ASAM Technical Seminar 2020

Release of ASAM XIL 2.2

Christian Sczyrba-Neumann

dSPACE GmbH

Friday, Oct 09, 2020

ASAM Association for Standardization of Automation and Measuring Systems

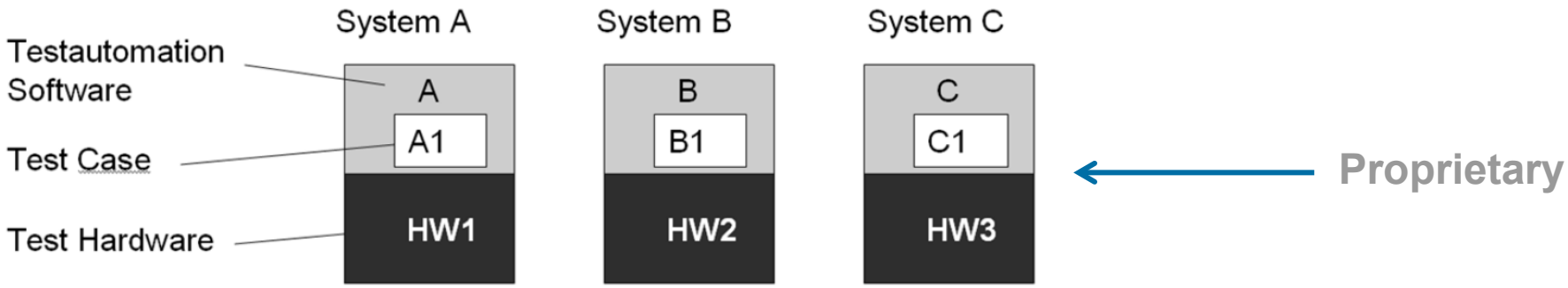
Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward Compatibility
6	Deliverables

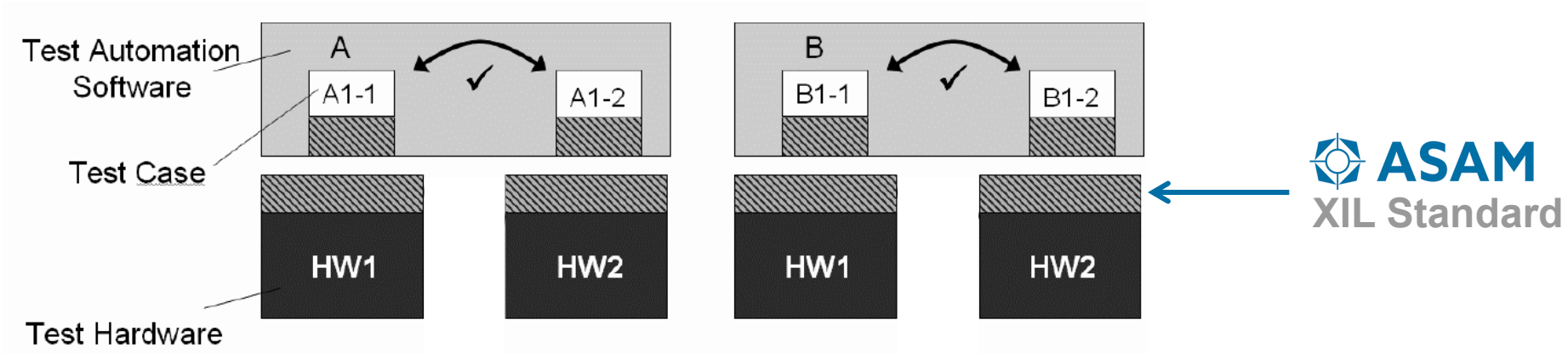
Agenda

1	**Introduction**
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	Deliverables

