TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Model Development and Data Provision for Formal Analysis of Cause Effect Chains

Project Work

Presented by,
Tejas Pravin Phadnis [406696] [1] [2]

Guided by,
Prof. Dr. Klaus Schneider[1]
Mr. Max Jonas Friese[2]
Mr. Hannes Walz[2]

1 Technische Universität Kaiserslautern
Department of Commercial Vehicle Technology
And Department of Computer Science

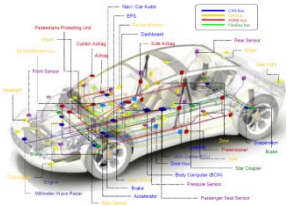
2 Daimler AG
Mercedes Benz Car Development

Contents

- Motivation
- Introduction
- Purpose
- Problem Statement
- Concept
- Tool Overview
- Agile Implementation
- Software Topology
- Hardware Topology
- Database Management
- Conclusion & future scope

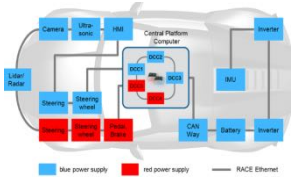
Motivation

- Automobile as a cyber physical system



- Complexity of network

Drive for the development …!!

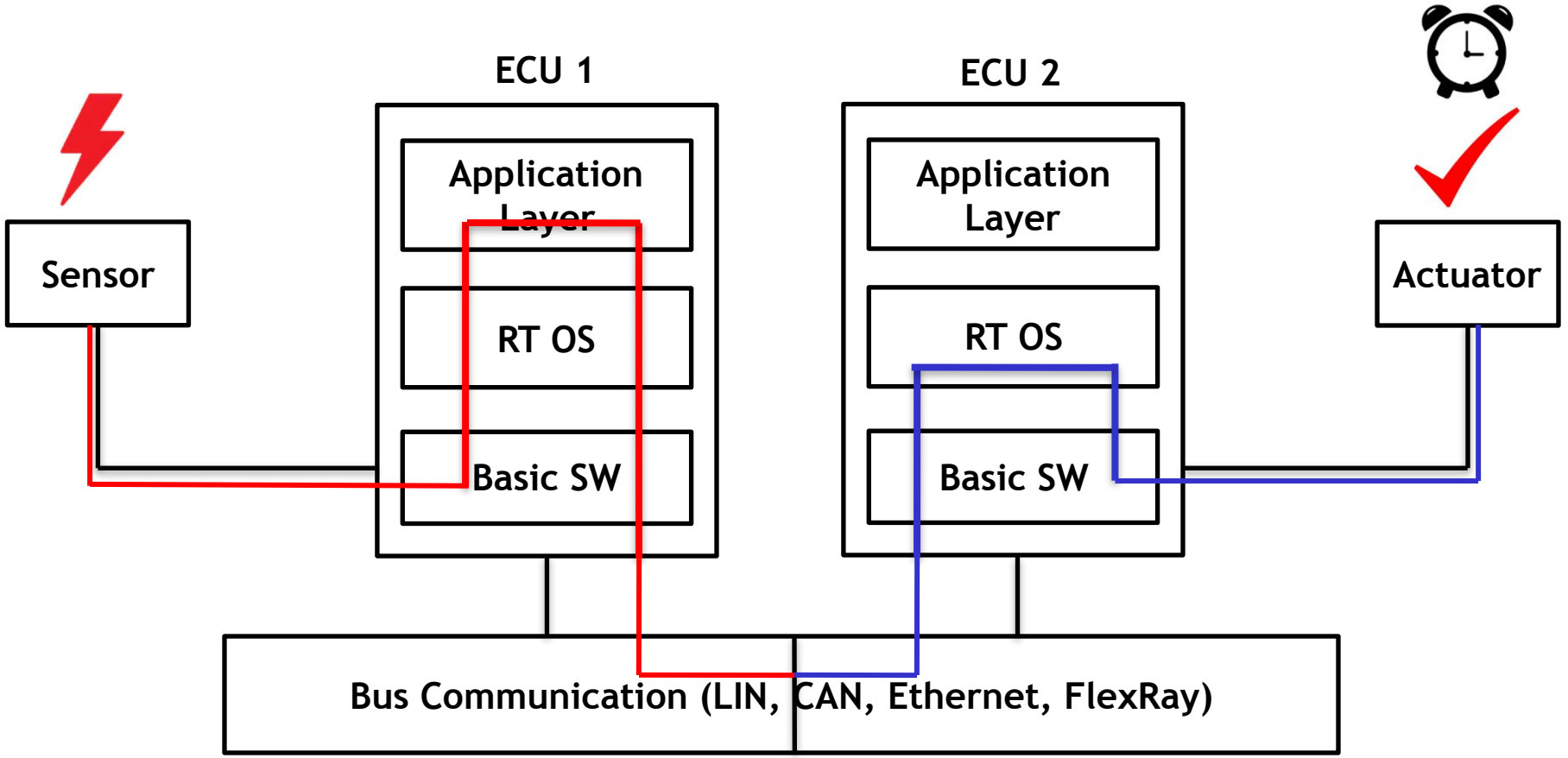


- E/E architecture

- Network Communication – Bus topology

Source: www.mercedes-benz.com
www.bosch-semiconductors.com

Introduction - Cause Effect Chain



Purpose

Major challenge in the embedded system of powertrain

- ➢ Improving performance of the powertrain
- ➢ Performance analysis and verification of hardware and software system
- ➢ Minimization of the reaction time
- ➢ Predicting reaction time before start of the system development and the implementat
- ➢ Finding out the estimations of the worst case
- ➢ Prediction can be achieve by formal analysis methods

Problem Statement

Problems addressed in this work:

1. Which data must be collected for a continuous, formal description of the cause effect chain?
2. What exchange formats are available and how can information that changes with different frequencies (hardware vs. software development) be brought together in a consistent way?
3. How can system model and formally described cause effect chain be brought together? At which points is a ‘cut’ of the analysis of the chain possible?

