ASAM XIL 3.0
Release Presentation

Ralph Trogus

AUDI AG

Sep 30, 2024
Ingolstadt

ASAM Association for Standardization of Automation and Measuring Systems

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Deliverables

Introduction

ASAM XIL (Generic Simulator Interface) was developed to allow to exchange combinations of test automation software and virtual or physical testbenches.

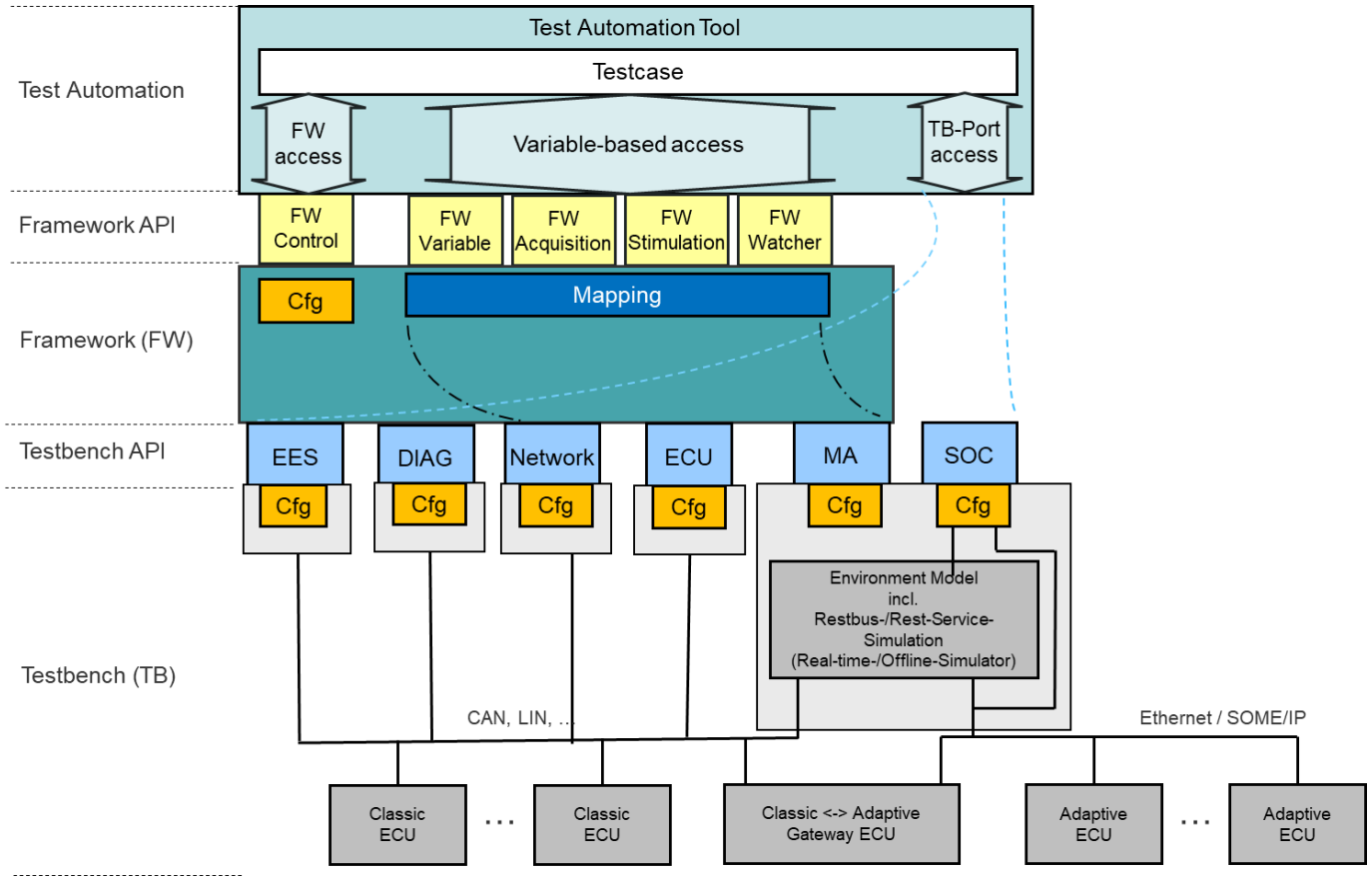
Therefore, the standard primarily describes the interface between test automation software and virtual or physical testbenches.

Introduction

Motivation of the XIL Standard

- Standardize the communication between test automation software and X-in-the-loop test benches.
  -> Interaction of different test automation tools with different test bench tools via one interface.
  -> Reduction of total costs.

- XIL Concept of Ports and Framework.
  - Testbench API (Port Access).
  - Framework API (Variable Access).



Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Deliverables

Motivation for New Release

- Support of Service oriented communication (SOC) in the new release. SOC has become an important foundation in automotive software architecture. And its importance will further increase in the coming years.
- Integration of requirements from the VDA project group “SIL Standardization” such as Traceability and Event Monitoring.
- Consolidation of ECUM and ECUC Testbench ports into a common ECU Testbench port.
- Usage of XIL on LINUX platforms.
- Support of advanced data structures to be used with modern interfaces.
- Basic concepts for streaming and event handling are required, as they are important for modern interfaces.

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Deliverables

New features

Concerning Support of SOC (primary scope SOME/IP)

SOC – SOCPort

- The new SOCPort allows testing of service-oriented communication in vehicle electronics (scope SOME/IP).
- The SOCPort can be configured with service description files (i.e. arxml) and service simulation configurations.
- Support interrogation of dynamic service instances.

SOC – Enable the call of service methods

- Synchronous or asynchronous call of SOME/IP methods on arbitrary service instances.

SOC – Handling of fields

- Synchronous or asynchronous handling of field values (getter and setter calls).
- Subscription for field value change notifications and reception of the notifications.

SOC – Event listener

- Subscription for service events and reception of the notifications.

New features

Concerning Support of SOC (primary Scope SOME/IP)

SOC – Monitoring of SOC messages

- Live surveillance of SOME/IP messages.
- Filtering of monitored messages based on message meta data (i.e. IDs of methods, events, etc.).

SOC – Capturing SOC communication

- Capturing of arbitrary SOME/IP messages.
- Recording to MDF files in raw or interpreted format according to MDF associated standard SOC Data Logging.
- Filtering of captured messages based on message meta data.
- Triggering of capture start and stop based on message meta data and message payload (GES-Expression).

SOC – Control the service simulation

- Start and stop of configured service simulations.
- Enabling or disabling the simulation of service instances, methods, field or events.
- Explicit triggering of service notifications (events and fields).

New features

Concerning ECUPort

ECUPort – support of multiple devices with one ECUPort instance

- Configuration and initialization of multiple devices.
- Synchronized start of all configured captures of all devices.

ECUPort – generalizing the handling of measurement and calibration variables

- Capturing of calibration variables including curves and maps.
- One recorder can capture measurement and calibration variables in one synchronized recording.
- Support of writeable measurements.

ECUPort – support of calibration data files

- Calibration data files (e.g. ASAM CDF files) can be written to an ECU device.

ECUPort – merging of the former existing ECUCPort and ECUMPort

- Optimized handling of devices and their resources.

New features

Cross sectional

Data streaming

- Standardized monitor concept to enable efficient streaming. The concept is based on queues and prevents the server from being blocked by client activities.
- The monitor concept is integrated in the SOCPort, MAPort, ECUPort.

Traceability (Error handling, Logging)

- Streaming of log and error messages to a client application is enabled by a new log monitor interface. This is also based on the monitor concept for streaming.

Complete redesign of the ValueContainer

- Support for arbitrary dimensional arrays.
- Support for dynamic structures.
- Support for unions.
- Generalization of maps and curves to lookup tables.

New features

User defined events (MAPort)

- Support of user defined events, which allow the monitoring of parallel events. This is also based on the monitor concept for streaming. The event monitor is supported by the MAPort.

Support of custom methods

- Definition of custom interface methods on testbench and ports.
- Allows the server to provide non-standard functionality for specific use cases.

Other changes

Support of LINUX environments

- All assemblies are built against .NET Standard 2.0 which is compatible to Windows and LINUX platforms
- Dependencies to GAC (Global Assembly Cache) are removed
- Structure of Manifest files has been adapted

Python compatibility

- Python classes are not provided
- The adaptions in Python are described in the C# To Python Mapping Rules
- The Python interface follows the C# interface
- Python enumerations that follow the C# enumerations are provided
- C# to Python mapping and interoperability is described in the C# To Python Mapping Rules

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Deliverables

Backward compatibility

ASAM XIL 3.0.0 is a major release and not backward compatible to previous versions.

In addition of removing deprecated elements, the major release of this standard contains further breaking changes to the previous release.

The breaking changes are listed in an appendix of the ASAM XIL Specification.

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Deliverables

Relation to other standards

ASAM Data Types

- ASAM data types are used in the specification and mapped to native data types for the respective technology references.

ASAM General Expression Syntax

- GES can be used for parameter definitions of SignalDescriptionSets, Watcher conditions, ConstSymbol expressions and in PayloadCondition of the SomeIPWatcher.

ASAM MDF BS

- Framework measurement data and Testbench capture data are stored in MDF.

ASAM MDF AS Data Logging of Service Oriented Communication*

- Describes the mapping of SOC Data to ASAM MDF data structures.

AUTOSAR SOME/IP Protocol Specification

- The SOC Port supports the SOME/IP protocol.

*is announced for spring 2025

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Relation to Other Standards
7	Deliverables

Deliverables

Documents

- Part 1: Specification
- Part 2: C# To Python Mapping Rules
- ASAM XIL C# Interface Reference
- Schema files

Supplementary Files

- Technology Reference for C#
- C# Examples
- Preconfigured setup for Microsoft Visual Studio Code