# Monitoring and Fault Injection of X-by-Wire Communication Networks

Roman Pallierer
Dependable Computer Systems
DECOMSYS GmbH
Stumpergasse 48/28, A-1060 Vienna
pallierer@decomsys.com

Martin Horauer
University of Applied Sciences
Technikum Wien
Höchstädtplatz 5, A-1200 Vienna
horauer@technikum-wien.at

Andreas Steininger
University of Technology Vienna
ECS Group E182-2
Treitlstr. 3, A-1040Vienna
a.steininger@ecs.tuwien.ac.at

## Introduction

Automotive electronics is the key innovation driver in the automotive domain. Currently the automotive industry considers the replacement of mechanical or hydraulic implementations of safety-critical automotive systems such as braking and steering by electronic counterparts for safety, comfort, and cost reasons. As the name suggests, these X-by-wire systems highly depend on the underlying communication network, cf. [LH02]. An industrial consortium of leading automotive and electronic manufacturers has established a communication network, termed FlexRay[1], to address the stringent requirements of such safety-related applications.

While by-wire systems will enable an unprecedented boost of functionality of such systems, however, customers will eventually hesitate to accept this completely new technology. Therefore it is of utmost importance to establish confidence in automotive electronic systems by all possible means. Undoubtedly, one of these means will be testing. Efficient methods for test and diagnosis will be required to check the functional integrity of all involved components and hence prevent failures.

In cooperation with the Vienna University of Technology and the University of Applied Sciences, DECOMSYS – one of the major developing partners of the FlexRay consortium – has setup a research project “Systematic Test of Embedded Automotive Communication Systems” (STEACS)[2]. This project positioned between fundamental and applied research aims at developing diagnosis and test methods for the communication network FlexRay.

This paper presents an overview of the scope of our technical approach. First we detail a suitable layer model of time triggered communication systems based on the FlexRay protocol. Next, we show how this model can be applied to identify faults in the communication system. To that end, we illustrate some first prototype implementations before we conclude the paper with an outlook to further work.

[1] http://www.flexray.com/

[2] The STEACS-project (http://embsys.technikum-wien.at/steacs.html) received support from the Austrian “FIT-IT[embedded systems” initiative, funded by BMVIT and managed by the FFF under grant 807146.

## Scope

The main scientific challenge of diagnosis and test methods lies in the elaboration of a systematic test approach to handle the complexity given by the various different configuration possibilities and network topologies. In particular, due to the distributed nature of the communication services we cannot perform the test in a node-by-node manner. Performing an unstructured functional test, on the other hand, is not reasonable, since the complexity of the distributed system is substantially higher than that of a single node. Since test efforts rise more than linearly –- typically $O(n^2)$ – with system complexity this would either result in excessive test duration or in poor test coverage. To that end, a systematic test approach shall provide in-depth observability required from the different points of user views (e.g., OEM/system integrator view, supplier/application engineer view). The fundamental purpose of such a test is to check whether the system under test provides all services as specified. Should any service deviate from the specification or be completely missing, we have encountered a failure, and this is exactly what the test is intended to discover. If we want to structure the test, it will probably be a good strategy to identify and separate all services, break them into sub-services (which we call mechanisms) as far as possible and focus the test to each of these mechanisms separately.

Therefore, in the context of the STEACS project we have developed a layer-based model to address these systematic test requirements for time-driven communication networks such as FlexRay. The rationale why we chose to employ a layer model as the central point of our test concept is the following:

1. The layer definition is generally based on the notion of services and hence suits naturally to our intended test strategy. In particular the layer model shall reflect all services (and the respective mechanisms) provided by the system with a fine (ideally atomic) granularity
2. In the process of model elaboration the decomposition of the complex global communication service into smaller mechanisms is performed in a very systematic manner. This aids in achieving a complete picture of all relevant mechanisms (i.e. not forgetting mechanisms)
3. The comprehensive layer model not only shows the individual services and mechanisms but also their interrelations. Given this picture it is easy to identify all relevant inputs of the particular service or mechanism as well as the receivers of the service outputs in the next level. This can substantially simplify diagnosis and helps in identifying potential sources of error (fault model).
4. By using the layer model it is easy to determine the abstraction level at which monitoring should be performed to test a particular mechanism. Thus overheads in terms of additional hardware and time can be saved.
5. If potential error signals issued by a mechanism are included in the model, a hierarchy of error signals can be constructed that further eases diagnosis.

