WoPeD –
A tool for teaching, analyzing and visualizing workflow nets

Andreas Eckleder[1], Thomas Freytag[2]

[1] Nero AG, Karlsbad, Germany
eckleder@woped.org
[2] University of Cooperative Education (Berufsakademie), Karlsruhe, Germany
freytag@woped.org

Abstract. *WoPeD (Workflow Petrinet Designer) is a Java-based, open-source software tool developed at the University of Cooperative Education ("Berufsakademie") Karlsruhe, Germany. WoPeD uses the Petri Net Markup Language (PNML) as interchange format and is able to edit, simulate and analyze plain P/T nets and in particular "van-der-Aalst"-like workflow nets. The main focus of WoPeD lies on simplicity, interactiveness and visual expressiveness, giving teachers and researchers a powerful and flexible e-learning and e-publishing instrument.*

Introduction

WoPeD is an easy-to-use software tool for editing, simulating and analyzing workflow nets as well as plain place-transition Petri nets. WoPeD is being developed continuously by faculty members and students at the University of Cooperative Education ("Berufsakademie") Karlsruhe/Germany in conjunction with a group of graduates who now are IT professionals in local partner companies.

WoPeD is published under the LGPL, an open-source license which allows free access to its source code and underlying design documents. The Java-based tool runs on all operating systems for which Java version 5.0 or higher is available - in particular on all Windows platforms, most Linux distributions and MacOS X.

WoPeD primarily focuses on educational and publishing purposes, supporting lecturers, researchers and students in the area of Petri nets and workflow management. WoPeD is relating strictly to the well-established "van-der-Aalst" notation [Aal02] and can visualize both structure and dynamics of workflow nets, providing a deeper understanding of the underlying principles, notational concepts, key properties and algorithms.

By this, WoPeD can be considered as an e-learning or e-publishing instrument. It is a supporting tool for creating lecture materials or setting up and performing student exercises, assignments and case studies. Additionally, it can be used to illustrate scientific publications through tailored examples or other visualizations. A lot of positive feedback from all over the world has shown that WoPeD has successfully been applied in all of these areas.

A couple of publications have accompanied the emerging development of the software, giving additional information on the underlying architecture [FrL03], on the used algorithms [Eck06], on some experimental visualization concepts [FlF06] and on new functional features [EcF08].

The development of WoPeD started in May 2003. The first public beta was brought-up in March 2005, since then several major releases have been made available. The current release 1.6 was provided in April 2008, the next major release 2.0 will be published by the end of 2008. Source code and installer packages are hosted on Sourceforge, a well-established development platform for open-source projects. Via Eclipse, the whole source code (all releases since 2005, including the current development branch) can be accessed in a straightforward way by anonymous synchronisation with the CVS repository.

WoPeD's architecture is based strictly on the Model-View-Controller (MVC) pattern. The data model is represented by standard PNML, the code to export to and import from the underlying file system is automatically generated from the XML schema. The graphical view is based on a Petri-net-specific implementation of the well-known JGraph framework [JGr08], a widely-used open-source-solution for drawing, transforming and visualizing graphs.

Main features

The followingt section gives a brief overview on the most important features of WoPeD, especially in the light of using the tool in the area of academic teaching as a "blended learning" instrument.

Graphical process editor with sub-process support

The major development goals of WoPeD are simplicity, lean architecture and intuitive usability. The graphical net editor is not only capable to represent traditional place-transition Petri nets, but also offers support for the class of workflow nets including special operators like XOR-split/join, AND-split/join, trigger symbols and resource assignments. It also provides functions to define hierarchical sub-processes as annotated transitions which act as placeholders for an embedded workflow net. By this, even huge process models can be represented graphically and analyzed by applying the associated algorithms only on the local process level (e. g. soundness check, reachability/coverability graph construction).