Motivation of the XIL Standard



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



Concept of Ports

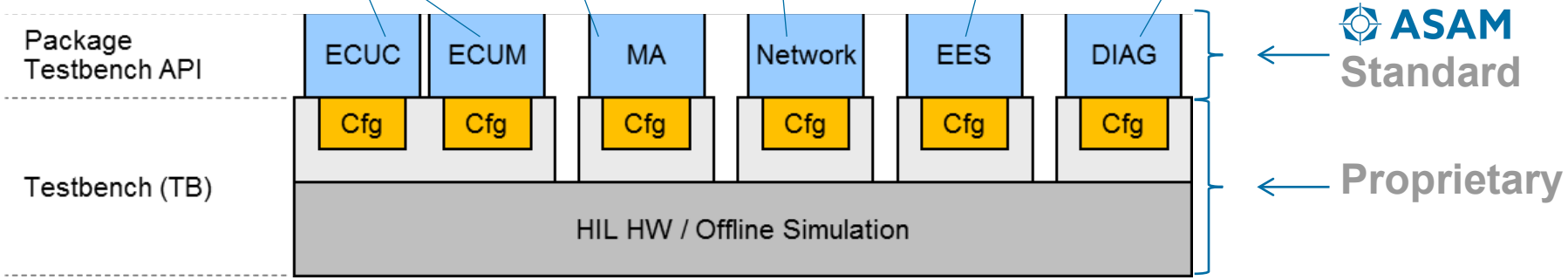
MAPort
Model Access port provides access to the simulation model read and write parameters, capture and generate signals.

NetworkPort
provides access to field bus systems such as CAN. E. g. Allows measurement (monitoring) and transmission (single transmit or replay) of bus data.

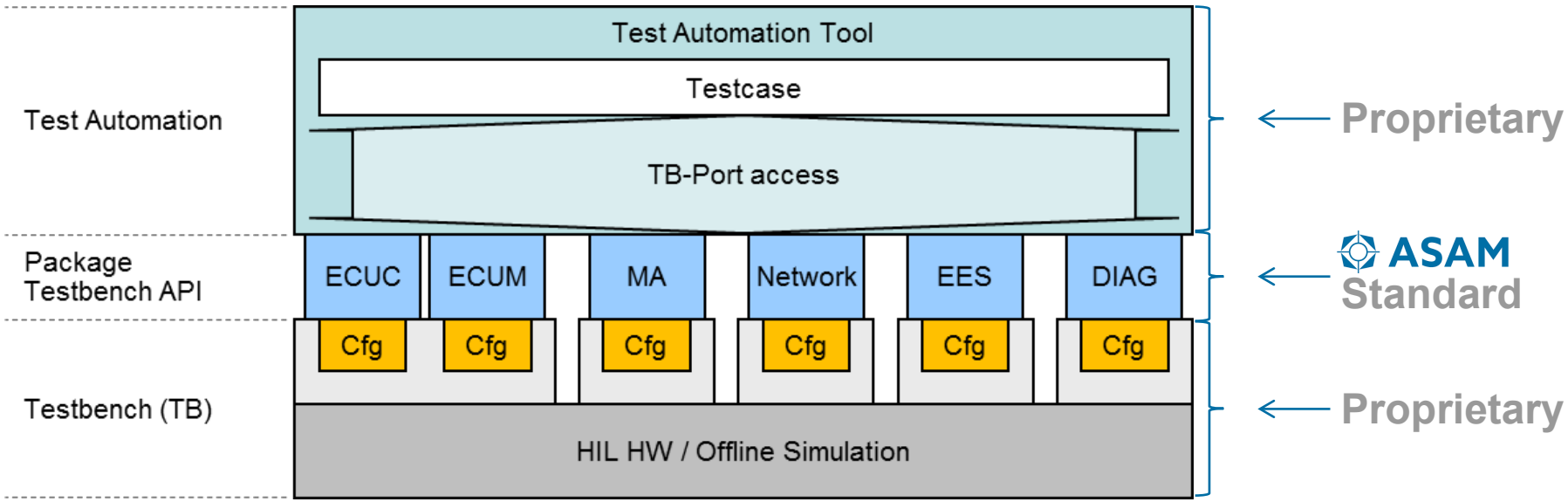
EESPort
Electrical Error Simulation port controls electrical error simulation hardware. It allows the setup of different types of errors (e. g. short cuts).

ECUPorts (M+C)
The ECUM port allows capturing and reading of **M**easurement variables. The ECUC port is used for **C**alibration.

DiagPort
Diagnostic port communicates with a diagnostic system, reads data via diagnostic services from an ECU.