Furthermore, in order to test the fault tolerance- and error detection capabilities of the communication system, e.g. redundant message transfer, we will employ fault injection, see [HTI97]. Again the layer-based model can be helpful to provide a systematic structure for planning these experiments.

## *Layer-based Model*

In the course of our project we developed a layer-based model specifically targeted for time triggered communication systems – in particular for FlexRay – in order to ease detection and diagnosis of faults and aid in bus injection, cf. [ASHP04b].

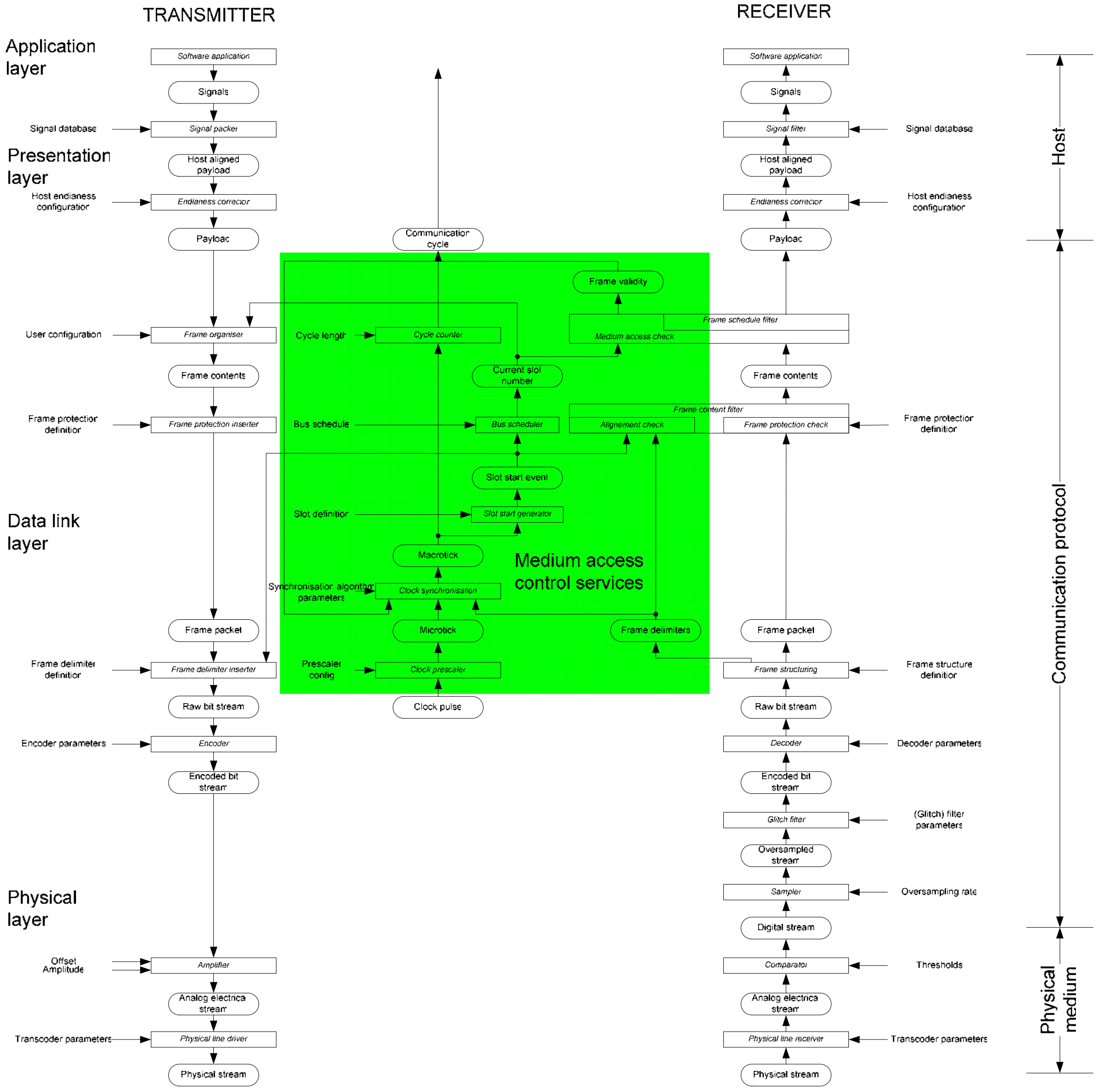

Figure 1 shows an overview of the model.