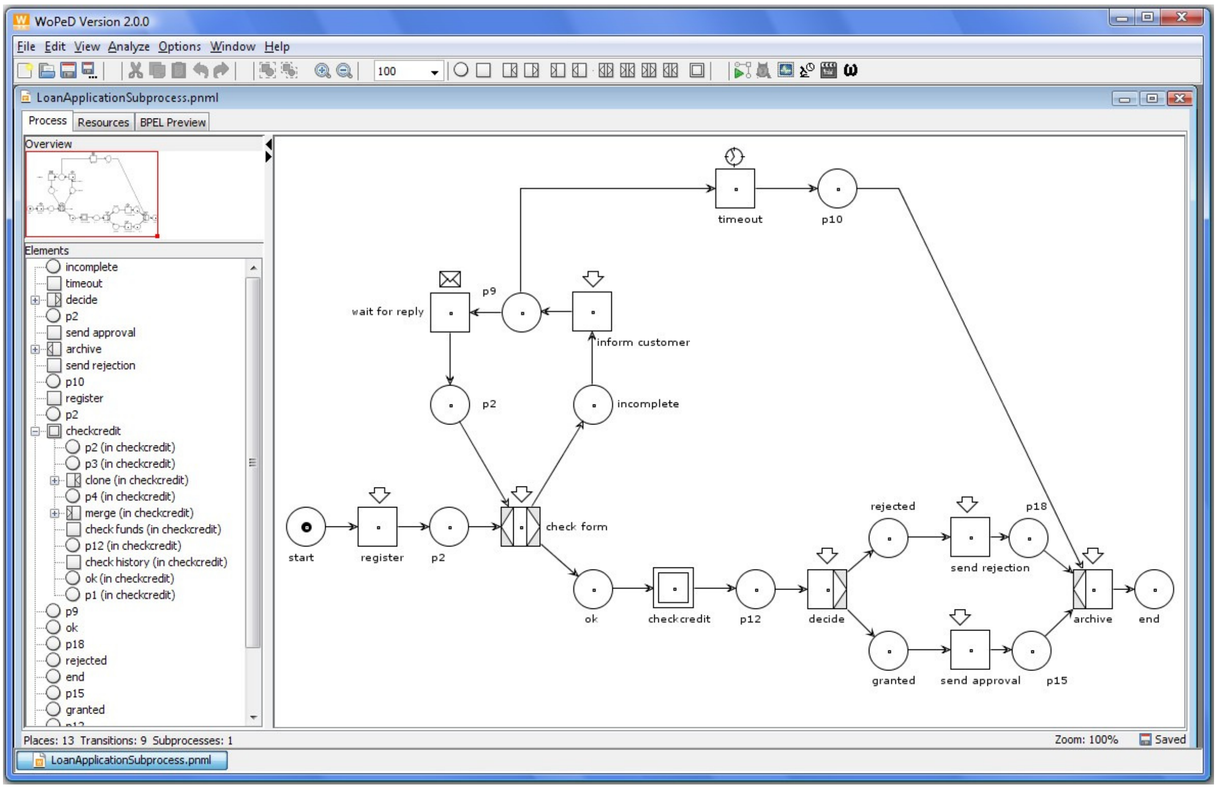


Fig. 1: A workflow net with a sub-process and process structure sidebar

PNML-compliant file format and multiple export formats

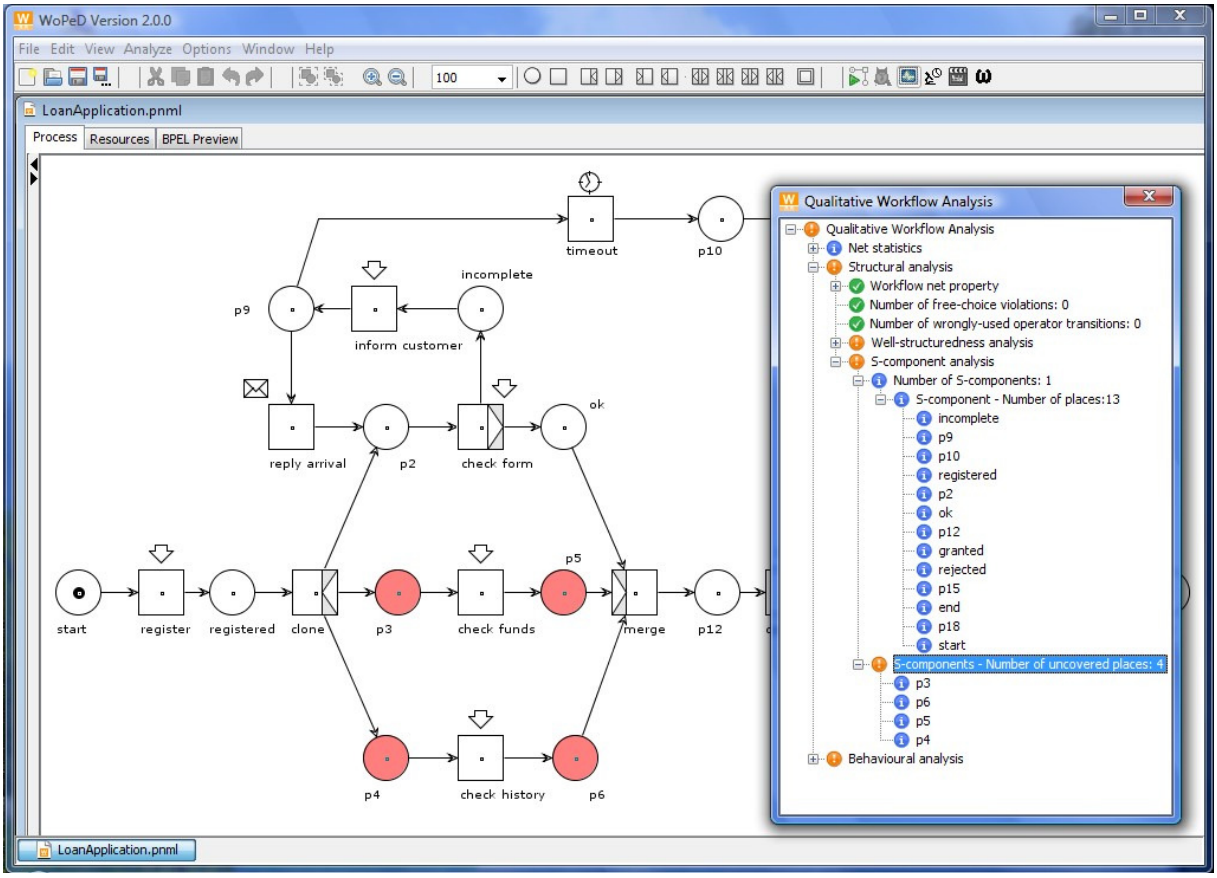
The standard file format of WoPeD is PNML [WeK03], allowing export or import of net models to and from other (PNML-based) Petri net tools. In order to guarantee a maximum of compatibility, the workflow net control flow extensions are internally translated into plain Petri net constructs and thus can be loaded also by tools unaware of the workflow-specific extensions. Additionally, WoPeD provides several export interfaces, including the JPEG, GIF and PNG graphics formats and textual formats like TPN (for Woflan interaction) and BPEL (see below).

Resource view editor and resource mapping

WoPeD contains a separate graphical resource modelling component to create and manage resource classes (i. e. groups and roles) and their contained resource objects (i. e. workflow participants) for each modelled process. Inside the process editor, each resource-triggered transition can be associated with a role and a group. The notation used for this also relates almost completely to the "van-der-Aalst/van Hee"-suggestions [AaH02]. By following this graphical approach, students intuitively get an idea on how the execution conditions of process models are connected to their organizational context.

