FIBEX – An Exchange Format for Networks Based on Field Busses

Josef Krammer, Pascal Bornat, Günter Dengel, Stephan Kuttenkeuler, Rainer Lukas, Klaus Oberhofer, Jens Ruh, Joachim Stroop, Hans Quecke

FIBEX – An Exchange Format for Networks Based on Field Busses

Authors
Josef Krammer, BMW AG, Pascal Bornat, Cadence, Günter Dengel, Berata GmbH, Stephan Kuttenkeuler, Sulzer GmbH, Rainer Lukas, Robert Bosch GmbH, Klaus Oberhofer, IXXAT GmbH, Jens Ruh, DaimlerChrysler AG, Joachim Stroop, dSPACE GmbH, Hans Quecke, Vector Informatik GmbH

Abstract:

FIBEX (field bus exchange format) is based on XML. It has been developed to support the description of frame based field busses. FIBEX can describe complete networks comprised of clusters with different protocols and gateways connecting them. This multi protocol capability provides the means for the design and management of large heterogeneous networks. As an open standard (in ASAM MCD-2[FBX], see [1]) it is intended to bridge the existing gap between the large variety of tools. Great emphasis has been given to a comprehensive description for an easy access to FIBEX.

1. Introduction

In the last decades an increased number of intelligent devices and electronic control units (ECU) has been used in many applications. In today's high-end cars a few dozens ECUs are used to control the large variety of functions whereby many real time applications are distributed via several ECUs. The need to interconnect these units forming clusters and networks for sharing information has risen dramatically.

A single cluster (the term cluster is used as synonym for data bus) can transfer several hundred or even thousands of shared signals. A network can contain several clusters that are interconnected via so called gateways that transfer signals between them. Many protocols used within the different clusters have been developed to support the different needs of a wide range of applications. Among them are CAN, LIN [2], FlexRay [3], and MOST© [5]. Obviously, the importance of communication technologies has been increased dramatically.

Databases are necessary to store the information of all signals and their parameters in order to manage efficiently these networks. That data is used for various design and verification steps. Many dedicated tools have shown up on the market to support that. However, their application focus may differ. Some support specific tasks (e.g. bus monitoring), others are more general for design of frames, configuration of software, and rest bus simulation (a commonly used term for hardware in the loop simulation, where the bus behavior if some ECUs are simulated, while others are real).

Unfortunately, no common format for the exchange of data between different tools is presently available. This situation is shown in figure 1. The growing number of signals in a reasonable network yields an increasing demand for a straightforward way to exchange data to avoid an error prone manual handling of redundant data. Furthermore, the fast-growing communication requirements of the implemented functions result into an increasing number of new, extended, and more dedicated tools. This increases the need for an exchange format that supports better data handling.

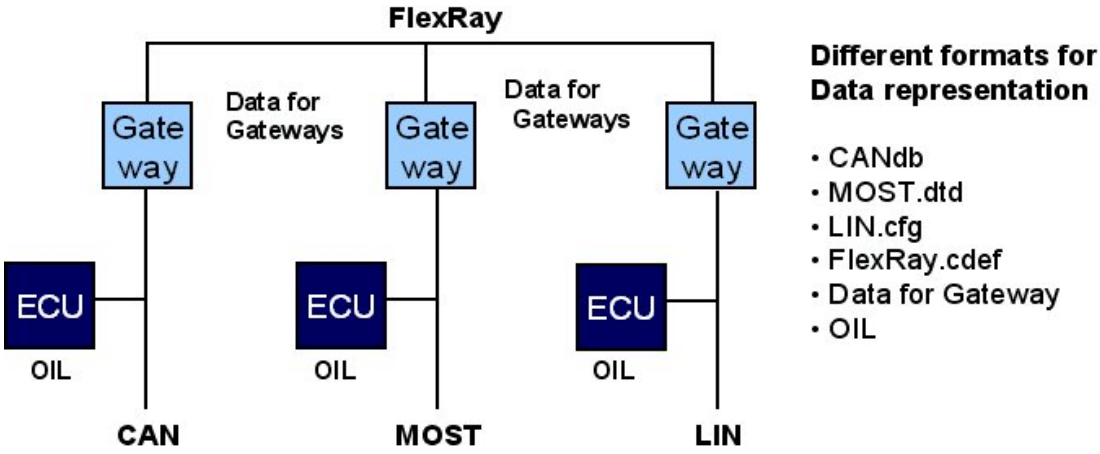

Figure 1: Managing distributed real time systems by conventional tools and data bases.

