

Experimentation and Output Interchange for Petri Net Models

Marc Melià
Catalina M. Lladó
Universitat de les Illes Balears
Departament de Ciències
Matemàtiques i Informàtica
Ctra de Valldemossa, Km. 7.6,
07071
Palma de Mallorca, Spain
marcnma@gmail.com,
cllado@uib.es

Connie U. Smith
Performance Engineering
Services
PO Box 2640 Santa Fe
New Mexico, 87504-2640 USA
www.spe-ed.com

Ramon Puigjaner
Universitat de les Illes Balears
Departament de Ciències
Matemàtiques i Informàtica
Ctra de Valldemossa, Km. 7.6,
07071
Palma de Mallorca, Spain
putxi@uib.es

ABSTRACT

Performance model interchange formats are common representations for data that can be used to move models among modeling tools. The Experiment Schema Extension (Ex-SE) provides a means of specifying performance studies (model runs and output). It is independent of a given tool paradigm, e.g., it works with PMIF, S-PMIF, LQN, Petri nets and other paradigms. Petri nets are different from the other paradigms, however, because they provide additional representation and analysis capabilities in addition to performance analysis. Examples include constraints on tokens in places, invariant analysis, reachability analysis, and so on. So to capitalize on these additional capabilities, this paper presents a specific, extended instantiation of the Ex-SE for Petri nets (PN-Ex). The viability of the approach is demonstrated with a case study carried out using PIPE2 (Platform Independent Petri net Editor 2) with an experimental framework.

Categories and Subject Descriptors

D.2.8 [Software Engineering]: Metrics—*performance measures*; C.4 [Modelling Techniques]: Experimentation

General Terms

Performance, Experimentation

1. INTRODUCTION

XML-based interchange formats for performance models provide a mechanism whereby performance model information may be transferred among modeling tools. This makes it possible for a user to create a model in one tool, perform

some studies, and then move the model to another tool for other studies that are better done in the second tool.

For example, interchange formats have been specified for: queueing network models: PMIF (*Performance Model Interchange Format*) [16], LQN (*Layered Queueing Networks*), UML (*Unified Modelling Language*), Petri Nets, and others. These formats specify the model and a set of parameters for one run. For model studies, however, it is useful to be able to specify multiple runs, or experiments, for the model.

In [17] an XML interchange schema extension, called Experiment Schema Extension (Ex-SE), defines a set of model runs and the output desired from them. This extension to an interchange schema provides a means of specifying performance studies that is independent of a given tool paradigm. In [17], the use of the Ex-SE is illustrated with an instance of the extension in which the interchange schema is the PMIF, called PMIF-Ex.

Petri nets are different from the other paradigms, however, because they provide additional representation and analysis capabilities in addition to performance analysis. Examples include constraints on tokens in places, invariant analysis, reachability analysis, and so on. So to capitalize on these additional capabilities, this paper presents a specific, extended instantiation of the Ex-SE for Petri nets (PN-Ex). It then demonstrates the viability of the approach with a case study solved using the Experiment for PIPE2 (Platform Independent Petri net Editor 2) [15].

The paper first covers related work in several areas. Then we present an overview of the Experiment Schema Extension (Ex-SE) followed by a description of the specific extensions in the Petri Net instance (PN-Ex). Next, a case study illustrates the usefulness of the approach.

2. RELATED WORK

Related work falls into two categories: interchange formats for Petri net models and tool interfaces.

An overview of work that has addressed the abstract information requirements for modeling studies is presented in [17]. It is not covered here due to lack of space.

2.1 Interchange Formats for Petri Nets

The Petri net community interest in developing a Petri net standard interchange format began with a proposal in the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WOSP'08, June 24–26, 2008, Princeton, New Jersey, USA.
Copyright 2008 ACM 978-1-59593-873-2/08/06 ...\$5.00.

