

Combining a Reachability Graph and a Reduction Rule Approach for Verification of EPCs

J. Mendling

Vienna University of Economics and Business Administration, Austria,
jan.mendling@wu-wien.ac.at

Abstract. This demonstration shows how two complementary tools can be combined for the efficient verification of EPC business process models. In a first step, the tool *xoEPC* is used to identify control-flow errors in an EPC based on reduction rules. While this approach is efficient, it may result in a partially reduced EPC since the set of reduction rules is not complete. In a second step, this partially reduced EPC can then be loaded into ProM. In ProM there is a conversion plug-in that calculates the reachability graph and verifies EPC soundness. Though this approach is complete, it can be quite time-consuming. Using the two tools in sequence balances efficient processing of the reduction rules approach and the completeness of the reachability graph analysis.

1 Introduction

Conceptual business process models such as Event-driven Process Chains (EPCs) play an important role in the business process management life cycle. Since they are often used as an early formalization of process requirements, they should be subject to quality assurance in order to avoid costly rework in later in the implementation phase [9, 11]. Verification is related to one particular quality aspect, namely, that business process models should guarantee that every case can be processed with a proper completion. In comparison to other business process modeling languages that are based on Petri nets, EPCs have to features that makes it difficult to do verification with standard Petri net technologies. Firstly, EPCs may have multiple start and end events. Since Soundness [1], as a prominent correctness criterion, is only defined for Workflow nets, i.e. a Petri net class with exactly one source and one sink, it is not directly applicable for EPCs. Secondly, EPCs may have OR-join connectors representing conditional synchronization semantics. Some of these verification challenges have been addressed with the relaxed soundness criterion, and a mapping of EPCs to Petri nets as defined in [3]. Still, proper completion is not guaranteed for a relaxed sound EPC.

For this demonstration we take the novel EPC semantics definition and a corresponding notion of EPC soundness as a starting point (see [7, 6]). EPC soundness basically requires (1) that there is a set of initial markings such that every start event is included in at least one of these initial markings, (2) that for every marking, that is reachable from a marking in the set of initial markings, there exists a final marking that can be reached, and (3) that final markings are the only markings reachable from a marking in the set of initial markings such that no node can fire [6, p.106]. In Section 2,

we briefly sketch the functionality of *xoEPC* and discuss its different outputs. In Section 3 illustrate the plug-in that calculates the reachability graph for an EPC, and gives the result of the verification of EPC soundness. Section 4 summarizes the paper.

2 Application of Reduction Rules with *xoEPC*

xoEPC a tool for EPC verification based on reduction rules.¹ It is written in the object-oriented scripting language XOTcl [10], which is an extension of Tcl. *xoEPC* loads all *.xml files from the current directory and checks whether they are ARIS XML files [4]. If they are, the XML is processed with the tDOM package, a Tcl implementation of the DOM specification.² For each EPC model that has at least one event and one function, *xoEPC* checks syntactical correctness and applies the reduction algorithm. The internal data structure of *xoEPC* uses an adjacency matrix representation of the EPC, and the reduction methods work on this data structure. *xoEPC* produces three types of files:

1. All EPCs that cannot be reduced completely are written to the reducedEPCs.epml file. These EPCs can be further analyzed by tools that can read EPML [8].
2. Errors encountered by reduction rules are recorded in the errorresults.xml file. This file also records the processing time of the reduction, metadata of the model, as well as the size of the original and the size of the reduced EPC.
3. Finally, an XHTML file with an embedded SVG graphic³ is generated for each EPC based on the position information in the ARIS XML file. It projects the errors back to the model by highlighting the involved connectors.

The advantages of *xoEPC* are the following four points. Firstly, due to its reduction rule approach it is performative, providing a result for many EPCs within less than one second [6]. Secondly, since it is based on EPC soundness, it returns a precise result also for EPCs with multiple start and end events as well as with OR-joins. Thirdly, the program runs in a batch mode, and is therefore able to analyze not only one, but several EPC models, without user interaction. Finally, *xoEPC* projects the errors back onto the visual model such that they can be easily corrected. Still, it might happen that the original EPC is not reduced completely. In this case, the partially reduced EPC can be further analyzed with ProM by importing the reducedEPCs.epml file.

3 Calculation of Reachability Graph with ProM

Based on the EPC semantics defined in [6], we have implemented a conversion plug-in for the *ProM* (Process Mining) framework [2]. ProM was originally developed as a tool for *process mining*, but meanwhile its functionality was extended to include other types

¹ Some of the functionality of *xoEPC* is available at <http://wi.wu-wien.ac.at/epc>.

² For an overview of the various DOM specifications of the World Wide Web Consortium refer to <http://www.w3.org/DOM/DOMTR>.

³ For the respective specifications, see <http://www.w3.org/TR/xhtml1> and <http://www.w3.org/TR/SVG>.

of analysis, model conversions, model comparison, etc. This was enabled by the plug-able architecture of ProM, that allows to add new functionality without changing the framework itself, and the fact that ProM supports multiple modeling languages. Since ProM can interact with a variety of existing workflow management systems, simulation tools, ERP systems, and analysis tools (cf. [2]), the plug-in for the new EPC semantics can easily be used for the analysis of existing models. Currently, there are more than 150 plug-ins in release 4.1. ProM basically supports five kinds of plug-ins for mining, importing, exporting, conversion, and analysis.