The FIBEX working group has been established in 2002 with the goal to overcome these difficulties. The founders and working group members are either responsible to manage automotive networks, or are involved in the development of tools. In June 2003 the group joined ASAM [1] that provides a good platform for these kinds of standardization activities (ASAM MCD-2[FBX] working group). At that time several additional companies joined the group and contributed to the now very well aligned and harmonized result.

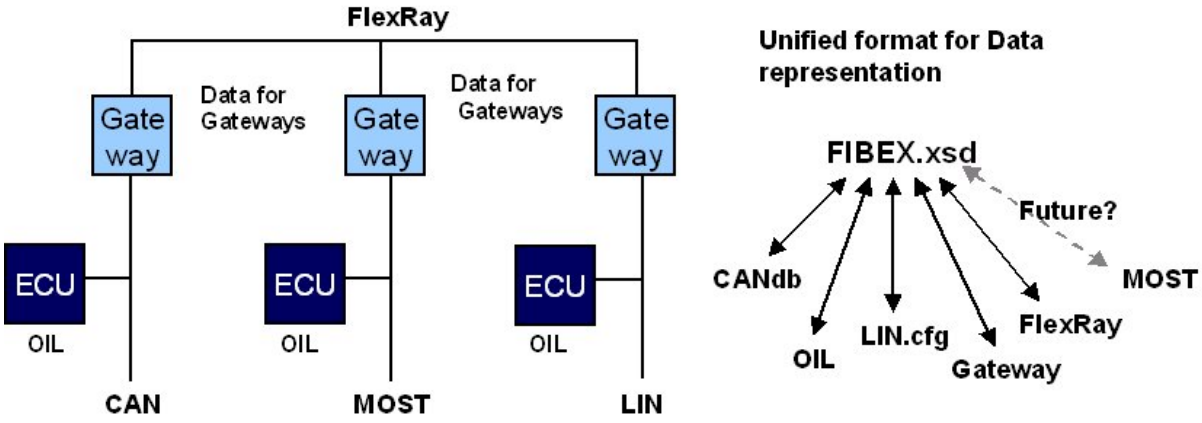

Figure 2: Describing a complete network with FIBEX

As illustrated in Figure 2, FIBEX is an XML data structure that can describe a complete network within one file. This data generally includes definitions of signals, frames, clusters, and ECUs sending and receiving the signals. FIBEX can be used to transfer network information between different tools. It is not intended to replace but to supplement established standards that are often used to store data locally. The coexistence with existing formats is regarded as an advantage since available tools and standards can still be used. FIBEX provides a bridge between them.
FIBEX' strength is seen in the field of tool integration, data exchange, and data integrity. It is "more powerful" than most of the more dedicated formats. For example a single instance can contain the description of clusters with different protocols as well as the transfer function of the gateway between them. Note, that the formats of Figure 1 can represent the data of one type of protocol only. In this case the data integrity of the definitions can be checked efficiently. Hence, FIBEX should be used as an extension whenever the specific format is not sufficient. Since many tool vendors have announced to support FIBEX, a conversion between FIBEX and specific formats should be well supported.

FIBEX is also a continuation of preceding activities in MSR (MSRnet.dtd) [1, 4]. The pure description of CAN clusters has been extended and experiences have been brought in.
The paper is structured as follows: In section 2 the use cases and requirements that guided the definition of FIBEX are explained. Section 3 describes the content and capabilities. A guideline for usage and an example are completing the paper (Section 4).

2. Use-Cases and Requirements

Many activities are supported by FIBEX. Figure 3 illustrates some of the most important ones.

