
TIMMO2_{use}

TIMMO-2-USE

Timing Model – Tools, algorithms, languages, methodology, USE cases

Report type	Deliverable D9.2
Report name	Use Cases
Report status	Consortium Confidential
Version number	1.0
Date of preparation	2012-01-16

TIMMO-2-USE Partners

AbsInt Angewandte Informatik GmbH
Arcticus Systems AB
Chalmers University of Technology
Continental Automotive GmbH
Delphi France SAS
dSpace GmbH
INCHRON GmbH
Institute National de Recherche en Informatique et Automatique
INRIA
Mälardalen University
Rapita Systems Ltd, UK
RealTime-at-Work
Robert Bosch GmbH
Symtavision GmbH
Technische Universität Braunschweig
University of Paderborn
Volvo Technology AB

Project Coordinator

Dr. Daniel Karlsson

Volvo Group Trucks Technology
Advanced Technology & Research
Dept 6260, M2.7
405 08 Göteborg
Sweden
Tel.: +46 31 322 9949
Email: Daniel.B.Karlsson@volvo.com

Copyright © 2010-2012: The TIMMO-2-USE Consortium

Authors

Stefan Kuntz, Continental Automotive GmbH (Editor)

Document History

Version	Date	Description
1.0	2012-01-16	First version.

Table of Contents

TIMMO-2-USE Partners	2
Authors.....	3
Document History	4
Table of Contents	5
1. Introduction	7
1. Use Cases	9
1.1 Main Use Cases	9
UC#0001 - Specify Time Budgets.....	9
UC#0002 - Specify Mode Dependent Timing Information	11
UC#0003 - Revise Erroneous Timing Information	12
UC#0004 - Negotiate Time Budgets	13
UC#0005 - Develop Control Applications	14
UC#0006 - Specify Variability and Timing Information	16
UC#0007 - Develop Application and Infrastructure	17
UC#0008 - Exchange Models	17
UC#0009 - Perform Post-Build Parameterization	18
UC#0010 - Specify Synchronization Timing Constraints.....	19
UC#0011 - Specify Probabilistic Timing Properties	20
UC#0012 - Integrate Re-usable Component	21
UC#0013 - Specify System Dimensions	22
1.2 Specific Use Cases.....	22
Capture, Analyze, and Utilize Uncertain Timing Information.....	22
Capture, Analyze, and Utilize Worst Case Timing Information	23
Develop Body Controller following Top-down Approach	23
Develop Cruise Control following Top-down Approach	24
Develop Engine Management System On Implementation (AUTOSAR) Level.....	25
Exchange Timing Information between Control Engineer and Software Engineer.....	26
Exchange Timing Information with worst case verification tool	27
Explore Design Alternatives for Control Applications	27
Control Scheduling Co-Design with Fixed Rates.....	28
Control Scheduling Co-Design with Flexible Timing Structure	29
Generate Test bench for Non-functional Properties	29
Handle Timing Information in Simulation Based Analysis Activities.....	30
Integrate Re-usable SW-Component.....	31
Integrate Several Control Applications on a Target Platform	32

Perform Timing Analysis On Code-Level	33
Process Timing Information for HIL-based Simulation	34
Specify End-to-End Latencies	35
Transform Continuous Time Model to Discrete Time Model	35
Transform Timing Information from Vehicle Level to Analysis Level.....	36
Transform Timing Information from Analysis Level to Design Level	37
Transform Timing Information from Design Level to Implementation Level.....	38
Verify Timing Constraints	39

1. Introduction

Purpose

The purpose of this document is to document all use cases that have been defined during the course of work package 1 “Use Cases and Requirements”.

Scope

This document contains the use cases important for work package 2 “Language” to work package 5 “Validation”.

Format of Use Cases

Every use case in this document is described using the format as described below:

Name: This field contains the use case identification (Main Use Cases), or an active-verb goal phrase that names the goal of the use case’s actor (Specific Use Cases).

Alias: An active-verb goal phrase that names the goal of the use case’s actor.

Description: Objective/Goal: This section states the objective/goal of the use case.

Description: This section describes the use case in more detail and provides further information about the background important to know.

Actors: A list of actors involved in the use case.

Stakeholders: A list of stakeholders interested in the results of the use case respectively the fact that the use case is performed.

Originator: The originator of the use case, when the field “Author” does not contain a name. This section is optional

Status: This field contains one of the following states: Proposed, Approved, Rejected, and Implemented.

Author: This field contains the author’s name who contributed the use case. Otherwise the name of the editor of this document is listed.

Explanation: In the course of the TIMMO-2-USE project this field is used later to provide further information on the requirement’s status.

Type: UseCase

Relations: This field lists the requirements associated with this use case. In case of a Main Use Case this field lists all the Specific Uses Cases assigned to this use case.

Abbreviations and Acronyms

The abbreviations and acronyms listed in the table below are used in this document.

Abbreviation/ Acronym	Description
AUTOSAR	Automotive Open System Architecture

1. Use Cases

1.1 Main Use Cases

UC#0001 - Specify Time Budgets

Name: UC#0001 - Specify Time Budgets

Alias: UC#0001

Description: **Use Case Title/Name:** Specify Time Budgets

Objective/Goal: An E2E function normally spans over several ECUs and across the responsibility of multiple suppliers. OEMs need to divide the overall end-to-end latency to the ECUs and the communication channels, and assign these timing budgets to the suppliers.

Description: At the beginning of a project, the OEM must properly decide the time budgets for each ECU and communicate the specification to the suppliers. During the development process, the OEM and the suppliers want to keep the two-way feedback. When the suppliers have refined solutions at the proper abstraction level, the OEM can estimate if the time budgets are realistic, and may either ask the supplier to improve the solution or adjust the time budgets. On the other hand, given the timing estimates of the individual parts, the OEM may revise the timing requirements on vehicular functions to achieve optimal performance or cost of the entire vehicle.

We need to perform WCET analysis on all relevant levels of abstraction, although the cross-supplier issue arises mostly on the implementation level and possibly to some extent also on the design level.

- **Vehicle:** N/A
- **Analysis:** Simulink (or other behavioural) model using some hardware-independent time unit. This type of analysis can be used to determine properties on hardware needed to satisfy the timing budget.
- **Design:** Simulink model (or other behavioural) on hardware with given characteristics. Can we say something about the OS task and communication bus schedules?
- **Implementation:** C code on concrete hardware.

Implied TADL Support and Relevance to TIMMO-2-USE

This use case is related to the following work packages.

WP2 (Language)

- Structural extensions, including (1) the modeling constructs for timing budgets, hardware timing characterization, timing properties of executable code and communication channels, etc. (2) traceability between the analysis results at different abstraction levels.
- Algorithm specific extensions: Language constructs to support relevant methods for timing analysis, e.g., the estimation of WCET and communication latency.
- Methodological extensions: Effective communication between the OEM and the suppliers; Progressive negotiation on timing budgets; Support for different proposals.
- Semantical reasoning

WP3 (Algorithms & Tools)

- More precise timing analysis methods to obtain good timing estimates at the analysis and design levels.
- Improved timing information exchange between tools and stakeholders; optimized tool chain based on exchange format induced by the new TADL definitions.

WP4 (Methodology)

- Better support for collaboration through (1) the investigation on what information has to be shared between OEM and the suppliers while maintaining IP integrity of the collaborators. (2) the negotiation on the timing budgets among all collaborators.
- Virtual system integration: Estimation and validation of the overall timing requirements of the vehicular function based on the design models at the analysis and design levels, before the executable code is available.

