Institut für
Fahrzeugtechnik & Mobilität

Electronics & IT

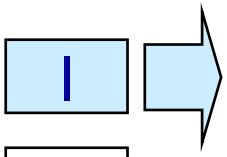
The FIBEX Standard and FIBEX Checker

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

Open Technology Forum at Testing Expo Europe 2007

TÜV NORD

Overview



I **Presentation of TÜV NORD IFM**

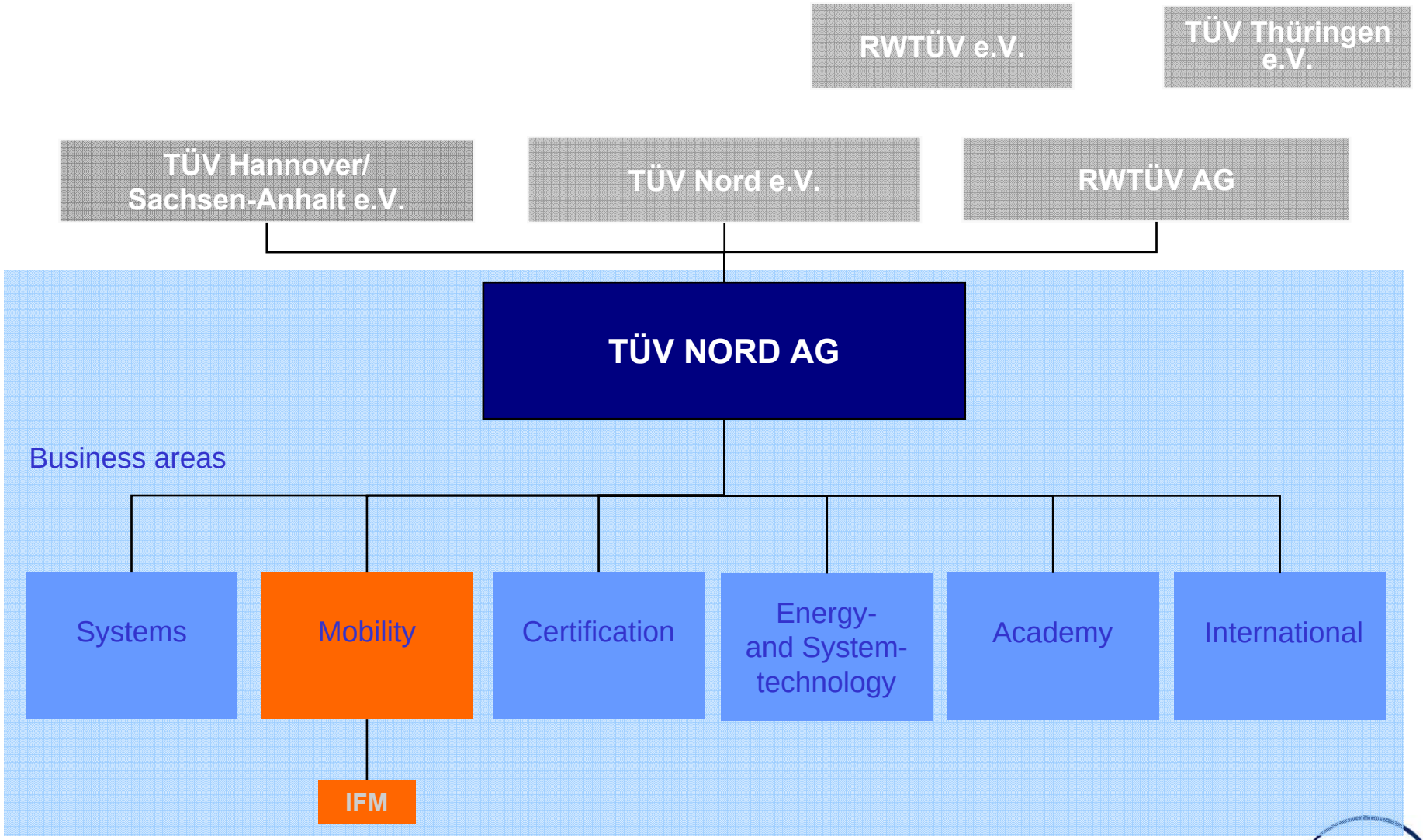
II The Fibex data exchange format

III Motivation for the Checker Tool

IV Overview of the Checker Tool

V Summary

The 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

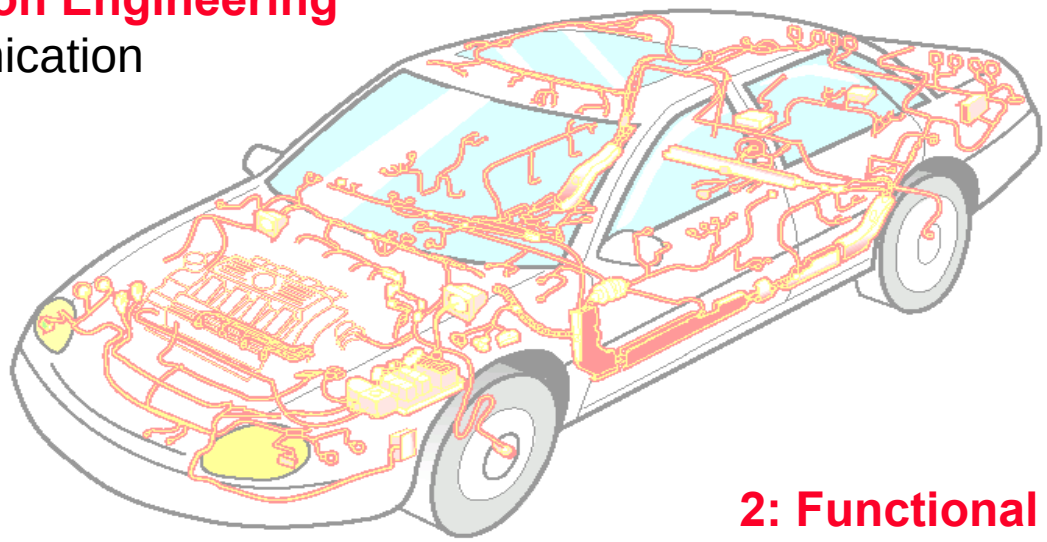
IFM – Institute for Vehicle Technology and Mobility