Interactive token game

WoPeD provides a graphically animated token game simulator which can be controlled through a "remote-control"-like interface. It is possible to step forward and backward in single or in multiple steps, create a log of the chosen firing sequence and restore it. When encountering a sub-process transition, the user can choose between stepping over or stepping into the sub-process, very much comparable to graphical debugging in software development tools. The comfortable and highly intuitive interface helps students to get a deep understanding of process behaviour and how it is semantically represented by the Petri net model and the firing rule.



```
Qualitative Workflow Analysis
  Net statistics
  Structural analysis
    Workflow net property
    Number of free-choice violations: 0
    Number of wrongly-used operator transitions: 0
    Well-structuredness analysis
    S-component analysis
      Number of S-components: 1
        S-component - Number of places:13
          incomplete
          p9
          p10
          registered
          p2
          ok
          p12
          granted
          rejected
          p15
          end
          p18
          start
      S-components - Number of uncovered places: 4
        p3
        p6
        p5
        p4
  Behavioural analysis
```

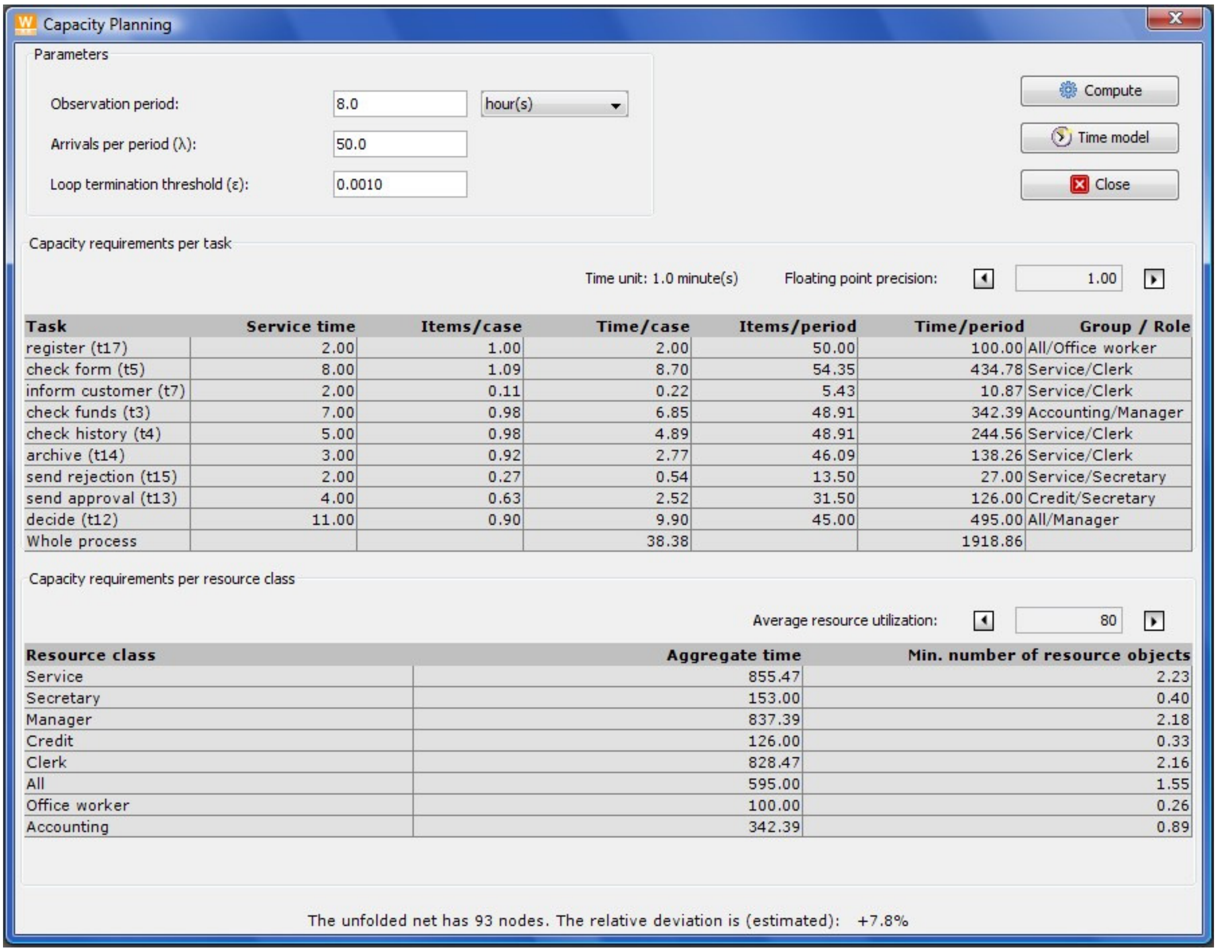
Fig. 2: Visualization of soundness violations, here: places uncovered by an S-component