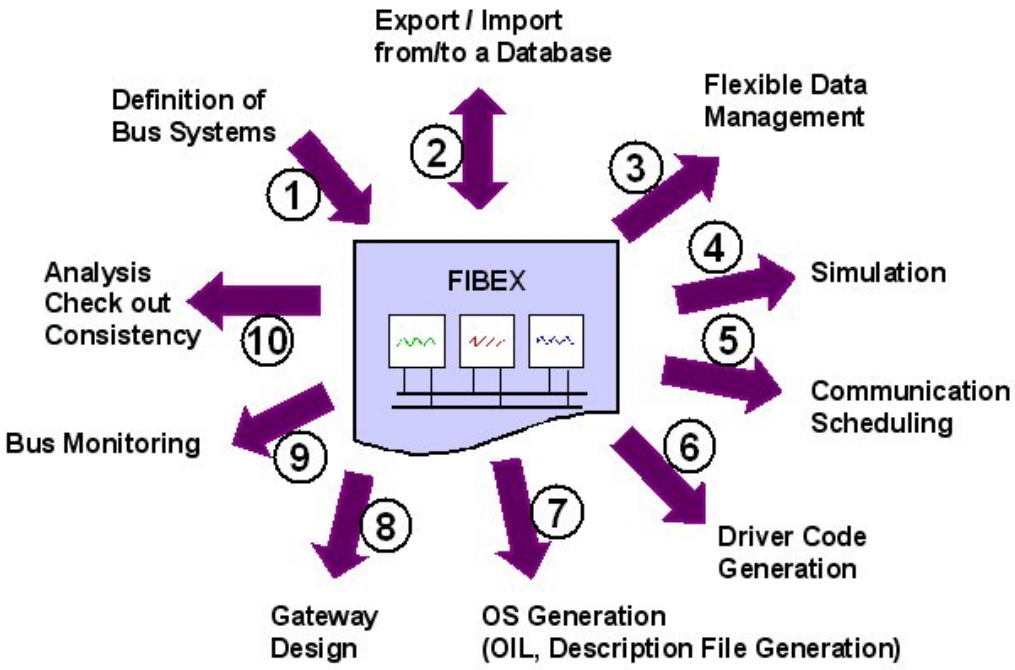

Figure 3: Use Cases for FIBEX

1. The definition of a bus system is a classic task. Generally, a bus with the connected ECUs, the signals, and frames is defined by using an appropriate design tool. The result should be written to a FIBEX XML file.
2. Import/Export to/from a database. This use case has to support the merging of typical network data with other data that may be stored in a more general database.
3. A flexible data management supports the distribution and gathering of data and portions of it between different companies (OEM and sub suppliers).
4. FIBEX may be used for offline simulation of the distributed application's behavior (down to the time domain, by taking into account the frame timing information)
5. Within the Communication scheduling the timing of the transmitted signals and frames is defined. In synchronous systems such as FlexRay, this is an essential design step that may be performed by specialized tools.
6. With a driver code for software interfaces can be generated that allow the application to access the signals via an abstract API.
7. FIBEX information can be completed with task information and HW information, and can be used to describe ECU configuration (e.g. in OIL – OSEK Implementation Language) for code generation tools
8. Since gateways can be described in FIBEX, their configuration and mapping functions can be derived from FIBEX.
9. The network- and symbolic information of the frame contents can be passed to bus analyzing (bus monitoring) tools via FIBEX.
10. FIBEX can provide statistical information, e.g. for bus load and other analyses.

Derived from the use cases and experiences made with existing formats several essential requirements have been set as prerequisites for the development of FIBEX.

- The exchange format has to support (nearly) all communication systems in the car (CAN, LIN, FlexRay, byteflight, TTCAN, MOST*…)
- It has to be standardized, free, and open for all development partners without any charge (OEM - Tier1-Tool Supplier -...)
- It must be capable to specify the complete network (including gateways) in one file
- The (logical) HW-topology (communication systems, ECUs, channels, gateways, …) must be included (no information about the physical topology included in FIBEX 1.0)
- The data must contain information about signals, frames, routing, …, and
- project information (version, vehicle type, ...).
- Capabilities to exchange (import - export) specific subsets of networks (e.g. configuration for one ECU) to support OEM-supplier relationship must be possible.
- Exchange format has to be user extendable
- Exchange format has to be user friendly (e.g. readable, documented, ...)
- Format has to be defined using commonly used, well tested and well supported technologies

3. Content and capabilities of FIBEX

FIBEX can be understood as an object oriented decomposition of communication systems into the following major elements:

ECU	Electronic Control Units contain the functions that communicate via the clusters. Generally an ECU contains one or more functions.
Function	A function is a part of the distributed application running in an ECU. It is assigned to an arbitrary number of input ports (input signals) and output ports (output signals).
Signal	The signal entity serves two purposes: From the “functional” perspective, it is an input or output parameter of a function representing a physical or logical value of a specific data type. From the “communication” point of view it is a stream of bits of a defined length.
Cluster	A cluster describes the ensemble of ECUs, which are linked by a communication medium of arbitrary topology (bus, star, ring, ...). The nodes within the cluster share the same communication protocol, which may be event-triggered, time-triggered or a combination of both.
Channel	The communication medium of a cluster consists of one or more (redundant) communication channels. The channel entity interlinks the ECUs of a cluster physically: An ECU is part of a cluster if it contains at least one controller that is connected to at least one channel of the cluster. In single-channel systems, this entity is known as “bus”; to avoid confusion we recommend not using this term any more.
Frame	A frame is the smallest piece of information that is exchanged over the communication system. A frame is a datagram with a header section and a payload section of a certain length in bytes, which contains an arbitrary number of non-overlapping signals and/or multiplexers/sub frames. The send-stimuli of a frame are described in the “frame triggering” elements of a frame.
Gateway	A gateway is an ECU that is connected to two or more clusters (channels, but not redundant), and performs a frame or signal mapping function between them.

