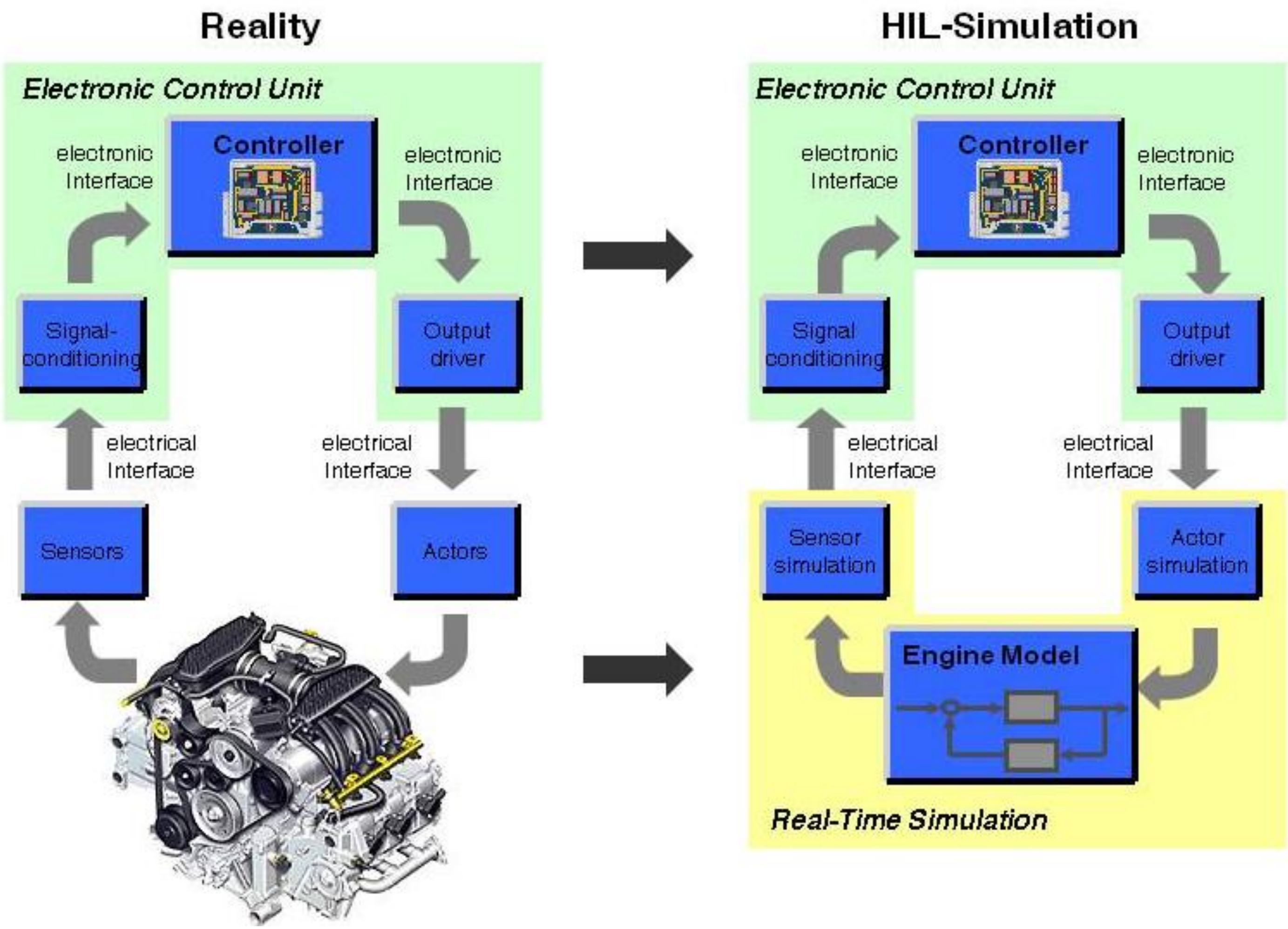
What‘s new

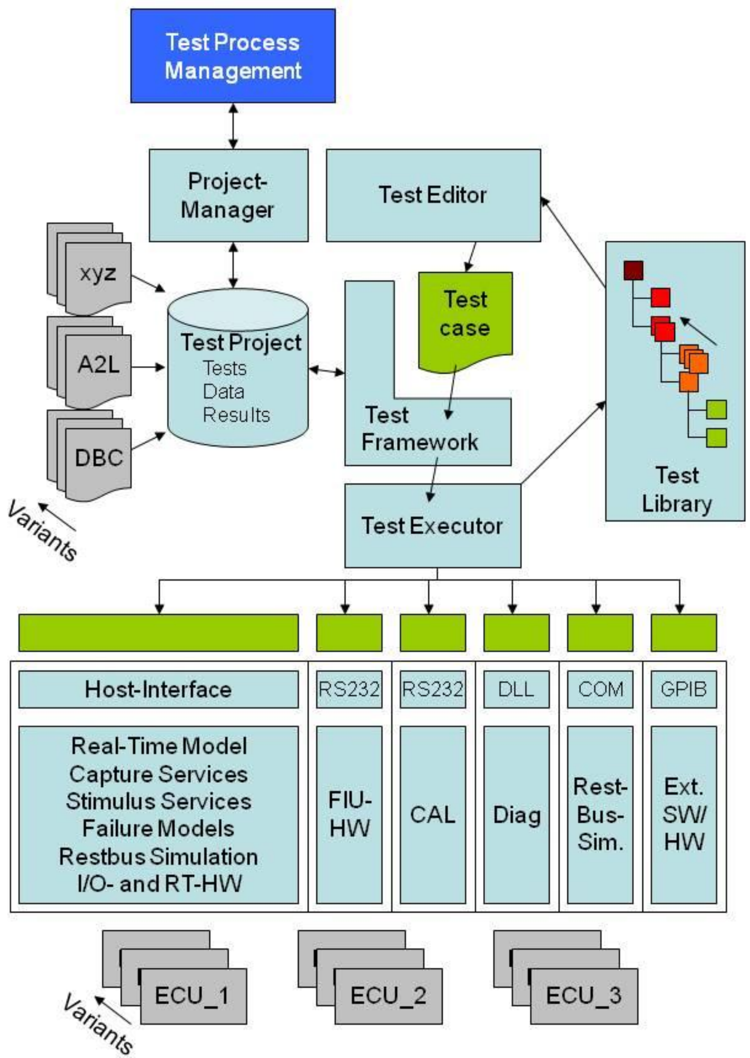
in ASAM AE HIL API V1.0.0 ?

Darmstadt, 2009-06-23 (ASAM TSC Meeting)