+ **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**

Units of the TÜV NORD IFM Department „Electronics & IT“

1: Communication Engineering

Databus communication
in vehicles
FlexRay™,
TTP,
CAN



2: Functional Safety

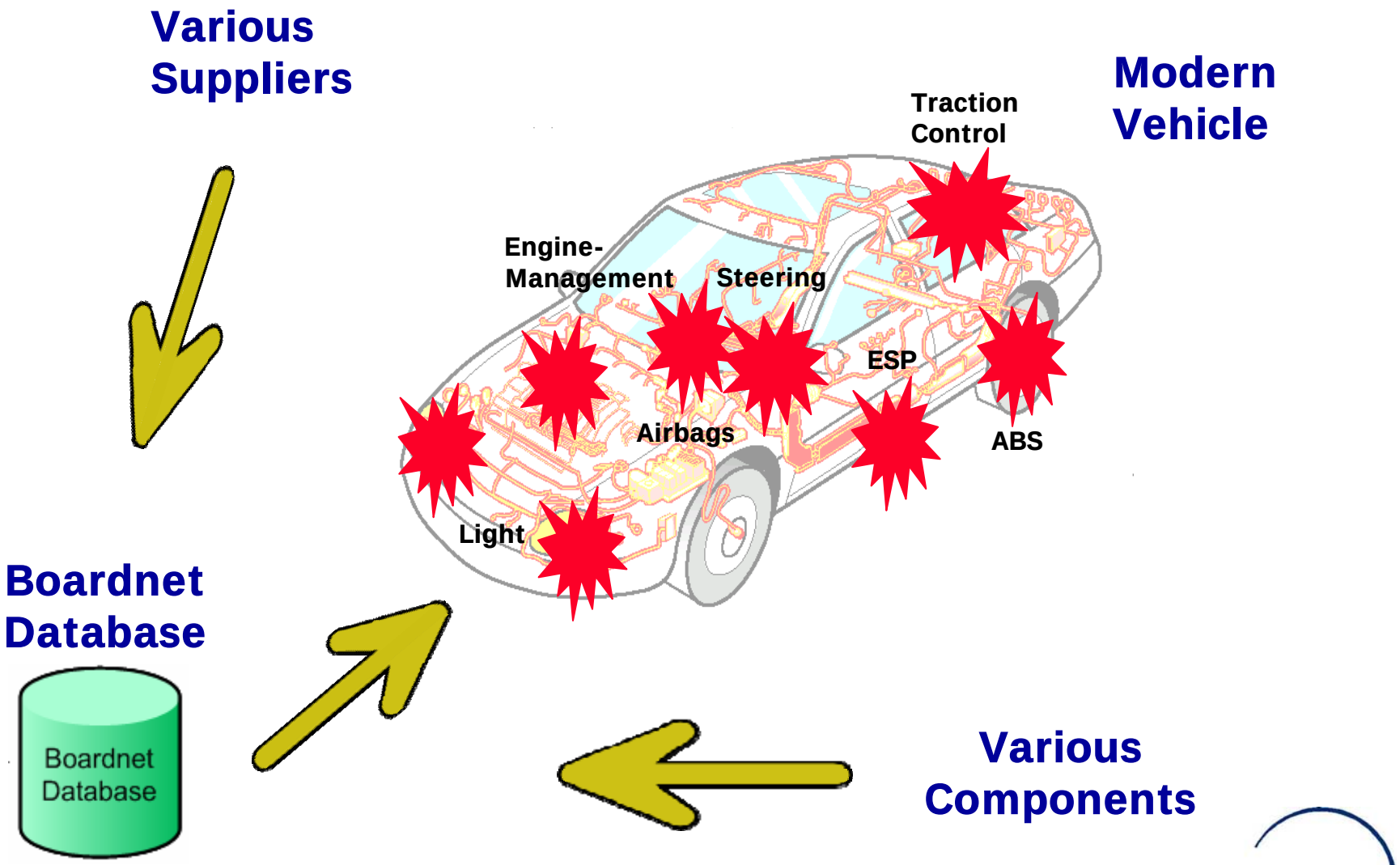
Technical support, consulting
with regard to the standard IEC 61508

3: Validation Tools and Test Systems

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

Contents

Introduction



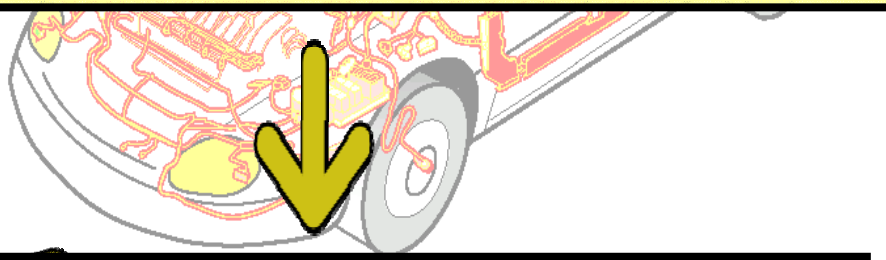
Introduction

Various Suppliers

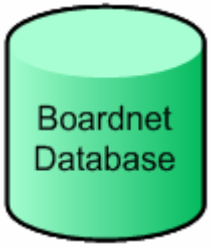
Modern Vehicle

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 = **Fi**eld **B**us **Ex**change 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, . . .)