Qualitative analysis

WoPeD can investigate a variety of qualitative properties, e. g. free-choice situations, S-component coverage, well-structuredness and soundness. Most of these properties are checked by built-in algorithms, except some runtime-critical parts of the soundness check which are computed by using an interface to the (non-graphical) Woflan tool [Wof08] which is developed at the TU Eindhoven. This "WoPeD-Woflan"-bridge allows the direct representation of the analysis results inside the associated workflow net graph and its nodes, visualizing control flow errors in process models and possible reasons for it on a graphical base[1]. The process editor forces all sub-processes to be workflow nets by themselves, and thus restricting them to have exactly one input and one output place. This has the

[1] The "WoPeD-Woflan" bridge is currently available for Microsoft Windows® operating systems only

interesting consequence that dynamical properties including soundness can be checked locally and separately for each sub-process.



Observation period: 8.0 hour(s)

Arrivals per period (λ): 50.0

Loop termination threshold (ε): 0.0010

Capacity requirements per task

Time unit: 1.0 minute(s) Floating point precision: 1.00

Task	Service time	Items/case	Time/case	Items/period	Time/period	Group / Role
register (t17)	2.00	1.00	2.00	50.00	100.00	All/Office worker
check form (t5)	8.00	1.09	8.70	54.35	434.78	Service/Clerk
inform customer (t7)	2.00	0.11	0.22	5.43	10.87	Service/Clerk
check funds (t3)	7.00	0.98	6.85	48.91	342.39	Accounting/Manager
check history (t4)	5.00	0.98	4.89	48.91	244.56	Service/Clerk
archive (t14)	3.00	0.92	2.77	46.09	138.26	Service/Clerk
send rejection (t15)	2.00	0.27	0.54	13.50	27.00	Service/Secretary
send approval (t13)	4.00	0.63	2.52	31.50	126.00	Credit/Secretary
decide (t12)	11.00	0.90	9.90	45.00	495.00	All/Manager
Whole process			38.38		1918.86	

Capacity requirements per resource class

Average resource utilization: 80

Resource class	Aggregate time	Min. number of resource objects
Service	855.47	2.23
Secretary	153.00	0.40
Manager	837.39	2.18
Credit	126.00	0.33
Clerk	828.47	2.16
All	595.00	1.55
Office worker	100.00	0.26
Accounting	342.39	0.89

The unfolded net has 93 nodes. The relative deviation is (estimated): +7.8%

Fig. 3: Capacity planning report computed from workflow and resource model

Quantitative analysis and capacity planning

WoPeD is able to store and visualize an average service time value with each resource-trigged transition and an average branching probability with each outgoing arc of an implicit or explicit XOR-split operator. This allows the computation of capacity planning scenarios derived from both process and resource models, assigning each transition the expected number of work items per case, and each resource class the minimum required number of members under a given resource utilization rate. The algorithm to compute the number of work items per case is based on a finite prefix of the net unfolding being capable to handle possibly infinite loops. Apart from this, WoPeD contains a discrete event simulator which is able to compute quantitative results like average waiting time and resource usage based on random number generators with various distribution functions. The simulations use the underlying process and resource model and can be configured by a variety of parameters such as average arrival rate of cases, average service time of transitions and average branching probability of XOR-splits.

Reachability and coverability graph visualization

The new release 2.0 of WoPeD implements automatic reachability and coverability graph construction and visualization. The current implementation provides two simple layout mechanisms for the visual representation, leaving the manual "fine-tuning" of node positions to the user. Once displayed in a satisfying way, the coverability graph can also be exported to the most common graphics formats such as JPEG, BMP and PNG. By this, WoPeD can be used to create sample graphs e. g. for lecture material or other publications.

