

Translation from UML to SPN Model: A Performance Modeling Framework

Razib Hayat Khan and Poul E. Heegaard

Norwegian University of Science & Technology
7491, Trondheim, Norway
{rkhan, poul.heegaard}@item.ntnu.no

Abstract. This work focuses on the delineating a performance modeling framework for a communication system that proposes a translation process from high level UML notation to Stochastic Petri Net model (SPN) and solves the model for relevant performance metrics. The framework utilizes UML collaborations, activity diagrams and deployment diagrams to be used for generating performance model for a communication system. The system dynamics will be captured by UML collaboration and activity diagram as reusable specification building blocks, while deployment diagram highlights the components of the system. The collaboration and activity show how reusable building blocks in the form of collaboration can compose together the service components through input and output pin by highlighting the behavior of the components. Later a mapping between collaboration and system component identified by deployment diagram will be demonstrated. Moreover the UML models are annotated to associate performance related quality of service (QoS) information for solving the performance model for relevant performance metrics to generate performance evaluation results.

Keywords: UML, SPN, Performance attributes.

1 Proposed Performance Modeling Framework

Our proposed performance modeling framework utilizes tool suite Arctis which is integrated as plug-ins into the eclipse IDE [1]. The proposed framework is shown in Fig.1 where steps 1 and 2 are the parts of Arctis tool suite. A developer first consults a library to check if an already existing collaboration block or a collaboration of several blocks solves a certain task. Missing blocks can also be created from scratch and stored in the library for later reuse. The building blocks are expressed as UML models. The structural aspect, for example the service component and their multiplicity, is expressed by means of UML 2.2 collaborations [2]. For the detailed behavior, we use UML 2.2 activities [2]. They express the local behavior of each of the service components as well as their necessary interactions in a compact and self-contained way using explicit control flows [1].

In the second step, building blocks are combined to more comprehensive service by composition. For this composition, we use UML 2.2 collaborations and activities as well. While collaborations provide a good overview of the structural aspect of the

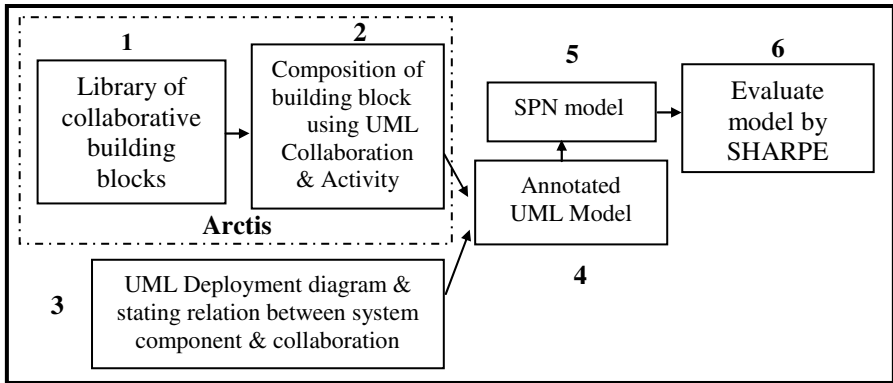


Fig. 1. Proposed performance modeling framework

composition, i.e., which sub-services are reused and how their collaboration roles are bound, activities express the detailed coupling of their respective behaviors [1]. Each sub-service is represented by a call behavior action referring to the respective activity of the building blocks. For each activity parameter node of the referred activity, a call behavior action declares a corresponding pin. By connecting the individual input and output pins of the call behavior actions, the events occurring in different collaborations can be coupled.

In the next step the UML deployment diagram of our proposed system as well as the relationship between system component and collaboration will be outlined to describe how service is delivered by the joint behavior of the system components [2].

Activity diagram and the deployment diagram are annotated in the following step according to the UML Profile for Schedulability, Performance and Time [2] to incorporate performance related quality of service information.

By considering all the above mentioned steps probable states and the parameter for triggering the change between the states of the performance model will be derived and later will be solved by the SHARPE tool [4].

2 Conclusion

However the size of the underlying reachability set is major limitation for large and complex system. Further work includes automating the whole translation process from our UML specification style to generate performance model and the way to solve the model through our proposed framework as well as tackling the state explosion problems of reachability marking for large system.

References

1. Kramer, F.A.: ARCTIS, Department of Telematics, NTNU, <http://arctis.item.ntnu.no>
2. OMG UML Superstructure, Version-2.2
3. OMG 2005, UML Profile for Schedulability, Performance, & Time Specification, V – 1.1 (2005)
4. Trivedi, K.S., Sahner, R.: SHARPE Performance evaluator. Duke University, Durham