ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

# ITEA 2 - 09033


# TIMMO-2-USE

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

| | |
|---|---|
| Report type | Deliverable D1.1 |
| Report name | Requirements |
| Report status | Consortium Confidential |
| Version number | Version 1.0 |
| Date of preparation | 2011-02-21 |

## *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 Technology AB

Dept 6270, M2.7

405 08 Göteborg

Sweden

Tel.: +46 31 322 9949

Email: Daniel.B.Karlsson@volvo.com

## *Authors*

Stefan Kuntz, Continental Automotive GmbH (Editor)

## Document History

| Version | Date | Description |
|---|---|---|
| 1.0 | 2011-02-21 | First version. |

## Table of contents

# 1 Introduction

## Purpose

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

## Scope

This document contains the requirements stated for work package 1 “Use Cases and Requirements” to work package 5 “Validation”.

## Format of Requirements

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

**Name:** This field contains the unique requirement identification (RID) which unequivocally signifies a requirement. It consists of the partner’s abbreviation, for example CAG (capital letters) immediately followed by the number sign ’#’ and immediately followed by a four digit number including leading zeros ’0’. For example CAG#0005 or CAG#0078 or CAG#0987 or CAG#1024.

**Alias:** This field contains the short name of the requirement in order to convey – at least – the particular purpose of the requirement. The field “Name” containing the requirement identification is not very expressive and the alias should help to convey the purpose of the requirement.

**Description:** The field “Notes” contains the sentence that states the specific requirement. In addition it can contain more information providing more background on the requirement, like reason, observation, rationale, etc.

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

**Priority:** This field corresponds with the degree of necessity as described below. The Enterprise Architect predefines three priorities: High, Medium, and Low. In our context *High* corresponds to *Essential*, *Medium* corresponds to *Conditional*, and *Low* corresponds to *Optional*.

**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:** Requirement

**Relations:** In the course of the TIMMO-2-USE project this field is used later to indicate the solution that is associated with the requirement.
In some cases – specifically the requirements that state the necessity of a use case – this field contains references to other requirements.

# 2 Requirements

## 2.1 WP1 – Use Cases and Requirements

### ABS#UC0002 - Perform Timing Analysis On Code-Level

**Name:** ABS#UC0002 - Perform Timing Analysis On Code-Level
**Alias:** Perform Timing Analysis On Code-Level
**Description:** Determine a reasonable value of the WCET of a runnable entity

Implies the necessity of requirements ABS#0003 – ABS#0013.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### CAG#0022 - Transition from DL to IL

**Name:** CAG#0022 - Transition from DL to IL
**Alias:** Transition from DL to IL
**Description:** The methodology shall describe the steps to be taken to transform timing information on the EAST-ADL design level to timing information on the EAST-ADL implementation level.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

### CAG#0023 - Transition from AL to DL

**Name:** CAG#0023 - Transition from AL to DL
**Alias:** Transition from AL to DL
**Description:** The methodology shall describe the steps to be taken to transform timing information on the EAST-ADL analysis level to timing information on the EAST-ADL design level.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

### CAG#0028 - Integrating a component

**Name:** CAG#0028 - Integrating a component
**Alias:** Integrating a component
**Description:** The methodology shall describe the steps to be taken in order to integrate a component in a given system.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0029 - Exchange a component

**Name:** CAG#0029 - Exchange a component
**Alias:** Exchange a component
**Description:** The methodology shall describe the steps to be taken in order to exchange a component between two peers.

Rational: Typically, the exchange of a component is not only passing an artifact/artifacts from one peer to another, but it also involved steps to be taken prior to this exchange (configuration), and after this exchange (validation).
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## TUBS#0002 - Uncertain parameters

**Name:** TUBS#0002 - Uncertain parameters
**Alias:** Uncertain parameters
**Description:** The use cases shall specify uncertain parameter values using mathematical functions.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#UC001 - OEM-Supplier time budgeting

**Name:** VTEC#UC001 - OEM-Supplier time budgeting
**Alias:** OEM-Supplier time budgeting
**Description:** OEM decomposes the overall end-to-end latency to the timing budgets of individual ECUs and network  bus. OEM then assigns these timing budgets to the suppliers.

***UC Name:*** OEM_Supplier_Timing_Analysis
***Source:*** VTEC
***Motivation:*** An E2E function normally spans over several ECUs and across the responsibility of multiple suppliers. OEM needs to divide the overall end-to-end latency to the ECUs and the communication channels, and negotiate these timing budgets with the suppliers.
***Stakeholders:*** Automotive OEM and suppliers.
***Process context:*** The initial negotiation between the OEM and the suppliers to specify the requirements; During the development when the OEM wants to verify or modify the timing budget.
***Product context:*** Any function that requires more than one supplier.
***Organization context:*** The OEM is responsible for the specification and

verification.

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 behavioral) 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 behavioral) 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.

**Status:** Approved
**Priority:** Medium
**Explanation:** None

**Type:** Requirement
**Relations:** VTEC#0003
VTEC#0004

## VTEC#UC002 - Mode-specific timing characterization

**Name:** VTEC#UC002 - Mode-specific timing characterization
**Alias:** Mode-specific timing characterization
**Description:** Developers specify the timing characterization for each running mode of the application.

***UC Name:*** Mode_Specific_Timing_Characterization
***Source:*** VTEC
***Motivation:*** 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.
***Stakeholder:*** Function owner and developer
***Process context:*** Specification and development phases
***Product context:*** All vehicle embedded systems
***Organization context:*** OEMs and suppliers

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.

**Status:** Approved

**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#UC003 - Change Management

**Name:** VTEC#UC003 - Change Management
**Alias:** Change Management
**Description:** Engineers effectively manage the change to the system

***UC Name:*** Change_Management
***Source:*** VTEC
***Motivation:*** Product development is mostly about modifying or improving an existing system with new functionality. Even for new product development, the process consists of several iterations with a lot of modifications between iterations. It is therefore of crucial importance to establish efficient change management.
***Stakeholders:*** OEM and function developers
***Process context:*** Development of new functions; Evolvement of existing functions
**Product context:** Both new and existing functions
***Organization context:*** Function owners and developers

This use case discusses the change management process from the time perspective. Relevant issues to consider are
What other parts of the system (functions/features, ECUs, busses) are affected by a certain change in timing characteristics?
How can suppliers be notified and timing budgets/contracts most conveniently are negotiated again?
When should one notify a change to others and who should receive the notification? Note that notification/change request implies additional cost.
How can one capture several design alternatives in the same model?

**Implied TADL Support and Relevance to TIMMO-2-USE**
The following work packages are related to this use case.
WP2 (Language)
Methodological extensions: Cost estimates and other information needed for the change process.
Semantical reasoning.

WP3 (Algorithms & Tools)
Improved timing information exchange between tools and stakeholders.
Tool support for storing and comparing several design alternatives.

WP4 (Methodology)
An effective collaboration process for managing the change request and maintain the system consistency.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#UC004 - Iterative Design Process

**Name:** VTEC#UC004 - Iterative Design Process
**Alias:** Iterative Design Process
**Description:** The development process of vehicle electronic systems is always iterative.

***UC Name:*** Iterative_Design_Process
***Source:*** VTEC
***Motivation:*** 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.
***Stakeholders:*** Function owners and developers.
***Process Context:*** Introduction of new functions and evolvement of existing ones.
***Product Context:*** All vehicle electronic systems.
***Organization Context:*** OEM and suppliers.

This use case is closely related to the use case of Change Management. 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 having in mind that there are uncertainties related to future functions that might affect the overall time aspects of the system.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#UC005 - Control application development

**Name:** VTEC#UC005 - Control application development
**Alias:** Control application development
**Description:** 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.

