FORGE: Friendly Output to Results Generator Engine

J. Llodrà, C. M. Lladó, R. Puigjaner
Computing and Maths Department
Universitat de les Illes Balears
07071 Palma de Mallorca, Spain

jlg.hrtc@gmail.com,cllado/putxi@uib.cat

Connie U. Smith
Performance Engineering Services
PO Box 2640
Santa Fe, NM 87504-2640 USA

www.spe-ed.com

ABSTRACT

Performance model interchange formats provide a mechanism for automatically moving performance models among modelling tools. The Experiment Schema Extension (Ex-SE) specifies performance studies to be run on the model and the XML performance metrics desired. Moreover, the Results Schema Extension (Results-SE) specifies how to automatically transform the output metrics into useful results. This paper presents the tool FORGE (Friendly Output to Results Generator Engine), a GUI based transformation tool from output to results which simplifies the specification of common tables for the output files the user provides. A case study demonstrates the use of the tool.

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 Model Interchange Formats (MIFs) that specify a model and a set of parameters for one run have been defined for several performance modelling formalisms. Figure 1 shows our vision of the model interoperability framework for creating and evaluating performance models using these MIFs. The Experiment Schema Extension (Ex-SE), shown in the top-right part of the figure, defines a set of model runs and the output desired from them [3]. It has the expressive power to specify iterations, alternations, assignments of values, actions based on model results and more. The schema extension provides a means of specifying performance studies that is independent of a given tool paradigm. Additionally, the Output Schema Extension (Output-SE) [2] defines the XML output file format to be produced by the modelling tools and the Results Schema Extension (Results-SE) [2] specifies how to automatically transform the output metrics into useful results (which metrics, how those are combined when they come from different runs, etc.). These results can be generated as xls files or LaTex tables. The ResultsGen tool is a command line based tool that has as input

parameters the name of the input and output files (Results-SE instance and xls or LaTex table files respectively) and the type of output (xls, LaTex or both).

This paper presents the tool Friendly Output to Results Generator Engine (FORGE), a GUI-based transformation tool that simplifies the creation of common tables for the output files the user provides. A case study demonstrates the use of the tool. Figure 1 shows the overall view of the framework and the portion where FORGE may streamline the process. In addition, we have developed the tool EXOSS, (EXperiment Output to SpreadSheet), also shown in the figure. Differently to FORGE, EXOSS takes the XML output from one or more experiments and produces a spreadsheet (xls) file for easily viewing the performance output (it does not transform the output, it is only an automatic way of getting one or more output xml files in xls format).

This first version of FORGE works with output from queueing network modeling (QNM) tools, that follows the Output-SE. It is, however, designed to work with other types of output(e.g., Petri nets output). Those will be handled in future versions.
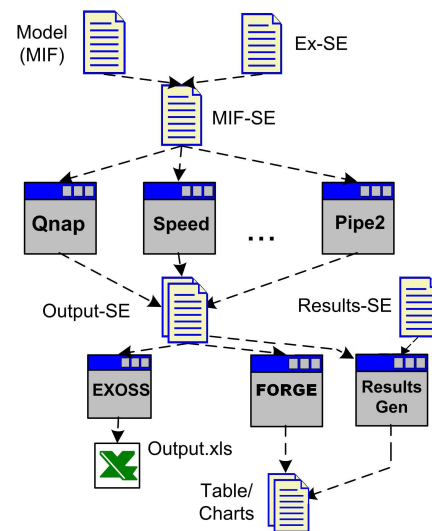

Figure 1: FORGE in the interoperability framework

In [2] we examined a sample of papers from the 25th anniversary compilation edition of the CMG proceedings and we found three general types of results: *tables*, graphs or *charts* as they are called in spreadsheet tools, and metric values that are embedded in the text of the paper. Note

that our focus is on performance results and not other uses of QNM. Some examples of common tables are:

- Response times for n nodes

- Response times for n workloads from m different workload intensities

- Node Utilization and Wait time for multiple workloads

- Utilization for several workloads

- Combinations of the above

All of these can be generated from output files with FORGE using input that is easily selected by the user. Future work will deal with charts created from these tables.