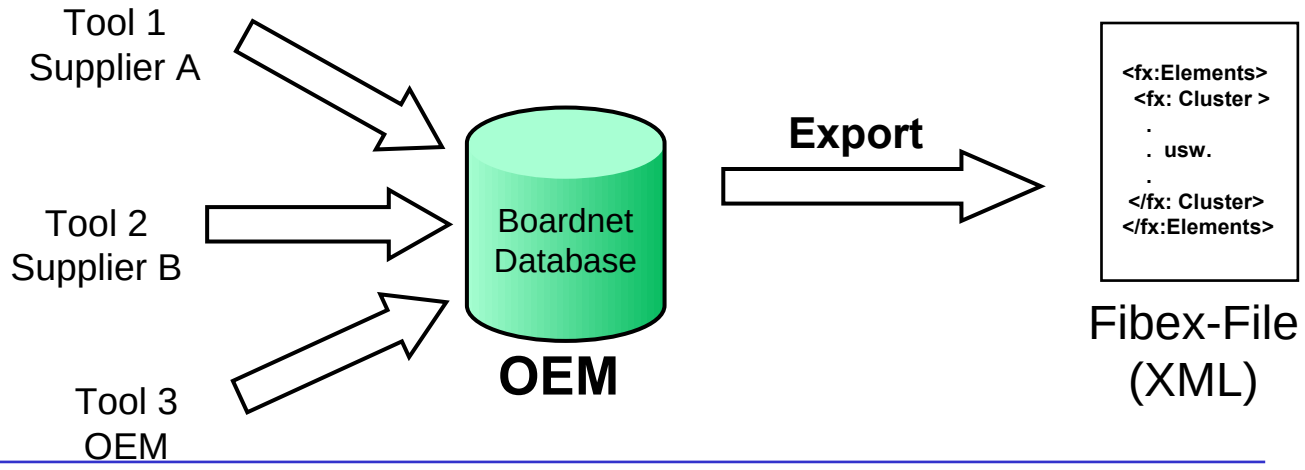
FIBEX Use Cases

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

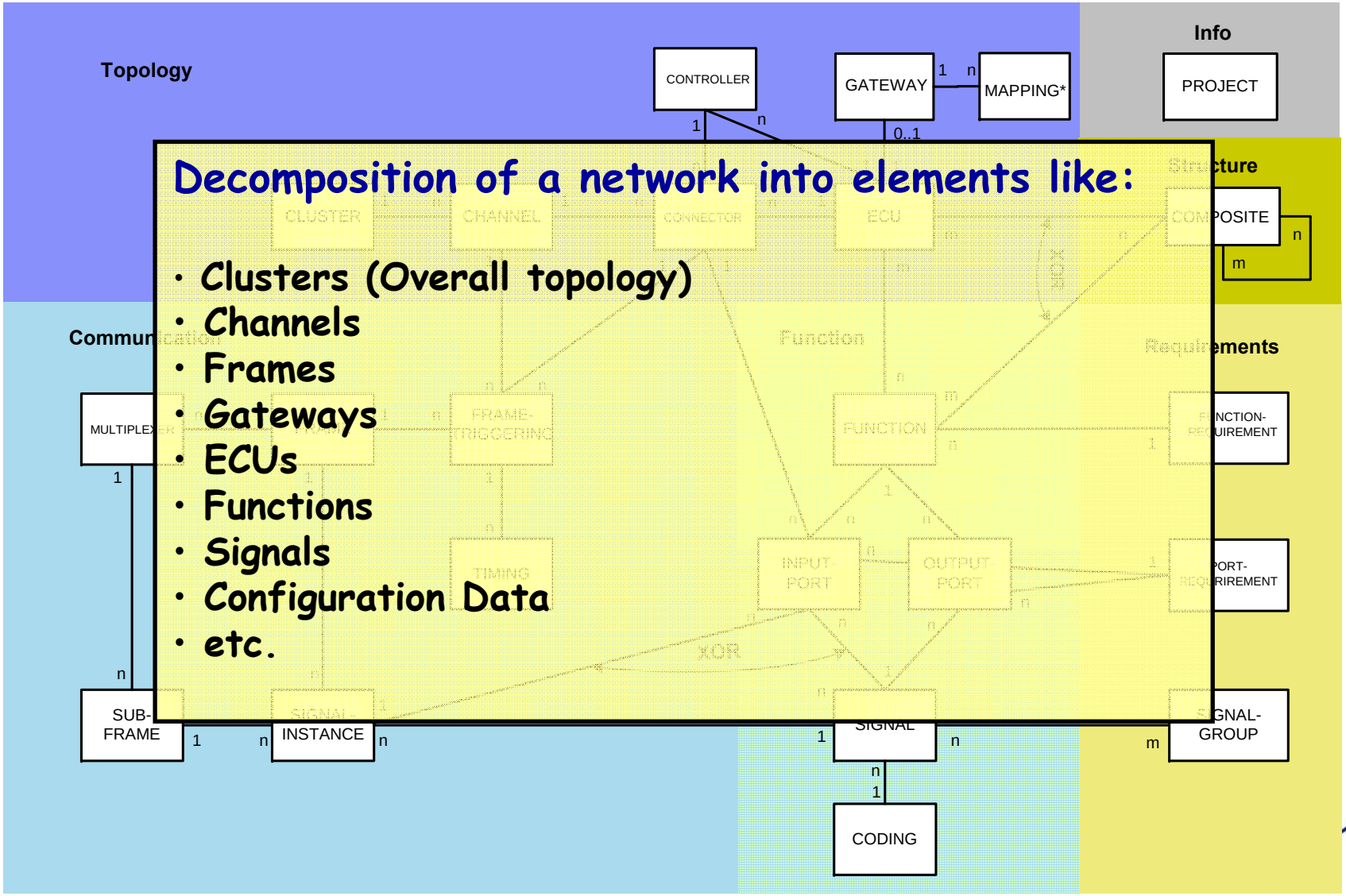
Communication properties

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

Functional network (optional)

