MODELING APPROACH FOR NETWORKED EMBEDDED SYSTEMS WITH HETEROGENEOUS COMMUNICATION
Modeling Common Gateway Functionalities for Interconnection

Johannes Klöckner, Marcus Müller, Wolfgang Fengler and Yixin Huang
Computer Architecture Group, Ilmenau University of Technology, P.O. Box 100565, Ilmenau, Germany

Keywords: Model based design, Fieldbus, FlexRay, CAN, Building blocks, Network simulation, MLDesigner, Gateway.

Abstract: This paper presents a method to create system level models of gateway facilities to combine embedded fieldbus networks. Here automotive applications are used as representative with CAN and FlexRay. A strategy is introduced to create gateway models, which provide high flexibility in terms of reusability, replaceability, extensibility and flexibility. Todays systems often incorporate a very complex heterogeneous communication structure with real time requirements. These characteristics have an influence on the system design process. The goal of avoiding design errors in early development stages requires the validation on hierarchical functional models of various abstraction levels using simulation based analysis prior to hardware design. A current approach includes problem oriented system analysis. This approach is extended to analyze the system behavior of heterogeneous embedded systems focusing on switching mechanisms. Here lies the challenge on the connection of subsystems with distinct fieldbuses realizing different paradigms. Regarding real time aspects paradigm switches can be a source of errors in the system design.

1 INTRODUCTION

As a result of the increase in both system complexity and the amount of incorporated functionality a single embedded system has to be considered as a composition of several heterogeneous subsystems. Due to this the intra-system networking of an embedded system provides new challenges in system design. Developing such complex systems requires a lot of planning and design decisions on a profound knowledge of system behavior. The heterogeneity in a compound system stems from differences in hardware, software and subsystem architectures. In networked embedded systems the communication is an important aspect. Based on the complexity of systems and distributed applications the amount of communication between individual subsystems grows. Using communication technologies with different paradigms, e.g. event or time triggered, influence the system behavior.

The heterogeneity of large systems represents a great challenge in the system design. This high grade of variability complicates the development process. Mistakes in early system design stages can negatively influence the performance and development costs. Avoiding these errors requires the validation on functional models of various abstraction levels using simulation based analysis prior to hardware design.

To analyze the performance on system level (Henia et al., 2005) use an approach based on a scheduling analysis. A model based design approach (Salzwedel, 2004) enables an efficient top down development of a complete system using hierarchically composed building blocks, thus providing a high grade of reusability and exchangeability.

Aim of the presented work is the extension of an existing modeling approach studying networked embedded systems to allow the modeling of heterogeneous networked systems in a common way and the analysis of system behavior. It is based on the tool MLDesigner (MLDesign Technologies Inc., 2007) and fulfills the requirements of a model based design approach allowing system analysis and development. An example for a complex heterogeneous networked embedded system is an automobile. It is a system with a large amount of functionality, real time requirements and distributed characteristics. The complex communication infrastructure contains several protocols with different paradigms, e.g. FlexRay (FlexRay Consortium, 2005) and Controller Area Network (CAN). The mentioned properties can be found in a lot of systems belonging to areas engaged with automation and control. This paper is

Klöckner J., Müller M., Fengler W. and Huang Y. (2009).
MODELING APPROACH FOR NETWORKED EMBEDDED SYSTEMS WITH HETEROGENEOUS COMMUNICATION - Modeling Common Gateway Functionalities for Interconnection.
In *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics - Signal Processing, Systems Modeling and Control*, pages 230-233
DOI: 10.5220/0002207902300233

organized as follows. Section 2 gives a short description of the used modeling tool. Section 3 introduces related work and the basic modeling strategy. Section 4 describes the modeling concept. Section 5 presents the drawn conclusions and a brief overview of further development steps.

2 MODELING TOOL

Currently many tools exist, that realize model-based design to support the development process. In this work the tool MLDesigner by MLDesign Technologies, Inc. is used. The tool is dedicated to improving the design process from early concepts to implementation with mission and system level design. MLDesigner extends the Ptolemy project of UC Berkeley (The Ptolemy Project, 2009) with modeling paradigms. Different models of computation so called domains are provided, e.g. *discrete event domain* (DE) and *finite state machines* (FSM), *synchronous data flow domain* (SDF). In this work the *discrete event domain* and *finite state machines* are used. These two are well suited to modeling networked embedded systems.

Now a short introduction to the terms of MLDesigner. The *System* is the top level element in the modeling hierarchy. The building blocks can be atomic blocks called *Primitives*, specified as FSM or in C/C++ code, or hierarchical *Modules*. To communicate with their environment building blocks can use linked variables or ports. *Ports* are represented as arrows on the bounding box of a building block and are interconnected by signal paths. So called *Wormholes* allow the embedding of building blocks belonging to different domains.