Actors: Function owners, function developers, system testers

Stakeholders: Automotive OEM and suppliers

Status: Approved

Author: Daniel Karlsson

Explanation: None

Type: UseCase

Relations: VTEC#0012 - Identification of design decision

CAG#0034 - Automation

VTEC#0005 - Access right to TIMMO model

VTEC#0034 - Application-independent timing information

INRIA#0004 - Functional time

VTEC#0014 - Tool support for comparing alternative timing solutions

INRIA#0005 - Executable models

CAG#0002 - Event chains between LoA

VTEC#0025 - Model integration

VTEC#0001 - Traceability between design and implementation levels

VTEC#0013 - Effect of a selected solution

VTEC#0032 - Infrastructure-independent timing information

VTEC#0011 - Support of multiple solutions

CAG#0001 - Events between LoA

VTEC#0002 - Decomposition of time budget

VTEC#0038 - Tool integration

Perform Timing Analysis On Code-Level

Generate Test bench for Non-functional Properties

Develop Cruise Control following Top-down Approach

Transform Timing Information from Analysis Level to Design Level

Develop Engine Management System On Implementation (AUTOSAR) Level

VTEC#0004 - Timing budget negotiation between OEM and supplier

CAG#0005 - Hardware

Transform Continuous Time Model to Discrete Time Model

Transform Timing Information from Design Level to Implementation Level

VTEC#0006 - Different interpretations of timing information

INRIA#0002 - Time bases
CAG#0038 - Timing Analyses
VTEC#0003 - Methods for estimating WCET at analysis and design levels
VTEC#0035 - Methods for timing characterisation of hardware
CAG#0051 - Reuse of timing constraints
INRIA#0001 - Multiform concepts of Time
VTEC#0033 - Methods for timing characterisation of behavior/algorithm
CAG#0025 - Safety (timing)

UC#0002 - Specify Mode Dependent Timing Information

Name: UC#0002 - Specify Mode Dependent Timing Information

Alias: UC#0002

Description: **Use case Title/Name:** Specify Mode Dependent Timing Information
Objective/Goal: A function behaves differently in time depending on the present vehicle mode. The vehicle mode, such as the vehicle is running or parked, determines the active states of software components and power states of ECUs and networks. It hence has a great impact on the timing performance of the vehicle functions.
Developers specify the timing characterization for each running mode of the application.

Description: The mode has an impact on the state of software component, the OS task schedule, and the network schedule. One may need to specify the timing property of the functions for each mode. For best performance, it is even preferable to find the optimal task and bus schedules for each mode.

When the mode needs to be changed, the change event or change request must be propagated to the related components via the network. To maintain global mode consistency and high performance, the mode manager must arbitrate the mode switch requests, decide the proper target mode, and respond to the affected components. This mode request-decision-reply process must be bound by a deadline. As a side-effect, this process may also significantly increase the transient bus traffic.

Implied TADL Support and Relevance to TIMMO-2-USE

This use case is related to the following work packages.

WP2 (Language)

- Structural extensions: Mode-dependent timing descriptions; Mode-dependent descriptions on task and bus schedules at the implementation level; Timing constraints on mode management operations, etc.
- Algorithm-specific extensions: Mode-dependent bus scheduling parameters; Requirements on mode management.

WP3 (Algorithms & Tools)

- Methods and tools to obtain good task and bus schedules based on modes.
- Methods and tools to manage the mode consistency and optimize system performance.

WP4 (Methodology)

- Collaboration on the mode-dependent function distributed to multiple suppliers.

Actors: Function owners, function developers

Stakeholder: Function owner and developer

Status: Approved

Author: Danieal Karlsson

Explanation: None

Type: UseCase

Relations: VTEC#0009 - Method and tool support for mode-dependent bus scheduling
CAG#0002 - Event chains between LoA
CAG#0001 - Events between LoA
VTEC#0008 - Timing specifications depending on modes
Perform Timing Analysis On Code-Level
Develop Engine Management System On Implementation (AUTOSAR) Level
Transform Timing Information from Analysis Level to Design Level
Transform Timing Information from Design Level to Implementation Level
Develop Cruise Control following Top-down Approach
VTEC#0007 - Timing constraints dependent on modes
VTEC#0010 - Methodology support for mode-aware design
BOSCH#0006 - Mode dependencies
BOSCH#0005 - Mode dependent timing requirements for control applications
CAG#0025 - Safety (timing)

UC#0003 - Revise Erroneous Timing Information

Name: UC#0003 - Revise Erroneous Timing Information

Alias: UC#0003

Description: **Use Case Title/Name:** Revise Erroneous Timing Information

Objective/Goal: Product development is mostly about modifying and improving an existing system. Even for new product and function development, the process consists of several iterations with a lot of modifications between iterations. Such alterations may lead to conflicting timing information and unfortunately erroneous timing information. It is therefore of crucial importance to establish an efficient and transparent methodology dealing with timing information, as well as to uncover erroneous timing information as early as possible.

Description: This use case discusses the steps to be taken to correct erroneous timing information. Relevant issues to consider are:

- What other parts of the system, like features, functions, ECUs, busses, are affected by a correction of an erroneous timing information?
- Which types of timing faults lead to the discovery of erroneous timing information?
- What are plausible causes of the timing fault?

- What measures can be taken to correct a specific timing fault?

Implied TADL Support and Relevance to TIMMO-2-USE

The following work packages are related to this use case.

WP2 (Language)

- Semantic definitions than can lead to identification of erroneous timing information?

WP3 (Algorithms and Tools)

- Effective tool support to discover and analyze timing faults.

WP4 (Methodology)

- An effective collaboration process for finding and correcting timing faults.

Actors: Function owners, function developers, system integrators, system testers

Stakeholders: OEM and function developers

Status: Approved

Author: Danieal Karlsson

Explanation: None

Type: UseCase

Relations: VTEC#0014 - Tool support for comparing alternative timing solutions

VTEC#0015 - Methodology support for change management

VTEC#0013 - Effect of a selected solution

VTEC#0012 - Identification of design decision

CAG#0037 - EAST-ADL XML

VTEC#0011 - Support of multiple solutions

Develop Cruise Control following Top-down Approach

Generate Test bench for Non-functional Properties

Develop Engine Management System On Implementation (AUTOSAR) Level

INRIA#0003 - Timing expressions

UC#0004 - Negotiate Time Budgets

Name: UC#0004 - Negotiate Time Budgets

Alias: UC#0004

Description: **Use Case Title/Name:** Negotiate Time Budgets

Objective/Goal: The development process of vehicle electronic systems is always iterative. Even when a completely new architecture is being developed, different functions are added at different times. Consequently, developers and function owners must keep on negotiating time budgets iteratively.

Description: This use case is closely related to the use case "Change Existing Timing Information". The emphasis is on the double way communication between the developers of different functions. The introduction or modification on one function requires negotiation and compromise with other related functions.

Handling timing requirements along such iterative design processes needs to be addressed in a systematic way, for example when deciding a time budget for the different functions, one must anticipate the uncertainties imposed by future functions that may affect the overall time aspects of the system.

Actors: Function owners, function developers, system integrators

Stakeholders: Function owners and developers.