- functions
- signals

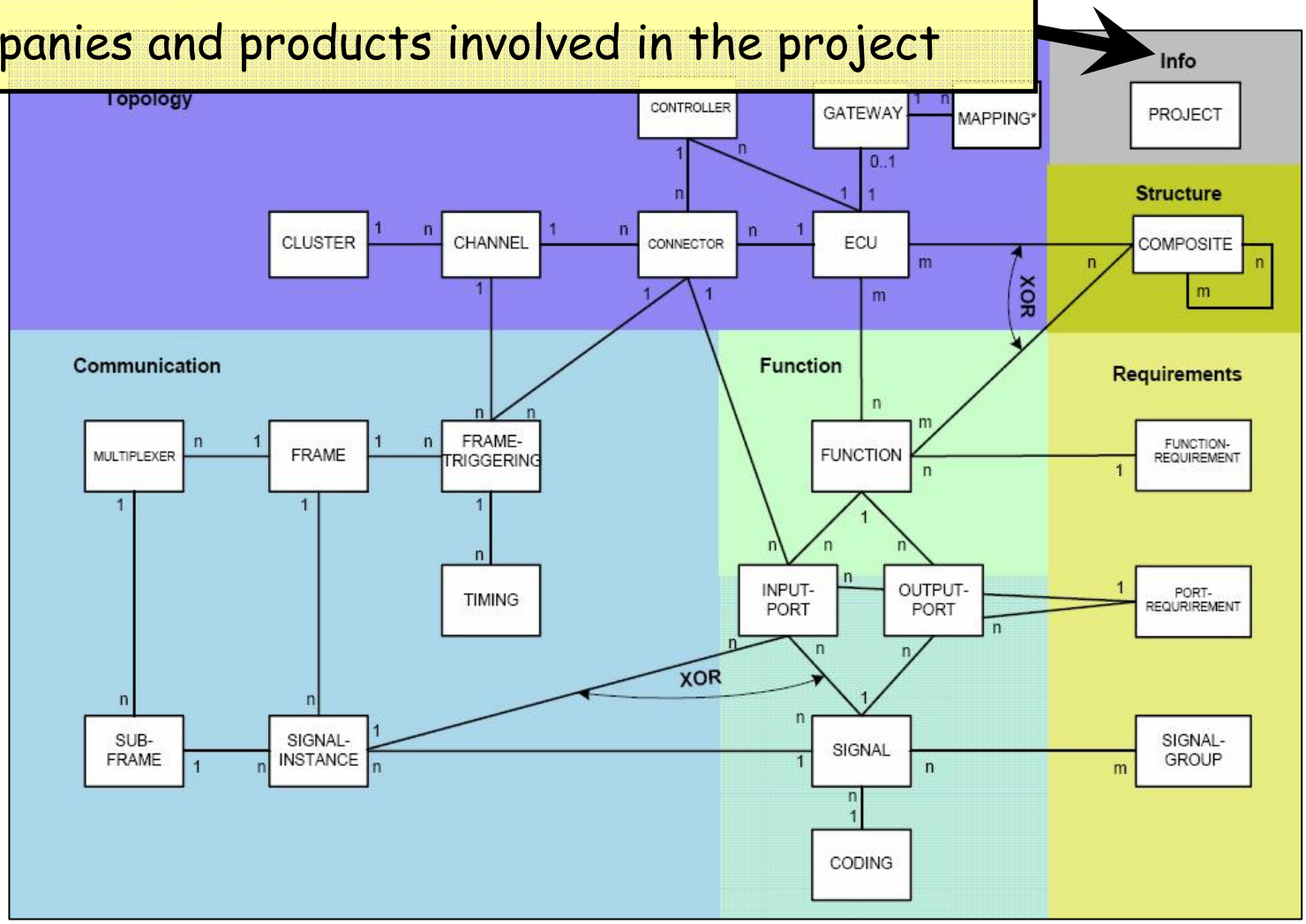
FIBEX Entities



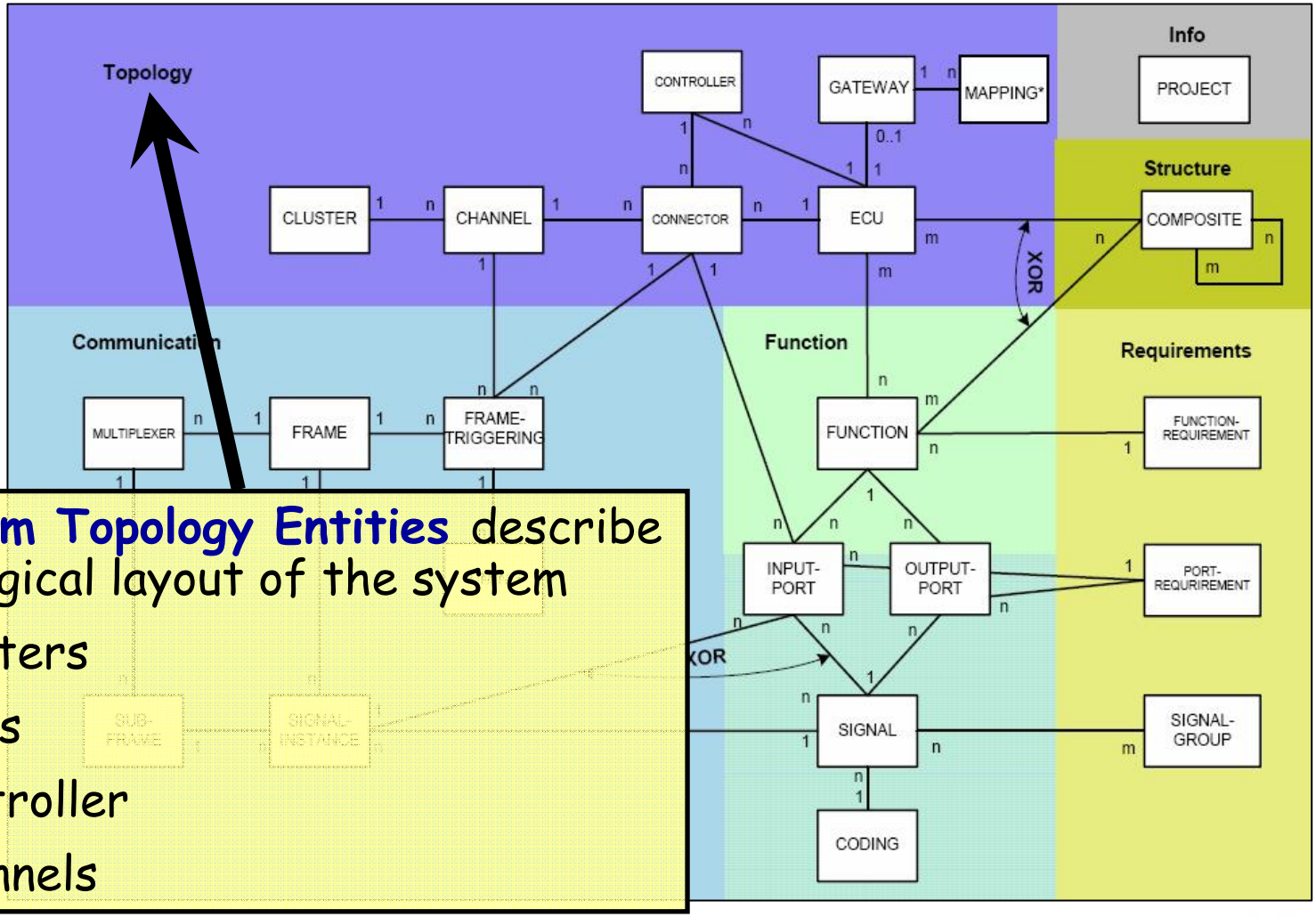
FIBEX Entities

Information Entities add Metadata to the project

- Companies and products involved in the project



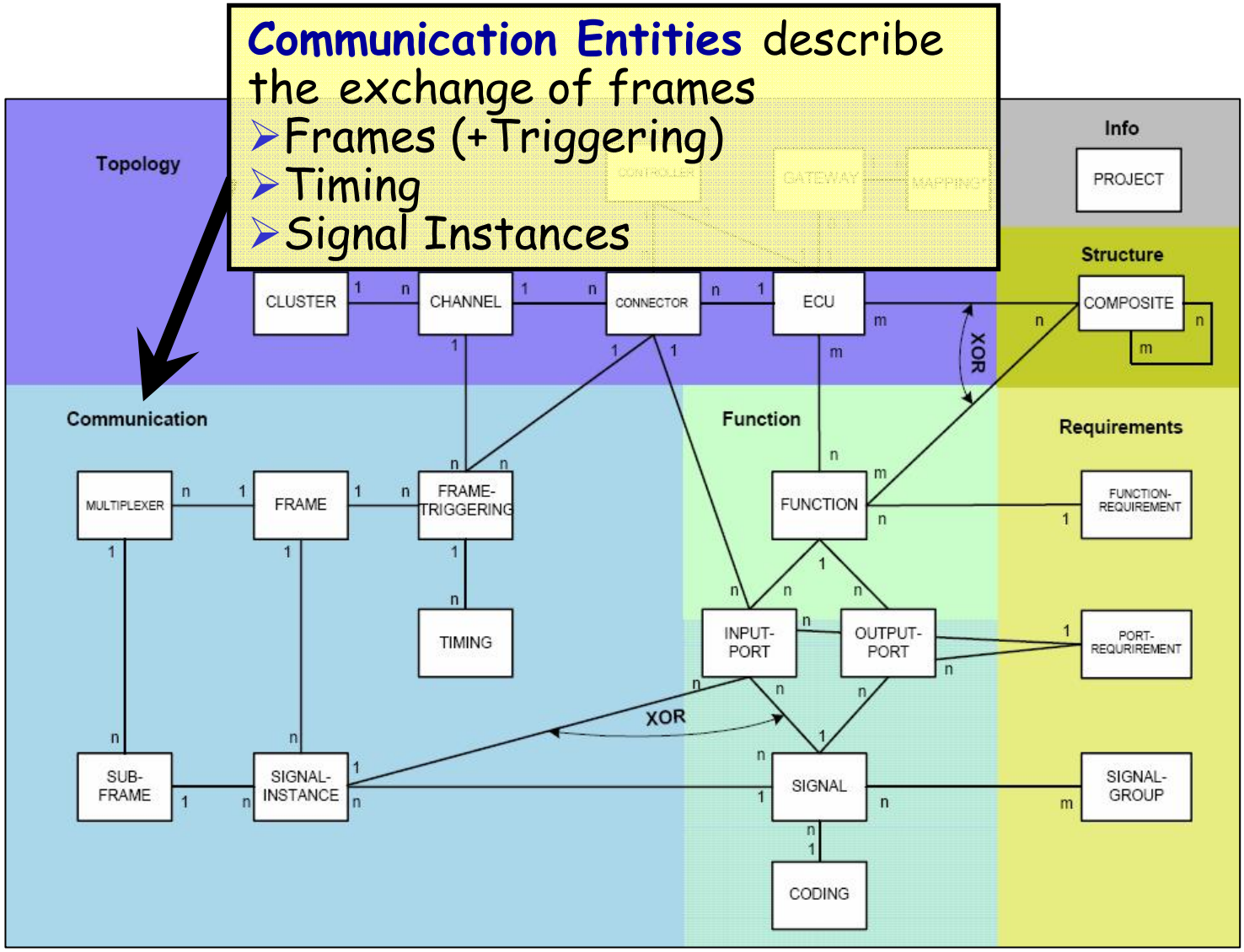
FIBEX Entities



System Topology Entities describe the logical layout of the system

- Clusters
- ECUs
- Controller
- Channels

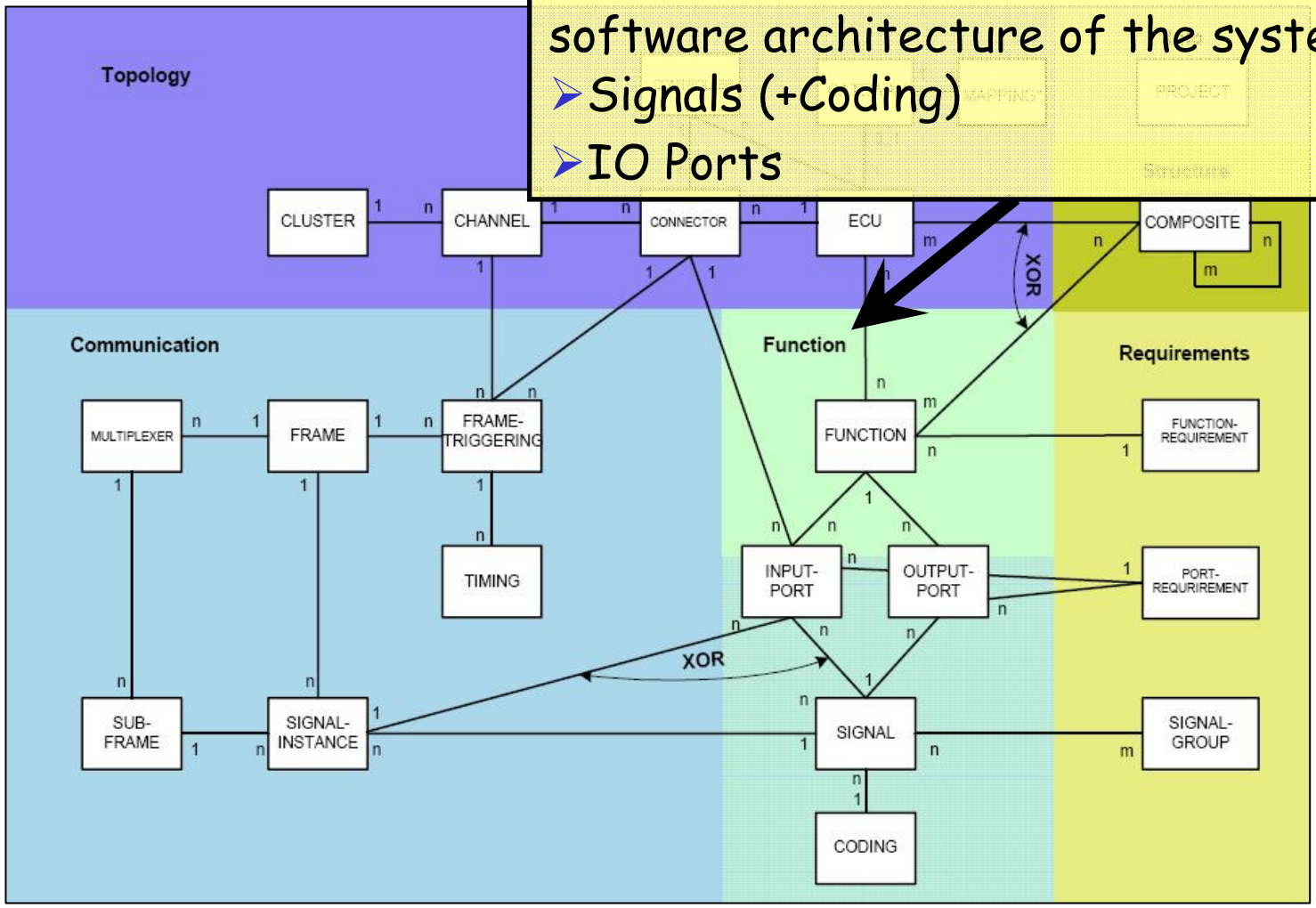
FIBEX Entities



