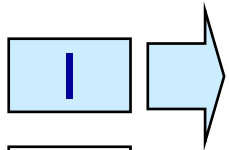


The FIBEX Standard and FIBEX Checker

Early validation of the Board-Net Database
in distributed development processes

Open Technology Forum at Testing Expo Europe 2007

Overview



Presentation of TÜV NORD IFM



The Fibex data exchange format



Motivation for the Checker Tool

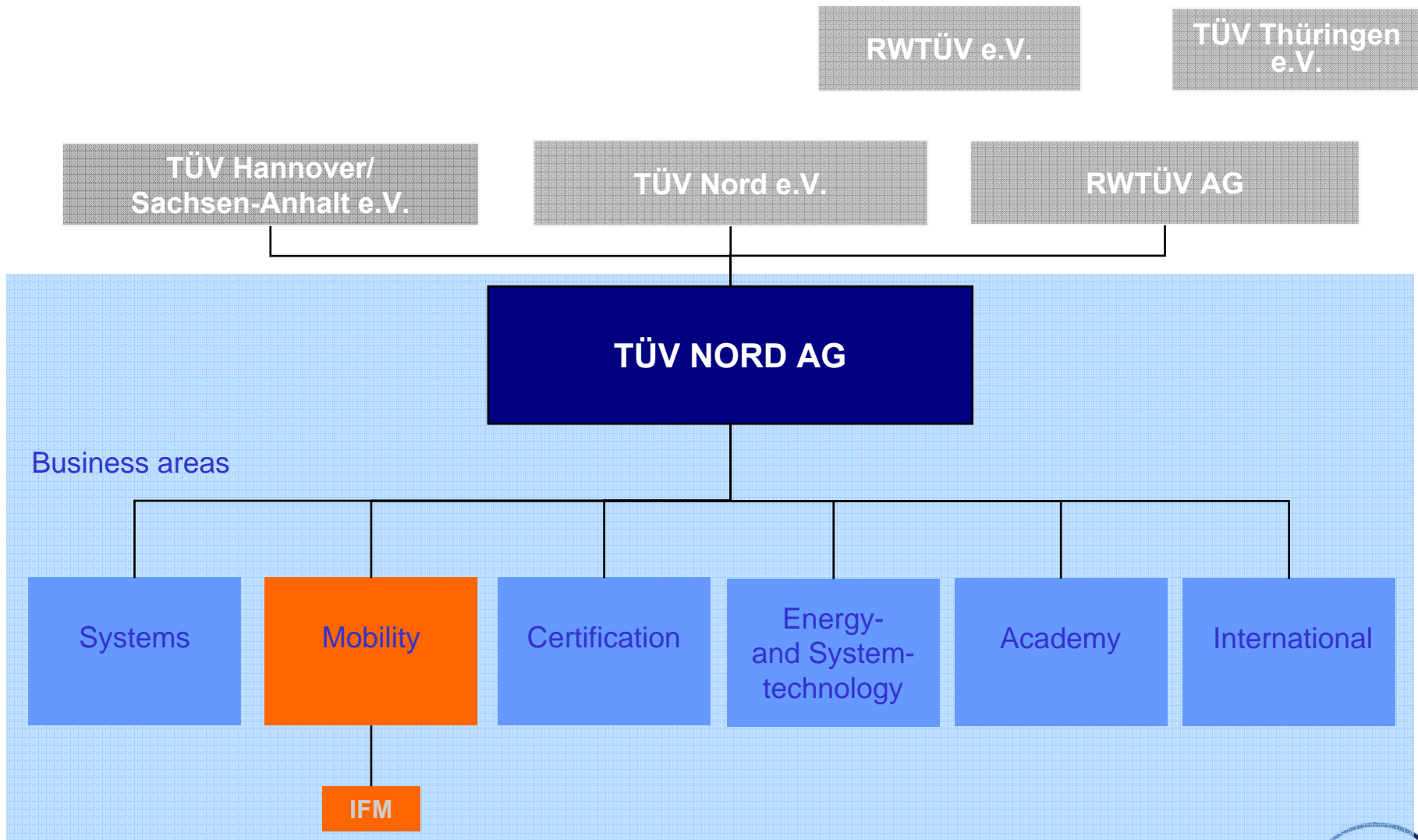


Overview of the Checker Tool



Summary

The structure of the TÜV NORD Gruppe



TÜV Nord Group



750 Mio. €



7.000

Institute for Vehicle Technology and Mobility (IFM)



13 Mio. €



130

Division Electronics & IT



1,2 Mio. €



15

IFM – Institute for Vehicle Technology and Mobility

+ Mechanical, mechatronic Functionalities in the Vehicle

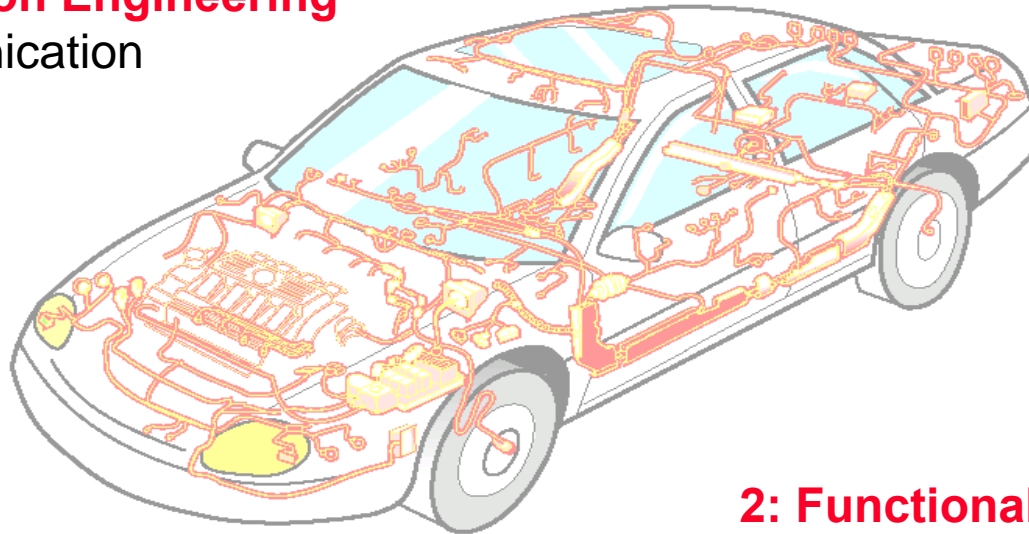
- Brake systems
- Emission management
- Engine, Powertrain
- Tyres and other Components

+ Electronics & Car-IT

- Communication Engineering
- Functional Safety
- Research & Development
- Validation Tools and Test Systems
- Training, Workshops, Seminars

1: Communication Engineering

Databus communication
in vehicles
FlexRay™,
TTP,
CAN



2: Functional Safety

Technical support, consulting
with regard to the standard IEC 61508

3: Validation Tools and Test Systems

FlexRay™ Consortium: Protocol Conformance Test
AUTOSAR: Development Member (Test Specifications and Systems)
ASAM: FIBEX Checker