Status: Approved

Author: Alejandro Cortes

Explanation: None

Type: UseCase

Relations: VTEC#0005 - Access right to TIMMO model

VTEC#0018 - Reduction of design iteration

CAG#0021 - Virtual integration (timing)

VTEC#0012 - Identification of design decision

VTEC#0013 - Effect of a selected solution

VTEC#0017 - Identification of dependency

VTEC#0011 - Support of multiple solutions

VTEC#0016 - Definition of dependency

CAG#0037 - EAST-ADL XML

VTEC#0004 - Timing budget negotiation between OEM and supplier

Develop Engine Management System On Implementation (AUTOSAR) Level

Develop Cruise Control following Top-down Approach

CAG#0038 - Timing Analyses

CAG#0020 - Revising timing constraints

UC#0005 - Develop Control Applications

Name: UC#0005 - Develop Control Applications

Alias: UC#0005

Description: **Use Case Title/Name:** Develop Control Applications

Objective/Goal: Developers of automotive control programs use TADL to specify both continuous time characterizations of the abstract controller and the discrete-time characterization of the implementation.

Description: In control engineering, the controller is usually designed using continuous or discrete time methods without considering the implementation and final deployment. In real implementation, various delays caused by computation time, resource contention, communication, and so on, may violate timing constraints and deteriorate the control performance.

Consequently TADL should be able to describe the timing requirements of the original controller and maintain the traceability between the controller and its implementation. For the original controller, TADL should support the description of its timing properties, e.g., settling time, rise time, allowable sampling period, etc. To account for the inevitable delays caused by implementation, the allowable delays within the control loop should also be

captured in the TADL model. These high-level timing requirements on the control application will be converted to the timing requirements on the implementation components. Such information includes, for instance, WCET, computation deadline, maximal end-to-end delay, etc.

A high-level control design can be decomposed to individual software components in many ways and the components can be allocated to the ECUs in different ways. The decomposition and allocation significantly influence the timing performance of the control application. While subject to practical constraints, the possible combinations may still be numerous. It is an interesting topic for TIMMO2 to study the algorithm for choosing the optimal combination. choosing the optimal combination.

Actors: Function owners, control engineers, software developers, hardware developers

Stakeholders: OEM; Supplier of the control application

Status: Approved

Author: Lei Feng

Explanation: None

Type: UseCase

Relations: CAG#0039 - Sequence Constraint

INRIA#0004 - Functional time

INRIA#0005 - Executable models

VTEC#0024 - Optimization of component mapping

VTEC#0023 - Verification of component mapping

VTEC#0025 - Model integration

VTEC#0022 - Conversion from continuous time to discrete time

VTEC#0027 - Schedulability analysis

VTEC#0019 - Continuous time specifications

VTEC#0021 - Traceability between continuous and discrete time specifications

VTEC#0026 - Internal and external triggers

Explore Design Alternatives for Control Applications

Develop Engine Management System On Implementation (AUTOSAR) Level

Transform Timing Information from Design Level to Implementation Level

Control Scheduling Co-Design with Flexible Timing Structure

Integrate Several Control Applications on a Target Platform

Exchange Timing Information between Control Engineer and Software Engineer

Control Scheduling Co-Design with Fixed Rates

Develop Cruise Control following Top-down Approach

Transform Continuous Time Model to Discrete Time Model

INRIA#0002 - Time bases

INRIA#0001 - Multiform concepts of Time

VTEC#0003 - Methods for estimating WCET at analysis and design levels

VTEC#0020 - Time delays for control applications

BOSCH#0006 - Mode dependencies

UC#0006 - Specify Variability and Timing Information

Name: UC#0006 - Specify Variability and Timing Information

Alias: UC#0006

Description: **Use Case Title/Name:** Specify Variability and Timing Information

Objective/Goal: Variability is an important source for complexity in automotive systems because it leads to a very large number of possible combinations and therefore becomes difficult to handle.

Description: Variability on timing specifications can arise at different abstraction levels. At vehicle level, vehicle configurations are typically defined, each configuration being defined as the features or functions available for the end customer in that particular vehicle configuration. Knowledge on the possible vehicle configurations is often exploited to devise smart design solutions: not all the vehicle functions are present in a given vehicle configuration (that is, no vehicle will be manufactured with all the functions that are possible in that type of vehicle, the end customer cannot freely choose whatever combination, but there are a number of pre-defined vehicle configurations). This implies that it is possible to design the system in such a way that, when considering all the functions, the time budget exceeds 100%, yet the time constraints are fulfilled, simply because we know that there exist vehicle configurations that put constraints on which functions are present on the same vehicle.

It is therefore important to take into account variability, for instance how to capture timing information at the very high levels of abstraction (vehicle level) knowing that vehicle configurations do influence timing at lower levels. For instance, a system has commonly several variants of a specific functionality. This can be implemented as that one or more components are replaceable. The overall timing requirements must be met for each variant and support for specification and analysis at all abstraction levels is necessary.

Actors: Function owners, system architects, function developers

Stakeholders: Function developers; system engineer responsible for integration.

Status: Approved

Author: Henrik Lönn

Explanation: None

Type: UseCase

Relations: VTEC#0030 - Exploitation of vehicle configurations

VTEC#0029 - Configuration at the vehicle level

VTEC#0031 - Scheduling based on vehicle configuration

VTEC#0028 - Timing information and variability

CAG#0036 - Variability

Develop Engine Management System On Implementation (AUTOSAR) Level

Develop Cruise Control following Top-down Approach

UC#0007 - Develop Application and Infrastructure

Name: UC#0007 - Develop Application and Infrastructure

Alias: UC#0007

Description: **Use Case Title/Name:** Develop Application and Infrastructure

Objective/Goal: Developers separately design and implement the applications and the infrastructure. The separation is an effective way to manage the complexity and to enable the easy re-allocation of applications to ECUs. TADL model reflects the separation and also describes the binding effect of the two.

Description: The challenge for this process is to capture timing aspects while keeping the separation of application and infrastructure. For example, the end-to-end latency of an event chain depends on both the application (e.g. the control algorithm) and the infrastructure (e.g. the target hardware). We need a smooth way to bind the application-specific timing information and the infrastructure-specific timing information.

Actors: System architects, software developers, hardware developers, system integrators

Stakeholders: System architects and system integrators; Vendors of platform and/or middleware.

Status: Approved

Author: Lönn Henrik

Explanation: None

Type: UseCase

Relations: VTEC#0034 - Application-independent timing information

VTEC#0037 - Methodology for development

INRIA#0004 - Functional time

CAG#0039 - Sequence Constraint

VTEC#0032 - Infrastructure-independent timing information

Develop Cruise Control following Top-down Approach

Develop Engine Management System On Implementation (AUTOSAR) Level

INRIA#0002 - Time bases

INRIA#0001 - Multiform concepts of Time

VTEC#0035 - Methods for timing characterisation of hardware

VTEC#0036 - Support of flexible timing definition

VTEC#0033 - Methods for timing characterisation of behavior/algorithm

UC#0008 - Exchange Models

Name: UC#0008 - Exchange Models

Alias: UC#0008

Description: **Use Case Title/Name:** Exchange Models

Objective/Goal: Engineers are able to exchange models between different tools.

Description: TADL language works as the universal intermediate format. The timing information in other model formalisms can be extracted and automatically transformed into the TADL model and the TADL model can be transformed into other model formalisms for analysis and testing. The transformation must be done in such a way that the existing timing specifications are preserved.

