COMPL\textsubscript{e}T\textsubscript{e}—
A COMmunication Protocol vaLidation Toolchain

Sven Gröning, Christopher Rosas, and Christian Wietfeld

Communication Networks Institute (CNI), TU Dortmund University,
Dortmund 44227, Germany
\{sven.groening,christopher.rosas,christian.wietfeld\}@tu-dortmund.de
www.cni.tu-dortmund.de

Abstract. Because of shorter software development cycles for communication protocol stacks, the risk of design failures rises. Therefore, even within the protocol specification phase, appropriate validation should be performed in order to detect failures as early as possible. In the light of electric vehicle integration in a smart grid environment, the complexity of charging processes increases e.g. for demand management, and thus also complexity of requirements for associated communication protocols increases. Accordingly, it lends to describe the behavior of communication protocols by abstraction in form of models. The use of model checking processes can validate properties of future behavior, hence failures may be detected earlier. COMPL\textsubscript{e}T\textsubscript{e} is a toolchain for validation of communication protocols, represented in an adapted version of UML-Statecharts. The toolchain uses the SPIN model checker and its composition is based on techniques of Model-Driven Software Development (MDSD).

Keywords: Communication Protocol Validation, COMPL\textsubscript{e}T\textsubscript{e}, SPIN, UML-Statecharts, Electric Mobility.

1 Introduction

Communication protocols in general, define the way of information exchange between devices or other entities on a network. To reach an agreement by involved parties about the way of information flow, the protocol description should be developed as a technical standard. Some standards already include a formal description, however only in rare cases. Furthermore, the description of the protocol behavior may also have a high level of complexity.

Especially in the context of electric mobility, a future widespread use of electric vehicles requires the deployment of reliable, uniform and comprehensive battery charging infrastructures. Therefore, the communication between all systems becomes an important factor for future acceptance.

By use of model checking techniques, the behavior of new communication protocol standards can be validated within the specification process. For this purpose, it is required to describe the behavior in a formal description language, which can be used by state-of-the-art model checking tools like SPIN\textsuperscript{[5]}. 

© Springer-Verlag Berlin Heidelberg 2013
COMPL\textsubscript{e}T\textsubscript{e} combines the possibility of an abstract behavior description represented as Unified Modeling Language (UML)-Statechart models, with a formal representation in PROMELA, which is used by SPIN as input language. Accordingly, COMPL\textsubscript{e}T\textsubscript{e} facilitates the formal description process.

The remainder of this paper is structured as follows. In Section 2, the concept and design of the toolchain is described and their functionality is illustrated. Section 3 closes with a conclusion and an outlook including future work.

## 2 Concept and Design of COMPL\textsubscript{e}T\textsubscript{e}

COMPL\textsubscript{e}T\textsubscript{e} realizes a COMmunication Protocol vaLidation Toolchain, by using formal and model-based specifications and descriptions. The concept takes the following requirements into account.

First, the support for creation and modification of graphical models which represent the behavior of communication protocols. This is realized by the front-end component, which is depicted at the left part of Figure 1. Moreover, an automatic transformation of the constructed graphical models to the input language of a corresponding model checker is realized. This transformation builds the link between the front-end and the back-end component in Figure 1. The back-end component integrates a model checker tool. Furthermore, editing the transformed models, based on the input language of the model checker is supported for evaluation purposes. In addition, properties can be defined with which models are checked against.

Secondly, beside the more functional requirements the toolchain is used within the Eclipse Integrated Development Environment (IDE). This ensures that components within the toolchain can easily be exchanged or modified (modularity and extensibility) and new components can be integrated in a simple way. Furthermore, open-source or free available existing tools are used in order to consider reusability.

![Fig. 1. Architecture of COMPL\textsubscript{e}T\textsubscript{e}](image-url)