LITERATURE REVIEW IN THE FIELDS OF STANDARDS, PROJECTS, INDUSTRY AND SCIENCE

DOCUMENT TYPE: **DELIVERABLE**

DELIVERABLE N^0: **D1.1**

DISTRIBUTION LEVEL: **PUBLIC**

DATE: **11/07/2016**

VERSION: **FINAL**

AUTHOR(S): LEONID LICHTENSTEIN, FLORIAN RIES, MICHAEL VÖLKER, JOS HÖLL, CHRISTIAN KÖNIG (TWT); JOSEF ZEHETNER (AVL); OLIVER KOTTE, ISIDRO CORAL, LARS MIKELSONS (BOSCH); NICOLAS AMRINGER, STEFFEN BERINGER, JANEK JOCHHEIM, STEFAN WALTER (DSPACE); CORINA MITROHIN, NATARAJAN NAGARAJAN (ETAS); TORSTEN BLOCHWITZ (ITI); DESHENG FU (LUH); TIMO HAID (PORSCHE); JEAN-MARIE QUELIN (RENAULT); RENE SAVELSBERG, SERGE KLEIN (RWTH); PACOME MAGNIN, BRUNO LACABANNE (SIEMENS); VIKTOR SCHREIBER (TU-IL); MARTIN KRAMMER, NADJA MARKO, MARTIN BENEDIKT, STEFAN THONHOFER, GEORG STETTINGER, MARKUS TRANNINGER (VIF); THIES FILLER (VW)

REVIEWED: BRUNO LACABANNE (SIEMENS), JEAN-MARIE QUELIN (RENAULT), VALENTIN IVANOV (TUIL), JOSEF ZEHETNER (AVL), NADJA MARKO (VIF)

APPROVED: LEONID LICHTENSTEIN (TWT), MARTIN BENEDIKT (VIF)

PROJECT ACRONYM:	**ACOSAR**
PROJECT TITLE:	**ADVANCED CO-SIMULATION OPEN SYSTEM ARCHITECTURE**
ITEA PROJECT N^0:	**14004**
CHALLENGE:	**ENGINEERING**
PROJECT DURATION:	**01/09/2015 - 31/08/2018**
PROJECT WEBSITE:	**WWW.ACOSAR.EU**
COORDINATION:	**VIRTUAL VEHICLE RESEARCH CENTER (AT)**
PROJECT LEADER:	**DR. MARTIN BENEDIKT**

1 Contents

2 Executive summary

The main goal of ACOSAR is to develop a non-proprietary real time (RT) co-simulation interface. This document presents a survey of the relevant state of the art and state of practice. Different technical domains that are relevant for the ACOSAR project were analysed, reviewed and summarized. A short summary of every chapter is presented in the following.

Discrete-Event Simulation is a simulation method, where the operation is represented as a chronological sequence of events. There are various synchronization algorithms which are optimized for different systems and environments. These algorithms are relevant for ACOSAR in terms of the synchronized exchange of information between simulators and RT systems and have to be considered during the specification of the advanced co-simulation interface (ACI).

Continuous Simulation serves as fundamental basis for numerical investigation of dynamic system behaviours and powerful explicit and implicit numerical solvers were developed for dedicated classes of systems. In contrast to numerical simulation, recently several approaches were published with respect to event-based simulation of continuous system dynamics, leading to significant benefits in terms of accuracy and simulation time reduction. Both approaches will be considered and incorporated into ACOSAR results.

Hybrid Simulation represents an approach, where continuous and discrete system behaviours are handled. As cross-domain considerations of embedded systems and mechatronic products is mandatory hybrid system analysis become more and more relevant. Recently discussed approaches for simulation of hybrid systems are considered for ACI specification within ACOSAR to enable and support a common approach for integration if Real-Time Systems.

Real-time simulation aims at providing the right information and data at the right time, while an actual system is operating. This type of simulation is typically employed in the later stages of a system development cycle, when real components are integrated and run in parallel to simulated systems. A large variety of real-time simulation approaches exists in the literature and practice. ACOSAR's goal is, therefore, to identify, tailor, and extend existing techniques in real-time simulation to generate a standard interface for such simulations.

In a co-simulation scenario, the **communication layer** realizes the data transmission between the different participating systems. ACOSAR's goal is to define a certain level of communication protocol abstraction such that various existing communication protocols can be supported. The large group of existing communication standards can be roughly categorized in internet protocol (IP) based, automotive and industrial standards. Each category has its own applications. The ACI communication layer should be able to support a large variety of communication standards to make sure that all desired use cases can be properly addressed. The ACI communication layer should therefore be flexible enough to support the given communication standards while it can still be extended to support some additional communication standards in the future.

A **real-time system** is a combination of soft- and hardware that must process information and produce a response within a specified time. Usually it consists of a physical component and a controller, an entity observing and eventually controlling the physical part. Structured approaches and requirements have been reported in the literature for networked real-time systems, e.g., in terms of timeliness, predictability, efficiency, robustness, fault tolerance, and maintainability. Within ACOSAR, the challenge will be to close gaps between existing solutions at the levels of the simulation tool interface, communication protocol and the hardware interface.

Several **interoperability standards** exist for the integration of RT systems and non-RT systems. The review shows that there is an ongoing strong interest and effort to standardize interfaces of systems to further streamline the systems development process. None of the reviewed standards concerns a generic interface for executing RT co-simulations. Nevertheless, within ACOSAR, some concepts from existing standards shall be considered for reuse.

System modelling languages, like the Unified Modelling Language (UML), support requirements engineering, specification, analysis, design as wells as verification and validation of systems. Many system problems result from inadequately defined interfaces. These problems are often detected too late in the development process. The ACI aims at providing clear interface specifications for RT system integration at different development stages. The use of modelling languages for interface specification is favourable as it provides a way to continuous refinement through the use of recurring interface information.

A large set of recent and ongoing **research projects** was reviewed. A clear trend towards integrated, holistic approaches to model-based systems engineering in real time, across domains and enterprises with a focus on openness and standards was determined. ACOSAR aims for exactly this openness and flexibility in the systems development process. However, ACOSAR also faces the challenge of suiting a large group of divers use cases.

ACOSAR partners have provided information on their **commonly used tools** which clearly reflect the current state of practice within the consortium. The integration of RT systems is, however, limited due to the use of proprietary interfaces. ACOSAR aims on designing an open RT co-simulation interface. Further, the usage of the ACI can be ensured if all tool vendors of the consortium implement the ACI for their tools.

The comprehensive list of related standards and projects shows that there is a strong attempt to improve interoperability between tools and devices. None of the existing standards provides a generic and flexible interface which enables RT co-simulation. ACOSAR's goal is to specify such an interface standard allowing a large amount of applications in different domains. To suit different use cases and domains the ACI has to provide the possibility to use different kinds of communication channels and therefore requires an abstraction of the communication layer.

3 Introduction

This report describes the results of the work undertaken in Task 1 of Work Package 1 of the ACOSAR project.

The overall aim of ACOSAR is to develop a non-proprietary real-time co-simulation interface. This "Advanced Co-Simulation Interface" (ACI) will be a substantial contribution to international standardization activities (e.g. FMI) and will demonstrate a systems integration methodology that will save effort compared to the current state-of-practice. In Work Package 1 our goal is to define a set of requirements to the ACI that are well-structured and feasible. For a better judgement of feasibility and practicality, we have thoroughly investigated the relevant state-of-the-art and the state-of-practice, as described in Task 1.1. The results of the investigations are presented in this report.

The goal of this deliverable is to present a detailed survey of the relevant state-of-the-art and the state-of-practice. This document constitutes an important and valuable contribution both to the project and to the community. The overview will be a useful reference for the further course of the project.

To compile the report, the ACOSAR consortium has first identified an extensive collection of scientific publications, research projects, industrial standards and systems engineering tools that are relevant for the project. Subsequently, all these elements were reviewed, their important results and outcomes summarized, and the planned contributions of ACOSAR positioned. Finally, all subdomain specific information was further condensed and conclusions drawn.

The report is divided into two main chapters: state-of-the-art (Chapter 4) and state-of-practice (Chapter 5). Chapter 5 is further split into the following subdomains:

- (Distributed) Discrete-Event Simulation
- (Distributed) Continuous Simulation
- (Distributed) Hybrid Simulation
- (Distributed) Real-Time Simulation
- Communication Standards
- Real-Time Systems
- Interoperability and Related Standards
- Requirements, Modelling, Design, and Specification for Integration

4 State of the Art

4.1 (Distributed) Discrete-Event Simulation

4.1.1 Introduction

Discrete Event Simulation (DES) is a simulation method, where the operation of the system is represented as a chronological sequence of events. Each event occurs at an instant of time and triggers a change of system states. It is expected that DES takes advantage of the multi-processor environment as many other applications do. As consequence, approaches exist that try to reduce the execution time of DES in multi-processor environments. In the Distributed Discrete Event Simulation (DDES), multiple Physical Processes (PPs) in one computer or several loosely coupled computers are considered. The whole simulation model is partitioned into several components and each of them is implemented as a Logical Process (LP), which is assigned statically or dynamically to a PP. Compared with other simulation methods, DES benefits from the conservative estimation of the time of the next system operation. It provides a fast time advance for many systems, especially when the change of the system is hard to calculate but the time of decisive change is easy to estimate.

Causality error is the most challenging topic for DDES. It refers to the situation that one LP runs faster than another LP, and it receives an event from that LP with a time stamp that is smaller than its current local time, i.e. this event should be processed locally “in the past”. Generally, there are two ways to resolve the causality error. When an optimistic time-warp algorithm is applied, the LP rolls back to the time right before the time stamp of the event violating causality, while conservative synchronization algorithm synchronizes all LPs by their local virtual time thereby avoiding such inconsistencies.

4.1.2 Relevant State of the Art Items

4.1.2.1 Parallel and Distributed Simulation Systems

Publication name	Parallel and Distributed Simulation Systems
Authors	Richard M. Fujimoto
Year	2001
Reference	http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=977259 Print ISBN: 0-7803-7307-3
Summary	This paper discusses the general structure of distributed discrete event simulation and the basic algorithms to avoid / solve causality error. It gives an overview of technologies to distribute the execution of simulation programs over multiple computer systems. Particular emphasis is placed on synchronization (also called time management) algorithms as well as data distribution techniques.
Project	N/A.
ACOSAR relevance	This paper discusses the general structure of distributed discrete event simulation. The Advanced Co-Simulation Interface (ACI) developed in ACOSAR should provide necessary mechanism so that the structure and the related algorithms could be applied to the simulators connected to ACI.
Summarized by	Desheng Fu (LUH)

4.1.2.2 Distributed Simulation: A Case Study in Design and Verification of Distributed Programs

Publication name	Distributed Simulation: A Case Study in Design and Verification of Distributed Programs
Authors	Chandy, K.M.; Misra, J.
Year	1979
Reference	http://ieeexplore.ieee.org/xpl/articleDetails.jsp?&arnumber=1702653 ISSN: 0098-5589
Summary	In this paper, the most important CMB algorithm is presented. It described a basic architecture of the distributed discrete event simulation without shared variable by all parts of the system. The processes communicate only through messages with their neighbors. The CMB algorithm is aimed to synchronize the processes based on the look-ahead, and provided a solution to solve the deadlock between the processes through null-messages. The correctness of a distributed system is also proven within the paper.
Project	N/A.
ACOSAR relevance	In this paper, the most important CMB algorithm (null-message) algorithms is presented. It is still applied today with a few optimizations. This algorithm shows a typical model that how the logical processes (the simulators) communicate with each other to achieve the synchronization. In the ACOSAR project, we should provide a similar channel, so that the simulators could exchange same information. Such an exchange channel might be implemented as a special state exchange in FMI (or similar co-simulation interfaces, e.g. ACI). This might be discussed in the relevant WPs.
Summarized by	Desheng Fu (LUH)

4.1.2.3 Distributed Simulation: Non-committal Barrier Synchronization

Publication name	Non-committal Barrier Synchronization
Authors	Nicol, D. M.
Year	1995
Reference	Parallel Computing 21: 529 - 549
Summary	This paper described a very important method for the distributed barrier synchronization. In such systems, all process must be synchronized after the barrier primitive is executed. And they enter the next barrier after the global synchronization. A very popular algorithm for distributed barrier synchronization, called butterfly barrier algorithm is presented in this paper. It provides a balanced solution between the delay of the synchronization and the amount of messages to exchange.
Project	N/A.

ACOSAR relevance	The simulators / processes connected to the ACI, which will be developed in the ACOSAR project, must be synchronized periodically for the state exchange. The synchronization between the simulators is similar as the synchronization in discrete simulation, especially when a state variable will be exchanged only if it has been modified. Thus, the distributed barrier synchronization might also be considered to replace the synchronization organized by a global controller for ACI. In this way, the delay for the synchronization can be reduced, and not all processes must be blocked and wait for the global synchronization. On the other side, the bandwidth of the network has to be considered, so that a short transmission time is guaranteed. The butterfly barrier algorithm presented in this paper provides a balanced solution for barrier synchronization and should be considered in this project.
Summarized by	Desheng Fu (LUH)

4.1.2.4 Hybrid Simulation Using SAHISim Framework

Publication name	Non-committal Barrier Synchronization – A Hybrid Distributed Simulation Framework Using Waveform Relaxation Method Implemented Over the HLA and the Functional Mock-up Interface
Authors	Awais, M. U.; Gawlik W.; De-Cillia G.
Year	2015
Reference	http://dl.acm.org/citation.cfm?id=2832219 ISBN: 978-1-63190-079-2
Summary	In this paper the hybrid simulation based on the SAHISim Framework is presented. The hybrid simulation combines the most popular standards HLA and FMI. The most challenging topic of such hybrid simulations is the synchronization between different simulation components. A solution is discussed and a case-study is presented.
Project	N/A.
ACOSAR relevance	A simulation based on ACI can also be considered as a hybrid (distributed) simulation, since ACI is aimed to integrate various types of virtual model and physical hardware. Most important, we have to provide a solution to handle the discrete events. In this paper, a basic solution is presented. First of all, the iteration where the discrete event was detected will be aborted. The FMUs (or other simulators) go back to the state at the end of last communication step. After that, the step size will be minimized. The iteration will be repeated and the exact time that the event occurs will be detected. The state change caused by the discrete event will be propagated at the next communication step, which is very close to the occurrence of the event. After that, the step size will be reset. In the ACOSAR project, the iteration cannot be aborted easily at the online mode. We have to consider a different solutions to solve the problem as good as possible.
Summarized by	Desheng Fu (LUH)

4.1.3 Summary and Conclusion

In this chapter, we reviewed some of important papers in the area of DES / DDES, which give us an overview of structure and the basic algorithms that are applied on DDES.

In ACOSAR, we will provide a general Advanced Co-simulation Interface (ACI) supporting DES / DDES. Since ACOSAR and the ACI is optimized for the real-time simulation and the combination of virtual model and physical model (hardware), the optimistic time-warp algorithm cannot be applied and thus only the conservative synchronization will be considered. As shown in the papers, there are various synchronization algorithms which are optimized for different systems / environments. In the ACOSAR project, we will consider to support most of the algorithms.

At the co-simulation level, the DES / DDES works in the same way as the other simulation systems. Thus there are only few things to do to support DES / DDES:

1. Support information exchange between simulators.
2. Identify the type of the simulators, so that the best synchronization methods could be chosen automatically.
3. Define the semantic of the synchronization message related to the synchronization methods.

4.2 (Distributed) Continuous Simulation

4.2.1 Introduction

Continuous simulation represents the fundamental basis for numerical simulation with the exclusive focus on continuous dynamics analysis. Already in 1970 first attempts towards modular (distributed) continuous simulation were published, were the focus was on a significant reduction of simulation time for numerical analysis of large-scale electrical circuits. However, during the last decade the motivation of modular simulation changed completely, as the focus now is manly on integration of domain-specific simulation tools, typically coming from dedicated engineering department or 3rd party component suppliers. FMI pursues this trend since 2010 due to the Functional MockUp Interface Standard, which specifies an interface to simulation tools (FMI for Co-Simulation).
This section gives a brief overview about published co-simulation approaches and continuous simulation in general with main focus on coupling and synchronization challenges as well as the relevance to the ACOSAR project.

4.2.2 Relevant State of the Art Items

4.2.2.1 Master for Co-Simulation Using FMI

Publication name	Master for Co-Simulation Using FMI
Authors	Jens Bastian; Christoph Clauß; Susann Wolf; Peter Schneider
Year	2011
Reference	https://modelica.org/events/modelica2011/Proceedings/pages/papers/05_2_ID_165_a_fv.pdf ISBN: 978-91-7393-096-3 ISSN: 1650-3686
Summary	This paper discusses a proposal for co-simulation in a master-slave concept. In this context the slave participants simulate sub-systems and the master coordinates the exchange between the slaves. Therefore the existing generic interface FMI (Functional Mock-up Interface) was developed to merge different simulation tools, such as SimulationX. Also different solver and integral methods for ODE's/DAE's (ordinary differential equation/differential algebraic equation) like Gauß-Seidel or Newton's method are shortly discussed.
Project	MODELISAR (founded by ITEA2/BMBF)
ACOSAR relevance	This publication overlaps some work packages of ACOSAR. In particular, the WP3 (Simulation Tool Interface) is related to issues like parameter definitions, Co-Simulation Description Schema and Prototype simulation tool interfaces (comparable with ACI) and associated solutions. This paper could be a basic study for developing the advanced Co-Simulation Interface in WP6 (Advanced Co-Simulation Interface). However, the proposed Master Algorithm was not realized for the FMI standard. Merely, a prototypical ANSI C code for masters and slaves was realized. Additionally, the Master interface is not real-time capable. After a critical review it can be noted that the implemented interface is working rudimentary, as seen on the simulation results (the simulated speed is oscillating not plausible, see Fig. 9 in the paper) [R22221].
Related WP	WP3 (Simulation Tool Interface). WP6 (Advanced Co-Simulation Interface).

Summarized by	Viktor Schreiber (Technische Universität Ilmenau)

4.2.2.2 Guidelines for the Application of a Coupling Method for Non-iterative Co-Simulation

Publication name	Guidelines for the Application of a Coupling Method for Non-iterative Co-Simulation
Authors	Martin Benedikt and Anton Hofer
Year	2013
Reference	[BMH2013]
Summary	A successful co-simulation application is based on efficient coupling technologies as subsystems are independently solved and synchronized at coupling time instants (cp Figure 1, page 245 in [BMH2013]). The mandatory extrapolation of coupling signals due to bidirectional dependencies introduces coupling errors which have to be handled by the coupling mechanisms. This work introduces a nearly energy preserving coupling approach to handle these errors in non-iterative co-simulation problems. The successful coupling error compensation depends on the correct parameterization of the coupling scheme which in turn is strongly related to the given subsystem configuration (i.e. no general setting available). In order to meet this target parameterization, guidelines for the nearly energy preserving coupling element are outlined.
Project	During ICOS tool development at VIRTUAL VEHICLE Research Center
ACOSAR relevance	This work describes a possible coupling algorithm for non-iterative co-simulation which compensates coupling errors caused by the extrapolation of coupling signals. After the development of the ACI every user can integrate his expert knowledge in form of coupling technologies in the Functional Framework, which implements smart integration strategies, to solve the specific problem setting. As this work describes one possible coupling scheme (Smart Function) including guidelines for the correct parameterization, it is directly related to the application of the ACI Functional Framework.
Related WP	WP3 (Simulation Tool Interface).
Summarized by	Georg Stettinger, Nadja Marko (Virtual Vehicle)

4.2.2.3 Parallel Co-Simulation for Mechatronic Systems

Publication name	Parallel Co-Simulation for Mechatronic Systems
Authors	Markus Friedrich
Year	2011
Reference	ISBN 9783843903363 • http://www.mw.tum.de/fileadmin/w00bpv/www/Die_Fakultaet/TdF2012/Friedrich_Zusammenfassung.pdf • http://mediatum.ub.tum.de/doc/1063436/1063436.pdf

Summary	This doctoral thesis treats multi-domain coupled simulation, (especially mechanical, hydraulic and electric ones) using a weak and therefore explicit co-simulation approach. The main focus is on stability improvements and parallelization in order to make simulations on multi CPU computers more time efficient. Therefore, different extrapolation methods (using polynomials, hermite splines or linear combination) are reviewed and their influence on the different physical domains is analysed. Furthermore, different methods of inter-process communication (IPC) are compared (e.g. files, sockets, pipes or shared memory). Finally, a co-simulation framework is implemented and applied to different use cases
Project	none
ACOSAR relevance	Since ACOSAR will deal with different physical domains both in the online and the offline world, the presented analysis might come in handy when defining the requirements for the ACI. For example one such requirement would be a method to apply arbitrary extrapolation schemes to signals transmitted by the ACI-element (e.g. via plug-ins). The influence of different extrapolation schemes (= master-algorithms) on the different physical domains as well as the error induced by this extrapolation will limit the makro step size and therefore dictate the required performance of the ACI-element in terms of coupling frequency, signal amount and round trip time.
Related WP	Most of the conclusions in this thesis are related to performance, extrapolation and stability issues, which could be interesting during the design phase of the ACI-element in WP6.
Summarized by	Timo Haid (Porsche AG)

4.2.2.4 Parallel Simulation Process for Virtual Prototyping in the Automotive Industry

Publication name	Parallelisierte Simulationsprozesse für virtuelles Prototyping in der Automobilindustrie [parallel simulation process for virtual prototyping in the automotive industry]
Authors	Mathias Hommel
Year	2006
Reference	full text: http://rzbl04.biblio.etc.tu-bs.de:8080/docportal/servlets/MCRFileNodeServlet/DocPortal_derivate_00002878/Hommel_Parallelisierte_Simulationsprozesse.pdf
Summary	The thesis focuses on the clustered simulation of a full hybrid vehicle with several software-in-the-loop controllers, distributed across multiple computers. Therefore, a detailed analysis of computer interconnection and data transfer methodologies (physical layer and protocol) regarding bandwidth, latency and communication frequency is conducted. Furthermore, the connection between controller and controlled system in terms of interface definition and communication type is examined. Also practical problems like start and termination of the clustered simulation as well as data synchronization. As use case a Volkswagen BHEV prototype is modelled as clustered simulation using different simulation tools and the co-simulation framework EXITE.
Project	none

ACOSAR relevance	Since ACOSAR will deal with interface standardisation across different physical communication layers, the conclusions reached in this thesis regarding data transfer methodologies as well as interface definition should prove applicable. Regarding ACOSAR the thesis reached the conclusion, that the most performant communication methods were those which bypass the operating system kernel and its safety measures as well as the ISO-OSI-standard. Additionally, it is concluded, that the method which yields the best bandwidth and latency is not necessarily performing best when running a real clustered simulation across multiple computers. Furthermore, practical problems in test automation like start and termination of clustered simulations, that ACOSAR will address too, are discussed in this work.
Related WP	During ACI design in WP6 as well as implementation in WP5 the relevant conclusions should be considered. Especially working around the OS to achieve the best possible performance. Further, it is relevant for the requirements specification in WP1.
Summarized by	Timo Haid (Porsche AG)

4.2.2.5 Investigation of Communication Intervals in Coupled Simulations

Publication name	Untersuchung des Kommunikationsintervalls bei der gekoppelten Simulation [Investigation of Communication Intervals in Coupled Simulations]
Authors	Lars Völker
Year	2011
Reference	Dissertation, Karlsruher Institut für Technologie [Karlsruhe Institute of Technology] (KIT) http://dx.doi.org/10.5445/KSP/1000021208
Summary	Coupling simulation platforms via co-simulation requires the exchange of data (coupling data) at specific points in time, more precisely at the end of a communication interval (macro step size). The performance and accuracy of co-simulations highly depends on the size of the macro step. The smaller the macro step size the greater the accuracy of the simulation. The greater the macro step size the greater the performance of the simulation. So there is a trade-off between simulation performance and accuracy. Normally the optimal macro step size has to be chosen manually by a user based on depth knowledge of the sub models. Within the literature a sophisticated approach is presented which determines the (fixed) macro step size based on frequency analysis of sub models. The presented approach is prototypical implemented in MATLAB/Simulink as co-simulation master and SIMPACK, AMESim and DSHplus as co-simulation slaves.
Project	-
ACOSAR relevance	Within the ACOSAR project a Functional Framework will be defined as part of the ACI Communication Layer. The implementation of the Functional Framework can include Smart Functions e.g. coupling strategies (see Figure 1). When coupling RT and RT/non-RT systems over a communication system (EtherCAT, CAN, etc.), efficient coupling strategies (which lead to a bandwidth reduction for example) like the presented

above could be useful to improve the simulation performance by avoiding unnecessary communications while simulation accuracy is not affected.

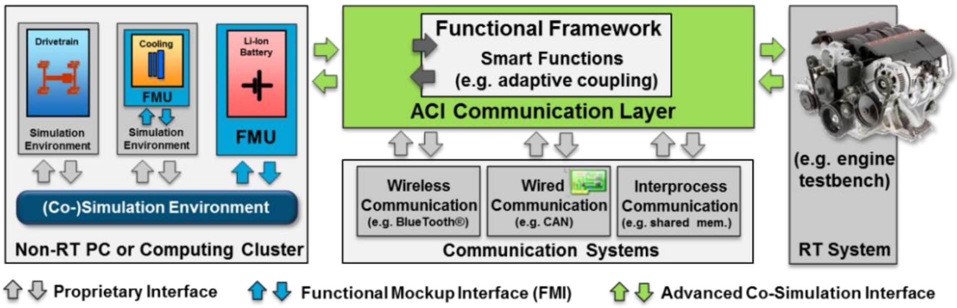


Figure 1: Functional Framework as part of ACI Communication Layer. The Functional Framework can include Smart Functions (Source: http://acosar.eu/overview.php)

Related WP	This publication is relevant for WP6, especially for the prototypical implementations of Smart Functions but also for the specification of the Functional Framework API. For example the API can provide the possibility to specify the frequency of sub models. With this information ACI users are able to implement efficient coupling strategies. Further, it is relevant for the requirements specification in WP1.
Summarized by	Nicolas Amringer (dSPACE)

4.2.3 Summary and Conclusion

Within this section some relevant publications about modular System Simulation, in terms of distributed continuous co-simulation, are presented. The reviewed articles clearly indicate the performance improvement using distributed co-simulation and that subsystem integration (coupling) represents a serious challenge and has to be handled in an adequate manner: for different classes of systems specific coupling algorithms are mandatory. Thus, for handling of IPR-protected (black-box) subsystems appropriate information at subsystem interfaces has to be made available (e.g. FMI).

4.3 (Distributed) Hybrid Simulation

4.3.1 Introduction

Systems becomes more and more complex and typically different domains of system science have to be virtually integrated to enable overall system development and analysis (including subsystem interactions) as well as cross-domain system optimization. Especially the integration of (control) software and physical systems requires the simultaneously consideration of discrete and continuous simulation aspects. This approach is referred to as Hybrid Simulation, where continuous simulation and discrete-event simulation are combined, i.e. modal models are simulated.
This Section gives an overview of collected State of the Art hybrid simulation topics relevant to ACOSAR. It covers the hybrid simulation, requirements for hybrid simulation, formalization attempts and existing hybrid simulation frameworks.

4.3.2 Relevant State of the Art Items

4.3.2.1 Continuous System Simulation

Publication name	Continuous System Simulation
Authors	François E. Cellier; Ernesto Kofman
Year	2010
Reference	ISBN: 978-1-4419-3863-3 ISSN: 0-387-30260-3 (eBook)
Summary	This book introduces the concept of numerical simulation of physical systems that are described by differential and algebraic equations. Plenty of simulation tools are available nowadays and in most cases they “do well”. It is highly important, though, that we understand the mechanism behind the simulation tools. This allows us to understand and correct simulation errors that may arise and to reduce the execution time. After introducing the basic principle of numerical integration, the book focuses on various algorithms which use time-discretization to solve ordinary and partial differential equations. These methods are then extended to the simulation of discontinuous systems, where zero-cross detections must be performed. Finally, the main points of real-time simulation are given. The second part of the books deals with discrete event simulation and its application to quantized state simulation.
Project	-
ACOSAR relevance	In order to specify an innovative interface and innovative methods for simulation of hybrid system, it is highly important to understand in detail the numerical methods applied for continuous time simulation.
Related WP	WP3 (Simulation Tool Interface w.r.t continuous and discrete-event simulation).
Summarized by	Daniela, Dejaco, Martin Benedikt (Virtual Vehicle)

4.3.2.2 Requirements for Hybrid Cosimulation

Publication name	Requirements for Hybrid Cosimulation
Authors	David Broman, Lev Greenberg, Edward A. Lee, Michael Masin, Stavros Tripakis and Michael Wetter
Year	2014
Reference	Technical Report No. UCB/EECS-2014-157 http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-157.html
Summary	This paper defines a suite of requirements for future hybrid co-simulation standards, i.e. requirements for interface definitions that enable diverse simulation tools to interoperate. In addition, the paper defines a set of test compositions of components, i.e. test cases. Examples of such test cases are proper integration in the presence of discontinuous signals and glitches, zero-delay feedback, and piecewise constant signals. These test cases define requirements for coordination between components.
Project	COSMOI (NSF): Compositional System Modelling with Interfaces ExCAPE (NSF): Expeditions in Computer Augmented Program Engineering
ACOSAR relevance	The paper defines a set of test cases that a standard for hybrid co-simulation must support, i.e. it defines a set of requirements. In ACOSAR, a standard is developed that shall, among other things, support a coupling between continuous and discrete models. The set of test cases that is defined in this paper may help to formulate requirements for ACI, and to validate parts of the ACI standard.
Related WP	WP 1 (the requirements proposed in this paper may translate into requirements for ACI) WP 6 (the test cases proposed in this paper may help with the validation of ACI)
Summarized by	Oliver Kotte (Bosch)

4.3.2.3 Formalization of global simulation Models for Continuous/Discrete Systems

Publication name	Formalization of global simulation models for continuous/discrete systems
Authors	Gheorghe, Luiza; Bouchhima, Faouzi; Nicolescu, Gabriela; Boucheneb, Hanifa
Year	2007
Reference	Gheorghe. L et al, A Formalization of global simulation Models for Continuous/Discrete Systems, Proceedings of the 2007 Summer Computer Simulation Conference, ISBN # 1-56555-316-0 , July15 -18, 2007
Summary	The paper provides a formal definition of a co-simulation model with focus on continuous/ discrete global synchronization and simulation interfaces. The simulation interface behaviour is represented as a timed automata and the synchronization functionality is distributed to the simulation interfaces.