ISO project [13]. One of the outcomes of this effort is the Petri Net Markup Language (PNML) [2, 6], an XML-based interchange format for Petri nets. Workshops focussed on PNML have followed, for example [14], though unfortunately, there is not yet a well-accepted interchange format for Petri nets nor many working tools supporting the format. The PNML specification [2] distinguishes between general features of all types of Petri nets and specific features of a specific Petri net type. The specific features are defined in a separate Petri Net Type Definition (PNTD) for each type. The only PNTD published so far is the *Petri net type for Place/Transition nets*, which defines the basic features of Place/Transition Petri nets as in [11].

For performance analysis, we need to specify Petri nets with temporal specifications, for example GSPN (Generalised Stochastic Petri Nets) [11]. Unfortunately, there is no PNTD that defines the necessary performance specifications. Moreover, we would also like to be able to specify features such as capacity restriction, priorities or inhibitor arcs since they improve the modelling power of a tool and those are not included in the PNTD published.

On the other hand, TimeNet (TIMEd Net Evaluation Tool) [5, 20], a tool for modelling and analysis of stochastic Petri nets uses an xml based format for the specification of the nets that defines the specifications necessary for performance analysis. This format is defined by an xml schema, *eD-SPN.xsd*, that can be integrated very easily with the Experiment Schema Extension to create a specification for Petri Net experiments and output (PN-Ex).

2.2 Petri Net Tools Supporting Experiments

Tool interfaces may also include experimental capabilities that define the input, output, and control information that can be used to perform modeling studies. For example, the Möbius modeling environment includes quite sophisticated features for experiment design [9]. It supports two types of two-level design strategies: Plackett-Burman and two-level factorial design. It also supports two types of response surface designs (central composite and Box Behnken) to further refine the empirical model of the system once the user has collected enough information using the factorial designs.

Solutions of these models are currently conducted within the Möbius environment. These experiments are higher-level specifications for model solutions than those specified by the Ex-SE. It would be possible to create an equivalent specification using Ex-SE. For example, it would have the multiple combinations of upper and lower bound parameter values in the two-level factorial design and specify the results desired from each solution. Another option is to use a ToolCommand specification for those tools that recognize the higher level experimental design specifications.

Sharpe [18, 12] (Symbolic Hierarchical Automated Reliability and Performance Evaluator): a tool for specifying and analyzing performance, reliability and performability models that allows for different model types, including Petri nets. Output measures of a model can be used as parameters of other models which facilitates the hierarchical combination of different model types. Sharpe supports a subset of the experiments that can be defined with the Ex-SE. For example, an experiment can be specified for a single variable with a *Start value*, *Stop value* and *Increment value*. The desired output from the analysis can be also specified in Sharpe, see

section 4.1 for more details. For example Sharpe supports plots of output values for the single variable experiment.

3. EXPERIMENT SCHEMA EXTENSION (EX-SE)

The Experiment Schema Extension (Ex-SE) is an extension to performance model interchange formats for defining a set of model runs and the output desired from them [17]. It incorporates elements from each of these approaches to provide a comprehensive solution for experimentation. The Ex-SE provides a means of specifying performance studies that is independent of a given tool paradigm, e.g., it works with PMIF, S-PMIF, LQN, and other paradigms. It requires only that a tool support the Ex-SE or have an interface that is capable of reading/writing extended interchange files.

The Experiment schema specifies (A) Variable declaration(s):

- Variables - assign values to an attribute of the model, iterate over it, and so on.
- LocalVariables - expressions that combine variables in solution specifications
- OutputVariables - concrete results from a solution that determine subsequent actions,

(B) Solution specification(s) - the models runs and parameter values desired:

- Assignment - assigns values to model attributes
- Iteration - one or more Range of values to be assigned, StopWhen conditions for termination, and Solve specifications
- Alternation - a test and actions to be taken when true
- Solve - the type of solution desired and the point(s) in the experiment where it should be executed
- ToolCommand - control parameters that are not included explicitly because they depend on the tool,

and (C) Output specification(s) - the variables to be written, the results (e.g., throughput or utilization), and possibly tool specific output.