Contents

I

Presentation of TÜV NORD IFM

II



The Fibex data exchange format

III

Motivation for the Checker Tool

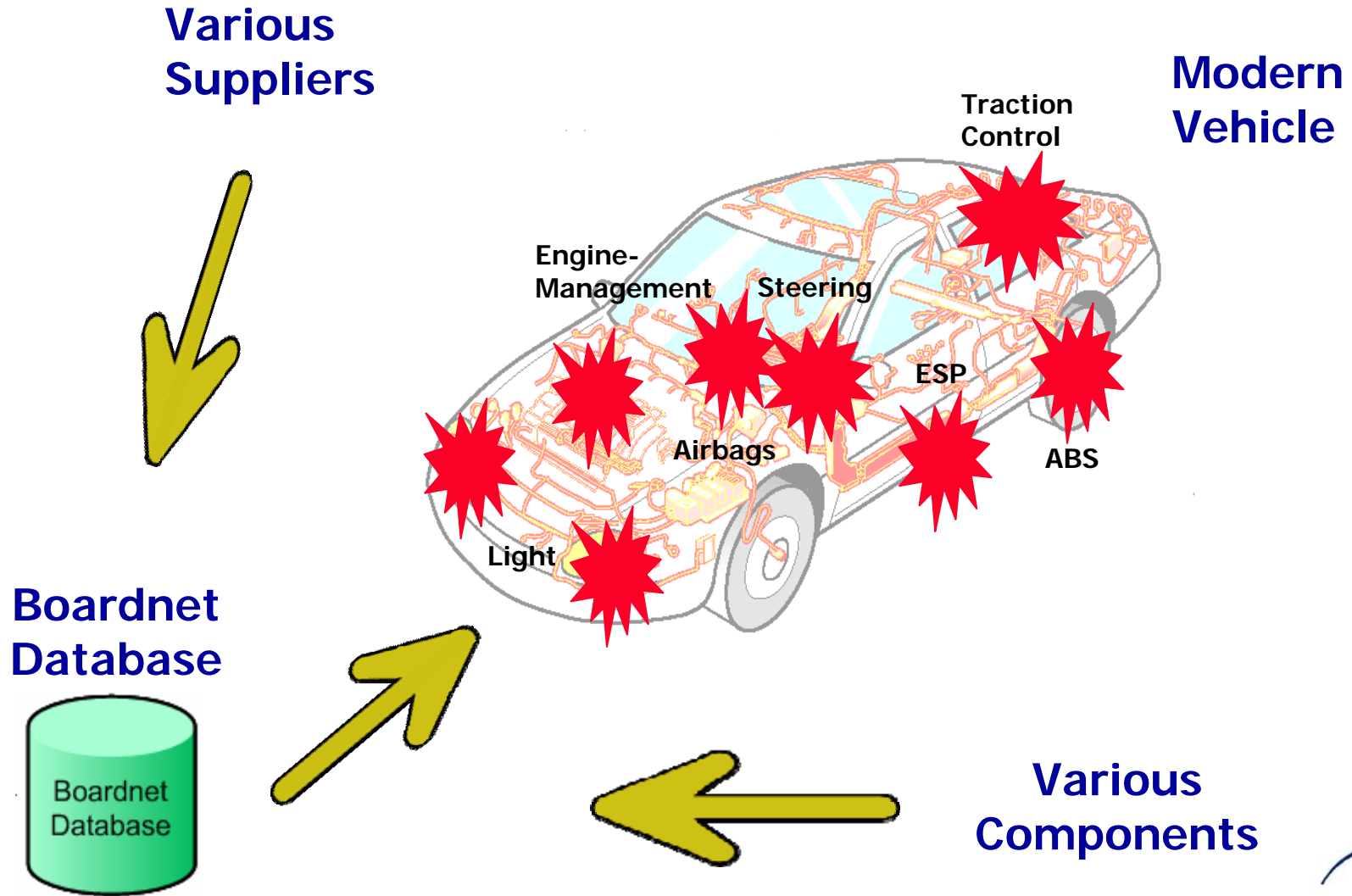
IV

Overview of the Checker Tool

V

Summary

Introduction



Introduction

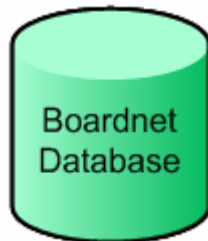
Various
Suppliers

Modern
Vehicle

Problems in the Development Process

- Different Tools
- Different Design Philosophy
- Consistent Boardnet Database ?

Boardnet
Database



Solution

FIBEX provides a standardised way for data exchange.

Various
Components

FIBEX facts

- **Meaning:** Fibex = **F**ield **B**us **E**xchange Format (ASAM e.V.)

- **Purpose:** XML exchange format for use with tools that deal with message-oriented bus communication systems

- **Multi-Bus:** FlexRay™, CAN, Most, Lin

- **Tools:**
 - OEM Toolchains for series implementation (e.g. BMW, Daimler, . . .)
 - Tool provider (e.g. Decomsys, Vector, Softing, dSpace, Sulzer, . . .)

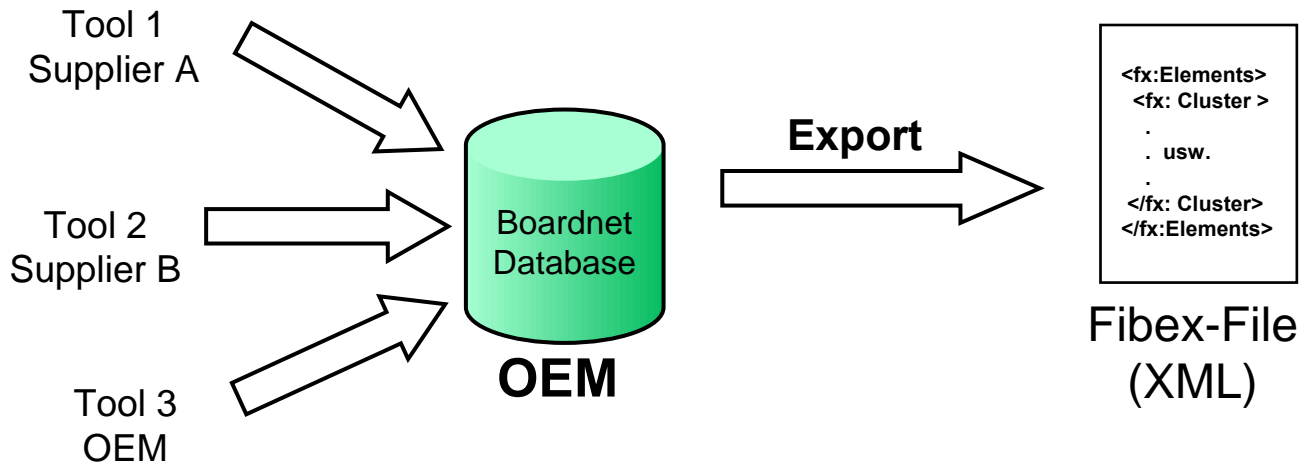
FIBEX Use Cases

OEM:

- Integration (Data of numerous suppliers)
- Data-Management (Supplier ↔ OEM)

OEM & Supplier:

