All electronic communication relies on communication protocols. It is therefore very important that protocols are correct and that protocol implementations are reliable. Coloured Petri Nets (CPNs) have been widely used to model, analyse and verify communication protocols. However, relatively limited work has been done on transforming CPN model to protocol implementations. The goal of the thesis is to be able to automatically generate high-quality implementations of communication protocols based on CPN models.

In this thesis, we develop a methodology for generating implementations of protocols using a sub-class of CPNs, called Pragmatics Annotated CPNs (PACPNs). PACPNs give structure to the protocol models and allows the models to be annotated with code generation pragmatics. These pragmatics are used by our code generation approach to identify and execute the appropriate code generation templates. The templates hold the information needed to transform the model to a fully working protocol implementation for a target platform. The code generation approach coupled with PA-CPNs provide a flexible way to perform code generation for communication protocols. The code generation approach has been implemented in a prototype tool called PetriCode.

We defined several criteria for our code generation approach, the approach should be scalable so that is can be used to generate code for industrial sized protocols. The models should be verifiable and it should be possible to perform efficient verification on the models. The approach and the models that are employed for code generation should be platform independent in the sense that it should be possible to generate code for a wide range of platforms based on the same model. The generated code should be integrable meaning that it should be able to use different third party libraries and the code should be easily usable by third party code. Finally, the code should be readable by developers with expertise on the considered platforms.

In this thesis, we show that our code generation approach is able to generate code for a wide range of platforms without altering the PA-CPN model that describe the protocol design. The generated code is also shown to be readable and we demonstrate that a generated implementation can be easily integrated with third party software. We also show that our approach scales to industrial sized protocols by applying our approach to generate code for the WebSocket protocol. The WebSocket protocol creates a message-based two-way channel that can be used by web applications. This allows web applications to communicate with the server much more efficiently than using the traditional request-response pattern for certain application types such as games and rich web applications. Finally, we conclude the evaluation of the criteria of our approach by using the WebSocket PA-CPN model to show that we are able to verify fairly large protocols.

General information
State: Published
Organisations: Department of Applied Mathematics and Computer Science, Software and Process Engineering, Western Norway University of Applied Sciences
Contributors: Simonsen, K. I., Kristensen, L. M., Kindler, E.
Number of pages: 296
Publication date: 2015

Publication information
Place of publication: Kgs. Lyngby
Publisher: Technical University of Denmark (DTU)
Original language: English
(DTU Compute PHD-2014; No. 345).
Electronic versions:
phd345_Simonsen_KIF.pdf
Research output: Research › Ph.D. thesis – Annual report year: 2015