The experiment schema definition is included in the host schema, e.g., PMIF. An OutputFormat schema is also needed in the host schema to specify the XML format to be used for output from the experiments. Thus, the Ex-SE allows specification of:

- Changes in parameter values from one execution of a model to the next
- Specification of control in performing model studies, including iteration and alternation
- Variables that are local to the experiment to be used in computations and output
- Model-results dependent execution
- Use of previous output as input to subsequent runs
- Specification of the output metrics to be returned
- Solution type specifications.

The schema specifies the syntactic characteristics of the Experiment. Additional semantic constraints and assumptions used in PMIF-Ex are provided at: www.spe-ed.com/pmif/pmif-ex_readme.htm.

4. PETRI NET EXPERIMENT AND OUTPUT SPECIFICATIONS

Petri nets provide representation and analysis capabilities in addition to performance analysis. In order to determine the capabilities that should be included in the specification, we examined the capabilities of current Petri net tools that support performance evaluation. The first section describes the tools considered. The second section describes the eD-SPN format that is integrated with the Ex-SE to create the the Petri Net experiment specification (PN-Ex), detailed in the third section.

4.1 Petri Net Tools for Performance Evaluation

Petri net theory has been widely used to implement a variety of modeling and evaluation tools; most of them can be found the Petri Net Tool Database [3]. Unfortunately, many of the tools described in the database or in literature are no longer maintained or available.

In order to specify a general interchange format for experimentation using Petri nets we have examined the following tools:

- GreatSPN 2.0 [1, 8] (GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets): a software package for the modeling, validation, and performance evaluation of distributed systems using GSPNs and their colored extension: Stochastic Well-formed Nets.
- Sharpe v1.01 [18, 12]: a tool for specifying and analyzing performance, reliability and performability models that allows for different model types. These include queueing networks, Markov and semi-Markov reward models as well stochastic Petri nets.
- TimeNet 4.0 [5, 20] (TIME Net Evaluation Tool): a tool for the modeling and analysis of stochastic Petri nets with non-exponentially distributed firing times. Version (4.0) contains a JAVA-based user interface.
- PIPE 2.5 [4, 7]: an open source, platform independent tool for creating and analyzing Petri nets including GSPNs. It is implemented entirely in Java thus providing platform independence as well as an easy-to-use graphical user interface. PIPE also offers a full suite of analysis modules to check behavioral properties, produce performance statistics, and some less common features such as Petri net comparison and classification.
- Möbius [10]: a discrete-event system modeling and analysis tool supporting multiple modeling formalisms and multiple solution techniques, including GSPNs.

Besides quantitative analysis that provides performance metrics, Petri net tools also have capabilities for structural analysis. Table 1 shows the structural analysis methods provided by the tools considered. *Structural Properties* refers to boundedness, safeness and deadlock freedom. Möbius is not in the table because it does not do structural analysis. Based on the solutions offered by the tools we have decided to include the specification of additional types of solutions in an experiment.

Table 2 shows the steady state quantitative results that the Petri net tools under study provide. Sharpe and TimeNet

allow the user to specify the output to be obtained from the quantitative analysis, the other tools have default values instead. Sharpe gives a list with the possible outputs to choose, however in TimeNet an expression needs to be defined for each output wanted using a specific syntax [20]. As default, TimeNet shows only throughput of timed transitions.