- Simulation
- Scheduling
- Generation of Driver Code
- Gateway Design
- Bus Analysis



FIBEX Design Goals

FIBEX covers different aspects of the communication system:

System topology

- logical layout of the system

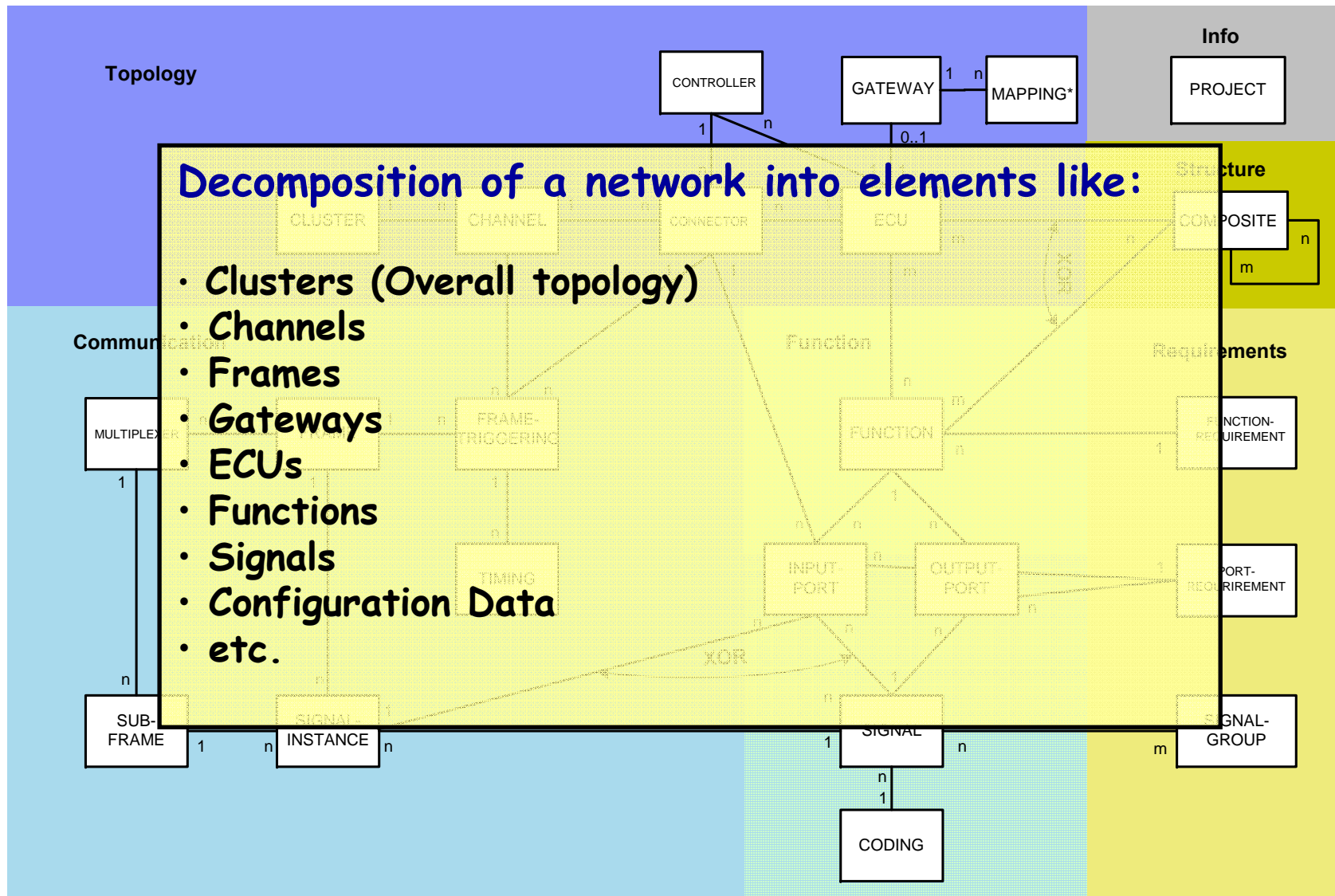
Communication properties

- exchange of frames
- the structure of frames
- timing characteristics

Functional network (optional)

