multiagentensystemen - Grundlagen und Anwendungen, *Agent Technology - Theory and Applications*, vol. 2. Logos Verlag, Berlin (2004). <http://logos-verlag.de/cgi-bin/engbuchmid?isbn=0768&lng=eng&id=>
15. Valk, R.: Petri nets as token objects: an introduction to elementary object nets. In: Desel, J., Silva, M. (eds.) *ICATPN 1998*. LNCS, vol. 1420, pp. 1–25. Springer, Heidelberg (1998)