***UC Name:*** Control_Application_Development
***Source:*** VTEC
***Motivation:*** Validating and improving TADL's capability of describing the timing requirements of control applications
***Stakeholders:*** OEM; Supplier of the control application
***Process Context:*** Communication between OEM and supplier on the control function
***Product Context:*** Introducing a new feature or improving an existing one
***Organization Context:*** OEM departments who are responsible for control functions; Suppliers who are experts on physical modeling and control engineering

**Implied TADL Support and Relevance to TIMMO-2-USE**

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.

***Comment:*** This use case might be identical to the use cases from Bosch and Continental. If so, we may delete it. Keep.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#UC006 - Variability

**Name:** VTEC#UC006 - Variability
**Alias:** Variability
**Description:** OEM and suppliers use TADL to define the design variation at the vehicle level. This variation is broken down to different SW/HW configurations and the traceability is maintained by TADL.

**UC Name:** Variability
***Source:*** VTEC
***Motivation:*** 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.
***Stakeholders:*** Function developers; system engineer responsible for integration.
***Process context:*** Initial system specification and component allocation; System integration.
***Product context:*** Functions with different requirements and configuration for different vehicle models.
***Organization context:*** OEM and suppliers who maintain multiple versions of vehicle functions.

**Implied TADL Support and Relevance to TIMMO-2-USE**

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.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#UC007 - Separation of Application and Infrastructure

**Name:** VTEC#UC007 - Separation of Application and Infrastructure
**Alias:** Separation of Application and Infrastructure
**Description:** Developers separately design and implement the applications and the platform. TADL model reflects the separation and also describes the binding effect of the two.

***UC Name:*** Separation_of_Application_and_Infrastructure
***Source:*** VTEC
***Motivation:*** The separation of applications and infrastructure is an effective way to manage the complexity and to enable the easy re-allocation of applications to ECUs.
***Stakeholders:*** System architects and system integrators; Vendors of platform and/or middleware.
***Process context:*** Making decisions on embedded system architecture
***Product context:*** System architecture; All aspects of the embedded system development.
***Organization context:*** Departments responsible for platform and architecture.

**Implied TADL Support and Relevance to TIMMO-2-USE**
However, a challenge is to capture timing aspects while keeping the separation of application and infrastructure: an example is latency times which depend both on the application (e.g. the control algorithm) and the infrastructure (e.g. the target hardware). It is needed a smooth way to couple application-specific timing information and infrastructure-specific timing information.

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.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:** VTEC#0003
VTEC#0004

## VTEC#UC008 - Model exchange

**Name:** VTEC#UC008 - Model exchange
**Alias:** Model exchange
**Description:** The engineer exchanges models between different tools.

***UC Name:*** Model_Exchange
***Source:*** VTEC
***Motivation:*** TADL model 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.
***Stakeholders:*** System integrators.
***Process context:*** Deploying software applications on the platform.
***Product context:*** Configurations on the software build and the hardware platform.
***Organization context:*** Department responsible for system integration and configuration.

**Implied TADL Support and Relevance to T2U**

**Comment:** Could be combined with UC001 and UC008.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:** VTEC#0036

## VTEC#UC009 - System parameterization

**Name:** VTEC#UC009 - System parameterization
**Alias:** System parameterization
**Description:** The engineer gets support from tools for conveniently setting up system parameters.

***UC Name:*** Efficient_System_Post-build_Parameterization
***Source:*** VTEC
***Motivation:*** The developer must fill in a large number of system parameters for implementation, and these parameters such as the size of Tx/Rx 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.
***Stakeholders:*** System integrators.
***Process context:*** Deploying software applications on the platform.
***Product context:*** Configurations on the software build and the hardware platform.
***Organization context:*** Department responsible for system integration and configuration.

**Implied TADL Support and Relevance to T2U**

With the post-build feature, AUTOSAR allows parameterization on all levels of implementation, including functional features, routing tables, ECU and network characteristics. T2U 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, T2U 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 T2U.

Comment: this use case is closely related to Change Management.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#UC010 - Synchronization

**Name:** VTEC#UC010 - Synchronization
**Alias:** Synchronization
**Description:** The engineer specifies synchronization constraints on system timing.

***UC Name:*** Synchronization
***Source:*** VTEC
***Motivation:*** 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.
***Stakeholders:*** Function developer.
***Process context:*** Development and V&V phase.
***Product context:*** Advanced function.
***Organization context:*** Suppliers responsible for the function.

**Implied TADL Support and Relevance to T2U**

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.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:** VTEC#0039

## VTEC#UC011 - Probabilistic timing properties

**Name:** VTEC#UC011 - Probabilistic timing properties
**Alias:** Probabilistic timing properties
**Description:** 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.

***UC Name:*** Probabilistic_Timing_Properties
***Source:*** VTEC
***Motivation:*** Probabilistic timing properties are often given for and even preferred by soft real-time applications, where certain amounts of constraint violations are acceptable. For such systems, we need to only guarantee the safety with a probability. This relaxed safety requirement admits tremendous flexibility to stakeholders.
***Stakeholders:*** OEMs, suppliers, and all function developers.
***Process context:*** System specification; the development process; the verification and testing process.
***Product context:*** Non-time-critical applications.
***Organization context:*** All stakeholders

**Implied TADL Support and Relevance to T2U**
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.

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 occasions.

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.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement

**Relations:**

## 2.2 WP2 – Language

### 2.2.1 Constraints

#### BOSCH#0001 - Control Timing Requirements

**Name:** BOSCH#0001 - Control Timing Requirements
**Alias:** Control Timing Requirements
**Description:** The TADL shall be capable of modeling intrinsic timing requirements of the plant to be controlled.
Examples are:
-Analytical time constants of the plant
-Shannon threshold frequency of the plant

Rationale: These timing requirements span the possible design space for a discrete software solution of the controller (e.g. selection of a sampling rate and a discretization method).

Comment: Isn't this the transition from the functional to the SW/HW domain, is it? WP4 shall describe how to accomplish this with the given EAST-ADL / TADL elements. Supporting co-engineering between function and HW/SW development.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

#### CAG#0003 - Age constraint on port

**Name:** CAG#0003 - Age constraint on port
**Alias:** Age constraint on port
**Description:** The language shall provide the capability to specify the age of data on a port.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

#### CAG#0004 - Synchronization constraint on ports

**Name:** CAG#0004 - Synchronization constraint on ports
**Alias:** Synchronization constraint on ports
**Description:** The language shall provide the capability to specify synchronization constraints of data on more than one port.
**Status:** Approved

**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0012 - Semantics of event chains (component)

**Name:** CAG#0012 - Semantics of event chains (component)
**Alias:** Semantics of event chains (component)
**Description:** The language shall specify the semantics of event chains spanning across components (from input ports to output ports).
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0013 - Semantics of event chains (connector)

**Name:** CAG#0013 - Semantics of event chains (connector)
**Alias:** Semantics of event chains (connector)
**Description:** The language shall specify the semantics of event chains spanning connectors (from output ports to input ports).
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0014 - Composability of runnable entities

**Name:** CAG#0014 - Composability of runnable entities
**Alias:** Composability of runnable entities
**Description:** The language shall support the capability to compose runnable entities based on given runnable entities.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0015 - Assumptions on target systems

**Name:** CAG#0015 - Assumptions on target systems
**Alias:** Assumptions on target systems
**Description:** The language shall provide means to specify the assumptions made on possible target systems.

Rational: A given system imposes requirements on a component that is operated in this system. However, a component makes some assumptions on the target system in order to operate properly.
**Status:** Approved
**Priority:** Medium

**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0025 - Safety (timing)

**Name:** CAG#0025 - Safety (timing)
**Alias:** Safety (timing)
**Description:** The language shall provide capabilities with regard to timing in order to address safety concerns.