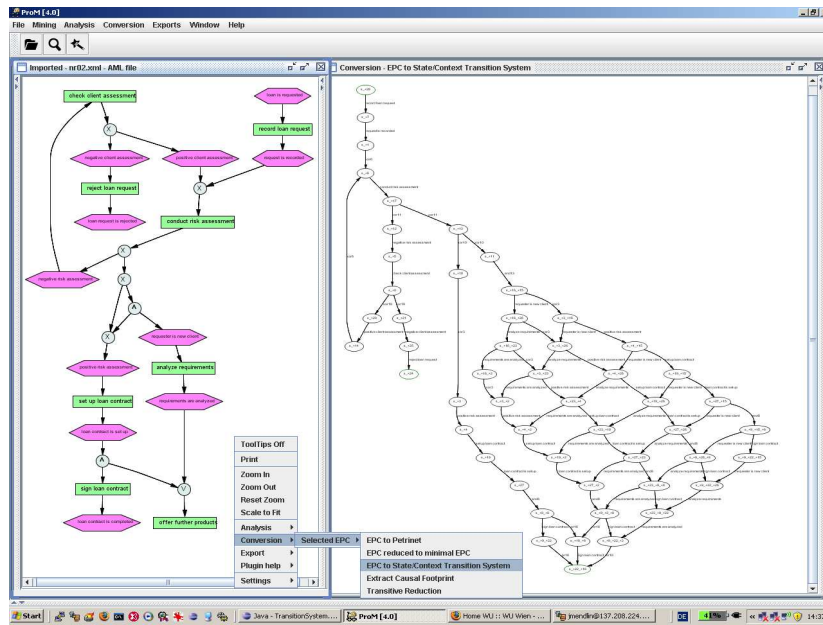


Fig. 1. Calculating the reachability graph in ProM

The conversion plug-in maps an EPC to the transition systems package that was developed for an implementation of the incremental workflow mining approach by *Kindler, Rubin, and Schäfer* [5]. Figure 1 illustrates how the conversion plug-in works. First, one has to load an EPC business process model into ProM, for instance, by using the import plug-in for the ARIS XML format [4] or for the EPC Markup Language [8]. In the figure, the EPC example model for a loan request process is loaded. Since ProM generates a new layout automatically, the model might look different compared to the original layout. Once the EPC is displayed in ProM, one can click on it, trigger the conversion plug-in “EPC to State/Context Transition System”, and the reachability graph is calculated and shown in a new ProM window. The dense network of states and transitions on the right-hand side stems from the concurrent execution, if there is both a positive risk assessment for the loan request and the requester is a new customer. There are two markings that do not serve as a source for another transition in case if

the request is rejected or accepted. Both these markings are displayed with a green border since they are proper final markings. If they were deadlocks, they would be drawn with a red border. The advantage of the reachability graph analysis is that the result is complete.

4 Conclusion and Future work

In this demonstration, we have presented a combination of two tools for the verification of EPCs. Firstly, *xoEPC* provides a fast analysis of EPCs with the shortcoming that not all EPCs might be reduced completely. In a second step, only partially reduced EPCs can be further analyzed with a ProM plug-in that explicitly calculates the reachability graph. Since partially reduced EPCs tend to be quite small [6], this combination of tools efficiently circumvents the state explosion problem frequently encountered in process model verification. Further details on both tools are reported in [6].

References

1. W.M.P. van der Aalst. Verification of Workflow Nets. In Pierre Azéma and Gianfranco Balbo, editors, *Application and Theory of Petri Nets 1997*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426. Springer Verlag, 1997.
2. P. Barborka, L. Helm, G. Köldorfer, J. Mendling, G. Neumann, B.F. van Dongen, H.M.W. Verbeek, and W.M.P. van der Aalst. Integration of EPC-related Tools with ProM. In M. Nüttgens and F.J. Rump and J. Mendling, editor, *Proceedings of the 5th GI Workshop on Business Process Management with Event-Driven Process Chains (EPK 2006)*, pages 105–120, Vienna, Austria, December 2006. German Informatics Society.
3. J. Dehnert and W.M.P. van der Aalst. Bridging The Gap Between Business Models And Workflow Specifications. *International J. Cooperative Inf. Syst.*, 13(3):289–332, 2004.
4. IDS Scheer AG. *XML-Export und -Import (ARIS 6 Collaborative Suite 6.2 Schnittstellenbeschreibung)*. <ftp://ftp.ids-scheer.de/pub/ARIS/HELPDESK/EXPORT/>, 2003.
5. E. Kindler, V. Rubin, and W. Schäfer. Process Mining and Petri Net Synthesis. In J. Eder and S. Dustdar, editors, *Business Process Management Workshops*, volume 4103 of *Lecture Notes in Computer Science (LNCS)*, pages 105–116. Springer Verlag, September 2006.
6. J. Mendling. *Detection and Prediction of Errors in EPC Business Process Models*. PhD thesis, Vienna University of Economics and Business Administration, 2007.
7. J. Mendling and W.M.P. van der Aalst. Formalization and Verification of EPCs with OR-Joins Based on State and Context. In J. Krogstie, A.L. Opdahl, and G. Sindre, editors, *Proceedings of the 19th Conference on Advanced Information Systems Engineering (CAiSE 2007)*, volume 4495 of *Lecture Notes in Computer Science*, pages 439–453, Trondheim, Norway, 2007. Springer-Verlag.
8. J. Mendling and M. Nüttgens. EPC Markup Language (EPML) - An XML-Based Interchange Format for Event-Driven Process Chains (EPC). *Information Systems and e-Business Management*, 4(3):245 – 263, 2006.
9. D.L. Moody. Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, 55(3):243–276, 2005.
10. G. Neumann and U. Zdun. XOTcl, an Object-Oriented Scripting Language. In *Proc. of the 7th USENIX Tcl/Tk Conference, Austin, Texas, USA*, 2000.
11. M. Rosemann. Potential pitfalls of process modeling: part a. *Business Process Management Journal*, 12(2):249–254, 2006.