Project	Part of Ph.D. Thesis from Luiza Gheorghe, Continuous/Discrete Co-Simulation Interfaces - From Formalization To Implementation, University of Montréal, Ècole Polytechnique de Montréal, 2009
ACOSAR relevance	The paper outlines synchronization of a co-simulation interface for a continuous-discrete simulation. The timed automata approach formulates the behaviour of a continuous/ discrete co-simulation interface. It also proposes a methodology from definition of operational semantics and verification of execution models. This generic methodology can be considered as basis in definition of properties for the simulation tool interface in the ACOSAR project.
Related WP	This paper is relevant for WP3, WP4 and WP6. The interface specification in ACOSAR requires a faithful interface definition. An implementation of the specification on one end of the interface requires an acknowledgement or reception of the information conveyed (principle of handshaking in terms of data, notification and physical signal transfer). Further, it is relevant for the requirements specification in WP1.
Summarized by	Natarajan Nagarajan, ETAS

4.3.2.4 Collaborative Design for Embedded Systems-Co-modelling and Co-simulation

Publication name	Collaborative Design for Embedded Systems- Co-modelling and Co-simulation
Authors	John Fitzgerald; Peter Gorm Larsen; Marcel Verhoef
Year	2014
Reference	ISBN: 978-3-642-54117-9 ISSN: 978-3-642-54118-6 (eBook) DOI: 10.1007/978-3-642-54118-6 Springer-Verlag Berlin Heidelberg
Summary	This book aims to give readers a solid base for developing multi-disciplinary systems. Usually, the developers of embedded systems are from disparate engineering fields, such as software, mechanical and control engineering. In order to allow engineers from each different discipline to continue work within familiar formalism, the concept of "co-simulation" is introduced. It focuses mainly on "co-modelling" of continuous time plants and their discrete-event controller to be implemented on computers. The discrete events in the controller arise for example because we want the controller to switch between several modes or for safety reasons. First, the causality-based method 20-sim for modelling the continuous time dynamics is presented. Second, the object-oriented Vienna Development Method formalism is exposed and is used to implement a supervisory discrete-event controller. Finally, the two formalisms are linked by the Crescendo tool, which allows information exchange among the different models.
ACOSAR relevance	This book is particularly interesting for the project, because it describes how the discrete-event formalism can be embedded in "co-simulation".

	Furthermore it explains how the object-oriented programming can enforce a control mechanism based on supervisory control. It is particularly interesting because it eases fault detection and because it allows to give more structure to the controller.
Related WP	WP 2 (co-modelling and systematic approach for system representation) WP 3 (requirements for interfacing simulation tools)
Summarized by	Daniela Dejaco, Martin Benedikt (ViF)

4.3.2.5 Modelling and simulating cyber-physical systems using CyPhySim

Publication name	Modelling and simulating cyber-physical systems using CyPhySim
Authors	Edward A. Lee; Mehrdad Niknami; Thierry S. Nouidui; Michael Wetter
Year	2015
Reference	International Conference on Embedded Software (EMSOFT), Amsterdam, 4-9 Oct. 2015 DOI: 10.1109/EMSOFT.2015.7318266 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7318266
Summary	CyPhySim is an open-source simulator for simulating Cyber Physical Systems, pursued and developed by the EECS Department from UC Berkeley. Within this paper an overview is given and typical challenges w.r.t CPS are discussed. Different approaches (continuous simulation, quantized state simulation) for handling of the different classes of systems (ordinary differential equations, modal models (hybrid systems), discrete-event models, FMU's, discrete-time (periodic) systems, and algebraic loops). Especially the benefit of the Quantized State Simulation approach is highlighted in connection to "smooth tokens" according simulation time and accuracy.
Project	Center for Hybrid and Embedded Software Systems (CHESS)
ACOSAR relevance	Co-simulation represents modular System-Simulation and thus, the integration of different kinds and classes of systems. Cyber-physical systems are considers integrated the communication mediums where discrete-event simulation is strictly mandatory. This contribution is considering several modern approaches for simulation of CPS, which strongly relates to ACOSAR targets and is of interest for adequate (supporting modern solvers) specification of the ACI interface.
Related WP	WP3,4,5 (requirements for ACI w.r.t to tool interfaces and the communication protocol.
Summarized by	Martin Benedikt (Virtual Vehicle)

4.3.3 Summary and Conclusion

Especially for large-scale, cross-domain and complex systems co-simulation comes into the play and represents a promising possibility to handle the resulting system development complexity by modular system representation. This section indicates, that Hybrid Simulation is strongly necessary and several attempts are already done (e.g. FMI or Quantized State Simulation) to handle the integration challenge for (non-real-time, offline) co-simulation. With respect to ACOSAR, the ACI has to support system integration algorithms (i.e. master algorithms) to be able to establish overall "hybrid" systems. In particular, current (novel) approaches in continuous and discrete-event simulation has to be considered during ACI specification. One big challenge within ACOSAR will be the transfer of current approaches to the real-time domain.

4.4 (Distributed)Real-Time Simulation

4.4.1 Introduction

Real-time simulation aims at providing the right data at the right time, while an actual system is operating. In the V-cycle, it typically occurs on the right side of the V, when real components are integrated and run in parallel to simulated systems.
Distributed real-time simulation refers to the approach of assigning the entire real-time simulation code to different processors or machines. To each machine a physical system can be coupled which consumes and provides data to the simulated system.
There are four main reasons of distributing simulation (see Fujimoto R., 2010):

1. **Reducing execution time**. By operating in parallel, *N* such processors have the potential of speeding up a simulation run by a factor of *N*. However, this is rarely feasible since the execution must be synchronized. Thus, one slow processor may delay the rest either because its output is necessary for them to continue execution or because of inefficiencies in the way the code was distributed. Nonetheless, this approach is necessary for application where multiple processors are needed to complete the simulation computation fast enough so that the simulation results are provided in real-time to the actual systems.
2. **Geographical distribution.** Executing the simulation on geographically distributed computers enables to create real-time systems with multiple participants which are physically located at different sites, a situation which often occurs when e.g. OEMs and supplier are jointly working.
3. **Integrating simulators that execute on machines from different manufacturers.** The current picture in the automotive industry is a heterogeneous one, meaning that simulation tools, as well as accompanying real-time systems or test benches, were developed for testing and validating certain vehicle components. Rather than porting these programs on a single computer, it may be more cost-effective to couple them to create a new distributed real-time system.
4. **Fault tolerance.** Another benefit which comes with distributing simulation is the increased fault tolerance. If one of the involved processors or computers fails, it might be that another processor picks up the work of the failed machine, allowing the simulation computation to proceed and thus avoiding e.g. expensive hardware failures.

In this chapter we are providing an overview on the state-of-the-art techniques for distributed real-time simulation, mention some of the results obtained within industrial projects in the area, and emphasize their relevance for ACOSAR.

4.4.2 Relevant State of the Art Items

4.4.2.1 Parallel and Distributed Simulation Systems

Publication name	Parallel and Distributed Simulation Systems
Authors	Richard M. Fujimoto
Year	2000
Reference	http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471183830.html
Summary	The book is a state-of-the-art guide for the implementation of parallel and distributed simulation technologies. It summarizes techniques for speeding up the execution of simulations across multiple processors and dealing with data distribution over wide area networks, including the Internet. Especially relevant for ACOSAR is the use of parallel and distributed computers in both the modelling and analysis of system behaviour and the

	creation of distributed virtual environments (meta-notation for e.g. Hardware in the Loop Systems).
Project	-
ACOSAR relevance	The book is relevant for WP2, WP3, WP4 and WP5 since it summarizes already available techniques in the area of distributing simulation, also for real-time application. In ACOSAR, we can build upon those techniques to provide solutions to the specific requirements and use cases considered in the project. Further, it can be relevant for the requirements specification in WP1.
Related WP	WP1, WP2, WP3, WP4 and WP5
Summarized by	Corina Mitrohin, ETAS

4.4.2.2 Distributed Simulation in Industry

Publication name	Distributed Simulation in Industry
Authors	Csaba Attila Boer
Year	2005
Reference	http://repub.eur.nl/pub/6925
Summary	The PhD thesis compares techniques and state-of-the-art in different industries with respect to distributed simulation. It offers a list of requirements for designing and developing distributed simulation architectures that would help the acceptance and spread of simulation techniques across industries, and provides an architecture for coupling simulation models and test its appropriateness in industry. The thesis elaborates on the benefits of distributed simulation, applicable for distributed real-time simulation as well. It helps mastering the complexity of simulated systems, it provides techniques for spitting and integrate collaborative work and results, and it supports information hiding by e.g. keeping the sensitive information on dedicated machines.
Project	-
ACOSAR relevance	While ACOSAR is mainly focusing on automotive industry, the cited thesis compares distributed simulation techniques and their application in several industries. The thesis is relevant for WP2, WP3, and WP6. The industry surveys, with respect to application of distributed simulations for industrial use cases, is also a valuable insight for ACOSAR.
Related WP	WP2, WP3 and WP6
Summarized by	Corina Mitrohin, ETAS

4.4.2.3 Real-time co-simulation for the control of an engine test bench

Publication name	Real-time co-simulation for the control of an engine test bench

Authors	Josef Zehetner, Georg Stettinger, Helmut Kokal, Bart Toye
Year	2014
Reference	http://www.atzonline.com/Artikel/3/17730/Real-time-Co-simulation-for-the-Control-of-an-Engine-Test-Bench.html
Summary	The paper shows the first industrial application of ACORTA error handling strategies for real-time co-simulation, developed in the research project ACoRTA at Virtual Vehicle Research Center, together with AVL, Porsche and TU Graz. The topics compensation of latency effects, noisy measurement signals, error compensation for RT-simulation are handled. The application of real-time co-simulation for two ACOSAR relevant real-time testing environments are shown: engine testbench (AVL) and HiL testbench (Porsche).
Project	Research project ACoRTA, Austrian K2 COMET Project, partners Virtual Vehicle, AVL, Porsche, TU Graz
ACOSAR relevance	In this usecase, proprietary interfaces via CAN and UDP for the testbeds are used. These should be replaced by ACOSAR-conform, standardized interfaces.
Summarized by	Josef Zehetner, AVL

4.4.2.4 Distributed Modular Real-Time Simulations of Complex Systems

Publication name	Verteilte modulare Echtzeitsimulation komplexer Systeme. [Distributed Modular Real-Time Simulations of Complex Systems]
Authors	Hubert B. Keller
Year	1988
Reference	http://rd.springer.com/chapter/10.1007%2F978-3-642-74051-0_10
Summary	This paper describes a solution for coupling simulation models (virtual sensors) when these models are placed in distributed systems. The proposed solution enables real time performance of the coupled model system and considers the network transmission delays and different local discretization times of the models. Hence, on the one hand an asynchronous coupling method is proposed– i.e. with variable communication time step size – and where local explicit solvers are used for the numerical integration of every model in the system. On the other hand, signal reconstruction is used to solve problem of the communication delay in coupled-simulation. It provides the required availability of the coupling quantities as well as enables asynchronous data flow between models. The solution was implemented in a single computer, for which a stochastic data transmission delay and a fixed data history (buffer) were adopted.
Project	-
ACOSAR relevance	Argumentation on asynchronous coupling methods and on explicit numerical integration for real time simulation can provide orientation on optimal coupling method for HiL-SiL simulation and therefore for ACI interface definition.

Summarized by	Isidro Corral (Bosch)

4.4.2.5 Methods for real-time simulation of Cyber-Physical Systems: application to automotive domain

Publication name	Methods for real-time simulation of Cyber-Physical Systems: application to automotive domain
Authors	Cyril Faure, Mongi Ben Gaid, Nicolas Pernet, Morgan Fremovic, Grégory Font, Gilles Corde
Year	2011
Reference	7th International Wireless Communications and Mobile Computing Conference (IWCMC), Issue Date: 4-8 July, 2011 http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?tp=&arnumber=5982695&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5982695
Summary	The shortening of time to market and the reduction of development costs calls for precise simulation models with growing computational complexity in (early) validation stages. To fit real-time execution requirements (e.g. in HiL simulations) often a model reduction takes place which is conflicting with the growing need of precise simulation models. In the literature several alternative methods are presented to succeed in real-time simulation of precise models, making model reduction not necessary. It is shown that through the usage of fixed-step (ODE) solvers, multirate integrations, multi-core technologies, inter/intra sub model parallelization (may including the relaxation of precedence constraints) it is possible to execute precise simulation models in real-time. Core objects being pursued are the reduction of computation and the exploitation of parallelism while not harming simulation results.
Project	-
ACOSAR relevance	The paper describes the computation of precise models through inter/intra sub model parallelization. This parallelization can be reached by coupling multiple RT-systems (e.g. HiL systems) within a real-time co-simulation. The ACOSAR project focuses on the coupling of RT-systems via standardized ACI interface. As described in the literature, real-time co-simulation requires the definition and meeting of real-time constraints and the description of inter/intra sub model dependencies to infer parallelization potential. The ACI standard could therefore provide a description format for sub model dependencies and real-time constraints.
Related WP	This publication is relevant for WP2, because WP2 focuses, among other things, on the development of a methodology to transfer the knowledge e.g. from MiL-simulation (e.g. intra sub model dependencies) to real-time co-simulation. Further, it is relevant for WP1 because of important information on sub model dependencies and RT constraints.
Summarized by	Nicolas Amringer (dSPACE)

4.4.2.6 Models for Distributed Real-Time Simulation in a Vehicle Co-Simulator Setup

Publication name	Models for Distributed Real-Time Simulation in a Vehicle Co-Simulator Setup

Authors	Anders Andersson, Peter Fritzson
Year	2013
Reference	Proceedings of the 5th International Workshop on Equation-Based Object-Oriented Modelling Languages and Tools (EOOLT'2013). Linköping University Electronic Press, April 2013
Summary	The paper reports on how a car model in Modelica is used in a new setup for distributed real-time simulation. This co-simulation setup can be used in a number of configurations where hardware in the loop can be interchanged with software in the loop. The paper further elaborates on parameter estimation and numerical issues; it also addresses the specifics of a communication link that is used to connect a moving base car simulator with a real car via a fiber optic communication link.
Project	HiPo RTSIM MODRIO
ACOSAR relevance	This paper describes previous work on the interchangeability of hardware in the loop and software in the loop configurations in the context of co-simulation. It describes the use of a specific HiL setup and how the Modelica model was prepared for the coupling. The described system could serve as an additional use case for ACOSAR, or at least provide further insight into the requirements for ACI.
Related WP	The report on successful experiments with interchanging hardware and models in a distributed real-time simulation setup is worth looking at when addressing the use cases in WP7.

4.4.3 Summary and Conclusion

The development in the last decades yield to fundamental research results in distributed real-time simulation, validated via both academic and industrial use cases. Architectures like HLA (high-level architecture), dedicated communication protocols and even concrete solutions were designed and implemented for distributed real-time simulation. A drawback of that plenty of available techniques is that they weren't been uniformly adopted in available tool and real-time systems.

The immediate task within our project would be to identify, tailor, and extend existent techniques to meet ACOSAR requirements in a uniform manner, meaning that all participating tool vendors and real-time systems providers will consistently adopt them in their products.

For the concrete distributed real-time simulation this means to define and implement an architecture of the distributed system, a concept for time and data management, and a communication protocol between eventually geographically distributed machines and real-time systems.

4.5 Communication Standards

4.5.1 Introduction

An essential part of the ACI Architecture is the ACI Communication Layer. This layer realizes the data transmission between the different systems in a co-simulation scenario. The main goal of the ACI Communication Layer is to abstract the used communication protocol from the ACI software interface. That way, different communication protocols can be supported without changing the ACI. In order to design the ACI Communication Layer properly, it is necessary to consider several state-of-the-art communication protocols from different application domains.
This section considers existing communication standards from the automotive as well as the industrial automation domain. Additionally, some IP-based standards are reviewed. For each communication standard, the most important aspects will be described. Additionally, the relevance for ACOSAR will be determined.

4.5.2 Relevant State of the Art Items

4.5.2.1 AUTOSAR

Standard Name	AUTOSAR (Automotive Open System Architecture)
Standardization Committee	The AUTOSAR development partnership has implemented a three tier structure. Rights and duties are allocated to the various tiers and has been outlined in appropriate agreements [AUT2016].
Chronology	Currently active versions: 3.2 (started in 2011) 4.x (started in 2009) Latest version: 4.2.2 (2015) Source: [AUT2016]
Consortium	The AUTOSAR partnership consists of more than 160 partners (Status: November 2013). For a complete list of all partners refer to [AUT2014].
Major goals	Development of a standardized open software architecture to handle the growing complexity of automotive software and foster their re-usability, scalability, transferability and modularity [AUT2016].
Description	The intention of AUTOSAR is to reduce costs, risk and capacities in automotive software development through the standardization of a layered open software architecture. Thereby suppliers are able to reduce version proliferation and to re-use of software modules over a wide range of OEMs. On the other hand OEMs are able to easily integrate software modules of different suppliers by the introduction of standardized software layers and interfaces [AUT2016].
Available materials	The standard specifications are available under: http://www.autosar.org/specifications/ The AUTOSAR standard specifications consists of two parts: Standard specification Auxiliary material

	The standard specification contains normative results of the AUTOSAR development partnership whereas the auxiliary material is recommended to read and/or use for a better understanding or harmonized usage of the AUTOSAR standard [AUT2016]. Notice that the use of material contained in the specifications requires membership within the AUTOSAR development partnership. Further information regarding the AUTOSAR standard can also be found under: http://www.autosar.org/events-publications/publications/papers-presentations/
ACOSAR relevance	AUTOSAR provides a description format and a methodology for the integration of complete ECU systems in all vehicle domains [AUT2016]. In contrast ACOSAR focuses on the integration of RT-systems (system-of-systems) into co-simulation environments and the specification of a proper methodology. ACOSAR also introduces the so called ACI Communication Architecture which abstracts via a communication layer (ACI Communication Layer) from concrete communication systems (e.g. EtherCAT or CAN) on application level. In AUTOSAR, an abstraction of communication systems also takes place [AUT2013].
Related WP	WP1 is concerned as the abstraction of communication could be a requirement for the ACI. In this context also WP5 is relevant, because WP5 focuses on the development of the ACI Communication Layer for abstraction of typically used communication systems. Because of the definition of a RT-system integration methodology WP2 can also be considered.
Summarized by	Nicolas Amringer (dSPACE)

4.5.2.2 Ethernet UDP

Standard Name	Ethernet UDP - User Datagram Protocol – RFC768
Standardization Committee	The Internet Engineering Task Force (IETF)
Chronology	Release – 28 August 1980
Consortium	Total number of partners: No Formal membership
Major goals	The User Datagram Protocol (UDP) is part of the Internet protocol suite. The protocol provides a datagram communication between programs and computer in a network. As a part of Internet protocol suite (IP), it requires the IP as underlying network protocol. The "protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed." [RFC768]
Description	An UDP message is composed of the UDP-Header and the user data. As part of the IP, a UDP message is embedded into an IP message and the UDP header is directly integrated into the IP header (pseudo header). The network connection (e.g. between two computer) is managed by the IP and the transport of the data between two programs is realized by the UDP. The UDP Header is composed of:

	• 16 Bit frame: Source Port (optional) • 16 Bit frame: Destination Port • 16 Bit frame: Length (length of header and payload in octests) • 16 Bit frame: Checksum ("16-bit one's complement of the one's complement sum of pseudo header") An UDP message is send by the source and can be received by the destination. Due to no acknowledgment or handshake a correct delivery is not guaranteed. Therefore it is not reliable but very lean and fast. [RFC768]
Available materials	User Datagram Protocol [RFC768]
ACOSAR relevance	Very simple communication protocol based on: • Internet layer: IP (IPv4, IPv6) • Link layer: Ethernet Advantages for ACOSAR (performance and latency): • Designed for time-sensitive applications and real-time systems. • Minimum protocol overhead Disadvantages for ACOSAR (stability and robustness): • Unreliable – When a UDP message is sent, it cannot be known if it will reach its destination; it could get lost along the way. There is no concept of acknowledgment, retransmission, or timeout. • Not ordered – If two messages are sent to the same recipient, the order in which they arrive cannot be predicted. • No congestion control – UDP itself does not avoid congestion. Congestion control measures must be implemented at the application level.
Summarized by	Serge Klein (RWTH)

4.5.2.3 Automotive Ethernet

Standard Name	BroadR-Reach
Standardization Committee	OPEN Alliance SIG (ieee802)
Chronology	2014
Consortium	Total number of partners: >200 Promoters: • BMW • Broadcom • Continental • Daimler AG • General Motors Co. • HARMAN • Hyundai Motor Company • Jaguar Land Rover (JLR) • NXP • Realtek Semiconductor Corp. • Renault SA • Renesas • Robert Bosch GmbH • Toyota • Volkswagen Group

	• Volvo Cars Adopters: In April 2016 284 companies of automotive, communication technology and automatization sector are part of the consortium. Please consult http://www.opensig.org/ for complete and up to date list.
Major goals	With Automotive Ethernet respectively BroadR-Reach the OpenSIG Alliance establishes standardization for high speed 100Mbps single pair Ethernet physical layer. Automotive Ethernet meets the in vehicle requirements for automotive communication.
Description	"The BroadR-Reach automotive Ethernet standard realizes simultaneous transmit and receive (i.e., full-duplex) operations on a single-pair cable instead of the half-duplex operation in 100BASE-TX, which uses one pair for transmit and one for receive to achieve the same data rate. " [OSA2016] Within Automotive Ethernet Project interoperability of different vendors or semiconductor suppliers are standardized.
Available materials	• License to specification is available to all OPEN Alliance members under RAND terms via a license from Broadcom Corporation [BRC2016] • BroadR Reach Standard [BRR2016]
ACOSAR relevance	Upcoming automotive control unit communication standard. Future replacement or addition for eg. CAN. High-Speed and low cost communication layer for automotive applications.
Summarized by	Rene Savelsberg (RWTH)

4.5.2.4 Ethernet TCP-IP

Standard Name	Ethernet TCP – Transmission control program – RFC675
Standardization Committee	The Internet Engineering Task Force (IETF) / Cerf's networking research group at Stanford
Chronology	Release – December 1974
Consortium	Total number of partners: No Formal membership
Major goals	Transmission Control Protocol (TCP) is part of the Internet protocol suite. "TCP/IP provides end-to-end connectivity specifying how data should be packetized, addressed, transmitted, routed and received at the destination." [RFC675]. The goal is a reliable communication between two programs.

Description	Owing to the major goal of reliable communication, TCP includes various mechanisms to guarantee a proper communication.
	• Establishing connection:
	  ○ Initiator(client) sends request to responder (server)
	  ○ A proper connection will be established by a three way handshake.
	• During transmission and correct established connection:
	  ○ Acknowledgment of packets
	  ○ Duplicate detection
	  ○ Scheduling order of letters
	  ○ Open closing of ports to maintain correct order
	  ○ Flow control
	  ○ Sequencing

	◦ Checksum • Closing connection: ◦ Connection is closed by a three way handshake. Major advantage and disadvantage is that each packet has to be acknowledged. This process provides a reliable communication but results in slow and non-deterministic behaviour. [RFC1122]
Available materials	Specification of internet transmission control program [RFC675] Requirements for Internet Hosts Communication Layers [RFC1122]
ACOSAR relevance	Advantages for ACOSAR (performance and latency): • Very populate communication protocol. • Mechanisms to ensure data integrity Disadvantages for ACOSAR (stability and robustness): • Not designed for real time application • Non deterministic behaviour • Slow
Summarized by	Serge Klein (RWTH)

4.5.2.5 EtherCAT

Standard Name	EtherCAT IEC 61158 / IEC 61784-1 CPF12
Standardization Committee	IEC International Electrotechnical Commission
Chronology	The EtherCAT Technology Group is an official partner of the IEC (International Electrotechnical Commission) working groups for digital communication. Specification: IEC/PAS 62407 (2005) Replaced by: IEC 61158, IEC 61784-2, IEC 61800-7 (SERCOS and CANopen drive profiles)(2007) XML description: ISO 15745-4 Safety over EtherCAT: IEC 61784-3 (2010) Installation Profile: IEC 61784-5 (2008)
Consortium	“The EtherCAT Technology Group (ETG) was established in 2003, and is the industrial Ethernet user organization with the most members in the world today. A wide range of industrial controls vendors, OEMs, machine builders, and technology organizations from around the world comprise the ETG member roster.” [ECA2016] Total number of partners: 3505 (in 2016)

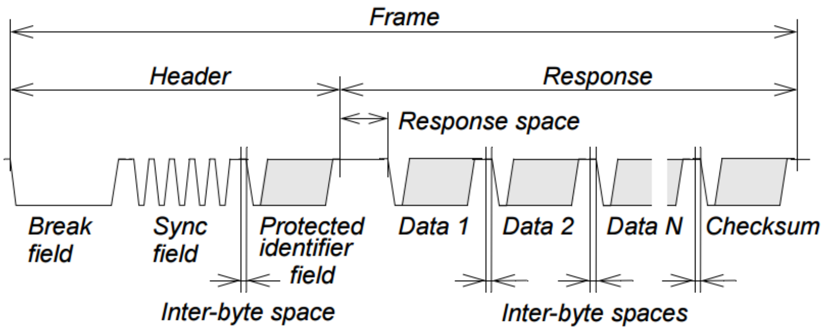
Major goals	"EtherCAT is a real-time Industrial Ethernet technology originally developed by Beckhoff Automation." [ECA2016] EtherCAT protocol which is meets hard and soft real-time requirements of automation technology. "The main focus during the development of EtherCAT was on short cycle times (≤ 100 µs), low jitter for accurate synchronization (≤ 1 µs) and low hardware costs. "[ECA2016]
Description	The functional principle of the master and multi slave communication protocol is described in the standardization mentioned above. "The EtherCAT master sends a telegram that passes through each node. Each EtherCAT slave device reads the data addressed to it "on the fly", and inserts its data in the frame as the frame is moving downstream. " [ECA2016] This communication concept achieves a high performance and minimum latencies and jitter. The EtherCAT protocol uses standard Ethernet Identifier (0x88A4) and standard EtherCAT physical transport layer. "In addition to cyclical data, further datagrams can be used for asynchronous or event driven communication." [ECA2016] Beside of master to slave communication, EtherCAT also supports slave to slave communication. Line, tree, star, or daisy-chain topologies are supported. By synchronizing all nodes with distributed clock feature the jitter of EtherCAT is less than 1µs. Data integrity is ensured by checksum evaluation of the moving frame. Debugging of communication errors are possible with simple Ethernet sniffing tools.
Available materials	• EtherCAT Consortium [ECA2016] • IEC 61158-1:2014: Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series [IEC2014] https://webstore.iec.ch/publication/4624
ACOSAR relevance	• **Real-Time Capable:** Bus system designed for real-time applications, can be used as communication protocol (below ACI communication layer) between RT systems or RT and offline systems. • **Performance: Very** short cycle times smaller than 100 µs and smaller jitter than 1µs are beneficial for numerical stable ACOSAR coupling of RT-Systems • **Common Fieldbus:** Common and highly distributed standard in automation systems. • **Debugging:** Standard Ethernet frames can easily be debugged during development of ACOSAR protocol. • **Co-Simulation Scenarios:** Line, tree, star, or daisy-chain bus topologies are possible for complex Co-Simulation setups. • **Coupling Quality:** Minimal jitter and very low latencies are advantageous for stable and robust Co-Simulations with distributed real time systems.

- **Reliability:** Data integrity is ensured by checksum evaluation.
- **ACOSAR master-slave system handling**: Master-Slave concept of communication channel suits ACOSAR Master-Slave requirements. By use of e.g. event triggered frames the Co-Simulation initialization, start and stop can be handled. For measurement data and safety mechanisms as well as Co-Simulation data exchange cyclic frames can be used.

Summarized by Rene Savelsberg (RWTH)

4.5.2.6 Local Interconnect Network

Standard Name	LIN (Local Interconnect Network), ISO-Norm 17987 Part 1-7
Standardization Committee	ISO/TC 22 (Road Vehicles)/SC 31 (Data Communication)
Chronology	LIN 1.0 1999-07-01 Initial Version of the LIN Specification LIN 2.2A 2010-12-31 Corrected wakeup signal definition in chapter 2.6.2
Consortium	Total number of partners: 7 core members OEM level • AUDI AG • Bayerische Motoren Werke AG • Daimler AG • Volkswagen AG • Volvo Car Corporation Supplier level • Motorola • Freescale Halbleiter Deutschland GmbH • Mentor Graphics Corporation Tool vendors • Volcano Communications Technologies (VCT) Academic partners • none
Major goals	LIN-BUS is a serial network protocol used for connecting of components/sensors and vehicle. This is a low cost alternative for the CAN-Bus protocol standard. [LIN01]

Description	The LIN is a serial communications protocol which efficiently supports the control of mechatronics nodes in distributed automotive applications. The intention of this specification is to achieve compatibility with any two LIN implementations with respect to the scope of the standard, i.e. from the application interface, API, all the way down to the physical layer. LIN provides a cost efficient bus communication where the bandwidth and versatility of CAN are not required. The specification of the line driver/receiver is based on the ISO 9141 standard with some enhancements regarding the electromagnetic interference (EMI) behavior. The LIN standard is structured in master and slave task. The master controls which message is to be preferred to transmit it via the bus. He determines the order/priority of messages and monitors data (checks bytes and controls error handler). However, a slave task is one of 2 – 16 members on the bus. He receives or transmits data when an appropriate ID is sent by the master.  **Figure 2: Structure of LIN bus [LIN01]** The LIN consortia have concluded its work with the finalization of the LIN Specification 2.2.A. Currently the latest LIN specification is being transcribed to the ISO (International Organization for Standardization) as part of the process to be accepted as ISO standard ISO 17987 Part 1-7. For this reason the community size cannot estimated. But they are still typical applications for the LIN standard, such as connecting sensors and actuators with low requirements. Usually LIN buses are installed in comfort systems, e.g. car sets, car doors, air conditioner and sideview mirrors. [LIN01]
Available materials	http://www.iso.org/iso/catalogue_detail.htm?csnumber=61227 http://www.cs-group.de/fileadmin/media/Documents/LIN_Specification_Package_2.2A.pdf
ACOSAR relevance	Related to WP4 and WP5 this widely used LIN Bus communication protocol should be an objective of ACOSAR project. For verification and testing of sensor and actuator in automotive applications, a LIN RT-Interface and RT-Simulation is necessary. Anyway, the use cases of ACOSAR do not make use of the LIN standard.
Related WP	WP4: RT-System Interface WP5: Communication System Protocol
Summarized by	Viktor Schreiber (Technische Universität Ilmenau)

4.5.2.7 ProfiBus

Standard Name	ProfiBus (Process Field Bus), IEC 61158 Type 3 and IEC 6178
Standardization Committee	IEC
Chronology	Initially the FieldBus standard was founded and promoted by BMBF (German department of education and research), 1987-1989 [Bus01] In 1989, ProfiBus manufacturers and users created the ProfiBus User Organization (PNO). This group was, and still is, a non-commercial venture. A larger group was formed in 1995 and named PROFIBUS and PROFINET International, or PI. As one of the largest Fieldbus user association in the world, PI is able to undertake many tasks vital to the progression of PROFIBUS. Like the PNO, PI educates users on PROFIBUS and helps advance its placement throughout the world. The ProfiBus standard was developed historic in in three versions. 1989 the ProfiBus-FMS (Fieldbus Message Peripherals) was established for controlling and automation systems. In terms of high-speed communication and decentralized peripheral devices this version was advanced to ProfiBus-DP (Decentralised Peripherals) 1993. For this reason the FMS protocol is not anymore a part of the FieldBus standardization. The latest version ProfiBus-PA (Process Automation) was developed in 1995 to create intrinsically safe installations and devices with bus powered supply. [Bus02] Further developing plans are not recognized.
Consortium	Total number of partners: over 1400 members in the Profibus & Profinet International (PI) worldwide http://www.profibus.com/nc/pi-organization/members/
Major goals	Basically there are two specifications of the ProfiBus standard (ProfiBus DP/PA). ProfiBus DP: Is used for connecting a controller with sensors and actuators. This Version is the common used specification. As an option, it is possible to execute diagnostics. Furthermore, the ProfiBus DP is separated in the functionalities /options DP-V0 (cyclic data exchange), DP-V1 (acyclic data exchange) and DP-V2 (isochronous mode and data exchange broadcast / slave-to-slave communication). ProfiBus PA: The PA specification is designed to monitor measuring equipment in an automation process. Especially under explosion/hazardous conditions. The ProfiBus cable standard allows supply of power/energy – not only signals. Explosive hazard potential is avoided by limiting the current flow – even in case of a malfunction. The number of attached devices is limited.
Description	The ProfiBus standard covers the robust, fast and cheap communication between controller and sensors/actuators. The ProfiBus standard is structured in master and slave devices. It can accommodate a maximum of 32 participants (masters or slaves) to each segment and without DP/PA coupler. You cannot connect DP and PA on the same segment. Therefore a specific gateway to converters the protocols required. [Bus03]Master Devices: • master devices determine communication permission of slave devices • master participants share the bus permissions

- if a master participant is in possession of bus permission (token), a master device is authorized to send a message without external request or demand
- typical master devices are controller

Slave Devices:

- a slave device has no bus permissions
- it is only authorized to cancel requested messages or to send a message on request to the demanding master
- typical slave participants are peripheral devices like actuators and sensors

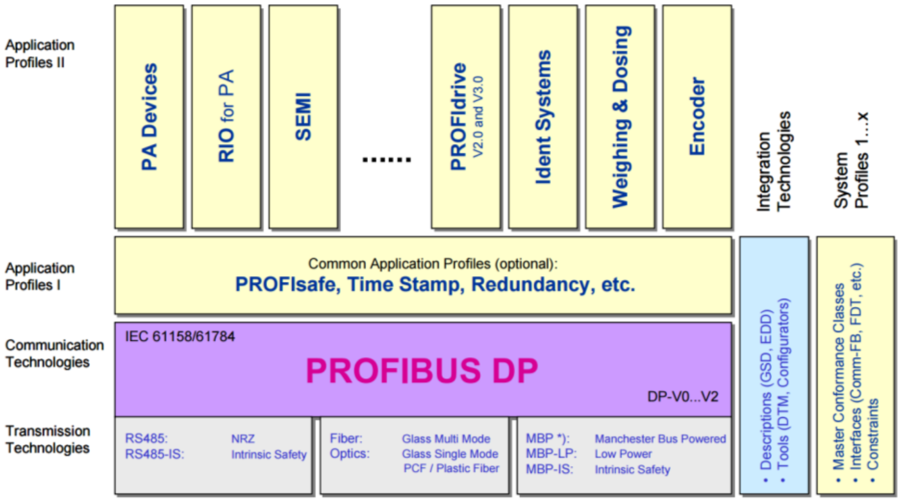


Figure 3: ProfiBus structure [Bus04]

The PI community is still active. A further development is not known. But this standard is supported by the committees and Project/Working Groups, which carry out Marketing, Quality, Application Profiles and Integration.

