DaGen: A Tool for Automatic Translation from DAML-S to High-Level Petri Nets

Daniel Moldt and Jan Ortmann

University of Hamburg, Computer Science Department
Vogt-Kölln-Str. 30, D-22527 Hamburg
{moldt,ortmann}@informatik.uni-hamburg.de

Abstract. The Internet provides possibilities for distributed execution of business processes and Web Services. This caused the emergence of a variety of Web Services that might be composed to accomplish tasks. To efficiently compose these tasks a simple workflow description no longer suffices. We therefore suggest a description based on high-level Petri nets called reference nets, allowing for the consideration of pre- and post-conditions of services offered on the Internet. We demonstrate how DAML-S models can be automatically translated into high-level nets and thus can be directly executed in such contexts, including Petri net based MAS.

Keywords: high-level Petri nets, nets within nets, reference nets, RENEW, workflow, web service, business process, DAML-S, process ontology

1 Introduction

The modelling of workflows and processes with Petri nets has been thoroughly investigated (see [2, 16, 1]). Petri nets offer a clear semantics and a rather intuitive way to process modelling. This makes them both easy to handle for humans and unambiguously processable by machines. Based on RENEW¹ (Reference Net Workshop, see [13]), a Petri net simulator, written in Java, several approaches for the modelling of workflows with Petri nets have been made (see [7, 16]). We adopt those approaches and transfer them to the area of Web Services. The composition of Web Services is strongly related to the modelling of workflows.

RENEW allows for the execution of Java code and is therefore capable of not only graphically describing a composition of Web Services but also of executing the described composition. Since process descriptions in DAML-S use XML, they are hard to read without support. Additionally DAML-S allows for concept hierarchies, so that a sub-concept might inherit properties from a super-concept. This makes it rather difficult to grasp the meaning of a DAML-S description. Based on RENEW, we developed a tool to read DAML-S descriptions, show them as Petri net (reference net) and execute them from within the RENEW simulator, following the ideas introduced in [14], that showed that DAML-S descriptions can be modelled as Petri nets.

¹ RENEW is freely available from <http://www.renew.de/>

2 Conceptual Background

This tool heavily relies upon the RENEW simulator. RENEW is a Petri net simulator that allows for the graphical drawing and for the execution of reference nets (see [9, 12, 11, 10]). *Reference nets* represent an extension of the Coloured Petri net (CPN) formalism (see [8]) adding both the concept of nets within nets introduced by [15] and the concept of synchronous channels (see [5]). Additionally they provide means to execute Java code directly from within the Petri net through inscriptions on net elements. Furthermore, RENEW has a plug-in mechanism, so that it can easily be extended by new functionality. Furthermore new net classes can be created, loaded as net instances and can therefore be executed.

Having started the RENEW-application, two windows appear: a toolbar window and a drawing window. The former allows to pick a drawing tool which can then be used in the latter window. By picking places, transitions, arcs and inscriptions, the user can easily draw a Petri net. These Petri nets can then be executed, which will then be displayed in an extra window, so that the user can follow the execution of the net. Figure 1 shows the RENEW-simulator with both windows mentioned above. Here a synchronous channel and execution of Java code (via the action inscription) is modelled. Via synchronous channels different net instances can communicate.

The RENEW simulator fully supports the execution of reference nets. **action** inscriptions make the simulator execute the Java code following the **action** keyword. This way arbitrary code can be executed.

DAML-S² is based on DAML+OIL³ and provides the framework for a process ontology to describe Web Services. Just like DAML+OIL, DAML-S is closely related to description logics. It therefore allows for the discovery of Web Services through inference. A general category of Web Services can be investigated and the services best suited for a given task can be picked. Besides this categorisation of Web Services, DAML-S also provides information about the input and the output parameters as well as the preconditions and the effects of a service (IOPE).

3 Tool

The tool introduced here is a plug-in for RENEW. It allows to read a DAML-S description, which will then be displayed as a set of Petri nets. The Petri nets generated allow different views on the process, such as the control flow, the data and the operations. These views were partly adopted from the views on workflows discussed in [3]. The different nets, that have been generated can be executed by the simulator. To exchange data, they communicate with each other through synchronous channels. In order to get a quick overview, the user is able to merely look at the control flow of a process and in order to get a deeper insight it is possible to consider each process separately along with its IOPE.

² <http://www.daml.org/services/>

³ <http://www.daml.org/language/>

Fig. 1. RENEW

The Petri nets of a process description are assembled through templates. These templates specify the basic net layouts of the control constructs allowed in DAML-S and they contain the templates for atomic and for composite processes. These templates are placed in the order given by the process description.

To load a DAML-S description one can simply pick the file the service description is located in. If there is only one service description in this file, this description will be opened. If there are several such descriptions, the user can pick the one he or she wants to open.

4 Current Use

Within our Petri net based FIPA⁴ compliant multi-agent framework CAPA (see [6]), this tool can as well be used to have agents invoke Web Services dynamically. This way we have enabled a FIPA compliant multi-agent platform to interact with Web Services. Through an integration of CAPA into the Agentcities.net framework we plan to offer these services to a broader community.

⁴ <http://www.fipa.org>

Apart from them we are currently investigating in how far Web Services can be modelled through workflow techniques. Through this tool, workflows using Web Services can be executed without any need of other tools.

5 Conclusion and Outlook

We use a special kind of high-level Petri nets – reference nets – as the basic description technique and formalism for processes and workflows. The technical implementation is directly possible with the RENEW-tool. Our integration into the agent domain as an open framework results by adding our FIPA-compliant extension CAPA.