Actors: System architects, function developers, system integrators, system testers

Stakeholders: System integrators.

Comment: This use case is closely related to UC#0001 and UC#0008.

Status: Approved

Author: Thomas Söderqvist

Explanation: None

Type: UseCase

Relations: CAG#0029 - Exchange a component

CAG#0031 - HW/SW Co-design (Methodology)

CAG#0032 - HW/SW Co-design (Language)

INRIA#0005 - Executable models

VTEC#0038 - Tool integration

CAG#0015 - Assumptions on target systems

CAG#0037 - EAST-ADL XML

CAG#0003 - Age constraint on events

VTEC#0040 - EAST-ADL compliance

VTEC#0039 - AUTOSAR compliance

Develop Cruise Control following Top-down Approach

Develop Engine Management System On Implementation (AUTOSAR) Level

CAG#0004 - Synchronization constraint on events

UC#0009 - Perform Post-Build Parameterization

Name: UC#0009 - Perform Post-Build Parameterization

Alias: UC#0009

Description: **Use Case Title/Name:** Perform Post-Build Parameterization

Object/Goal: The developer must fill in a large number of system parameters for implementation, and these parameters such as the size of transmit and receive buffers and the configuration of network affect the timing performance. It is however not easy to choose the right values and is also a tedious job to manually fill in them. TIMMO-2-USE should provide tools for conveniently setting up system parameters.

Description: With the post-build feature, AUTOSAR allows parameterization on all levels of implementation, including functional features, routing tables, ECU and network characteristics. TIMMO-2-USE should study how to model these parameters using TADL. The optimal values are better determined by a dedicated tool and the values in the model can be automatically transferred

to the AUTOSAR development tool with minimal human effort.

Because of the large number of parameters and their profound effect, TIMMO-2-USE can set limitations on what level of parameters TADL should handle. Allowing post-build for all possible parameters is unrealistic. A modest objective is to identify the parameters in the model and allow their values to be automatically transferred to the AUTOSAR implementation. The algorithm and tool to decide the optimal values of these parameters can be investigated by TIMMO-2-USE.

Actors: Function developers, system integrators

Stakeholders: System integrators.

Status: Approved

Author: Robert Karlsson

Explanation: None

Type: UseCase

Relations: VTEC#0041 - Model parameters

VTEC#0042 - Automatic model reuse

VTEC#0043 - System parameters

VTEC#0013 - Effect of a selected solution

UC#0010 - Specify Synchronization Timing Constraints

Name: UC#0010 - Specify Synchronization Timing Constraints

Alias: UC#0010

Description: **Use Case Title/Name:** Specify Synchronization Timing Constraints

Objective/Goal: An application may have synchronization requirements on the arrival time or age of multiple events from distinct sources and routes. Failure of the synchronization requirement may jeopardize the function of the application.

Description: A typical application with this synchronization requirement is the Electronic Stability Control (ESC). ESC continuously monitors the slipping conditions of the wheels. The signals from all wheels must represent the conditions of the wheels at the same time. Owing to the disturbance in the ECU and the network, one or more wheel slip signals might be delayed and the time synchronization is not preserved at ESC. The consequence of this is that the stability actions are not the optimal. In addition, along the other signal flow direction, the actuation signals from ESC to the wheels must also be synchronized. TADL must support the engineer to specify and analyze synchronization timing constraints to prevent such problems.

Actors: Control engineers, function developers, system testers

Stakeholders: Function developers.

Status: Approved

Author: Thomas Söderqvist

Explanation: None

Type: UseCase

Relations: CAG#0039 - Sequence Constraint

VTEC#0044 - Analysis of the synchronization constraint
VTEC#0046 - Methodology for synchronization issues
VTEC#0045 - Signal age
CAG#0027 - Synchronization constraint per runnable entity
Develop Cruise Control following Top-down Approach
Develop Engine Management System On Implementation (AUTOSAR) Level
INRIA#0003 - Timing expressions
INRIA#0001 - Multiform concepts of Time
CAG#0004 - Synchronization constraint on events

UC#0011 - Specify Probabilistic Timing Properties

Name: UC#0011 - Specify Probabilistic Timing Properties

Alias: UC#0011

Description: **Use Case Title/Name:** Specify Probabilistic Timing Properties

Objective/Goal: Timing properties and constraints may not be deterministic. They can be given as probabilistic values with distribution functions. Stakeholders need to describe and analyze systems with such timing properties.

Description: Probabilistic timing properties are often given for and even preferred by soft real-time applications, where certain amount of constraint violations are acceptable.

Deterministic timing properties, e.g. WCET and deadline, are critical for hard real-time systems; however, the majority of the applications are soft, i.e., certain amount of constraint violations are acceptable. Timing properties of these soft real-time applications may be given as probabilistic values of certain distribution functions. The safety constraint need only to be guaranteed with a probability. This relaxed safety requirement admits tremendous flexibility to stakeholders.

TADL shall allow developers to describe such probabilistic timing properties of events and event chains. The safety constraints of the system should then be associated to probabilities. For example, the end-to-end delay of an event chain must be smaller than 10 ms in 99% of the cases.

Methods and tools for analyzing timing properties must be adapted. For example, the schedulability test cannot only return true or false. The answer should be the probability of the schedulability.

Development methodology must be adapted to allow the new type of specifications and analysis.

Actors: Function owners, function developers, system testers

Stakeholders: OEMs, suppliers, and all function developers.

Status: Approved

Author: Lei Feng

Explanation: None

Type: UseCase

Relations: TUBS#0002 - Uncertain parameters

VTEC#0048 - Analysis of probabilistic timing specifications

VTEC#0047 - Specification of probabilistic timing information

TUBS#0004 - Obtain uncertain timing information

VTEC#0049 - Methodology to work with probabilistic timing specifications

TUBS#0003 - Timing analysis using uncertain timing information

TUBS#0001 - Uncertainty

UC#0012 - Integrate Re-usable Component

Name: UC#0012 - Integrate Re-usable Component

Alias: UC#0012

Description: **Use Case Title/Name:** Integrate Re-usable Component

Objective/Goal: A component is integrated into an existing system on all levels of abstraction.

Description: A re-usable functionality is integrated into an existing system. This results in integrating the corresponding artifacts on all EAST-ADL levels of abstraction:

- Vehicle Level: A Vehicle Feature is integrated into an existing Technical Feature Model.
- Analysis Level: An Analysis Function [Prototype] is integrated into an existing Functional Analysis Architecture.
- Design Level: A Design Function [Prototype] and/or Hardware Component [Prototype] is integrated into an existing Functional Design Architecture and/or Hardware Design Architecture respectively.
- Implementation Level: An AUTOSAR Software Component [Prototype] is integrated into an existing AUTOSAR Virtual Function Bus model. (This case is already addressed by the specific use case "Integrate Re-usable SW-Component").

Since the component subject to be integrated is an existing and re-usable component the assumption is that on all EAST-ADL levels of abstraction the types of the components already exist.

Actors: System architects, function developers, system integrators, system testers

Stakeholders: System integrators.

Comment: The specific use case "Integrate Re-usable SW-Component" is related to this main use case and it describes the the integration of a AUTOSAR Software Component on the EAST-ADL Implementation Level.

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations:

UC#0013 - Specify System Dimensions

Name: UC#0013 - Specify System Dimensions

Alias: UC#0013

Description: **Use Case Title/Name:** Specify System Dimensions

Objective/Goal: ...

