Chapter 3
Towards a Multi-Formalism Multi-Solution Framework for Model-Driven Performance Engineering

Catalina M. Lladó
Universitat de les Illes Balears, Spain

Pere Bonet
Universitat de les Illes Balears, Spain

Connie U. Smith
Performance Engineering Services, USA

ABSTRACT

Model-Driven Performance Engineering (MDPE) uses performance model interchange formats among multiple formalisms and tools to automate performance analysis. Model-to-Model (M2M) transformations convert system specifications into performance specifications and performance specifications to multiple performance model formalisms. Since a single tool is not good for everything, tools for different formalisms provide multiple solutions for evaluation and comparison. This chapter demonstrates transformations from the Performance Model Interchange Format (PMIF) into multiple formalisms: Queueing Network models solved with Java Modeling Tools (JMT), QNAP, and SPE ED, and Petri Nets solved with PIPE2.

INTRODUCTION

Our approach to Model-Driven Performance Engineering (MDPE) is based on model interchange formats for creating and evaluating performance models. The goal is to transfer design specifications into a performance model that may be automatically solved by a variety of modeling tools. The overall approach begins with a formal specification of a system and/or software design and transforms it into a *Model (MIF)* as shown in the upper left of Figure 1. The *Model* may be created by translating design models into performance models (Balsamo & Marzolla, 2005) (Smith, Cortellessa, Di Marco, Lladó, & Williams, 2005), or created with one performance modeling tool

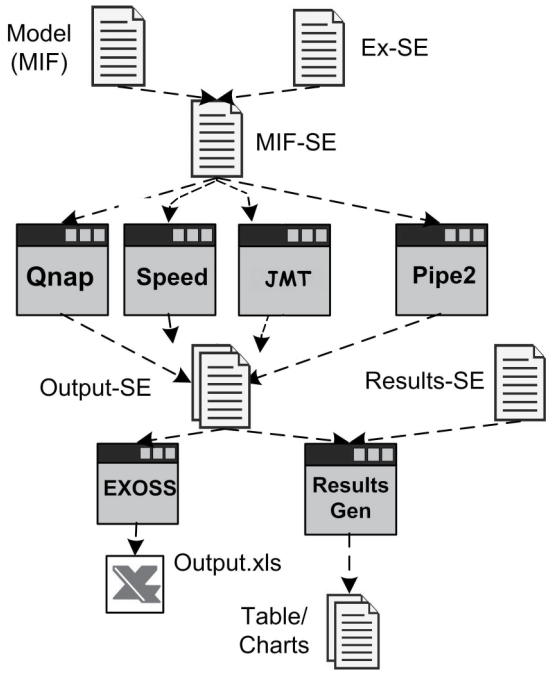
DOI: 10.4018/978-1-4666-4659-9.ch003

then converted into the MIF format. MIF formats have been defined for queueing network models (PMIF), software performance models (S-PMIF), layered queueing networks (LQN), UML, PNs and other types of models.

The upper right section of Figure 1 shows the Experiment Schema Extension (*Ex-SE)* that defines a set of model runs that vary parameters and the *output metrics* desired (Smith, Lladó & Puigjaner, 2011). It may be used to study how performance metrics change for different workload mixes, increasing service demands, etc. The *Model* and *Ex-SE* files together are used as input for one or more performance modeling tools; Qnap (Potier, D., & Veran, M., 1985), SPE-ED (www.spe-ed.com), JMT (Java Modeling Tools, jmt.sourceforge.net.), and PIPE2 (Bonet, P., Lladó, C.M., Puigjaner, R. & Knottenbelt, W.J., 2007) are illustrated but many such tools can be used.

Each tool generates the performance metric output as specified for each experiment and an Output Schema Extension (Output-SE) specifies

Figure 1. Model interoperatibility framework



the XML format for these *output metrics*. The tool EXperiment Output to SpreadSheet (EXOSS) has been also developed *EXOSS* (Llodrà, J., Lladó, C.M., Puigjaner,R. & Smith.C.U., 2011), which takes the XML output from one or more experiments and produces a spreadsheet (xls) file for easily viewing the performance output. The last step transforms the output into desired results as specified by the Results-SE (Smith, Lladó & Puigjaner, 2011).

Note that Figure 1 illustrates abstract classes for the interchange formats that are instantiated into concrete realization(s). For example, the *Model* abstract class (or MIF) may be realized with the Performance Model Interchange Format (PMIF) (Smith & Williams, 1999) (Smith & Lladó, 2004). The performance model interchange format (PMIF) is a common representation for system performance model data that can be used to move models among modeling tools that use a Queueing Network Model (QNM) paradigm. A user of several tools that support the format can create a model in one tool, and later move the model to other tools for further work without the need to laboriously translate from one tool's model representation to the other. Without a MIF two tools would need to develop a custom import and export mechanism. A third tool would require a custom interface between each of those tools resulting in an order N^2 requirement for customized interfaces. With PMIF, tools export and import with the same format so the requirement for customized interfaces is reduced to $2{\cdot}N$. The remainder of this paper assumes a familiarity with the MIF approach; see (Smith, Lladó & Puigjaner, 2010) for background information

The MIF could also be realized with the Software Performance Model Interchange Format (S-PMIF) (Smith, Cortellessa, Di Marco, Lladó, & Williams, 2005) for exchanging software performance model information. Similarly, the *Ex-SE* may have an instance for PMIF called PMIF-Ex (in other words, QN oriented), or for Petri Nets (PN) called PN-Ex (Smith, Lladó & Puigjaner, 2011).