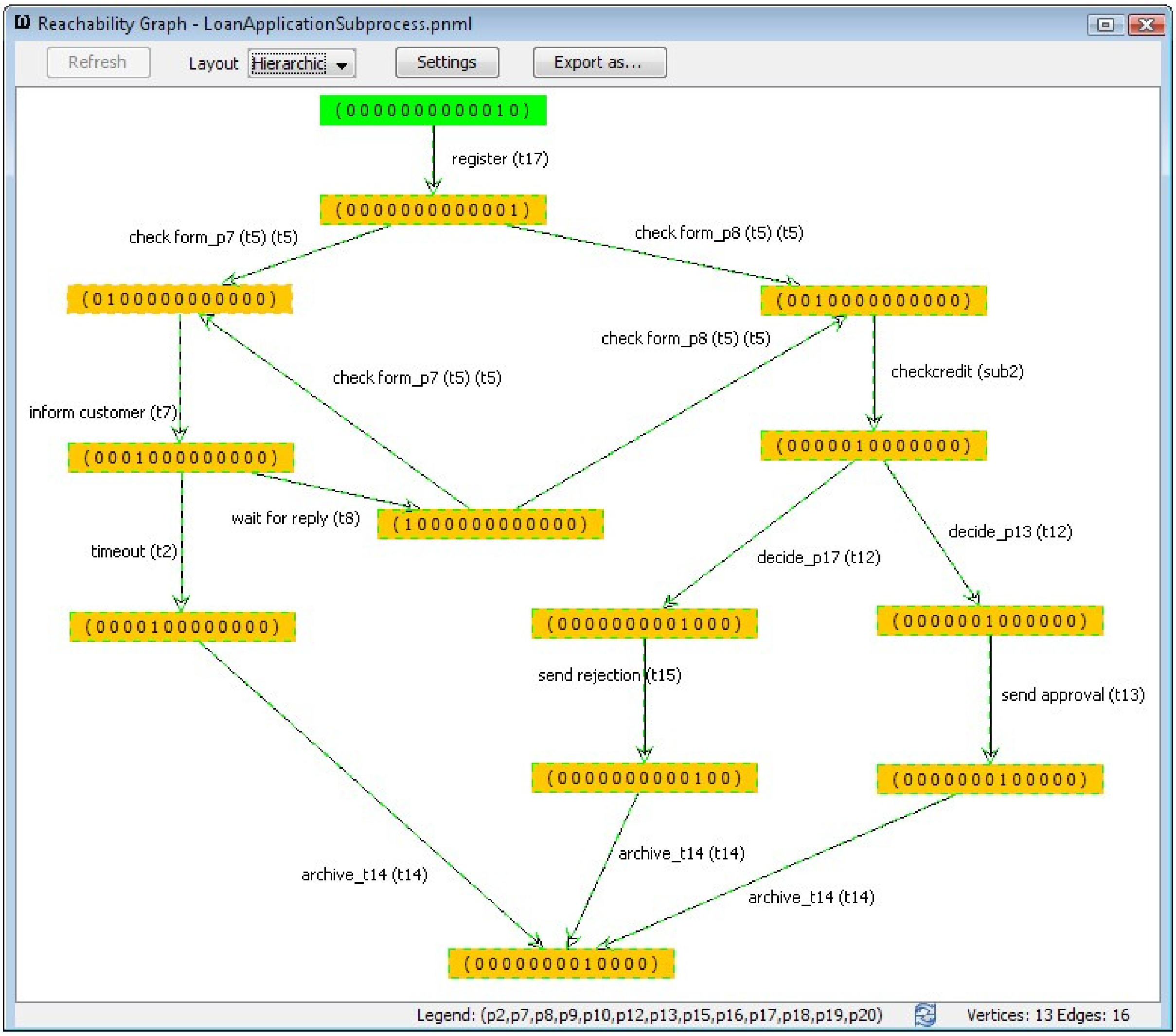


Fig. 4: Coverability tree construction and visualization

As the only known Petri net tool, WoPeD is able not only to construct behavioural information in memory, but also to visualize the associated graph on the screen. By, this, students learn about the impact of their modelling decisions on the associated state space and get a good visual impression of Petri net behaviour in general and the "state explosion" phenomenon in particular.