Description: ...

Actors: ...

Stakeholders: ...

Comment: This use case originated from a discussion the OEM Advisory Board.

Status: Proposed

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations:

1.2 Specific Use Cases

Capture, Analyze, and Utilize Uncertain Timing Information

Name: Capture, Analyze, and Utilize Uncertain Timing Information

Alias:

Description: **Objective/Goal:** Capture, analyze, and utilize uncertain timing information during the development process

Description: Traditional scheduling algorithms and analysis methods (e.g. for processor utilization or response time), provide deterministic timing guarantees (i.e., all task instances meet their deadline) which take into account worst-case scenarios that may be very rare in practice. This is needed in hard real-time systems but too restrictive for soft real-time systems --- and even for some hard real-time systems where the application allows for a given failure rate (e.g. the probability of missing a deadline could be as small as the probability of hardware failure). One solution to this issue is to analyze the system under uncertainty, while ensuring that the deadline miss ratios predicted by the approximated analysis are greater than (or equal to) the real ones.

This use case addresses the following topics:

- How to capture the uncertain timing information (methodology) and how to describe this information in a formal way (language)?
- How to analyze the captured uncertain timing information (methodology) and what statements can be made on the results of this analysis?
- How and what conclusions can be drawn from the uncertain timing information in order to utilize them in different phases of the development process (methodology)?

Actors: Timing Analyst(s), Timing Expert(s)

Stakeholders: VFM Architects, FAA Architects, [FDA | HDA | MWA] Architects, [VFB | System | ECU | Component] Architects

For more details about the mentioned roles refer to the TIMMO deliverable D7.

Originator: Symtavision

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations:

Capture, Analyze, and Utilize Worst Case Timing Information

Name: Capture, Analyze, and Utilize Worst Case Timing Information

Alias:

Description: **Objective/Goal:** Utilize worst case timing information during the development process

Description: The primary purpose of this use case is to deal with worst case timing information throughout an development process. The main questions are:

- How to capture the worst case timing information (methodology) and how to describe this information in a formal way (language)?
- How to analyze the captured worst case timing information (methodology) and what statements can be made on the results of this analysis?
- How and what conclusions can be drawn from the worst case timing information in order to utilize them in different phases of the development process (methodology)?

Actors: Timing Analyst(s), Timing Expert(s)

Stakeholders: VFM Architects, FAA Architects, [FDA | HDA | MWA] Architects, [VFB | System | ECU | Component] Architects

For more details about the mentioned roles refer to the TIMMO deliverable D7.

Originator: Symtavision

Status: Approved

Author: Kai Richter

Explanation: None

Type: UseCase

Relations:

Develop Body Controller following Top-down Approach

Name: Develop Body Controller following Top-down Approach

Alias:

Description: **Objective/Goal:** The goal of the use case is to derive the timing requirements of a body controller from the requirements specification until the control application level.

Description: Specific scenarios of a real-world case study will be used to validate the results from work package 2 to work package 4 on the use of timing information across different abstraction levels respectively development phases, with particular emphasis on efficient collaboration between system, software and control engineers.

This use case will address the following topics, among others:

- How to derive (transform) the timing requirements following the EAST-ADL methodology and how to describe this information in a formal way (language)?
- How to analyze and validate the timing requirements in models?
- How to enable co-engineering between system, software and control engineers (exchange information between tools)?
- How to address the safety aspect?

Actors: System Architect, Function Architect, Software Designer, Control engineer, Timing Analyst

Stakeholders: System integrators, Supplier of the control application

Originator: Delphi

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations:

Develop Cruise Control following Top-down Approach

Name: Develop Cruise Control following Top-down Approach

Alias:

Description: At the moment no description is available, because the work package 1 could not draw a conclusion on the objective/goal of this use case. The decision has been postpone to the beginning of the work packages 2 through 5.

Objective/Goal: ...

Description: ...

Actors: ...

Stakeholders: ...

Originator: Continental Automotive

Status: Approved
Author: Stefan Kuntz
Explanation: None
Type: UseCase
Relations: CAG#0035 - Task synthesis
CAG#0039 - Sequence Constraint
CAG#0009 - Scheduling Analysis
CAG#0011 - Time bases relation
CAG#0010 - Time bases
CAG#0038 - Timing Analyses
CAG#0025 - Safety (timing)

Develop Engine Management System On Implementation (AUTOSAR) Level

Name: Develop Engine Management System On Implementation (AUTOSAR) Level

Alias:

Description: **Objective/Goal:** Validate the results of work package 2 through 4 in the context of developing a combustion engine management systems and/or specific scenarios of this development

Description:

#1: Validation

The use case shall demonstrate how the results from work package 2 through 4 are applied respectively utilized in the software/hardware development on the EAST-ADL implementation level (AUTOSAR) and assesses their applicability.

In order to obtain reasonable results a case study is conducted using a real-world example Combustion Engine Management System.

During the course of the TIMMO-2-USE project possible results, specifically from work package 2, 3, and 4) are validated using this example and the applicability of these results are assessed and demonstrated.

Note: Indeed, the use case does not cover the entire development of a combustion engine management system, but identifies specific scenarios within this development where the application of the results are obvious and leads to an improvement of the development.

#2: Timing Requirements/Timing Constraints and Timing Properties

When developing according to the EAST-ADL methodology the timing requirements that shall be considered during the Implementation Phase (AUTOSAR) are given as a result from the Design Phase in the first place. Since AUTOSAR provides different views on the software system (Virtual Function Bus, System, ECU, and Component) the question to be answered is how the timing requirements are reflected in the various AUTOSAR timing views and how these are handled in subsequent steps during the implementation phase.

Actors: [Architect | Designer | Implementer | Tester | Integrator | Timing

Analyst | Timing Expert] [VFB | System | ECU | Component]
For more details about the mentioned roles refer to the TIMMO deliverable D7.

Stakeholders: Customer

Originator: Continental Automotive

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations: CAG#0008 - Multi-Core

CAG#0039 - Sequence Constraint

CAG#0016 - Use of AUTOSAR timing views

CAG#0035 - Task synthesis

CAG#0024 - Multi-Core (Scheduling Analysis)

CAG#0022 - Transition from DL to IL

CAG#0026 - Age constraint per runnable entity

CAG#0009 - Scheduling Analysis

CAG#0005 - Hardware

CAG#0027 - Synchronization constraint per runnable entity

CAG#0011 - Time bases relation

CAG#0038 - Timing Analyses

CAG#0010 - Time bases

CAG#0014 - Composition of runnable entities

CAG#0025 - Safety (timing)

Exchange Timing Information between Control Engineer and Software Engineer

Name: Exchange Timing Information between Control Engineer and Software Engineer

Alias:

Description: **Objective/Goal:** The goal of this use case is to enable control engineers and software engineers to exchange timing information in order to perform co-engineering. This includes in particular extending the TIMMO-2-USE TADL.

Description: In order to perform the co-engineering use cases described in [Exploration of Design Alternatives for Control Applications](#), [Control Scheduling Co-Design with Fixed Rates](#), [Integration of Several Control Applications on the Target Platform](#) and [Control Scheduling Co-Design with Flexible Timing Structure](#) the control engineer and the software engineer need a means for exchanging timing information. The concrete information that shall be exchanged has to be determined by these use cases.

