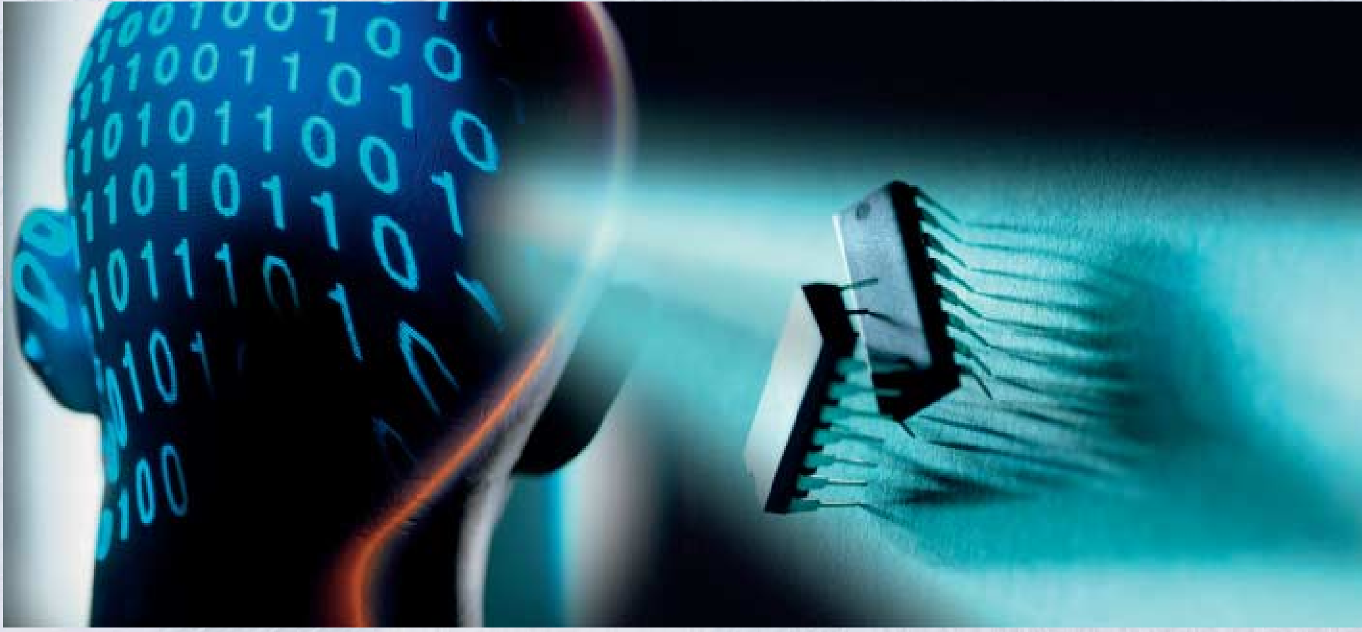


Taking concept models

from standardization to silicon

Modern communication protocols used in safety-relevant systems require a continuous verification process from the early concepts of the protocol to the finished product. This article describes how this was addressed by Philips for the FlexRay protocol using a two-pronged approach that included a SystemC model, which later became the core of a FlexRay executable model, and a Cadence Specman/e model, which provided improved coverage for product testing in comparison to directed tests.

As a Core Partner of the FlexRay Consortium, Philips has been involved in the FlexRay development and standardization right from the beginning. The challenge: taking FlexRay to success both from a technical perspective and from a business point of view.

First of all, tool support was needed for protocol development experts that enabled the proper definition and accurate specification of protocol algorithms matching the requirements. Secondly, for Philips as a semiconductor manufacturer, the fast and correct translation of the protocol specification into product implementations was of essential importance. Finally, an integrated approach based on reuse of gained experience and components from the protocol development phase had to be defined.

The complexity of the FlexRay design called for a process that included advanced verification methods and the deployment of appropriate tools.

The Process

For the process of verifying and implementing the FlexRay protocol specification several models of varying cognitive distance from the specification were realized as shown in **Figure 1**. The first model employed was a SystemC model. It was constructed by defining and applying translation rules from SDL to SystemC which achieved a reproducible and equivalent conversion. Initially the model was employed for validating the protocol specification using directed tests. But FlexRay's configuration space encompasses approximately 70 interdependent parameters. Similarly, the test space explodes due to several interacting Finite State Machines (FSM) in the specification. Thus using directed tests for ensuring that the protocol behaves as intended in every situation was much too costly and not a sustainable option.

A second model was constructed based solely on how the protocol should behave, largely ignoring the details of the SDL diagrams. So e.g. instead of modeling the FlexRay Startup procedure as several FSMs, this model expects a certain sequence of frames with correct timing before moving the node into normal operation mode. The model was written using Specman Elite/e, a solution for random-driven testing which removed the problem of crafting thousands of test-cases. Mutually verifying both models in random configurations and situations ensured their correctness, as well as that of the protocol specification itself. Philips discovered several incorrect configuration constraints as well as errors and gaps in the SDL description itself. One example was that the clock synchronization FSM sometimes overeagerly used sync frames even though those had been declared late by another FSM.

The two validated models were then put to work in different ways: The SystemC model of the protocol engine became the core part of a FlexRay executable model. The Specman/e model was again used to verify expected behavior in every situation, this time to ensure the correctness of the FlexRay controller RTL model.