Rational: Error modeling.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0026 - Age constraint per runnable entity

**Name:** CAG#0026 - Age constraint per runnable entity
**Alias:** Age constraint per runnable entity
**Description:** The language shall be capable of specifying the age constraint of data processed by every runnable entity of a SW-C.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0027 - Synchronization constraint per runnable entity

**Name:** CAG#0027 - Synchronization constraint per runnable entity
**Alias:** Synchronization constraint per runnable entity
**Description:** The language shall be capable of specifying the synchronization constraint of data processed by every runnable entity of a SW-C.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0039 - Sequence Constraint

**Name:** CAG#0039 - Sequence Constraint
**Alias:** Sequence Constraint
**Description:** The language shall provide means to specify sequence constraints.

Rational: The execution order constraint on the implementation level (AUTOSAR, ARTE) is too strict, because it requires that the runnable entities part of the execution order constraint are executed in subsequent order.
**Status:** Approved
**Priority:** Medium

**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0041 - TADL in Modeling Languages

**Name:** CAG#0041 - TADL in Modeling Languages
**Alias:** TADL in Modeling Languages
**Description:** The base concepts of events and event chains shall be expressed in EAST-ADL and SysML. It shall be possible to organize event chains hierarchically along the Function Hierarchy of EAST-ADL or the Block structure of SysML (End2End constraints). Also constraints cross cutting a hierarchical model, as required for synchronization constraints shall be possible. Sample Usages should be worked out.(e.g. in demonstrators)
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0045 - Constraint Language

**Name:** CAG#0045 - Constraint Language
**Alias:** Constraint Language
**Description:** A constraint language is required, in order to be able to do constraint checking. The constraint language should contain basic arithmetic expressions like =,<,>, -, +, and, or, ...It seems not necessary to define a new constraint language, since OCL is available in UML tooling or language specific expressions can be evaluated in runtime environments as C or Modelica. A language neutral expression should be part of an event chain, the related events should be the arguments of the constraint expression.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## INRIA#0004 - Functional time

**Name:** INRIA#0004 - Functional time
**Alias:** Functional time
**Description:** The TADL shall be capable modeling functional relations at the different levels of a design (precedence, causality, concurrency…) independently from a physical time reference.

Comment: Clarify
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## RTAW#0005 - Synchronized schedules

**Name:** RTAW#0005 - Synchronized schedules
**Alias:** Synchronized schedules
**Description:** TADL shall allow describing which schedules of tasks or frames run synchronously or may drift against each other. In the latter case it should be possible to describe the drift speed.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0004 - Delay and jitter

**Name:** UPB#0004 - Delay and jitter
**Alias:** Delay and jitter
**Description:** The Timing Model shall support the specification of delay and jitter conditions

Comment: implemented already in TIMMO
**Status:** Rejected
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:** UC#0001 - Specify Time Budgets

## UPB#0007 - Execution times

**Name:** UPB#0007 - Execution times
**Alias:** Execution times
**Description:** The Timing Model shall support the specifications of execution times (best, average, worst case).

Comment: Already implemented by TIMMO and AUTOSAR.
**Status:** Rejected
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:** UC#0001 - Specify Time Budgets

## UPB#0016 - Network frame modeling

**Name:** UPB#0016 - Network frame modeling
**Alias:** Network frame modeling
**Description:** The Timing Model shall support modeling of network frames and their timing constraints.

Comment: See also UPB#0003
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0017 - Synchronization

**Name:** UPB#0017 - Synchronization
**Alias:** Synchronization
**Description:** The Timing Model shall provide mechanism to describe synchronization.

Rational: It is required to specify the synchronization between for example tasks and bus communications, synchronously sending and receiving signals.

Comment: See time bases UPB#0005 and UPB#0024
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:** UC#0010 - Specify Synchronization Timing Constraints

## UPB#0022 - Software instruction level

**Name:** UPB#0022 - Software instruction level
**Alias:** Software instruction level
**Description:** The Timing Model shall support timing specification at SW instruction level

Comment: See UPB#0020 What means SW instruction level - processor's instruction set and/or C programming language level.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0004

**Name:** VTEC#0004
**Alias:** Support timing characteristics of behavior/algorithm
**Description:** TADL shall support the specification of execution time for an activity/component. Worst case, best case, average case execution time should be supported.

Rationale: In order to conveniently elaborate with different behavior and different algorithms, and in order to get a good timing estimation, it is necessary to have a means to characterize the timing performance of the algorithms used for realizing part of the functionality involved in the event chain of the timing budget.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0006

**Name:** VTEC#0006
**Alias:**
**Description:** The Timing Model shall support different interpretations of the timing information. Interpretations that shall be supported are:

Timing requirements - A requirement imposed to govern a correct timing behavior of a high-level function or application.

Timing constraint - A requirement imposed as part of a design decision.

Timing property - A value measured or derived for an already existing component that is known to hold.

Timing requirements and constraints are imposed as part of a top-down design methodology where they set the limits for a component yet to be developed, e.g., maximum allowed execution time. Timing properties are used in a bottom up approach when components are reused, e.g., worst case execution time.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0007

**Name:** VTEC#0007
**Alias:** Mode-dependent timing constraints
**Description:** TADL shall support different timing constraints on the same event chain depending on system mode.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0019

**Name:** VTEC#0019
**Alias:**
**Description:** TADL must be able to specify continuous-time properties of the control design.

Comment: Representative timing properties are settling time, rise time, allowable sampling period, etc.

**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0020

**Name:** VTEC#0020
**Alias:**
**Description:** TADL must be able to define the allowable time delays within the control loop.

Comment: The delay may be due to longer execution time of a computation, preemption, or communication.

**Status:** Approved

**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0027

**Name:** VTEC#0027
**Alias:**
**Description:** TADL shall support industrial state-of-the-art schedulability analysis methods to be used for verification of timing behavior.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0028

**Name:** VTEC#0028
**Alias:**
**Description:** Timing constraints shall be composable and possible to maintain in a scenario with variable elements
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0032

**Name:** VTEC#0032
**Alias:**
**Description:** TADL shall support infrastructure-independent timing information for applications.

Explanations:
TADL shall support the specification of execution time for an activity/component. Worst case, best case, average case execution time should be supported.

Rationale: In order to conveniently elaborate with different behavior and different algorithms, and in order to get a good timing estimation, it is necessary to have a means to characterize the timing performance of the algorithms used for realizing part of the functionality involved in the event chain of the timing budget.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0041

**Name:** VTEC#0041

**Alias:**
**Description:** TADL shall be able to capture the important implementation parameters in the model.

Comment: These parameters are relevant for timing properties.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0045

**Name:** VTEC#0045
**Alias:**
**Description:** TADL shall support the specification of signal age.

Comment: For example, a component may provide an output derived from a sensor value. This value may at the finish time of the component already be of a certain age, e.g., sensor value read at exec-time/2 and released after some post-processing at exec-time.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

### *2.2.2 Distribution*

## TUBS#0001 - Uncertainty

**Name:** TUBS#0001 - Uncertainty
**Alias:** Uncertainty
**Description:** TADL shall provide the facilities to model relevant timing system parameters with uncertain parameter values and enable the characterization of the uncertainty by means of mathematical functions.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0047

**Name:** VTEC#0047
**Alias:**
**Description:** TADL shall support the specification of probabilistic timing properties and constraints.

Comment: an evident distinction between the deterministic and probabilistic timing specifications must be visible in the model.

**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

### 2.2.3 Events

## BOSCH#0007 - Explicit and implicit events

**Name:** BOSCH#0007 - Explicit and implicit events
**Alias:** Explicit and implicit events
**Description:** The TADL shall be capable of distinguishing explicit and implicit events.

Explicit events: events originating from the use of the plant or events in sequence control, e.g. driver hits brake pedal.