BPEL export

An important new feature of WoPeD 2.0 is the export of well-structured, free-choice workflow nets into the widely-used BPEL format. The process control flow elements are converted into the associated BPEL constructs and single transitions can be used as placeholders for basic BPEL operations (*assign, invoke, receive, reply, wait*). A global namespace is supported for defining state variables which can be used as parameters when interacting with web-services. By this, WoPeD allows the orchestration of arbitrary web services identified by partner links entered manually or imported from UDDI business registries. The parser used to convert the workflow net control flow into an executable BPEL script is based on the ideas published in [AaL08] and [Las06].

Conclusion

WoPeD is an evolving software tool with lots of useful functional features. The editing component supports a process model view as well as a resource model view. WoPeD fully supports the original workflow net notation and contains algorithms for checking qualitative properties (soundness) as well as quantitative properties (capacity planning). With its high degree of intuitiveness and its easy-to-use interface, WoPeD is an ideal instrument for "blended learning" in the context of teaching and publishing in the area of workflow management and process analysis. For a closer look at the tool's functionality, including download links, screenshots and documentation, please refer to the website [WoP08].

References

[AaH02] W. M. P. van der Aalst, K. van Hee. Workflow Management – Models, Methods and Systems. MIT Press, Cambridge, 2002.

[AaL08] W. M. P. van der Aalst, K. B. Lassen: Translating unstructured workflow processes to readable BPEL: Theory and implementation. Information & Software Technology 50(3): 131-159 (2008).

[FlF06] C. Flender, T. Freytag - Visualizing the Soundness of Workflow Nets.
13th Workshop *Algorithmen und Werkzeuge für Petrinetze* (AWPN), Hamburg, 2006.

[EcF08] A. Eckleder, T. Freytag. WoPeD 2.0 goes BPEL 2.0. 15th Workshop *Algorithmen und Werkzeuge für Petrinetze* (AWPN), Rostock, 2008.

[FrL03] T. Freytag, S. I. Landes. PWFtool – a Petri net workflow modelling environment. 10th Workshop *Algorithmen und Werkzeuge für Petrinetze* (AWPN), Eichstätt, 2003.

[JGr08] The JGraph Homepage: www.jgraph.com, visited on Oct 1st, 2008.

[Las06] K. B. Lassen, W. M. P. van der Aalst: WorkflowNet2BPEL4WS: A Tool for Translating Unstructured Workflow Processes to Readable BPEL. OTM Conferences (1): 127-144 (2006).

[WeK03] M. Weber, E. Kindler. The Petri Net Markup Language. In: H. Ehrig, W. Reisig, G. Rozenberg, H. Weger (Eds.). Petri Net Technology for Communication Based Systems, LNCS 2472, Springer 2003.

[VBA01] H. M. W. Verbeek, T. Basten, W. M. P. van der Aalst: Diagnosing Workflow Processes using Woflan. Computer Journal 44(4): 246-279 (2001).

[Wof08] The Woflan Homepage: www.is.tm.tue.nl/research/woflan, visited on Oct 1st, 2008.

[WoP08] The WoPeD Homepage: www.woped.org, visited on Oct 1st, 2008.

Quick facts (from Petrinet tools database)

Homepage	www.woped.org
Email contact	info@woped.org
Availability	Free (open-source)
Platforms	Independent (Java) One-file-installers for Windows, Linux and MacOS
Net class	Workflow nets, place-transition nets
File formats	PNML, TPN (export), JPEG (export), GIF (export), PNG (export), BPEL (export)
Features	Graphical editor, sub-processes, interactive simulator, qualitative analysis, soundness check, capacity planning, quantitative simulation