The structure of FIBEX can represent a nearly arbitrary subsets of systems. Hence it can support any stage of the design process, independent from the succession of the definition steps for the various entities.
There are also some restrictions. FIBEX is restricted to represent one system at one time only. This means that there is no multi-vehicle or multi-version handling.

* not yet covered in FIBEX version 1.0, but planned for a future release

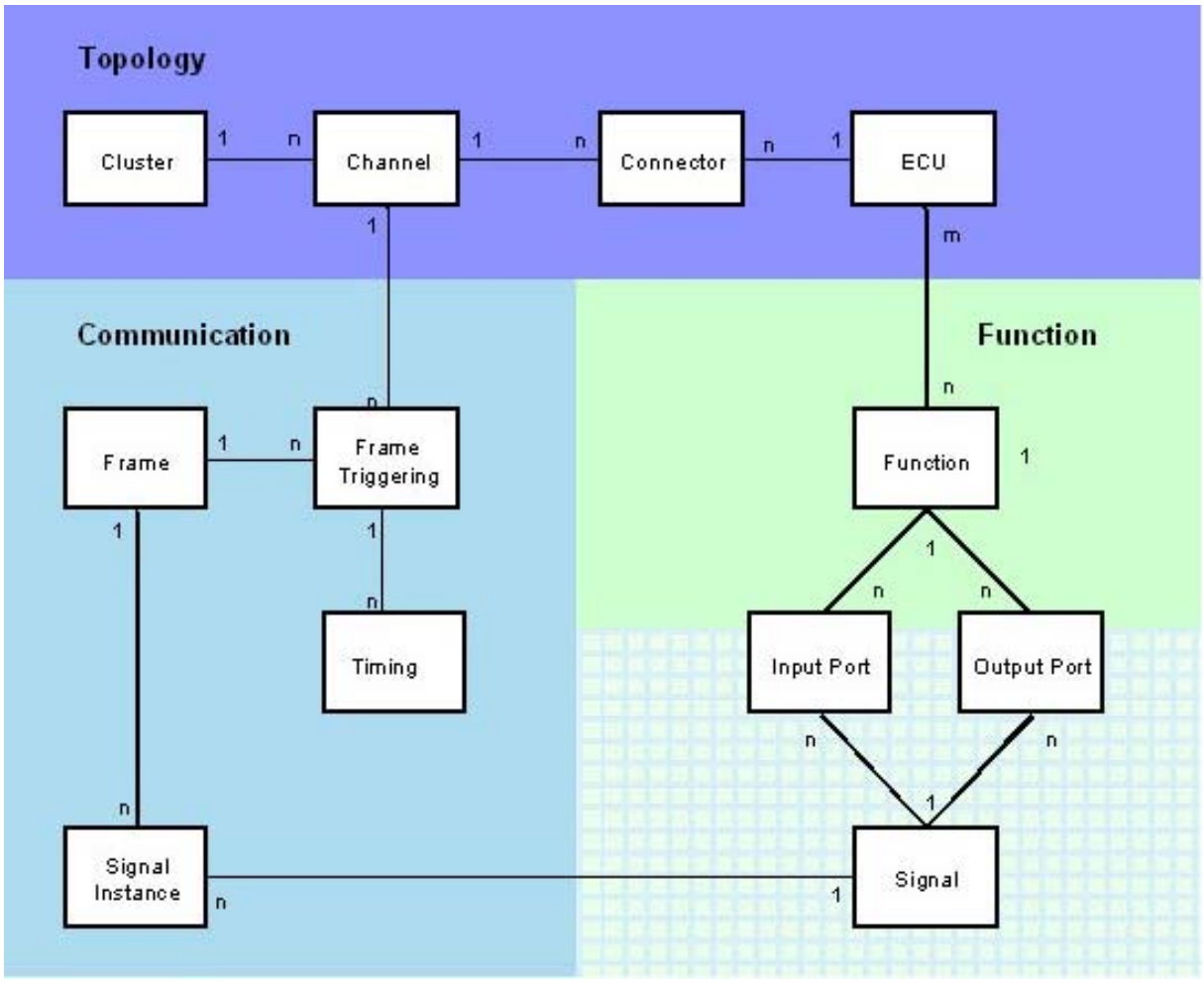

Figure 4: Reduced ER diagram that can describe already many systems

Figure 4 shows the relationship between some entities. For a better understanding, some features such as requirements, gateway, multiplexing within frames, and project information are not shown. The numbers at the relation between the entities characterize their cardinality. For example, the 1 to n relation between the Cluster and Channel expresses that a cluster can contain an arbitrary number of channels. But a Channel can be member of one Cluster only. A typical situation would be a FlexRay Cluster with two redundant Cannels. Note, a CAN Cluster is described by a cluster with one channel connected by a one to one relation.
The entities are specified in more details by appropriate attributes. These attributes consist of some general ones like name, short name, and description. Others are specific to the corresponding entity. To give an example, the Cluster would have the speed attribute since this is the same for all related Channels, whereas the Channel would have attributes defining the network management, as it might be Channel specific. But both of them might have a name and a description.
FIBEX consists of a basic definition (fibex.xsd) that is used for any protocol. Protocol specific extensions are described by additional definitions (fibex4can.xsd, fibex4lin.xsd, fibex4flexray.xsd, …), see Fig. 5.

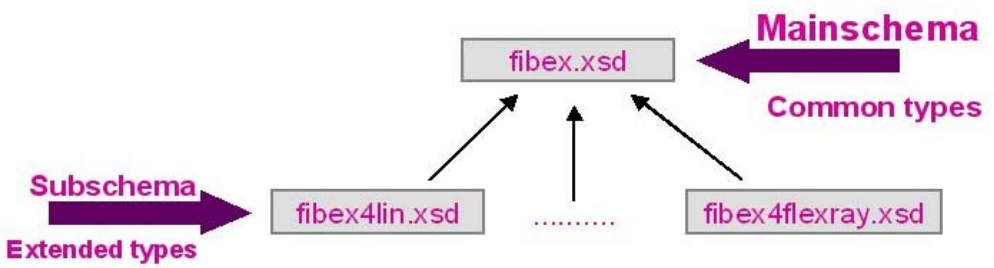

Figure 5: Protocol specific extensions in FIBEX

Manufacturer specific extensions
Almost any mid- and top-level element contains an optional <MANUFACTURER-EXTENSION> element. This acts as a container for an arbitrary sequence of elements that are not part of the FIBEX specification. Within these containers manufacturer specific extensions are possible for data that is not represented by standard FIBEX. The user should be aware that tools on the market would not support those extensions. Nevertheless, the structure of FIBEX together with the capabilities of the xml schema allows validation of FIBEX instances with or without protocol and/or manufacturer specific extensions.