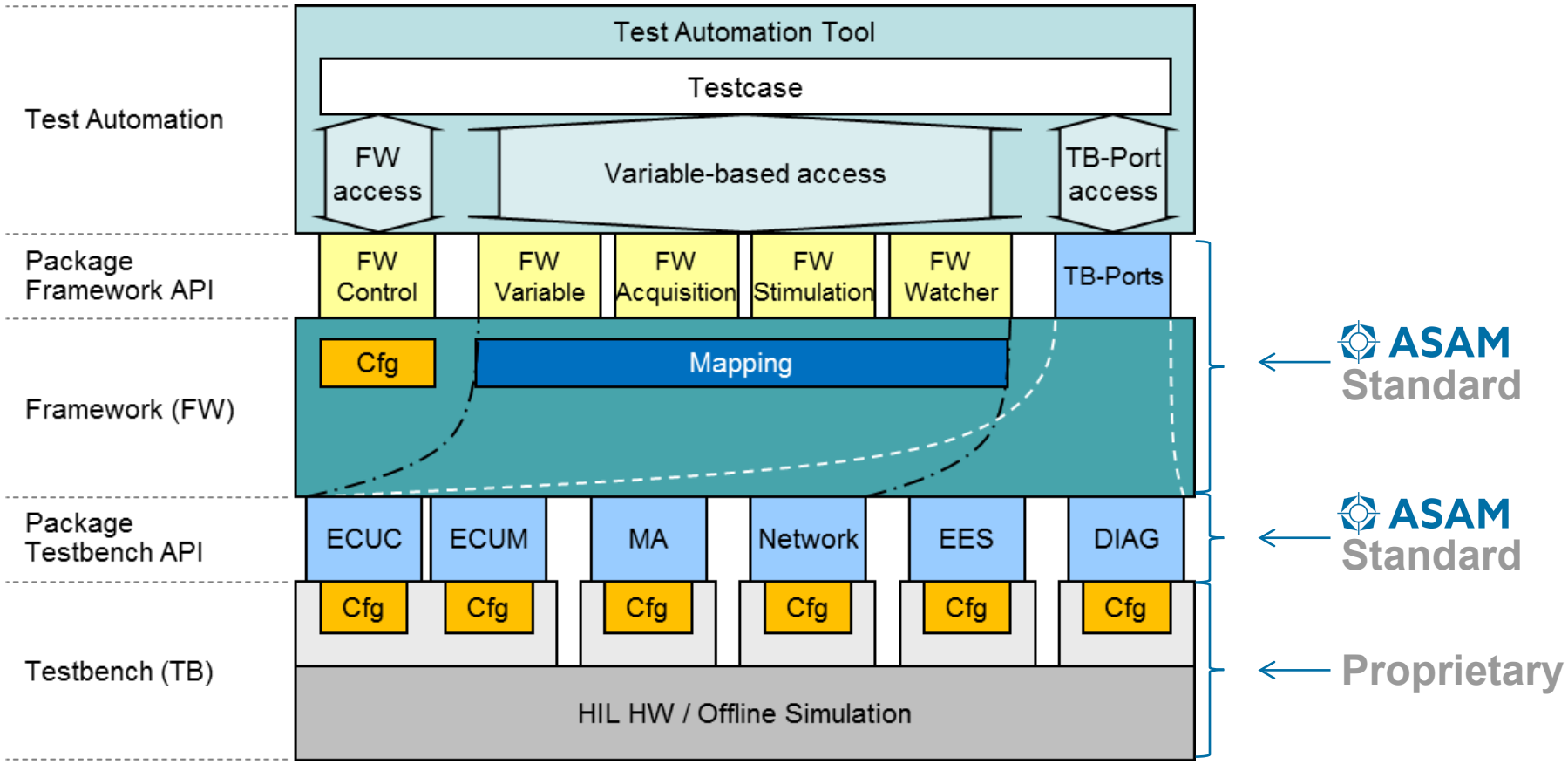


Port-Based Access of a Test Automation Tool



Variable-Based Access of a Test Automation Tool

- Port independence of test cases by using an object-oriented access to variables
- Test Developer can use both: Testbench Port access and variable based access by the Framework



Agenda

1. Introduction
2. **Motivation for New Release**
3. New Features
4. Other Changes
5. Backward-Compatibility
6. Deliverables

Motivation for ASAM XIL 2.2

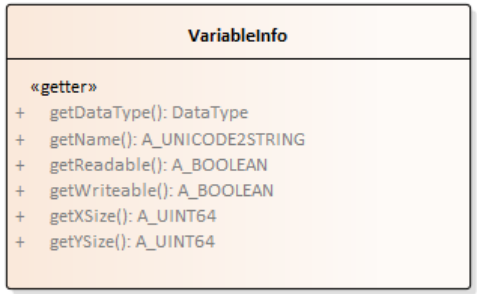
- Address the feedback from cross tests in 2016 and 2017
- Incorporate the vendor feedback from the ASAM XIL 2.1 implementation
- Remove errors in the standard
- Close gaps in the specification about the expected behavior in certain areas
- Add new features requested by the users (e.g. capturing support of vector and matrix data types)