2. FORGE DEVELOPMENT

FORGE's user interface, shown in Figure 2, is divided into 4 parts: (1) File selection, (2) Results selection, (3) Table type selection, and (4) Generate button. The user can include in the project as many output files as needed using the file selection. The file names and an overview of what they contain (for example the workloads it refers to) is shown in part (1). FORGE reads the output file/s and depending on what they contain gives a number of options for metrics to be displayed in the table as well as some formatting options shown in (2). The user selections can be saved and re-used later if needed. Finally, (3) allows for the generation of the table file, being possible to create xls, LaTex or both. Once the user has chosen among the various options, the Generate button (4) causes FORGE to generate the table/s. In order to generate the user demanded table, FORGE generates first an Results-SE XML file following an explicit specification which is an update w.r.t the explicit specification described in [2]. An extended version of [2] that will include this update will be soon published in the Computer Journal. FORGE is a pure Java application, that uses the SWING toolkit to build a friendly and cross-platform user interface. It also relies on POI-HSSF [1], the POI Project's pure Java implementation of the Excel '97(-2007) file format. In addition, it uses standard Java built-in libraries for parsing and validating XML files (SAX, DOM and XPath). FORGE is developed in three modules, one reads and interprets the output files, the second one is the GUI itself and the later generates the Results-SE instance and invokes the embedded ResultsGen core engine to produce the xls and/or LaTex results file(s). It is a user friendly tool. The Results-SE xml file and xls file generation is very quick (1 s). Moreover, if the user has Excel installed, the xls is automatically open, to make the work easier and quicker for the user. FORGE and some Output examples can be downloaded from `dmi.uib.es/~cllado/mifs/`.

3. CASE STUDY

The case study from [3] demonstrates FORGE capabilities. It is an actual, complex model that compares output from *SPE·ED* [4] and Qnap tools. We aim to produce the results shown by Table 3 from [3] automatically using FORGE and two Output.xml files (one from *SPE·ED* and one from Qnap). Figure 2 shows the FORGE Gui when it has already read the output and displays the options the user has to create the results table. It also shows what options the
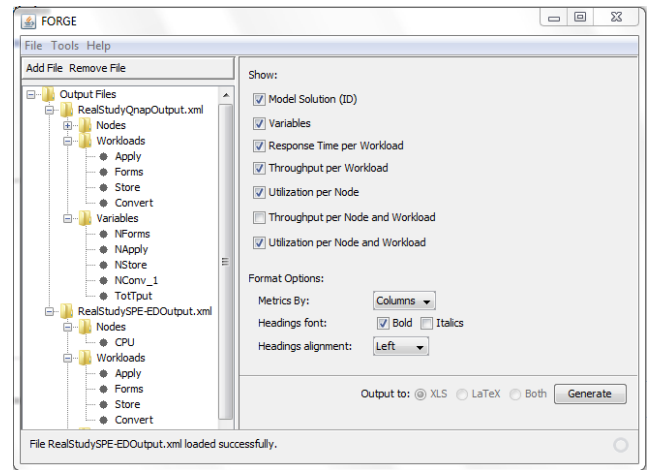

Figure 2: FORGE Gui

user would choose to generate the table we are aiming for. Table 1 is part of the FORGE generated one (missing some rows and columns due to lack of space). Some minor differences can be detected though in essence is the same table.

Table 1: Validation results for RealStudy

		Response Time		Throughput	
Experiment		*SPE·ED*	Qnap	*SPE·ED*	Qnap
Run1					
Forms	18	451	446	0.005	0.005
Apply	32	194	177	0.014	0.016
Store	50	66	21	0.019	0.017
Convert	30	20	21	0.012	0.010
Run1					
Forms	27	1125	1034	0.006	0.006
Apply	48	226	243	0.021	0.023
Store	75	236	40	0.027	0.026
Convert	45	69	61	0.017	0.015

4. CONCLUSIONS

We have presented FORGE, a GUI-based transformation tool that easily creates common tables from PMIF output.xml files. Future versions of the tool will handle other types of output (e.g., Petri nets), produce charts from tables, use formulae, highlight outputs of interest (e.g., "utilization higher than 95%"), and other capabilities.

5. REFERENCES

[1] Apache POI. Java api to access microsoft format files. `poi.apache.org/`.

[2] C. U. Smith, C. M. Lladó, and R. Puigjaner. Automatic generation of performance analysis results: Requirements and demonstration. In Jeremy T. Bradley, editor, *6th European Performance Engineering Workshop*, volume 5652 of *LNCS*, pages 73–78. Springer, 2009.

[3] 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.

[4] SPE-ED. LS Computer Technology Inc. Performance Engineering Services Division. `www.spe-ed.com`.