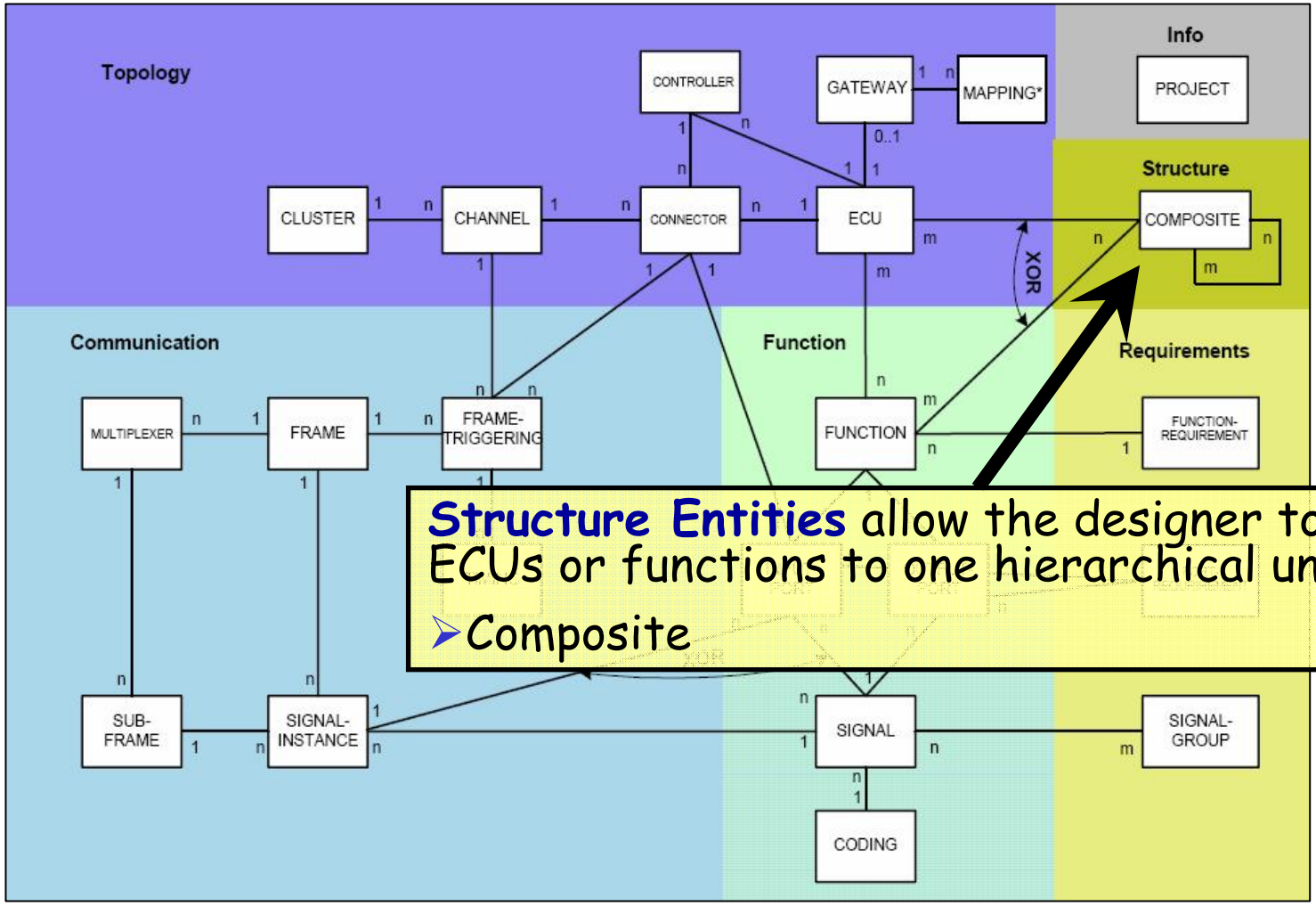
FIBEX Entities

Functional Level Entities describes the software architecture of the system.

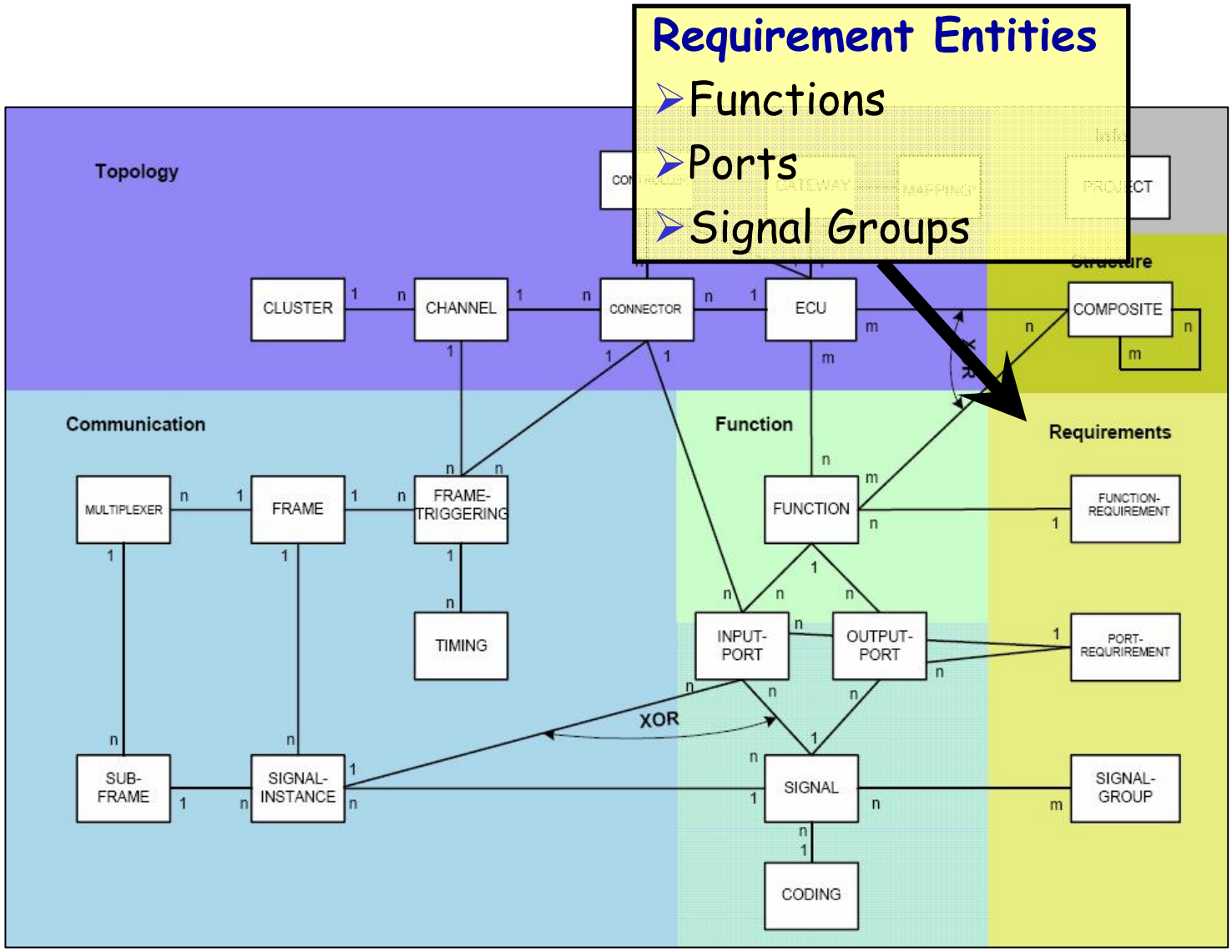
- Signals (+Coding)
- IO Ports



FIBEX Entities



FIBEX Entities



FIBEX Extensions

FIBEX allows two types of extensions to be used:

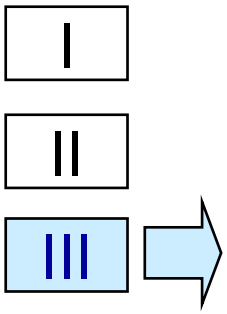
Manufacturer Extensions:

- add new functionality to FIBEX.

Platform Specific Extensions

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

Overview



Presentation of TÜV NORD IFM

The Fibex data exchange format

Motivation for the Checker Tool

Overview of the Checker Tool

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.
  - etc.

Checks for numerous mathematical relations
and configuration constraints that do exist with
respect to the communication parameters of a bus system.

connected to the channel the signal is being transmitted on.