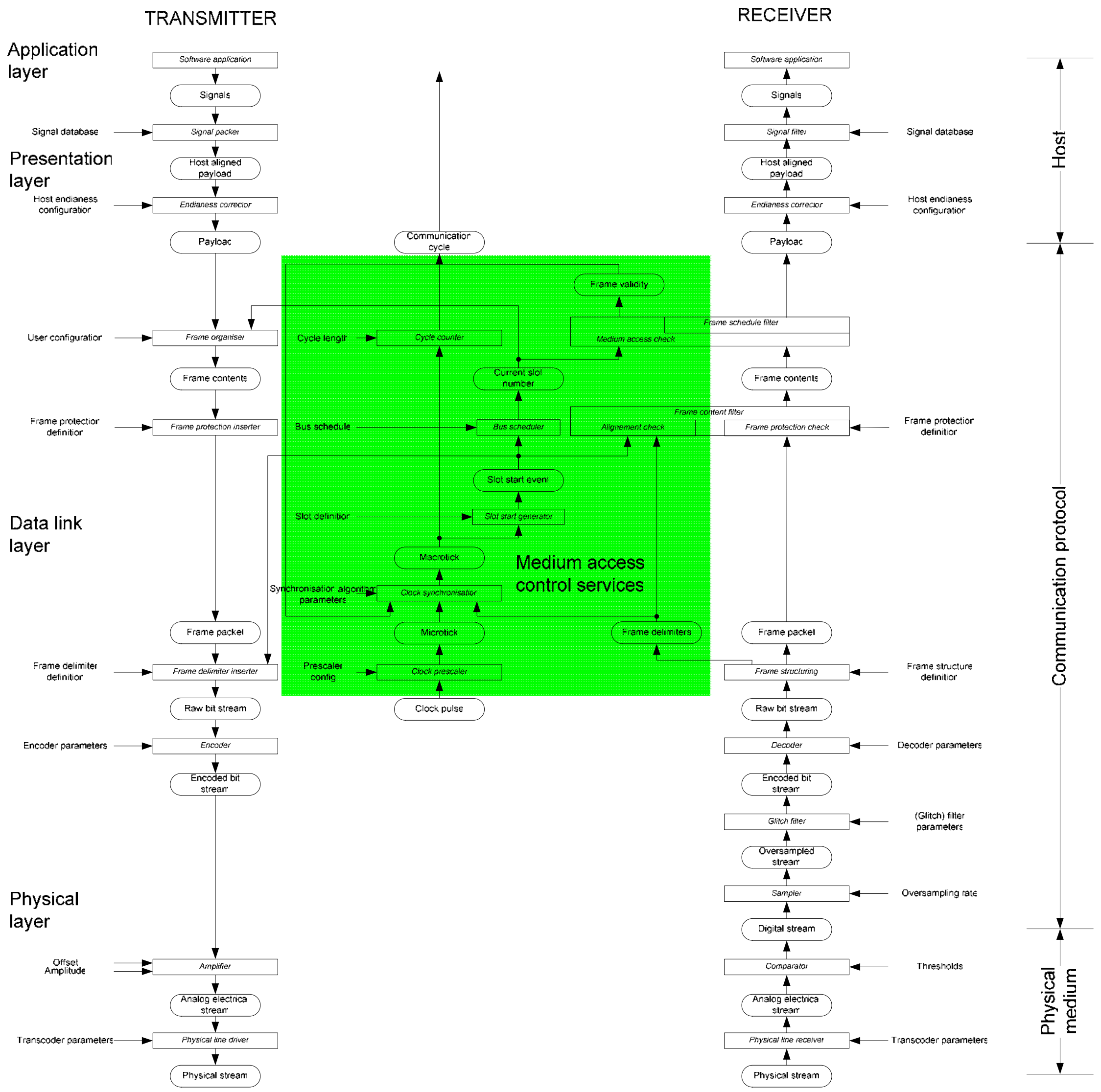

Figure 1: Layer model for time triggered architectures

The model structure is structured in relation to the OSI reference model. In particular, the proposed layers can be mapped to four of the seven layers of the OSI reference model. Each layer is further refined by the functional properties mechanism and abstraction level given by an actual implementation of the FlexRay protocol. Every mechanism performs an input to output transformation, i.e. it maps the information of one abstraction level to the next one. To that end the mechanisms are configurable to perform the intended operations. Although the layers are arranged in a hierarchical order, the mechanisms within the layers are not necessarily hierarchically organised and some loops might appear. These loops make the system more difficult to test and diagnose because the inputs of a mechanism might be indirectly dependent on its outputs. However, by breaking the loop, the mechanism can be observed directly at its outputs and controlled directly at its inputs and therefore totally tested. It is an additional benefit of our model to make these loops clearly visible.

The left column of the model represents the transmit path while the right column represents the receive path. The “time” path in the centre column is specific to our model and reflects the fact that our model is targeted to time triggered communication systems. The lowest layer is the *physical layer* whose service is to transmit raw bits from a node to another through a communication channel. Its interface to the upper layers are serial bit streams received from or to be sent through the communication system.

The next layer is the *data link layer*, whose services are to organize the bit stream into a frame structure with several parts, to detect potential transmission errors, and to control the medium access. In contrast to the data link layer defined in the OSI reference model, this layer does not include an acknowledgement system to automatically initiate re-transmission of faulty frames. Upper layers can access it through the frame payload, in other words frames containing the data to transmit. In addition this layer provides a view of the network synchronized time.