Implicit events: events originating from the physics of the plant or the environment, e.g. back pressure valve switching in dependence of pressure ratio or switch-off heating element when a maximum temperature is reached.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0002 - Event type a-periodic

**Name:** UPB#0002 - Event type a-periodic
**Alias:** Events Types
**Description:** The Timing Model shall support the event type a-periodic and specify the definition of the event type's semantic.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0026

**Name:** VTEC#0026
**Alias:**
**Description:** TADL shall support both internal and external triggering of events and signals. I.e., both software and hardware generated triggers.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### 2.2.4 Exchange

## CAG#0037 - EAST-ADL XML

**Name:** CAG#0037 - EAST-ADL XML
**Alias:** EAST-ADL XML
**Description:** The language shall provide a description of the EAST-ADL XML.

Rational: In order to ensure interoperability and seamless model migration an interchange format shall be specified including the steps to be taken to convert a model from one version of EAST-ADL XML to another, most likely the subsequent one.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0050 - TADL profile for dynamic UML diagrams

**Name:** CAG#0050 - TADL profile for dynamic UML diagrams
**Alias:** TADL profile for dynamic UML diagrams
**Description:** As being the widely recognized standard for modeling of behavior, within TIMMO-2-USE TADL profiles for the dynamic UML diagrams: State Machine diagram, Activity diagram and Sequence diagram shall be developed. These diagrams are also suitable for describing operational scenarios on a high level of abstraction, and thus timing conditions should be part of these models.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0001 - Code level exchange

**Name:** DSP#0001 - Code level exchange
**Alias:** Code level exchange
**Description:** It shall be possible to exchange code level descriptions including necessary meta-information for a timing analysis to be performed on this code.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## INRIA#0005 - Executable models

**Name:** INRIA#0005 - Executable models
**Alias:** Simulation
**Description:** It shall be possible to execute a TADL model to obtain a schedule (partial or totally ordered set of events).

Comment: Is this different from scheduling analysis? What comes first define a

schedule and then analyzing scheduling?
What information contained in the model/s is required for performing such schedule definition?
What does execution means in this context?

**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## RTAW#0002 - Extension of AUTOSAR 4.0

**Name:** RTAW#0002 - Extension of AUTOSAR 4.0
**Alias:** Extension of AUTOSAR 4.0
**Description:** TADL-2 shall be compliant with AUTOSAR 4.0 or higher and adhere to AUTOSAR meta-modeling rules.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0008 - FIBEX Compliance

**Name:** UPB#0008 - FIBEX Compliance
**Alias:** FIBEX Compliance
**Description:** The Timing Model shall be compliant to FIBEX (cluster) timing specifications

Comment: Comparison between AUTOSAR and FIBEX standard. Comparison on the XML schema level and determine missing element, as well as transformations required to exchange information using these standards.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:** UC#0008 - Exchange Models

## UPB#0009 - AUTOSAR compliance

**Name:** UPB#0009 - AUTOSAR compliance
**Alias:** AUTOSAR compliance
**Description:** The Timing Model shall be compliant to AUTOSAR standard 4.#.

Comment: AUTOSAR R4.0.# including ARTE
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0010 - AUTOSAR Timing Extensions

**Name:** UPB#0010 - AUTOSAR Timing Extensions
**Alias:** AUTOSAR Timing Extensions
**Description:** The Timing Model shall be compliant to AUTOSAR Timing Extensions.

Comment: See also UPB#0009
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0029

**Name:** VTEC#0029
**Alias:**
**Description:** TADL shall support the definition of configurations at vehicle level.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0039

**Name:** VTEC#0039
**Alias:**
**Description:** The semantics and syntax of TADL shall be compatible with the AUTOSAR approach.

Comment: That is, meta-meta model (=modeling guidelines) and meta-model (=software component template).
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0040

**Name:** VTEC#0040
**Alias:**
**Description:** The semantics and syntax of TADL shall be based on EAST ADL2 (ATESST, MAENAD) timing requirements wherever applicable.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### 2.2.5 Hardware

## CAG#0005 - Hardware

**Name:** CAG#0005 - Hardware
**Alias:** Hardware
**Description:** The language shall provide the capability to specify timing information covering hardware entities.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0008 - Multi-Core

**Name:** CAG#0008 - Multi-Core
**Alias:** Multi-Core
**Description:** The language shall provide means to describe timing information in the context of multi-core systems.

Rational: It shall be checked whether the TADL language elements are sufficient to specify timing information for multi-core systems. Possibly, the semantics of some language elements need to be sharpened or new elements shall be introduced.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0024 - Multi-Core (Scheduling Analysis)

**Name:** CAG#0024 - Multi-Core (Scheduling Analysis)
**Alias:** Multi-Core (Scheduling Analysis)
**Description:** The language shall provide means to describe the timing information required for performing scheduling analysis for applications executed on multi-core systems.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

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

**Name:** CAG#0032 - HW/SW Co-design (Language)
**Alias:** HW/SW Co-design
**Description:** The language shall provide means to support the HW/SW co-design.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement

**Relations:**

## RTAW#0001 - ECU partitioning

**Name:** RTAW#0001 - ECU partitioning
**Alias:** ECU partitioning
**Description:** T2U shall allow describing the dependencies between safety requirements and ECU partitioning that are defined to meet these requirements.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0003 - Bus communication

**Name:** UPB#0003 - Bus communication
**Alias:** Bus Communication
**Description:** The Timing Model shall support modeling of timing properties for bus communication (e.g. FlexRay, CAN, RTEthernet)

Comment: AUTOSAR compliance and introduction of Ethernet in AUTOSAR Release 4.0
Possibly define more requirements how to related timing properties already defined in the AUTOSAR standard with TADL timing guarantees (WP2 and WP4)
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0011 - Multi-core support

**Name:** UPB#0011 - Multi-core support
**Alias:** Multi-core support
**Description:** The Timing Model shall be applicable to multi-core systems.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0019 - Hardware relation

**Name:** UPB#0019 - Hardware relation
**Alias:** Hardware relation
**Description:** The Timing Model shall be capable of describing timing properties that are related to hardware, like clock rates, sampling rates, bit-duration, etc.

Comment: To what extend shall this be described? EAST-ADL provides on the Design Level Sensor, Actuator, ECU, processor, etc. It shall be checked what AUTOSAR already specifies and which information is missing in order to perform Worst Case Execution Time analysis (cache, ...).

**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0003

**Name:** VTEC#0003
**Alias:** Support timing characterization of hardware
**Description:** TADL shall support generic modeling of timing behavior related to commonly used hardware devices, such as microprocessor, CAN, Flexray, etc.

Rationale: In order to conveniently elaborate with different hardware, and in order to get a good timing estimation, it is necessary to have a means to characterize the timing performance of the hardware devices used in the timing budget.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0034

**Name:** VTEC#0034
**Alias:**
**Description:** TADL shall support application-independent timing information for infrastructure components (e.g. hardware).

Explanations:
TADL shall support generic modeling of timing behavior related to commonly used hardware devices, such as microprocessor, CAN, Flexray, etc.