Example for a rule (A)

Rule A:

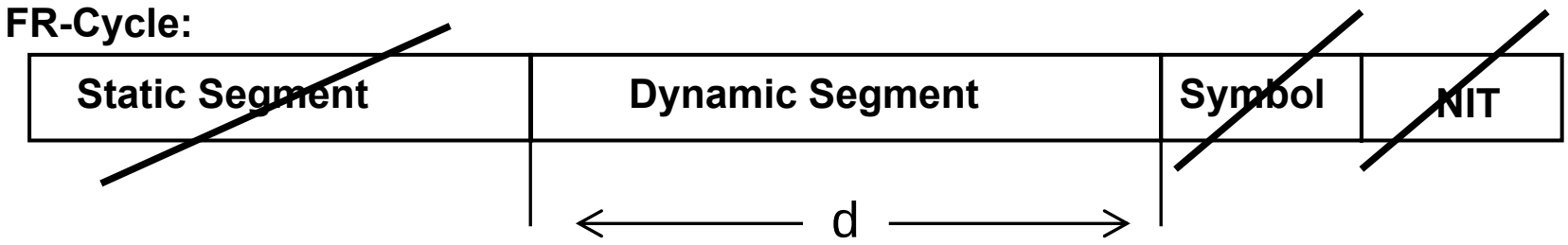
FR-Cycle:

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:



d ≤ gNumberOfMinislots ?

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:

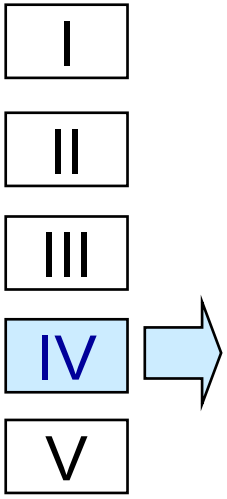
Frame (payload):