Typical applications are automated plants, facilities, equipment or devices in the fields of Production, Process and Civil Engineering.

Available materials

https://webstore.iec.ch/searchform&q=61158

http://www.profibus.com/

http://www.feldbusse.de/Profibus/profibus.shtml

ACOSAR relevance

In relation to ACOSAR ProfiBus is a common industrial standard for connecting and automating of networked controller-, sensor- and actor-devices. The connection of HiL-controller-devices with test-rigs may an important part for the use case scenarios.

The PI community is very big worldwide, i.e. in the future the meaning is still significant. But ProfiBus standard is not applicable in modern vehicles. Although the ProfiBus standard is not used in modern vehicles, but for automation of automotive test-rigs. Therefore, it may contribute the connecting of simulation and test-rig platforms in the ACOSAR project.

Related WP

WP4: RT-System Interface

WP5: Communication System Protocol

Summarized by

Viktor Schreiber (Technische Universität Ilmenau)

4.5.2.8 Media Oriented Systems Transport

Standard Name	MOST (Media Oriented Systems Transport), no standard name known
Standardization Committee	MOST is an approved standard for multimedia networking in vehicles with organizations such as ISO, SAE, ITS, AMI-C etc. SMSC and MOST are registered trademarks of Standard Microsystems Corporation ("SMSC"), now owned by Microchip Technology.
Chronology	First Release 1998 MOST Specification Framework Rev. 1.1, 1999 MOST MAMAC Specification Rev. 1.1, 2003 MOST Dynamic Specification Rev. 1.2, 2006 MOST Specification Rev. 2.5, 2006 The MOST Standard is updated continuously by the MOST Cooperation
Consortium	Total number of partners: 5 core members OEM level • Audi • BMW • Daimler Supplier level • HARMAN Automotive Division • Microchip Technology Currently the consortium also consists of more associated partners like system integrators (e.g. Porsche, Volvo, Volkswagen, Jaguar) and suppliers (e.g. Bosch, DENSO, Alpine, Bose). [MST01]
Major goals	MOST is a high-speed multimedia ring network technology with a big bandwidth optimized for automotive applications such as DAB, Navigation Systems, etc. The MOST Bus uses synchronous/asynchronous data communication for transmitting a Bit-stream. Depending on the bandwidth the MOST is classified in MOST25, MOST50 and MOST150. A MOST network is able to provide up to 64 devices. Safety critical applications use a double ring topology in respect to redundancy.
Description	The MOST technology supports a reliable and simple solution to cover audio, video and data communications. MOST defines the protocol, hardware and software layers necessary to allow for the efficient and low-cost transport of control, real-time and packet data using a single medium (physical layer). Media currently in use are fiber optics, unshielded twisted pair cables (UTP) and coax cables. MOST also supports various speed grades up to 150 Mbps. MOST • uses a single interconnection to transport audio, video, data and control information • supports different physical layers (fiber-optic, UTP, coax) • Supports 25, 50 and 150 Mbps • provides the connectivity backbone to network a variety of multimedia interfaces There are three communication channels open to applications: • Control Channel: for event-oriented transmission with low bandwidth (10 kBits/s) and short package length

- Asynchronous Channel: Packet oriented transmission with large block size and high bandwidth
- Synchronous Channel: Continous data streams that require high bandwidth

A MOST network must have a number of masters for different functions. The masters can be contained in the same device:

- Timing Master: controls the timing of the network and thereby the synchronization between the devices
- Network Master: sets up the network and allocates addresses to the devices
- Connection Master: sets up the synchronous communication channels between devices
- Power Master: monitors the power

A MOST device consists of three parts:

- Physical interface
- Network Services: a Network Interface Controller (NIC) handles these services. Modern NICs have a built in processor and are called INICs, Intelligent NICs
- Function Blocks (FBlocks): these take care of the services that the device can supply

A MOST device is not connected to a bus in the common sense. It has an inport and an outport and passes the information from the inport to the outport.

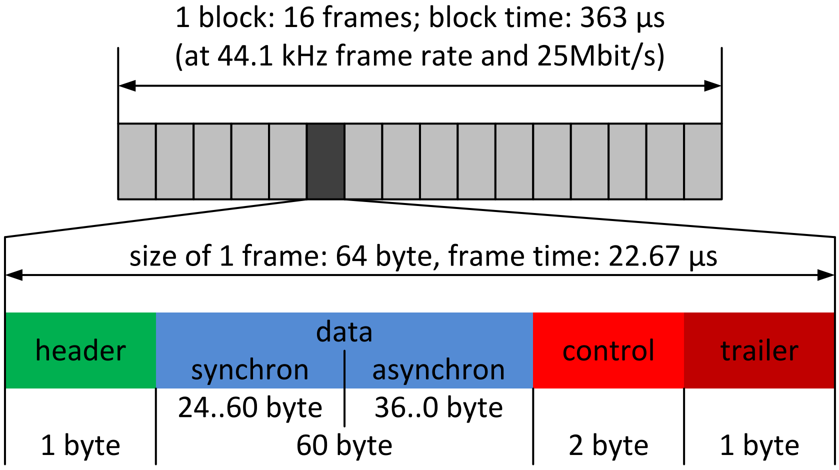


Figure 4: Structure of MOST standards [MST02]

The consortium is still active and keeps the MOST standard up to date.

http://www.mostcooperation.com/membership/members-list/

DAB, Navigation Systems, CD-/DVD Player, Mobile Phone, Bluetooth Devices, TV-Tuner

Available materials	http://www.mostcooperation.com/publications/specifications-organizational-procedures/ http://www.mostcooperation.com/
ACOSAR relevance	The MOST standard relates to multimedia communication for vehicles. The use cases in ACOSAR do not deal with this issue. In case of requirements

	for high Bit-stream and a high bandwidth an interface for this technology should be included.
Related WP	MOST is not related to ACOSAR WP or use cases.
Summarized by	Viktor Schreiber (Technische Universität Ilmenau)

4.5.2.9 CANopen

Standard Name	CANopen, CiA 301, EN 50325-4
Standardization Committee	EN European Standard CAN in Automation (CiA)
Chronology	**1993** - Pre-development of CANopen within an Esprit project under the chairmanship of Bosch **2011** - Publication of CiA 301, CANopen application layer and communication profile 4.2 (public) Whole chronology can be found at: http://www.can-cia.org/can-knowledge/canopen/canopen-history/
Consortium	Total number of partners: 627 in CiA International User's and Manufacturer's group. All partner can be found at: http://www.can-cia.org/about-us/members/
Major goals	CANopen is a higher layer protocol based on CAN (Controller Area Network). The goal is to enable interoperability between different devices based on CAN communication [CAN2016].
Description	CANopen is a standardized application for distributed automation systems based on CAN. The main features of CANopen are [CAN2016]: • Transmission of time-critical process data via PDO (Process Data Object). PDOs are used for real-time communication • Standardized device description (data, parameters, functions, programs) enables accessing all important data of a device from the "outside", i.e. via the CAN bus • Standardized services for device monitoring, error signalisation and network coordination • Standardized system services for synchronous operations (central time stamp message) • Standardized help functions for configuring baud rate and device identification number via the bus • Standardized assignment pattern for message identifiers for simple system configurations • EDS (Electronic Datasheet) and DFC (Device Configuration File) specify the CANopen devices in a machine readable form. This makes it easy to integrate a new device into a CANopen network • Each device sends a "Boot-up" message after start-up to show its availability. Every node in the CANopen network gets informed about the presence of a new node. • Network Management is organized via Master/Slave principle. There is only one master for network management, all other participants are slaves.

	• Node monitoring includes two different concepts: – "heartbeat" principle: Automatic transmission of a "heartbeat" message at regular intervals by the network nodes – "node guarding" principle: Cyclic querying of the node state by the master • Emergency telegrams indicate error situations. **Typical Applications:** medical equipment, off-road vehicles, maritime electronics, railway applications, or building automation [CAN2016]
Available materials	Specification and further information: http://can-cia.org CAN Newsletter: http://can-newsletter.org/ CANopen Basics: http://www.canopensolutions.com/index.html **Books:** • Embedded Networking with CAN and CANopen, [PAK2008] • CANopen: Das standardisierte, eingebettete Netzwerk, [ZEL2008] • CANopen Implementation: Applications to Industrial Networks, [FAB2000]
ACOSAR relevance	The main relation to ACOSAR is given by communication aspects defined in CANopen. The following concepts may be helpful for ACOSAR: – The CANopen standard makes it possible to connect devices from different manufacturers via a standardized device description – There's the possibility to automatically detect new nodes – The standard provides standardized services for device monitoring (e.g. "heartbeat") – The standard provides system services for synchronous operations (synchronization message, central time stamp message) The standard is applicable at the SW/HW development stage. In contrast to CANopen, the main goal of ACOSAR is its abstraction of the communication layer. Nevertheless, some mentioned concepts of CANopen may be transferred to ACOSAR in order to develop the ACI.
Related WP	WP5 (Communication System Protocol).
Summarized by	Markus Tranninger, Nadja Marko (Virtual Vehicle)

4.5.2.10 OPC Unified Architecture

Standard Name	OPC UA (Open Platform Communications Unified Architecture) IEC 62541
Standardization Committee	IEC – International Electrotechnical Commission
Chronology	**2010 -2012:** IEC/TR 62541 Parts 1 to 10 **12/2015:** EtherCAT Technology Group (ETG) and OPC Foundation recently started a cooperation to develop a common interface to ensure a consistent communication between EtherCAT and OPC UA for industry 4.0.
Consortium	IoT@Work Total number of partners: 6

OEM level:

Centro Ricerche FIAT (CRF)

Tool vendors:

Siemens AG

Academic partners:

City University London (CITY), inIT - Institut Industrial IT, CRS TXT Division (Corporate Research and Innovation), European Microsoft Innovation Center (EMIC)

Major goals

OPC UA is an industrial machine to machine communication protocol for interoperability which supports a secure, reliable and platform-independent exchange of information. The data exchange is possible between products of different manufacturers and various operating systems [OPC2015].

The main features of OPC UA are [OPC2015]:

- Platform and vendor independent
- Standardized communication via internet and firewalls
- Service-oriented architecture
- Protection against unauthorized access
- Accessibility and reliability
- Simplification by unification

Description

The OPC-UA standard is based on specifications that were developed in close cooperation between manufacturers, users, research institutes and consortia, in order to enable safe information exchange in heterogeneous systems. The specification is a multi-part specification and consists of the following parts [OPC2016]:

1. Concepts
2. Security Model
3. Address Space Model
4. Services
5. Information Model
6. Mappings
7. Profiles
8. Data Access
9. Alarms and Conditions
10. Programs
11. Historical Access
12. Discovery
13. Aggregates

OPC-UA was designed to support a wide range of systems. These systems can be diverse in terms of size, performance, platforms and functional capabilities. In order to meet these objectives, the following basic functionalities were specified for OPC-UA [OPC2015]:

- Transport – enables the data exchange mechanisms between OPC-UA applications. Two protocols are specified for this purpose. One is a binary TCP protocol, optimized for high performance and the second is web service-oriented.

	• Meta model – specifies the rules and basic components for publishing an information model via OPC UA. • Services – represent the interface between a server as information provider and clients as information users. OPC UA uses universally applicable technologies. It enables data exchange from production to ERP systems, including geographically distributed devices.
Available materials	[OPC2016] - Specification, books, examples etc. [OPC2015] – OPC UA and Industrie 4.0
ACOSAR relevance	OPC UA addresses communication aspects regarding real-time systems as well as its interfaces. OPC UA is a standard which includes a lot of specifications which will help to specify the ACI too. The standard is applicable at the SW development stage. OPC UA APIs are available in several programming languages. (Commercial SDK is available for C, C++, Java and .NET. Open-source stacks are available at least for C, C++, Java, Javascript (node) and Python.) In contrast to OPC-UA, which focuses on TCP/IP, the main goal of ACOSAR is its abstraction of the communication layer. Nevertheless, all helpful pre-work should be considered. Aspects that can be useful for ACOSAR are: • Platform independence • Extensibility: ability to add new features without affecting existing applications • OPC Data Access • OPC Alarms & Events • OPC Historical Data Access
Related WP	WP 4 (RT-System Interface), WP 5 (Communication System Protocol).
Summarized by	Markus Tranninger, Nadja Marko (Virtual Vehicle)

4.5.2.11 FlexRay

Standard Name	FlexRay
Standardization Committee	International Organization for Standardization (ISO)
Chronology	Start in 2000, expansion from 2001-2004. Consortium lead acquired by Freescale in 2004. Consortium lead acquired by NXP in 2006. End of consortium in 2010. Currently no known future developments.
Consortium	OEM level: BMW, Daimler, GM, VW Supplier level: Motorola, Philips, Bosch, NXP
Major goals	Flexray was initially planned to overcome the limitations of event-triggered bus systems like CAN, to provide a real-time capable, time-triggered bus system with sufficient bandwidth.
Description	Flexray is a serial, deterministic and fault tolerant bus system. Version 3.0.1 consists of the following parts:

	• Protocol specification • Electrical physical layer application notes • Electrical physical layer conformance test specification • Electrical physical layer specification • Protocol conformance test specification The consortium vanished in 2006 after ISO standardization in ISO 17458-1 to ISO 17458-5. Many research activities and European projects around 2010. Typical applications of the standard are e.g. in the powertrain area.
Available materials	The consortium website went offline. ISO standard 17458 available. Quite a lot of scientific papers available, targeting various aspects of Flexray. Flexray basics available in [RAU2007]
ACOSAR relevance	Next to CAN, Flexray is a state-of-the-art target communication medium to potentially realize the ACI communication layer for wired communication, due to the fact that Flexray supports time-triggered real-time communications.
Related WP	WP 5
Summarized by	Martin Krammer (Virtual Vehicle)

4.5.3 Summary and Conclusion

In this section, several communication standards from different application domains have been investigated. Some of the communication standards are IP-based while other standards come from the automotive or industrial domain. Especially the relevance of each communication standard for the ACOSAR project has been considered.

This section has listed IP-based communication standards, namely TCP and UDP. These standards specify well established protocols that are widely used, especially for the communication between personal computers via local area networks (LAN) or wide area network (WAN). Coupling systems via the internet is one use case that shall be addressed in this project, hence IP-based communication standards should be taken into account.

Next to IP-based protocols, this section also considered communication standards that are particularly used in the industrial automation domain. Coupling different systems such that these systems can communicate under real-time constraints is a typical scenario here. There already exist some well-established standards which fulfil those communication requirements, e.g. EtherCAT. Since one purpose of the ACI is to interconnect real-time systems, it makes sense to consider some of the most common communication standards in that area and to find out whether they can be adapted for this project.

Lastly, this section covered several automotive communication standards, especially bus protocols like CAN and LIN. Most of the use cases that are considered in the ACOSAR project are driven by OEMs, suppliers and tool vendors from the automotive domain. Furthermore, most of the partners in the project come from this domain. To make sure that the ACI is useful for the partners and can be used in their use cases, the ACI communication layer should support automotive bus protocols. Additionally, the AUTOSAR standard has been taken into account. This standard involves a concept to abstract the communication topology from the used communication protocol. This concept is similar to the desired architecture of the ACI communication layer. Especially this standard should therefore be taken into account when this layer is planned.

All in all, the ACI communication layer should be able to support a large variety of communication standards in order to make sure that all desired use cases can be addressed properly. In this context, the given list of communication standards can only be seen as a starting point to get an overview over the different communication standards and architectures. The ACI communication layer should therefore be flexible enough to support the given communication standards while it can still be extended to support some additional communication standards in the future.

4.6 Real-Time Systems

4.6.1 Introduction

A real-time system is a system whose behaviour is a function of physical time. It usually consists of a physical component, and a controller, an entity observing and eventually controlling the physical part. Parallel to these two entities, a virtual reality can be "created" via simulation environments running simulated models.
Knowledge on control systems was available already in the Hellenic period (see "A Brief History of Automatic Control", by Stuart Bennett). The first automatic control systems is often cited to be the water clock of Ctesibius in Alexandria, at around third century BC. At the time writing, we envision cyber physical environments, where smart objects connect to each other, ad-hoc, and build the Internet of Things.
From engineering perspective, different contributions are needed to make such visions true. The methodological contribution accompanies the engineer on the way from an idea to a final product. The technological contribution provides tools, both software and hardware, which enables the engineer to iteratively build up new product.
In this chapter, we present an overview on both methodological and technological advancements, which were achieved in the area of real-time systems. This overview shall help us to identify the existing gaps and develop the necessary interfaces for real-time systems, such that the state of the art in real-time applications can be improved by the new ACI.

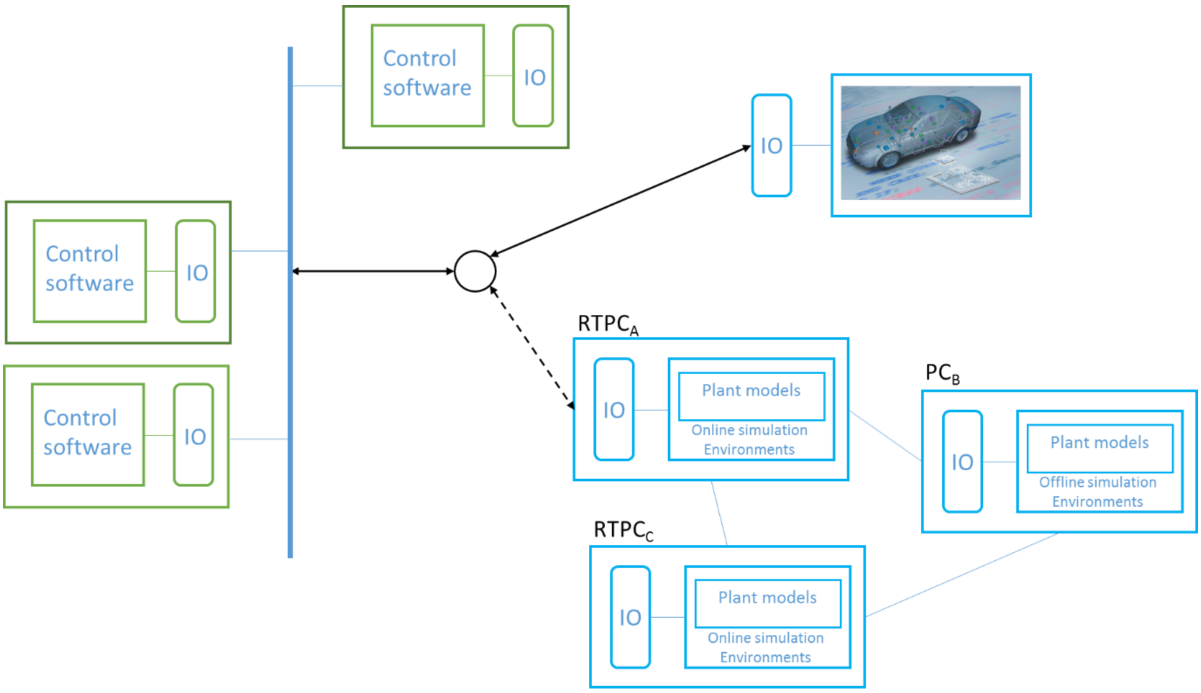


Figure 5: Generic picture of a real-time system. It consists of real hardware (top-right) controlled by control software (*controllers* on the left-side). In parallel, a simulated environment can run in parallel, consume and provide data.

4.6.2 Relevant State of the Art Items

4.6.2.1 2.6.2.1 Bus-based Architecture Concept HIL Test Benches

Publication name	Busbasiertes Architekturkonzept für Hardware-in-the-Loop-Prüfstände
Authors	Constantin Brückner, Bettina Swynnerton

Year	2014
Reference	[BRS2014]
Summary	The article describes a bus-based architecture concept for building modular, heterogeneous HiL- test benches. It describes the emergent need for multiple HiL-test benches in the future car development to master e.g. the growing complexity. The authors propose an architecture built upon a data-oriented middleware, instead of a message-oriented one. Each component provides data and consumes data. It can observe the data and check if the time constraints were fulfilled. Concretely, the OMG (Object Management Group) Data Distribution service (DDS) is used for data exchange. A test bench manager with data models and a software library couples individual HiL modules to the HiL bus.
Project	-
ACOSAR relevance	The article shows how HiL-systems provided by different suppliers can be coupled using an open standard (DDS) for data exchange. Furthermore, the implementation of the standard (RTI Connext DDS) is robust, scalable and real time compatible. The approach followed in the paper can serve as orientation in ACOSAR, with respect to communication protocols. It could be also used as an additional use case to show how HiL-systems can be coupled via ACI.
Related WP	WP1: The publication describes requirements for possible industrial use cases of ACOSAR such as combining multiple test benches and models, abstractions for central test automation, recording and stimulation of data between test benches. WP4+5: The publication shows a possible way to handle data transfer and communication between test benches. The DDS has a system view accessible to each test bench. In case of errors such as delays, feedback is provided to the user. The definition of the exchanged data is proprietary. WP7: A similar environment could be used for a use-case to validate ACI and compare it with the DDS approach.
Summarized by	Thies Filler (Volkswagen)

4.6.2.2 2.6.2.2 Real-time Systems: Design Principles for Distributed Embedded Applications

Publication name	Real-time Systems: Design Principles for Distributed Embedded Applications
Authors	Hermann Kopetz
Year	2011
Reference	[KOP2011]
Summary	The book offers an excellent overview on real-time systems. It starts with basic definitions, and goes over to scheduling algorithms and Internet of Things. According to Kopetz, a real-time system changes as a function of physical time. A real-time system can be decomposed into subparts, called clusters: the physical plant or machine that is to be controlled (*the controlled cluster*), the real-time computer system (*the computational cluster*) and, the human operator (*the operator cluster*) (cp. Figure in [KOP2011], page 2).

	Within ACOSAR, all those clusters together with the standardized compatibility among each other play a role. On the computational cluster, several simulation tools, running on eventually different machines, both online and offline will simulate models, providing results to the controlled objects and to the operators. The communication within the computational cluster and from the computational cluster to the controlled cluster shall be optimized in such a way that the real-time requirements are fulfilled. The Controlled Objects, being test benches, real ECUs, or other hardware, coming from different suppliers, can be coupled via the ACI, as well.
Project	-
ACOSAR relevance	Within ACOSAR, the book will help to fix and stick on correct terminology and to build upon the current state-of-the art in real-time systems. It summarizes established techniques in the area of real-time communication, real-time systems, synchronization, and thus makes it relevant for the entire project, from WP1 to WP 7.
Related WP	WP1 to WP 7
Summarized by	Florian Pölzlbauer, Nadja Marko (Virtual Vehicle)

4.6.2.3 The Complexity Challenge in Embedded System Design

Publication name	The Complexity Challenge in Embedded System Design
Authors	Hermann Kopetz
Year	2008
Reference	[KOP2008]
Summary	The paper offers some impulses to reduce the cognitive complexity of embedded computer systems. The author argues that given the limited cognitive capabilities of humans, complex systems can be can be modeled at different levels of detail by models that are simple enough for the human mind to understand. As simplification strategies, following capabilities are enumerated: - Abstraction, the formation of a higher-level concept that captures the essence of the problem-at-hand and reduces the complexity of the scenario by omitting irrelevant details. - Partitioning (or separation of concerns), the spatial division of the problem scenario into nearly independent parts that can be studied in siolation. - Segmentation, the temporal decomposition of complex behavior into smaller parts that can be processes sequentially, one after the other. The author identifies and distinguishes between computational components (can be in turn, either time-triggered or event-triggered) and interface components. Such a structured approach will be helpful in ACOSAR, as well, to help mastering the complexity and variety of tool, protocols, and hardware involved.
Project	-

ACOSAR relevance	The paper is useful as a guideline for the entire project, in that only sticking to certain rules as abstraction, partitioning, segmentation will help us to come up with a sound yet understandable and thus implementable ACI. The structure of the work-packages within ACOSAR already go into the right direction of partitioning and segmentation; these principles have to be further followed when refining the specification of ACI.
Related WP	WP1 to WP7
Summarized by	Corina Mitrohin, ETAS

4.6.2.4 Hard Real-Time Computing Systems

Publication name	Hard Real-Time Computing Systems
Authors	Giorgio Butazzo
Year	2011, 3rd edition
Reference	http://www.springer.com/us/book/9781461406754
Summary	The book offers a comprehensive overview in the challenges and algorithms for hard real-time computing systems. The use cases addressed within ACOSAR will yield to systems of real-time systems, which are in turn real-time systems. This means that we have to deal and define solutions for the typical real-time challenges. These are (see also Page 12 in the reference): - Timeliness. Results have to be correct not only in their value but also in the time domain. As a consequence, the operating system must provide specific kernel mechanisms for time management and for handling tasks with explicit timing constraints and different criticality. In ACOSAR, this becomes more tedious, since a network of real-time computers will have to ensure the timing requirements, i.e., cooperative real-time operating systems has to agree and guarantee timeliness. - Predictability. To achieve a desired level of performance, the system must be analysable to predict the consequences of any scheduling decision. If some task cannot be guaranteed within its time constraints, the system must notify this fact in advance, so that alternative actions can be planned to handle the exception. In a networked real-time system, it has to be defined which component will be in charge of guaranteeing the predictability of the system. A related question will be under which circumstances a global predictability can be derived from local predictability. - Efficiency. Most of real-time systems are limited in computational power and memory resources. In these systems, an efficient management of the available resources by the operating system is essential for achieving a desired performance. In networked real-time systems, the trade-off between communication overhead and distributed computation shall be investigated and a conscious decision shall be made as of which scenario is the best for the use case in turn. - Robustness. Real-time systems must not collapse when they are subject to peak-load conditions, so they must be designed to manage all anticipated load scenarios.

	- Fault tolerance. Single hardware and software failures should not cause the system to crash. With a network, the communication channels become a further component which shall be made fault-tolerant. - Maintainability. The architecture of a real-time system should be designed according to a modular structure to ensure that possible system modifications are easy to perform. Even though certain design principles were followed in that direction and nowadays successful real-time applications exist, within ACOSAR we shall strive at harmonizing the coupling interfaces, enabling this modularity beyond one tool vendor or real-system provider.
Project	-
ACOSAR relevance	The book is relevant as conceptual backbone for the entire ACOSAR project. It summarizes fundamental algorithms for scheduling and real-time operating systems, which are component parts of ACOSAR, as well.
Related WP	WP1 to WP7
Summarized by	Corina Mitrohin, ETAS

4.6.3 Summary and Conclusion

In this chapter, we revised some of the relevant publications and results in the field of real-time systems. There is a plenty of results we can benefit and build upon. Also projects were specific users and tool providers succeeded to realize complex, networked real-time systems. Structured approaches and requirements are available for networked real-time systems, e.g., timeliness, predictability, efficiency, robustness, fault tolerance or maintainability. The next step within ACOSAR will be to harmonize and close existing solutions, and gaps, at three levels:

1. **Simulation tool interface, in the computational cluster.** Which kind of communication (data and time) is needed such that the real-time requirements are fulfilled? Which kind of network model suits best to the networked computational cluster nodes?
2. **Communication protocol, within the entire real-time system.** Which kind of protocol and communication medium suits best to the communication needs between the physical components and the controllers, between the controllers themselves, and between the computational cluster, controllers and physical components?
3. **Hardware interface.** How can the IO modules be standardized in such a way that the real-time system can be built up modularly and heterogeneously?

All these questions will be followed up and answered in corresponding ACOSAR work packages. Further, these references will help us to get a common vocabulary and common definition of real-time systems.

4.7 Interoperability and Related Standards

4.7.1 Introduction

As ACOSAR enhances interoperability between real-time (RT) and non-real-time (non-RT) simulations, existing interoperability standards may give input to ACOSAR. Several related interoperability standards for integration of RT systems and non-RT systems are described in this chapter. Hence, communication architectures, communication abstraction as well as data access are important aspects that have to be analysed in ACOSAR. From the related interoperability standards, FMI has the closest relation to ACOSAR as it is a common standard to exchange simulation models and execute co-simulations via a standardized interface. However, FMI focuses on simulation tools and not on RT systems. In addition to related interoperability standards, the application and integration into an organization is relevant. Thus, ProSTEP iViP is described which represents an organization that supports the development of processes and best practices for applying, among others, such standards. More detailed information about related interoperability standards and literature is described in the following chapter.