Concept

Formal analysis with abstract model

↳ Time verification of behaviour

↳ Expensive in terms of computational resources

↳ To deal with such a problem a tool support is needed

In tool

➢ A developer can work on hardware and software topologies together

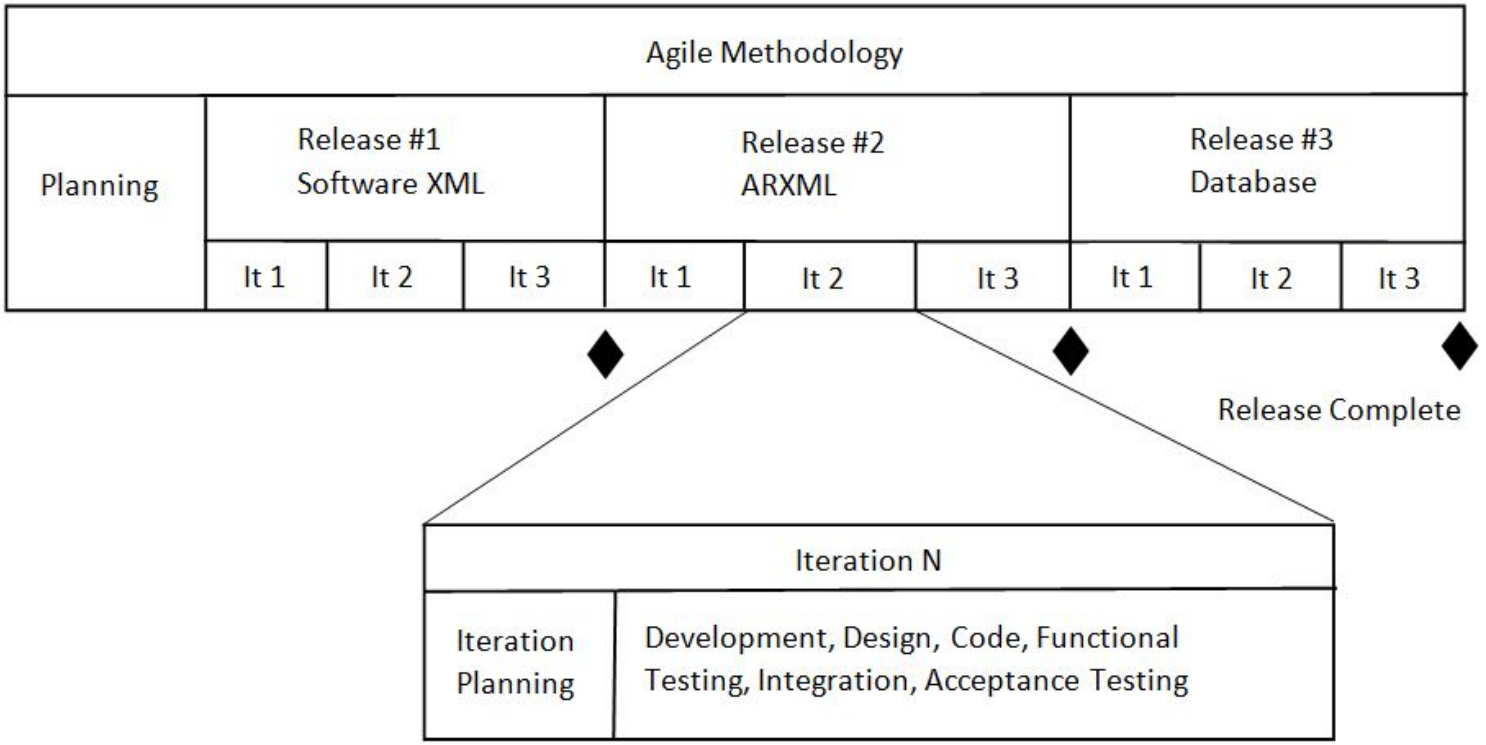
➢ Understanding of flow of the signal

➢ Identification of network resources

➢ Performance analysis which finally supports design decisions of cause-effect chain

Agile Implementation

Agile is the methodology of software development based on the incremental and iterative development where requirements and solutions evolved in each step.