4.2 e-DSPN.xml

The e-DSPN.xml is a schema to specify Extended Deterministic and Stochastic Petri Nets. It is composed of the following elements and its attributes¹:

- Places: label², initialMarking.
- Exponential transitions: label, delay, serverType, preemptionPolicy, DTSPNpriority.
- Immediate transitions: label, priority, weight, enablingFunction.
- Deterministic transitions: label, delay, preemptionPolicy, DTSPNpriority.
- General transitions: label, delay, preemptionPolicy.
- Arcs: fromNode, id, toNode, inscription, handle.
- Inhibitor Arcs: fromNode, id, toNode, inscription, handle.
- Measures: name, expression, result. (Note that these are output measures specified by the user since if no output specified, TimeNET only shows throughputs of timed transitions)
- Definitions: defType (integer or real), name, expression. (These are specifications of labels associated with a value that can be used in the model definition)

4.3 PN-Ex Definition

Contrary to queueing networks, Petri nets have two different types of nodes (places and transitions), and each node type has its own attributes. Places may have capacity limitations or an initial marking, whereas transitions have weights or delays, and priorities. Petri net arcs may have a weight attribute which indicates the number of tokens added to or removed from a place. While Experiment Variable definitions could still be “generic” and refer to nodes and arcs as in the general Ex-SE, we prefer to refer to places, transitions or arcs for both readability and for the additional syntactic validation (against the schema) that is possible with the customized schema. Therefore, PN-Ex needs a different instance of the Variable specification allowing the reference of places, transitions and arcs. For the variable specification, the ID attribute (in the *label* element) that places and transitions have in e-DSPN will be used since it is the only one that is unique.

As specified in Ex-SE, output variables refer to results of previous executions of the model. The results obtained when using Petri Nets that can be assigned to output variables are:

¹attributes related to the drawing of the model are not shown since they are not of interest for this paper.

²labels are in fact elements with attributes name and ID. For simplicity, it is not described.

| | GreatSPN 2.0 | Sharpe v1.01 | TimeNet 4.0 | PIPE 2.5 |
|-------------------------------|--------------|--------------|-------------|----------|
| Net Comparison | No | No | No | Yes |
| Place Invariant Analysis | Yes | No | Yes | Yes |
| Transition Invariant Analysis | Yes | No | No | Yes |
| Minimal Siphons | Yes | No | Yes | Yes |
| Minimal Traps | Yes | No | Yes | Yes |
| Structural Properties Check | All | None | None | All |

Table 1: Structural analysis in Petri net tools

| | GreatSPN 2.0 | Sharpe v1.01 | TimeNet 4.0 | PIPE 2.5 | Möbius |
|--|--------------|--------------|-------------|----------|------------|
| Token probability density of places | Yes | No | Customized | Yes | Customized |
| Average number of tokens of places | Yes | Yes | Customized | Yes | Customized |
| Steady state distribution of tangible states | No | No | Customized | Yes | Customized |
| Utilization of places | Yes | Yes | Yes | Yes | Customized |
| Throughput of timed transitions | Yes | Yes | Yes | Yes | Customized |
| Throughput of immediate transitions | Yes | Yes | No | No | No |
| Sojourn time for tangible states | No | No | No | Yes | No |
| Utilization of transitions | No | Yes | No | No | Customized |

Table 2: Performance output metrics in Petri net tools

throughput for transitions, utilization and average number of tokens for places.

The PN-Ex uses most of the same solution specifications in the Ex-SE, such as assignment, iteration and alternation. However, in addition to the analytic and simulation solution types for the Solve command, the following solution types can also be requested in PN-Ex: PlaceInvariantAnalysis, TransitionInvariantAnalysis, MinimalSiphons, MinimalTraps and StructuralPropertiesCheck.

In the EX-SE output specifications, *WriteVariable* (see [17]) remains the same, but *WriteOutput* for PN-Ex includes additional structural analysis results and different quantitative results.

As a result of the tool comparison shown in the previous subsection, the possible values for Petri net performance metrics are: TokenProbabilityDensity, Throughput, AverageNumberOfTokens, Utilization, Delay and TimeUnits. On the other hand, the possible values for StructuralAnalysis results are: PlaceInvariants, TransitionInvariants, MinimalSiphons, MinimalTraps, and StructuralProperties.