Actors: Control Engineer, Function Developer, Software Engineer

Stakeholders: OEM and supplier
Status: Approved
Author: Stefan Kuntz, Arne Hamann
Explanation: None
Type: UseCase
Relations: BOSCH#0003 - Tracing of control timing requirements
BOSCH#0004 - Collaborative Engineering of Control Applications
BOSCH#0009 - Specification of events in the continuous environment
Integrate Several Control Applications on a Target Platform
Explore Design Alternatives for Control Applications
BOSCH#0008 - Concepts of Time
BOSCH#0001 - Control Timing Requirements
BOSCH#0007 - Explicit and implicit events

Exchange Timing Information with worst case verification tool

Name: Exchange Timing Information with worst case verification tool
Alias:
Description: **Objective/Goal:** Bidirectional exchange of timing information between the tools that are used to design or to describe a system and the tools that are used to verify timing constraints based on worst-case analysis.

Description: Several specialized tools are used for different purposes at different steps of the development process. In the descending branch of the V-development cycle, timing constraints can be verified through worst-analysis at several levels (functional design, implementation). In order to be able to perform the analysis, the description of the system and the constraints to be verified need to be exported from some system description tool to the specialized verification tool. And the results of the analysis need to be exported back to the architecture description tool for tracing purposes.

Actors: System Architect, Function Architect, Software Designer
Stakeholders: Automotive OEMs, Suppliers
Originator: RealTime-at-Work

Status: Approved
Author: Jörn Migge
Explanation: None
Type: UseCase
Relations:

Explore Design Alternatives for Control Applications

Name: Explore Design Alternatives for Control Applications

Alias:

Description: **Objective/Goal:** The goal of this requirement is to enable the design team to explore alternative software realization for a given control task at hand.

Description: Computer-based control theory is based on equidistant sampling and negligible input-output latencies that can be ignored. However, in reality execution times vary due to preemption, blocking, data-dependencies, caches, pipelines, network communication, etc. This results in sampling interval jitter as well as non-negligible and varying latencies. To solve this problem, software solutions for control tasks must be co-engineered between control and software engineers.

The solution space for the software realization of a given control task is vast. Thereby, the chosen solution influences on the one hand the control performance, and on the other hand the overall timing performance of the system.

1. The control engineer prefers small sampling and execution rates to achieve close-to-optimal control performance. This, however, leads to high system load, and consequently high system cost (in terms of hardware).
2. The software engineers prefers large sampling and execution rates to increase the composability and extensibility of the system. This, however, leads to decreased control performance.

This conflict of objectives spans the co-design area shown in the picture below. Extending the TIMMO-2-USE TADL and methodology to systematically explore the trade-off between control quality and composability is the main aim of this use case.

Actors: control engineer, function developer, system architect

Stakeholders: OEM and supplier

Status: Approved

Author: Arne Hamann

Explanation: None

Type: UseCase

Relations: BOSCH#0010 - Methodology for timing design of control applications

CAG#0007 - Use of SystemC

BOSCH#0002 - Solution dependent and solution independent timing requirements

Control Scheduling Co-Design with Flexible Timing Structure

Control Scheduling Co-Design with Fixed Rates

Integrate Several Control Applications on a Target Platform

BOSCH#0004 - Collaborative Engineering of Control Applications

Control Scheduling Co-Design with Fixed Rates

Name: Control Scheduling Co-Design with Fixed Rates

Alias:

Description: **Objective/Goal:** The goal of this use case is to enable the design team to realize a control application on a target system using an operating system with fixed rates, such as OSEK or AUTOSAR-OS.

Description: This use case describes one context for the parent use-case [Exploration of Design Alternatives for Control Applications](#). Here the goal is to realize the control application on an operating system offering periodic tasks and processes. Examples for such operating systems are OSEK and AUTOSAR-OS.

Actors: control engineer, function developer, system architect

Stakeholders: OEM and supplier

Status: Approved

Author: Arne Hamann

Explanation: None

Type: UseCase

Relations:

Control Scheduling Co-Design with Flexible Timing Structure

Name: Control Scheduling Co-Design with Flexible Timing Structure

Alias:

Description: **Objective/Goal:** The goal of this use case is to enable the design team to realize a control application on a target system using an operating system that allows flexible time structuring, i.e. going beyond periodic tasks and processes.

Description: This use case describes one context for the parent use-case [Exploration of Design Alternatives for Control Applications](#). Here the goal is to realize the control application on operating systems allowing flexible time structuring. The question to be answered is if better trade-offs between control quality and composability can be achieved compared to operating systems with fixed rates.

Actors: control engineer, function developer, system architect

Stakeholders: OEM and supplier

Status: Approved

Author: Arne Hamann

Explanation: None

Type: UseCase

Relations:

Generate Test bench for Non-functional Properties

Name: Generate Test bench for Non-functional Properties

Alias:

Description: **Objective/Goal:** Definition and analysis of non-functional properties (timing) derived from requirements specifications.

Description: This use case describes generation of a test bench for non-functional properties. This contains the definition and analysis of these properties, focusing on timing, derived from requirement specifications. Therefore a test bench implemented by means of IEEE PSL-Timing specifications transformed from TADL constraints, which are derived from requirement specifications. Hereby, this use case is focusing on one ECU considering AUTOSAR Basic Software Timing and the timing of control applications.

Actors: System Tester

Stakeholders: Requirement Engineer, System Integrator, System Architect

Originator: University Paderborn

Status: Approved

Author: Kay Klobedanz

Explanation: None

Type: UseCase

Relations: UPB#0012 - Black box behavior
UPB#0001 - Abstraction levels
UPB#0019 - Hardware relation
UPB#0022 - Software instruction level
UPB#0006 - Transformation

Handle Timing Information in Simulation Based Analysis Activities

Name: Handle Timing Information in Simulation Based Analysis Activities

Alias:

Description: **Handle Timing Information in Simulation Based Analysis Activities**

Objective/Goal: Describe how timing information is handled in various simulation activities, and how timing information is exchanged with design and implementation tools in a roundtrip development process. It shall be identified which timing information is needed/provided by simulation tools and if the required data can be handled with TADL2 or the AUTOSAR timing extensions.

Description:

Developers perform various simulation activities

- ... to validate timing information, which has been assumed during the design and implementation phase, and
- ... to obtain additional information about timing and resource consumption by measurement on the target platform.

The activities include:

- Validation of time budgets (e.g. reaction time) in HIL and offline simulations.
- Comparison of timing behavior with reference simulations.
- Identifying event chains with critical/suspect timing.
- Measuring timing data and resource consumption on the target processor.
- Automated HIL test series.
- Optimization of the timing by step-wise modification of the implementation.
- Visualization of measured timing, for example, in sequence charts with timing annotation.

The tools applied in this activities are:

- Offline and HIL simulators,
- Experimentation and test automation tools,
- Profilers and Debuggers,
- Special tools for analysis and visualization

These tools consume timing data (reference data) and they provide new measured timing data which must be interchanged with other tools in a roundtrip development process.

Actors: Integrator, Tester, Implementer, Timing Analyst

Stakeholders: OEM, Supplier

Originator: dSPACE

Status: Approved

Author: Ulrich Kiffmeier

Explanation: None

Type: UseCase

Relations:

Integrate Re-usable SW-Component

Name: Integrate Re-usable SW-Component

Alias:

Description: **Objective/Goal:** A re-usable SW-Software Component is integrated into an existing system

Description: This use case describes how an available AUTOSAR Software Component is integrated into an existing system (VFB), which means that all required steps before the integration, during the integration, and after the integration are described.

