Stochastic Modeling and Analysis Using QPME: Queueing Petri Net Modeling Environment v2.0

Simon Spinner, Samuel Kounev, and Philipp Meier

Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany
{simon.spinner,kounev}@kit.edu, mail@philippmeier.com

Abstract. Queueing Petri nets are a powerful formalism that can be exploited for modeling distributed systems and analyzing their performance and scalability. By combining the modeling power and expressiveness of queueing networks and stochastic Petri nets, queueing Petri nets provide a number of advantages. In this paper, we present our tool QPME (Queueing Petri net Modeling Environment) for modeling and analysis using queueing Petri nets. QPME provides an Eclipse-based editor for building queueing Petri net models and a powerful simulation engine for analyzing these models. The development of the tool started in 2003 and since then the tool has been distributed to more than 120 organizations worldwide.

Keywords: Queueing Petri nets, stochastic modeling and analysis, simulation.

1 Introduction

Introduced in 1993 by Falko Bause [1], the Queueing Petri Net (QPN) formalism combines the modeling power and expressiveness of queueing networks and stochastic Petri nets. QPNs are commonly used for the performance evaluation of computer systems because they provide a number of benefits compared to traditional queueing networks and stochastic Petri nets. QPNs enable the integration of hardware and software aspects of system behavior in the same model. In addition to hardware contention and scheduling strategies, QPNs make it easy to model simultaneous resource possession, synchronization, asynchronous processing and software contention. Another advantage of QPNs is that they can be used to combine qualitative and quantitative system analysis. A number of efficient techniques from Petri net theory can be exploited to verify some important qualitative properties of QPNs. The latter not only help to gain insight into the behavior of the system, but are also essential preconditions for a successful quantitative analysis [4]. The main idea behind the QPN formalism was to add queueing and timing aspects to the places of Colored Generalized Stochastic Petri Nets (CGSPNs). This is done by allowing queues (service stations) to be integrated into places of CGSPNs. A place of a CGSPN that has an integrated queue is called a queueing place and consists of two components, the queue and a depository for tokens which have completed their service at the queue. For a detailed description of the QPN formalism see [1].
The major goal of QPME (Queueing Petri net Modeling Environment) is to support the modeling and analysis of QPN models. The presented tool provides user-friendly graphical editors enabling the user to quickly and easily construct QPN models. It offers a highly optimized simulation engine that can be used to analyze QPN models efficiently. The simulation engine enables the analysis of QPN models too large to be analyzable with analytical techniques due to the state space explosion problem \[8\]. QPME also offers advanced features for processing and visualizing the results of simulating a QPN model. In addition, being implemented in Java, QPME runs on all major platforms and is widely accessible. The QPN formalism can be used for stochastic modeling in many domains. One major area of application is the performance analysis of computer systems. The tool has been successfully used in several performance modeling studies, e.g. in \[10\,11\,14\,16\].

The development of QPME started in 2003 at the Technische Universität Darmstadt and has been continuously extended since then. Currently, the tool is developed and maintained by the Descartes Research Group \[3\] at Karlsruhe Institute of Technology (KIT). Since May 2011, QPME is available in version 2.0 under an open-source license (Eclipse Public License 1.0). QPME has been distributed to more than 120 universities and research organizations worldwide so far.

The rest of this paper is organized as follows: Section 2 provides an overview of the functionality provided by QPME. Section 3 gives some technical insights into its implementation. Section 4 describes typical use cases of the tool and Sect. 5 provides a comparison with other tools for QPNs. Finally, the paper is wrapped up with some concluding remarks in Sect. 7.

2 Functionality

2.1 Queueing Petri net Editor (QPE)

QPE is a graphical editor for QPNs. The user can create QPN models with a simple drag-and-drop approach. Figure 1 shows the QPE main window which is comprised of four views. The Main Editor View displays the graphical representation of the currently edited QPN. The palette contains the set of QPN elements that can be inserted in a QPN model by drag-and-drop, such as places, transitions, and connections. Furthermore, it provides editors for the central definition of colors and queues used in a QPN model. In the Properties View the user can edit the properties of the element currently selected in the QPN model. For queueing places, for instance, the user can specify a scheduling strategy and service time distributions for each color in this view. The Outline View shows a list of all elements in the QPN model. The Console View displays the output when simulating a QPN model.

In a QPN, a transition defines a set of firing modes. An incidence function specifies the behavior of the transition for each of its firing modes in terms of tokens destroyed and/or created in the places of the QPN. Figure 2 shows the

\[\text{http://www.descartes-research.net}\]