Using Colored Petri Nets for System Specifications and as a System Under Test Prototype

Thomas Schnattinger and Andrej Pietschker
Giesecke & Devrient GmbH
Banknote Processing
Prinzregentenstrasse 159
andrej.pietschker@gi-de.com

Abstract

Automated testing is an essential part in the software development process. It requires, however, the system under test (SUT) to be at least partially implemented. Also the practical verification of the test cases themselves cannot start before the SUT is available. Based on the usage of Colored Petri Nets (CPNs) as a specification tool and their integration with TTCN-3, we present an approach allowing the application of test cases already before the start of the implementation phase. As an additional benefit, the well-defined semantics of Colored Petri Nets enforces completeness and consistency of the system specification. The implementation of the described approach is based on the development of the machine control software for banknote processing machines.

Introduction. The software architecture of a banknote processing machine is split into hardware controllers and a system control component which provides an HMI and the overall coordination of the hardware controllers. All components are connected through the CAN bus. Specifying the overall control is a major development step. To ensure completeness and consistency of the specification, the formal system of Petri Nets has been chosen. In parallel the TTCN-3 tests were created, specifying how the overall control will interact with the hardware controllers. These tests are designed to check the message flow between overall control and hardware components. The TTCN-3 tests use a CAN adapter and CAN message codec to achieve this. To gain more confidence in the correctness of the specification as well as the test implementation, the idea was to apply the tests against the formal, executable model.

Connecting CPN with TTCN-3. In our approach, selected banknote processing applications have been specified as CPNs using the high level Petri Net software CPN Tools. The CPN specifications can be used as executable models. Execution of the CPN is triggered by the exchange of CAN messages with the TTCN-3 test automation platform. To achieve the necessary integration of CPN Tools with TTCN-3, the CPN Tools library Comms/CPN has been used. It allows TCP/IP based message exchange. In addition, the necessary encode and decode functions had to be provided. On the TTCN-3 test side we had to exchange the codec and the adapter to work with TCP/IP and string encoded messages rather than binary messages through a CAN hardware box. Otherwise the tests were left unchanged.

Specification of the system control component as CPN Colored Petri Nets are an established tool for the modeling of distributed systems. Unlike many other formal notation systems, they have a rather intuitive notation and the CPN can be designed so that the visible places and transitions have a very close correspondence to the elements and processes of the application domain. Technical details, for instance the implementation of the message exchange with external tools, can be hidden in sub-nets using a hierarchical decomposition of the Petri Net architecture.

The behavior of the central system control component of the machine is implemented in CPN, while the behavior of single components (e. g., sensors or engine control) is tested at the level of individual CAN messages in the TTCN-3 test case. Since the CPN model provides a more abstract view on the application than individual CAN messages, a single transition in the CPN is often based on an aggregated number of CAN messages send from the system components (i.e., TTCN-3 tests) to the central system control unit (i.e., CPN Tools). This allows to use the CPN model at a much more abstract level than individual CAN messages, which is necessary for the system specification phase.

Summary. An adequate and right level of abstraction is offered by CPN to formalize the system control component. In addition, the used CPN Tools software provides built-in functionality for state space analysis, simulation statistics, or timing/performance aspects. If desired, also manual interaction with the CPN model is possible during execution of a test case. The availability of an executable SUT model already at the end of the specification phase allows a much earlier application of automated testing and a better verification of the test cases themselves. When the real system is implemented, two independent but interchangeable SUT components are available.