The Release package contains several basic essentials.

1. The specification points out the goals, requirements, and use cases. It introduces the entities with the underlying model, and describes the relations between them.
2. The attributes are fully documented. In order to guaranty consistency between delivered essentials, the documentation is generated from the XML-Schema.
3. The XML-Schema contains the declaration of the expected XML-structure of FIBEX documents. There are several files. One for the main scheme (fibex.xsd) and one for each protocol (fibex4can.xsd, fibex4lin.xsd, ..).
4. The HTML documentation is a navigable visualization of the XML-Schema. These HTML pages are also generated from the schema.
5. The examples show how FIBEX is used for various communication technologies. The delivered XML files are described in the specification and can be used for first evaluations.

4. Usage and Examples

In general the user will not be confronted with the details of FIBEX. He just will export and import data represented in FIBEX. The benefit for him will be an easy exchange of data between tools.
However, we will give a short tutorial for those people who have a deeper interest in FIBEX, e.g. for anybody who writes software for FIBEX. For that we use a very little example. We show how to find out what an entity (a frame) is and how it is represented in one of the examples.

1. At first we look at the FIBEX specification. The Document gives a specification as follows: A frame is the smallest piece of information that is exchanged over the communication system. A frame is a datagram with a payload section of a certain length in bytes, which contains an arbitrary number of non-overlapping signals and/or multiplexers/sub frames. The send-stimuli of a frame are described in the “frame triggering” elements of a frame.

2. We analyze the location and relations in the ER diagram; see Figure 4 in the previous section. The ER diagram shows how the frame is embedded into the FIBEX structure. We see that the Frame has a relation to FrameTriggering (1 to n relation) and to Signal instance (1 to n relation) and which refers to the signal (n to 1 relation). The frame is refered to one or more FrameTriggerings. Each of them is referred to one channel. With this structure it is possible to express that one frame is transferred via several channels with different timings. With the analog reference to signal instance and signal we can express a frame that contains an arbitrary number of signals.