What is Hardware-in-the-Loop Simulation ?



Architecture of a HIL Test Bench



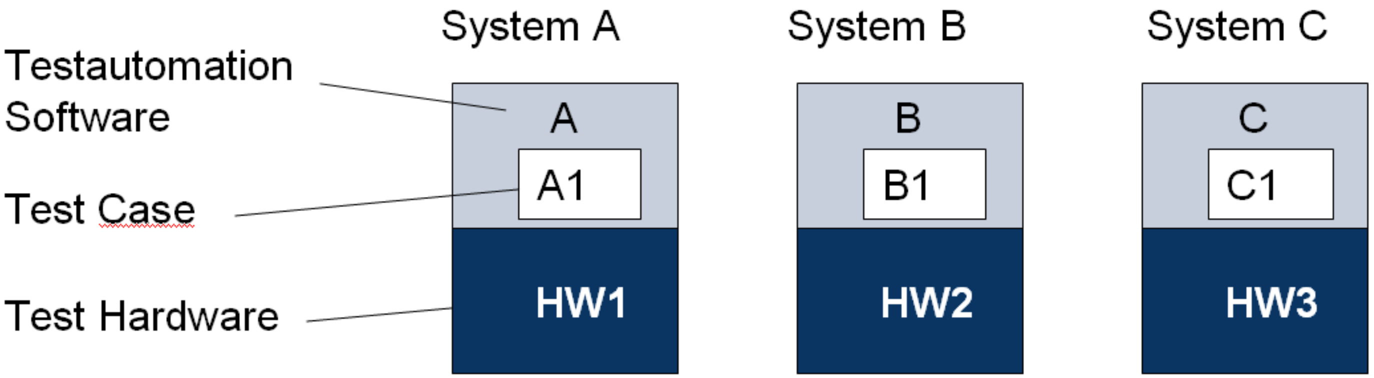
Host Software for
- Test Process Management,
- DataManagement and
- Test Automation