Agenda

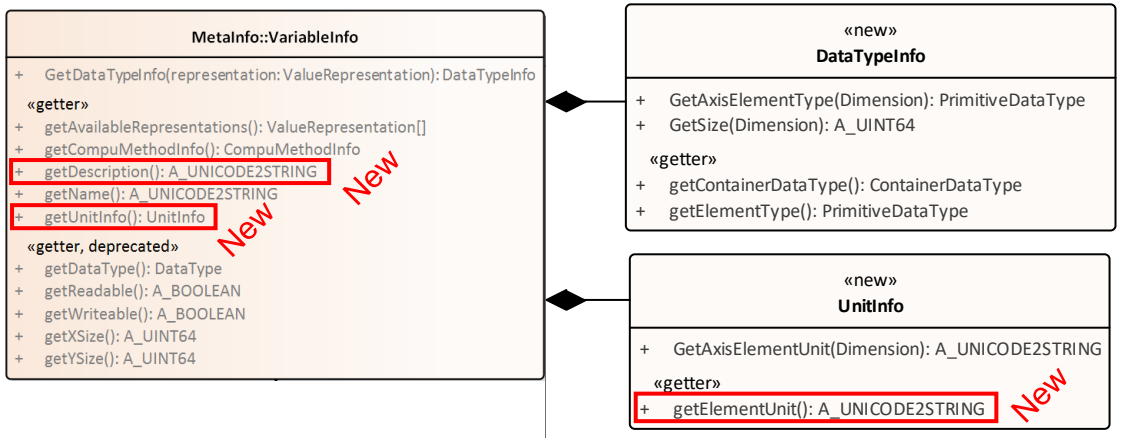
Extension of the VariableInfo Interface 1/4

Improvement

ASAM XIL 2.1



ASAM XIL 2.2



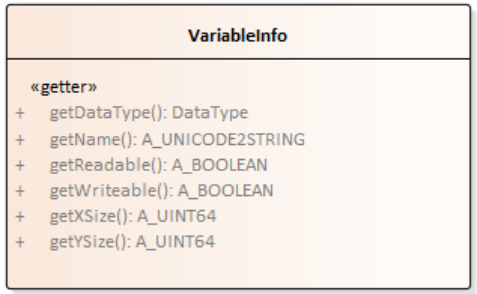
- The `VariableInfo` interface was not able to return a variable description text and the physical unit directly
- It took multiple function calls to get this information in ASAM XIL 2.1
- This information is now available directly through the new getters `getDescription()` and `getUnitInfo()`
- Improves performance and usability at the same time

ASAM

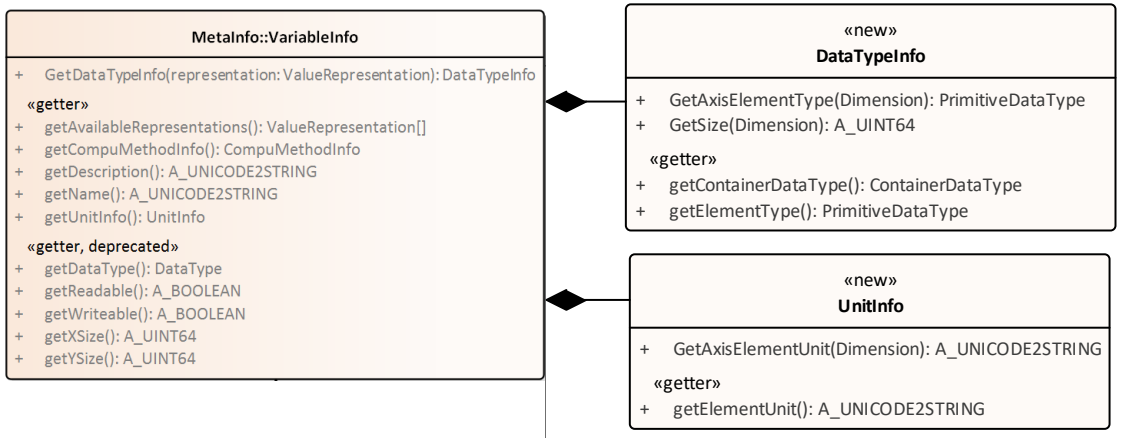
Extension of the VariableInfo Interface 2/4