Rationale: In order to conveniently elaborate with different hardware, and in order to get a good timing estimation, it is necessary to have a means to characterize the timing performance of the hardware devices used in the timing budget.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### *2.2.6 Mode*

## BOSCH#0005 - Mode dependent timing requirements for control applications

**Name:** BOSCH#0005 - Mode dependent timing requirements for control applications
**Alias:** Mode dependent timing requirements for control applications
**Description:** The TADL shall be capable of modeling mode dependent timing requirements

for open and closed loop control applications
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## BOSCH#0006 - Mode dependencies

**Name:** BOSCH#0006 - Mode dependencies
**Alias:** Mode dependencies
**Description:** The TADL shall be capable of modeling dependencies between modes and conditions for mode transitions
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0046 - Mode dependent Application

**Name:** CAG#0046 - Mode dependent Application
**Alias:** Mode dependent Application
**Description:** The systems and applications of one or more demonstrators should be mode dependent, in order to show how to handle modes in the TADL language. The modes could be error modes, system modes or modes on vehicle level (driving profile, energy mode)
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0047 - Mode dependent End2End delay

**Name:** CAG#0047 - Mode dependent End2End delay
**Alias:** Mode dependent End2End delay
**Description:** In order to demonstrate the ability to deal with modes, demonstrator should contain an End-to-End event chain, which is dependent on a certain mode. The constraint is only violated, if the violation happens, the system is in the depending mode. In TADL it has to be defined, if only events, and event chains, or also constraints have to be mode-dependent
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0048 - Mode Switch in Stimulus/Response

**Name:** CAG#0048 - Mode Switch in Stimulus/Response
**Alias:** Mode Switch in Stimulus/Response
**Description:** In a further example dealing with modes, an End2End event chain, where the

stimulus event occurs in a mode different from the mode, where the response event occurs. This is only possible, when between the stimulus event and the response event; one or more mode changes take place. A constraint on the event chain is only violated, if the mode switch takes place at run time, between stimulus and response event.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0049 - Timing Constraint on Mode Switch

**Name:** CAG#0049 - Timing Constraint on Mode Switch
**Alias:** Timing Constraint on Mode Switch
**Description:** Timing conditions on a mode switch shall be expressed by a repetition rate constraint. The constraint shall ensure, that a mode switch must be from mode A to mode B, if the time span between the two mode switches is smaller than a threshold (t), and the mode switch must take place from mode A to mode C, if the time span between the two mode switches is bigger than a threshold (t).
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0018 - Redundancy/safety

**Name:** UPB#0018 - Redundancy/safety
**Alias:** Redundancy/safety
**Description:** The Timing Model shall support specification of redundancy (safety).

Comment: It shall be possible to describe the re-execution of any executable entity, like tasks, runnable entities, functions, components and the related timing. This should be possible for different modes and mode transitions, like normal to error mode transition, etc.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0008

**Name:** VTEC#0008
**Alias:** Mode-dependent task and bus schedule descriptions
**Description:** TADL shall support the description of different task and bus schedules depending on system modes.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

### 2.2.7 Reuse

## CAG#0017 - Reuse of events

**Name:** CAG#0017 - Reuse of events
**Alias:** Reuse of events
**Description:** The language shall provide means to reuse existing events.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0018 - Reuse of event chains

**Name:** CAG#0018 - Reuse of event chains
**Alias:** Reuse of event chains
**Description:** The language shall provide means to reuse existing event chains.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0020 - Revising timing constraints

**Name:** CAG#0020 - Revising timing constraints
**Alias:** Revising timing constraints
**Description:** The language shall provide means to alter the timing constraints of events and event chains being reused.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0051 - Reuse of timing constraints

**Name:** CAG#0051 - Reuse of timing constraints
**Alias:** Reuse of timing constraints
**Description:** The language shall provide means to reuse existing timing constraints.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0011

**Name:** VTEC#0011
**Alias:** Several alternative timing solutions
**Description:** TADL shall support defining several alternative solutions.

Rationale: In the negotiation process with a supplier, it is important to simultaneously capture all alternatives currently being discussed. This will allow both supplier and OEM to efficiently evaluate and compare them.

**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### *2.2.8 Time*

## BOSCH#0008 - Concepts of Time

**Name:** BOSCH#0008 - Concepts of Time
**Alias:** Concepts of Time
**Description:** The TADL shall be capable of distinguishing different concepts of time: solution timing (timing resulting from the chosen solution), environment timing, process timing (physical process of the plant)

Rationale: For system understanding and co-engineering considerations it is important to explicitly model the transition from the continuous world to the discrete SW world.

Comment: See BOSCH#0002

**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## BOSCH#0009 - Specification of events in the continuous environment

**Name:** BOSCH#0009 - Specification of events in the continuous environment
**Alias:** Specification of events in the continuous environment
**Description:** The TADL shall be capable of specifying (implicit) events originating in the continuous environment.
Currently, only crankshaft speed is supported.
In TADL it shall be possible to specify arbitrary events referencing arbitrary (continuous) physical processes.
Example: Event is fired when a maximum temperature is reached (and heating is switched off).

Physical process: change of a continuous signal over time

Comment: WP4 use case is how to obtain time values for a timing constraint.

**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0010 - Time bases

**Name:** CAG#0010 - Time bases
**Alias:** Time bases
**Description:** The language shall provide capabilities to specify multiple/several time bases.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## INRIA#0001 - Multiform concepts of Time

**Name:** INRIA#0001 - Multiform concepts of Time
**Alias:** Multiform concepts of Time
**Description:** The TADL shall be capable of modeling time evolution on any repetitive event (camshaft, meters, seconds, etc.).

Comment: different time domains
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## INRIA#0002 - Time bases

**Name:** INRIA#0002 - Time bases
**Alias:** Time bases
**Description:** The TADL shall be capable of expressing in a common model different time bases.

Comment: See UPB requirement on time bases
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## INRIA#0003 - Timing properties

**Name:** INRIA#0003 - Timing properties
**Alias:** Timing properties
**Description:** The TADL shall be capable of expressing complex constraints based on different time bases.

Comment: See UPB requirements on time bases
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0005 - Global time base

**Name:** UPB#0005 - Global time base
**Alias:** Global time base
**Description:** The Timing Model shall support the specification of a global time base.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:** UC#0010 - Specify Synchronization Timing Constraints

## UPB#0014 - Time- and event triggering

**Name:** UPB#0014 - Time- and event triggering
**Alias:** Time- and event triggering
**Description:** The Timing Model shall support time- and event-triggered mechanisms.

Comment: Is already implemented in TIMMO.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0024 - Relationship between time bases

**Name:** UPB#0024 - Relationship between time bases
**Alias:** Relationship between time bases
**Description:** The Timing Model shall support describing relationships between various time bases.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

### *2.2.9 Tracing*

## BOSCH#0002 - Solution dependent and solution independent timing requirements

**Name:** BOSCH#0002 - Solution dependent and solution independent timing requirements
**Alias:** Solution dependent and solution independent timing requirements
**Description:** The TADL shall be capable of distinguishing timing requirements originating in the problem space (e.g. from plant to be controlled see BOSCH#0001) and solution dependent timing requirements (e.g. from the discrete software solution for the control task).
**Status:** Approved
**Priority:** High

**Explanation:** None
**Type:** Requirement
**Relations:**

## BOSCH#0003 - Tracing of control timing requirements

**Name:** BOSCH#0003 - Tracing of control timing requirements
**Alias:** Tracing of control timing requirements
**Description:** The TADL shall be capable of tracing (solution dependent) timing requirements to 1) the chosen solution (e.g. discretization method), and 2) the underlying intrinsic timing requirement originating in the problem space (see Bosch#0001).

Comment: WP4
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0001 - Events between LoA

**Name:** CAG#0001 - Events between LoA
**Alias:** Events between LoA
**Description:** The language shall provide means to specify the relation between events on different levels of abstraction.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0002 - Event chains between LoA

**Name:** CAG#0002 - Event chains between LoA
**Alias:** Event chains between LoA
**Description:** The language shall provide means to specify the relation between event chains on different levels of abstraction.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0011 - Time bases relation

**Name:** CAG#0011 - Time bases relation
**Alias:** Time bases relation
**Description:** The language shall provide capabilities to relate different time bases.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement

**Relations:**

## DSP#0004 - Target processor dependence

**Name:** DSP#0004 - Target processor dependence
**Alias:** Target processor dependence
**Description:** It shall be possible that code level timing analysis according to DSP0002 can be performed for a required target processor.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0001 - Abstraction levels

**Name:** UPB#0001 - Abstraction levels
**Alias:** Abstraction Levels
**Description:** The Timing Model shall define and handle several levels of abstraction as defined by the EAST-ADL.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0012 - Black box behavior

**Name:** UPB#0012 - Black box behavior
**Alias:** Black box behavior
**Description:** The Timing Model shall support description for black-box timing behavior on the interface of a component.