As the major contribution here we see the visualisation of DAML-S process descriptions as well as their execution and their integration into a FIPA compliant multi-agent system. Furthermore we see reference nets as a major modelling technique for processes. They have a formal semantics, their modelling is supported by a tool – RENEW – and they have successfully been applied to a wide range of different fields.

What is missing now is the direction from reference nets which have been modeled directly or which have been derived from other models (e.g. from UML or AUML interaction diagrams (see [4])) to DAML-S descriptions. Apart from that it would be of great interest to verify certain properties of the nets generated or of those to be exported as DAML-S.

Another challenge would be the integration of ontology tools to support the development of process ontologies in DAML-S as well as the development of static ontologies in DAML+OIL. Here an integration of tools like Protege⁵ or OilEd⁶ is envisioned.

In the context of Petri nets workflow patterns need to be integrated into nets simplifying the creation of workflows. Here some approaches exist within the scope of agent protocols (see [4]).

References

1. Wil M.P. van der Aalst and Arthur H.M. ter Hofstede. Workflow pattern homepage. Technical report, URL: <http://tmitwww.tm.tue.nl/research/patterns/>, 2003.
2. Wil van der Aalst. Verification of workflow nets. Number 1248 in Lecture Notes in Computer Science, pages 407–426, Berlin Heidelberg New York, 1997. Springer Verlag.
3. Wil van der Aalst, Daniel Moldt, Rüdiger Valk, and Frank Wienberg. Enacting Interorganizational Workflows Using Nets in Nets. In Jörg Becker, Michael zur Mühlen, and Michael Rosemann, editors, *Proceedings of the 1999 Workflow Management Conference Workflow-based Applications, Münster, Nov. 9th 1999*, Working Paper Series of the Department of Information Systems, pages 117–136, University of Münster, Department of Information Systems, Steinfurter Str. 109, 48149 Münster, 1999. Working Paper No. 70.

⁵ <http://protege.stanford.edu/index.html>

⁶ <http://oiled.man.ac.uk/>

4. Lawrence Cabac, Daniel Moldt, and Heiko Rölke. A proposal for structuring petri net-based agent interaction protocols. In Wil van der Aalst and Eike Best, editors, *Proc. of 24th International Conference on Application and Theory of Petri Nets 2003 (ICATPN 2003)*, Eindhoven, NL, Berlin Heidelberg New York, 2003. to be published in Lecture Notes in Computer Science, Springer-Verlag.
5. Søren Christensen and Niels Damgaard Hansen. Coloured Petri Nets Extended with Channels for Synchronous communication. Technical Report DAIMI PB-390, Computer Science Department, Aarhus University, DK-8000 Aarhus C, Denmark, April 1992.
6. Michael Duvigneau, Daniel Moldt, and Heiko Rölke. Concurrent architecture for a multi-agent platform. In Fausto Giunchiglia, James Odell, and Gerhard Weiß, editors, *Third International Workshop, AOSE 2002, Bologna, Italy, July 15, 2002, Revised Papers and Invited Contributions*, volume 2585 of LNCS, Berlin Heidelberg New York, 2003. Springer.
7. Thomas Jacob. Implementierung einer sicheren und rollenbasierten Workflowmanagement-Komponente für ein Petrinetzwerkzeug. Diplomarbeit, Universität Hamburg, Fachbereich Informatik, 2002.
8. K. Jensen. High Level Petri Nets. In A. Pagoni and G. Rozenberg, editors, *Applications and Theory of Petri Nets*, number 66 in Informatik Fachberichte, pages 166–180, Berlin, 1983. Springer-Verlag.
9. Olaf Kummer. Simulating synchronous channels and net instances. In Jörg Desel, Peter Kemper, Ekkart Kindler, and Andreas Oberweis, editors, *5. Workshop Algorithmen und Werkzeuge für Petrinetze*, Forschungsbericht Nr. 694, pages 73–78. Fachbereich Informatik, Universität Dortmund, 1998.
10. Olaf Kummer. Introduction to Petri nets and reference nets. *Sozionik Aktuell*, 1:1–9, 2001. ISSN 1617-2477.
11. Olaf Kummer. *Referenznetze*. Logos Verlag, Berlin, 2002.
12. Olaf Kummer, Annette Laue, Matthias Liedtke, Daniel Moldt, and Heiko Rölke. Höhere Petrinetze zur kompakten Modellierung und Implementierung von Verhalten. In Holger Giese and Stephan Philippi, editors, *Visuelle Verhaltensmodellierung verteilter und nebenläufiger Software-Systeme*, 8. Workshop des Arbeitskreises GROOM der GI Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung, 13.-14. November 2000, Universität Münster, pages 27–32, November 2000. Techreport 24/00-I.
13. Olaf Kummer, Frank Wienberg, and Michael Duvigneau. Renew – The Reference Net Workshop. WWW-Dokument unter <http://renew.de/>, 2003.
14. Srinu Narayanan and Sheila A. McIlraith. Simulation, verification and automated composition of web services. In *Proceedings of the eleventh international conference on World Wide Web*, pages 77–88, Honolulu, Hawaii, 2002. ACM Press.
15. R. Valk. Petri nets as token objects - an introduction to elementary object nets. In J. Desel and M. Silva, editors, *Proc. Application and Theory of Petri Nets, Lisbon, Portugal*, number 1420 in LNCS, pages 1–25, Berlin, 1998. Springer-Verlag.
16. W. M. P. van der Aalst and K. Anyanwu. Inheritance of interorganizational workflows to enable business-to-business E-commerce. In *Proceedings of the Second International Conference on Telecommunications and Electronic Commerce (ICTEC'99)*, pages 141–157. Nashville, Tennessee, October 1999.