4.7.2 Relevant State of the Art Items

4.7.2.1 ASAM GDI

Standard Name	GDI (Generic Device Interface)
Standardization Committee	ASAM ISO 20242
Chronology	Initial efforts – ? Draft – ? Release –Version 4.2.0, released in 2000 Release – current Version 4.5.0, released in 2011 Future developments, plans – ?
Consortium	Following information taken from [ASAM2016]: Total number of partners: 17 OEM level BMW AG, Porsche AG, Daimler AG, General Motors Company, Renault S.A., Volkswagen AG Tool vendors AVL List GmbH, dSPACE GmbH, FEV Automatisierungssysteme GmbH, imc Meßsysteme GmbH, Elektrobit Automotive GmbH, HORIBA Automotive Test Systems GmbH, MFP GmbH, M&K GmbH, National Instruments Corporation, rd electronic GmbH, Siemens AG
Major goals	The targets of ASAM GDI are the reduction of costs and efforts for running complex automation systems and related measurement devices. Plug-and-play and minimal integration effort for new systems integrated into an automation loop is mentioned as "ideal" target. A 3rd party device can be integrated in application software of a system integrator via driver software provided by the device vendor. With ASAM GDI the solution is independent of the operating system (integrator) and of the communication interfaces. For this a so-called "universally usable platform adapter" is provided.
Description	Description of standardization efforts What does the standard cover? ASAM GDI relies on a layer model and defines 4 layers [ASAM2016]:

	• Layer 4 - Coordinator: connect application programs and devices, allows re-routing • Layer 3 - Device Driver: provides access to the devices via clearly defined functions and states • Layer 2 - Platform Adapter: OS dependent integration for a device • Layer 1 - Transport Layer: e.g. via IPv4, USB, SoftSync, COM or LPT. What is its intention? The intention of the standard is to simplify the integration of devices for measurement and control applications into (the application software of) complex automation systems. It does an abstraction of operating systems as well as (needed) communication systems (busses, protocols). Is the consortium still active? Is it still developed? Yes Rough size of community? >20 companies involved/using standard >20 products supporting standard Name typical applications of the standard. It is used on chassis dynamometers, engine dynamometers, emission test benches or transmission test beds, for in car assembly lines, e.g. for fluid-filling stations, or for the integration of data loggers or measurement modules for supplier-independent device configuration. A list of applications is provided in [ASAM2016].
Available materials	Availability of standard, working sheets, models, etc. Download of standard (fee-based) is available via ASAM & ISO website
ACOSAR relevance	Connections to ACOSAR ACOSAR wants to provide a standardized interface for real-time systems to execute continuous real-time co-simulation whereas ASAM-GDI focuses on the integration of devices or measurement systems from the standpoint automation. Nevertheless, since these systems typically rely on real-time systems the ASAM GDI standard will be related to ACOSAR project. ASAM-GDI could be used at ACI SW integration level. Tool provider might use existing ASAM-GDI implementations for their products to extend these with ACI functionality. ACOSAR might re-use some of the concepts of GDI, e.g. layer concept.
Related WP	Deeper evaluation • of layer concept during WP2 • Of layer 4 during WP2, "master" • Of layer 2 & 3 during WP4 • Of layer 1 during WP5
Summarized by	Josef Zehetner (AVL)

4.7.2.2 ASAM XiL-MA

Standard Name	ASAM XiL-MA: Generic Simulator Interface for Simulation Model Access
Standardization Committee	ASAM
Chronology	ASAM HIL 1.0.0, 2009 ASAM XIL 2.0.0, 2013 ASAM XIL-MA 1.0.0 (republication of parts of ASAM XIL 2.0.0 in cooperation with ITEA project Modelisar, and as part of the FMI [Functional Mock-Up Interface] set of standards), 2014 ASAM XIL-MA 2.0.1, 2015
Consortium	Total number of partners: 18 <u>OEM level</u> Audi AG, BMW AG, Daimler AG <u>Supplier level</u> Robert Bosch GmbH, Continental Automotive GmbH <u>Tool vendors</u> AVL List GmbH, dSPACE GmbH, ETAS GmbH, Berner & Mattner Systemtechnik GmbH, D2T, HORIBA Automotive Test Systems GmbH, M&K Mess- und Kommunikationstechnik GmbH, MBtech Group GmbH & Co KGaA, National Instrument Corporation, RA-Consulting GmbH, Softing Automotive Electronics GmbH, TraceTronic GmbH, Vector Informatik GmbH
Major goals	With ASAM XIL-MA, test automation tools can control simulation tools and access simulation models. The API abstracts test automation tools from simulation tools and the underlying hardware (e.g. hardware-in the-loop systems). This allows for the integration of tools from different vendors. Additionally, different tools may exchange data with each other, which means that test cases can seamlessly run with different simulators. Users may freely choose testing products with ASAM XIL-MA support and integrate them seamlessly into their toolchain.
Description	ASAM XIL is a standardized API for the communication between test automation tools and test benches (model-in-the-loop (MIL), software-in-the-loop (SIL) and hardware-in-the-loop (HIL). ASAM XIL-MA has been extracted from ASAM XIL V2.0.0. Aside from some common functionality that is shared by the two standards, ASAM XIL-MA contains as its major component the specification of the MAPort (Model Access Port), an object-oriented API. This API provides read- and write-access to the model, means to capture simulation data, generate stimulation signals, define triggers, store data, manage model variables, handle errors, and methods for simulation control. ASAM XIL-MA is sometimes referred to as ´functional mock-up interface for (test automation) applications´. The ASAM XIL-MA standard is distributed in two packages: • Standard Package: The standard itself and the corresponding UML model.

	• Implementation Package: Templates and examples for multiple applications. The intention is to facilitate the development of standard-compliant tools. The standard is in on-going development.
Available materials	The Standard Package is free for the public. The Implementation Package is free only for members of ASAM; non-members may purchase this package. To obtain the Standard and/or Implementation Package, contact info@asam.net .
ACOSAR relevance	ASAM XIL-MA and ACI supplement FMI, and each other. • ASAM XIL-MA is an API that separates test automation tools from simulation tools and the underlying hardware. It supplements FMI as "FMI for applications". • ACI will supplement FMI as "FMI for real-time". It is highly desirable that co-simulations with ACU's (Advanced Co-simulation Units) can be subject to test automation via the model access port defined by ASAM XIL-MA. The compatibility should help with the validation of ACI during the testing stage of the ACOSAR project. A full compatibility of ACI with ASAM XIL-MA might even be considered a pre-requisite for a later adoption of the standard; it should probably be considered as a requirement for ACI. The model access port (defined by ASAM XIL-MA) will likely be realized through a wrapper around an ACU. The wrapper would then map calls to the model access port onto calls to the ACI. Therefore, it needs to be ensured in the ACOSAR project that the ACI standard allows for such a mapping to be possible, i.e. ACI should allow the wrapper read- and write-access to the model, to set up capturing of simulation data and generation of stimulation signals, definition of triggers, storage of data, management of model variables, error handling and methods for simulation control. Where applicable, parts of ASAM XIL-MA may even be reused in ACOSAR in order to achieve full compatibility between the two standards.
Related WP	WP1 (Open System Architecture Requirements) WP4 (RT-System Interface) Requirements should be ASAM XIL-MA compatible;ASAM XIL-MA may be useful for automated testing of the ACI.
Summarized by	Oliver Kotte (Bosch), based on information available at http://www.asam.net

4.7.2.3 ASAM MCD-1 XCP

Standard Name	ASAM MCD-1 XCP (Universal Measurement and Calibration Protocol)
Standardization Committee	ASAM (Association for Standardization of Automation and Measuring Systems)
Chronology	XCP 1.0 – 2003

	• Specification of the base standard features measurement & calibration, synchronous stimulation, page switching, reprogramming. • Specification of the transport layers for CAN, Ethernet (UDP and TCP/IP), SPI and USB. XCP 1.1 – 2008 • Addition of the transport layer for FlexRay. XCP 1.2 – 2013 • Addition of the A2L-IF_DATA description for calculation of the estimated ECU resource consumptions. XCP 1.3 – 2015 • Addition of the base standard features ECU States, Bypassing Error Handling and Time Correlation
Consortium	Total number of partners: 8 <u>OEM level</u> Daimler AG <u>Supplier level</u> Robert Bosch GmbH, Accurate Technologies Inc. and RA Consulting GmbH <u>Tool vendors</u> Continental Automotive GmbH, ETAS GmbH, Vector Informatik GmbH, dSPACE GmbH and CSM GmbH
Major goals	<u>Main Focus</u> The standard interoperates between calibration systems and ECUs using XCP. It enables an OEM to : a. Calibrate a vehicle comprising of multiple ECUs from different suppliers. b. Use several calibration systems for the identical ECU provided by the supplier The standard was designed with two major goals : 1. Minimize the high requirements on ECU resources such as CPU load, RAM consumption and flash memory, for the XCP slave. 2. Achieve a maximal data transmission rate over the communication link and reduce the impact on bus communication. In order to achieve the above goals the XCP was designed. The XCP fulfils the following objectives: • Minimal resource usage in the ECU • Efficient communication • Simple Slave implementation • Plug-and-play configuration with just a small number of parameters • Scalability: XCP is also very scalable in its implementation, it is not necessary to implement every command described by the standard • Transferability: XCP can be implemented on major bus systems and is open for future bus systems
Description	This standard is a bus-independent, master-slave communication protocol which connects ECUs with calibration systems and accesses parameters and measurement variables using memory addresses. It includes a base standard, which describes memory-oriented protocol services with indirect dependencies on specific bus systems. Several

	standards in the close vicinity contain the transport layer definitions for CAN, FlexRay, Ethernet (UDP/IP and TCP/IP), serial links (SPI and SCI) and USB. The A2L is the accepted file format, standardised through ASAM MCD-2 MC standard. Using the A2L file, one can access a specific parameter or variable, without hardcoding the access in the ECU application software. Benefit: Several calibration and measurement tasks can be created for different configurations of the calibration system, reusing (without recompiling and reprogramming) the ECU application code. Applications: • Calibration of ECU parameters • Measurement of ECU variables • Stimulation of ECU variables • ECU programming The Consortium is active. All major calibration tool suppliers support XCP, typically for all bus systems supported by the standard. Relationship with other existing standards ASAM XCP closely relates to ASAM MCD-2-MC (also known as ASAP2), which is a description format for internal ECU variables. The ASAP2 Meta Language describes the ECU specific protocol parameters within interface descriptions. XCP on FlexRay requires the information of the FlexRay network cluster. For more information on this standard please refer to the link. XCP is closely relates the XCP BSW module specification of AUTOSAR 4.0, which implements the protocol stack and defines an RTE interface for sampling XCP measurement data.
Available materials	The technical contents of the standard can be found in https://wiki.asam.net/display/STANDARDS/ASAM+MCD-1+XCP A downloadable version of the standard can be found in http://www.asam.net/nc/home/standards/standard-detail.html?tx_rbwbmasamstandards_pi1[showUid]=3144
ACOSAR relevance	1. Standardise the ACI parameter access and calibration to allow runtime stimulation of parameters. 2. This standard is applicable at System level (during system tests to verify a right ACI behaviour for the specific system) 3. This standard uses XCP (an industry standard data exchange method) – uses Ethernet and USB protocols to communicate with PC. Refer to XCP on Ethernet V1.3.0, XCP on USB V1.3.0
Related WP	This standard is relevant for the work package 5. The standard talks about several possibilities (Ethernet, USB, CAN, FlexRay) of communication at physical layer with PC and real-time platforms using XCP data-exchange method. At this point, we need to analyse how we can re-use this standard into our ACI communication with real-time systems and PC.
Summarized by	Natarajan Nagarajan (ETAS)

4.7.2.4 HLA

Standard Name	HLA: High Level Architecture

Standardization Committee	IEEE (Institute of Electrical and Electronics Engineers)
Chronology	Initial efforts, 10/93 - 01/95 Draft, 03/95 – 08/96 Release of baseline HLA definition (HLA 1.0), 09/96 Release of first complete version (HLA 1.3), 02/98 First release of IEEE 1516, 09/00 Latest release of IEEE 1516, 2010
Consortium	Total number of partners: unknown Government: US Department of Defense (Defense Modelling and Simulation Office, renamed Modelling and Simulation Coordination Office) Architecture Management Group Industry: Representatives of major US DoD simulation programs Universities, Research centers: DARPA, Georgia Institute of Technology
Major goals	Definition of a simulator infrastructure to support interoperability and reuse of defense simulations.
Description	HLA is a general purpose architecture to integrate multiple (>100) of mostly independent simulators or real systems into one distributed simulation system or scenario (system of systems approach). In order to achieve great interoperability and reusability, HLA is a rather complex construct, to which a more formal and detailed definition can be found in the material listed below [IEE2000] [DAH1997] [DAH1999]. A brief structural overview is given in figure 11 (adapted from [STR2006]). Down to its core HLA is based on the following key ideas: • **Peer-to-peer**: there is no master-system or central means of control. All data and functionality is stored decentralized and shared upon request. • **Hierarchical clustering**: multiple systems (called federates in HLA) are grouped together and form a federation, which provides a superset of the information from the contained federates to other federations. • **Object oriented**: much like in object oriented programming, HLA object models (of different types) define how federates or federations have to represent themselves to one another (expose internal data, attributes, associations and interactions of the specific federate or federation) • **Template based**: object models are based on a so called object model template which specifies the tables that need to be documented (Attribute/Parameter Table, Interaction Class Structure Table etc.). The object model template defines the “language” of HLA. • **Strict set of rules**: HLA rules specify the properties and behaviour of a federation and its federates. These rules represent the foundation of HLA. There are 5 basic rules for federates and federations each [IEE2000].

- **Standardized interface:** definition of services which federates can use and which they have to provide. These services are described by the HLA Interface Specification and is divided into 6 subcategories (e.g. time management or data distribution management)
- **Abstract communication layer:** Federates send and receive data or events through predefined services via a common runtime infrastructure (RTI) [FUJ1998]. To implement this bi-directional communication scheme, the interface uses "ambassadors" as intermediaries between federate and RTI. From a programming point of view these ambassadors are objects and communication is done by calling methods (=services) of these objects.
- **Subscription system:** because in HLA there is no central element, where all data and information is stored, HLA uses a subscription and broadcasting system called declaration management services. Federates declare which data, attributes, information or methods they provide and which they are interested in.

There are implementations of the standard in Java and C++, plus Ada and FORTRAN for the older HLA v.1.3

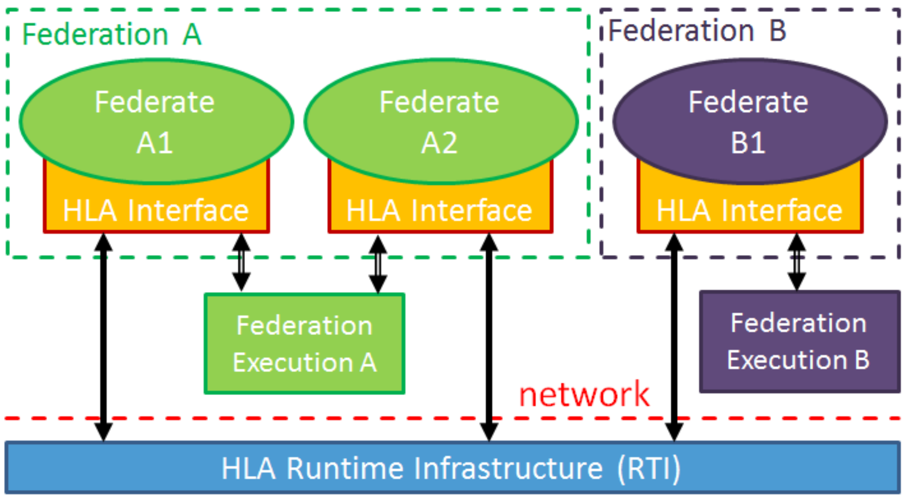


Figure 1: Functional view of a distributed simulation under HLA

As it was the major design goal, a typical application of the standard is the integration of a wide variety of different simulators or real systems (e.g. ships, aircraft, ground vehicles etc.) to a complex scenario for:

- tactical simulations (e.g. war-games)
- training on platform- and command-level

system testing and evaluation

In general HLA is applicable when multiple autonomous entities interact and/or exchange information but do not depend on each other on a basic functional level (multi-agent systems). Because of the complexity the standard incorporates it only seems efficient for larger amounts of these entities. Minor applications in the civilian sector are factory planning and smart transportation systems (e.g. airport or airspace control simulations).

Available materials	http://msco.mil/hla.html STR2006

	IEE2000 DAH1997 DAH1999 FUJ1998
ACOSAR relevance	Although HLA targets a completely different type of distributed simulation (multi-agent systems) as ACOSAR (strongly coupled subsystems), there are some elements which can be adopted or used as guideline: • Standard interface between systems, based on a known template with a predefined set of methods or services. • Encapsulate complex systems in simpler object models, which just expose and demand necessary information. • Interaction between object models based on strict rules • Abstract communication layer independent from the actual means of data transfer (physical layer e.g. CAN)
Related WP	WP6 (Advanced Co-Simulation Interface) Regarding the relevant points listed above HLA can provide a guideline during the design of the ACI-Element. WP2 (RT-System Integration Methodology) When discussing system integration methodology, HLA can serve as one possible way to do this. Especially when it comes to the specification and the interfaces of subsystems (T2.2 & T2.3) of which HLA features a pretty advanced method (template based & object oriented). . WP3 (Simulation Tool Interface) and WP4 (RT-System Interface): T3.1, T4.1 & T4.2 (slightly related) WP5 (Communication System Protocol) During implementation of the ACI element one of the open source implementations of HLA (e.g. Open HLA) might provide some inspiration.
Summarized by	Timo Haid (Porsche AG)

4.7.2.5 FMI

Standard Name	FMI: Functional Mock-up Interface for Model Exchange and Co-Simulation
Standardization Committee	Modelica Association www.modelica.org
Chronology	Initial effort, 2008 (within MODELISAR project) FMI for Model Exchange Version1.0, January 2010 FMI for Co-Simulation Version 2.0, October 2010 FMI for Model Exchange and Co-Simulation, July 2014 Development follows the FMI development process rules [FMI2015]. As soon as quality gates are fulfilled, test implementations are available and the FMI Steering Committee agrees, new versions will be published.
Consortium	FMI is one of currently 4 Modelica Association Projects [MOD2016]. Total number of partners: 24 Members of the Steering Committee: BOSCH, Daimler, Dassault Systèmes, dSPACE, IFP EN, ITI, LMS, Maplesoft, Modelon, QTronic, Siemens, SIMPACK

	Members of the Advisory Committee: Altair, Armines, AVL, DLR, ETAS, Fraunhofer (IIS/EAS First, SCAI), IBM, ITK Engineering AG, Open Source Modelica Consortium, Synopsys, TWT, University of Halle [FMD2016]
Major goals	Functional Mock-up Interface (FMI) is a tool independent standard to support both model exchange and co-simulation of dynamic models using a combination of xml-files and compiled C-code [FMI2016].
Description	FMI defines an API in programming language C and a XML format description for the interface data that allows for exchange of simulation models and co-simulation. A component which implements FMI is called FMU. It is a zipped file that contains the XML model description, the implementation in binary and/or source code representation and additional data. The FMI specification defines the mathematical representations and the allowed calling sequence of the API functions for each stage of the solution process. The consortium is active; it currently develops new features in several working groups and works on a FMI 1.0 maintenance release. At the moment 82 tool vendors support FMI [FMZ2016]. FMI is for example applied for model exchange between suppliers and OEMs for system simulation and optimization. In that way each partner is able to use the best fitting tool or modelling approach for their component and simulation task.
Available materials	FMI specifications are publically available free of charge on: www.fmi-standard.org
ACOSAR relevance	FMI for Co-Simulation is a widely accepted standard. It provides functionalities which allow for simple and sophisticated co-simulation master algorithms (for example variable communication step-size, iterative algorithms that repeat communication steps, directional derivatives, extrapolation techniques). It does not specify any network protocol for distributed co-simulation architectures, but can be used in such environments if the network communication is implemented by other components. The standard which is developed in ACOSAR could fill the gap of the missing network protocol. It should allow to transport as much as possible of the data which are necessary for certain FMI features (for example data that allow for extrapolation of inputs, directional derivatives). FMI defines certain causalities (inputs, outputs, parameters, local variables) and variabilities of that data. Additionally, it defines an expansible set of meta data for that signals. These definitions are the results of requirements from a manifold of applications and discussions of groups of experienced experts in the field of modelling and simulation. ACOSAR should adopt these definitions and the way how they are represented as much possible and extend it if necessary.
Related WP	WP1 (Open System Architecture Requirements): FMI features should be considered for requirements specification WP2 (RT-System Integration Methodology): The method should consider compatibility with FMI WP3 (Simulation Tool Interface): ACI should be compatible with FMI WP4 (RT-System Interface): ACI should be compatible with FMI

Summarized by Torsten Blochwitz (ITI)

4.7.2.6 OSI-Model

Standard Name
Open System Interconnection (OSI) Model

Standardization Committee
International Organization for Standardization (ISO)

Chronology
Release: 1984 – now *

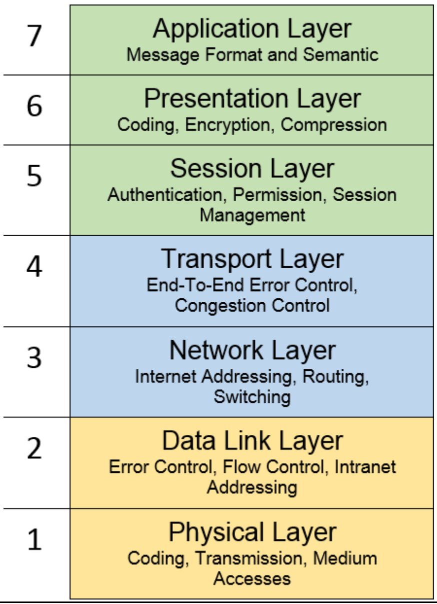
* The OSI project is made of a series of ISO standards, which are maintained and updated separately. The main standard, which describes the abstract model (OSI model), is defined in ISO/IEC 7498-1:1994.

Consortium
Hubert Zimmermann (ISO/IEC 7498:1984)

Major goals
To define a unifying standard for the architecture of networking systems and the interoperability of diverse communication systems with standard protocols.

Description
The OSI project is made of a series of ISO standards, which are maintained and updated separately. It defines a unifying standard for the architecture of networking systems and the interoperability of diverse communication systems with standard protocols.The core of OSI project is a general architecture of message-based communication protocols. It is known as OSI model and enables message exchange in an abstract and easy way. The complete networking can be considered as a black-box for the application.

In practice, almost all communication protocols and all hardware for communication (including the operating systems) are designed with respect to the OSI architecture. However, there are often slight modifications for the in-station communication between the protocols, and sometimes the architecture is not fully implemented.



As shown in the figure, the OSI architecture is made of 7 layers (layer 1 - 7). The layer N is built on the layer (N-1) and provides more features and functionalities (e.g. error control, congestion control, multiple access of shard medium) for the communication than the layer (N-1). In general, layer N simplifies the communication procedure for the application / protocols built above it, compared with communication procedure on layer (N-1). In addition, it also provides better compatibility than layer (N-1).

Available materials

- List of public available ISO standards (including OSI) http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

	• Document for ISO/IEC 7498-1:1994 * http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip
ACOSAR relevance	OSI defines an architecture, which describes how the message-based communication is achieved in various environment. In the practice, a similar architecture, which is known as TCP/IP model, will be applied instead of OSI model in most cases. In the ACOSAR project, Support of the abstract architecture (ACI or TCP/IP) and the related protocols is a most important requirements for all applied hardware (including simulation servers and coupled real-time hardware as a part of the model). In an advanced co-simulation system, different kinds of hardware are connected as a networking system. They must exchange messages for further cooperation (synchronization, exchange of variable, etc.). The abstract architecture and the related protocols provide a general message-based communication channel between these hardware. ACI will be built upon the channel, and it defines the semantic of the messages to exchange for co-simulation. In other words, ACI will define the semantic of the message to exchange (e.g. the data format of variable to exchange). The message itself, as sequence of binary data, will be exchanged through the channel in a pre-defined way. The OSI architecture provides different ways for an application to connect to the architecture. The application can be built on almost any layer in the OSI architecture in theory. As mentioned earlier, connecting at a higher layer provides better compatibility of the hardware, since all protocols with respect to the OSI architecture (from the lowest layer to the layer of integration) are supported automatically, or can be supported in an easy way. Connecting at a lower layer avoids support of unnecessary features and functionalities (e.g. support of certain hardware environment, detection of transmission error / packet loss and automatic re-transmission, internet routing, and addressing), and further enables hardware-depended optimization. It leads to a better performance (shorter run-trip time, better efficiency, and lower cost of hardware, etc.), but some necessary features might have to be defined and implemented by ACI. The ACI is designed to provide a universal co-simulation interface, thus the support of different environments will be considered. ACI will also support various existing communication protocols to maximize the compatibility. Therefore, different coupling points to the OSI architecture will be defined. Each coupling point provides different features and functionalities. In addition, each hardware might also define the minimal set of features and functionalities which must be supported by the communication channel and / or ACI. Based on the information the hardware is connected physically and the ACI is configured.
Related WP	The OSI standard is mainly related to WP5. As mentioned above, the abstract architecture and the related protocols provides a general messaged-based communication channel, which is the base to define the semantic. However, we also have to provide a global solution for addressing, which is related to WP3, WP4 and WP6.
Summarized by	Desheng Fu (LUH)

4.7.2.7 OSLC

Standard Name	OSLC: Open Services for Lifecycle Collaboration
Standardization Committee	De-facto standard OSLC is an open standard, i.e. everybody can participate. Nevertheless, there are technical committees compliant to the OASIS specification.
Chronology	2008 start of OSLC initiative 2013 member of OASIS
Consortium	As OSLC is an open standard there is no fixed number of participants. However, a lot of tool vendors are members of OSLC. The actual participants can be found at: http://open-services.net/organizations/.
Major goals	OSLC is an open community building practical specifications for integrating software. The main goal is to provide a standardized interface for development tools in order to improve interoperability and traceability between different and heterogeneous tools.
Description	OSLC consists mainly of two types of specifications: OSLC Core specification and several OSLC domain specifications. The OSLC Core specification defines common resource types and properties whereas the domain specifications define what resources and services are required as well as resource types, properties and relations in that domain. Domain specifications include: • Requirements Engineering • Architecture Management • Asset Management • Automation • Change Management • Configuration Management • Quality Management • Performance Monitoring • Reconciliation • Estimation and Measurement All current specifications can be found at http://open-services.net/specifications/ Hence, the standard covers all main development lifecycle artefacts. The intention is to improve interoperability between different lifecycle tools and enabling an automated traceability between development artefacts across different tools. OSLC is based on standard web technologies such as HTTP, RDF/XML, URI and RESTful Web Service. Further, it uses the LinkedData concept to enforce traceability. The idea is to implement an OSLC interface for tools that want to communicate. Having for example two tools, one provides data and one consumes data, both tools have to implement the OSLC interface. The tool providing the data has to offer an OSLC provider and the tool using the data has to have an OSLC consumer. Typical applications are for example:

	• Exchange requirements between Requirements Management tools • Link Requirements and Test Cases across different tools (Requirements Management tool and Test Management tool) • Show linked requirements as a preview in an Architecture/Modelling tool (Requirements from a different Requirements Management tool) Main advantages of OSLC are: • no common database or tool is needed during development but data can be stored in the intended tool • traceability needs not to be managed manually across different tools • no proprietary but standardized interfaces facilitate data exchange, i.e. data exchange with several tools providing the OSLC interface is possible. As the consortium is still active and a lot of tool vendors have the interface, or parts of the interface, implemented, OSLC is constantly improved.
Available materials	All related information regarding the standard can be found at http://open-services.net/ [OSL2016].
ACOSAR relevance	OSLC is a standard format for exchanging lifecycle data of development tools whereas ACOSAR's goal is to standardize an interface for real-time simulations. Hence, the field of interest and challenges are different. Although both, OSLC and ACOSAR, standardize interfaces, OSLC does not support features, such as real-time requirements, needed in ACOSAR. Further, OSLC's challenge is to exchange data of different domains with different properties whereas ACOSAR focuses on simulation. However, the extension mechanisms (how to extend standardized properties that are exchanged) of OSLC may be interesting for ACOSAR as well. In other aspects there is no close relation to ACOSAR.
Related WP	WP2 (RT-System Integration Methodology) WP3(Simulation Tool Interface)
Summarized by	Nadja Marko (Virtual Vehicle)

4.7.2.8 Co-simulation, coupled simulation or coupling of simulators?

Publication name	Co-Simulation, gekoppelte Simulation oder Simulatorkopplung? [Co-simulation, coupled simulation or coupling of simulators?]
Authors	Marcus Geimer, Thomas Krüger and Peter Linsel
Year	2006
Reference	[GKL2006]
Summary	This work deals with the definition of co-simulation based on the number of involved integrators and simulation tools. The delimitation of the term co-simulation to other related terms like simulator coupling, coupled simulation, classical simulation etc. is discussed on model- as well as on program layer. Using the coupling on model-layer, all models have to be merged to one overall model which is typically solved using one integrator. Coupling approaches on program-layer use their specific simulation tool including their solvers (integrators). In this case the coupling data exchange

	between the involved simulation models is realized via one common interface. As a conclusion the co-simulation approach represents the most flexible and modular approach because the number of integrators and simulation tools is at least greater than 1 which results in distributed simulation and modelling.
Project	-
ACOSAR relevance	This work deals with the definition of co-simulation which is important to start with the same overall picture in defining requirements for the ACI. The understanding of the co-simulation problematic is essential to point out the critical aspects (such as internal-loops, extrapolation and scheduling) occurring in a distributed simulation and modelling environment which in turn is important for the ACI specification.
Relevant WP	WP1 (for common understanding and description of perimeter) WP3 (Simulation Tool Interface)
Summarized by	Georg Stettinger, Nadja Marko (Virtual Vehicle)

4.7.2.9 ProSTEP iViP

Project Name	ProSTEP iViP (with focus on project SmartSE: Smart Systems Engineering)
Project period	ProSTEP: 1993 – present; ProSTEP iViP: 2003 – present; ProSTEP iViP project Smart Systems Engineering (SmartSE): 2012-2018
Consortium	ProSTEP iViP Total number of partners: 169 Project budget: 1 m. € (Technical Program) Industry (User): 42 % IT-Companies: 38 % Research and other bodies: 20 % ProSTEP iViP project Smart Systems Engineering OEMs: Airbus Deutschland GmbH, Audi AG, BMW AG, Daimler AG, Ford Werke GmbH, Volkswagen AG Supplier AVL LIST GmbH, Continental Automotive GmbH, Robert Bosch GmbH, Siemens AG, Siemens PLM, ZF Friedrichshafen AG Tool vendors & Consultancies Accenture PLM GmbH, Dassault Systèmes, dSPACE GmbH, :em engineering methods AG, ETAS GmbH, ITI GmbH, Parametric Technology GmbH, PROSTEP AG, Unity AG Academic partners Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik (IPK), TU Darmstadt (DiK), TU Kaiserslautern (VPE)
Major project goals	The mission of the international ProSTEP iViP Association is to solve problems and develop standards for product data management and virtual product creation. ProSTEP iViP takes the interests of manufacturers, suppliers and IT vendors into account, and cooperates with research and

	science institutes. It provides its members with more efficient processes, methods and systems in order to establish a long-term competitive advantage. The scope of ProSTEP iViP is huge. For an overview, see http://www.prostep.org/. For ACOSAR, the user-driven ProSTEP iViP project "Smart Systems Engineering" (SmartSE) is probably most relevant. The SmartSE project focuses on best practices for the exchange of behavioural models across disciplines and enterprises. The Functional Mockup Interface (FMI) has been chosen as the standard that is used for the exchange. Current focus is on developing additional test scenarios for the use of FMI.
Standardization	The SmartSE project has chosen FMI as the standard that is used for the exchange of behavioural models across disciplines and enterprises. The project develops best practices for the process of exchanging behavioural models between enterprises.
Major project outcomes	The standardization efforts of ProSTEP iViP have been ongoing since more than a decade. The major project outcome of the SmartSE project so far are recommendations for the exchange of behaviour models between two companies.
Available materials	The results are documented in a ProSTEP iViP Recommendation for "Behavior Model Exchange". The document describes several use cases, a reference process for the exchange and two accompanying templates. Members may download this document free of charge. Non-Members need to pay a fee (approx. 50 €) to gain access.
ACOSAR relevance	For some use cases in ACOSAR, models are exchanged between enterprises. In order to facilitate these exchanges, the processes developed in the SmartSE project should be applied.
Related WP	WP 2 (RT-System Integration Methodology)
Summarized by	Oliver Kotte (Bosch)

4.7.2.10 DIS

Standard Name	DIS: Distributed Interactive Simulation
Standardization Committee	IEEE
Chronology	Initial efforts, 1989 Release of IEEE-1278, 1993 Current version: IEEE-1278.1, 2012 NATO standard STANAG 4482, 1995-2010
Consortium	Standardization via SISO (Simulation Interoperability Standards Organization) Consortium not known, involved parties are: DoD (US Department of Defense), USAF Distributed Mission Operations Center (DMOC) and DARPA (Defense Advanced Research Project Agency)
Major goals	Standard for the communication between distributed simulators while no central computer controls the entire simulation.