The network layer, transport layer and session layer defined in the OSI reference model are not present in our system model because the corresponding services are not particularly pronounced in FlexRay: With a communication based on broadcast channels, every frame transmitted is seen by all the receivers and the addresses are implied by the schedule; therefore explicit routing as provided by the network layer is not needed. Typical services for the transport layer are splitting large data when sending and re-organization when receiving, or reliable communication. These services are typically not needed by applications in automotive field and therefore this layer is not implemented. Finally the session layer is not present because establishment and control of sessions between two nodes are implied by the time static schedule.

The *presentation layer* is located upon the data link layer. Its services are to format the data contents according to the needs of the application. The optional implementation of the FTCOM layer according to OSEK[3] is also located in the presentation layer. The layer presents a signal based interface to the application layer.

Finally the *application layer* is situated on top of the model and makes use of the services of the lower layers to communicate from a node to another.

One technical approach of our STEACS project is to develop a FlexRay node that allows monitoring the information at the presented abstraction levels. By adding this node to a FlexRay cluster and assuming proper functionality of this test node it will be possible to uncover faults originating from remote nodes.

## Prototype Implementations

Based on the layer model we started with prototype implementations to analyse the feasibility of our approach. After a detailed requirements analysis and several discussions with automotive partners of the FlexRay consortium, we decided to start by focusing on the data link and presentation layer for the first implementations, cf. [ASHP04a]. Besides of the use in the STEACS project, the first implementations of our monitoring solution are presently evaluated by the automotive partners in parallel. Next, we will include aspects of the physical layer as well.