- functions
- signals

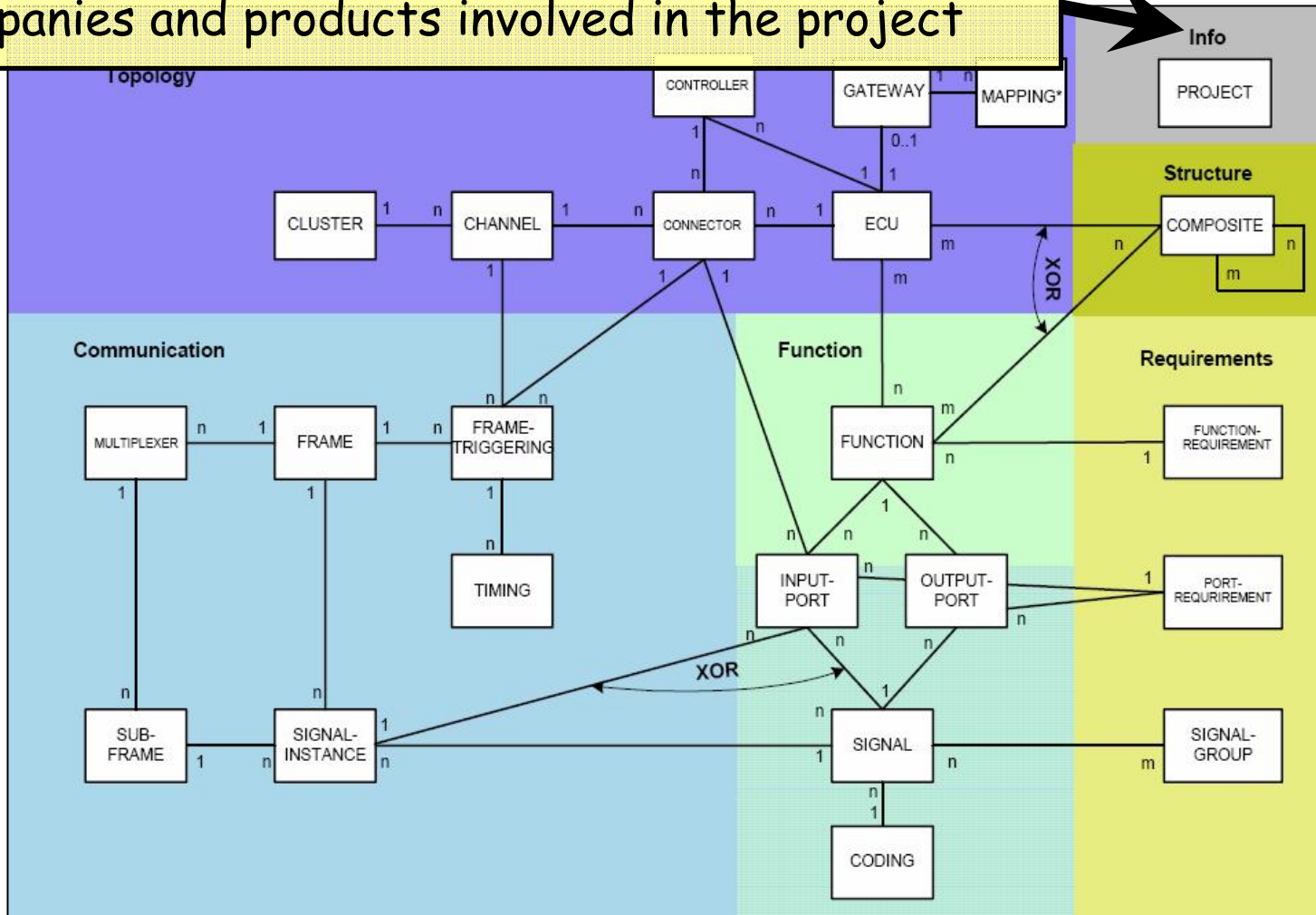
FIBEX Entities



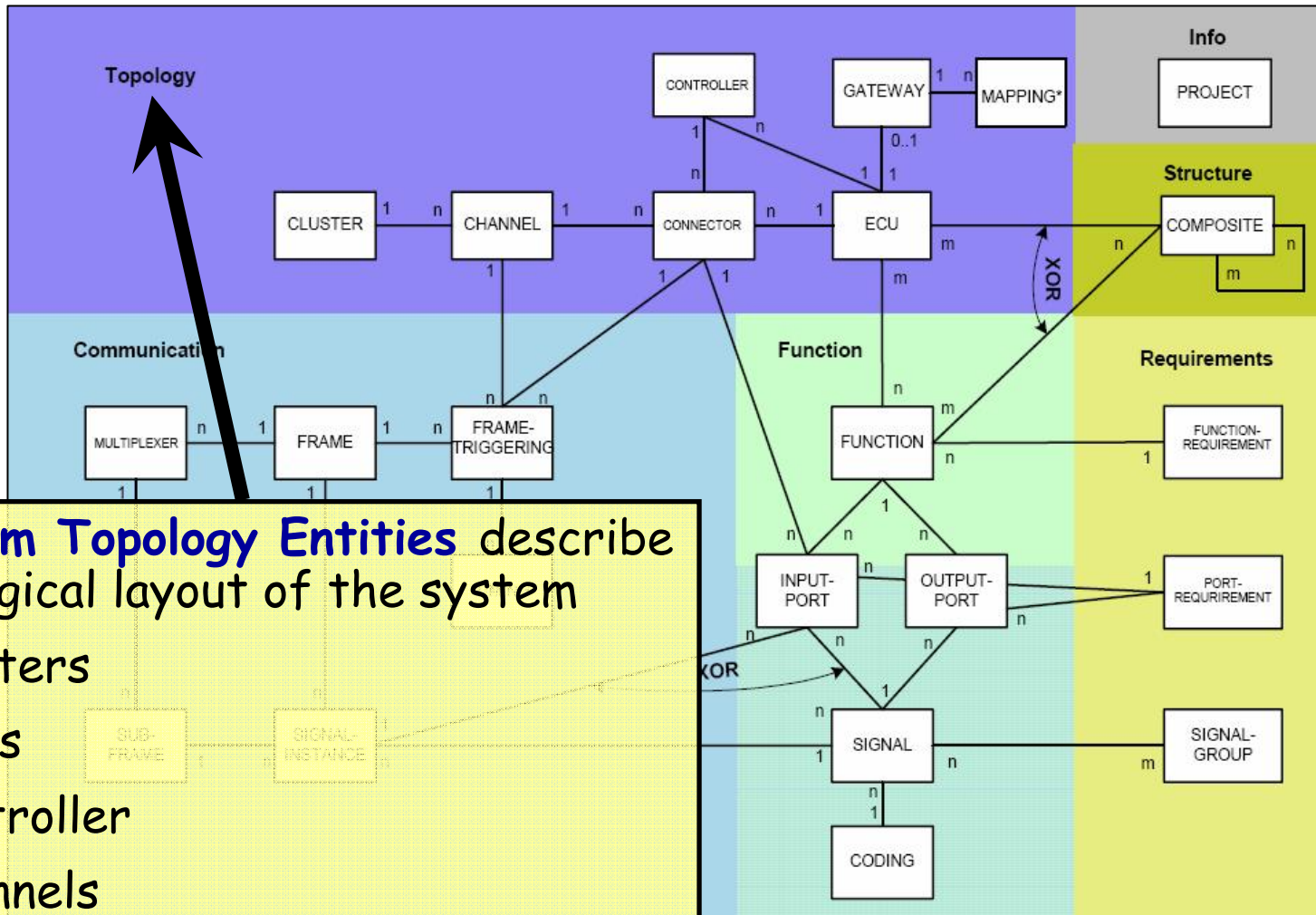
FIBEX Entities

Information Entities add Metadata to the project

➤ Companies and products involved in the project



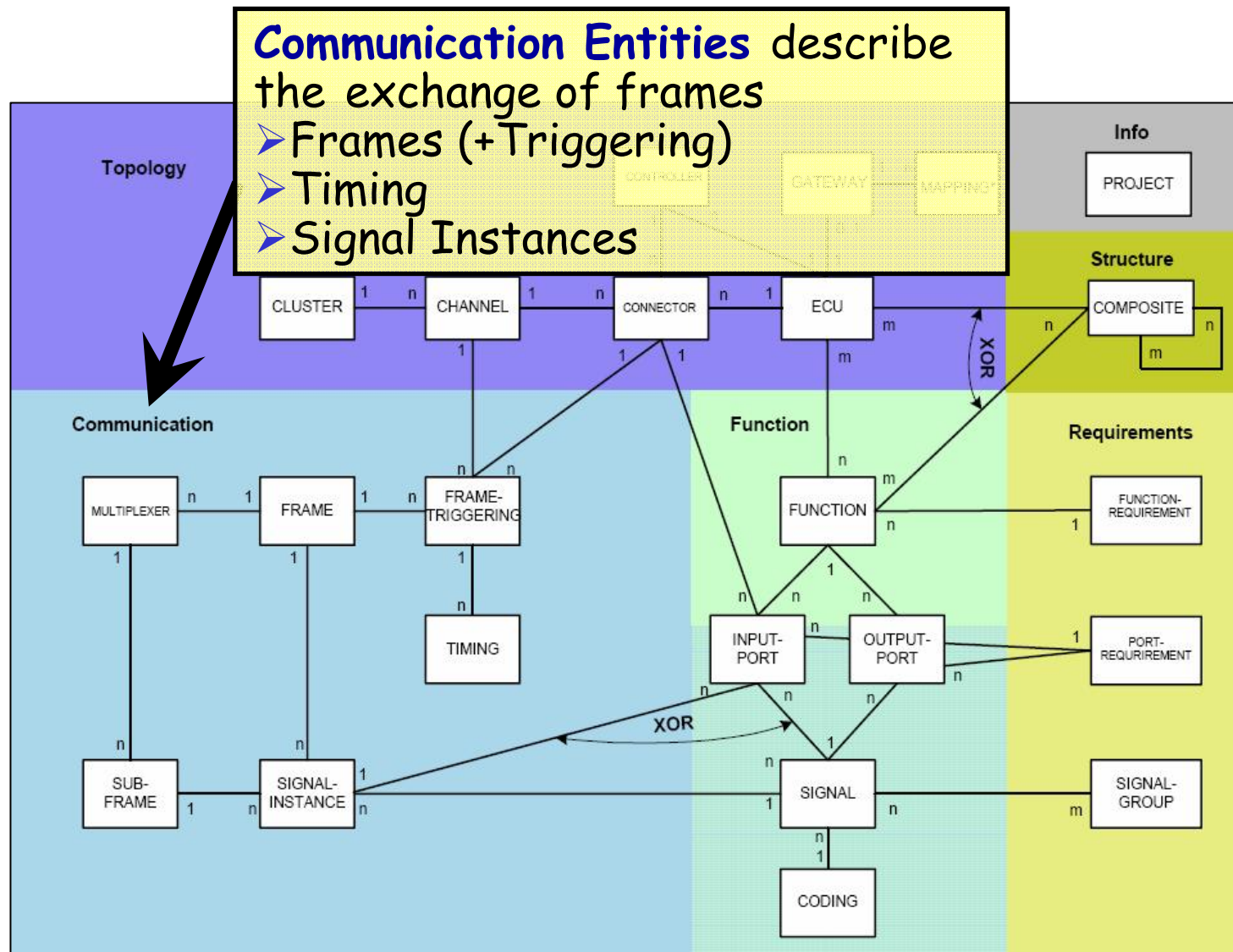
FIBEX Entities



System Topology Entities describe the logical layout of the system

- Clusters
- ECUs
- Controller
- Channels

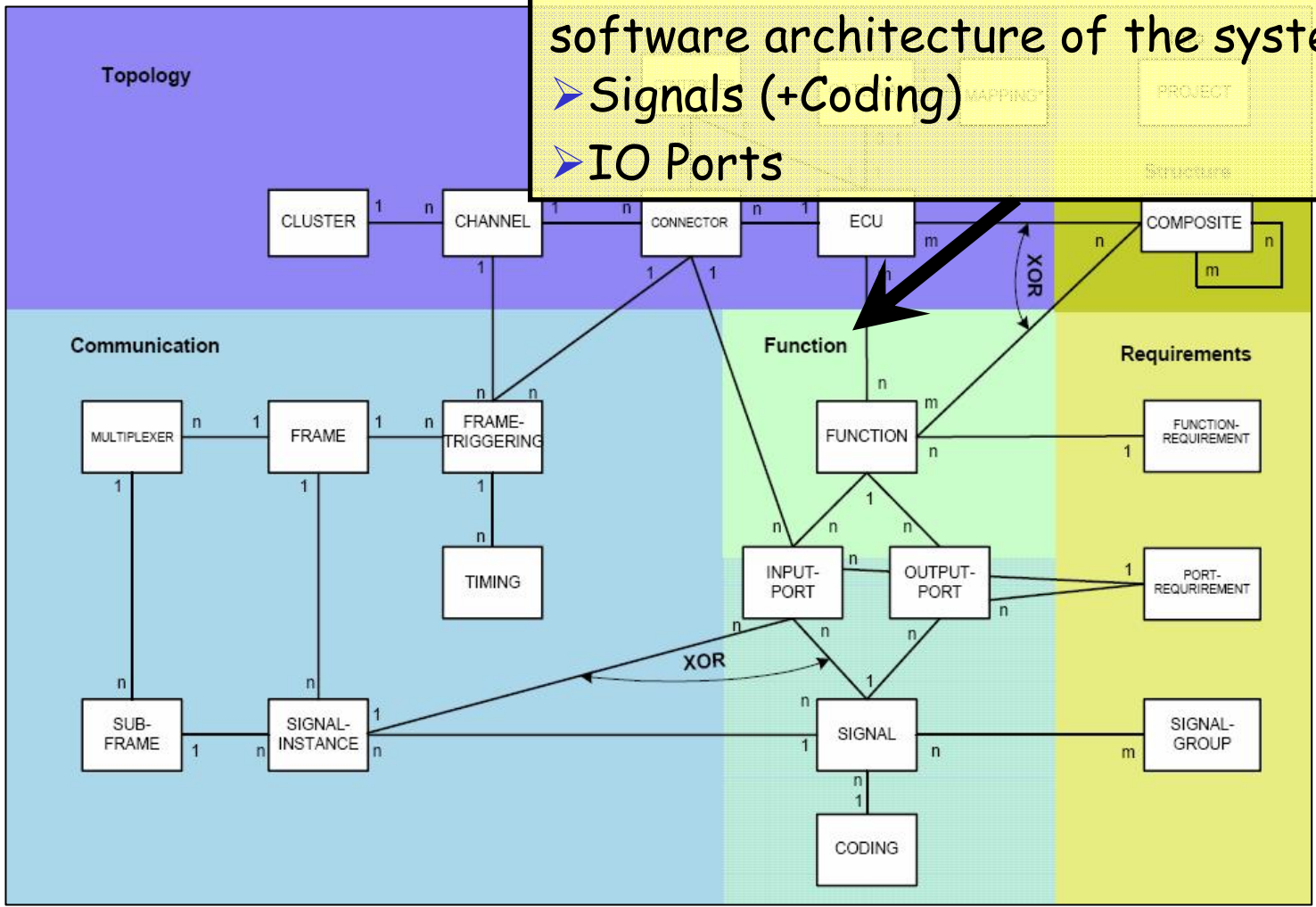
FIBEX Entities



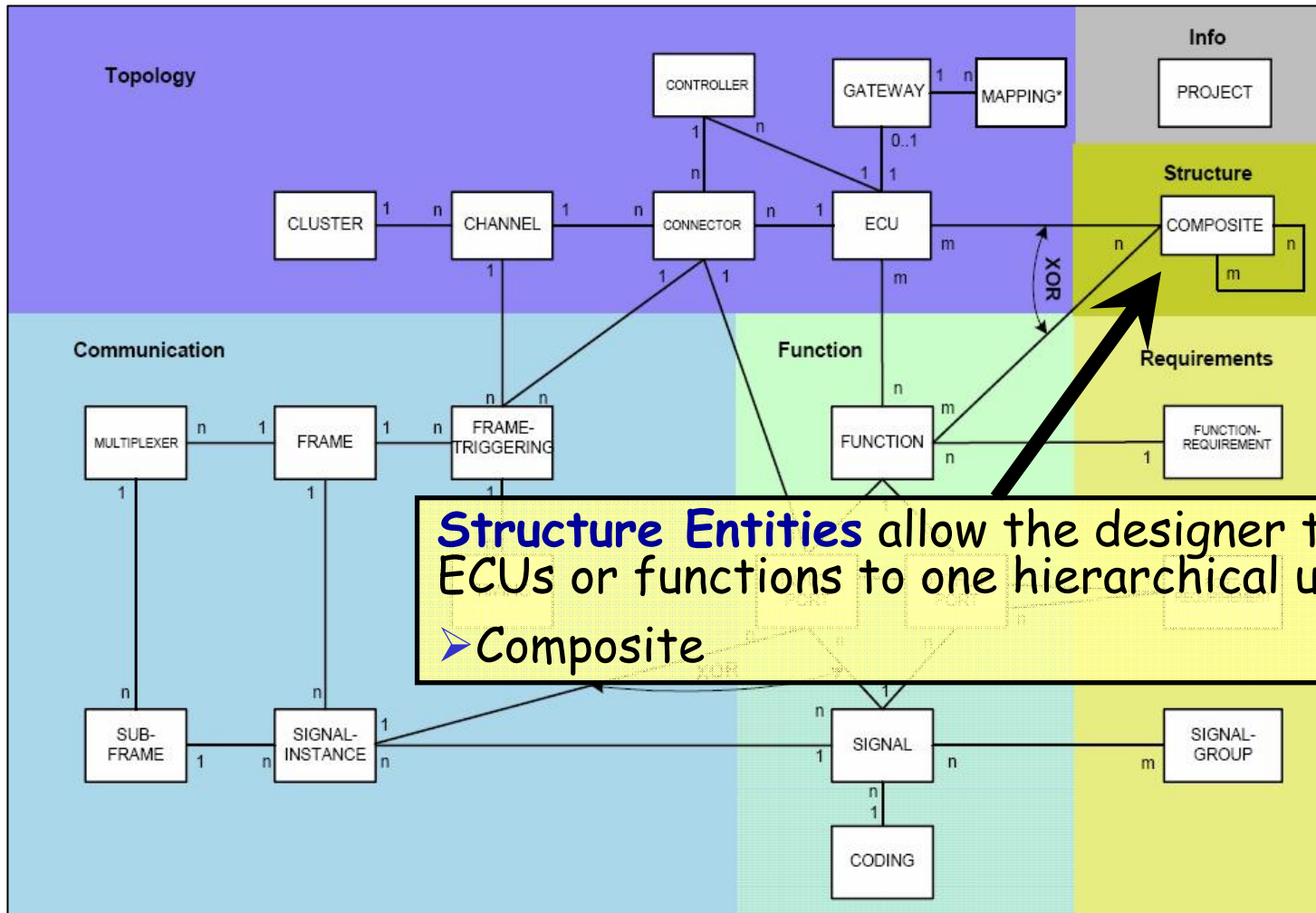