Description	The development of DIS was originally promoted by the DARPA in order to define a standard for linking the interactive, free-play activities of people in military exercises. However, DIS supports both offline simulators as well as real-time (including "human-in-the-loop") components. DIS enables the data exchange between simulators without a central coordinating computer, which is usually missing in warfare simulations. DIS defines 72 types of PDUs (Protocol Data Units) which represent the messages to be transmitted. The 13 groups of PDUs are: • Entity information/interaction • Warfare • Logistics • Simulation management • Distributed emission regeneration family • Radio communications family • Entity management family • Minefield family • Synthetic environment family • Simulation management with reliability family • Live entity family • Non-real time family • Information Operations family The DIS standard requires PDUs to be sent using the UDP/IP broadcast format, where all PDUs are sent to all players. This means that each connected entity receives all transmitted messages. However, filters (e.g. for PDU type, entity type, or frequency) may be applied. Due to the usage of broadcast, network communication can get slow when using a large number of simulators. In 2010, DIS was cancelled as NATO standard in favour of HLA.
Available materials	[DIS2012]
ACOSAR relevance	DIS defines a common communication protocol for both real-time as well as offline simulations. Main differences to ACOSAR are that DIS uses as communication channel only UDP and it focuses on military applications. The latter influences the data/properties that are defined in DIS. However, the communication protocol to combine online and offline simulation can be used as input for ACOSAR. The most interesting PDU groups therefore are non-RT protocol, simulation management, live entity information/interaction protocol as well as the DIS extension method. Note: Definition of real-time simulation can be different to ACOSAR
Related WP	WP5 (Communication System Protocol)
Summarized by	Stefan Thonhofer, Nadja Marko (Virtual Vehicle)

4.7.2.11 DSS

Standard Name	DSS: Distributed Simulation Systems
Standardization Committee	OMG (Object Management Group)
Chronology	Release – June 2000 Release of version 2.0 – November 2002

Consortium	Consortium not known
Major goals	Specify a mapping of HLA to CORBA/IDL
Description	The main objective of DSS was to use CORBA (Common Object Request Broker Architecture) within HLA (High Level Architecture). CORBA is an OMG standard for facilitating the communication between systems on different platforms (or at least written in different programming languages). Within CORBA, an IDL (Interface Definition Language) and a programming API are defined that enable client/server communication. Throughout the 1990s, CORBA became a frequently used standard in industrial applications. DSS defines IDL interfaces for all HLA routines (see HLA standard definition for further information). The SISO (Simulation Interoperability Standards Organization) never incorporated DSS and hence, it never became widely spread [TOL2002].
Available materials	[DSS2002]
ACOSAR relevance	DSS enhances interoperability by specifying uniform interface specifications. DSS enables simulations implemented in CORBA IDL to be operated within HLA. Hence, HLA can be important for ACOSAR but not DSS itself as it is a platform specific (CORBA) integration mechanism.
Related WP	-
Summarized by	Stefan Thonhofer, Nadja Marko (Virtual Vehicle)

4.7.2.12 SSP

Standard Name	SSP: System Structure and Parametrization
Standardization Committee	Modelica Association www.modelica.org
Chronology	The Project SSP was founded on initiative of BMW, Bosch and ZF in 2014
Consortium	SSP is one of currently 4 Modelica Association Projects [MOD2016]. Consortium (partially): AVL, Bosch, Dassault Systemés, dSPACE, ITI, ZF A public list is not available.
Major goals	• Define a standardized format for the connection structure for a network of components • Define a standardized way to store and apply parameters to these components. [KOK2015]
Description	• The developed standard / APIs should be usable in all stages of development process (architecture definition, integration, simulation, test in MiL, SiL, HiL). • The work in this project shall be coordinated with other standards and organizations (FMI, ASAM, OMG). [KOK2015]

Available materials	The current drafts are not publically available.
ACOSAR relevance	The goal of this project is to provide a standardized format to represent a connected set of components. These components can for example be simulation models (FMUs, or in other representations), software components (for example controller code) or co-simulation entities. In ACOSAR we should check whether this format is suited to represent the structure of a co-simulation problem. The SSP specification is still in development. If ACOSAR specific extensions are necessary, we could try to influence the specification process. A close connection between ACOSAR should be given since some of the ACOSAR partners (AVL, Bosch, dSPACE, ITI) are SSP project members too.
Related WP	WP 2: RT-System Integration Methodology
Summarized by	Torsten Blochwitz (ITI)

4.7.3 Summary and Conclusion

The ongoing efforts to standardize interfaces of systems show that there is a need to become more independent and flexible in development. With interoperability standards the complex product development should become more flexible with respect to new opportunities for cooperation between business partners. However, in ACOSAR the standardization of an ACI is focused which will enable co-simulation of RT and non-RT systems. Related interoperability standards, which are described in the previous section, focus on RT system interfaces, on simulation tool interfaces or can only be applied in special domains. There is no generic standard for co-simulation of RT systems and non-RT systems.

ASAM GDI and ASAM MCD-1 XCP provide standardized interfaces for RT devices to be integrated into application programs. However, they do not concentrate on simulation but on other development activities. HLA and DIS describe architectures for distributed simulation including RT systems but they are specialized for multi-agent simulations and focus on military applications. DSS is a CORBA implementation for HLA. Hence, these standards do not fulfill requirements necessary for strongly coupled RT simulations. ASAM XiL MA (FMI for applications) and FMI for model exchange and co-simulation are closely related to ACOSAR. However, at this moment the FMI standard does not guarantee the compatibility of an FMU with RT systems. OSLC provides standardized interfaces for lifecycle development tools and is therefore neither applicable for simulations nor for RT systems. The OSI model is a generic architecture for communication over a network. It is not specialized for a specific application but is a general purpose architecture that can be a basis for ACOSAR. Finally, ProStep iViP proposes processes and best practices for applying FMI for model-exchange and co-simulation (SmartSE project).

Though all standards have connection points to ACOSAR, none of them concern a generic interface for executing RT co-simulations. There are standard interfaces for RT devices, for non-RT co-simulation or restricted RT co-simulation (no strong coupling, mainly applicable in defence). Nevertheless, some concepts may be reused from existing standards. The abstraction of communication that is supported by ASAM GDI, ASAM MCD-1 XCP and HLA can be useful for ACOSAR since ACOSAR also requires independence of communication medium. Furthermore, the architecture of HLA and DIS may be relevant since these architectures support concepts such as hierarchical clustering of simulation models or independence of a master system. ACOSAR aims in developing an ACI that supports these features as it should be as flexible as possible. However, also the OSI model may be a reference for the ACOSAR communication architecture.

ASAM XiL MA and FMI standardize an interface for exchanging and accessing models and their execution. Methods from these standards can be reused for simulation tool interfaces. Moreover, the ACI has to be compliant to FMI as it is an established standard for co-simulation. ACI can be added to FMI as "FMI for RT co-simulation". Further, ACI can also extend ASAM GDI to use ASAM GDI compatible devices for co-simulation.

4.8 Requirements, Modelling, Design, and Specification for Integration

4.8.1 Introduction

Since UML laid the foundation for model based software development in the 1990s, model based development is on the rise in many different industry sectors. While complexity was rather imminent in the software domain, others realized the need for modelling with the advent of electric and electronic systems in their domain. For the modelling of complex systems, SysML was created on top of UML as a general purpose modelling language. The AutomationML targets production systems within the Industry 4.0 trend. Finally, MARTE is a UML2 profile intended for modelling of real-time and embedded systems.

All these languages have in common that they are not only used for modelling and specification of software, but rather entire systems. They support requirements engineering, specification, analysis, design, as well as verification and validation steps within a process or product life cycle.

For ACOSAR, these modelling languages are relevant for different reasons. First of all, ACOSAR targets the interface to RT systems in general. Many system problems result from inadequately defined interfaces, and problems often surface during integration and test. That means that they are discovered rather late in the development process. Considering the ACI in the development process, the integration of RT systems requires clear interface specifications – however, they must be established as early as possible to support not only HiL tests, but rather specification, MiL and SiL test phases. Recurring use of interface information also leads to a continuous refinement. Thus, the use of modelling languages for interface specification is favourable.

Second, as the ACI is basically a software reference specification, the use of a semi-formal notation helps to achieve a clear and comprehensible specification. This enforces a fast and efficient implementation.

4.8.2 Relevant State of the Art Items

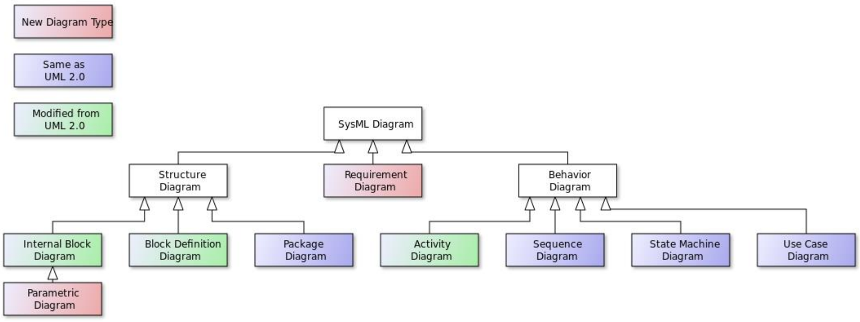
4.8.2.1 Unified Modelling Language (UML)

Standard Name	Unified Modelling Language (UML)
Standardization Committee	Standardized by OMG
Chronology	Initial efforts – 1990 First standard in 1997 by OMG Current version: 2.5, June 2015
Consortium	No classic consortium
Major goals	Development of a common, graphical language for specification and documentation of software.
Description	UML consists of several diagram types that support the specification and documentation of software. UML diagrams are divided into two general types of diagrams: structural diagrams and behavioural diagrams. Structural diagrams focus on representing the static structure/architecture of software whereas behavioural diagrams are aimed at presenting the behaviour of software. UML structural diagrams are: • Package Diagram • Component Diagram • Class Diagram • Deployment Diagram • Composite Structure Diagram • Object Diagram • Profile Diagram UML behavioural diagrams are:

	• Use Case Diagram • Sequence Diagram • Activity Diagram • Timing Diagram • State Machine Diagram • Interaction Overview Diagram • Communication Diagram The language concepts of UML are described with UML language elements itself. The conceptual description consists of 4 levels, M0 to M3. Higher levels (M2-M3) are considered important for tool providers, whereas modelling activities usually are conducted on levels M0-M1. However, the UML standard was adopted by many tool vendors, implementations still vary due to different code generation methodologies, hardware targets and target applications. The consortium is still active, activity spread across many groups.
Available materials	[UML2015]
ACOSAR relevance	The relevance for the ACOSAR project is twofold. First, UML technology might be interesting for software development in context of development of the ACI, enforcing traceability and modularity, perhaps also code generation features. Second, it might be relevant for the configuration of simulation models and simulation scenarios. This is also in close connection with the SysML standard.
Related WP	WP1, WP2, WP7
Summarized by	Martin Krammer, Nadja Marko (Virtual Vehicle)

4.8.2.2 SysML

Standard Name	SysML – Systems modelling language
Standardization Committee	OMG
Chronology	Initial efforts – 2001 OMG WG Release – 2006 Final Adopted Specification 09/2007 OMG SysML 1.0 released 11/2008 SysML 1.1 06/2015 SysML 1.4 (active development)
Consortium	SysML partners and contributors can be found at http://sysml.org/sysml-partners/.
Major goals	Development of a common, graphical notation and communication language for Systems Engineering.
Description	SysML is a modelling language based on the UML 2 specification. SysML describes diagrams and modelling elements like UML but with focus on systems development. As a result SysML covers diagrams that are the

	same like in UML, new diagrams and modified UML diagrams (see Figure 6).  Figure 6: SysML diagrams Consortium is active (with changed partners), also different working groups are active in related fields, e.g. functional safety for the description of System Safety Modelling Language Sys(S)ML Typical application of the standard are: Requirements engineering, System architecture descriptions, Functional Safety (specification, architecture, system design, safety analyses, etc.), behavioural descriptions.
Available materials	[SML2015]
ACOSAR relevanc	SysML is a language candidate for the description of overall co-simulation/experimentation scenarios, as it is capable for the description of RT systems and complex system architectures. Related work for the field of co-simulation with ICOS from VIRTUAL VEHICLE exists. As ACOSAR considers additional safety functions for halting or cancelling simulations/experimentations, the capabilities of SysML may also be useful in this field. The use cases in ACOSAR may also benefit from use case diagrams.
Related WP	WP2, WP7
Summarized by	Martin Krammer, Nadja Marko (Virtual Vehicle)

4.8.2.3 AutomationML

Standard Name	AutomationML
Standardization Committee	ABB, Daimler, Siemens, VW, Thyssen-Krupp, Mitsubishi Electric, Fraunhofer (IOSB, IPA), KIT, RWTH (MMI), TU Wien
Chronology	Founded 2006 registered association since 2009 Current developments can be tracked on the web page at https://www.automationml.org
Consortium	Total number of partners: 34 Complete list of partners can be found at https://www.automationml.org/o.red.c/mitglieder.html.
Major goals	AutomationML is the most comprehensive data format of plant engineering. It is already used in the field and is also available in several products. Main purposes are:

	• standardise data exchange in the engineering and manufacturing process • support the data exchange in a heterogeneous engineering tools landscape (interconnect engineering tools in their different disciplines) • use already existing formats that can be extended, adapted, and merged in a proper way
Description	AutomationML is about production systems and industry automation. Within IEC 62714 all parts of AutomationML are going to be standardised internationally. AutomationML serves as a file format (XML based) to aid an integration framework based engineering network. There are 5 white papers, which are subject to standardization: • Part 1 - Architecture and general requirements • Part 2 - Role class libraries • Part 3 - Geometry and Kinematics • Part 4 - Logic Description • Part 5 – Communication Consortium is very active and has current projects running. Among others, AutomationML is combined with OPC Unified Architecture [AML2016b].
Available materials	Specification, further publications, available tools, etc. can be found at [AML2016a]
ACOSAR relevance	Its process model features several phases where the ACI could have an impact on. Phases include for example requirements engineering, behaviour simulation, robot programming and simulation or test [AML2015]. The ACI can possibly support these phases, as RT systems are part of the manufacturing process. Behaviour simulation, control engineering, etc. could benefit from reduced integration efforts. Testing could be simplified as the interface to robotics, etc. is well defined and monitoring and diagnosis could also benefit from the ACI. However, the model transformation from AML to ACI needs to be established. Currently the configuration of the ACI is under development, including its data format. Regarding the interconnection of FMI with AML, scientific papers are available, e.g. [GKN2013]
Related WP	WP2
Summarized by	Martin Krammer, Nadja Marko (Virtual Vehicle)

4.8.2.4 MARTE

Standard Name	MARTE (Modelling and Analysis of Real Time and Embedded systems)
Standardization Committee	OMG
Chronology	MARTE specification version 1.0 (formal/2009-11-02) MARTE specification version 1.1 (formal/2011-06-02)
Consortium	21 contributors, all contributors can be found at http://www.omgmarte.org/node/5.
Major goals	Modelling of real-time systems and embedded systems with UML2

Description	The standard covers the description and analysis of real-time systems using UML2. It consists of 4 parts: foundations, design model, analysis model and annexes. The status of the consortium is unknown. But there seems to be some activities regarding application. Limited tool support is available: Modelio, Papyrus UML, IBM Rational Rhapsody, Magic Draw
Available materials	[MAR2011]
ACOSAR relevance	MARTE focuses on real time systems, however, the application capabilities of plain UML or SysML have already been demonstrated by others as well. Due to limited tool support, MARTE is not commonly used in industry today. Application within the automotive industry is rather low but subject to some research projects. At the moment MARTE is considered relevant for ACOSAR, but its practical use might be limited.
Related WP	WP2
Summarized By	Martin Krammer (Virtual Vehicle)

4.8.3 Summary and Conclusion

Modelling languages for the description of various aspects of RT systems and their related software are available today. All investigated languages feature a meta-model based on UML and were extended for their specific purpose. The usage of a modelling language can bring benefits when describing a system from MIL to HIL to automatically create configuration files.
All of the languages however are still *languages*. That means that a certain methodology must be established to gain a benefit from them. All of the referred languages are specified, however, their applicability often depends on tool implementations. Profiles or tool extensions are often available. Most tools also offer mechanisms to extend them. Model transformations are commonly used to generate code or other specific data files out of a model. This typically puts a modelling language or tool in the middle of the used process, as information flows in and out of the system model.

4.9 Related Research Projects

4.9.1 Introduction

As part of gaining an overview of the current state of the art, we identified 17 recent or ongoing research projects that are of relevance to the ACOSAR project. For each of these projects, we give an overview of the project period, the consortium, the available materials, and we sum up the major project goals and outcomes. For the ACOSAR project, it is of particular interest how other research projects use or develop standards; we therefore have a specific look at the standardization usage and effort of each project and how these relate to ACOSAR and the ACI. Next to standardization, we have a look at the various other ways in which a research project may relate to the ACOSAR project. For each project, we investigate the specific relevance to the ACOSAR project and, in a next step, assign the identified topics of relevance to ACOSAR work packages.

4.9.2 Relevant State of the Art Items

4.9.2.1 OPENPROD

Project Name	OPENPROD - Open Model-Driven Whole-Product Development and Simulation Environment
Project period	2009 - 2012
Consortium	Total number of partners: 28 Project budget: 11.4 M€ OEM level EADS Innovation Works, Electricité de France, Nokia Corporation, Peugeot Citroen Automobiles SA, Siemens Industrial Turbomachines AB, Siemens AG Supplier level SKF Sverige AB, Bosch-Rexroth AG, Metso Automation, Pöyry Finland Öy Tool vendors APPEDGE, Equa Simulation AB, LMS Imagine, Modelon AB, Plexim GmbH, TLK Thermo GmbH, Wolfram Mathcore AB, XRG Simulation GmbH Academic partners Bielefeld University of Applied Sciences, CEA LIST, ETH Zurich, Fraunhofer FIRST, IFP New Energy, INRIA Rocquencourt Sophia Antipolis, Linköping University, Technical University Clausthal, University of Lyon-INSA, VTT Technical Research Centre of Finnland Ltd.
Major project goals	The project integrated most aspects of product development into a holistic approach with a focus on open source. The major goals were • A model-driven rapid development and design environment for both software and hardware. • Open source tools and components for open reusable solutions. • Standardized model representation of products primarily based on Modelica and UML.
Standardization	The project focussed heavily on open source tools (e.g. OpenModelica) and advanced these. The project developed a standard for model representation of products that is primarily based on Modelica and UML. The project contributed to the FMI 2.0 specification.

Major project outcomes	The project achieved both its industrial and academic goals. • The project delivered a range of solutions, such as the ModelicaML Modelica-UML profile, Modelica3D, ontology-based simulation, simulation PDM/PLM integration using business product modelling and model-driven product optimization. • Eclipse was integrated with OpenModelica and other industrial modelling and simulation tools, and UML. • The OpenModelica model compiler was enhanced (vendor specific tool additions, many more models, OMPython, FMI import, 3D graphics animation lib, Fluid lib) • The speed of Modelica applications was improved to real-time (Gas Turbine, Engine and Hydraulics Simulators from Siemens, IFP and Bosch-Rexroth, respectively) for HiL. This improvement in speed has been achieved through translation of Modelica code into a numerically efficient target, through efficient parallelisation, and through optimization of solvers. • (Executable) Modelica models were integrated into (descriptive) SysML models.
Available materials	A collection of links to documents is available from https://itea3.org/project/openprod.html
ACOSAR relevance	OPENPROD developed a standard for model representation of products (primarily based on Modelica and UML); with this standard, it was possible to improve existing tooling and integrate the tools into a holistic approach. This integrated, holistic approach considers both hardware and software by means of a HiL setup. Modelling and simulation with model components in different formalisms were enhanced by the established interoperability between tools as well as the better re-use of tools components. This is in line with the goals of ACI. OPENPROD focussed on open source solutions as closed proprietary solutions were perceived as a hindrance to widespread dissemination and uptake; ACOSAR also favors openness. The project improved the speed of Modelica applications to make these available in a HiL context; Acosar is concerned with making soft-realtime systems seamlessly available in HiL systems. The project contributed to the FMI 2.0 standard, which also ACOSAR seeks to enhance.
Relevant WP	WP3 (this project is concerned with the simulation tool-side of a HiL setup)
Summarized by	Oliver Kotte (Bosch)

4.9.2.2 MODELISAR

Project Name	MODELISAR From System Modelling to S/W running on the Vehicle
Project period	2008 - 2011
Consortium	Total number of partners: 29 Project budget: 26 M€ OEM level Daimler, VW, Volvo

	Supplier level AVL, Altran, TWT, Trialog, Triphase, Verhaert Tool vendors Atego, David, Dassault Systemès, Geensoft, ITI, LMS, QTronic, Simpack, Academic partners AIT, Armines, ATB, DLR, Fraunhofer, IFP, Inspire, Uni Halle
Major project goals	The purpose of MODELISAR is to introduce Functional Mock-up Interface (FMI), a next generation of methods, standards and tools to support collaborative design, simulation and test of systems and embedded software [MOD2011].
Standardization	Development of FMI specification version 1.0, which is today the quasi industry standard for model exchange and co-simulation.
Major project outcomes	Release of FMI for Model Exchange in January 2010 and of FMI for Co-Simulation in October 2010. Proof of concept within industrial use-cases. Fast adoption by MODELISAR and non MODELISAR tool vendors. Start of FMI 2.0 development.
Available materials	Project data: itea3.org/project/MODELISAR.html FMI specifications and further information: www.fmi-standard.org
ACOSAR relevance	In MODELISAR a consortium of tool vendors, users and research organization developed a tool independent interface specification in close cooperation. Tool vendors provided test implementations early which were used in industrial use cases. In this way a proof of concept and an adaption to specific requirements were possible. The FMI specifications were made publically available for non MODELISAR participants immediately after their finalization. Results were presented at conference as soon as possible. This boosted the adoption of FMI by non-MODELISAR participants. After finalization of the project the achieved results were transferred to the Modelica Association which organized maintenance and further development in the founded Modelica Association Project FMI. Project participation is open for Modelica and non-Modelica tool vendors, users and research organizations. In this way a stable and agile development is ensured. ACOSAR has a similar structure and intention. We should try to reuse as much as possible of the MODELISAR procedures and experiences.
Related WP	The MODELISAR experiences are relevant for the following work-packages: **WP 6: Advanced Co-Simulation Interface (ACI):** At the beginning of MODELISAR nearly each of the partners had different ideas about what should be included in the FMI standard. In order to concentrate and share the work load, we founded different working groups: FMI for Applications, FMI for PLM, FMI for Model Exchange and FMI for Co-Simulation. The groups interacted if possible and necessary. Each group published its own specification. Version 1.0 of the FMI for Co-Simulation specification reused parts of the earlier published FMI for Model Exchange specification. Both specifications have been unified and combined later in the FMI 2.0 specification. Splitting into groups according

to well defined tasks can be an advantage for achieving results in an efficient way.

Even in these working groups there have been different opinions on the content and functionality of FMI. For example there have been attempts to describe and specify co-simulation master algorithms within the FMI for Co-Simulation specification. At the end this was only done for a very simple one in order to describe the nature of co-simulation. What the consortium did was to analyse a certain amount of algorithms and made sure that these algorithms can be implemented using FMI for Co-Simulation. As a consequence at an early stage of ACI definition we need to define clearly what is included and what is not included in the specification.

One reason for the fast adoption of FMI by tool vendors is the simplicity and lightness of the API (for example compared to the AUTOSAR specification). This was achieved by concentration on the core functionality and a conscious omitting of features which are nice to have but only for a small number of use cases (at least for the moment). This should be done in ACOSAR too.

WP 3: Simulation Tool Interface:

Before FMI specifications have been released, test implementations of most of the features have been done and were presented. Later on the consortium noticed that test implementation for all features are necessary. This should be done in ACOSAR too.

WP 7: Application Use Cases:

Like in MODELISAR the use case should be used at the beginning of the project for gaining requirements and feature request of ACI. Later on (already during development of the ACI specification) they can be used for test implementations and first proofs of concept. During dissemination the use cases should be used as demonstrators of the newly developed interface.

WP 8: Dissemination and Exploitation:

Project results should be published as soon as they are available. ACI specifications should be provided to non-project partners too as soon as first test implementations are available that proof the usability its concept.

At the end of MODELISAR the consortium asked itself what to do with the project results. In a very late stage we decided to transfer the FMI related results to a new Modelica Association Project which owns the IP and is responsible for maintenance and further development. In ACOSAR this should be done in a similar way, but a bit earlier in order to avoid the 6 month "organizational vacuum" we had between MODELISAR end and official adoption by Modelica Association, since bylaws had to be changed and infrastructure needed to be established.