### *Monitoring Architecture*

For a fast and efficient implementation of the monitoring device we decided to make best use of an existing FlexRay FPGA implementation. This approach relieves us from implementing the standard FlexRay controller which allows us to concentrate our efforts on the monitoring aspects. Furthermore, this avoids compatibility problems with the standard controllers we use as monitoring target. Moreover, using already tested components guarantee functionality, interoperability and save time and effort in developing and testing, cf. [ASHPF04].

[3] http://www.osek-vdx.org/

Figure 2 shows a typical small FlexRay cluster built using a set of COTS FlexRay nodes. In the presented example we have added our test node with the modified FlexRay modules highlighted. In particular, a FlexRay cluster consists of a number of nodes that are interconnected using one or two physical layer channels. Each node contains a communication controller part with bus drivers, protocol engine and controller host interface, and a host part. For the test node we added monitoring support for the relevant abstraction layers found within the protocol engine and replaced the controller host interface by a dedicated monitoring interface.

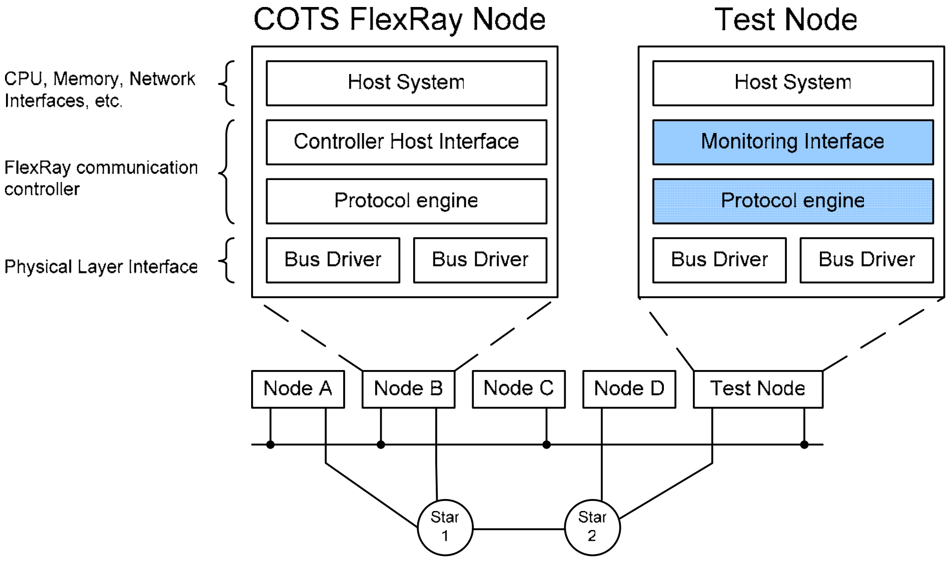

Figure 2: FlexRay Cluster with a Test Node

Figure 3 shows the some adaptation details made for the implementation of our monitoring interface. The main problem we are facing with this approach is the attainable data rate. The COTS FlexRay controller architecture has been optimized for the data rates occurring during normal bus operation. According to our monitoring requirements it should be able to handle multiple data streams in parallel when used as a monitoring device which means that a much higher data rate must be handled. This issue turned out as one of the most critical aspects in our architectural considerations.