Rational: Sometimes it is necessary to hide the internal of a component due to IP reasons respectively due to abstraction purposes.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0023 - AUTOSAR views

**Name:** UPB#0023 - AUTOSAR views
**Alias:** AUTOSAR views
**Description:** The language shall support the AUTOSAR views, like System, Virtual Function Bus, ECU, Component.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0001

**Name:** VTEC#0001
**Alias:**
**Description:** TADL shall support the traceability between communication timing on implementation level and connector timing on design level

Requirement 30 from TIMMO:
The Timing Model shall support the traceability (from upper abstraction levels) of a timing requirement/constraint through the different abstraction layers.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0002

**Name:** VTEC#0002
**Alias:** Timing budget support
**Description:** TADL shall support dividing an end-to-end delay constraints into smaller pieces (time budgeting), where the smaller pieces can be assigned to stakeholders for implementation.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0012

**Name:** VTEC#0012
**Alias:** Decision of timing solution
**Description:** TADL shall support the clear identification of the final decision/agreement of one of several proposed alternative solutions.

Rationale: When a timing budget has been finally agreed with a set of suppliers, the agreed alternative shall be marked so that it is possible to see the final decision. The alternatives shall not be deleted, in the case there will be a renegotiation.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0013

**Name:** VTEC#0013
**Alias:** Effect of a timing solution
**Description:** TADL shall support the description of the consequences of a certain solution on the system.

Rationale: An effect can relate to bus and task schedules and their implications

on timing performance of the system.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0016

**Name:** VTEC#0016
**Alias:**
**Description:** TADL shall define the dependency on timing properties between functions and components.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0021

**Name:** VTEC#0021
**Alias:**
**Description:** TADL must maintain the traceability between the continuous timing properties of the control design and the discrete-time specifications on the software implementation
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### *2.3 WP3 – Algorithms and Tools*

## ABS#0003 - Executable for WCET analysis

**Name:** ABS#0003 - Executable for WCET analysis
**Alias:** Executable for WCET analysis
**Description:** WCET analysis requires a compiled linked executable file.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0004 - Mapping to Source Code for WCET analysis

**Name:** ABS#0004 - Mapping to Source Code for WCET analysis
**Alias:** Mapping to Source Code for WCET analysis
**Description:** WCET analysis requires the following information in order to map the analysis results to the source code:
- availability of source code,

- either debug information in the executable or a map file with information linking source code and binary code.

**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0005 - Analysis Start Point for WCET analysis

**Name:** ABS#0005 - Analysis Start Point for WCET analysis
**Alias:** Analysis Start Point for WCET analysis
**Description:** WCET analysis requires a start point as an address or a routine name
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0006 - Loop Bounds for WCET analysis

**Name:** ABS#0006 - Loop Bounds for WCET analysis
**Alias:** Loop Bounds for WCET analysis
**Description:** WCET analysis requires information on loop bounds that cannot be found by automatic analysis.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0007 - Recursion Bounds for WCET analysis

**Name:** ABS#0007 - Recursion Bounds for WCET analysis
**Alias:** Recursion Bounds for WCET analysis
**Description:** WCET analysis requires information on bounds for recursion.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0008 - Function Pointers for WCET analysis

**Name:** ABS#0008 - Function Pointers for WCET analysis
**Alias:** Function Pointers for WCET analysis
**Description:** The Code-Level Timing Analysis requires information on possible values of function pointers.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0009 - Volatile Variables for WCET analysis

**Name:** ABS#0009 - Volatile Variables for WCET analysis
**Alias:** Volatile Variables for WCET analysis
**Description:** The Code-Level Timing Analysis requires declaration of volatile variables.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0010 - Improving precision of WCET analysis by additional parameters

**Name:** ABS#0010 - Improving precision of WCET analysis by additional parameters
**Alias:** Improving precision of WCET analysis by additional parameters
**Description:** WCET analysis requires for better precision the following information:
- range information (lower and upper bounds) for input variables,
- addresses of memory accesses that cannot be found by automatic analysis,
- infeasible code snippets,
- flow facts that consist of linear constraints for the execution counts of several program points
- modes (exclusive execution paths of the runnable).

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0011 - Supported Processor for WCET analysis

**Name:** ABS#0011 - Supported Processor for WCET analysis
**Alias:** Supported Processor for WCET analysis
**Description:** WCET analysis requires that the code is compiled for a supported processor.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0012 - Processor Configuration for WCET analysis

**Name:** ABS#0012 - Processor Configuration for WCET analysis
**Alias:** Processor Configuration for WCET analysis
**Description:** WCET analysis requires information on the configuration of the processor.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## ABS#0013 - Processor-Specific Settings for WCET analysis

**Name:** ABS#0013 - Processor-Specific Settings for WCET analysis
**Alias:** Processor-Specific Settings for WCET analysis
**Description:** Processor specific parameters are required to perform WCET analysis
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0034 - Automation

**Name:** CAG#0034 - Automation
**Alias:** Automation
**Description:** The tools shall automate ...
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0003 - Code level results

**Name:** DSP#0003 - Code level results
**Alias:** Code level results
**Description:** It shall be possible to exchange results of a code level analysis according to DSP#0002 back to an originating tool for documentation and comparison.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## RTAW#0003 - Scenario based analysis

**Name:** RTAW#0003 - Scenario based analysis
**Alias:** Scenario based analysis
**Description:** T2U shall provide the means to describe (worst-case) scenarios of frame transmission events for sets of frames.

Rational:
Some CAN communication matrices, are based on a lot of event driven frame transmissions (i.e. ComTxModeMode= Direct or Mixed), as opposed to periodic transmissions (i.e. ComTxModeMode=periodic).
A per frame (worst-case) model of the transmission events based on the ComTxIPduMinimumDelayTimeFactor, leads generally to an over pessimistic estimation of worst-case frame transmission delays.
The specification of a Worst-Case scenario of (transmission) event occurrences for a set of frame is a solution to obtain less pessimistic worst-case bound.
How to determine such scenario is probably out of scope, but TADL-2 shall provide the means to describe these scenarios.
**Status:** Approved

**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## RTAW#0004 - Data converters

**Name:** RTAW#0004 - Data converters
**Alias:** Data converters
**Description:** T2U shall ease the exchange of data between tools through data format converters between most commonly used open data formats.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## TUBS#0003

**Name:** TUBS#0003
**Alias:**
**Description:** Formal system level performance analysis shall be able to analyze systems with uncertain parameter values.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

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

**Name:** VTEC#0003 - Methods for estimating WCET at analysis and design levels.
**Alias:** Methods for estimating WCET at analysis and design levels.
**Description:** TIMMO-2-USE shall explore efficient methods for estimating WCET at analysis and design levels.

Rationale: In order to really take timing into account in early development phases, efficient methods for estimating WCET in these phases. Early development phases often imply high abstraction levels.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0009 - Method and tool support for mode-dependent bus scheduling

**Name:** VTEC#0009 - Method and tool support for mode-dependent bus scheduling
**Alias:** Method and tool support for mode-dependent bus scheduling
**Description:** TIMMO-2-USE shall explore methods for obtaining good task and bus schedules based on modes.

Rationale: In order to make the bus communication more efficient, it might be changed depending on the mode of functions communicating over it. This imposes challenges for synchronizing the connected ECUs so that they all assume the same schedule.