In particular, the use case focuses on the timing information required to be present and exchanged between the customer and supplier in the mentioned three steps.

The major topic in this use case is that not only the system, a software component is integrated into, imposes requirements on the software component; but also the software component, to be integrated into a system, imposes requirements on this system. In other words, the use case raises the question how are assumptions taken from the software component's view are

described using a language and what kind of assumptions shall be described in order to support: a) the selection of a component, and b) the integration of the software component.

Actors: System Integrator (AUTOSAR Role)

Stakeholders: OEM, First Tier Supplier

Originator: DENSO - Now taken care by dSPACE and INCHRON

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations:

Integrate Several Control Applications on a Target Platform

Name: Integrate Several Control Applications on a Target Platform

Alias:

Description: **Objective/Goal:** The goal of this use case is to extend the TIMMO-2-USE TADL and methodology to support the integration of several control applications on the same hardware platform.

Description: This use-case is closely related to the use case [Exploration of Design Alternatives for Control Applications](#). However, here the focus lies on the integration of several control applications considering their interdependencies. For instance, it might be necessary to "widen" the sampling rate and execution rate of one control application to be able to accommodate another application on the same ECU. Thereby, the same trade-off between control quality and composability has to be considered.

Actors: control engineer, function developer, system architect

Stakeholders: OEM and supplier

Status: Approved

Author: Arne Hamann

Explanation: None

Type: UseCase

Relations: BOSCH#0004 - Collaborative Engineering of Control Applications
CAG#0021 - Virtual integration (timing)
BOSCH#0010 - Methodology for timing design of control applications
CAG#0028 - Integrating a component
CAG#0008 - Multi-Core
CAG#0037 - EAST-ADL XML
CAG#0015 - Assumptions on target systems
CAG#0051 - Reuse of timing constraints

Perform Timing Analysis On Code-Level

Name: Perform Timing Analysis On Code-Level

Alias:

Description: **Objective/Goal:** Determine WCET of non-interrupted tasks as input for system-level verification of timing properties.

Description: Deterministic timing properties, e.g. WCET and deadline, are critical for hard real-time systems.

WCET can be determined by timing tools **aiT** and **TimingExplorer** by AbsInt on the **Implementation Level**.

Timing analysis is performed on compiled executables for concrete hardware. For reliable and not too pessimistic results the WCET analysis needs a number of configuration parameters, which can be grouped as

- **hardware parameters** (cache, ECU configuration, etc) - for Implementation Level only
- **software parameters** (loop bounds, etc) - for Implementation Level only
- **system-level parameters** (software modes) - for Design, Analysis and Implementation Levels

Some of parameters are optional and are used for better precision, others are mandatory.

Relation to Work Packages

This use case is related to the following work packages.

WP2 (Language)

- Extensions of TADL for specification of system-level parameters like modes.

WP3 (Algorithms & Tools)

- Identify hardware parameters that are missing in AUTOSAR and create a separate work package if needed.
- Identify software parameters that should be provided by software developers.
- Identify system-level parameters that should be specified by TADL.
- Perform integration to other tools to get necessary parameters. Integration needs agreement on exchange formats. There are exchange formats already available. Agree on extensions needed to these exchange formats.

WP4 (Methodology)

- Due to necessity of getting potentially confidential information on software implementation for WCET analysis, discuss the probable interaction between software developers and system integrators/testers.

Actors: system integrators, software developers, system testers

Stakeholders: system integrators, software developers, system testers

Status: Approved

Author: Olha Honcharova

Explanation: None

Type: UseCase

Relations: ABS#0010 - Improving precision of WCET analysis by additional parameters (modes, etc)

ABS#0001 - Timing Analysis in Implementation Phase

ABS#0008 - Function Pointers for WCET analysis

ABS#0006 - Loop Bounds for WCET analysis

ABS#0011 - Supported Processor for WCET analysis

ABS#0009 - Volatile Variables for WCET analysis

ABS#0004 - Mapping to Source Code for WCET analysis

ABS#0003 - Executable for WCET analysis

ABS#0013 - Processor-Specific Hardware and Software Settings for WCET analysis

ABS#0002 - Perform Timing Analysis On Code-Level

ABS#0007 - Recursion Bounds for WCET analysis

ABS#0012 - Processor Configuration for WCET analysis

ABS#0005 - Analysis Start Point for WCET analysis

Process Timing Information for HIL-based Simulation

Name: Process Timing Information for HIL-based Simulation

Alias:

Description: **Objective/Goal:** Building up a mixed analysis environment of HIL and restbus simulation.

Description: This use case is related to building up a mixed analysis environment for the integration of a new component into an existing HIL system by means of a restbus simulation. This contains analysis objectives like the modeling and development of control applications considering the system's real-time requirements and behavior as well as definition and validation of timing constraints for the component integration. Therefore, several timing constraints are modeled and examined by means of the TADL events and event chains. These constraints can be, e.g. delay constraints between existing sensors and actuators – with and without restbus simulator integration – or the synchronization between the HIL system and the restbus simulator connected via and/or a implemented as a communication bus, like FlexRay.

Actors: System Integrator, System Architect

Stakeholders: Control Engineer, Function Developer, Software Developer

Originator: University Paderborn

Status: Approved

Author: Kay Klobedanz

Explanation: None

Type: UseCase

Relations:

Specify End-to-End Latencies

Name: Specify End-to-End Latencies

Alias:

Description: **Objective/Goal:** Use multi-form time to express end-to-end latencies

Description: The TIMMO project only considered end-to-end latency values given in the unit of seconds [s]. For example, given data shall be processed within 500 ms. However, in many cases it is more convenient to express timing requirements in physical units, like "... when the temperature increased by 10° C." or "... at position 275° of the crankshaft ...", "... the vehicle shall stop motion after 50 m.", etc.

The use case described the steps to be taken to state such timing requirements and what additional information is required to put it into the specific context. Furthermore, it describes how the multi-form time is translated respectively transformed into a time measured in seconds.

Actors: To be defined.

Stakeholders: Requirements Engineer

Originator: INRIA

Scope: Event- and time triggered communications

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations: INRIA#0004 - Functional time

INRIA#0005 - Executable models

INRIA#0002 - Time bases

INRIA#0003 - Timing expressions

INRIA#0001 - Multiform concepts of Time

Transform Continuous Time Model to Discrete Time Model

Name: Transform Continuous Time Model to Discrete Time Model

Alias:

Description: **Objective/Goal:** The goal of this use case is to extend the TIMMO-2-USE TADL and methodology to support the transformation step between a continuous time model to a discrete time model.

Description: Typically, the control engineer first devices a continuous solution for a given control task. Then, in order to prepare a discrete software solution, the control engineer chooses a sampling rate and a discretization

method (based on information including the analytical time constants of the plant, and the Shannon threshold frequency of the plant). Note that due to these choices several timing requirements can be derived for the discrete software solution.

The software engineer and the system integrator then prepare a software solution and test it in the overall system.

This use case is the precondition to enable round-trip engineering between the control engineer and the software engineer.

For instance, in case of timing constraints violations or poor control quality, the control engineer can choose a different sampling rate or discretization method.