Summarized by	Torsten Blochwitz (ITI)

4.9.2.3 ASTERICS

Project Name	ASTERICS (Ageing and efficiency Simulation & TEsting under Real world conditions for Innovative electric vehicle Components and Systems)
Project period	10.2012–09. 2015
Consortium	Total number of partners: 10 Project budget: 4.3 M EUR

	OEM level Volvo, Centro Recerche Fiat Supplier level THIEN eDrives, Gustav Klein Tool vendors AVL GmbH, LMS International (now Siemens Industry Software NV), LMS Imagine (new Siemens Industry Software SAS), Academic partners FH Joanneum, University of Ljubljana, Università degli Studi di Firenze
Major project goals	• Development of an approach for the design • development and testing phases of E-drivelines in Fully Electric Vehicles • reduction of the overall development and testing efforts by 50% • improvement and optimisation of the overall efficiency and performance of electric vehicles by at least 20%
Standardization	Modelisar-FMI was used. No further Standards were used, but specifications for e-motor, inverter an battery testing have been established
Major project outcomes	Achievement of objectives • Advanced testing methodologies and models for E-driveline components • Development of accurate high fidelity model for batteries, inverters and electric motors • Development of procedures for accelerated ageing of battery, inverter and electric motor to shorten the testing time • Complete vehicle simulation model, including virtual driver setup and driving cycle description • Real world environment and conditions based drive cycles • Advanced testing methodologies and models for E-driveline components • Descriptive/predictive models for battery subsystem, power electronics and electric motor • Total system (e-driveline and FEV) • development times are improved by 40% • New state of the art 3D-models for e-components with multi-physics (thermal, electric, mechanical, magnetic) and ageing algorithms developed
Available materials	Material is available online on http://www.asterics-project.eu/
ACOSAR relevance	In ASTERICS, couplings of models and HiL-test benches with different properties (e.g. thermal, battery management) were realized. Components and in/output definitions available. MiL/SiL/HiL combinations were realized in the project, which is also focus in ACOSAR. The co-simulation setup can be used to derive requirements for ACOSAR. On the other hand, a use case using ACI with the models developed in the project could be setup. Development and demonstrations of interest for ACOSAR:

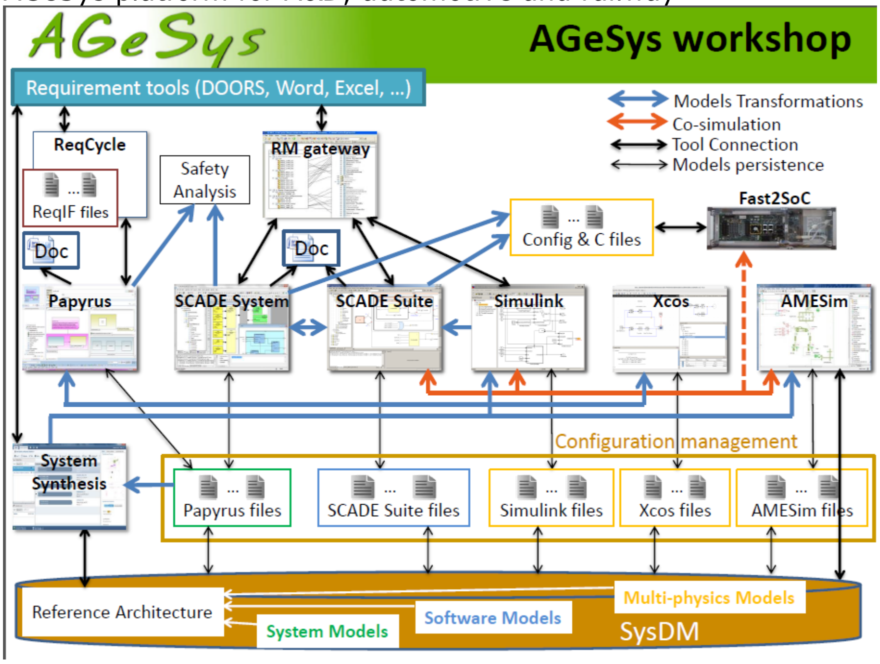
	- FMI vs interoperability testing and validatation between Amesim and Cruise (FMU cs) - System integration (FCA Gofast and Volvo GSP platform, Amesim) and simulation of heterogeneous systems (S-functions, FMI) with substitutable models from different tools providers (Siemens, AVL, Mathworks…) - New validated Amesim models (inverters, emotors) with RT capabilities Some lessons learnt so far: - Structured models IOs configuration required for proper model exchange and substitutability. - Challenges in co-simulation performance understanding and management from a user point of view.
Related WP	-WP2: RT models integration with test benches -WP3: models coupling and heterogeneous system integration
Summarized by	Pacôme Magnin (Siemens)

4.9.2.4 AGeSys

Project Name	AGeSys (Atelier de Génie Système)
Project period	2012 – 2015
Consortium	Total number of partners: 15 Project budget: 21.2 M€ OEM level Airbus, Alstom, PSA, Renault Supplier level Continental, Sagem, Scaleo chip, Snecma, Thales, Valeo Tool vendors Atos, thales R&T, LMS IMAGINE (Now a siemens subsidiary), ESTEREL TECHNOLOGIES(now an ANSYS subsdiary), SCILAB Academic partners CEA
Major project goals	The objective of the AGeSys project is the development of a systems engineering workshop dedicated to embedded systems, for all industrial sectors without restriction. This workshop is made of open, integrated tools that allow a model based design of embedded systems and software, and coupled with multi-physics models. The objective is to allow an industrial process with a clean definition of the system/software/hardware steps.
Standardization	Several tools share common technical components (Such as the Papyrus platform from CEA LIST) and public standards compliance (SysML, FMI)

Major project outcomes

AGeSys platform for A&D, automotive and railway



Available materials

http://www.systematic-paris-region.org/fr/projets/agesys

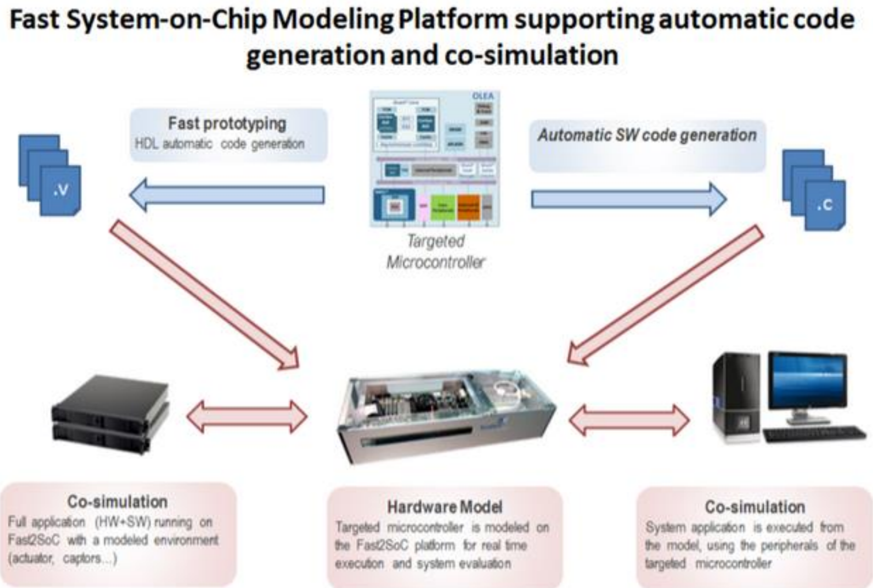
http://www.systematic-paris-region.org/en/news/agesys-bgle-project-ocds-project

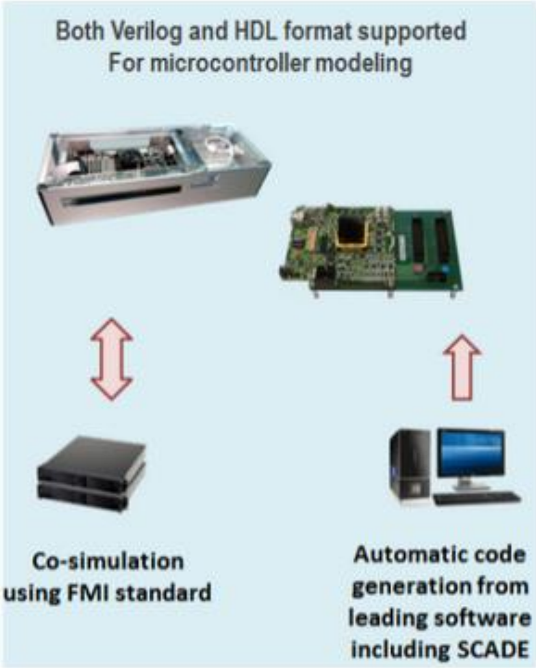
ACOSAR relevance

Usage of FMI for co-simulation:

- 1 to 1 coupling (Amesim, Scade suite, scade display, scade rapid prototyper, XCOS)
- Real-time co-simulation of multiple FMUs integrated in Amesim (https://www.youtube.com/watch?v=IJXqO0Ug3LA)
- Multi-level FMU

Co-Simulation using hardware model running in RT on Fast2Soc platform coupled with a RT compliant FMI cs through a wrapper:





	Generation of heterogeneous co-simulation targets (Simulink, Amesim, cosimulation buses) based on configured simulation architecture (Amesim, Simulink, FMI) in Siemens System Synthesis. Lesson learnt so far: - IT challenges with remote co-simulation (security-related constraints) - Numeric stability co-simulation challenges for RT co-simulation - Structured system configuration required for proper model management, assembly and target generation (implemented in Siemens System Synthesis)
Related WP	- WP2: system-level configurations - WP3: coupling interfaces - WP6: heterogeneous RT cosimulation
Summarized by	Pacôme Magnin (Siemens)

4.9.2.5 IMPROVE

Project Name	IMPROVE (Integration and Management of Performance and Road Efficiency Of Electric Vehicle Electronics)
Project period	2013 - 2016
Consortium	Total number of partners: 10 Project budget: unknown OEM level TOFAS Turk Otomobil Fabrikasi AS Supplier level

	Continental Temic Automotive Electric Motors GmbH, IDIADA Automotive Technology SA, Brusa Elektronik AG Tool vendors LMS Imagine (now Siemens Industry Software SAS), SIC! Software GmbH Academic partners Virtual Vehicle Competence Center, Fraunhofer-Gesellschaft Zur Förderung Der Angewandten Forschung E.V, Czech Technical University in Prague (CTU), The Università degli Studi di Firenze (UNIFI)
Major project goals	Improvement of BEVs with respect to driving comfort and performance, overall energy consumption, reduction of production and operation cost, sufficient real life driving range, transferable technology and optimized charging time.
Standardization	-
Major project outcomes	-
Available materials	http://improve-fp7.eu/
ACOSAR relevance	Co-simulation with ICOS involving Amesim (vehicle systems, subsystems, controls). Amesim model reduction for fast complex system simulation (done for MPCs model integration on ECUs but may be of interest for RT HiL too). MiL testing using Amesim vehicle and subsystems models. Lesson learnt so far: - IT challenges with remote co-simulation (security-related constraints) Model reduction processes and strategies for very complex models.
Related WP	Experience with model reduction strategies might be helpful in WP7, when former offline models have to be trimmed to achieve soft real-time conditions. IT challenges encountered and solved in the project might provide information on how to avoid similar problems during the design phase of the ACI element and the implementation (WP6 & WP5). Further, it is relevant for WP2 because of the model reduction topic.
Summarized by	Timo Haid (Porsche AG) Pacome Magnin (Siemens)

4.9.2.6 TRANSFORMERS

Project Name	TRANSFORMERS (Configurable and Adaptable Trucks and Trailers for Optimal Transport Efficiency)
Project period	01.09.2013 – 28.02.2017
Consortium	Total number of partners: 13 Project budget: 7.9 M€ (5.2 M€ EC funded)

OEM level

VOLVO, DAF, Schmitz Cargobull, Van Eck, End User level, P&G, IRU Projects

Supplier level

Bosch, Uniresearch B.V.

Academic partners

Fraunhofer, TNO, FEHRL, IFSTTAR, Virtual Vehicle

Major project goals

The TRANSFORMERS project developed a modular approach for optimization of "rightsizing by means of hybridization, truck engine downsizing and a trailer design that addresses simultaneously aerodynamics and load efficiency improvements". "The overall goal is to achieve 25% energy load efficiency (in energy/km.tn) in a real world application taking into account the needs to maintain road infrastructure and traffic safety". [PrTR03] Accordingly, the following developing and designing tools were established: Hybrid-on-demand driveline, Loading efficiency toolbox, Aerodynamic design toolbox (Figure 7) and a Hybrid-on-demand Framework.

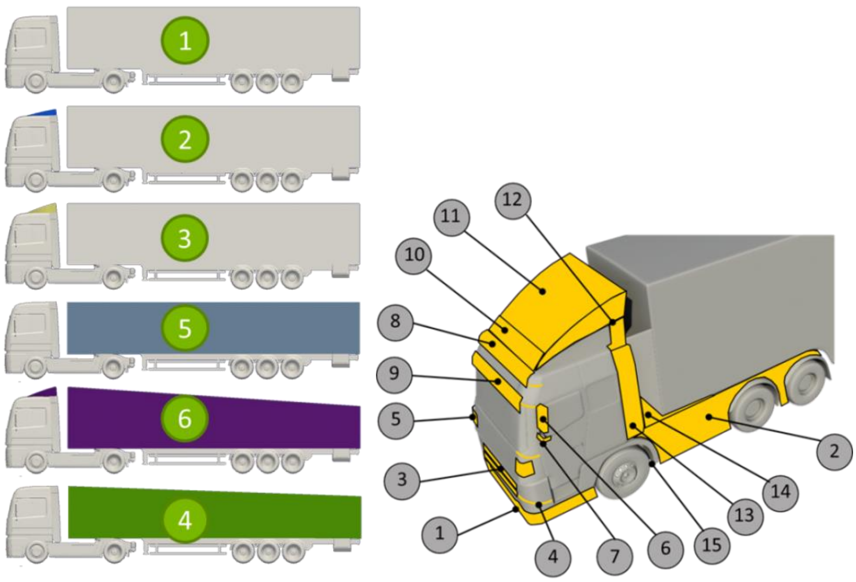


Figure 7: Aerodynamic semi-trailer design and targeted innovations [PrTR01]

Picture: demonstrator for Hybrid-on-Demand Driveline for Truck-trailer Combination ➔ fuel consumption reduction by 18% total:

- Load optimisation / efficiency up to 8% by adjusting the trailer shape / roof height ➔ aerodynamic toolbox: reducing the cross-section area and air resistance
- 2% fuel consumption reduction by downsizing the engine
- 8% by the Hybrid-on-Demand driveline

Standardization

No specific standards used – only use of optimizations functions, their solution and models, for example: trailer depreciation, trailer maintenance cost cost/km, trailer tare weight, etc.

The only standards (VSI 2700 ff, EN 12195, EN 12642, EN12642 XL, ISO/TC 22 Road vehicles) which are used, do not related to the ACOSAR project

	No significant efforts to establish a specification or standard
Major project outcomes	• Key Performance Indicators and End User requirements: An innovative truck-trailer/tractor-semitrailer driveline with up to 8% energy saving per tonne-km in long haulage and up to 12% energy saving per tonne-km for distribution traffic compared to state-of-the-art truck-trailer/tractor-semitrailer vehicles • Hybrid-on-demand Driveline: First-time demonstration of a distributed hybrid driveline concept with an internal approach on rightsizing the driveline for each transport mission • Loading efficiency toolbox • Aerodynamic design toolbox: Design and development of mission adapted aerodynamic solutions giving up to 10% in reduction of fuel • Hybrid-on-demand Framework
Available materials	http://www.transformers-project.eu/mainmenu/home/#.VmWKGHYvfmE
ACOSAR relevance	*This Project relates to ACOSAR by its modular vehicle architecture and library in Matlab, AVL Cruise and IPG CarMaker for holistic and experimentation and by its layer architecture (Information / Management / Physical/Application Layer):*

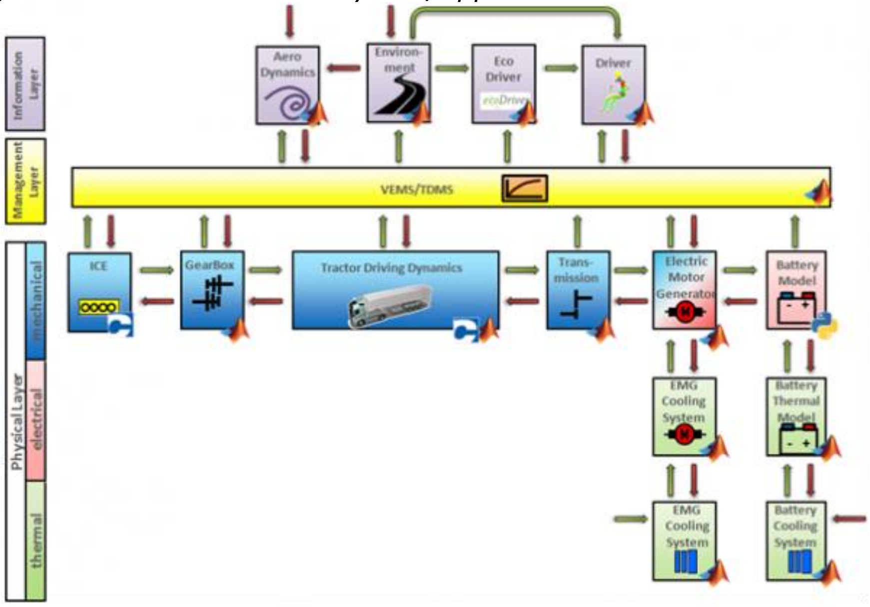


Figure 8: Modular Vehicle Architecture [PrTR02]

All these models are integrated in a non-real-time capable co-simulation environment ICOS. According to the ACOSAR use cases a real-time capability is required to connect HiL-systems, especially with a strong coupling of physicals phenomena. However, the TRANSFORMER project does not including explicit HiL use cases.

Related WP	➢ Relating to the ACOSAR project the proposed architecture and interface layering for co-simulation can be a starting point for WP 3: Simulation Tool Interface and WP 6: Advanced Co-Simulation Interface or to advance this suggestion in WP 2: RT-System

	Integration Methodology, WP 4: RT-System Interface and WP 6: Advanced Co-Simulation Interface.
Summarized by	Viktor Schreiber (Technische Universität Ilmenau)

4.9.2.7 Cossim

Project Name	COSSIM - A Novel, Comprehensible, Ultra-Fast, Security-Aware Cyber Physical Systems (CPS) Simulator
Project period	2015 - 2018
Consortium	*Total number of partners:* 8 Project budget: 2.8 M€ *Supplier level* ST Microelectronics, Search-Lab, Maxeler Technologies *Tool vendors* Synelixis Solutions Ltd. *Academic partners* Telecommunication Systems Institute, Tecnalia, Politecnico di Milano, Chalmers University of Technology
Major project goals	The COSSIM project has 5 major project goals [COS2016]: • Simulate CPS using a self-developed open-source simulation framework using, e.g., multi-core CPUs and complex and heterogeneous networks. • Speed-up the simulation of CPS by at least one order of magnitude. • Increase the precision of power estimations. • Support security levels and features in a CPS tool. • Build two real-world demonstrators from two domains to show the successful implementation of the goals.
Standardization	We are not aware of any standardization activities of the COSSIM consortium.
Major project outcomes	As the project just started, the major outcomes are not published yet.
Available materials	http://www.cossim.org/ [COS2016]
ACOSAR relevance	The framework that is going to be designed in COSSIM exhibits overlap with ACOSAR's Functional Framework (e.g. regarding network communications). Further, the aspects of communication and co-simulation are also relevant for ACOSAR.
Related WP	WP 5, WP 6
Summarized by	Leonid Lichtenstein (TWT)

4.9.2.8 Mosaik

Project Name	MOSAIK
Project period	2011 - now
Consortium	Total number of partners: unknown Project budget: unknown

	Academic partners OFFIS Institute for Information Technology
Major project goals	MOSAIK is an open source project that was initiated by OFFIS. The goal is to "reuse and combine existing simulation models and simulators to create large-scale Smart Grid scenarios". [MOS2016]
Standardization	We are not aware of standardization activities related to MOSAIK.
Major project outcomes	There is a number of project outcomes, summarized in the following: • The MOSAIK API allows the user to integrate different simulators into MOSAIK, independent of the language of their implementation. • A simple API enables the user to create large-scale simulation scenarios. • MOSAIK coordinates the execution of the simulators by employing an event-discrete simulation.
Available materials	• [SSS2012] • https://mosaik.offis.de/ [COS2016] (open source software available)
ACOSAR relevance	Mosaik's framework version 2.1 supports real-time simulations. For ACOSAR it might be helpful to align with the way the real-time functionalities are implemented.
Related WP	WP 4
Summarized by	Leonid Lichtenstein (TWT)

4.9.2.9 AVANTI

Project Name	AVANTI - Test methodology for virtual commissioning based on behaviour simulation of production systems
Project period	2013 - 2016
Consortium	Total number of partners: 12 Project budget: 4.6 M€ OEM level • Daimler AG (DE) • Arcelik A.S. A.S. (TR) Supplier level • Moventas Gears Oy (FI) • Festo (DE) • TWT GmbH Science & Innovation (DE) • KaTron Defence Aerospace and Simulation Technology (TR) • tarakos GmbH (DE) • EKS GmbH (DE) • WWP-Systeme GmbH (DE) Academic partners • Lappeenranta University of Technology LUT (FI) • Institute für Automation und Kommunikation e.V. Magdeburg – IFAK (DE) • Koc University (TR)
Major project goals	The AVANTI project has three major objectives:

	• "Behavioural description of production lines and their individual components • Establishing formal test methods for Virtual Commissioning • Improving the liability and quality of the production system" [AVA2015]
Standardization	Industry standards in use: FMI, AutomationML No activities towards standardization planned.
Major project outcomes	The following list shows some of the project outcomes: • "Realistic physics-based production equipment simulation • Automated test generation & execution • Seamless standard CAx process chain" [AVA2015]
Available materials	http://avanti-project.de/ [AVA2016] https://itea3.org/project/avanti.html [AVA2016a]
ACOSAR relevance	In AVANTI, a programmable logic controller of a production plant is linked as a HIL to the co-simulation of different mechatronic components and a physics-based engine using the FMI standard. This application has real time requirements. An exchange of experience with the partners might be helpful in defining the requirements for the ACI. Further, Avanti partners might be interested in joining the ACOSAR Industrial Follower Group.
Related WP	WP 3 and 6: ACOSAR consortium can learn from the simulation tool interface implementation experience of the AVANTI consortium. Further an exchange of experience is helpful for defining the ACI. Further, an exchange of experience with the partners might be helpful in defining the requirements for the ACI in WP1.
Summarized by	Leonid Lichtenstein (TWT)

4.9.2.10 INTO-CPS

Project Name	INTO-CPS Integrated Toolchain for Model-based design of Cyber-Physical Systems
Project period	2015 - 2017
Consortium	Total number of partners: 11 Project budget: 8 M€ OEM level - Supplier level ClearSy, UTRC, TWT, Agro Intelligence Tool vendors Controllab Products, Verified, Softeam Academic partners Aarhus University, University of Newcastle, University of York, University of Linköping, University of York
Major project goals	The main objectives of INTO-CPS are listed in the following [ICP2016]:

	• Create an open tool chain for the multidisciplinary design and modelling of CPS covering the full development life cycle. • Provide a good semantic foundation for the CPS tool chain including contributions to, e.g., FMI and SysML. • Develop a methodology to support the tool chain in the form of practical guidelines and patterns. • Show the effectiveness of the implementation in an industrial setting by conducting four case studies from different domains.
Standardization	Standards used: FMI, SysML Contributions to a further development of FMI are planned, but have not yet been brought into the FMI project. In particular, it is planned to contribute to the specification of hybrid systems (mixed continuous time – discrete event models).
Major project outcomes	Project is ongoing, therefore the achievement of goals cannot yet be assessed.
Available materials	Public deliverables and documents will be posted on the website as soon as they will be available: http://into-cps.au.dk/ [ICP2016]
ACOSAR relevance	INTO-CPS is developing a FMI-based Co-simulation engine, also considering HiL tests. HiL is, however, a minor topic of the project. In general, INTO-CPS and ACOSAR are complementary to each other and INTO-CPS partners are open for an exchange of general methodology and ideas.
Related WP	WP 3 & 5: the INTO-CPS project also deals with simulation tool interfaces and communication protocols.
Summarized by	Christian König, Leonid Lichtenstein (TWT)

4.9.2.11 CPSE Labs

Project Name	CPSE Labs - Cyber-Physical Systems Engineering Labs
Project period	na
Consortium	Total number of partners: 9 Project budget: na Industry partners: Indra Sistemas Academic partners: fortiss, KTH, ONERA, LAAS-CNRS, Newcastle University, OFFIS, Technical University of Madrid, Steinbeis Europa Zentrum
Major project goals	CPSE Labs makes technical support and funding available to European technology businesses. CPSE Labs is a European Union-funded initiative designed to provide support for engineering and technology businesses in Europe.
Standardization	-
Major project outcomes	CPSE Labs primarily supports businesses by funding experiments. Organizations working in several engineering domains can design and propose research experiments to CPSE Labs. CPSE Labs will review the proposed experiments and the best will receive funding [CPS2016].

Available materials	[CPS2016]
ACOSAR relevance	CPSE Labs is no classical project but an opportunity to get funding for experiments in the field of cyber-physical systems. There are some case studies (http://www.cpse-labs.eu/casestudies.php) concerning simulation, but as CPSE Labs does not provide concrete results no concrete relation can be shown. However, there may be experiments that are interesting for ACOSOR or ACOSAR can submit a proposal for a real-time co-simulation experiment.
Related WP	WP7: Application Use-Cases & Assessment
Summarized by	Nadja Marko (Virtual Vehicle)

4.9.2.12 ViProMa

Project Name	ViProMa
Project period	na
Consortium	Total number of partners: na Project budget: na Industry: Rolls-Royce Marine; Vard Group; OSC; and DNV GL Academic partners University College of Ålesund, SINTEF,MARINTEK
Major project goals	"The objective of this KPN project is to develop a common framework for the co-simulation and virtual prototyping of marine systems and operations for cases that are as realistic as possible." [VIP2016b]
Standardization	FMI is used for co-simulation
Major project outcomes	Major project outcomes are the release of a first version of the co-simulation bus based on the de-facto standard FMI (2014). Further, project partners have been able to submit, set up and demonstrate co-simulation for the virtual prototyping of their own simulation models working together with models from other partners (2014) [VIP2016b].
Available materials	There is project site [VIP2016a] and a project summary [VIP2016b] where the project site is only available in Norwegian language. No detailed information about project is available.
ACOSAR relevance	As there is no detailed information about the project, the relation to ACOSAR cannot be described in detail. ACOSAR focuses on real-time simulation, whereas ViProMa targets the development of a co-simulation bus. However, the development of a co-simulation bus based on FMI is strongly related to ACOSAR.
Related WP	WP2: RT-System Integration Methodology WP3: Simulation Tool Interface
Summarized by	Nadja Marko (Virtual Vehicle)

4.9.2.13 ACoRTA

Project Name	ACoRTA (Advanced Co-Simulation Methods for Real-Time Applications) 1 & 2
Project period	01/2012-12/2017
Consortium	Total number of partners: 6 Project budget: ACoRTA-1: 1M€, ACoRTA-2: 1.1 M€ OEM level: Porsche AG, Volkswagen AG Supplier level: AVL List GmbH Academic partners: Alpen Adria University Klagenfurt, Graz University of Technology, VIRTUAL VEHICLE Research Center
Major project goals	A self-evident extension of the (non-real-time, offline) co-simulation approach represents the integration of real-time systems. In this case, one or more components that are available as real hardware are directly implemented into the existing system model. In order to manage the various challenges, the experts of the ACoRTA project developed new coupling algorithms that ensure stable closed-loop characteristics of the entire system. The so-called model-based coupling methodology is based on models of the involved subsystems, identified during runtime. These models simulate the dynamic behaviour of the components models in the respective operating point. The focus of the ACoRTA-2 project lies on the development of model-based fault detection techniques to ensure save real-time co-simulations. In real-time co-simulation setups at least one real-time system is integrated in form of real-hardware. To avoid any damage on this real hardware caused by unstable offline simulation tools, extrapolation errors, etc. fault diagnosis techniques are essential to detect and handle such faults. If any faults are detected the hardware should immediately switch to a save operation independent of the simulation state of the offline world. Until now the configuration of such real-time co-simulations requires a lot of expert knowledge (experience) gained in the ACoRTA project. To make the ACoRTA approach generally applicable the usability has to be improved which is the second major goal of the ACoRTA-2 project.
Standardization	ACoRTA focuses on the coupling challenge and thus, industry standards were not investigated. However, during ACoRTA-1 the FMI Standards became more famous and were used to integrate some simulation models into the co-simulation platform ICOS, via "FMI for Co-Simulation". During ACoRTA-1 members of the consortium discussed several times possible standardisation attempts for using co-simulation throughout the entire development process. ACOSAR was initiated based on these discussions.
Major project outcomes	Researchers are now able to predict the future behaviour of the sub-models in the extrapolation stage to handle occurring dead times respectively data losses. In addition, the identified models of the subsystems are indispensable for efficient noise suppression typically introduced via measurement sensors.

Figure 2 shows the effects of the real-time co-simulation and the model-based coupling approach. The virtual prototype vehicle tracks the NEFZ (Neuer Europäischer Fahrzyklus) driving cycle (black line, short excerpt). The blue line represents the determined result of the offline co-simulation. As there are no latencies, it can be considered as a correct result. When directly connecting the control unit – which means the latencies are not compensated – strong oscillations of the vehicle's velocity are remarkable caused by the communication time-delays, see red line in Figure 9. With the help of model-based coupling techniques developed in the ACoRTA project, they can be compensated and the results of the real-time co-simulation (see green line in Figure 9) are correct due to the small deviations to the offline simulation result.

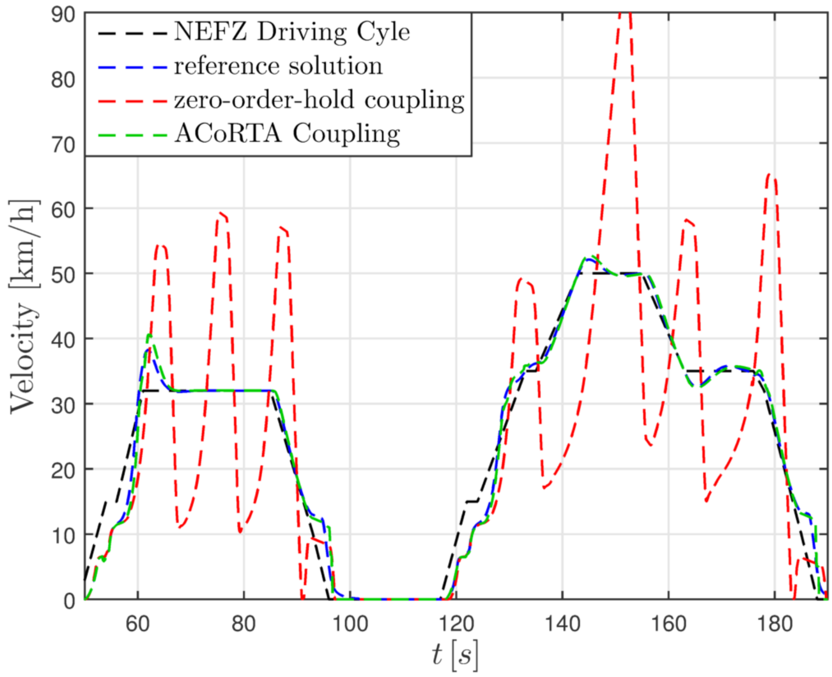


Figure 9: Effects of model-based coupling

Thanks to the successful extension of the typically offline co-simulation approach into the real-time domain, for the first time it is possible to consistently use the co-simulation approach throughout the entire product development process (e.g. using the V-model). In consequence, engineers are able to easily exchange virtual components and real hardware components. Moreover, the performance of existing test systems could be improved. For example, at an engine test bench at AVL List GmbH, the ACoRTA team proved that by reducing the communication dead time, the normal quality (scope) can be significantly increased.

The successful integration of real hardware components into the classical co-simulation approach (real-time co-simulation) is strongly related to the model-based coupling technology developed. This model-based coupling approach is designed to handle the three most important coupling imperfections caused by the integration of real hardware: communication time-delays, data-losses and noisy measurements. The successful compensation of these coupling imperfections is one possibility to adequately handle real-time co-simulation problems.

Available materials	General project information: [ACO2016] ACoRTA papers: [ZSK2014], [SZB2014], [TBZ2013], [SZB2013], [SBT2013]
ACOSAR relevance	ACoRTA focuses on solving hardware integration problems by designing special coupling techniques. In contrast, the ACOSAR project follows the idea of FMI and focuses on a standardized real-time system interface description to enable a highly efficient integration of real-time systems into simulation environments as well as improved usability. The major connecting factors or links to ACOSAR are • Real-time co-simulation problem • Real-time system integration • Real-time communication media • Communication time-delays
Related WP	WP2, WP3, WP4 and WP5
Summarized by	Georg Stettinger, Nadja Marko (Virtual Vehicle)

4.9.2.14 VeTeSS

Project Name	VeTeSS – Verification and Test to Support Functional Safety Standards
Project period	05-2012/04-2015
Consortium	Total number of partners: 24 Lead: Infineon UK All partners can be found at: http://vetess.eu/project-structure/partners/
Major project goals	Develop new methods, tools and processes supporting the automotive functional safety standard ISO 26262
Standardization	VeTeSS was completely built on the automotive domain, especially the rising safety standard ISO 26262.
Major project outcomes	The project enhanced the understanding of the new functional safety standard ISO 26262 by differentiating 5 different technical streams and 3 different work packages reflecting different levels of development (System, Software, and Hardware).
Available materials	[VET2015]
ACOSAR relevance	Functional safety: ACOSAR deals with RT systems where additional requirements apply which are at least safety relevant (or even critical) for the operation of RT systems. To ensure safe operation of RT systems, safety/RT requirements must be identified and satisfied. Systematic approach: RT co-simulation scenarios are considered complex systems featuring highly networked functions. On information level, a systematic approach for definition and description of single system components is sought. On instantiation level, a systematic approach for configuration and deployment of single system components in context of a complete system is sought. Requirement engineering: The ACI is currently considered as a software component where RT and non-RT requirements apply. Based on Use Cases introduced by project partners, these requirements should be derived in a traceable manner. As VeTeSS featured a traceable requirements process

	and continued innovation management, the experience gained from this context is considered a main asset for ACOSAR. For WP4 (System integration) and TS2 (Modelling & Simulation) Virtual Vehicle contributed a model based approach for co-simulation configuration, enhancing the systematic construction of system simulations. The model based approach as well as the experience gained with model based software and systems engineering tools serve as a basis to reach aforementioned goals within the ACOSAR project.
Related WP	This is relevant for ACOSAR in WP1 and 2. The relevance for WP1 stems from the fact that concepts for the construction of co-simulation scenarios and instantiation of the ACI are needed. This is also related to Master-Slave functionality. The main output will be sets of requirements (core & technical). WP2 investigates means for RT system integration, this includes model based methodologies.
Summarized by	Martin Krammer (Virtual Vehicle)

4.9.2.15 DACCOSIM

Publication name	FMI-based distributed multi-simulation with DACCOSIM
Authors	Galtier, Virginie; Vialle, Stéphane; Dad, Cherifa; Tavella, Jean-Philippe; Lam-Yee-Mui, Jean-Philippe; Plessis, Gilles
Year	2015
Reference	Proceedings of the Symposium on Theory of Modelling & Simulation: {DEVS} Integrative M&S Symposium, part of the 2015 Spring Simulation Multiconference, SpringSim '15, Alexandria, VA, USA, April 12-15, 2015, http://dl.acm.org/citation.cfm?id=2872971
Summary	The paper describes an architecture to perform distributed co-simulation based on FMI 2.0 standard. It is an instantiation of the high-level architecture (HLA) standardized by IEEE. The approach distinguishes between local master and global master, and between communication of data and control (for e.g. establishing next communication step size), optimizing the interaction between the involved FMUs.
Project	DACCOSIM (Distributed Architecture for Controlled CO-SIMulation) DACCOSIM is developed by the CentraleSupélec IDMaD research team and the EDF R&D MIRE department in the RISEGrid Institute.
ACOSAR relevance	The usecases of ACOSAR are targeting network HiLs, which will be in turn an example of a distributed real-time simulation. For real-time simulation is of crucial importance that the overhead of communication and control does not have negative impact on meeting the deadlines. From the paper reviewed here we can learn how we can split between local and global control, and how a greedy communication approach can be implemented. Especially interesting is the initialization approach, where information on the underlying strongly connected components and the directed acyclic graph is used to run iterative methods or propagate initial values, respectively.