Improvement

ASAM XIL 2.1



ASAM XIL 2.2



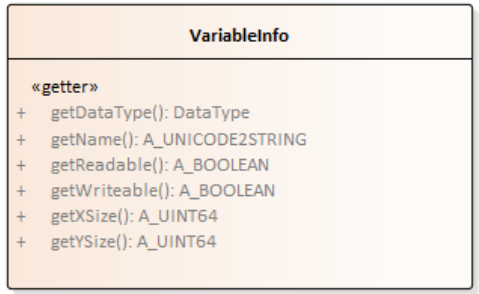
- The `VariableInfo` interface is now the unified location to get the variables data type and its readability and writability.
- The port specific functions are not needed anymore

ASAM

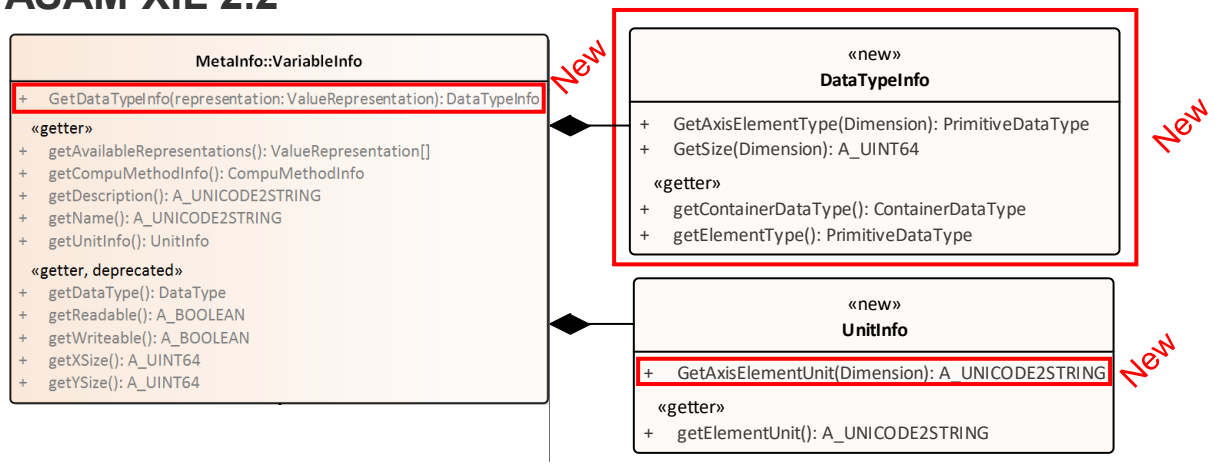
Extension of the VariableInfo Interface 3/4

New

ASAM XIL 2.1



ASAM XIL 2.2

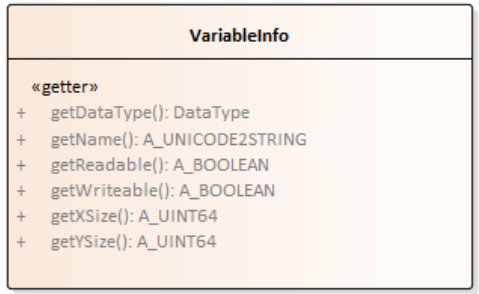


- The `VariableInfo` interface was also not able to retrieve any information on the element and axis types of curve, matrix and map variables
- The new `GetDataTypeInfo()` and `GetAxisElementUnit()` function provides a way to get this information of a complex variable

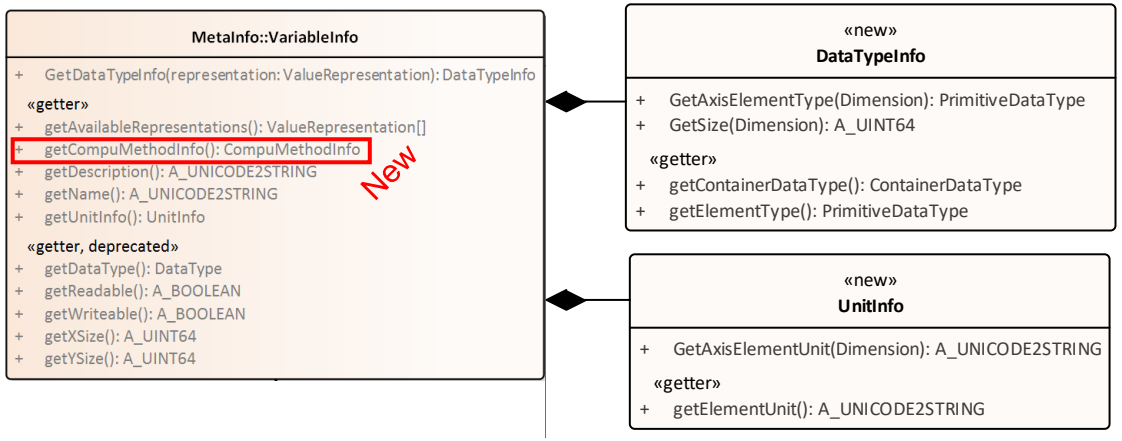
Extension of the VariableInfo Interface 4/4