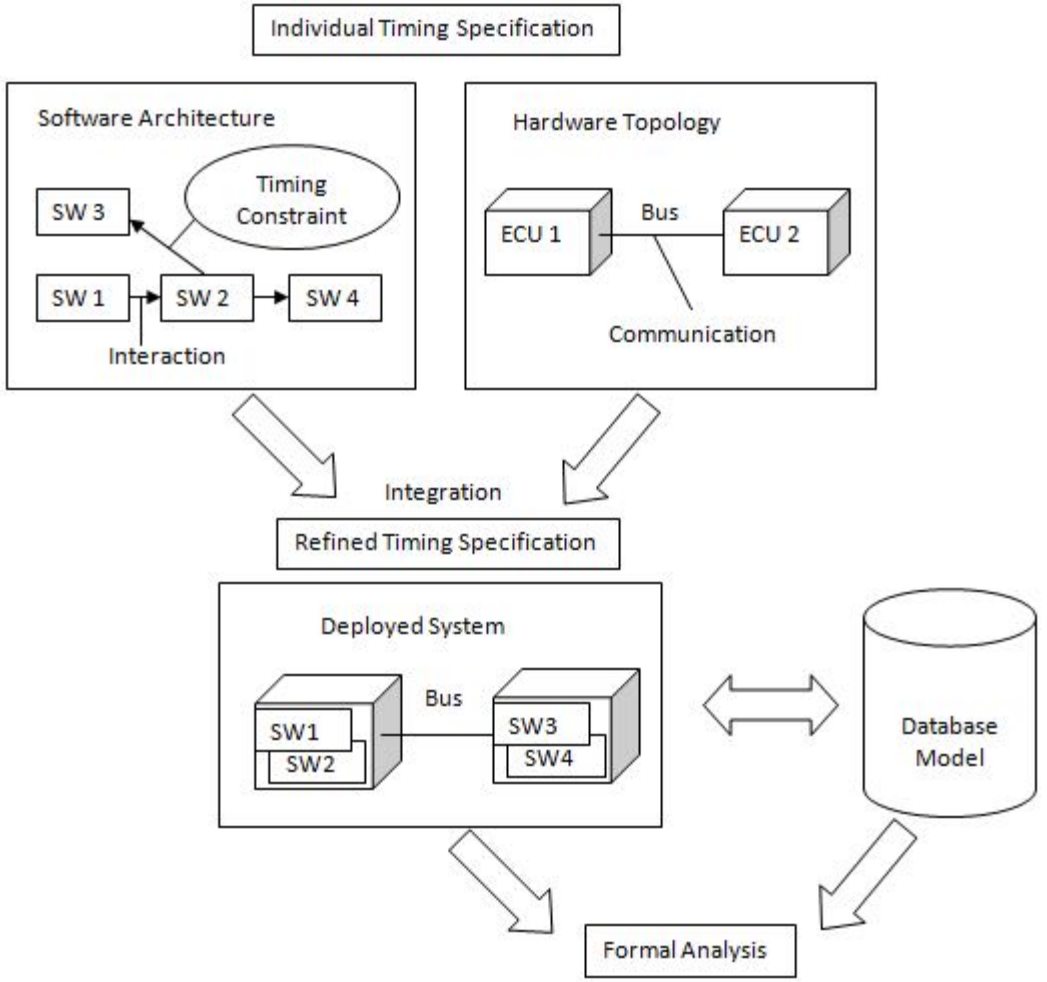
Tool Overview

Software Topology

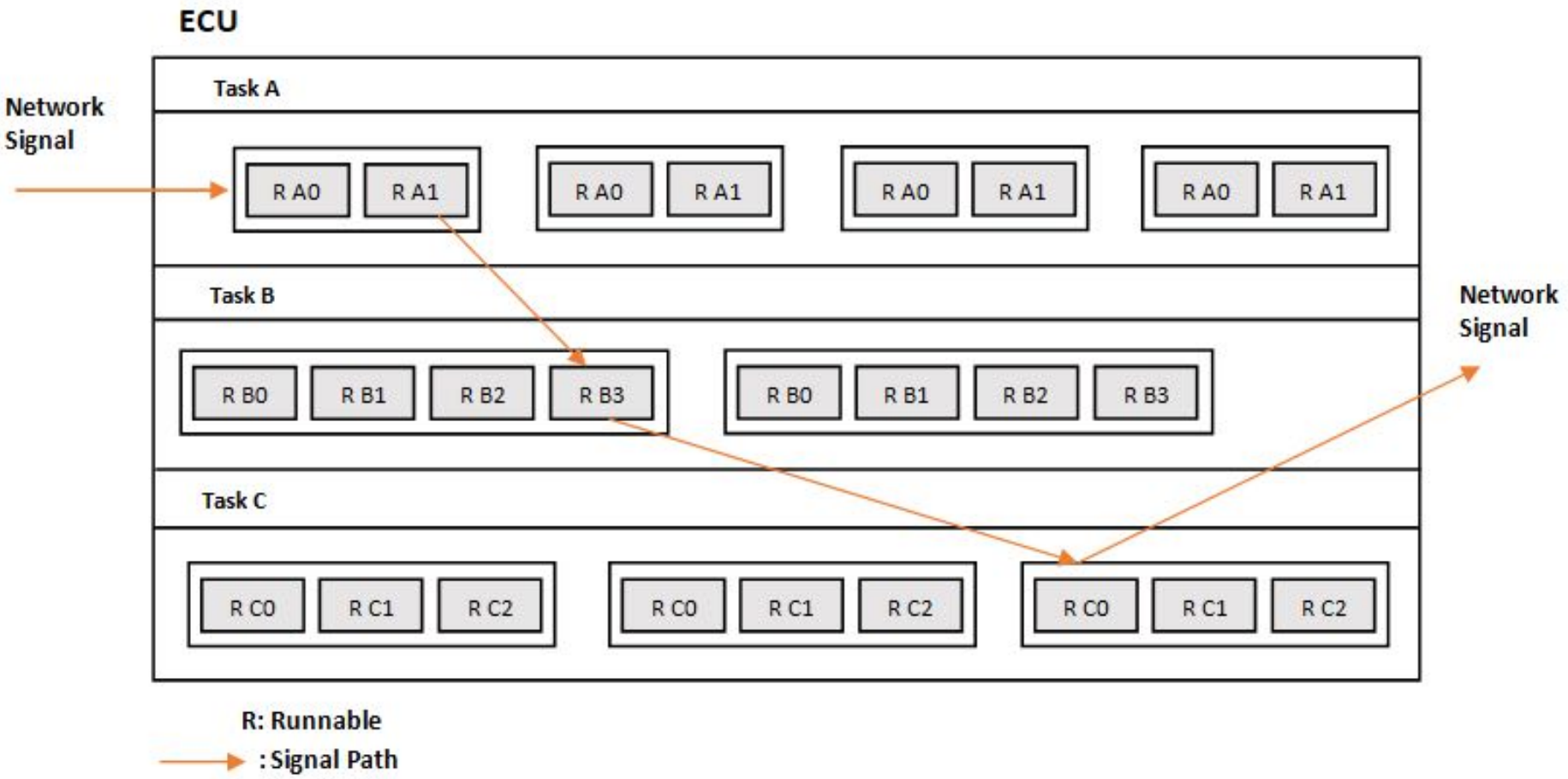
- ➢ Representation in SWXML file
- ➢ Information about ECUs, tasks, runnable entities
- ➢ Various types of signals

Hardware Topology

- ➢ Representation in ARXML file
- ➢ Represents communication cluster, busses, controllers, transport protocols
- ➢ Time relevant attributes and values



Software Topology



Software Topology

Challenges

- To identify the properties of a signal and path during transmission.
- It is important to find out which runnable is using which signal and finally how it is propagating
- Understanding of the XML structure and hierarchy as ECU contain tasks, task contains runnables and runnable reads/writes signals.