5. CASE STUDY

To prove the concept an Experimenter module has been designed and implemented for PIPE [7]. In order to validate the approach and Experimenter, we selected a published experiment that provides sufficient data on the model, the output, and the experiment for reproducibility. The experiment is published in Woodside and Li 91 [19] and represents the passing of messages following the *Courier* protocol. Figure 1 shows the software tasks and the data flows between them. A message originates at the Sender Side User task, and is conveyed, following the arrows, to the Receiver Side User task. The message is processed by two tasks implementing the ISO Session and Transport layers at each end, and is conveyed between users and layer tasks by *Courier* tasks. The detailed GSPN model for one-way data transfer and its characteristics is given in [19]. The experiment published varies the transport window size, called N and the fragmentation ratio of data at the transport level, called $(q_1 : q_2)$. The results presented are the throughput rate λ from the

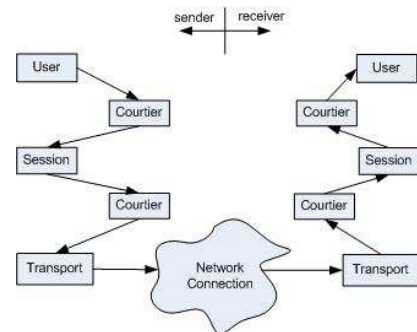


Figure 1: Tasking model of courier protocol software

users' point of view. Table 4 compares results published in [19] with results obtained with PIPE in one experiment, and for comparison, the ones obtained with TimeNET (4 different runs of the tool changing the corresponding parameters). TimeNET results for the 3rd run are not shown because it ran for more than 24 hours on three occasions and never ended.

The experiment in PIPE (1 single execution with 4 runs of the model) on a Pentium IV at 3GHz and with 512MB of DRAM ran for 20 minutes and 30 seconds. The run time for TimeNET on the same machine, is substantially longer as shown in Table 4.

It can be seen from the table that output measures have the same value for the first three runs but that surprisingly, for the fourth run ($N=2, q_1:q_2=2:1$) the output values published in [19] are noticeably different. This is the reason why we decided to also run the experiment with TimeNET. Results show that there is probably some sort of error in the published work since PIPE and TimeNET give basically the same result. It is not the first time that running experiments in an experimental framework discovers wrong results in published material (see [17]). This shows that it is easy to introduce errors in the models when different executions for each combination of model parameters have to be done. It is less likely when the experiment (different runs of the model)

can be automated. This is an additional advantage of an experimentation framework.

| Experiment | λ | Time |
|--|-------------------------------------|-------------|
| $N = 1 \ q_1 : q_2 = 1 : 1$ Woodside and Li 91 PIPE TimeNET | 74.346664 74.346677 74.346594 | 108 seconds |
| $N = 1 \ q_1 : q_2 = 2 : 1$ Woodside and Li 91 PIPE TimeNET | 50.432453 50.432461 50.432406 | 114 seconds |
| $N = 2 \ q_1 : q_2 = 1 : 1$ Woodside and Li 91 PIPE TimeNET | 120.372086 120.372102 | |
| $N = 2 \ q_1 : q_2 = 2 : 1$ Woodside and Li 91 PIPE TimeNET | 92.875839 81.052649 81.041052 | 15 hours |

Table 3: Validation results of Woodside and Li 91 experiment

6. CONCLUSIONS

The Experiment Schema Extension (Ex-SE) is an XML interchange schema extension that defines a set of model runs and the output desired from them. It provides a means of specifying performance studies that is independent of a given tool paradigm. In this paper we have presented a specific, extended instantiation of the Ex-SE for Petri nets (PN-Ex). Then, we have demonstrated the viability of the approach with a case study run with the Experimenter for PIPE2, a Petri net tool that provides structural and quantitative analysis modules. It is shown that PIPE2 with the Experimental framework is much more efficient than TimeNET, another Petri net modelling tool.

Further work includes adding transient analysis and its specific results to the PN-EX since some tools, for example Sharpe, TimeNET and PIPE through the dnamaca analyser, provide this type of analysis as well. Moreover, we also have in mind the creation of the analysis specifications that will transform the Output from the experiment into the desired results, and to address a general purpose Experimenter tool.