3. We check the attributes in the attribute table for the frame. That table shows all attributes and the detailed specification of the data type together with a textual description, see table below. The appearance specifies whether the attribute is mandatory or optional. We see that the frame has three mandatory attributes, SHORT-NAME, BYTE-LENGTH, and FRAME-TYPE.

FRAME

Attribute	Data Type	Appearance	Description
SHORT-NAME	[a-zA-Z0-9_]+	mandatory	The elements language independent denotation (usualy unique)
LONG-NAME	fx:FULL-STRING	optional	The elements full name in the language given as attribute
DESC	fx:LONG-STRING	optional	An elements description in the language and for the aspect given as attributes
BYTE-LENGTH	xs:unsignedShort	mandatory	A frames length in bytes
FRAME-TYPE	APPLICATION NM DIAG-STATE DIAG-REQUEST DIAG-RESPONSE TPL OTHER	mandatory	To distinguish application frames from certain services
MANUFACTURER-EXTENSION	fx:MANUFACTURER-FRAME-EXTENSION	optional	Reserved space providing schema extendibility to enable manufacturers to use FIBEX for their proper needs without affecting the FIBEX standard. See the derived example schema fibex4sulzer.xsd for a manufacturer speicifc extension enabling the manufacturer to validate the extended content against a proper specification.

4. In the next step we browse the HTML document to reconcile the XML schema, see Fig. 7. This document is a more graphical representation of the format. It shows the attributes as seen in step 3 and the references to other entities such as SIGNAL-INSTANCES. Please note also the grouping of some elements within COMMON-ELEMENT-DETAILS since these elements are used for many entities.