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

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

**Name:** VTEC#0014 - Tool support for comparing alternative timing solutions
**Alias:** Tool support for comparing alternative timing solutions
**Description:** TIMMO-2-USE shall provide tool support for comparing several timing solutions for the following two purposes:
1. help the designer in making a good decision
2. help in the negotiation between OEM and supplier.

**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0017

**Name:** VTEC#0017
**Alias:**
**Description:** TIMMO-2-USE shall investigate methods and tools to identify the dependency on timing properties between functions and components.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0022

**Name:** VTEC#0022
**Alias:**
**Description:** TIMMO-2-USE shall develop methods and tools that convert the continuous timing requirements of the control design to the discrete-time specification of the software components.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0023

**Name:** VTEC#0023

**Alias:**
**Description:** TIMMO-2-USE shall develop methods and tools to verify if a given way of decomposing the controller to software components and allocating them to ECUs and the allocation of tasks and bus-frames satisfies the timing requirements
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0024

**Name:** VTEC#0024
**Alias:**
**Description:** TIMMO-2-USE shall investigate the possibility of methods and tools for optimizing the decomposition of the controller and the allocation of the software components and the allocation of tasks and bus-frames.
**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0025

**Name:** VTEC#0025
**Alias:**
**Description:** TIMMO-2-USE shall allow the partners to integrate their component models so as to verify if the complete application can satisfy the timing requirements.

Comment: virtual integration support. The Timing Model shall support the notion of deadline as the last allowed finish time of an activity. Decomposition of deadlines should be possible, i.e., when an activity is broken down into sub-activities.
**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0030

**Name:** VTEC#0030
**Alias:**
**Description:** TIMMO-2-USE shall explore design methods for exploiting knowledge on vehicle configurations.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0031

**Name:** VTEC#0031
**Alias:**
**Description:** TIMMO-2-USE shall explore methods for scheduling based on vehicle configurations
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

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

**Name:** VTEC#0033 - Methods for timing characterization of behavior/algorithm
**Alias:** Methods for timing characterization of behavior/algorithm
**Description:** TIMMO-2-USE shall explore efficient methods for characterizing behavior/algorithms with respect to timing.

Rationale: By combining the behavior/algorithm characterization with the ditto of hardware, it is theoretically possible to achieve a concrete WCET.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0035 - Methods for timing characterization of hardware

**Name:** VTEC#0035 - Methods for timing characterization of hardware
**Alias:** Methods for timing characterization of hardware
**Description:** TIMMO-2-USE shall explore efficient methods for characterizing hardware with respect to timing.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0036

**Name:** VTEC#0036
**Alias:**
**Description:** TADL shall support constructs that allow binding application-specific and infrastructure-specific timing information into absolute timing values (e.g. ms)
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0038

**Name:** VTEC#0038
**Alias:**
**Description:** TIMMO-2-USE shall use EAXML and ARXML files for exchange between tools.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0042

**Name:** VTEC#0042
**Alias:**
**Description:** The values of the implementation parameters in TADL model shall be automatically transferrable to the implementation software.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0044

**Name:** VTEC#0044
**Alias:**
**Description:** TIMMO-2-USE shall explore the methods and tools to analyze the synchronization constraint.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0048

**Name:** VTEC#0048
**Alias:**
**Description:** TIMMO-2-USE shall investigate methods and tools to analyze the timing properties using the probabilistic information.

Comment: the timing characteristics of the events in an event chain may be mixed. Some events have deterministic specifications; others have probabilistic specifications. The methods must be able to handle both types of information in a universal way.
**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

## 2.4 WP4 – Methodology

### ABS#0001 - Timing Analysis in Implementation Phase

**Name:** ABS#0001 - Timing Analysis in Implementation Phase
**Alias:** Timing Analysis in Implementation Phase
**Description:** The T2U methodology shall be capable of validation of timing properties in the Implementation phase.
**Status:** Rejected
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### BOSCH#0004 - Collaborative Engineering of Control Applications

**Name:** BOSCH#0004 - Collaborative Engineering of Control Applications
**Alias:** Collaborative Engineering of Control Applications
**Description:** The TIMMO-2-USE methodology shall enable the collaboration of the control engineer and the software engineer to find an adequate software solution (with respect to timing, resource consumption, and control quality) for a given control task.

Rationale: Better control quality requires more frequent sampling; this in turn increases resource consumption and decreases composability.

Comment: use case is to determine the "limits" of a solution.
Collaboration of roles functional engineer and software engineer.
Possibly specify a collaboration use case.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

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

**Name:** BOSCH#0010 - Methodology for timing design of control applications
**Alias:** Methodology for timing design of control applications
**Description:** The methodology shall contain explicit steps for the timing design of open and closed loop control applications at all levels (i.e. requirements specification, design, implementation, and test)

Comment: See BOSCH#0009, etc.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

### BOSCH#0011 - Derivation of discrete timing requirements

**Name:** BOSCH#0011 - Derivation of discrete timing requirements
**Alias:** Derivation of discrete timing requirements
**Description:** The methodology shall support the transformation of intrinsic timing requirements for control applications stemming from the physics of the plant and the environment (see Bosch #0001) to discrete (solution dependent) SW timing requirements

Comment: see BOSCH#0009 and BOSCH#0010, etc.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0006 - Obtain timing information (closed-loop)

**Name:** CAG#0006 - Obtain timing information (closed-loop)
**Alias:** Obtain timing information (closed-loop)
**Description:** The methodology shall describe the steps to obtain timing information of closed-loop systems/applications.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0007 - Use of SystemC

**Name:** CAG#0007 - Use of SystemC
**Alias:** Use of SystemC
**Description:** The methodology shall describe the use of SystemC on different levels of abstraction.

Rational: T2U shall check whether SystemC can be used as Model of Computation in order to obtain timing information on various levels of abstraction.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0009 - Scheduling Analysis

**Name:** CAG#0009 - Scheduling Analysis
**Alias:** Scheduling Analysis
**Description:** The methodology shall describe what information from models on different levels of abstraction and views (AUTOSAR) are required to perform scheduling analysis.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0016 - Use of AUTOSAR timing views

**Name:** CAG#0016 - Use of AUTOSAR timing views
**Alias:** Use of AUTOSAR timing views
**Description:** The methodology shall describe the use of AUTOSAR timing views.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0021 - Virtual integration (timing)

**Name:** CAG#0021 - Virtual integration (timing)
**Alias:** Virtual integration (timing)
**Description:** The methodology shall describe the steps to be taken in order to perform virtual integration on every level of abstraction.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0030 - Distribute jitter

**Name:** CAG#0030 - Distribute jitter
**Alias:** Distribute jitter
**Description:** The methodology shall describe how jitter specified for a given event chain is distributed among event chain segments (time budgeting).
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

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

**Name:** CAG#0031 - HW/SW Co-design (Methodology)
**Alias:** HW/SW Co-design
**Description:** The methodology shall describe the steps to be taken to perform HW/SW co-design.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0033 - EPF

**Name:** CAG#0033 - EPF
**Alias:** EPF
**Description:** The methodology shall be described using Eclipse Process Framework (EPF).