3 RELATED WORK

In (Klöckner et al., 2008) a modeling strategy for networked embedded systems is discussed exemplified by a top-down developed generalized FlexRay protocol model. The communication system is defined as a composition of three different elements: *Host*, *Communication Controller* (CC) and *Channel* as shown in Figure 1. The combination of a host and a CC is called node. The corresponding protocol can be implemented within the CC, e.g. synchronization, error detection, message transmission and reception. In addition, the CC provides several services to the host to be interfaced by its application. The application of a node can be described within the host. The host

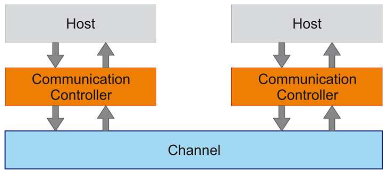

Figure 1: Basic Model Structure for networked embedded systems (Klöckner et al., 2008).

uses CC provided services to configure the CC, initiate the sending and receiving operations and process the received data. According to this the host contains a specialized sublayer realizing the protocol related access to the CC. At this point the division into different functional layers is visible, the host describes the functionality and the CC describes the type of communication. Nodes are grouped to communication clusters by connecting them to a channel.

The physical characteristics of the connection between nodes is described and modeled within a channel. On this level the physical delay of a transmission is determined by the respective message data length and a fault injection model simulates the transmission errors caused by physical medium and environmental influences.

The focus of the approach (Klöckner et al., 2008) lies on homogeneous communication systems. The model structure intends the extension of the approach to allow the modeling of heterogeneous systems.

4 GATEWAY MODEL

In order to provide concepts to monitor both the behavior of networked systems and the communication across different types of networks with different specifications and protocols, a concept for a gateway model which provides high flexibility in terms of reusability, replaceability, extensibility and flexibility to connect different communication technologies in a common way, is needed. Based on the model structures presented in (Klöckner et al., 2008) (Müller, 2007) and the libraries developed in these works, the creation of a customizable gateway module is possible. This allows the combination of heterogeneous subsystems in order to analyze the overall system behavior. Regarding the system architecture presented in Figure 1 a gateway is a special type of node. Normally a node is a combination of a host and a CC. A node realizing gateway functionality is now a combination of several *hosts*, associated CCs and a *Gateway Core*. The basic gateway structure is shown in Figure 2. The CCs contain the identical functionality as described before. A common strategy to model an

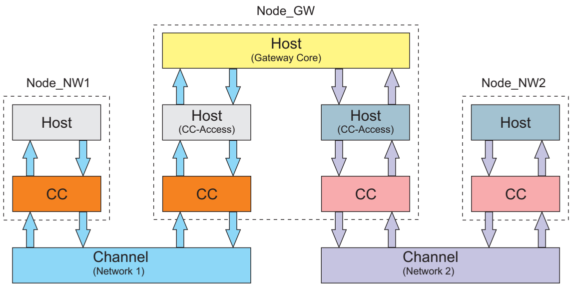

Figure 2: Extended Model Structure for networked embedded systems.

application is not yet designed. Therefore the hosts realize the specific sublayers to access the CCs. The *Gateway Core* can be seen as an application of the node providing the gateway base function. This concept fits the described modeling strategy. Anticipating a more detailed model containing different applications in a node divided in several tasks it is possible to extend this architecture to enable the analysis of the system behavior including other effects, e.g. properties of task scheduling.

Gateway facilities handle the data exchange between heterogeneous computer networks. Normally, they are designed to combine specified networks, and fulfill specified needs, which are based on the application. Hence, there is no predefined specification of a gateway. Although there are no fixed design approaches for gateway design, certain common functions can still be assumed for any specialized communications gateway: routing, protocol conversion, dataflow control, real time constraint checking for message transmission and message filtering or segmentation. The purpose of this paper is to develop a universal gateway model, which facilitates the communications between different networks. The design of the presented approach is similar to the idea of a time triggered gateway described in (Shaheen et al., 2007).

As described in (Fahmy, 1995), the gateway model is based on a shared medium approach, which uses an internal network to exchange uncommitted messages between *Host* and *Gateway Core*. Received frames are converted to an uncommitted protocol, after which the information is handled by the gateway module. Finally the information is converted to match the destination protocol. The uncommitted protocol contains three values: *Source Identifier*, *Destination Identifier* and *Datafield*. The routing information of each message can be identified by *Source Identifier* and *Destination Identifier*. The data of the received message is stored in the *Datafield* of the uncommitted message. To fulfill the flexibility requirement the gateway model is a composition of two kinds of modules: protocol related and unrelated modules. Only uncommitted gateway messages can be processed inside the protocol unrelated modules, and the protocol related modules process the specific protocol messages. In order to extend the gateway model to support a larger amount of protocols, only the protocol related modules need to be created, while the protocol unrelated modules remain unchanged. This approach provides enormous reusability, replace ability and extensibility. It allows an easy generation of a customized gateway to realize relevant system characteristics.

An important aspect of the gateway functionality refers to (Hörner, 2007). A gateway should provide at least two methods of data exchange - PDU based and signal based gateway functions. A PDU gateway routes the protocol data units (PDUs) unchanged between two networks. In this case the data carried on both networks, source and destination, are identical regarding content and length. It is also possible, that only signals, which are contained in the received PDUs from source network, are needed on the other network. In this case, the gateway does not transfer the entire PDU, but sends the individual signals to the corresponding destination network. To achieve this, a single received PDU is disassembled in signals according to the specification of the source network. Afterwards signals are grouped together congruent to configuration of the destination network and assembled to PDUs, which are send across the destination network.