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



Presentation of TÜV NORD IFM

The Fibex data exchange format

Motivation for the Checker Tool

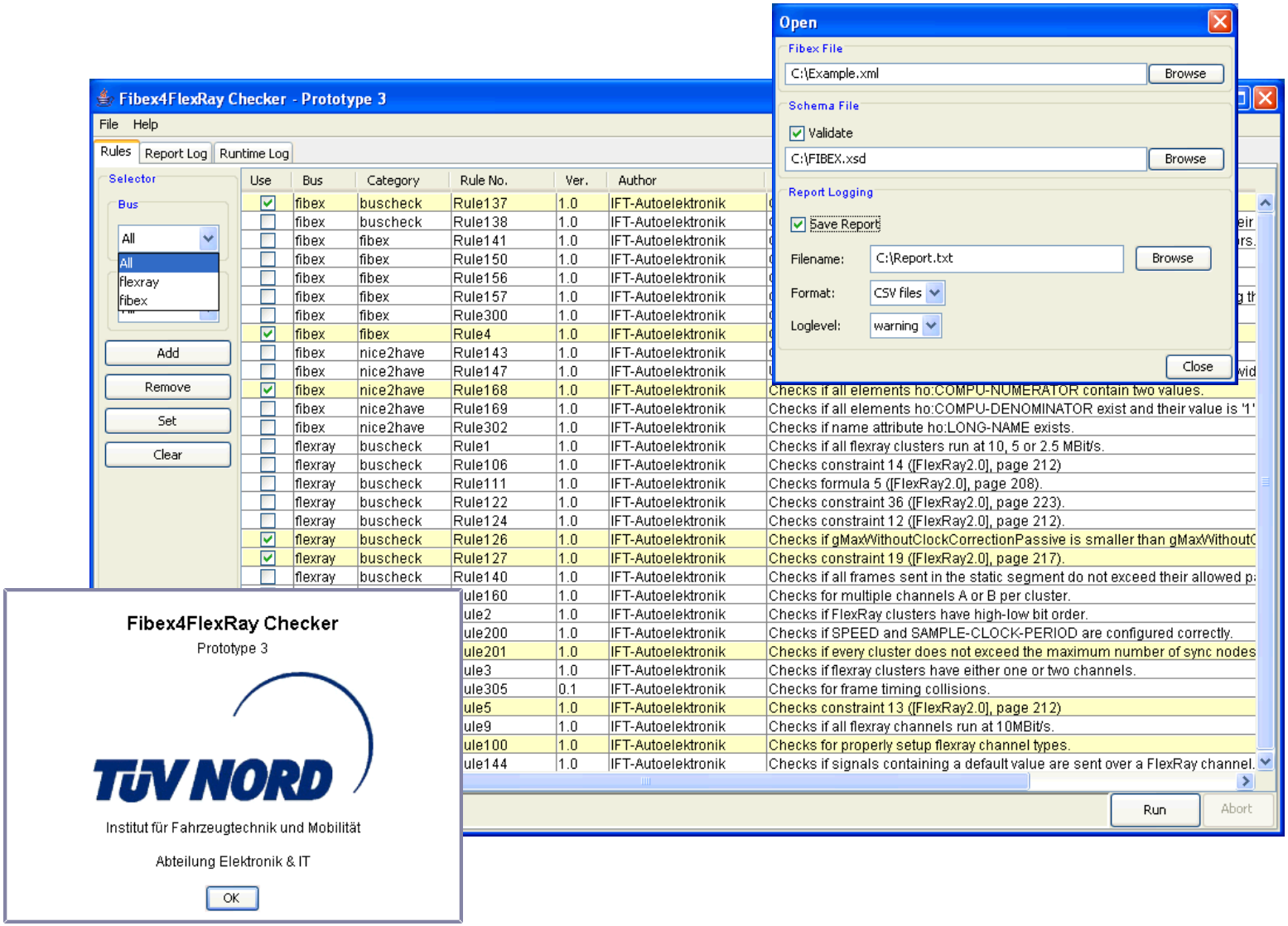
Overview of the Checker Tool

Summary

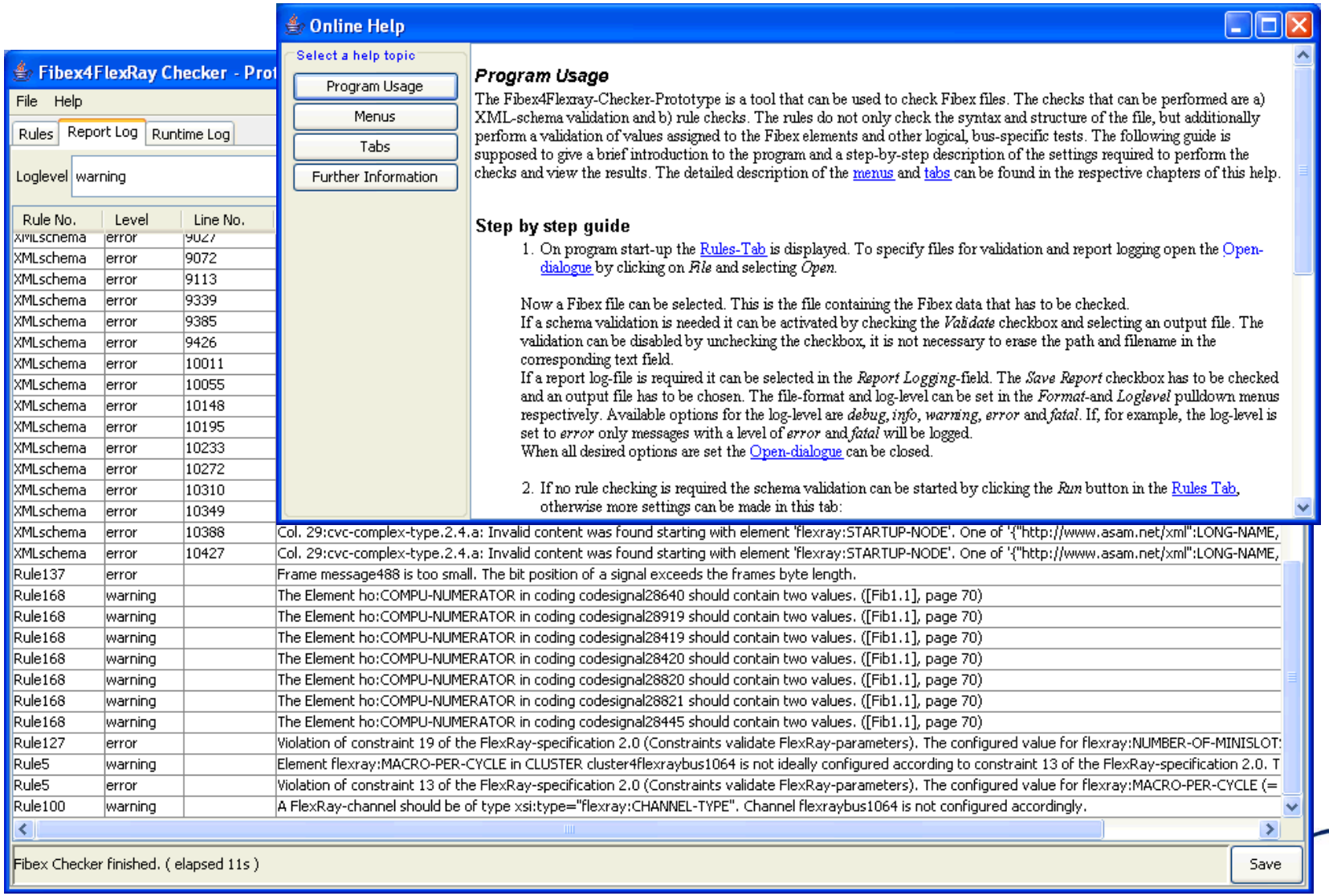
The “Checker-Tool”

- **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

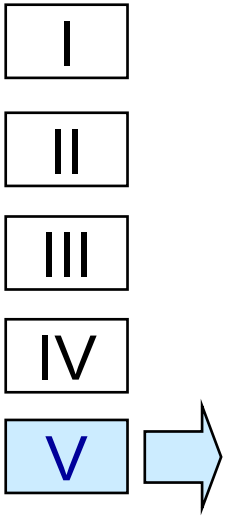
Screenshot: Selection of Validation Rules



Screenshot: Validation Results and Help System



Overview



Presentation of TÜV NORD IFM

The Fibex data exchange format

Motivation for the Checker Tool

Overview of the Checker Tool

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

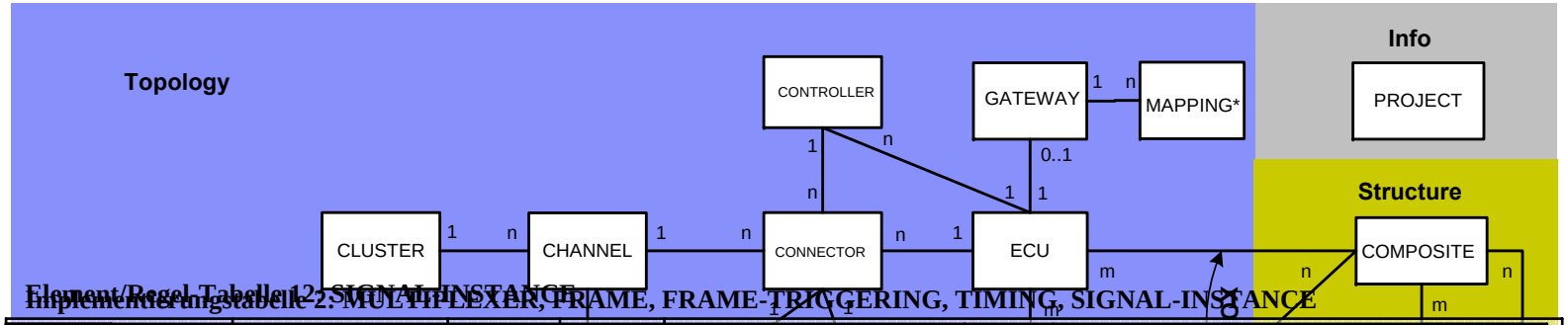
Institut für Fahrzeugtechnik & Mobilität

Fachgebiet „Electronics & IT“

Wolfgang Mickisch

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

Kurzvorstellung „Analyse“



Element/Regel-Tabelle 12: SIGNAL-INSTANCE

Implementierungstabelle 2: MULTIPLEXER, FRAME, FRAME-TRIGGERING, TIMING, SIGNAL-INSTANCE

R:	Regel formal	Kat.	Meldung / Kommentar
B138i	Ausgehend von allen Frames $frame (//fx:FRAMES/fx:FRAME/@ID) muß jeweils für die darin vorkommenden Signal-Instances $signalinst (//fx:FRAME/fx:SIGNAL-INSTANCES/fx:SIGNAL-INSTANCE/@ID) überprüft werden, ob die Anzahl der Bits zwischen der aktuellen BIT-POSITION (//fx:SIGNAL-INSTANCE[@ID=$signalinst]/fx:BIT-POSITION) und der BIT-POSITION der nächsten SIGNAL-INSTANCE größer oder gleich der BIT-LENGTH des referenzierten Signals ist: $signal (//fx:SIGNAL-INSTANCE[@ID=$signalinst]/fx:SIGNAL-REF/@ID-REF) $coding (//fx:SIGNAL[@ID=$signal]/fx:CODING-REF/@ID-REF) (//fx:CODING[@ID=$coding]/ho:CODED-TYPE/ho:BIT-LENGTH). Ist anstatt BIT-LENGTH eine MAX-LENGTH angegeben, muß diese überprüft werden (//fx:CODING[@ID=$coding]/ho:CODED-TYPE/ho:MAX-LENGTH).	F	Short: Signal $signal in frame $frame is too long. Verbose: Signal $signal in frame $frame is too long. The space between the BIT-POSITIONs configured in the corresponding signal instances is too small for the ho:BIT-LENGTH (or the ho:MAX-LENGTH) in the CODING of $signal.