Automatic Generation of Performance Analysis Results: Requirements and Demonstration

Connie U. Smith[1], Catalina M. Lladó[2], and Ramon Puigjaner[2]

[1] Performance Engineering Services, PO Box 2640, Santa Fe, New Mexico, 87504-2640 USA, `www.spe-ed.com`

[2] Universitat de les Illes Balears. Departament de Ciències Matemàtiques i Informàtica. Ctra de Valldemossa, Km. 7.6, 07071 Palma de Mallorca, Spain
`cllado@uib.es, putxi@uib.es`

Abstract. This paper presents a performance model interoperability framework that brings together performance model interchange formats and experiment specifications with the automatic generation of performance analysis results for presentation and publication. We define the Use Cases and requirements and survey output and results used in practice. We present the output specification, the issues in the output-to-results transformation, the results specification schema extension, and a prototype implementation. A proof of concept example demonstrates the framework.

Key words: Performance modelling, Tool interoperability, Queueing networks, Results

1 Introduction

The concept of performance model interoperability was first introduced in 1995 [6]. Methods and tools supporting model interchange formats have evolved rapidly since 2004 with the introduction of XML as a viable mechanism for supporting model interchange [4].

Performance model interchange formats (PMIF) provide a mechanism for automatically moving performance models among modeling tools. Use of the PMIF does not require tools to know about the capabilities of other tools, internal data formats, or even existence. It requires only that the importing and exporting tools either support the PMIF or provide an interface that reads/writes model specifications from/to a file. Interchange formats have also been defined for layered queueing networks (LQN), UML, Petri Nets and other types of models.

Another interchange format, the Experiment Schema Extension (Ex-SE), allows the user to define a set of model runs that vary parameters. The Ex-SE provides a means of specifying performance studies and the output desired from them that is independent of a given tool paradigm. Each tool generates the performance metric output, such as response time, utilization, etc., specified for each experiment. A performance analyst typically studies this output to form conclusions about the results of the experiments, then prepares a presentation and/or report to explain the results.

Other work (see [5] for its description) has recognized the need for this last step. Our work develops the concept and provides a concrete realization of it. This paper streamlines the last step by defining a Results Schema Extension (Results-SE) that enables a user-customized transformation from the output of an experiment into the desired results.

The contributions of this work are:

– A definition of the most frequent Use Cases for this model interoperability framework
– A review of the typical types of output and results produced for queueing network based performance models for these Use Cases
– Definition of a modeling-paradigm independent schema for specifying the output of experiments and the transformation to results
– Implementation of a prototype demonstrating the feasibility of the approach
– Demonstration that the approach works.

2 Requirements for Producing Results

First we discuss the typical situations, or *Use Cases,* for conducting modeling experiments and analyzing results. Next we identify typical output and results that are needed for those Use Cases. Then we present our approach to providing the output and results.

QNM may be used in a variety of fields from computer performance evaluation to any other field that is interested in the behavior of queues and servers. This paper addresses computer performance evaluation; other applications of QNM may require extensions to the analysis and results.

The most common reasons that performance analysts build and analyze QNM models are to:

1. Monitor and report on operational system performance
2. Analyze capacity requirements for future workload volumes
3. Evaluate problematic systems, identify causes and study options
4. Compare model results to measurements
5. Conduct technical investigations to compare results from: multiple tools, different solution algorithms, or even different types of solutions.

The next step is to determine the output metrics and results that are most often desired for these Use Cases.

We examined a sample of papers from the Computer Measurement Group 25th anniversary edition of the proceedings (1974 through 1999) [2] - the main source of practitioner modeling papers. Research results in other publications are similar.

We found three types of results: *tables*, graphs or *charts* in spreadsheet tools, and metric values embedded in the text of the paper. Some combinations of performance metrics occur frequently; examples are: service times and response times for several workloads; and throughput, response time and CPU utilization for several workloads.

Our conclusion is that the primary results are tables and charts. Charts are derived from tables, so they can be combined into one "result." Since there are many common combinations of both tables and charts, the specifications for those should be streamlined.

The most common format for tables and charts is xls [1] as in spreadsheet tools such as Excel and OpenOffice, and imported by most presentation and word processing packages. However, the most common document preparation system for research publications is LaTex. Our approach transforms the output metrics to tables and charts in xls and LaTex.

Additionally, we support two transformation modes: create a new table/chart and update an existing one. The update mode is convenient because it is unlikely that final results will be produced with one pass. It is also convenient when tables involve output from multiple tools. More importantly, it is easier to define table and chart formats by typing column and row headings or chart specifications directly into the spreadsheet rather than specifying transformation commands to create them.