Driver / Host Interfaces

Test bench components
- Simulator hardware
- External hardware
- External remote-controlled tools

System Under Test: ECU(s)

Motivation / Current Situation



- Test cases directly depend on the used Test Hardware
- Test Automation Software and Test Hardware are rigidly coupled
- Not always the ‚best' test software can be combined with the ‚best' testing hardware
- Know-how cannot be transferred from one test bench to the other

  (additional training costs for employees)
- Switch to the newest testing technology is difficult because of

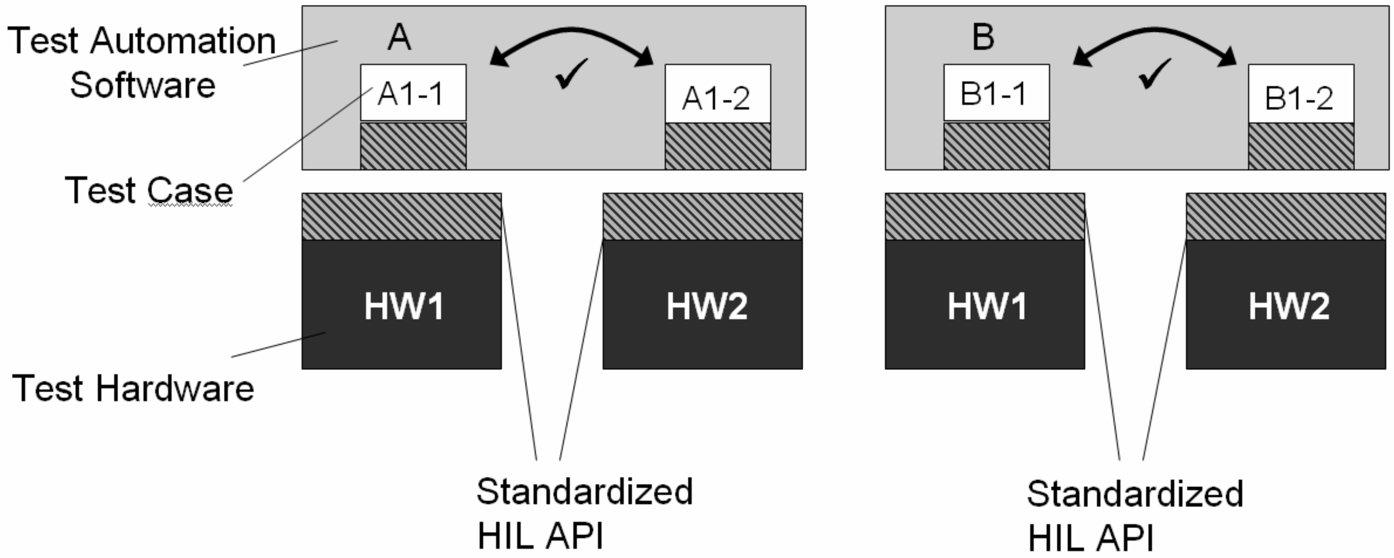
  tool specific formats and test hardware compatibility
- Test cases cannot be ported from one test system to another

  (e.g. test case exchange between OEM and supplier)