7. REFERENCES

- [1] GreatSPN 2.0. GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets. <http://www.di.unito.it/~greatspn/>.
- [2] Petri Net Markup Language. <http://www.informatik.hu-berlin.de/top/pnml/about.html>.
- [3] Petri Nets Tool Database. <http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html>.
- [4] Platform Independent Petri net Editor 2. <http://pipe2.sourceforge.net/>.
- [5] TimeNET 4.0. TIMEd Net Evaluation Tool. <http://pdv.cs.tu-berlin.de/~timenet/>.
- [6] J. Billington et al. The Petri Net Markup Language: Concepts, Technology, and Tools. http://www2.informatik.hu-berlin.de/top/pnml/download/about/PNML_CTT.pdf, March 2003.
- [7] P. Bonet, C. Llado, R. Puijaner, and W. Knottenbelt. Pipe v2.5: A petri net tool for performance modelling. In *Proc. 23rd Latin American Conference on Informatics (CLEI 2007)*, October 2007.
- [8] G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaud. GreatSPN 1.7: GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets. *Performance Evaluation, special issue on Performance Modeling Tools*, 24(1-2):47–68, November 1995.
- [9] T. Courtney, S. Gaonkar, G. McQuinn, E. Rozier, W. Sanders, and P. Webster. Design of experiments within the möbius modeling environment. In *Proc. of the Fourth International Conference on the Quantitative Evaluation of Systems*, pages 161–162, September 2007.
- [10] D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. Doyle, W. Sanders, and P. Webster. The möbius framework and its implementation. *IEEE Trans. on Software Engineering*, 28(10), October 2002.
- [11] J. Desel and W. Reisig. *Lectures on Petri Nets I: Basic Models*, chapter Place/Transition Petri Nets, pages 122–173. Springer, 1998.
- [12] C. Hirel, R. Sahner, X. Zang, and K. Trivedi. Reliability and performability modeling using sharpe 2000. In B. R. Haverkort, H. C. Bohnenkomp, and C. U. Smith, editors, *Proceedings 11th Int. Conf. on Computer Performance Evaluation Modelling Techniques and Tools, LNCS 1786*, pages 345–349, Berlin, March 2000. Springer Verlag.
- [13] ISO/IEC/JTC1/SC7. Subdivision of project 7.19 for a Petri net standard, October 1995.
- [14] E. Kindler, editor. *Petri Net Markup Language Forum 2006 (PNML 06)*, June 2006.
- [15] M. Melià, C. Llado, C. Smith, and R. Puigjaner. An experimental framework for PIPEv2.5. In *Submitted for publication as a tool paper to the 5th International Conference on Quantitative Evaluation of Systems*.
- [16] C. Smith and C. Llado. Performance model interchange format (PMIF 2.0): XML definition and implementation. technical report. Technical report, Performance Engineering Services, 2007.
- [17] C. Smith, C. Llado, R. Puigjaner, and L. Williams. Interchange formats for performance models: Experimentation and output. In *Proc. of the Fourth International Conference on the Quantitative Evaluation of Systems*, pages 91–100, September 2007.
- [18] K. Trivedi and C. Hirel. SHARPE: Symbolic Hierarchical Automated Reliability and Performance Evaluator. <http://www.ee.duke.edu/~chirel/IRISA/sharpeGui.html>.
- [19] C. Woodside and Y. Li. Performance petri net analysis of communications protocol software by delay-equivalent aggregation. *Petri Nets and Performance Models, 1991. Proceedings of the Fourth International Workshop on*, pages 64–73, 1991.
- [20] A. Zimmermann and M. Knoke. TimeNET 4.0. a Software Tool for the Performability Evaluation with Stochastic and Coloured Petri Nets. User Manual. Technical report, Technische Universität Berlin. Real-Time Systems and Robotics Group, August 2007.