The central unit of the gateway model is the module *Gateway Core* shown in Figure 3. This module contains protocol unrelated functions, e.g. buffering of received messages, routing, message segmentation and disassembling messages into signals. Based on the specification of function blocks with standardized interfaces a high adaptability of the architecture is achieved allowing the extension and substitution of single functions. Therefore it is possible to analyze the system performance of several realizations differing in their buffer strategy. In contrast to the gateway

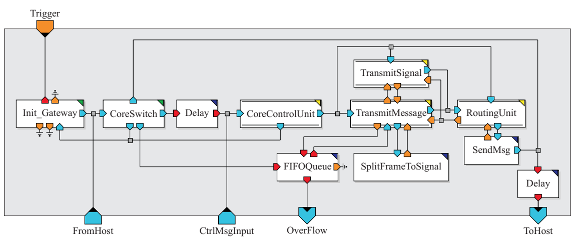

Figure 3: Module Gateway Core.

core, the protocol related gateway functions, such as protocol conversion, initialization of the communication controller, message transmission and message reception, etc. are realized within the respective *Host* module. The host connected to the source network receives a message via the CC and converts it into an uncommitted message. Afterwards the message is sent to the gateway core. Inside the gateway core, the uncommitted message will be processed and routed to the destination host. There it is converted into the corresponding standard. Each host has a distinct identifier. This allows the connection of an unlimited number of networks to the gateway model.

The example scenario shown in Figure 4 contains two CAN and two FlexRay networks which are connected by a gateway. The system itself is an extension of an application presented in (Hedenetz and Belschner, 1998). Each network contains two or more nodes. The behavior of the used modules CCs realizes the connection to CAN and FlexRay. The analysis of the simulation results showed the correct behavior of the created model.

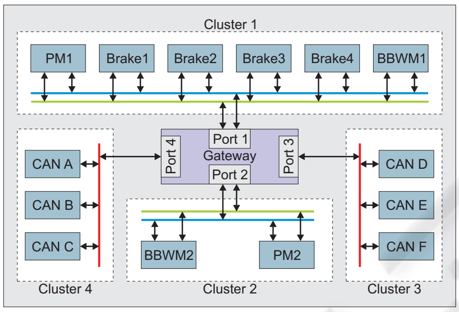

Figure 4: System used as reference for validation.

5 CONCLUSIONS

The modeling strategy and developed common gateway model presented in this paper allow the universal design and analysis of heterogeneous communication networks. Modularized components in the gateway library represent available functions of a gateway facility and allow an easy construction of a wide range of systems. The basic model concept behind the gateway model enables the analysis of communication and behavior of heterogeneous networked systems and supports the design process of such systems. In early stages of the development process the evaluation and verification of system properties can be provided prior to hardware design.

Up to now there are only few communication protocol models available. To extend the ability for system modeling it is necessary to extend the stock of models. At present this limits the systems which can be analyzed by the presented approach to CAN and FlexRay systems. To gain access to real world examples from within the automotive application domain the design of models realizing MOST (Media Oriented Systems Transport) and LIN (Local Interconnect Network) are required. Future work will also deal with the automated import of gateway routing information by using e.g. CAN Database and FIBEX (ASAM, 2007). This can be supplemented by concepts of automated model generation.

REFERENCES

ASAM (2007). *FIBEX - Field Bus Exchange Format Version 2.0.1*.

Fahmy, S. (1995). A survey of ATM switching techniques. CIS-788-95 Semester Report, Ohio State University, CIS Dep.

FlexRay Consortium (2005). *FlexRay Communications Systems - Protocol Specification Version 2.1*.

Hedenetz, B. and Belschner, R. (1998). Brake-by-wire without mechanical backup by using a ttp-communication network. page 2.

Henia, R., Hamann, A., Jersak, M., Racu, R., Richter, K., and Ernst, R. (2005). System level performance analysis - the symta/s approach. In *IEE Proceedings Computers and Digital Techniques*.

Hörner, H. (2007). Das universelle Gateway-Steuergerät. *Elektronik Automotive, Sonderausgabe AUTOSAR*, pages 33–35.

Klöckner, J., Köhler, S., and Fengler, W. (2008). Model based design of networked embedded systems. In *ICINCO-SPSMC*, pages 253–259. INSTICC Press.

MLDesign Technologies Inc. (2007). *MLDesigner Documentation, Version 2.7*. http://www.mldesigner.com/.

Müller, M. (2007). Dokumentation framebasiertes canmodell.

Salzwedel, H. (2004). Design technology development towards mission level design. In *49. Internationales Wissenschaftliches Kolloquium IWK'2004*.

Shaheen, S., Heffernan, D., and Leen, G. (2007). A gateway for time-triggered control networks. *Microprocess. Microsyst.*, 31(1):38–50.

The Ptolemy Project (2009). http://ptolemy.eecs.berkeley.edu/.