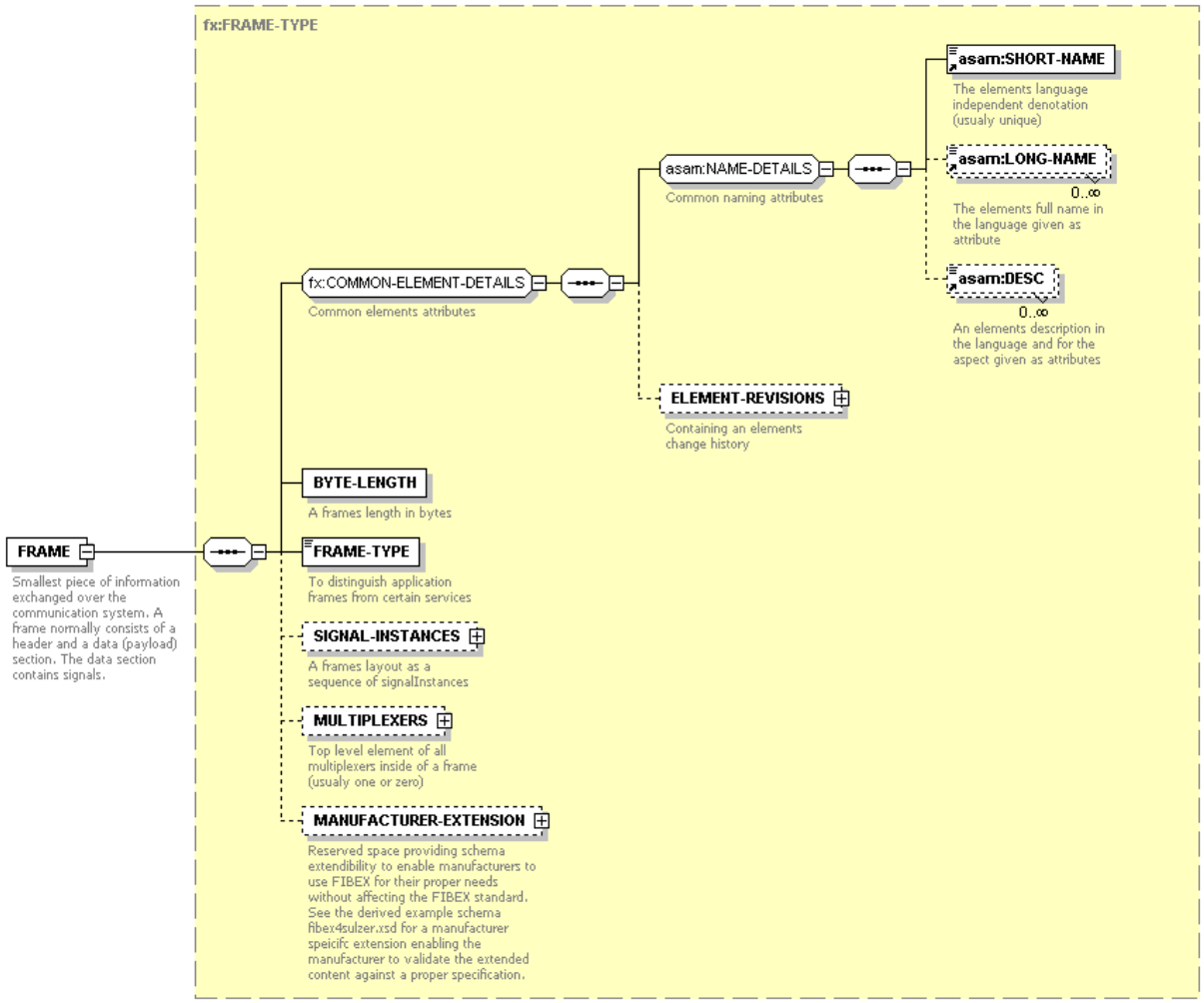

Figure 7: Section of the FIBEX structure that describes a frame.

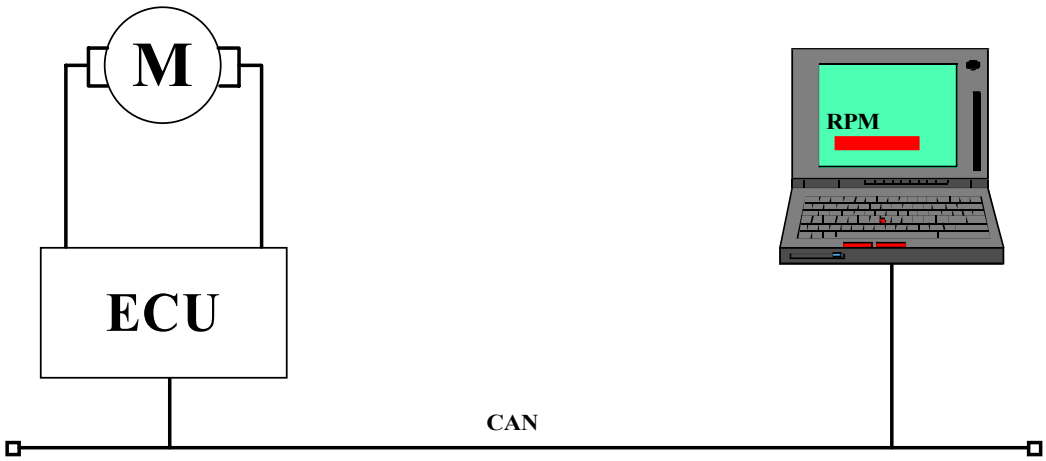

Figure 6: A simple CAN example

Frame	ID	contained signals
frmSetSpeed	0x10	sigTargetSpeed
frmSetRate	0x20	sigUpdateRate
frmReportSpeed	0x30	sigCurrentSpeed
frmRequestSpeed	0x30 (req)	

5. In the next step we look up a frame in the delivered CAN example (CANexample.xml), see Figure 6. In the frame frmSetSpeed, we recognize those attributes again. Some are not present, since they are optional and not necessary to describe that frame.

```
<FRAMES>
  <!------------------------ frmSetSpeed ------------------------>
  <FRAME xsi:type="can:FRAME-TYPE" ID="frmSetSpeed">
    <SHORT-NAME>frmSetSpeed</SHORT-NAME>
    <DESC>Sent from PC to set new speed</DESC>
    <BYTE-LENGTH>2</BYTE-LENGTH>
    <FRAME-TYPE>APPLICATION</FRAME-TYPE>
    <SIGNAL-INSTANCES>
      <SIGNAL-INSTANCE ID="instTargetSpeed">
        <BIT-POSITION>0</BIT-POSITION >
        <IS-HIGH-LOW-BYTE-ORDER>false</IS-HIGH-LOW-BYTE-ORDER>
        <SIGNAL-REF ID-REF="sigTargetSpeed"/>
      </SIGNAL-INSTANCE>
    </SIGNAL-INSTANCES>

    <can:RTR-FLAG>false</can:RTR-FLAG>

  </FRAME>

  ... further <frame> sections

</FRAMES>
```

The frame description has references to contained signals as ***<SIGNAL-INSTANCE>*** tags. The frame "frmSetSpeed" contains only one signal. It is necessary to separate the position of a signal within a frame from its description to be able to map one signal to several frames. Hence, the start of the signal (***<START>*** tag) is

specified within ***<SIGNAL-INSTANCE>*** and not directly at the signal. The “frmSetSpeed” frame also shows the usage of bus specific tags which are namespace prefixed, e.g. ***<can:RTR-FLAG>***. Here, the boolean value “false” specifies that the “remote transmission request” bit is not set for this frame. The prefix “can:” shows that this element is from the subschema fibex4can.xsd the others are from the main schema fibex.xsd.