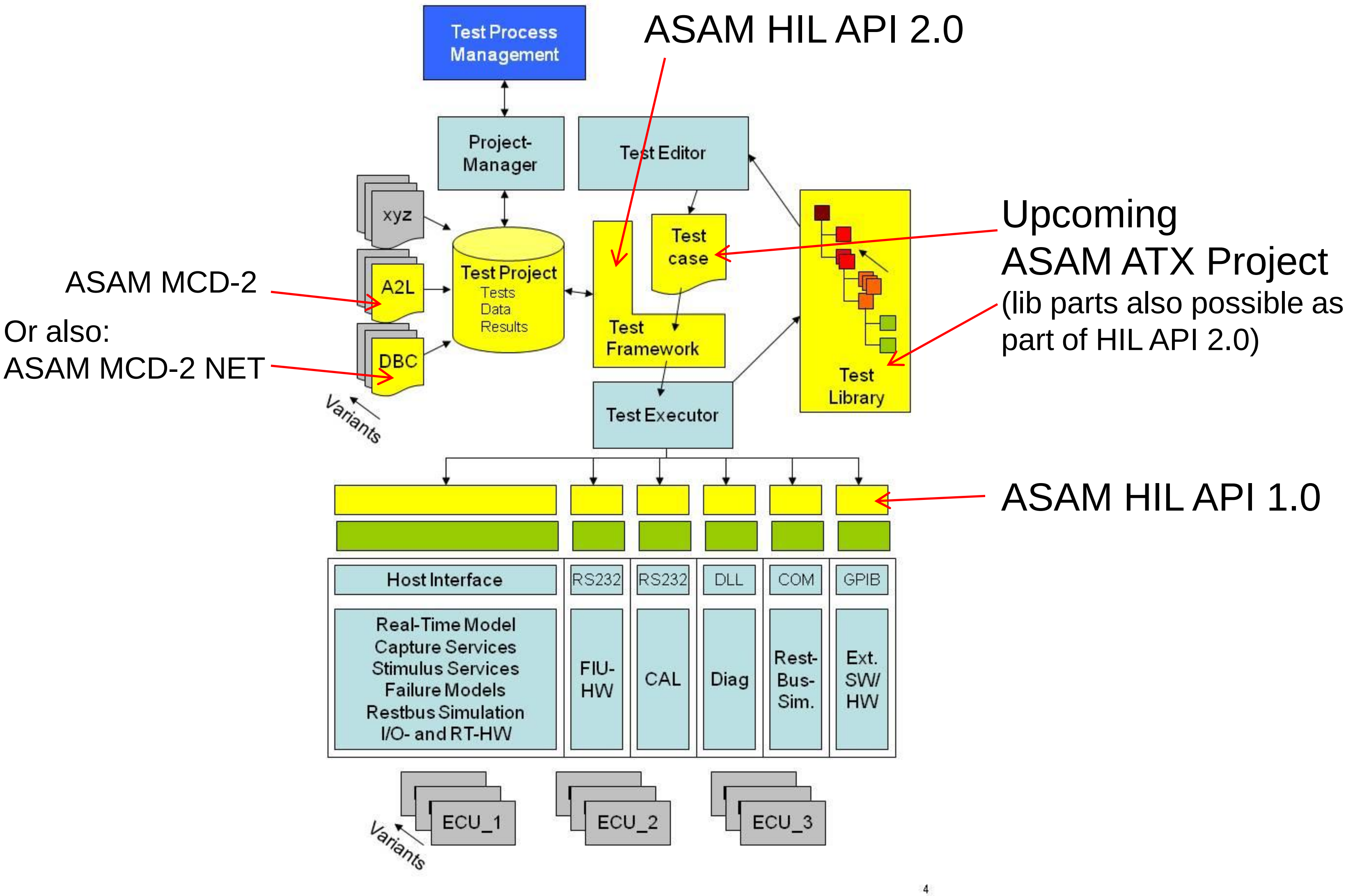
Project's Major Goals and Solution Approach

- Reuse of test cases within the same test automation software on different test hardwares
- Reduction of effort for test hardware integration into test automation software
- Long-term protection of software investments and test case development efforts

