Performance Evaluation of Software Architectures with Multiclass Queueing Network Models

Simonetta Balsamo
balsamo@dsi.unive.it
http://www.dsi.unive.it/~balsamo

Moreno Marzolla
marzolla@dsi.unive.it
http://www.dsi.unive.it/~marzolla

Department of Informatics University Ca’ Foscari of Venice via Torino, 155 30172 Mestre (Venice) ITALY

Software performance based on performance models can be applied at early phases of the software development cycle to characterize the quantitative behavior of software systems. We propose an approach based on queuing networks models for performance prediction of software systems at the software architecture level, specified by UML.

Starting from annotated UML Use Case, Activity and Deployment diagrams we derive a performance models based on multichain and multiclass Queueing Networks (QN). The UML model is annotated according to the UML Profile for Schedulability, Performance and Time Specification.

1. Use Case Diagrams describe workloads—either open or closed—applied to the system. Annotations specify the type of workload and its parameters (population or interarrival time).
2. Activity Diagrams describe the computations performed by the system in term of service requests to resources (processors). Annotations specify the service demand and the resource name on which the action executes.
3. Deployment Diagrams describe the resources (processors) available in the system. Annotations specify the processing rate.

Algorithm 1: Generation of a closed QN

\[ \text{Algorithm 1: Generation of a closed QN} \]

\[ \text{Requires:} \]
\[ \text{Activity diagram } G = (A, T), \text{ Deployment diagram } R, \text{ Actor } W \text{ stereotyped as } \langle \text{PlayClosedLoad} \rangle \]
\[ \text{Let count } [a] = 0 \]
\[ \text{for all } a \in A \text{ do} \]
\[ \text{count}[a] = \text{count}[a] + 1 \]
\[ \text{index}[a] = \text{count}[a] \]
\[ \text{end for} \]
\[ \text{Number of Classes} \]
\[ K = |R| + 1 \]
\[ \text{Number of Service Centers} \]
\[ C = |C| \]
\[ \text{new vector } \]
\[ \text{for all } t = (x, y) \in T \text{ do} \]
\[ \text{Define the routing matrix } P \]
\[ i \leftarrow \text{index}[x] \]
\[ j \leftarrow \text{index}[y] \]
\[ P[i][j] = \text{index}[a] \]
\[ \text{end for} \]
\[ \text{Define Service Rates } \mu \]
\[ \mu[i] = \mu \leftarrow \text{arrivalrate}[W] \]
\[ \mu[j] = \mu \leftarrow \text{arrivalrate}[W] \]
\[ \text{end for} \]

Algorithm 2: Generation of an open QN

\[ \text{Algorithm 2: Generation of an open QN} \]

\[ \text{Requires:} \]
\[ \text{Activity diagram } G = (A, T), \text{ Deployment diagram } R, \text{ Actor } W \text{ stereotyped as } \langle \text{PlayOpenLoad} \rangle \]
\[ \text{Let count } [a] = 0 \]
\[ \text{for all } a \in A \text{ do} \]
\[ \text{count}[a] = \text{count}[a] + 1 \]
\[ \text{index}[a] = \text{count}[a] \]
\[ \text{end for} \]
\[ \text{Number of Classes} \]
\[ K = |R| \]
\[ \text{Number of Service Centers} \]
\[ C = |C| \]
\[ \text{new vector } \]
\[ \text{for all } t = (x, y) \in T \text{ do} \]
\[ \text{Define the routing matrix } P \]
\[ i \leftarrow \text{index}[x] \]
\[ j \leftarrow \text{index}[y] \]
\[ P[i][j] = \text{index}[a] \]
\[ \text{end for} \]
\[ \text{Define Service Rates } \mu \]
\[ \mu[i] = \mu \leftarrow \text{arrivalrate}[W] / \text{servicecapacity}[\text{service}[a]] \]
\[ \mu[j] = \mu \leftarrow \text{arrivalrate}[W] / \text{servicecapacity}[\text{service}[a]] \]
\[ \text{end for} \]

Conclusions and future work

We propose an algorithm for automatic translation of annotated UML specifications into multiclass QN performance models. We consider UML specifications in terms of Use Case, Activity and Deployment diagrams annotated with quantitative, performance-oriented annotations according to the UML SPT Profile. The resulting multiclass QN model can be analyzed with well-known exact or approximate solution algorithms. Performance results are reported back into the software model as tagged values associated to the corresponding UML elements. We plan to extend the approach to UML 2.0 with the Profile for QoS.

WOSP’05, Palma de Mallorca, Spain, July 11—14, 2005