Methodology:

Development of user-friendly tool which converts XML data into the easily understandable

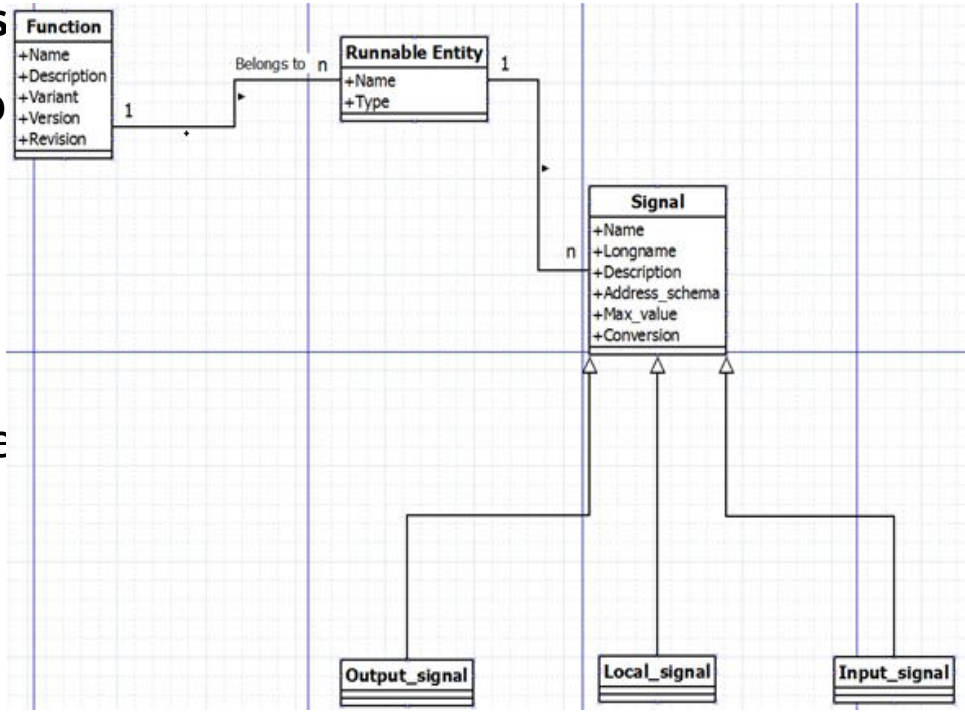
form and developed a GUI where the linkages between runnables can be visualized and chain between them can be formed.