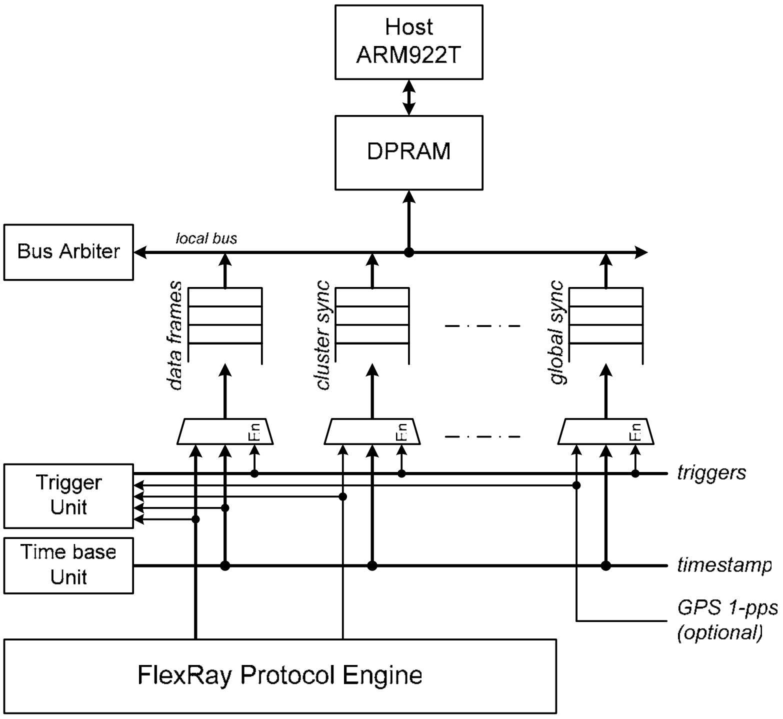

Figure 3: Monitoring Interface

To that end, we replaced the controller to host interface of a COTS FlexRay controller implementation as illustrated in Figure 2 with a large dual ported RAM interface and added several hardware units – in particular, a time base and a trigger unit – to the existing FlexRay controller. The block-diagram of the monitoring hardware implementation given in Figure 3 presents these modules without the interfaces required to configure and program these units, and the host interfaces (e.g. memory, Ethernet interface, etc.) required to transfer the data to a remote host computer for post-processing and visualization.

In principle, following a trigger every recorded event is encapsulated in a dedicated frame format optimized for the processing by the host (e.g. data frames, cluster synchronization frame, global synchronization frame, etc.) and stored in a dedicated queue. The latter are required because the information accumulates in parallel. A bus arbiter manages the local bus and when scheduled, copies the contents of the queues into the dual-ported memory. The dual ported memory is organized as a FIFO; whenever a certain threshold is reached an interrupt request is signaled to the host processor. In the respective interrupt service routine the processor empties the FIFO up to the threshold and stores the relevant information in a large external dynamic memory bank. Higher level software succinctly processes the stored information and copies it either to a large flash-disk or streams it via an Ethernet interface to a remote host for visualization and further processing.

Furthermore, for both a replay and fault injection mode a similar architecture is used, although, the data path is in the reversed direction and some additional units are added (e.g. to extract the required information from the frames).

### *Monitoring Implementation*

For the test node we chose a prototyping hardware platform for FlexRay, the DECOMSYS::NODE<ARM>, as basis for implementing the proposed monitoring functionality.

Figure 4: Prototype Hardware Platform

Using this hardware platform, we exploit the Altera Excalibur architecture. In particular, an EPA4 device offers an embedded ARM922T processor hardcore with several integrated peripherals and a 400k FPGA that is used to implement the FlexRay communications controller and the hardware support for monitoring, replay and fault injection.

The operating system software executing on this platform for our prototype is an RTAI Linux that runs the low level device driver for interfacing to our monitoring hardware and the high level software to handle the memory and the network interfaces.

### *Demonstrator Setup*

Another technical aim of the STEACS project is the setup of a demonstrator system to enable an experimental proof-of-concept and validate the correct functional behavior of the developed tools under real-life conditions. Figure 5 illustrates a possible setup of the demonstrator system.