Related WP	WP1, WP2, WP7
Summarized by	Corina Mitrohin, ETAS

4.9.2.16 MBAT

Project Name	MBAT - Combined Model-based Analysis and Testing of Embedded Systems
Project period	2011 - 2014
Consortium	Total number of partners: 38 Project budget: 34,5M€ OEM level: Daimler, Volvo, Airbus, Alenia Aermacchi, Alstom, EADS, Siemens, Thales Supplier level: ALES, AMET, Ansaldo STS, AVL, Infineon, Ricardo, Rockwell Collins, Seles Sistemi Integrati Tool vendors: Absint, All4Tec, BTC-ES, Elvior, Dassault Systems, MBTech, PikeTec, IBM Academic partners: Aalborg University, AIT, CEA List, École normale supérieure Paris, ENEA, Fraunhofer IESE, KTH, Mälardalen University, OFFIS, Technical University Graz, Technical University Munich, Virtual Vehicle Competence Center
Major project goals	Develop solutions to combine static analysis and testing as well as an interoperability platform based on OSLC.
Standardization	OSLC (Open Services for Lifecycle Collaboration): Used and contributed to standard
Major project outcomes	Several approaches combining the strengths of static analysis and testing have been proposed. Data management and exchange have been realized with OSLC and IBM Jazz.
Available materials	Deliverables and reports can be found at: [BAT2014]
ACOSAR relevance	Development of interoperability specification for the verification domain, implementation of OSLC adapters and integration into OSLC based platforms like IBM Jazz are the main relation points to ACOSAR. However, the interoperability specification and the implementation of it concerns lifecycle development tools and is not focused on simulation. Hence, there is no close relation to ACOSAR.
Related WP	WP2: RT-System Integration Methodology
Summarized by	Christian Schwarzl, Nadja Marko (Virtual Vehicle)

4.9.2.17 CRYSTAL

Project Name	CRYSTAL (CRitical sYSTem engineering AcceLeration)
Project period	2013-2016
Consortium	Total number of partners: 68 Project budget: about 82 million Euros

	All partners can be found at http://www.crystal-artemis.eu/partner.html
Major project goals	CRYSTAL's goal is to enable common interoperability among various life cycle domains. The project is "strongly industry-oriented and will provide ready-to-use integrated tool chains having a mature technology-readiness-level (up to TRL 7)" [CRY2016]. Further, CRYSTAL wants to support cross-domain reusability (Aerospace, Automotive, Health and Rail).
Standardization	In CRYSTAL the interoperability industry standard OSLC is used. Moreover, CRYSTAL drives the Interoperability Specification forward towards standardisation.
Major project outcomes	Within and across the industrial domains Aerospace, Automotive, Healthcare and Rail, major project outcomes that concern the entire software product life cycle are [CRY2016]: • New engineering methods for industrially relevant use cases and an maturity increase of existing concepts • Technical innovations with high maturity to support the use cases • Standardization of the interoperability specification • Support of SME integration into the embedded systems domain • Ready-for-use industrial tool chains As the project is still ongoing, the final results are not available yet.
Available materials	CRYSTAL project information, deliverables and further publications arisen from CRYSTAL can be found at [CRY2016]
ACOSAR relevance	As CRYSTAL covers the whole product life cycle, simulation practices and methods are part of CRYSTAL. However, the integration of lifecycle tools is different to the integration of simulation tools and RT systems which is focused by ACOSAR. Hence, in ACOSAR other integration challenges have to be handled than in CRYSTAL. Nevertheless, methods for integrating simulation in the development process are part of CRYSTAL where ACOSAR can base on.
Related WP	WP2: RT-System Integration Methodology WP3: Simulation Tool Interface
Summarized by	Nadja Marko (Virtual Vehicle)

4.9.3 Summary and Conclusion

When reading through the previous project summaries, one notices that certain topics underlie and connect these projects. These topics are made apparent by the following keywords, which appear repeatedly in the project summaries:

- seamless MiL / SiL / HiL tool (and process) chain
- tool integration
- integration of models; modularization; cross-domain reusability; interoperability
- (real-time) co-simulation
- reduction of software models (for real-time simulation)
- traceability of requirements
- standardization
- open source

One can thus discern a clear trend towards integrated, holistic approaches to model-based systems engineering in real-time, across domains and enterprises, with a focus on openness and standards (such as FMI, SysML, OSLC). The broad scope of this trend across industries is exemplified by such diverse application areas as electric vehicles, production lines, and marine systems – all of these are subjects of at least one of the above summarized research projects.

The existence of this trend, and its broad scope, underline the importance of the ACOSAR project – but they also stress its challenges: The ACI standard will have to suit a large group of diverse use cases.

Awareness of the above mentioned research projects by the ACOSAR consortium supports the ACOSAR project. On a technical level, some of the research projects may serve as use cases to define requirements for the ACI standard in an early state of the ACOSAR project; they may also serve as test cases and demonstrators for the ACI standard in a late state of the ACOSAR project. On a management level, ACOSAR should apply the lessons learned from the above summarized projects. This is particularly true for the MODELISAR project, which developed the FMI standard that the ACI standard may extend. The probably most important lessons learned include

- ACOSAR should consider splitting the consortium into groups according to well-defined tasks; this can be efficient.
- The ACOSAR consortium should discuss and define clearly what is included and what is not included in the ACI standard.
- The ACOSAR consortium should focus on the core functionality, so that the ACI standard will be simple and light – this will most likely facilitate a fast adoption of the standard by tool vendors.
- Test implementations for *all* features of ACI will probably be necessary.
- Use cases can be ‘recycled’ for gaining requirements, for testing of the standard, for first proofs of concepts, and for demonstrators.
- Project results should be published as soon as they are available to boost the adoption of ACI by non-ACOSAR members.
- The process of transferring the project results to another entity after completion of the project should be started well before the end of the ACOSAR project; this accounts for the time needed to create the additional (follow-up) infrastructure.
- Tools to test the adherence to ACOSAR/ACI specifications should be available as ‘open source’ so potential suppliers of tools can test their implementation and potential users can test the tools that suppliers are selling them.
- A demonstration example shall be available for testing the capabilities of ACI

5 Overview of State of Practice and Used Tools

5.1 Introduction

The idea of this chapter is to

- get an overview of the state of practice by looking at the tools the partners commonly use and the tools the partners distribute
- learn how the partners use specific tools in the context of systems engineering and systems integration
- collect important information for the future specification of the ACI
- specifically collect information on the use, the method, advantages and the relevance for the ACOSAR project
- to get a good picture of the current state of practice by looking at the common use of the different tools

For this purpose all ACOSAR partners were asked to provide information on their most important tools (either in self-use or in distribution). This information is presented hereafter in tabular form.

5.2 Used Tools

5.2.1 ISOLAR-EVE

Partner Name	Bosch
Summarized by	Isidro Corral, Oliver Kotte (Bosch)
Tool Description	ISOLAR-EVE ETAS, http://www.etas.com Virtual ECU for Software Development
Tool Use	This tool enables testing the behavior of controller functions generated e.g. in ASCET or Simulink in a virtual ECU. Hence, with this tool we integrate every software component of a virtual ECU, i.e. controller functions inside the application software, the operating system, the real time environment, etc. We generate the virtual ECU itself and a FMU of the virtual ECU. We integrate the FMU of the virtual ECU into a simulation tool, e.g. ASCET or Simulink, and run the virtual ECU together with a plant model for SiL tests.
Method	AUTOSAR conform code generated from the controller functions in ASCET or Simulink are imported with ISOLAR-EVE along with the ECU architecture (.arxml) files generated with ISOLAR-A. ISOLAR-EVE integrates these files to generate a virtual ECU for SiL testing.
Advantages	ETAS, the maker of the tool, is part of the consortium. Integration with other tools of the ETAS tool chain (for measurement and calibration, rest-bus simulation, automated testing).
Innovation potential	Support for ACU export of virtual controllers.
ACOSAR relevance	The tool could contribute in identifying the ACI SiL interface between the controller functions inside the ECU and the plant model. This information could be extracted from the .arxml-files generated for every "software entity" in the application SW concealing the controller functions. With integrated support for ACI, the tool would export virtual ACI-controllers that could be easily interchanged with hardware-ACI-controllers – the tool would then support a more seamless transition from SiL to HiL.

Partner Name	ETAS
Summarized by	Corina Mitrohin, Natarajan Nagarajan
Tool Description	ISOLAR-EVE – virtual ECU for Software Development
Tool Use	The ETAS Virtual ECU (Isolar EVE; Isolar stays for "**I**ntegrated **Sol**utions for **A**UTOSA**R**") is an open and expandable Eclipse-based tool environment, enabling the integration, configuration, and execution of ECU software on the PC. The key function of this solution is the generation of an ETAS Virtual ECU software and configuration, with the combination of actual ECU application software. This permits a PC-based behavioural simulation of a complete ECU within a Windows and Linux environment. http://www.etas.com/en/products/isolar_eve.php
Method	Function models, application software components and basic software modules from a variety of vendors can be integrated and tested. Different configurations of the basic software can be validated separately as standalone, as well as in interaction with the application software. In the course of development, the software can be validated in both an offline simulation and a real-time simulation on the PC: • Offline Simulation: The offline simulation provides an early means of testing the interaction of the application software components at the level of the AUTOSAR Virtual Function Bus. • Real-time Simulation: A real-time PC facilitates connections with actual sensors, actuators, and other ECUs by means of standard I/O cards. The ETAS Virtual ECU provides consistent support for established ECU interfaces – a capability that permits easy connections to existing tools. By re-using the testing artifacts and established methods, the ETAS Virtual ECU environment can be seamlessly integrated into existing development processes. The main use case of Isolar EVE lies in the Software in the Loop testing, with bridges to the "MiL" and "HiL" stages.
Advantages	• PC platform for ECU software validation at component, sub-system, or system level • Validate application software, basic software, and complete ECU software in a virtual environment • Run on the PC in virtual time (speed optimized or in user-defined speed) or in real-time • Integrate the virtual ECU in different environments, e.g. as an FMU (Functional Mockup Unit) or a Simulink® S-Function • Connect the virtual ECU to measurement and calibration tools, for example, INCA, via XCP • Connect the virtual ECU to tools for CAN bus simulation, such as CANoe or BUSMASTER • Automate testing, e.g. with ETAS RT2 or PikeTec TPT • MCAL (Microcontroller Abstraction Layer), OS (AUTOSAR operating system), and RTE (AUTOSAR Runtime-Environment) for the PC, based on production-proven embedded implementations of these components

	• User can configure MCAL implementations for virtual and real input and output interfaces on the PC • Deploy multiple virtual ECUs simultaneously • Openness and flexibility by using standards, such as AUTOSAR, Artop, Eclipse, XCP, and A2L
Innovation potential	The compliance with AUTOSAR, FMI, XCP, and further automotive standards delineates Isolar EVE as a necessary puzzle piece in the big ACOSAR picture, especially when it comes to production-ready control software and its coupling to plant models. The innovation potential relies in the area of standardized interfaces and a correct execution of production code, parallel to simulated plant models.
ACOSAR relevance	Isolar EVE is mainly covering the SiL use case and thus plays an important role latest for use case evaluation in ACOSAR (WP7) and the methodology work package (WP2). The combination of software, with its underlying discrete model of computation, with simulated continuous models, not necessary in the same tool environment but in a coupled scenario, will raise new challenges to assure the correctness of simulation results.

5.2.2 SCALEXIO

Partner Name	dSPACE
Summarized by	Steffen Beringer (dSPACE)
Tool Description	SCALEXIO® dSPACE, http://www.dspace.de
Tool Use	dSPACE SCALEXIO is a very versatile HiL simulator that provides highly flexible channels, can be extended to any required size and is completely software-configurable. Its application range covers all test domains, including the test of ECUs of electric drives.
Method	The integration of a SCALEXIO HiL System is entirely configured by the dSPACE software ConfigurationDesk®.
Advantages	The key benefits of SCALEXIO are: • Support of different workflows and user roles by separating I/O configuration, modelling and code generation. • Test of different ECU variants and types on a single system with minimal configuration effort. • Easily resizable to fit specific test tasks because component test systems and network systems are both built with the same standardized hardware components and connections. • Graphical configuration of I/O channels. • Use of virtual ECUs for HIL tests if the real ECU prototype is not available yet • Support of FMI.
Innovation potential	Refer to ACOSAR relevance.
ACOSAR relevance	With usage of the ACI a SCALEXIO HiL platform would achieve integration possibilities in heterogeneous test environments via a standard interface.

5.2.3 VEOS

Partner Name	dSPACE
Summarized by	Nicolas Amringer (dSPACE)
Tool Description	VEOS® dSPACE GmbH, http://www.dspace.de VEOS is a PC-based simulation platform for validating the software of ECUs in early development process stages.
Tool Use	The PC-based simulation platform VEOS enables function developers, software architects and testers to validate software of ECUs in early development process stages. Hereby VEOS works hand in hand with other dSPACE products to provide a complete tool chain for the development and testing process. This means that tools and models which are commonly used in RCP and HiL simulations can also be used in the virtual world. Similarly, layouts from HiL simulation can be reused in PC-based simulation with VEOS and vice versa.
Method	Tools and models which are used with VEOS can also be used with real-time systems like RCP or HiL systems.
Advantages	The key benefits of VEOS are: • PC-based simulation of models and the ECU network communication. • Openness through support of significant standards like AUTOSAR and FMI. • Seamless integration with RCP and HiL tool chains. • Open interfaces to connect and utilize existing tools. • Simulation of a wide range of different models – function models, FMUs, virtual ECUs, and vehicle models. • Importing, connecting and running any number of functions and plant models based on Simulink or FMI in multi-model scenarios.
Innovation potential	With the prototypical ACI implementation for VEOS, coupling of non-RT and RT-system can be addressed in early stages of the development/test process.
ACOSAR relevance	One of the major goals of ACOSAR is the coupling of non-RT and RT systems. As VEOS is a non-RT-simulator, dSPACE will contribute to the project with a prototypical ACI implementation for VEOS. With the prototypical ACI implementation for VEOS, coupling of non-RT and RT-system can be addressed in early stages of the development/test process. VEOS is a dSPACE tool; therefore it is possible for dSPACE to implement ACI specification quickly and to evaluate different ACI implementations. With ACOSAR the application areas of VEOS could be extended and new customer groups could be addressed.

5.2.4 ControlDesk Next Generation & ConfigurationDesk

Partner Name	Renault, dSPACE
Summarized by	Jean-Marie QUELIN (Renault)

Tool Description	ControlDesk Next Generation / Configuration Desk from dSPACE For configuring HIL benches two tools are needed: ConfigurationDesk (for new platform SCALEXIO) and ControlDesk Next Generation. For operating the HIL bench, ControlDesk Next Generation is used. For a full description of those two tools, see DSP2016 and DSP2016a
Tool Use	**ConfigurationDesk for SCALEXIO** Used to create the plant model, create bench configuration (assign I/O, link I/O to model I/O, CAN, …) **ControlDesk NG** Used to operate the HIL benches. ControlDesk can be seen as the HMI between the operator and the bench.
Method	Those tools are mandatory for preparing and operating dSPACE HIL benches. There is no alternative
Advantages	The tool (ConfigurationDesk, ControlDesk Next Generation) are designed to give the best benefit of dSPACE HIL. As an end-user, it's not possible to find advantage as there is no alternative for using dSPACE HILs
ACOSAR relevance	ControlDesk Next Generation, ConfigurationDesk are the two major tools for creating and using test environment on HIL benches. As ACOSAR intent is to couple RealTime system, those two tools should be compatible with the specifications of ACI

5.2.5 AMESim

Partner Name	Siemens Industry Software
Summarized by	Jean-Marie Quelin (Renault), Pacome Magnin (Siemens)
Tool Description	Siemens LMS Imagine Lab Amesim Siemens Industry Software, http://www.plm.automation.siemens.com/
Tool Use	LMS Imagine.Lab Amesim software offers engineers an integrated simulation platform to accurately predict the multi-domain performance of intelligent systems. LMS Amesim enables engineers to model, simulate and analyse multi-domain, controlled systems, and offers capabilities to connect to controls design, helping to assess and validate control strategies.
Method	Plant models can be coupled with control systems through various interfaces, and exported to various real-time targets like dSPACE, xPC Target, RTLab, Labcar, A&D, NI LabVIEW among others through Simulink coder or through Functional mock-up interfaces. Dedicated interfaces allow tight coupling with various other software as well as fmi (1.0 & 2.0) import /export for both model exchange and cosimulation. For Renault: AMESim is used for MIL validations and HIL validations. AMESim is used for upstream project and application project.

	AMESim is used for modelling system like engine, transmission, vehicle, batteries, etc, not ECU strategies. AMESim is coupled with Simulink to build complete model for MIL and HIL
Advantages	Thanks to its 38 libraries and more than 4,500 ready-to-use components, LMS Amesim provides a great scalability in the main physical domains (fluids, thermodynamics, electrics, electro-mechanical, mechanics and signal processing) as well as application libraries (cooling system, air-conditioning, internal combustion engine, and aerospace). These modelling capabilities can be extended using some additional software to code one's own components to address specific modelling/design. Dedicated analysis tools allow getting better understanding of designed systems, detecting CPU traps and accurate analysis of embedded dynamics. LMS Amesim thus allow users designing accurate and high fidelity model to get deep understanding of model behaviour but also ease simplification process to target RT platforms, ensuring a continuous model handling through the V-cycle conception scheme. At Renault, AMESim models are a part of a complete plant model for HIL validation. HIL system are from dSPACE (PHS, SCALEXIO in a near future)
Innovation potential	LMS Amesim being an integration platform, numerous developed interfaces with 3rd party software were developed on demand of our customers, as well as specific export to RT targets. The experience brought by these developments and projects allowed developing an expertise on software coupling, the related traps and issues, best practices, etc. In the same time Siemens PLM took part in various recent R&D project like Modelisar, Modrio or Agesys, and is also part of the FMI steering committee as well as Modelica association, participating to FMI standard definition and Modelica language evolution. Bringing results of these projects in LMS Amesim allows providing our customers the best tools and practices corresponding to their needs, reconciling innovation and user's daily work.
ACOSAR relevance	Siemens Industry Software will implement ACOSAR outputs in his tools, allowing in a first time, project partners using Amesim to test and validate prototype of ACOSAR defined interface during the project. Industrialization of such interfaces will then be processed in our product. AMESim models can be compiled for real-time target. As Renault plant model are assemblies of AMESim / Simulink, ACOSAR should allow the connection of a 'AMESim' real-time plant model to an another real-time plant model like dSPACE (based on Simulink). Potential benefits of the ACI specification: • To use same model from MIL to HIL • Simplify the deployment of AMESim thru the company It is further important to keep close contact to the AMESim developers.

5.2.6 Dymola

Partner Name	TWT
Summarized by	Michael Völker, Leonid Lichtenstein
Tool Description	Dymola, Dassault Systèmes [DYM2016] Dymola is a commercial development environment for the Modelica modelling language.
Tool Use	At TWT GmbH Dymola is used for the development of 1D models for complex cyber physical systems. In the Modelica modelling language, a model is characterized by a set of symbolic equations (algebraic, differential or discrete), as well a set of initial parameters. We use Dymola to study and optimize complex mechanical, thermal and electric systems, both in stand-alone simulations and in Co-Simulations via the FMU interface.
Method	Dymola provides support for real-time simulation using the Dymola-Simulink interface in connection with Matlab Real-Time Workshop and an additional RealtimeSim or SourceCodeGeneration license option.
Advantages	In the Modelica modelling language, component interfaces and equations are bi-directional and thus provide native feedback and self-consistency throughout large circuits. Fundamental conservation laws are held and there is an extensive library of basic blocks, physical components and mathematical tools. External C-Code can be included and the final model can be compiled both as an S-function or a stand-alone FMU.
Innovation potential	Dymola provides support for real-time simulation on only a limited number of platforms, mostly in connection with Simulink.
ACOSAR relevance	Dymola is a widely-used tool for systems engineering and has a big community. It is important to consider it for possible usage of the ACI standard including discussion with the tool vendor.

5.2.7 ADTF

Partner Name	TWT
Summarized by	Florian Ries, Leonid Lichtenstein (TWT)
Tool Description	ADTF, Elektrobit [ADT2016] ADTF means "automotive data- and time-triggered framework". It is a popular tool for rapid prototyping of advanced driver assistance systems (ADAS).
Tool Use	We use ADTF to evaluate driving assistance systems with recorded data offline, but also with the vehicle in the loop. Our focus is on environmental perception methods. Our customers use the tool in a similar way but covering more applications and larger ranges of tests.
Method	ADTF operates in real time, and also allows fractions/multiples thereof. It can retrieve and produce real-time vehicle data and is also designed to interact with other real-time processes.
Advantages	ADTF allows retrieving, storing and reproducing real vehicle data, and even allows live field tests with software or model prototypes.
Innovation potential	ADTF could be improved by allowing more modularity, stability and more tools for debugging.

ACOSAR relevance	ACOSAR should be usable with all kinds of automotive software. ACOSAR is mainly focused on engine test benches with the current consortium. However, ADAS is a very – and increasingly – important field of automotive software that should not be neglected. ACI would provide a standardized way to connect ADTF to all kinds of modelling and simulation software, as well as hardware prototypes, which so far do not exist. In this context, ADTF opens the way to including ADAS and real vehicles in the Co-Simulation. Our planned contribution of making ADTF able to retrieve ACI data packages will actually improve on the tool itself.

5.2.8 Simulink

Partner Name	TWT
Summarized by	Jos Höll, Leonid Lichtenstein (TWT)
Tool Description	Matlab-Simulink, Mathworks [MSI2016] Simulink is a toolbox for modelling and simulating cyber physical systems based on Matlab.
Tool Use	At TWT Matlab-Simulink is mainly used in the research and development process. It can be used for modelling and simulating cyber physical systems. Our customers and we use Simulink to model complex systems. In a complex simulation of a cyber physical system Simulink can play different roles: • Modelling parts of the system or a complete model. • As the master of a co-simulation. Simulink can be linked with other tools using S-functions. • Simulink can also be used for evaluation, analysis and presentation of simulation results.
Method	The Real-Time Workshop is a Matlab/Simulink extension for real time simulations. The Simulink-Coder can be used to export Simulink models to other targets. Using TLC-files users (3rd parties) are able to specify code generation rules and additional functionalities for own platforms. One can also use the TWT Simulink FMU-Interface to connect Simulink models to a Co-Simulation.
Advantages	Mathworks tools are used very intensively by many stakeholders. So there is a large community, a good support and a detailed documentation. Mathworks offers a large spectrum of products and extensions. Simulink works very well together with them.
Innovation potential	The wide range of Mathworks products opens many possibilities but the user has to buy all the products and toolboxes separately. Further, it is not straightforward to connect Simulink to other tools or even to export its models to an FMU.
ACOSAR relevance	Simulink is an established tool for systems engineering and co-simulation. As soon as Simulink supports the ACI with an implementation thereof, many existing models can be connected to other models and to hardware with real-time requirements. Simulink will be used in multiple use cases. If a prototypical implementation of an ACU can be shown to work in Simulink, the advantages of ACOSAR will become clear to industrial users.

If Simulink supports the ACI, it can be used for more applications than currently, e.g. HIL.

5.2.9 TWT CoSimLab

Partner Name	TWT
Summarized by	Jos Höll, Leonid Lichtenstein (TWT)
Tool Description	CoSimLab, TWT GmbH [TCS2016] The tool represents TWT's co-simulation framework.
Tool Use	TWT and its customers use the co-simulation framework to link different models and master co-simulation. Therefore many different types of models can be connected to the CoSimLab. We use the TWT Co-Simulation GUI to view the routed signals of all connected models and for configuration purposes. After and during the simulation, the GUI visualizes the simulation results intuitively.
Method	CoSimLab is used for system simulation purposes. One can test the model to be exported and verify its functionality.
Advantages	The main capabilities of the CoSimLab are • connect FMUs using the FMI standard, • connect external tools for simulation such as Matlab/Simulink, StarCCM+, Dymola/Modelica and Qucs, OpenFOAM, • connect the TWT Functional Mock-up Trust Centre to protect FMUs, • simulate online via a network.
Innovation potential	Currently, the CoSimLab does not support real-time co-simulation or the connection to HiL.
ACOSAR relevance	TWT is frequently using its co-simulation platform and will try to adapt it sustainably to the ACI. The experience with its co-simulation platform will help TWT to contribute to requirements for the ACI. The tool will be able to cover new areas, e.g. (distributed) real-time co-simulation, HIL, etc. As it is a TWT product, changes in the ACI can be quickly tested and implemented. We think that potential ACOSAR developments can substantially improve TWT's CoSimLab, e.g. the usage scope of the software can be extended and new customers addressed.

5.2.10 Enterprise Architect

Partner Name	Virtual Vehicle
Summarized by	Martin Krammer (Virtual Vehicle)
Tool Description	Enterprise Architect (EA) SparxSystems, http://sparxsystems.com/ In Europe represented by SparxSystems Central Europe, and LieberLieber GmbH, http://www.lieberlieber.com/ EA is basically a software development tool supporting the UML standard for a very long time now. Over the years, it has become a versatile system

	and process development tool. It is extendable and features an API, as well as model driven generation (MDG) technologies.
Tool Use	We use the tool at the following stages: • Requirements Management (structure, dependencies, analysis) • Preliminary System Architecture (all kinds of SysML diagrams) • System Design • Safety Concept (SysML, SPEM) VIF uses the tool for concept- and system-development. EA is primarily used: • in R&D projects • for prototype developments
Method	Until now, EA has been used for system designs and the generation of co-simulation configurations. The latter includes offline co-simulations and supports the FMI standard. The description of RT systems using SysML/UML concepts is well understood and requires a holistic approach.
Advantages	EA seems advantageous for us, because it • has affordable licenses • is extensible • can be adapted to own needs • is well documented and has a lively user community
Innovation potential	EA is a versatile tool, but it needs to be adapted to own needs. This requires some efforts for planning, implementation, testing, etc.
ACOSAR relevance	EA can be used in two ways in ACOSAR: (1) ACI development and (2) Configuration of simulation using ACI. (1) ACI development *Requirements engineering* Besides using spreadsheets in WP1, EA is a good alternative when it comes to requirements structuring and introduction of relationships. *ACI specification* The turn from requirements to specifications is often difficult. With EA this shift may be achieved efficiently. Requirements traceability could be helpful, but also code generation features of EA. *Safety Management* The ACI shall support safety features to ensure correct operation and to avoid failures and damages of connected physical systems. EA helps to analyse and structure the safety related information. Implementations regarding FMI concepts and specifications are already available for EA. The next step could be an implementation of the ACI. This would strengthen the position of EA as a system integration tool. ACOSAR can benefit through semi-formal and formal specifications of use cases, requirements and the ACI. This leads to more accurate specifications, avoiding systematic failures at early and later stages of development; and eases communication between different project teams and individuals. Furthermore, demonstrator applications may be built fast and efficiently.

(2) Configuration of co-simulation using ACI

RT system integration

EA provides ready to use implementations of SysML and offers extension capabilities, e.g. profiling mechanisms. The introduction of ACOSAR syntax and semantics to EA enables the integration of RT systems, and allows the representation of integration concepts and early implementations thereof.

5.2.11 Labcar

Partner Name	ETAS
Summarized by	Corina Mitrohin, ETAS
Tool Description	LABCAR Software Products http://www.etas.com/en/products/labcar_software_products.php http://www.etas.com/download-center-files/products_LABCAR_System_Components/ETAS_LABCAR_Flyer_EN.pdf The use of Hardware-in-the-Loop (HiL) systems requires software products for various purposes. These include configuration and administration of the HiL system, real-time computation on simulation targets, operation during manual testing, and execution of automated tests. LABCAR software is easy-to-use and reliable, increasing the user's efficiency. The LABCAR product family comprises the following products: - LABCAR-OPERATOR The LABCAR-OPERATOR software serves as the user interface for LABCAR – not only for the pre-configuration procedures but also for experiment design and execution. LABCAR operating software is running on every Microsoft Windows® standard PC. - LABCAR-RTPC (Real-Time PC) The PC-based simulation target calculates the models in real time. - LABCAR-AUTOMATION The LABCAR-AUTOMATION software provides for extensive test automation. It facilitates the economical and reproducible testing of safety-critical or time-intensive tests, such as OBD tests, e.g., in situations calling for the fulfillment of product liability requirements. The modular LABCAR tool concept allows tailoring and adaptation of the HiL system to specific needs. Open interfaces enable the integration with various other tools, for example for test design, modelling, measurement & calibration and diagnostics.
Tool Use	The tool is used in the ECU control software development (prototype development), verification and validation of electronic components (right half of the V-Cycle). Customer’s use-case: 1. System development 2. V&V activities 3. Test environment development 4. Test beds and control development

Method	The tool is used for integration, configuration, execution and simulation of native simulation models, FMUs and communication protocols in an ETAS real-time system (ETAS-RTPC).
Advantages	Open, modular system architecture for simulation models, software, hardware, test-automation and ECU access.
Innovation potential	Implementation of standardized interfaces (at simulation tool, communication and hardware level) in LABCAR components will make the entire HiL system compatible with solutions from different vendors. On customer side, existent HiL systems will become extensible, thus serving more complex use cases.
ACOSAR relevance	The LABCAR software products together with the appropriate hardware solutions constitute ETAS' HiL system, well-established and successful on the market. The relevance for ACOSAR is twofold: - make LABCAR software products compatible with other tools and solutions existent on the market, via standardized interfaces - make LABACR software products compatible and part of future XiL solution (Model-, Software- and Hardware in the Loop) The implementation of ACI in the affected LABCAR components will open and make ETAS' products compatible to other solutions from project partners. The build-up of e.g. network HiL will be possible without extra engineering effort, but in a plug-and-play manner. The usage of LABCAR software products offers alternatives or even round the use cases which are considered in ACOSAR. The concentrated effort in ACOSAR with partners coming with different perspectives, will help to reveal and close existent gaps when it comes to serve complex real-time applications. A simplified illustration of the benefit we will be achieving via the ACOSAR developments is the following. If prior to ACOSAR, LABCAR solutions were able to test and validate a system A, while another solution from another ACOSAR partner, let's denote it X, was used for the analysis of a system B, post ACOSAR the integrative analysis of A and B will be possible, provided that LABCAR and X are ACI-compliant.