Actors: Control Engineer, Function Developer, Software Developer, System Integrator

Stakeholders: OEM and supplier

Status: Approved

Author: Stefan Kuntz, Arne Hamann

Explanation: None

Type: UseCase

Relations: BOSCH#0010 - Methodology for timing design of control applications

BOSCH#0002 - Solution dependent and solution independent timing requirements

BOSCH#0003 - Tracing of control timing requirements

BOSCH#0011 - Derivation of discrete timing requirements

BOSCH#0009 - Specification of events in the continuous environment

Explore Design Alternatives for Control Applications

CAG#0030 - Distribute jitter

BOSCH#0008 - Concepts of Time

BOSCH#0001 - Control Timing Requirements

BOSCH#0007 - Explicit and implicit events

Transform Timing Information from Vehicle Level to Analysis Level

Name: Transform Timing Information from Vehicle Level to Analysis Level

Alias:

Description: **Objective/Goal:** Transform Vehicle Timing Requirements into Analysis Timing Requirements

Description: During the Analysis Phase the given Vehicle Timing Requirements, besides other functional and non-functional requirements, are the basis for taking decision to create the Functional Analysis Architecture FAA. The primary goal is to satisfy the Vehicle Timing Requirements and in addition to determine the important/relevant timing properties of this "design" - FAA. These timing properties are then transformed into the Analysis Timing Requirements which are a work product passed to the Design Phase.

The purpose of the Analysis Phase is to realize the features specified in the Vehicle Phase and to determine which Analysis Functions are required and

how they shall inter-operate in order to realize these features. The [external] visible behaviour of every Analysis Function is described, as well as their temporal characteristics.

It is important what kind/type of timing information describes the temporal characteristics of the Functional Analysis Architecture and how the timing information is transformed into timing requirements to be considered in the next phase.

Actors: FAA Architect, FAA Designer, FAA Implementer, Timing Analyst, Timing Expert

For more details about the mentioned roles refer to the TIMMO deliverable D7.

Stakeholders: VFM Architect, VFM Timing Expert

Originator: Continental Automotive

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations:

Transform Timing Information from Analysis Level to Design Level

Name: Transform Timing Information from Analysis Level to Design Level

Alias:

Description: **Objective/Goal:** Transform Analysis Timing Requirements into Design Timing Requirements

Description: During the Design Phase the given Analysis Timing Requirements, besides other functional and non-functional requirements, are the basis for taking decision to create the Functional Design Architecture FDA, Hardware Design Architecture HDA, and Middleware Architecture MWA. The primary goal is to satisfy the Analysis Timing Requirements and in addition to determine the important/relevant timing properties of these "designs" - FDA, HDA, MWA. These timing properties are then transformed into the Design Timing Requirements which are a work product passed to the Implementation Phase. The Design Timing Requirements describe the temporal characteristic of the Functional Design Architecture FDA, Hardware Design Architecture HDA, and Middleware Architecture MWA.

The purpose of the Design Phase is to realize the Analysis Functions specified in the Analysis Phase and to determine which Design Functions, Hardware Elements, and Middleware Services are required and how they shall inter-operate in order to realize the Analysis Functions. The internal [and external] behaviour of every Design Function is described, as well as their temporal characteristics.

It is important what kind/type of timing information describes the temporal characteristics of the Functional Design Architecture, Hardware Design Architecture, and Middleware Architecture; and how the timing information is transformed into timing requirements to be considered in the next phase.

In the Design Phase possible distributions of Design Functions in a given system topology are explored.

Actors: [FDA | HDA | MWA] Architect, [FDA | HDA | MWA] Designer, [FDA | HDA | MWA] Implementer, Timing Analyst, Timing Expert
For more details about the mentioned roles refer to the TIMMO deliverable D7.

Stakeholders: FAA Architect, FAA Timing Expert

Originator: Continental Automotive

Status: Approved
Author: Stefan Kuntz
Explanation: None
Type: UseCase
Relations: CAG#0034 - Automation
CAG#0023 - Transition from AL to DL
CAG#0031 - HW/SW Co-design (Methodology)
CAG#0032 - HW/SW Co-design (Language)
CAG#0002 - Event chains between LoA
CAG#0009 - Scheduling Analysis
CAG#0001 - Events between LoA
CAG#0030 - Distribute jitter
CAG#0005 - Hardware
CAG#0013 - Semantics of event chains (connector)
CAG#0012 - Semantics of event chains (component)
CAG#0025 - Safety (timing)

Transform Timing Information from Design Level to Implementation Level

Name: Transform Timing Information from Design Level to Implementation Level

Alias:

Description: **Objective/Goal:** Transform Design Timing Requirements into Implementation Timing Requirements

Description: The very first steps of the Implementation Phase is to transform the Functional Design Architecture, Hardware Design Architecture, and Middleware (functional view) into the corresponding software and hardware architecture - represented by the following views in AUTOSAR:

- Virtual Function Bus View
- Software Component View and Basic Software Module view
- System Topology and ECU Resource Descriptions

During this transformation the given Design Timing Requirements, besides other functional and non-functional requirements, are the basis for taking

decision how to transform/map the elements of the functional domain into the software/hardware domain.

The primary goal is to satisfy the Design Timing Requirements and in addition to determine the important/relevant timing properties of the various AUTOSAR views.

Since there are a lot of different ways in developing the various AUTOSAR views, this use case focuses on the transformation from the Design Level to the AUTOSAR level and generating the VFB view and the System Topology, only (See also EDONA project <http://www.edona.fr>).

Actors: [VFB | System | ECU | Component] [Architect | Designer | Implementer], [VFB | System | ECU | Component] Timing Analyst, [VFB | System | ECU | Component] Timing Expert
For more details about the mentioned roles refer to the TIMMO deliverable D7.

Stakeholders: [FDA | HDA | MWA] Architect, [FDA | HDA | MWA] Timing Expert

Originator: Continental Automotive

Status: Approved

Author: Stefan Kuntz

Explanation: None

Type: UseCase

Relations: CAG#0034 - Automation
CAG#0031 - HW/SW Co-design (Methodology)
CAG#0032 - HW/SW Co-design (Language)
CAG#0022 - Transition from DL to IL
CAG#0002 - Event chains between LoA
CAG#0009 - Scheduling Analysis
CAG#0001 - Events between LoA
CAG#0030 - Distribute jitter
CAG#0005 - Hardware
CAG#0012 - Semantics of event chains (component)
CAG#0013 - Semantics of event chains (connector)
CAG#0025 - Safety (timing)

Verify Timing Constraints

Name: Verify Timing Constraints

Alias:

Description: **Objective/Goal:** Verify timing constraints based on system models

Description: In the descending branch of the V development cycle, the system to be designed (and implemented) is described by models in an increasingly detailed manner. Based on the system models, timing

constraints can be verified through probabilistic and/or worst-case timing analysis. The confidence in the verification and the precision of the analysis increases with the increasing knowledge of system details. On the one hand, only rough estimates of execution times are available at function design level and thus only rough estimates of response time bounds can be obtained by the analysis at that level. On the other hand, very precise Worst Case Execution Time WCET can be obtained when the implementation code is available and thus very precise response time bounds can be computed.

Actors: System Architect, Function Architect, Software Designer

Stakeholders: Automotive OEMs, Suppliers

Originator: RealTime-at-Work

Status: Approved

Author: Jörn Migge

Explanation: None

Type: UseCase

Relations: CAG#0038 - Timing Analyses