This work does not address the metric values that are embedded in text. They have no tedious formatting requirements, and they might be best suited to the performance tree question/answer approach [7].

3 Model Transformation Approach

This section covers our approach for transforming the output of the performance model solutions into the desired results. The first section addresses the output. The next section explains the transformation by first describing the key issues and decisions, the approach for simplifying the generation of standard results, and some implementation issues.

The Output Schema Extension is in Fig. 1. The "ValueUsed" applies to *Ranges* or other variables used in the experiment specification and reports the value used for that particular solution.

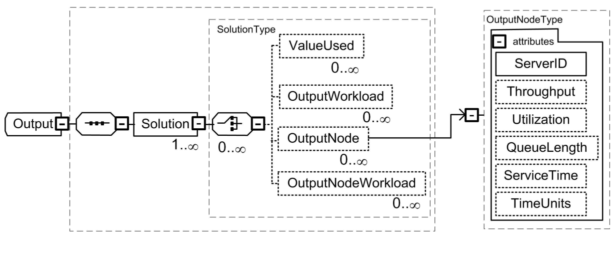


Fig. 1. Output Schema Extension

The metrics that may be produced are in the OutputWorkload (overall results by workload), OutputNode (overall results by Node), and OutputNode-Workload (results by Workload for Nodes). For more information see www.spe-ed.com/pmif/.

The output desired is specified in the Experiment specification. For each solution, the user may specify: WriteVariable, WriteOutput, or a ToolCommand that is passed to the tool unchanged. This allows users to print custom reports, visualization output, etc. particular to the tool, see [5].

The next step is to provide for an automatic conversion of the output into the table and chart results. We considered 2 options related to how those tables and charts would be expressed:

1. To use a "standard" xsd schema for spreadsheets for the results specification and transform the output into the xml format that follows such a schema.
2. To develop a transformation specification from output into the standard elements of a spreadsheet: rows, columns, and charts and transform the output into xls or LaTex format.

Some spreadsheet tools, such as OpenOffice, do not yet support xml import and export, and the "standard" schema does not include chart specifications. Option 1 would require an additional schema to specify the transformation. Thus we chose the second option because it provides a specification of tables and charts using familiar notation, e.g., numeric rows and alphabetic columns. Java tools support the creation of a spreadsheet in xls format.

Fig. 2 shows the Results-SE schema. The Output-SE has a collection of outputs for each OutputSolutionSpec (or Solve) in the Experiment-SE. So the Results-SE specifies how to process each of those, and specifies the file/s containing the output. It can specify one or more tables (in xls tables go into different worksheets). WriteResult specifies the type of output metric to use (Node, Workload, etc.), the metric (such as response time), and where to place the values in the table (row, column, etc). There is a placeholder for Chart specifications to be added in future work.

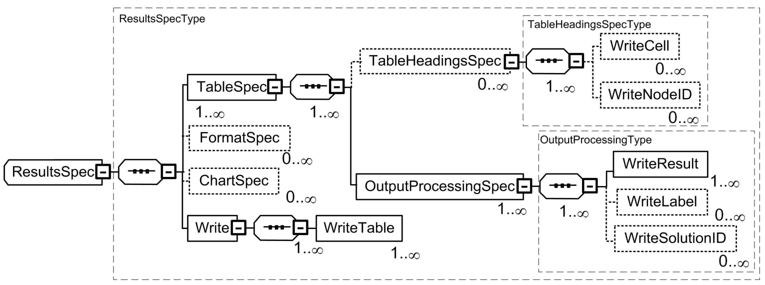


Fig. 2. Results-SE schema

The Transform prototype has been implemented in Java, using the Document Object Model (DOM) to read and validate the xml files (Output and ResultsSpec). We have also used the Apache POI APIs for manipulating MS Excel and OpenOffice file formats using pure Java.

4 Proof of Concept

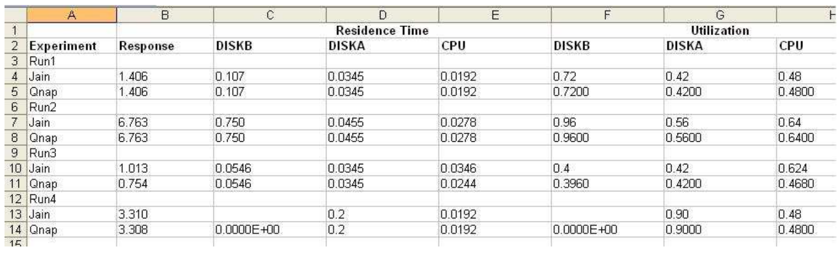
The proof of concept uses a case study previously published and replicates it with the model interoperabilty experimental framework. It is a technical paper (Use Case) that compares a published solution to solutions derived automatically from experiment specifications.