5.2.12 SimulationX

Partner Name	ITI
Summarized by	Torsten Blochwitz (ITI)
Tool Description	SimulationX ITI GmbH www.itisim.com SimulationX is a multi-domain system simulation tool. It supports the modelling language Modelica and FMI as exchange format for models and co-simulation.
Tool Use	We (ITI) are the developer of SimulationX. Our customers and ITI's engineering department use it for modelling and simulation of technical systems. Model libraries provided by ITI, the Modelica Standard Library, 3rd party libraries or the Modelica language are used for efficient modelling. The tool is applied in various stages of the design cycle: for early system

	design, analysis and optimization of components and systems, for development and test of control algorithms. SimulationX models can be exported as platform independent C-Code for various targets (Simulink S-functions, FMI for Model Exchange and Co-Simulation, ETAS Labcar, NI VeriStand, …). In this way the models can run in dedicated SiL or HiL platforms. SimulationX supports co-simulation via FMI or via specific network based interfaces to other tools.
Method	Real-time simulations with higher requirements regarding real time capabilities and determinism are addressed by code-export: the model (with or without a real-time capable fixed step solver) are exported as platform independent ready to use C-code which is compiled for the used real-time platform (for example dSPACE Scalexio, ETAS Labcar or NI VeriStand). Depending on the capabilities FMI or a target specific API is used for integration to those models. For lower requirements the simulation run inside SimulationX can be carried out synchronous to real time. This feature together with the SimulationX interfaces to automation systems is used for run up and test of PLC-devices (programmable logic controllers, for example Siemens S7 systems). Of course, for such real-time applications, models need to be real-time capable. That means the levels of complexity, dynamics and non-linearity have to allow an execution in a known run-time which is smaller or at least equal to real-time. SimulationX provides a number of performance analysis tools which support users in developing real-time capable models.
Advantages	SimulationX comes with a bunch of user friendly ready to use and industry proven model libraries for various physical domains (1D..3D). In addition to that the object oriented modelling language Modelica can be used to develop own models. ITI's engineering departments supports users in solving their analysis tasks. SimulationX is an open environment. By means of the object oriented modelling language Modelica, model libraries can easily be enhanced or written from scratch. SimulationX supports FMI 1.0 and 2.0 import and export which is a tool neutral interface for model exchange and co-simulation. Beside of simulation in time domain the tool provides analyses in frequency domain as well as several linear system analyses methods. Tracing and performance analyses help users debugging and optimizing their models.
Innovation potential	ITI participated in the Modelisar project [MOD2011] and was one of the main contributors to the FMI specification. SimulationX was one of the first simulation platforms with full FMI support. These examples show the innovation capabilities of ITI. SimulationX is an open tool, it supports Modelica as modelling language and the flexible External Function/Object interface for integration of external code. The innovation potential of SimulationX consists in the capability to add new extensions and interfaces in a fast way without recompilation of the whole tool. In this way a fast prototype implementation of ACI will be possible.
ACOSAR relevance	ITI participates in ACOSAR. SimulationX will implement the developed interfaces prototypically. SimulationX is used in one of the ACOSAR

demonstrators. SimulationX will be improved by ACOSAR due a closer connection to real time systems and better connectivity to other tools.

5.2.13 AVL Cruise

Partner Name	AVL
Summarized by	Josef Zehetner
Tool Description	AVL CRUISE™, developed at AVL List GmbH AVL CRUISE™ is a simulation tool targeting fuel efficiency, driving emissions or performance analysis simulation in the vehicle driveline development process, as well as for office and real-time applications.
Tool Use	AVL CRUISE™ currently is sold by AVL as a Vehicle System and Driveline Analysis tool. At AVL PTE (powertrain engineering) department AVL CRUISE™ is used to driveline and system simulation in customer projects. AVL ITS (instrumentation and test systems) uses AVL CRUISE™ to perform driveline simulation at the AVL testbeds. AVL CRUISE is typically used for simulation in following applications: Aftertreatment, Electrification, Energy Management, Function Development and Calibration, Model Based Testing, Thermal Management, Transmission and Driveline, Vehicle System Simulation AVL CRUISE is used by customers from OEM level down to tier 1 & 2 to perform above mentioned simulation applications.
Method	AVL CRUISE can either be provide real-time models for real-time systems (e.g. AVL ARTE, ETAS, dSpace, …) or can be coupled directly to real-time systems via Model.CONNECT.
Advantages	AVL CRUISE™ offers a streamlined workflow for all kinds of parameter optimization and component matching – guiding the user along the way to practical and attainable solutions. Due to its structured interfaces and advanced data management AVL CRUISE™ has established itself as a data communication and integration tool for various teams within world-leading OEMs and their suppliers. This facilitates consistent target definition and traceability of the decisions made in reaching the best overall results for the developed product.
Innovation potential	AVL CRUISE and its successor AVL CRUISE M are strongly going into the direction of multi-domain simulation, with highly specific component libraries, like Powertrain, Engine or Thermal.
ACOSAR relevance	AVL CRUISE is a tool which provides real-time capable simulation models for uses cases within ACOSAR. The ACI standard could simplify the interaction between AVL CRUISE and real-time systems and/or real-time simulation due to the standardized interface. The use cases described in ACOSAR can use component simulation models from AVL CRUISE.

The implementation of an ACI demonstrator with AVL CRUISE shows the real-time capability of the component models in CRUISE and therefore strengthen its position on the market.

5.2.14 AVL Model.CONNECT

Partner Name	AVL
Summarized by	Josef Zehetner
Tool Description	Model.CONNECT™, developed at AVL List GmbH Model.CONNECT™ is a co-simulation environment to enable all future model based development approaches. It brings together office simulation (tools and models) and real-world testing. Model.CONNECT™ contains the advanced co-simulation execution engine ICOS (Independent CO-Simulation, developed at VIRTUAL VEHICLE, www.v2c2.at) and ACoRTA (Advanced Co-Simulation Methods for Real-Time Application), which solves the complex interaction between your virtual and real components with advanced coupling algorithms. • Couple different simulation models from different tools which you already use • Model integration with HiLs and testbeds • Unique solver techniques (multi-rate) and minimizing coupling errors • Fully supporting interface standards and customized wrappers • More than 25 simulation tools for various domains supported (e.g. AVL CRUISE, AMESim, MATLAB, ECS Kuli,Dymola, MSC Adams, LS-DYNA)
Tool Use	Model.CONNECT™ currently is sold by AVL as a standalone integration tool for office- and real-time-simulation. At AVL PTE (powertrain engineering) department Model.CONNECT™ is used to perform complex system simulation tasks in customer projects. In AVL ITS (instrumentation and test systems) Model.CONNECT™ is used to bring advanced simulation models to the AVL testbeds. Typical use cases include integrated safety simulations, coupling of FEM for crash simulation with vehicle dynamics simulation and controller development, coupling of mechanical/electrical subsystems, coupling of electrochemical and thermal models, coupling of thermal, electrical and mechanical models for the development and optimization of energy management systems, or development and calibration of vehicle dynamics controller based on multi-body-simulations. Model.CONNECT is used by customers from OEM level down to tier 1 & 2 to perform co-simulation and real-time co-simulation applications. Research projects were conducted with Model.CONNECT being used on the hardware-level (integrated circuits) as well. Applications include all fields mentioned previously.
Method	*How does the tool relate to the integration of real-time systems?* Model.CONNECT contains interfaces and coupling methods to couple office simulation and real-time systems.

Advantages	Model.CONNECT acts as a system integration tool, for office and real-time. The tool itself does not degrade simulation performance, as the simulators run independently. Every domain specific engineering task can be solved within its own, specific tools.
Innovation potential	Model.CONNECT can currently integrate up to 25 different simulation tools, including FMI and real-time systems via CAN-bus and UDP/Ethernet via a proprietary interface. The ACI standard would support the integration along the path from MiL, SiL up to HiL during system development.
ACOSAR relevance	See innovation potential. The ACI standard simplifies the interaction between Model.CONNECT and real-time systems and/or real-time simulation due to the standardized interface. The use cases described in ACOSAR can instantly benefit from the availability of Model.CONNECT as a demonstrator platform. The addition and advantages of the ACI can easily be demonstrated. The implementation of an ACI demonstrator would highlight Model.CONNECT as a state-of-the-art simulation platform and strengthen its position on the market.

5.3 Summary and Conclusion

The gathered descriptions on the tools in use have revealed some important information for the ACOSAR project. First of all, it is useful to note the variety of usage domains of the tools in use:

- modelling of virtual ECUs
- development and validation of ECU software
- HIL simulators
- System modelling and simulation (mechanical, electric, thermal, etc.)
- development tools for advanced driver assistance systems
- co-simulation master / slave
- requirement management
- system architecture and design

The list of tools is also useful to get an overview of how these tools and the users thereof can benefit from the development of the ACI standard:

- extend the range of applications of a tool
- tools will benefit from a standard interface
- make tool RT capable
- couple RT and non-RT systems
- connecting two RT plants
- save time and effort during RT systems integration
- improve on compatibility to other tools
- perform integrative analysis of systems modelled in different tools
- improve tools due to experience in ACOSAR and the specifications of the ACI

Further, the ACOSAR project benefits from the experience of the partners with certain tools. This is especially relevant for:

- the development and the specification of the ACI
- prototypical implementation of the ACI in the different tools
- to test different ACI implementations
- to test the ACI specification
- for conducting the ACOSAR use cases

Overall, it is apparent that there are many different tools in use for modelling and simulation. However, there is not one single interface and not one single methodology for real time systems integration. The goal of ACOSAR is to create a substantial contribution to the community in this context by specifying an open advanced co-simulation interface and showing a time-saving systems integration methodology.

If all tool vendors, which are ACOSAR partners, implement ACI for their tools the usage for many fields of applications will be ensured. In addition to that at least test implementations as proof of concept for other commonly used tools (for example Matlab/Simulink) should be provided.

6 Discussion and Conclusion

The comprehensive list of related projects and standards show that there is an attempt to improve interoperability between tools and devices. Interoperability standards, such as the ASAM standards or FMI, want to improve and facilitate the development of products by providing a standard interface so that different tools and devices can communicate without high implementation efforts. Further, there are lots of projects that on the one hand have the goal to improve methods and tools and on the other hand use existing standards and validate their applicability. Existing co-simulation methods are advanced and provide a good basis for the verification of systems. There are co-simulation tools such as Model.CONNECT/ICOS that enable simulation of systems using different simulation tools and models. Many of the simulation tools support FMI for that purpose which is the de-facto standard for co-simulation and model exchange.

However, none of the existing standards and projects provides a generic ACI which enables co-simulation in RT. The existing standards describe RT system interfaces, simulation tool interfaces or RT co-simulation architecture/interfaces with focus on a special domain (e.g. military applications).

In contrast to that the goal of ACOSAR is to provide a standard interface for RT co-simulations which is generic and flexible. This ACI shall be usable for both simulation tools and RT devices. Further, it is necessary that the interface is not tailored for a special domain but is applicable in several ones. This brings some challenges for ACOSAR such as handling the different properties of simulation tools and RT systems, the coupling of different simulation models and RT systems in RT as well as the possibility for communication via various communication channels in RT.

However, the ACI shall provide the possibility to exchange data and to co-simulate devices in RT. FMI (cp. Section 4.7.2.5) as well as further state of the art literature (cp. Chapters 4.1 - 4.4) provide the basis for simulation tool interfaces as well as co-simulation methods and properties. They can be reused but have to be extended to meet RT requirements.

As RT devices support different kinds of communication channels, the ACI has to provide the possibility to use them in the co-simulation and requires therefore the abstraction of communication mediums. Further, the abstraction of communication targets at having minimal configuration efforts for using different communication mediums. Existing interoperability standards concerning RT devices (cp. Section 4.7) can be used as basis for this functionality. Nevertheless, they have to be adapted to meet co-simulation requirements. In addition to interoperability standards, the communication standards described in Chapter 4.5 give information about properties of communication mediums that have to be considered in ACOSAR.

Extension of standard with advanced methods (e.g. coupling methods) shall be possible. Existing architectures such as HLA can provide input for that.

Beside the technical implementation of ACI, its application including the configuration of a co-simulation scenario and guidelines and processes how to use it, are needed, in order to establish a practical framework. For the specification and configuration of co-simulation scenarios, existing well-established modelling languages can be used (cp. Chapter 4.8). Moreover, projects that include co-simulation methods and tools (cp. Chapter 4.9) can give information about challenges, practical use cases and needed applications. Further, organizations such as ProStep provide practical guidelines and processes for using new methods and standards (e.g. workflow for using FMI).

In addition to the State-of-the-Art, current practices applied by project partners (cp. Chapter 3) offer valuable clues to ACOSAR. Partners offer or use several simulation tools which to some extent have RT capabilities. Nevertheless, improvement of these capabilities and a standardization of co-simulation in RT, such as FMI for simulation tools, are desirable.

Concluding the analysis of SoA and SoP, the following recommendations for the ACOSAR project can be given.

WP2 – RT system integration methodology

The RT system integration methodology should provide the methodical basis for integrating RT systems into co-simulation. This means that process, methods and best practices for using the ACI are specified. D1.1 summarized existing projects and standards that contain similar integration methods which can be used as basis for WP2.

WP3 – Simulation tool interface

The ACOSAR simulation tool interface should be based on FMI as the current de-facto standard for co-simulation. However, some improvements and extensions are necessary regarding the coupling of RT systems. RT requirements and corresponding coupling methods have to be part of ACI. Results of existing papers and projects, such as the ACoRTA projects, can provide input for this purpose.

WP4 – RT System interface

Standardized RT system interfaces already exist, but these standards have various purposes. The ACI interface for RT systems has to provide functions needed for co-simulation, including controlling/ monitoring functions that guarantee that system failures are avoided. Nevertheless, some parts or rather ideas can be taken from existing work and adapted for co-simulation.

WP5- Communication System Protocol

The main goal of WP5 is to provide the possibility to use various communication mediums for exchanging data in co-simulations. This requires an abstraction of communication mediums. Existing standards can give information about abstraction methods, and communication standards (cp. Section 2.5) provide requirements for communication.

WP6 – Advanced Co-simulation Interface

Existing standardization projects and standards, that ACI has to comply with, are described in this deliverable. Moreover, there are methods and practices that can be reused from existing standards. WP6 should consider these contributions for the ACI in order to become an accepted standard in the field of co-simulation.

WP7 – Application Use Cases and Assessment

In WP7 the ACI will be applied in real Use Cases. For the Use Cases and mainly the assessment of the ACI, ACOSAR should adopt assessment methods of related projects. Further, a comparison of the actual state in RT simulations (e.g. practices described in Chapter 3) and the state when using the ACI could be reasonable.

7 Glossary

Acronym	Full definition
A2L	Address to label
ACI	Advanced Cosimuation Interfaces. The interface that ACOSAR is proposing to standardize.
ACoRTA	Advanced Co-Simulation Methods for Real-Time Applications
ACOSAR	Advanded Co-simulation Open System Architecture
ACU	Advanced Co-Simulation Unit
ADAS	advanced driver assistance systems
ADTF	automotive data- and time-triggered framework
API	Application Programming Interface
ASAM	Association for Standardization of Automation and Measuring Systems (ASAM e.V.)
ASAM MCD-2-MC	Description format of the internal ECU variables used in measurement and calibration
AUTOSAR	Automotive Open System Architecture
AVANTI	Test methodology for virtual commissioning based on behaviour simulation of production systems
CAN	Controller Area Network
CAx	Computer-aided technologies
CiA	CAN in Automation
CORBA	Common Object Request Broker Architecture
CoSimLab	TWT’s co-simulation framework
COSSIM	A Novel, Comprehensible, Ultra-Fast, Security-Aware Cyber Physical Systems Simulator
CPS	Cyber-Physical System
CPSE Labs	Cyber-Physical Systems Engineering Labs
CPU	Central Processing Unit
CRF	Centro Ricerche FIAT
CRYSTAL	Crititcal sYSTem engineering AcceLeration
DACCOSIM	Distributed Architecture for Controlled CO-SIMulation
DARPA	Defense Advanced Research Projects Agency
DDS	Data Distribution service
DFC	Device Configuration File
DIS	Distributed Interactive Simulation
DoD	Department of Defence
DSS	Distributed Simulation Systems

ECU	Electronic Control Unit
EDS	Electronic Datasheet
EMIC	European Microsoft Innovation Center
ETG	EtherCAT Technology Group
EtherCAT	Ethernet for Control Automtion Technology
FMI	Functional Mockup Interface
FMU	Functional Mock-up Unit
GUI	Graphical user interface
HiL	Hardware in the Loop, stage in the system development cycle (V cycle)
HLA	High Level Architecture
HMI	Human Machine Interface
HTTP	Hypertext Transer Protocol
I/O	Input/Output
ICOS	Independent Co-Simulation
IDL	Interface Definition Language
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPC	Inter-Process Communication
ISO	International Oranization for Standardization
LABCAR	Software and Harware solztion for Hardware in the Loop applications
LAN	Local Area Network
LIN	Local Interconnect Network
MAPort	Model Access Port
MC	Measurement and Calibration
MCD	Measurement and Calibration Data
MDG	Model driven generation
MiL	Model-in-the-Loop
NEFZ	Neuer Europäischer Fahrzyklus
non-RT	Non-Real-Time
ODE	Ordinary Differential Equation
OEM	Original Equipment Manufacturer
OMG	Object Management Group
OPC	Open Platform Communications
OpenFOAM	Open source Field Operation And Manipulation
OS	operating system
OSLC	Open Services for Lifecycle Collaboration
PC	Personal Computer

PDO	Process Data Object
RCP	Rapid Control Prototyping
RDF	Resource Description Framework
REST	Representational State Transfer
RT	Real Time
RTI	runtime infrastructure
SCALEXIO	Name of latest dSPACE platfom for HIL
SiL	Software-in-the-Loop
SISO	Simulation Interoperability Standards Organization
SME	Small Medium Enterprises
SPI	Serial Peripheral Interface
SysML	Systems Modelling Language
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TLC	target language compiler
TRL	Technology Readiness Level
UDP	User Datagram Protocol
UML	Unified Modelling Language
URI	Uniform Resource Identifier
USB	Universal Serial Bus
VeTeSS	Verification and Test to Support Functional Safety Standards
WAN	Wide Area Network
WP	Work Package
XCP	eXtended Communication Protocol
XIL	X-in-the-Loop
XML	Extensible Markup Language

8 References

Identifier	Full Reference
ACO2016	ACORTA project, www.acorta.info (08/03/2016)
ADT2016	Elektrobit's ADTF page (2016). Retrieved from https://www.elektrobit.com/products/eb-assist/adtf/
AML2015	N. Schmidt and A. Lüder, "AutomationML in a Nutshell", AutomationML e.V., 2015
AML2016a	AutomationML e.V., "AutomatinML", Retrieved from: https://www.automationml.org (08/03/2016)
ASA2015	ASAM XCP Protocol, http://www.asam.net/nc/home/standards/standard-detail.html?tx_rbwbmasamstandards_pi1[showUid]=3144
ASAM2016	http://www.asam.net/nc/home/standards/standard-detail.html?tx_rbwbmasamstandards_pi1%5BshowUid%5D=2564&start= , visited on 22.03.2016
AUT2013	AUTOSAR - The Global Automotive E/E Standard. (2013, October 1). http://www.autosar.org/fileadmin/files/presentations/AUTOSAR_IOA_2013-10-23.pdf (accessed 2016, March 8)
AUT2014	Current State / Future Trends. (2014, August 1). http://www.autosar.org/fileadmin/files/presentations/AUTOSAR_Seminar_Mentor_20140801.pdf (accessed 2016, March 8)
AUT2016	Offical AUTOSAR website. (2016, March 7). http://www.autosar.org (accessed 2016, March 8)
AVA2015	AVANTI Consortium (2015). Poster on ITEA Co-summit. Retrieved from https://itea3.org/project/poster/download/453/AVANTI Posters Co-summit 2015.pdf
AVA2016	AVANTI Consortium (2016). Homepage. Retrieved from http://avanti-project.de/
AVA2016a	AVANTI Consortium (2016). ITEA project page. Retrieved from https://itea3.org/project/avanti.html
BAT2014	MBAT project, http://www.mbat-artemis.eu/ (08/03/2016)
BOE2005	Csaba Attila Boer, Distributed Simulation in Industry, http://repub.eur.nl/pub/6925, 2005
BRC2016	Broadcom Corporation, Retrieved from: https://www.broadcom.com/
BRR2016	BroadR Reach Standard, Retrieved from: http://www.ieee802.org/3/1TPCESG/public/BroadR_Reach_Automotive_Spec_V3.0.pdf
BRS2014	C. Brückner and B. Swynnerton, "Busbasiertes Architekturkonzept für Hardware-in-the-Loop-Prüfstände", Springer Verlag, 2014
Bus01	Gevatter Grünhaupt: Handbuch der Mess- und Automatisierungstechnik. 2nd Edition. Springer Berlin Heidelberg, 2006, S. 552
Bus02	https://www.samson.de/pdf_de/l453de.pdf
Bus03	http://www.profibus.com/uploads/media/PROFIBUS_Planning_8012_V10_Aug09.pdf

Bus04	http://www2.htw-dresden.de/~huhle/micros/MC-Docs/profibus.pdf
BUT2011	Giorgio Butazzo, Hard Real-Time Computing Systems, 2011, 3rd edition, http://www.springer.com/us/book/9781461406754
COS2016	COSSIM Consortium (2016). Retrieved from http://www.cossim.org/
CPS2016	CPSELabs, http://www.cpse-labs.eu/ (08/03/2016)
CRY2016	CRYSTAL project, http://www.crystal-artemis.eu (08/03/2016)
DAH1997	J. Dahmann et al.: "THE DEPARTMENT OF DEFENSE HIGH LEVEL ARCHITECTURE", 1997
DAH1999	J. Dahmann et al.: "THE DoD HIGH LEVEL ARCHITECTURE: AN UPDATE", 1999
DSP2016	Description of ControlDesk Next Generation (2016, March 8). Retrieved from https://www.dspace.com/en/inc/home/products/sw/experimentandvisualization/controldesk.cfm
DSP2016a	Description of ConfigurationDesk (2016, March 8). Retrieved from https://www.dspace.com/en/inc/home/products/sw/impsw/scalexio_configurationdesk.cfm
DYM2016	Dymola Homepage (2016). Retrieved from https://www.3ds.com/products-services/catia/products/dymola
ECA2016	EtherCAT Consortium: https://www.ethercat.org
FMD2016	Development (2016, March 9th). Retrieved https://www.fmi-standard.org/development.
FMI2015	FMI Steering Committee (2015, July 15th): FMI Development Process And Communication Policy. Retrieved from: https://svn.fmi-standard.org/fmi/branches/public/docs/DevProcess/FMI_DevelopmentProcess_1.0.pdf
FMI2016	FMI Description (2016, March 9th). Retrieved https://www.fmi-standard.org/
FMT2016	FMI Support in Tools (2016, March 9th). Retrieved https://www.fmi-standard.org/tools
FRI2011	M. Friedrich, "Parallel Co-simulation for Mechatronic Systems", Phd thesis, TU München, 2011, ISBN 9783843903363
FUJ1998	R. Fujimoto and P. Hoare: "HLA RTI Performance in High Speed LAN Environments", 1998
FUJ2000	Richard M. Fujimoto, Parallel and Distributed Simulation Systems, http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471183830.html, 2000
GAV2015	Galtier, Virginie; Vialle, Stéphane; Dad, Cherifa; Tavella, Jean-Philippe; Lam-Yee-Mui, Jean-Philippe; Plessis, Gilles, FMI-based distributed multi-simulation with DACCOSIM , 2015
GHB2007	Gheorghe, Luiza; Bouchhima, Faouzi; Nicolescu, Gabriela; Boucheneb, Hanifa, A Formalization of global simulation Models for Continuous/Discrete Systems, Proceedings of the 2007 Summer

	Computer Simulation Conference, ISBN # 1-56555-316-0 , July15 -18, 2007
GKN2013	O. Graeser, B. Kumar, O. Niggemann, N. Moriz and A. Maier, "AutomationML as a basis for offline- and realtime-simulation: Planning, Simulation and Diagnosis of Automation Systems", Informatics in Control, Automation and Robotics, LNEE 174, 2013, pp.195-209
HOM2006	M. Hommel, "Parallelisierte Simulationsprozesse für virtuelles Prototyping in der Automobilindustrie", Phd thesis, TU Braunschweig, 2006
IEC2014	IEC, "IEC 61158-1:2014: Industrial communication networks - Fieldbus specifications - Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series", 2014, Retrieved from: https://webstore.iec.ch/publication/4624
ICP2016	INTO-CPS Consortium (2016). Homepage. Retrieved from http://into-cps.au.dk/
IEE2000	IEEE 1516-2000: "IEEE Standard for Modeling and Simulation High Level Architecture (HLA) - Framework and Rules", 2000
KOK2015	Jochen Köhler and Michael Kübler: FMI experience at ZF – Progress in MAP „System Structure and Parameterization," 2015, September 21st, presentation at the FMI User Meeting within the 11th International Modelica Conference 2015, Versailles, France. Retrieved from https://svn.fmi-standard.org/fmi/branches/public/docs/Modelica2015/FMI_User_Meeting_ModelicaConference2015.zip
KOP2008	Kopetz, Hermann, The Complexity Challenge in Embedded System Design, 2008, Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing
LAB2016	LABCAR Product Family, http://www.etas.com/en/products/labcar_software_products.php
LIN01	http://www.cs-group.de/fileadmin/media/Documents/LIN_Specification_Package_2.2A.pdf
MAR2011	OMG, "UML Profile for MARTE: Modeling and Analysis of Real-time Embedded Systems", 2011, Retrieved from: http://www.omgmarte.org (08/03/2016)
MOD2011	Modelisar (2011, December). Retrieved https://itea3.org/project/modelisar.html
MOD2016	Modelica Association (2016, March 9th). Retrieved https://www.modelica.org/association
MOS2016	MOSAIK Consortium (2016). Retrieved from https://mosaik.offis.de/
MSI2016	Mathwork's Simulink page (2016). Retrieved from http://www.mathworks.com/products/simulink/
MST01	http://www.mostcooperation.com/membership/members-list/
MST02	MOST Specification Rev. 3.0 Errata 2
OSA2016	Open Signal Alliance Consortium, Retrieved from: http://www.opensig.org/ (2016)
PrTR01	http://www.transformers-project.eu/articles/results/#.VvJrUuLhDIU

PrTR02	http://www.transformers-project.eu/articles/holistic-simulations-results/#.VvJmguLhDIU
PrTR03	http://www.transformers-project.eu/mainmenu/background/#.Vvz1AXonaQ8
R22221	Jens Bastian; Christoph Clauß; Susann Wolf; Peter Schneider: Master for Co-Simulation Using FMI. 8th International Modelica Conference, Dresden, 2011.
RFC675	Vinton Cerf, Yogen Dalal, Carl Sunshine, "Specification of internet transmission control program", Retrieved from: https://tools.ietf.org/pdf/rfc675.pdf
RFC768	J. Postel, „User Datagram Protocol", Retrieved from: https://tools.ietf.org/pdf/rfc768.pdf
RFC1122	Internet Engineering Task Force,R. Braden, "Requirements for Internet Hosts Communication Layers", Retrieved from: https://tools.ietf.org/html/rfc1122
SBT2013	Stettinger, G.; Benedikt, M.; Thek, N.; Zehetner, J., "On the difficulties of real-time co-simulation", In: Proccedings of International Conference on Computational Methods for Coupled Problems in Science and Engineering, CIMNE2013, pp.1-11
SSS2012	S. Schütte, S. Scherfke, M. Sonnenschein, "MOSAIK - SMART GRID SIMULATION API, Toward a semantic based standard for interchanging Smart Grid simulations", Proceedings of SMARTGREENS 2012 - International Conference on Smart Grids and Green IT Systems, http://www.smartgreens.org
STR2006	S. Straßburger, "Overview about the High Level Architecture for Modelling and Simulation and Recent Developments ", 2006
SZB2013	Stettinger, G.; Zehetner, J.; Benedikt, M.; Thek, N., "Extending Co-Simulation on the Real-Time Domain", In: SAE Technical Paper No. 2013-01-0421, 2013, pp. 1-8
SZB2014	Stettinger, G.; Zehetner, J.; Benedikt, M.; Kokal, H.; Paulweber, M.; Wierse, M.; Toye, B., "Control of an engine test-bench via hardware-software-co-simulation", 14th Stuttgart International Symposium, 2014, pp. 869-880
TBZ2013	Thek, N.; Benedikt, M.; Zehetner, J.; Stettinger, G., "CO-SIMULATION UNDER HARD-REAL-TIME CONDITIONS", In: Proceedings of the NAFEMS World Congress, 2013, pp. 1-18
TCS2016	TWT's CoSimLab page (2016). Retrieved from https://www.twt-gmbh.de/en/products/co-simulation/cosimlab.html
TOL2002	A. Tolk, "Avoiding another Green Elephant - A Proposal for the Next Generation HLA based on the Model Driven Architecture", Fall Simulation Interoperability Workshop, Orlando, 2002, pp. 1-12
VET2015	VeTeSS project, www.vetess.eu (08/03/2016)
VIP2016a	Viproma, http://www.sintef.no/prosjekter/viproma/ (08/03/2016)
VIP2016b	E.Pedersen, "ViProMa", Retrieved from: http://folk.ntnu.no/annikak/Bladbar_AR2014IMT/marine_2014/assets/common/downloads/page0016.pdf

ZSK2014 Zehetner, J.; Stettinger, G.; Kokal, H.; Toye, B., "Real-time co-simulation for the control of an engine test bench" In: ATZ worldwide 116 (2014) 2, 2014, pp. 24-29

9 Acknowledgment

This project is co-funded by the Austrian Research Promotion Agency (FFG)

This project is co-funded by the Federal Ministry of Education and Research (BMBF)

PROJECT PARTICIPANTS:
Kompetenzzentrum - Das virtuelle Fahrzeug, Forschungsgesellschaft mbH („VIRTUAL VEHICLE") (AT)
AVL List GmbH (AT)
Spath MicroElectronicDesign GmbH (AT)

Dr. Ing. h.c. F. Porsche AG (DE)
Volkswagen AG (DE)
Robert Bosch GmbH (DE)
ETAS GmbH (DE)
dSPACE GmbH (DE)
ESI ITI GmbH (DE)
TWT GmbH Science & Innovation (DE)
RWTH Aachen University (DE)
Technische Universität Ilmenau (DE)
Leibniz University of Hannover (DE)

Renault SAS (FR)
Siemens Industry Software SAS (FR)

DISCLAIMER
This ITEA3 Project has been made possible by a financial contribution by the Austrian Research Promotion Agency (FFG) and by the Federal Ministry of Education and Research. The Publication as provided reflects only the authors' view.

Every effort has been made to ensure complete and accurate information concerning this document. However, the author(s) and members of the consortium cannot be held legally responsible for any mistake in printing or faulty instructions. The authors and consortium members retrieve the right not to be responsible for the topicality, correctness, completeness or quality of the information provided. Liability claims regarding damage caused by the use of any information provided, including any kind of information that is incomplete or incorrect, will therefore be rejected. The information contained in this document is based on author's experience and on information received from the project partners.