➔ **Separation of Test HW and Test SW by means of standardized APIs**



Standardisation Context



Technological Approach

API Proposal Development

Enterprise Architect

ASAMs standard tool for UML

HIL API Reference Model

Technology References

C# for .NET Technology

Python Script Language Classes

JAVA

Others to be added without the need to modify the API model in UML

Work group members

Goal: Involvement by main stakeholders of HIL technology to increase the acceptance - and therefore the use of - the new standard in the market

Project team members:

➢ AUDI	(OEM - HIL end user)
➢ BMW	(OEM - HIL end user)
➢ Bosch	(ECU Supplier – HIL end user)
➢ Continental	(ECU Supplier – HIL end user)
➢ dSPACE	(HIL System and tool supplier) (project leader)
➢ ETAS	(HIL System and tool supplier)
➢ Vector Informatik	(HIL System and tool supplier)
➢ National Instruments	(HIL System and tool supplier)
➢ Berner & Mattner	(HIL System and tool supplier)
➢ Tracetronic	(HIL tool supplier)
➢ M&K	(ASAM contractor)

Consulting partners resp. temporary project partners:

AVL, MB-Tech, Softing, RA Consulting, D2T

Work group members

Goal: Involvement by main stakeholders of HIL technology to increase the acceptance - and therefore the use of - the new standard in the market

Project team members:

➢ AUDI	(OEM - HIL end user)
➢ BMW	(OEM - HIL end user)
➢ Bosch	(ECU Supplier – HIL end user)
➢ Continental	(ECU Supplier – HIL end user)
➢ dSPACE	(HIL System and tool supplier) (project leader)
➢ ETAS	(HIL System and tool supplier)
➢ Vector Informatik	(HIL System and tool supplier)
➢ National Instruments	(HIL System and tool supplier)
➢ Berner & Mattner	(HIL System and tool supplier)
➢ Tracetronic	(HIL tool supplier)
➢ M&K	(ASAM contractor)

Consulting partners resp. temporary project partners:

AVL, MB-Tech, Softing, RA Consulting, D2T association for standardisation of automation and measuring systems

Some ASAM HIL API 1.0 projects facts

- 1,5 years of development before becoming an ASAM project
- Nov. 2007 official ASAM project kick-off at AUDI, Ingolstadt
- 14 x 2-day meetings, 1 x 1-day meeting, 20 Web conferences
- estimated total effort of all participants: **> 12 my**

Content Overview (Data Value Container)

The ValueContainer package provides a set of data structures, strictly speaking container classes, that are commonly used by the HiL-API ports to gain and deliver data.

These container classes are divided into three categories.

1. The first one comprises all ordered container classes whose elements are accessed by an integer based index, e.g. vectors and matrices.
2. There are some derivatives of these data structures that are specialized for different domains, e.g. the Data Capturing, ECU calibration access (e.g. maps, curves, XYValues, …)
3. The third one consists of unordered, named collections, also known as associative arrays. Elements of these data structures are accessed by name which is a string and not an integer.

Content Overview (Data capturing, triggering, capture results)

The Capture resp. Capture Result package provides a set of operations that are used to define measurement (resp. capture) configurations, to control the execution of measurements and to obtain the measured data as results.

The preferred measurement file format is ASAM MDF 4.0.

Trigger conditions can be described following the ASAM General Expression Syntax 1.0

Other proprietary file formats, i.e. customer or supplier specific formats can be added consistently using the so-called DOCUMENT HANDLING concept.

Content Overview (Collection handling)

Collections are used to build lists of elements, resp. associative arrays. These classes are templates for typed collections.

➔ Major Use Case: Methods with in/out parameters of dynamic structure (property value pairs)

➔ There are different types of collections, such as: IndexedCollection, Named Collections, …

➔ Standardized set of methods to construct, retrieve, check or iterate

➔ Could be easily extended to new bind types

➔ ASAM HIL API collections are directly mapped to the native collections of the respective Technology References, i.e. to C#, Python and JAVA collections

➔ ASAM HIL API collections could and should be made new ASAM Harmonized Data Objects to be used in upcoming API definitions

Content Overview (Signal description handling)

In general the signal description is used to describe a signal for general purpose usage. A signal can be described by using synthetic waveform elements like ramp or sine and/or with elements which contain the signal points in form of numerical data.