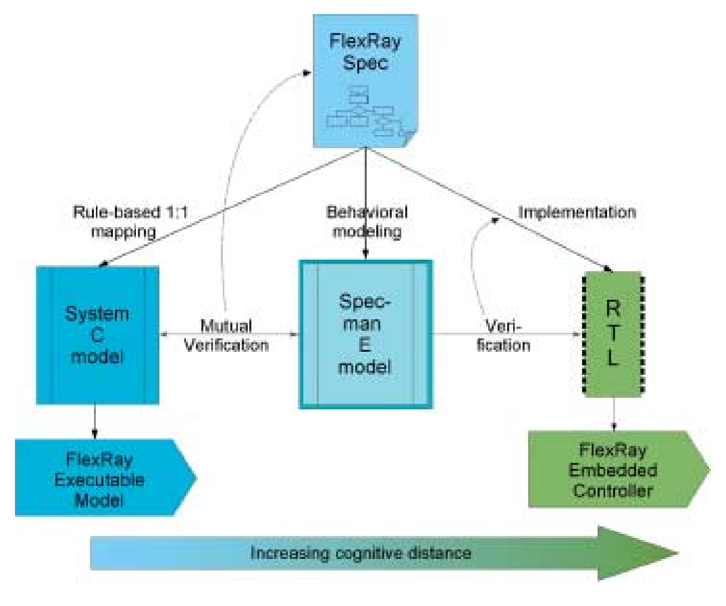


Figure 1: End-to-end verification: FlexRay specification reference and derived model representations]

© **auto**motive

Figure 2 shows the environment architecture developed to easily switch between verifying a RTL or a SystemC device under test (DUT). The Specman/e model and the respective DUT model receive the same input and their outputs are mutually compared. To cope with the different abstraction levels of the model outputs, adaptors were developed providing a common interface between the model layers. This set up a situation where the checking model becomes independent from specifics of the DUT.

The examples on the right side of figure 2 demonstrate the flexibility of this approach. Various model types can be easily connected to each other, forming a hybrid FlexRay node model. For example a verification environment for the protocol engine (PE) contains two parallel models in the PE layer -one being the DUT- as well as a fast model in the controller-host interface layer (CHI) providing the stimuli. So realistic PE stimuli scenarios are enabled at low simulation costs. Another example shown is a model of nodes which do not contain any DUT. This is crucial as FlexRay focuses on cluster-wide node interoperability. DUT-less nodes facilitate the setup of high node count environments with realistic bus activity as well as stress testing them by error injection.

Verification quality was measured in the Specman/e model using functional coverage grids. Since the model behaved in lock step to the DUT it was possible to analyze verified behavior and configurations of the DUT thoroughly even without the ability to look into DUT internals.

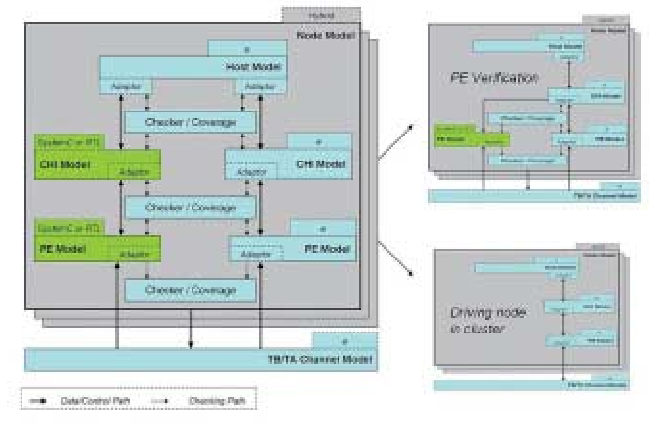


Figure 2: FlexRay node: combining different model architectures

© **auto**motive

Tooling and Methodology

The "concept to RTL" verification flow applied by Philips was enabled by:

- abstraction layer modeling and reuse
- random-directed test generation and
- metric driven verification.

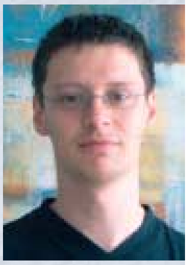
Taking concept models end-to-end from standardization to silicon requires dedicated and common code development for the various model abstractions. The Boolean-constrained aspect-oriented programming support of the e verification language enabled the explicit capturing of crosscutting concerns. These included reference behavior and the ability to selectively modifying existing attributes and methods according to the required model abstraction. The project used this technology together with the System Verification Methodology which significantly improved modularity as well as reusability of common code layers. A final challenge presented the huge state space of the FlexRay protocol specification. The stimuli and configuration generation was successfully managed with Specman's random-directed generation approach. It enabled e.g. the startup phase simulation of 1000's of FlexRay node configurations with only a hand full of test files. The vManager project-level process automation solutions handled the regression sessions as well as the process of measuring the verification completeness against a metric (verification plan). The tool automatically correlated incited functionality to verified DUT features. This was crucial to keep a high level overview on the FlexRay verification project.

Conclusion

Bridging the gap from basic specification documents to final realization in hardware requires an end-to-end verification flow. The described approach embedded to the Cadence Specman/Elite tool suite has been successfully demonstrated for the FlexRay development. The architecture delivered ease of use and mutual verification process for different models that found several unexpected errors during FlexRay protocol development phase. The approach was capable of safeguarding the implementation process and ensured correctness and fitness of Philips FlexRay products.

Dr. Markus Baumeister is working as Senior Scientist at Philips Research Labs, Aachen. He is member of the FlexRay protocol development group and the conformance test group

Dipl.-Ing. Peter Fuhrmann is leading the project cluster on dependable in-vehicle networks and architectures at Philips Research Labs, Aachen.



Dipl.-Ing. Frank Armbruster Is working as Senior Sales AE Manager for Cadence Verification Division in Central Europe, driving and executing customer engagements in functional verification projects.