FIBEX Entities

Functional Level Entities describes the software architecture of the system.

- Signals (+Coding)
- IO Ports

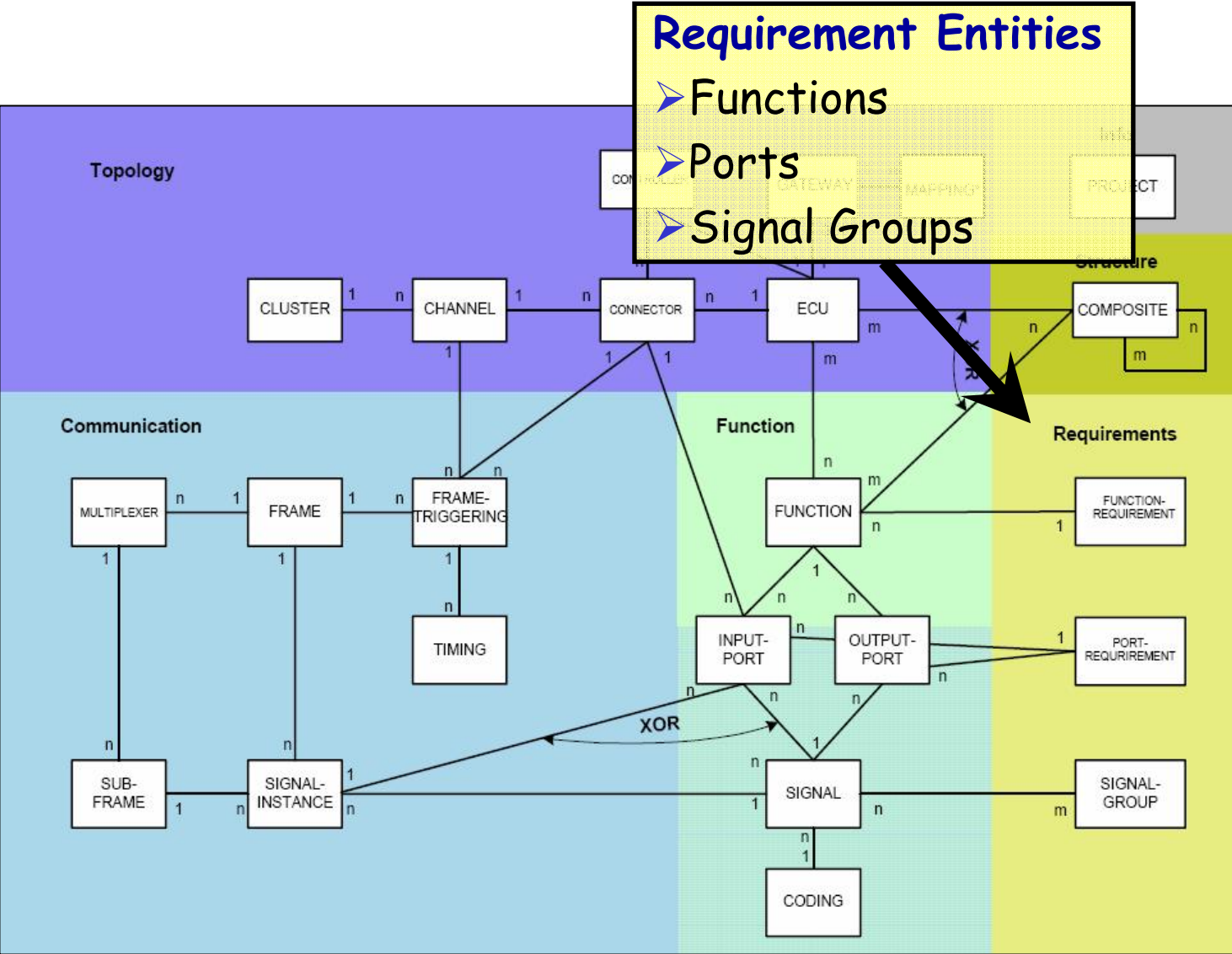


FIBEX Entities



Structure Entities allow the designer to group ECUs or functions to one hierarchical unit
 ➤ Composite

FIBEX Entities



FIBEX Extensions

FIBEX allows two types of extensions to be used:


Manufacturer Extensions:

- add new functionality to FIBEX.

Platform Specific Extensions

- allow the designer to extend the generic basic types by platform specific needs

Overview

- I Presentation of TÜV NORD IFM
- II The Fibex data exchange format
- III**  **Motivation for the Checker Tool**
- IV Overview of the Checker Tool
- V Summary

Schema validation + Rule based validations

➤ **Schema validation:** Based on XML schema files for Fibex.

The Checker performs a standard XML schema validation as a basic check:

- Correct order of elements.
- Range of values.
- Existence of referenced elements.
- etc.