These described signal descriptions can be applied e.g. by the signal generator of the Model Access Port or to define reference signals.

- Supported signal forms (segments) are: CONST, RAMP, SINE, SAW, PULSE, EXP, NOISE, IDLE, …
- Signal channels can also be added to or multiplied with other (➔ modulation resp. overlaying)

Content Overview (Simulation Model Access)

The MAPort (Model Access) facilitates the access to the variables of the simulation model.

Access means
- read and write of parameters and measurements (client-controlled)
- support of commonly used data types (scalar, array, 1D, 2D, …)
- stimulation of variables in real-time

As the HIL API project was initiated by Hardware-in-the-Loop simulation stakeholders this functionality is tailored but not limited to real-time simulation applications.

The BMBF project MODELISAR is highly interested to adapt this part of ASAM HIL API 1.0.

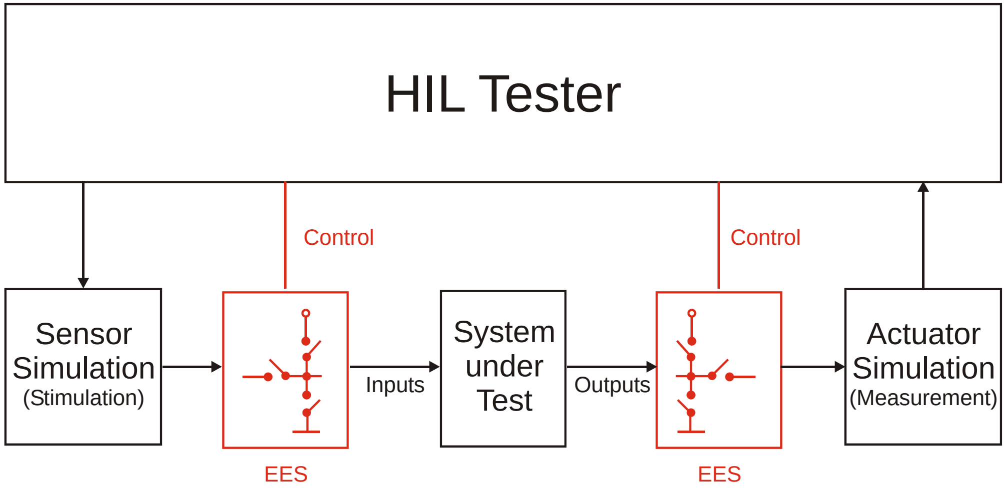
Nearly the complete MAPort functionality including its related common parts, such as *Capturing, Signal Descriptions* and *Signal Generator* can easily be ported or adapted also to non-real-time simulation environments.

Content Overview (Electrical Error Simulation)

Test cases may not only comprise checking the behaviour of the SUT in a fully functional environment. It is also important to check the SUT in case of electrical errors on the input and output pins.

Typical errors that have to be tested are electrical problems, mainly caused by wiring. The so-called electrical error simulator (EES) generates this class of errors at the connections between the SUT, its environment, and the HIL HW.

The HIL API EES port provides a general API for electrical error simulation hardware. The API hides the specific API of the used hardware, its driver software, and the communication between the machine the test is running on and the EES hardware.



Content Overview (ECU Diagnostics Access)

The Diagnostic Port facilitates the integration of diagnostic tools within the hardware-in-the-loop test automation setup. A diagnostic tool may consist of hardware, software or both that allow for ECU diagnostics.

The Diagnostic Port enables not only the integration of diagnostic tools within the test automation system but also the unification and standardization of the diagnostic tools' functional interface by defining a standardized API. By implementing the standardized Diagnostic Port API diagnostic tool providers make sure that the diagnostic tools are exchangeable.