Software Topology

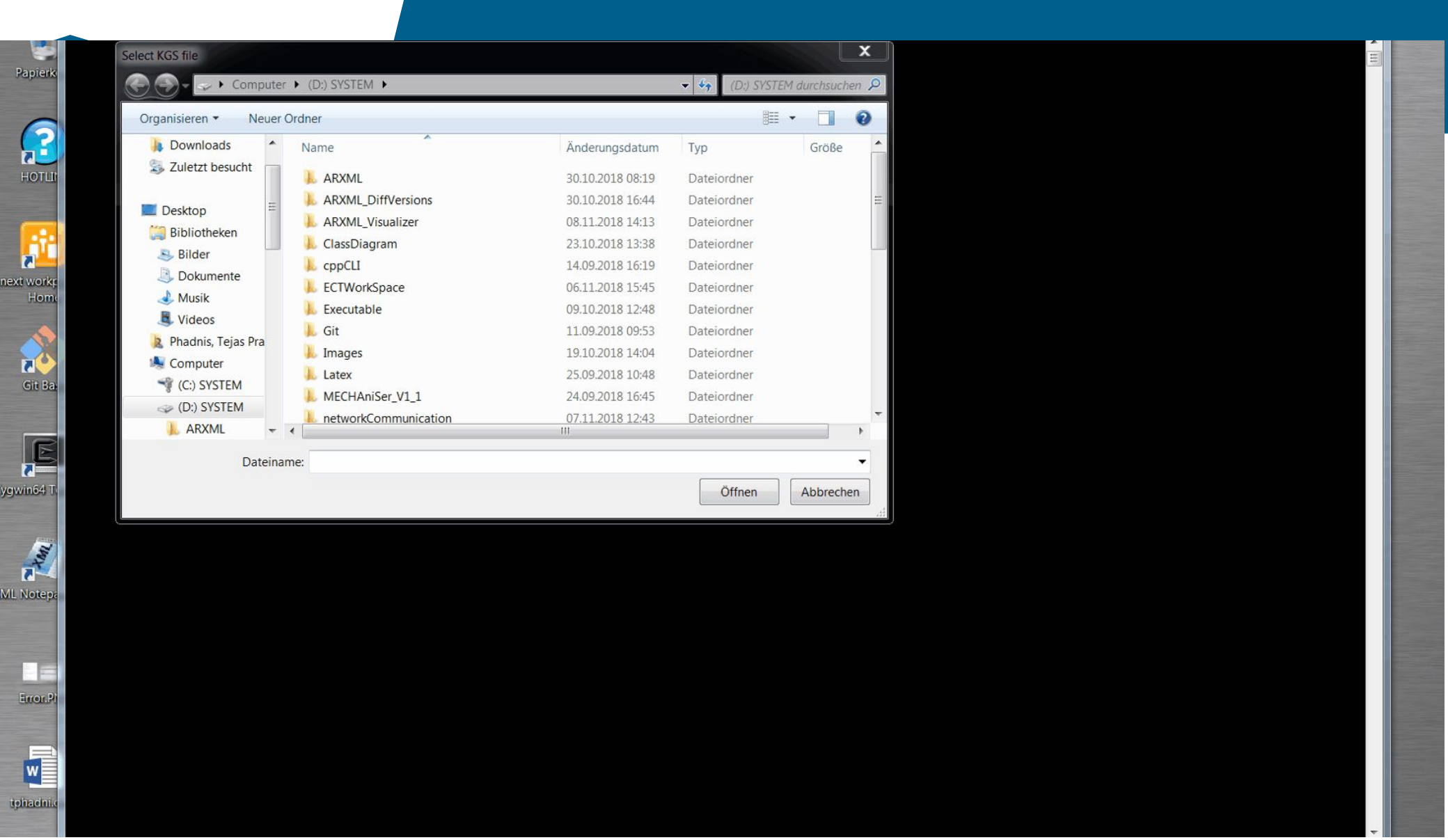
Automotive Software Development Concepts

UML: A diagram is the virtual representation o
the system.

1. **Class Diagram:** It helped for the better understanding of the schematics of the application. It states information about the interconnections, data type, inheritance, associations, etc.
2. **Use Case:** A methodology that describes how a user uses a system or application to accomplish a certain goal.