**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0035 - Task synthesis

**Name:** CAG#0035 - Task synthesis
**Alias:** Task synthesis
**Description:** The methodology shall describe the steps to be taken to automatically perform runnable entity to task mapping.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0036 - Variability

**Name:** CAG#0036 - Variability
**Alias:** Variability
**Description:** The methodology shall describe the steps to be taken to create a variant with regard to timing.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0038 - Timing Analyses

**Name:** CAG#0038 - Timing Analyses
**Alias:** Timing Analyses
**Description:** The methodology shall describe the possible timing analyses that can be carried out at the various levels of abstractions.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0040 - Verify timing constraints

**Name:** CAG#0040 - Verify timing constraints
**Alias:** Verify timing constraints
**Description:** It shall be possible to verify TADL constraints, when TADL is used within standardized modeling languages as EAST-ADL or SysML. TIMMO-2-USE shall ensure and give samples (e.g. in demonstrators) how TADL is applied in EAST-ADL and SysML TADL shall be extended, so that the verification of TADL constraints becomes possible, either dynamically or statically within these languages (e.g constraint evaluation, EMF-validation, plug-ins, etc.)
**Status:** Approved

**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0002 - Code level analysis

**Name:** DSP#0002 - Code level analysis
**Alias:** Code level analysis
**Description:** It shall be possible to perform a worst case execution analysis on code level according to DSP#0001.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0005 - Levels of abstraction

**Name:** DSP#0005 - Levels of abstraction
**Alias:** Levels of abstraction
**Description:** It shall be possible to specify timing requirements on different levels of abstraction (e.g. single functions, integration of functions with scheduling demands, integration of functions in a complete ECU)
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0006 - Expressiveness

**Name:** DSP#0006 - Expressiveness
**Alias:** Expressiveness
**Description:** It shall be possible to describe which timing-related information shall be exchange in order to support certain use cases. (This is kind of a methodology question, answering e.g. questions such as which subsets of data e.g. of the AUTOSAR standard should be exchanged).
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0007 - Generation of timing test units

**Name:** DSP#0007 - Generation of timing test units
**Alias:** Generation of timing test units
**Description:** It shall be possible to derive units under tests (preferably following the AUTOSAR methodology) to be analyzed with simulation methods with regard to timing requirements.
**Status:** Approved
**Priority:** High

**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0008 - Test framework

**Name:** DSP#0008 - Test framework
**Alias:** Test framework
**Description:** It shall be possible to derive a test framework suitable to perform timing related test activities (Which question can be answered? What kind of experiments need to be executed?)
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## DSP#0009 - Re-use of test descriptions

**Name:** DSP#0009 - Re-use of test descriptions
**Alias:** Re-use of test descriptions
**Description:** It shall be possible to re-use test descriptions for timing aspects at different phases of the development.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## TUBS#0004 - Obtain uncertain timing information

**Name:** TUBS#0004 - Obtain uncertain timing information
**Alias:** Obtain uncertain timing information
**Description:** The methodology shall describe the steps to obtain timing information in presence of uncertainty.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0006 - Transformation

**Name:** UPB#0006 - Transformation
**Alias:** Transformation
**Description:** The Timing Model specifications shall support transformations to constraints and assertions.

Rational: There is a need to transform timing constraints into Property Specification Language PSL in order to specify properties on various levels of abstraction.
**Status:** Approved
**Priority:** High

**Explanation:** None
**Type:** Requirement
**Relations:** UC#0003 - Change Existing Timing Information

## UPB#0013 - Reduced overhead

**Name:** UPB#0013 - Reduced overhead
**Alias:** Reduced overhead
**Description:** The Timing Model shall specify timing properties and constraints for multiple abstraction levels with minimal overhead.

Comment: It shall be demonstrated how the TADL and methodology are used in order to accomplish this (model only what is necessary for a specific purpose). Identify also possible use case for this.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0015 - Refinement and abstraction

**Name:** UPB#0015 - Refinement and abstraction
**Alias:** Refinement/Abstraction
**Description:** The Timing Model shall support refinements and abstractions between different levels of abstraction.

Comment: With regard to abstraction language elements need to be introduced to accomplish this.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0020 - Offline simulation

**Name:** UPB#0020 - Offline simulation
**Alias:** Offline simulation
**Description:** The Timing Model shall support offline simulation with respect to timing.

Comment: Non HIL simulation/simulation without hardware. Behavior model of the "hardware", Considering the environment (EAST-ADL). Executable model? What kind of executable models are required to perform such simulation? Possibly WP3 but check whether WP2 and WP4 are also involved.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## UPB#0021 - Communication simulation

**Name:** UPB#0021 - Communication simulation
**Alias:** Communication simulation
**Description:** The Timing Model shall support simulation of buses and networks, for example "Restbussimulation" [German].

Comment: See UPB#0020
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0004 - Timing budget negotiation between OEM and supplier

**Name:** VTEC#0004 - Timing budget negotiation between OEM and supplier
**Alias:** Timing budget negotiation between OEM and supplier
**Description:** The methodology shall support the interplay between OEM and supplier when it comes to negotiating and renegotiating timing budgets.
**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0005

**Name:** VTEC#0005
**Alias:**
**Description:** It shall be possible to assign different access rights to different users/developers (or types/groups of users/developers) of a model. Some may be allowed to modify or view only certain parts of a model.

Comment: This requirement was used in TIMMO.
**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0010 - Methodology support for mode-aware design

**Name:** VTEC#0010 - Methodology support for mode-aware design
**Alias:** Methodology support for mode-aware design
**Description:** TIMMO-2-USE shall explore the methodological implications of mode-aware design.

Rationale: Such implications could arise from the fact that functions are distributed on several ECUs, provided by different suppliers.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0015 - Methodology support for change management

**Name:** VTEC#0015 - Methodology support for change management
**Alias:** Methodology support for change management
**Description:** The methodology shall describe how change is best handled in the context of product timing properties.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0018

**Name:** VTEC#0018
**Alias:**
**Description:** When multiple functions or components need be modified, TIMMO-2-USE methodology shall help the developers plan the sequence of modification so that the number of iterations can be reduced.
**Status:** Approved
**Priority:** High
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0037

**Name:** VTEC#0037
**Alias:**
**Description:** TIMMO-2-USE shall find the effective methodology to guide the developers of the applications and the suppliers of the platforms.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0043

**Name:** VTEC#0043
**Alias:**
**Description:** TIMMO-2-USE methodology shall support efficient handling of system parameters.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0046

**Name:** VTEC#0046
**Alias:**

**Description:** TIMMO-2-USE shall define reference methodology to manage synchronization issues.
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## VTEC#0049

**Name:** VTEC#0049
**Alias:**
**Description:** TIMMO-2-USE shall develop a reference methodology to support probabilistic timing requirements and constraints.
**Status:** Approved
**Priority:** Low
**Explanation:** None
**Type:** Requirement
**Relations:**

### *2.5 WP5 – Validation*

## CAG#0042 - Static verification

**Name:** CAG#0042 - Static verification
**Alias:** Static verification
**Description:** In order to enable the verification of timing constraints, it must be clear, which concepts of TADL can be verified statically (without simulation) and dynamically (with simulation) Requirement VTEC#N0001 gives an important hint, how to verify TADL constraints statically. (requirement and property ? provided and required in AUTOSAR) A time property can be set on a measured execution time and directly connected events, a time requirement on an event chain connecting two events on top level of a system
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**

## CAG#0043 - Dynamic verification

**Name:** CAG#0043 - Dynamic verification
**Alias:** Dynamic verification
**Description:** Some aspects of TADL, like the synchronization constraint requires a simulation and thus a simulation environment in order to be verified. The aspects of TADL, which have to be verified dynamically, have to be worked out. SystemC or Modelica, which offer an execution semantics and runtime environments are good candidates to perform required simulations. A mapping of EAST-ADL or SysML on these languages is either possible or already defined.
**Status:** Approved
**Priority:** Medium
**Explanation:** None

**Type:** Requirement
**Relations:**

## CAG#0044 - Runtime Trace

**Name:** CAG#0044 - Runtime Trace
**Alias:** Runtime Trace
**Description:** As an alternative solution to verify dynamic aspects of TADL, a (data) trace can be written at runtime, and verification of TADL constraints can be done after execution on base of the written runtime data. In order to do so, a format for a TADL runtime trace has to be specified. The runtime trace should allow the verification of dynamic aspects of TADL
**Status:** Approved
**Priority:** Medium
**Explanation:** None
**Type:** Requirement
**Relations:**