➤ **Rule based validations:**

Additional validation rules can easily be defined by the user for e.g.:

- Verification of relations among communication parameters.
- Checking the logical structure of the target bus system.
- Enhanced validation of relations among elements.
- Enhanced validation of reference targets.

**Checks for numerous mathematical relations and configuration constraints that do exist with respect to the communication parameters of a bus system. s
v connected to the channel the signal is being transmitted on.**

Example for a rule (A)

Rule A:

FR-Cycle:

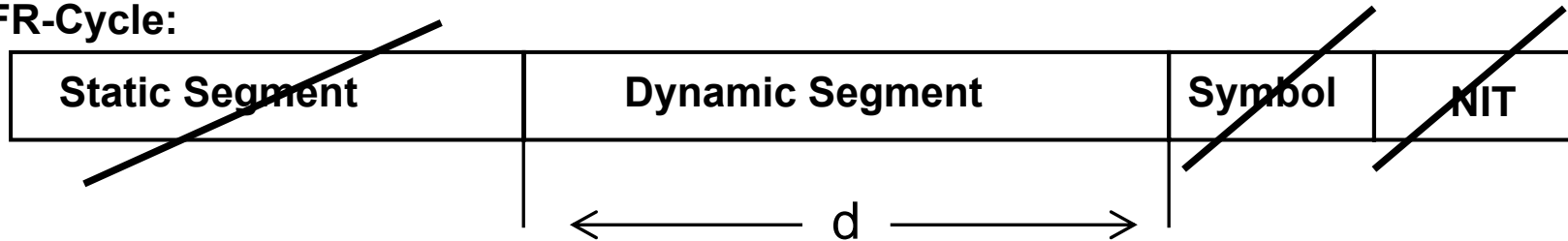
Static Segment	Dynamic Segment	Symbol	NIT
----------------	-----------------	--------	-----

Setting: A FlexRay™ cycle consists of the segments shown above. The length of the segments and the cycle is defined by the FR-parameters, which are contained in the Fibex file.

Example for a rule (A)

Rule A:

FR-Cycle:



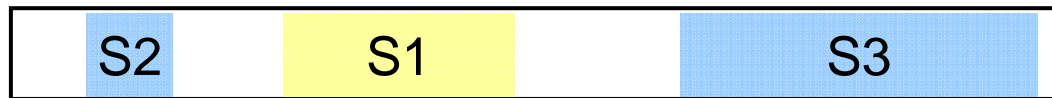
$d \leq g\text{NumberOfMinislots} ?$

Rule: If the length of the static segment, symbol window and NIT are subtracted from the cycle length, the length of the dynamic segment can be calculated. The result **d** can then be compared to the configured parameter **gNumberOfMinislots**.

Example for a rule (B)

Rule B:

Frame (payload):

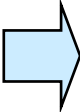


Setting: The payload section of a frame containing three signals is defined in a Fibex-file. The signal length and their respective positions inside the frame are defined as well.

Problem: A possible error could emerge when signals overlap.

The configuration data for frames, signals, their positions and length is spread over different parts of the Fibex file.

Overview

- I Presentation of TÜV NORD IFM
- II The Fibex data exchange format
- III Motivation for the Checker Tool
- IV**  **Overview of the Checker Tool**
- V Summary

The "Checker-Tool"


- **Java application:** ⇒ platform independency
- **Plug-in interface:** ⇒ Quick and easy implementation of rules
- **Basic operation:** ⇒ No time consuming orientation phase
- **Error recognition:** ⇒ Can be used at any stage of development to validate Fibex files and recognise errors as early as possible
- **Operation:**
 - **Command line:** ⇒ Automated testing, part of a tool chain.
 - **GUI:** ⇒ Comfortable operation

Screenshot: Selection of Validation Rules

The screenshot displays the 'Fibex4FlexRay Checker - Prototype 3' application. The main window features a 'Rules' tab with a table of validation rules. On the left, a 'Selector' panel allows filtering rules by bus type (All, Flexray, fibex). An 'Open' dialog box is overlaid on the right, showing file selection options for 'Fibex File' (C:\Example.xml) and 'Schema File' (C:\FIBEX.xsd). The dialog also includes checkboxes for 'Validate' and 'Save Report', and options for 'Filename' (C:\Report.txt), 'Format' (CSV files), and 'Loglevel' (warning).

Use	Bus	Category	Rule No.	Ver.	Author	Description
<input checked="" type="checkbox"/>	fibex	buscheck	Rule137	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	buscheck	Rule138	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	fibex	Rule141	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	fibex	Rule150	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	fibex	Rule156	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	fibex	Rule157	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	fibex	Rule300	1.0	IFT-Autoelektronik	
<input checked="" type="checkbox"/>	fibex	fibex	Rule4	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	nice2have	Rule143	1.0	IFT-Autoelektronik	
<input type="checkbox"/>	fibex	nice2have	Rule147	1.0	IFT-Autoelektronik	
<input checked="" type="checkbox"/>	fibex	nice2have	Rule168	1.0	IFT-Autoelektronik	Checks if all elements ho:COMPU-NUMERATOR contain two values.
<input type="checkbox"/>	fibex	nice2have	Rule169	1.0	IFT-Autoelektronik	Checks if all elements ho:COMPU-DENOMINATOR exist and their value is '1'
<input type="checkbox"/>	fibex	nice2have	Rule302	1.0	IFT-Autoelektronik	Checks if name attribute ho:LONG-NAME exists.
<input type="checkbox"/>	flexray	buscheck	Rule1	1.0	IFT-Autoelektronik	Checks if all flexray clusters run at 10, 5 or 2.5 MBit/s.
<input type="checkbox"/>	flexray	buscheck	Rule106	1.0	IFT-Autoelektronik	Checks constraint 14 ((FlexRay2.0], page 212)
<input type="checkbox"/>	flexray	buscheck	Rule111	1.0	IFT-Autoelektronik	Checks formula 5 ((FlexRay2.0], page 208).
<input type="checkbox"/>	flexray	buscheck	Rule122	1.0	IFT-Autoelektronik	Checks constraint 36 ((FlexRay2.0], page 223).
<input type="checkbox"/>	flexray	buscheck	Rule124	1.0	IFT-Autoelektronik	Checks constraint 12 ((FlexRay2.0], page 212).
<input checked="" type="checkbox"/>	flexray	buscheck	Rule126	1.0	IFT-Autoelektronik	Checks if gMaxWithoutClockCorrectionPassive is smaller than gMaxWithoutC
<input checked="" type="checkbox"/>	flexray	buscheck	Rule127	1.0	IFT-Autoelektronik	Checks constraint 19 ((FlexRay2.0], page 217).
<input type="checkbox"/>	flexray	buscheck	Rule140	1.0	IFT-Autoelektronik	Checks if all frames sent in the static segment do not exceed their allowed p.
<input type="checkbox"/>	ule160			1.0	IFT-Autoelektronik	Checks for multiple channels A or B per cluster.
<input type="checkbox"/>	ule2			1.0	IFT-Autoelektronik	Checks if FlexRay clusters have high-low bit order.
<input type="checkbox"/>	ule200			1.0	IFT-Autoelektronik	Checks if SPEED and SAMPLE-CLOCK-PERIOD are configured correctly.
<input type="checkbox"/>	ule201			1.0	IFT-Autoelektronik	Checks if every cluster does not exceed the maximum number of sync nodes
<input type="checkbox"/>	ule3			1.0	IFT-Autoelektronik	Checks if flexray clusters have either one or two channels.
<input type="checkbox"/>	ule305			0.1	IFT-Autoelektronik	Checks for frame timing collisions.
<input type="checkbox"/>	ule5			1.0	IFT-Autoelektronik	Checks constraint 13 ((FlexRay2.0], page 212)
<input type="checkbox"/>	ule9			1.0	IFT-Autoelektronik	Checks if all flexray channels run at 10MBit/s.
<input type="checkbox"/>	ule100			1.0	IFT-Autoelektronik	Checks for properly setup flexray channel types.
<input type="checkbox"/>	ule144			1.0	IFT-Autoelektronik	Checks if signals containing a default value are sent over a FlexRay channel.