6. In the last step we validate the example. Using freely available xml tools as described in Appendix of the FIBEX specification does this. Here, the strength of the schema is used for a thorough check of the instances including the values of attributes.

Current Projects

A variety of projects are under way within the companies that are members in the FIBEX working group. Among them are:

- BMW AG: Integration of dedicated communication tools (comprising both development and configuration) for FlexRay and LIN into a tool chain that is connected to a database that comprises general data for networking.
  The FIBEX support for the configuration of gateways is considered as one of the next steps.
- DC will use FIBEX as an internal exchange format to exchange harmonized Signal descriptions between the different communication systems.
- Robert Bosch GmbH: Usage of FIBEX as a means of modelling and parameterization for simulations yielding performance benchmarks of automotive-networked systems and topologies.
- Cadence: "Based on the information provided by FIBEX, our SysDesign simulation tool configures the virtual components of the vehicle network and provides after simulation, the metrics about the performances of the system."
- DECOMSYS GmbH is currently developing a FIBEX Import Module, which is translating FIBEX contents into the DECOMSYS tool chain internal representation.
  A prototype of this module has already been implemented for a previous version of FIBEX (0.9.3) in Q03/2003.
  DECOMSYS expects that the Import module will be available as a product in 2004.
- dSPACE GmbH: The development of FIBEX is supported by dSPACE because of its potential as a common exchange format for communication data. A first use of FIBEX is considered in the scope of a tool integration dedicated for the design and test of applications using the FlexRay protocol.
- Sulzer GmbH: XSLT converter from a company specific format into FIBEX
- Vector Informatik GmbH: Import of communication data into CANdb++ for usage in the development workflow (system simulation, generation of embedded code, system test, …). The source of the data may be an arbitrary design and/or data management system that can export the data in FIBEX format. CANdb++ can create DBC files that are widely used in the CAN area. At the moment the import is focused on CAN data but will be provided for other bus systems in the future.
  Because the FIBEX elements are assigned to several, largely independent parts (topology, communication, function, …), it is possible to describe and/or process only those aspect of the data needed to perform the different development tasks. In the case of the CANdb++ import only data describing the communication and topology of a system is used. The other parts (e.g. functions) may or may not be available in the FIBEX file. The independence of the different parts allows editing of the topology and the communication aspects of a system without influencing elements not covered by CANdb++.
  Manufacturer extensions described in an additional XML schema allow the exchange of data not covered in the basic FIBEX definition.
- The Volcano Communications Technologies (VCT) produced Network Design Tool (VNA) and its LIN only version (LNA) is now equipped with a FIBEX based interface. The FIBEX is used:
  - To import signals and their attributes where OEMs use a corporate signal database, and
  - to export complete vehicle network configurations from VNA including OEM specific attributes.
  In addition there is a development project ongoing where a FIBEX file (export/import) will be used between the VNA and a FlexRay Schedule Builder.

Furthermore, other tool suppliers have expressed ambitions and plans to support FIBEX as well.

5. Concluding Remarks

The current status of FIBEX is 1.0 with official ASAM release. With the official release it is fully open to anyone for usage. Several implementations have already shown the practicability of the format. The overwhelming good response from the involved companies leads to the expectation that FIBEX will diffuse into the landscape of tools and methods for field bus networks very quickly. FIBEX is expected to remain stable for a long time. We hope that future extensions can be implemented in a backward compatible way.
With the new standard format FIBEX it is possible to describe complex networks in a unified form. Currently, plans for extending the format to MOST (fibex4most.xsd) are in discussion. That will allow to describe complete automotive systems including MOST and gateways to MOST. Furthermore, ASAM provides other formats for the description of ECUs (ASAM MCD2, ODX), e.g. for diagnostic data description. Since the ASAM style guides have been applied for FIBEX a joint usage of these formats is assured.
Apart from these limitations, a variety of new tools and methods are supported, see use cases above. This should support the design of new tools or the improvement of existing ones. This supports the system designer by applying the improved and new methods. Finally it should be a major contribution to manage future requirements on automotive networks.

References:

[1] www.ASAM.net, ASAM MCD-2-FBX, FIBEX Expert Group
[2] ww.lin-subbus.de
[3] www.FlexRay.com
[4] www.msr-wg.de
[5] www.mostnet.de