The example was published in Jain's book [3] and subsequently used as an example of the experimental framework in [5]. It shows how to manually create a table, specify formats, enter the results from another source, then update the remaining values with the output from the experiment. The demonstration seeks to replicate the table in [5].

We run Qnap to produce the Output file of performance metrics specified in the experiment in [5]. We manually create an xls file with the formatting and Jain results taken from [3]. We then update the file using the results specification (an excerpt is in Fig. 4) to produce the table in Fig. 4.

```
<OutputProcessingSpec RowIncrement="3"FileToProcess="Jain574.xml">
  <WriteSolutionID Format="5" Row="3" Col="1" />
  <WriteLabel Value="Qnap" Row="5" Col="1" />
  <WriteResult Type="OutputWorkload" Metric="ResponseTime" Row="5" Col="2"/>
  <WriteResult Type="OutputNodeWorkload" Metric="ResidenceTime" Row="5" Col="3"
               ColIncrement="1"/>
  <WriteResult Type="OutputNode" Metric="Utilization" Row="5" Col="6" ColIncrement="1"/>
<OutputProcessingSpec>
```

Fig. 3. Excerpt of Results Specifications



	A	B	C	D	E	F	G	H
1				Residence Time			Utilization	
2	**Experiment**	**Response**	**DISKB**	**DISKA**	**CPU**	**DISKB**	**DISKA**	**CPU**
3	Run1							
4	Jain	1.406	0.107	0.0345	0.0192	0.72	0.42	0.48
5	Qnap	1.406	0.107	0.0345	0.0192	0.7200	0.4200	0.4800
6	Run2							
7	Jain	6.763	0.750	0.0455	0.0278	0.96	0.56	0.64
8	Qnap	6.763	0.750	0.0455	0.0278	0.9600	0.5600	0.6400
9	Run3							
10	Jain	1.013	0.0546	0.0345	0.0346	0.4	0.42	0.624
11	Qnap	0.754	0.0546	0.0345	0.0244	0.3960	0.4200	0.4680
12	Run4							
13	Jain	3.310		0.2	0.0192		0.90	0.48
14	Qnap	3.308	0.0000E+00	0.2	0.0192	0.0000E+00	0.9000	0.4800
15								

Fig. 4. Xls file automatically produced for Jain's case study

5 Conclusions

This paper has tied together previous work on performance model interchange formats and experiment specifications. It adds an output-to-results transformation to produce performance analysis results for presentation and publication.

Our general purpose approach was demonstrated with PMIF, however it also applies to other modeling paradigms, tools, and even measurement tools. It supports the automation of model studies from the creation of the performance model specification, the experiments to be conducted with the model, the execution of models and transformation of output to tables and charts for presentation and publication. It supports the Use Cases in Section 2.1 (i.e., analyzing capacity requirements, evaluating problematic systems, etc.) and streamlines typical tasks such as exploring output and identifying results for presentation. It is a standard format that can be used by multiple tools.

Future work will develop additional templates for the most frequent results and implement additional prototypes for updating tables and creating charts. We will apply the framework to other tools, and extend it to apply to real time systems. An interesting extension might include creating rules for specifying threshold values and highlighting results in tables that exceed the threshold. We also envision the integration of Performance Trees by relating the queries to the output in order to produce results.

Acknowledgments. This work is partially funded by the TIN2006-02265 QUASIMODO project of the *Ministerio de Educacion y Ciencia*, Spain. Smith's participation is sponsored by US Air Force Contract FA8750-C-09-0086.

References

1. Microsoft office binary (doc, xls, ppt) file formats. `www.microsoft.com/interop/docs/OfficeBinaryFormats.mspx`.
2. CMG. Computer Measurement Group. `www.cmg.org`.
3. R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling.* John Wiley, 1991.
4. C.U. Smith and C.M. Lladó. Performance model interchange format (PMIF 2.0): XML definition and implementation. In *Proc. of the First International Conference on the Quantitative Evaluation of Systems*, pages 38–47, September 2004.
5. C.U. Smith, C.M. Lladó, R. Puigjaner, and L.G. 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.
6. C.U. Smith and L.G. Williams. Panel presentation: A performance model interchange format. In *Proc. of the International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, 1995.
7. T. Suto, J. T. Bradley, and W. J. Knottenbelt. Performance trees: A new approach to quantitative performance specification. In *Proc. 14th Intl. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS06).* IEEE Computer Society, September 2006.