Fibex4FlexRay Checker
 Prototype 3



TUV NORD
 Institut für Fahrzeugtechnik und Mobilität
 Abteilung Elektronik & IT

OK

Screenshot: Validation Results and Help System

The screenshot displays the 'Fibex4FlexRay Checker - Prototyping' application window. The main window is divided into several sections:

- Menu:** File, Help
- Buttons:** Rules, Report Log, Runtime Log
- Loglevel:** warning
- Table:** A table showing validation results with columns for Rule No., Level, Line No., and a description of the error.
- Status Bar:** Fibex Checker finished. (elapsed 11s)

An 'Online Help' window is overlaid on the main application, showing the 'Program Usage' section. The help window includes a 'Select a help topic' sidebar with buttons for 'Program Usage', 'Menus', 'Tabs', and 'Further Information'. The main content area of the help window contains the following text:

Program Usage

The Fibex4Flexray-Checker-Prototype is a tool that can be used to check Fibex files. The checks that can be performed are a) XML-schema validation and b) rule checks. The rules do not only check the syntax and structure of the file, but additionally perform a validation of values assigned to the Fibex elements and other logical, bus-specific tests. The following guide is supposed to give a brief introduction to the program and a step-by-step description of the settings required to perform the checks and view the results. The detailed description of the [menus](#) and [tabs](#) can be found in the respective chapters of this help.

Step by step guide

1. On program start-up the [Rules-Tab](#) is displayed. To specify files for validation and report logging open the [Open-dialogue](#) by clicking on *File* and selecting *Open*.

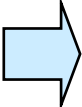
Now a Fibex file can be selected. This is the file containing the Fibex data that has to be checked. If a schema validation is needed it can be activated by checking the *Validate* checkbox and selecting an output file. The validation can be disabled by unchecking the checkbox, it is not necessary to erase the path and filename in the corresponding text field. If a report log-file is required it can be selected in the *Report Logging*-field. The *Save Report* checkbox has to be checked and an output file has to be chosen. The file-format and log-level can be set in the *Format*-and *Loglevel* pulldown menus respectively. Available options for the log-level are *debug*, *info*, *warning*, *error* and *fatal*. If, for example, the log-level is set to *error* only messages with a level of *error* and *fatal* will be logged. When all desired options are set the [Open-dialogue](#) can be closed.

2. If no rule checking is required the schema validation can be started by clicking the *Run* button in the [Rules Tab](#), otherwise more settings can be made in this tab:

The table in the main window shows the following validation results:

Rule No.	Level	Line No.	Description
XMLschema	error	9027	
XMLschema	error	9072	
XMLschema	error	9113	
XMLschema	error	9339	
XMLschema	error	9385	
XMLschema	error	9426	
XMLschema	error	10011	
XMLschema	error	10055	
XMLschema	error	10148	
XMLschema	error	10195	
XMLschema	error	10233	
XMLschema	error	10272	
XMLschema	error	10310	
XMLschema	error	10349	
XMLschema	error	10388	Col. 29:cvc-complex-type.2.4.a: Invalid content was found starting with element 'flexray:STARTUP-NODE'. One of '{"http://www.asam.net/xml":LONG-NAME,
XMLschema	error	10427	Col. 29:cvc-complex-type.2.4.a: Invalid content was found starting with element 'flexray:STARTUP-NODE'. One of '{"http://www.asam.net/xml":LONG-NAME,
Rule137	error		Frame message488 is too small. The bit position of a signal exceeds the frames byte length.
Rule168	warning		The Element ho:COMPU-NUMERATOR in coding codesignal28640 should contain two values. ([Fib1.1], page 70)
Rule168	warning		The Element ho:COMPU-NUMERATOR in coding codesignal28919 should contain two values. ([Fib1.1], page 70)
Rule168	warning		The Element ho:COMPU-NUMERATOR in coding codesignal28419 should contain two values. ([Fib1.1], page 70)
Rule168	warning		The Element ho:COMPU-NUMERATOR in coding codesignal28420 should contain two values. ([Fib1.1], page 70)
Rule168	warning		The Element ho:COMPU-NUMERATOR in coding codesignal28820 should contain two values. ([Fib1.1], page 70)
Rule168	warning		The Element ho:COMPU-NUMERATOR in coding codesignal28821 should contain two values. ([Fib1.1], page 70)
Rule168	warning		The Element ho:COMPU-NUMERATOR in coding codesignal28445 should contain two values. ([Fib1.1], page 70)
Rule127	error		Violation of constraint 19 of the FlexRay-specification 2.0 (Constraints validate FlexRay-parameters). The configured value for flexray:NUMBER-OF-MINISLOT:
Rule5	warning		Element flexray:MACRO-PER-CYCLE in CLUSTER cluster4flexraybus1064 is not ideally configured according to constraint 13 of the FlexRay-specification 2.0. T
Rule5	error		Violation of constraint 13 of the FlexRay-specification 2.0 (Constraints validate FlexRay-parameters). The configured value for flexray:MACRO-PER-CYCLE (=
Rule100	warning		A FlexRay-channel should be of type xsi:type="flexray:CHANNEL-TYPE". Channel flexraybus1064 is not configured accordingly.

Overview

- I Presentation of TÜV NORD IFM
- II The Fibex data exchange format
- III Motivation for the Checker Tool
- IV Overview of the Checker Tool
- V**  **Summary**

Summary

➤ Problem

- Fibex files can contain a multitude of errors that cannot be found manually.

➤ Validation and rules checking

- The Checker-Tool can be used to perform schema validation as well as execution of user defined rules.

➤ Plug-in interface

- The rules are implemented as plug-ins, so no deeper knowledge of the checker is needed.
⇒ fast and easy implementation of new rules

➤ Automated testing

- The checker can be run automatically using the command line control.
⇒ quick and standardised validation of Fibex files.

➤ Example of use

- Support of the first FlexRay™ series implementation at BMW.



Thank you for your interest

TÜV NORD

Institut für Fahrzeugtechnik & Mobilität
Fachgebiet „Electronics & IT“

Wolfgang Mickisch

Phone + 49 201 – 825 4101
Fax + 49 201 – 825 4204
e-mail wmickisch@tuev-nord.de

Kurzvorstellung „Analyse“