The Diagnostic Port API is not a replacement or an extension of existing diagnostic standards (e.g. ASAM MCD 3D) or diagnostic protocol standards.

Rather the Diagnostic Port API forms a programming interface that reflects the client's requirements related to a diagnostic tool in the context of hardware-in-the-loop test automation. In this sense ASAM HIL API Diag Port can be seen as a form of ASAM MCD 3D convenience layer.

Content Overview (ECU Measurement and Calibration Access)

The ECU Port facilitates the integration of ECU measurement and calibration tools within the hardware-in-the-loop test automation setup. An ECU measurement and calibration tool may consist of hardware, software or both.

The ECU Port Port enables reading and writing to ECU internal variables. This functionality is nearly identical to the methods of the Model Access Port.

Raster synchronous data measurement is handled via the *Capturing* and *Capture Result* parts of the common section of HIL API 1.0.

Stimulus features are not supported for ECUs because the needed real-time infrastructure is not available today .

The ECUCPort also allow for ECU memory page switching, e.g. to work with two different calibration data sets.

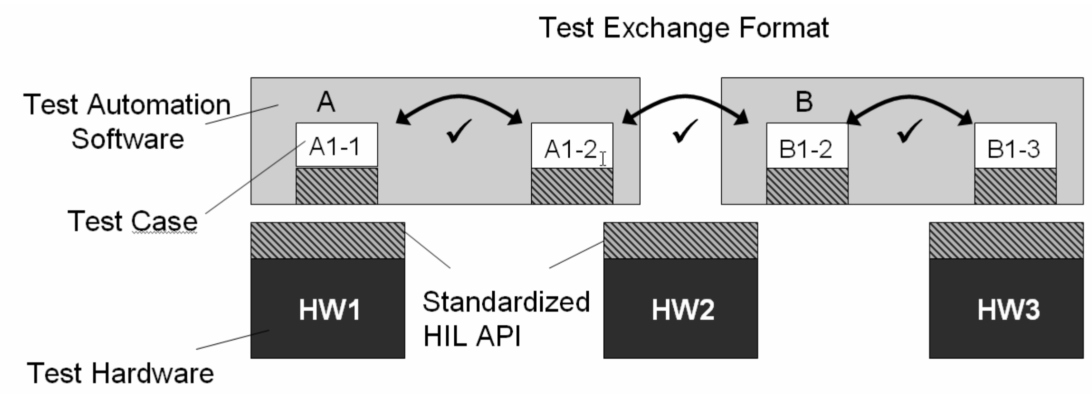
Concrete Deliverables

- ASAM HIL API 1.0 UML Model in Enterprise Architect
- ASAM HTML API Reference Guide
- ASAM HIL API 1.0 Programmer's Guide
- Technology Reference for C#, Python and Java including
  + Interface Defintions
  + Reference Implementations for some interfaces *(i.e. for the non competetive interfaces)* and Dummy Implementation for the rest
- Modelling Guides for C#, Python and Java (*‚How to map UML to the native languages ?')*
- Comprehensive set of sample codes for all HIL API functions in all 3 Technologies (sample code explained via UML sequence diagrams in the Programmer's Guide)

Additonal Outcome of the Project

- Draft for ASAM General Expression Syntax Standard (also used in MDF 4.0, ASAM MCD 2MC 1.6)
- Extension proposal for Harmonized Data Objects (i.e. Collections)
- MS Excel based utility for Enterprise Architect UML model documentation

Next Steps – Related Projects



Next step of test automation standardisation:

- Reuse of test cases between different test automation software systems
- Standardized exchangeability by a Test Exchange Format

Next Steps – Related Projects

ASAM

ISO International Organization for Standardization

ASAM ATX Part 1

(TDX+TAF Folder, Result, Suites - TAF Core)

OTX Core
ISO WD/13209-2

OTX Standard Libraries
(Lib-Com, Lib-Mmi, …)
ISO WD/13209-3

ASAM ATX Part 2 or
as part of ASAM HIL API 2.0
Standard Libraries for ASAM HIL API