Tejas Pravin Phadnis
Commercial Vehicle Technology
University of Kaiserslautern

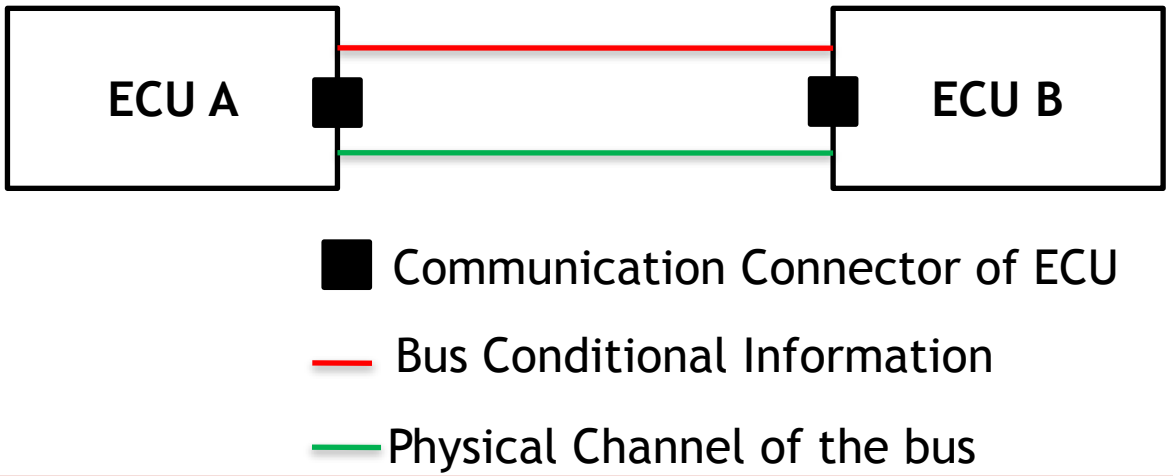

Select KGS file
Computer ▸ (D:) SYSTEM ▸
(D:) SYSTEM durchsuchen
Organisieren
Neuer Ordner
Downloads
Zuletzt besucht
Desktop
Bibliotheken
Bilder
Dokumente
Musik
Videos
Phadnis, Tejas Pra
Computer
(C:) SYSTEM
(D:) SYSTEM
ARXML
Name
Änderungsdatum
Typ
Größe
ARXML 30.10.2018 08:19 Dateiordner
ARXML_DiffVersions 30.10.2018 16:44 Dateiordner
ARXML_Visualizer 08.11.2018 14:13 Dateiordner
ClassDiagram 23.10.2018 13:38 Dateiordner
cppCLI 14.09.2018 16:19 Dateiordner
ECTWorkSpace 06.11.2018 15:45 Dateiordner
Executable 09.10.2018 12:48 Dateiordner
Git 11.09.2018 09:53 Dateiordner
Images 19.10.2018 14:04 Dateiordner
Latex 25.09.2018 10:48 Dateiordner
MECHAniSer_V1_1 24.09.2018 16:45 Dateiordner
networkCommunication 07.11.2018 12:43 Dateiordner
Dateiname:
Öffnen
Abbrechen

Hardware Topology

➢ **AUTOSAR** helped to manage the complexity of the interconnections and mapping between components and their communication network.

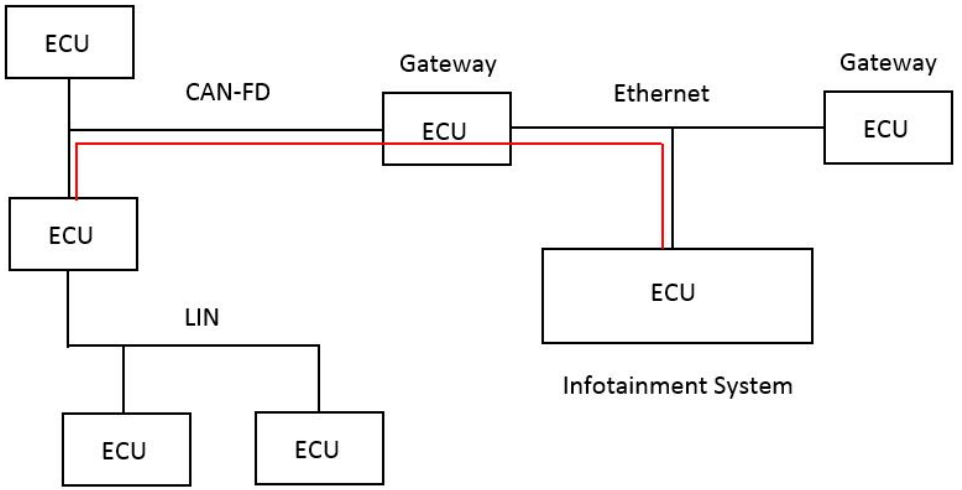
➢ **ARXML** is mainly focused on how ECUs are communicating with each other i.e. the entire communication cluster including signals and buses and their connections.

- **Focus** of the hardware topology



Hardware Topology

Aim: Time required to communicate between
2 ECU nodes via a specific bus.



For such estimation of time, it is important
to know

1. How the message is transmitted via buses
2. Which artifacts are involved in the process (PDUs, frames, connectors, etc.)
3. Description of bus protocols, and their latencies, etc.

Hardware Topology

Challenges in ARXML:

➢ To determine the bus conditional information i.e. how the frames or services, PDUs and signals are encapsulated in the packages of different buses

➢ Different ARXML versions file have different package structure

➢ ARXML file is focusing on a either a bus or ECU

➢ Frames of LIN, CAN are statically present and Ethernet frames are dynamically generated

➢ Need to handled many references and XPath in many communication module

Hardware Topology

Result:

```
Timing Attributes of Version 3 ARXML:

ECU: Main Controller
        Max-Number-of-Bits: 10
        Min-Number-of-Bits: 5
        Cycle-Offest: 0.001
        Reduced-Time: 4.0
        ECU Connector: Short-name
        ECU Controller: Short-name

BUS: CAN
        Delay: 0.001
        Activation-Time: False
        Sample-Point: 34
        Timeout: 1.5

Frame: Frame-name
        Length: 15

PDU: Pdu-name
        Activation-Time: True
        Repetition-Time: 0.002
```

