Stochastic models for quality of service of component connectors

Moon, Y.J.

Chapter 7

Conclusions and Future work

In this chapter, we conclude this thesis with a summary of what we presented in the previous chapters and a discussion of a number of future activities to extend the work presented in this thesis.

7.1 Conclusions

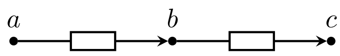
As the Internet has advanced in terms of accessibility, usability, and utility, the interest in using distributed services over networks for large-scale applications has increased. However, the composition of distributed applications is non-trivial because of their heterogeneity. When it comes to their quantitative aspects, it is challenging to specify and reason about the end-to-end QoS of composed applications.

In this thesis, we provided a specification model, Stochastic Reo, to describe coordination in such composition while considering non-functional (QoS) aspects. As an operational semantic model for Stochastic Reo, Quantitative Intentional Automata (QIA) were introduced. This semantic model describes the data-flows through connectors and the interaction with the environment of the connectors separately, thus, it is appropriate to specify and reason about the end-to-end QoS in a composed application. However, in general, QIA have a large number of states. In order to overcome this, we introduced Stochastic Reo Automata as an alternative semantic model to QIA. Stochastic Reo Automata are not only more compact, but they also enabled us to prove a compositionality result easily, which was lacking for QIA. Both semantic models serve as intermediate models for generating corresponding CTMCs for stochastic analysis. In order to consider more general QoS aspects, we have extended Stochastic Reo Automata with reward information, and this extension is also propagated to CTMCs as state rewards using the translation method. This translation method has been implemented, in the Reo2MC tool, in the Extensible Coordination Tools (ECT) [35]. As a plug-in for ECT, Reo2MC provides the following functionalities: 1) editor for Stochastic Reo 2) generating semantic models for Stochastic Reo, in particular, QIA, 3) deriving a CTMC model for a Stochastic Reo. That is, Reo2MC can be seen as an integrated tool of a Stochastic Reo editor and a CTMC generator from Stochastic Reo. We explained the implementation details of the Reo2MC, as well as its basic definitions and algorithms, and its usage. The output of the Reo2MC can be fed to other tools, such as PRISM, MATLAB, and Maple, for stochastic analysis. As a case study, we analyzed the ASK system, which is an industrial software and acts as a mediator between service consumers and service providers. The ASK system is specified using Stochastic Reo, whereby this model is translated into corresponding CTMC for analysis using PRISM. The rates used in this model were obtained by applying statistical analysis techniques on the raw values that we obtained from the real logs of an actual running ASK system. The results of this verification allowed us to draw conclusions about resource allocation and how the system installation can be adapted in order to improve its performance.

7.2 Future work

Stochastic Reo does not impose any restriction on the distribution classes of its annotated rates, such as the rates for request-arrivals at channel ends or data-flows through channels. However, for the translation from Stochastic Reo into an homogeneous CTMC model, we considered only exponential distributions for the rates. For example, in the case study using the ASK system, if the rates, obtained from statistical analysis on the raw values extracted from the real logs of a running ASK system, were not exponentially distributed, then we had to assume the obtained rates as exponential distributions or used other techniques, such as bootstrapping, to get meaningful rates. Thus, in order to support the general usage of Stochastic Reo, we want to consider non-exponential distributions such as phase-type distributions or using Semi-Markov Processes [50] as target models of our translation.

In addition, during the case study, we encountered models whose state spaces were too large to analyze. However, all the states in such a model are not meaningful because some of them are caused by the structure of Reo primitive channels, not the behavior of connectors. For example, consider the following connector:



The data-flows from the first buffer to node b and from node b to the second buffer are considered as two different events and represented sequentially in its corresponding CTMC model. In general, we may not be interested in which buffer is full when one of the two buffers is occupied. In this case, it is more meaningful to make these two data-flows immediate events, to reduce the configurations of this connector to include only *empty, half-full,* and *full.* For this purpose, we want to *hide* node b in this connector. However, hiding nodes is not trivial since it can lead to the loss of the structural information of connectors, which is used to generate corresponding CTMCs from the connectors. Thus, it is an interesting future work direction to find certain patterns of hiding nodes, which still allows to generate CTMCs with correct operational semantics. Moreover, the implementation of checking these pattens for hiding will be the next step in order to provide users with safe selection of nodes for hiding. This will help to mitigate the large state space of the derived CTMCs.

Compared to CTMCs, Interactive Markov Chains (IMCs) are compositional, thus, the IMC for a complex system can be built out of IMCs corresponding sub-systems constituting the complex system. Then, one might wonder why IMCs are not used as our stochastic target model without the translation via other operational semantic models. To answer this question, we discussed why IMCs are not an appropriate semantic model for Stochastic Reo since it generates unintended transitions that are produced in synchrony propagation scenarios. In addition, we showed the translation from Stochastic Reo into IMCs via Stochastic Reo Automata. A natural and interesting future work is to consider whether it is possible to adapt the composition operator of IMCs in order to delete the unintended transitions and still remain within a compositional framework.

So far, the Reo2MC tool uses QIA (instead of Stochastic Reo Automata) as an intermediate model for its translation. We are currently extending and improving these tools to use Stochastic Reo Automata, as well as the extension with reward information, so that the more compact sizes of the automata models will then allow us to analyze larger models.

The connection considered in this thesis is described without considering the overhead of establishing the coordination between components. That is, for a Reo connector, we considered only the interaction with the environment of the connector and its internal processing, i.e., data-flows between its boundary nodes, and the coordination was assumed to be established immediately. To be more realistic, we also need to consider the overhead of establishing the coordination between the boundary nodes before the internal processing of the connector occurs. In [54], Action Constraint Automata (ACA) were proposed to specify such a coordination processing in Reo connectors in a compositional manner. However, they do not include the interaction with the I/O requests of the connectors, and moreover, ACA do not account for the QoS aspects of the connectors. Thus, it will be an interesting and meaningful future work to provide a stochastic extension of the specification for the coordination processing in ACA and to, in turn, combine this extension and our specification approach, which enables us to analyze and reason about more realistic end-to-end QoS of a Reo connector.