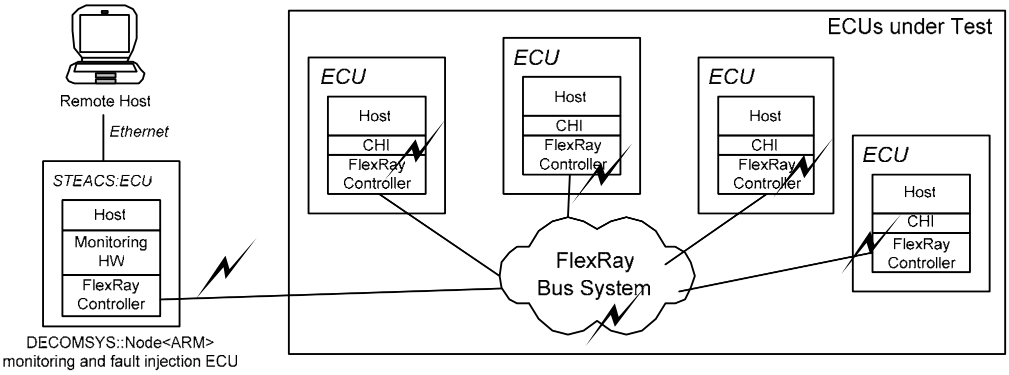

Figure 5: Demonstrator Setup

Herein the communications subsystem constitutes of several Communication-Host Interfaces (CHI), the FlexRay Controllers and, the bus system of the automotive distributed system under test. Faults originating within this communication subsystem are monitored by means of one or more dedicated test nodes. Such a test node can also inject faults into the system and monitor the effects and consequences. A remote host computer is used to read out and validate the collected information. Furthermore this remote host facilitates controlling of the various different fault scenarios that the system under test is subjected to.

## Conclusion and Outlook

From a user point of view it is of utmost importance to have the right tools available at the right time. Thus, tools development must be in line with the development of the technology and the corresponding requirements of possible users. In addition, cost-efficiency is always an issue – still providing best possible solutions to meet the high demands for monitoring and fault injection.

Based on requirements of our automotive partners we implemented a prototype that meets the current demands. Therefore, we utilized our layer-model focussing on the data and presentation layer. Further activities for the extension of this implementation to cover the physical layer are ongoing. A demonstrator setup will be used to perform extensive fault injection experiments.

So far, our technical approach proofed very suitable to cover the requirements of monitoring and fault injection. Due to the layer-based model, we are able to provide a scalable and extensible monitoring infrastructure focussing on the individual layers and services required.

These research activities and the resulting product developments will provide a necessary basis for monitoring and fault injection methods. In general they will support automotive OEM partners to establish and increase confidence in future safety-critical X-By-Wire applications.

For further information on the research project and the upcoming product developments please refer to our homepages http://embsys.technikum-wien.at/steacs.html and http://www.decomsys.com.

## References

[ASHPF04] E. Armengaud, A. Steininger, M. Horauer, R. Pallierer, and H. Friedl: A Monitoring Concept for an Automotive Distributed Network - The FlexRay Example. 7th IEEE Workshop on Design & Diagnostics of Electronic Circuits & Systems (DDECS-2004), Slovakia, April 18-21, pp.173 -178, 2004.

[ASHP04a] E. Armengaud, A.Steininger, M.Horauer and R.Pallierer: Design Trade-offs for Systematic Tests of Embedded Communication Systems, The International Conference on Dependable Systems and Networks (DSN-2004), Florence - Italy, June 28 - July 1, 2004. (to appear).

[ASHP04b] E. Armengaud, A. Steininger, M. Horauer, R. Pallierer: A Layer Model for the Systematic Test of Time-Triggered Automotive Communication Systems. 5th IEEE International Workshop on Factory Communication Systems, 2004 (submitted).

[LH02] G. Leen and D. Hefferman: In-Vehicle Networks, Expanding Automotive Electronic Systems. IEEE Transaction on Computers, pp. 88-93, January 2002.

[HTI97] M.C. Hsueh, T.K. Tsai and R.K. Iyer: Fault Injection Techniques and Tools. IEEE Transactions on Computer, Vol. 30, No. 4, pp. 75-82, 1997.