Database Management

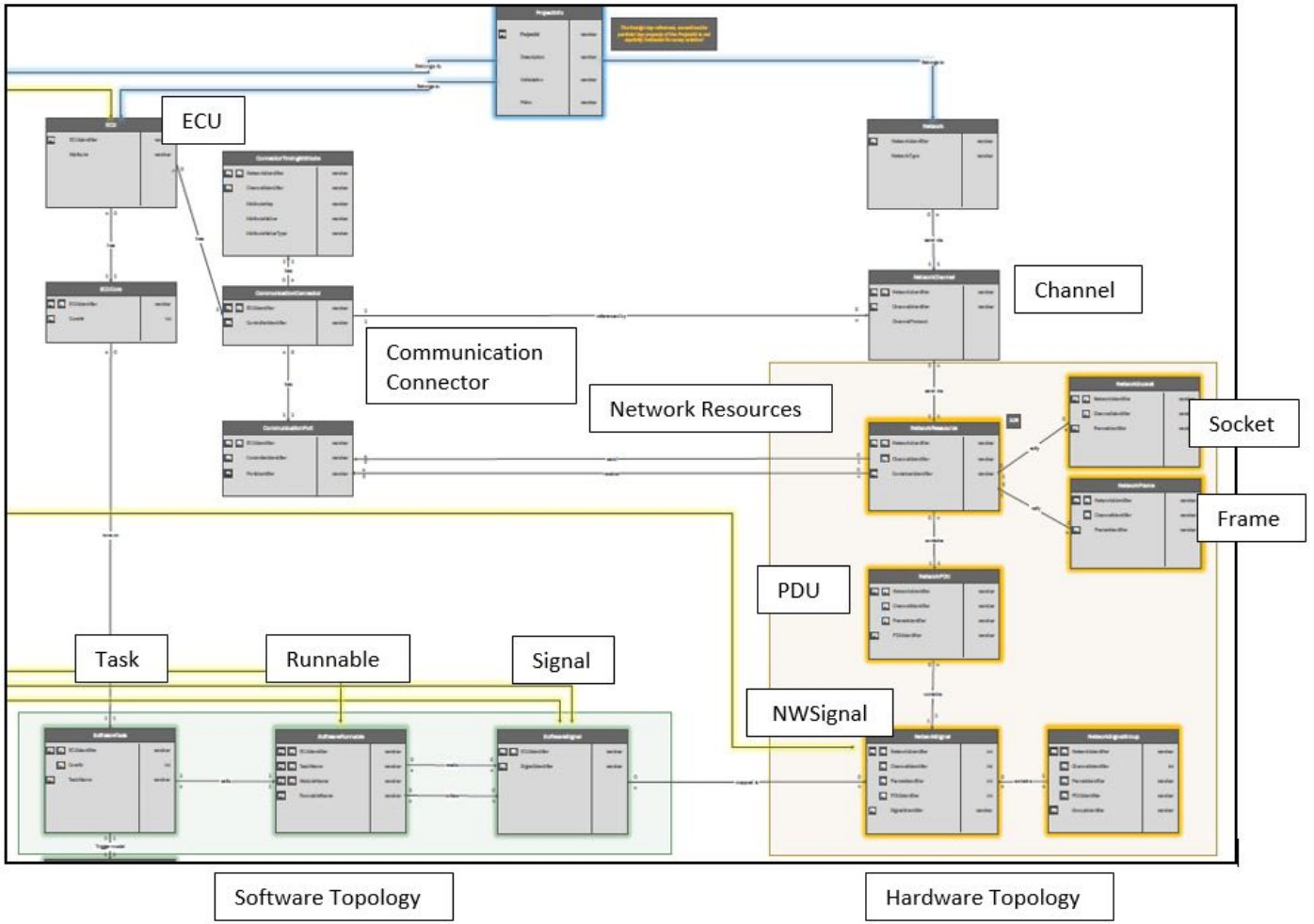
Necessity:

➢ Both the topologies are based on the real time embedded system where parameters are changing continuously.

➢ This data is further used for the formal analysis.

➢ Therefore, it is required to manage and store the data in a structural way.

Database Management



FACHBEREICH INFORMATIK
Tejas Pravin Phadnis
Commercial Vehicle Technology
University of Kaiserslautern

Conclusion and Future Work

Conclusion:

➢ The tool helped to deal with the complexity of ECU architecture and communication network by bringing hardware and software topologies together.

➢ The developer can visualize the complete data model of the communication network.

➢ All the timing attributes with their values are stored in the database also.

➢ The complete data provision which is required for the formal analysis is available in the tool.

Future Work:

➢ These data models are further used to find out the worst time scenarios in cause-effect chain.

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Thank you !

Questions ?