New

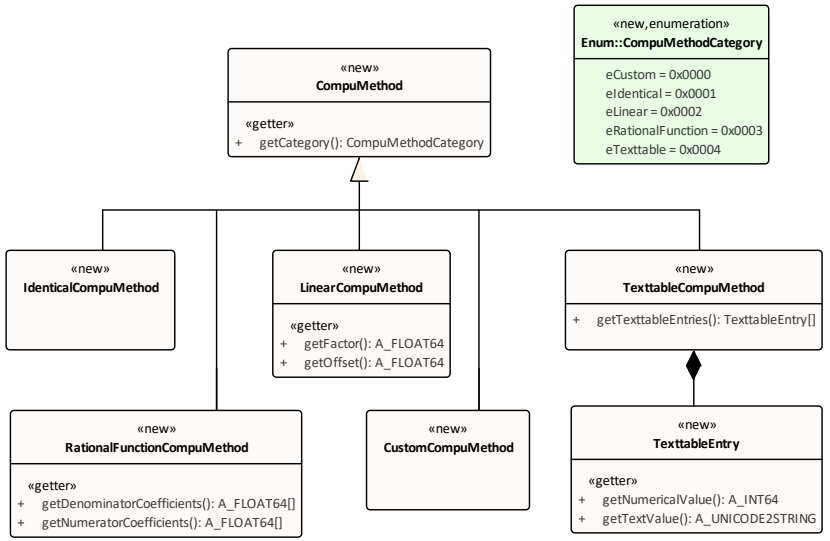
ASAM XIL 2.1



ASAM XIL 2.2



- The `VariableInfo` interface now distinguishes between physical value representation and raw value representation
  - It is now able to retrieve data type information on the physical value and the raw value of a testbench variable
  - It is now possible to retrieve information about the conversion method from raw value to physical value

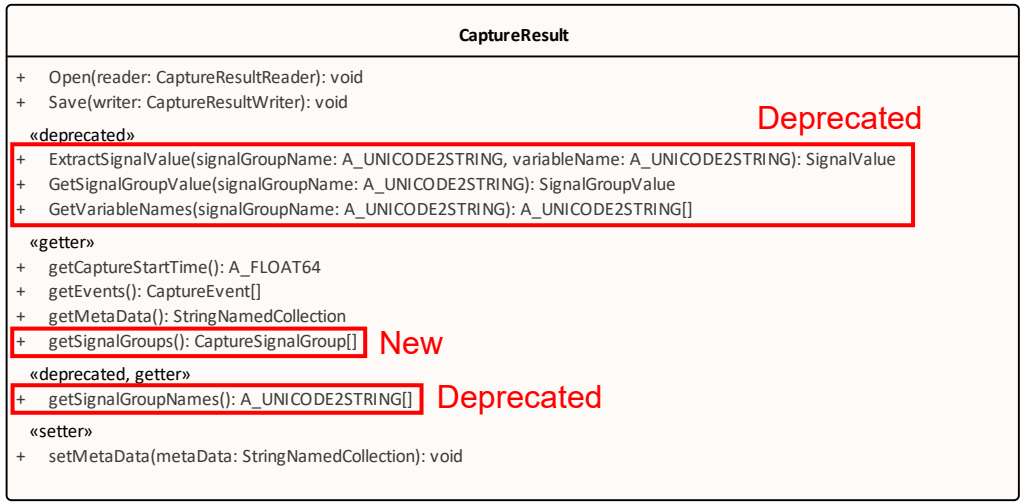


As result of this change, the user can now choose whether to read and write data from a testbench variable as raw value or physical value

ASAM

Capture Support For Complex Data Types 1/2

New

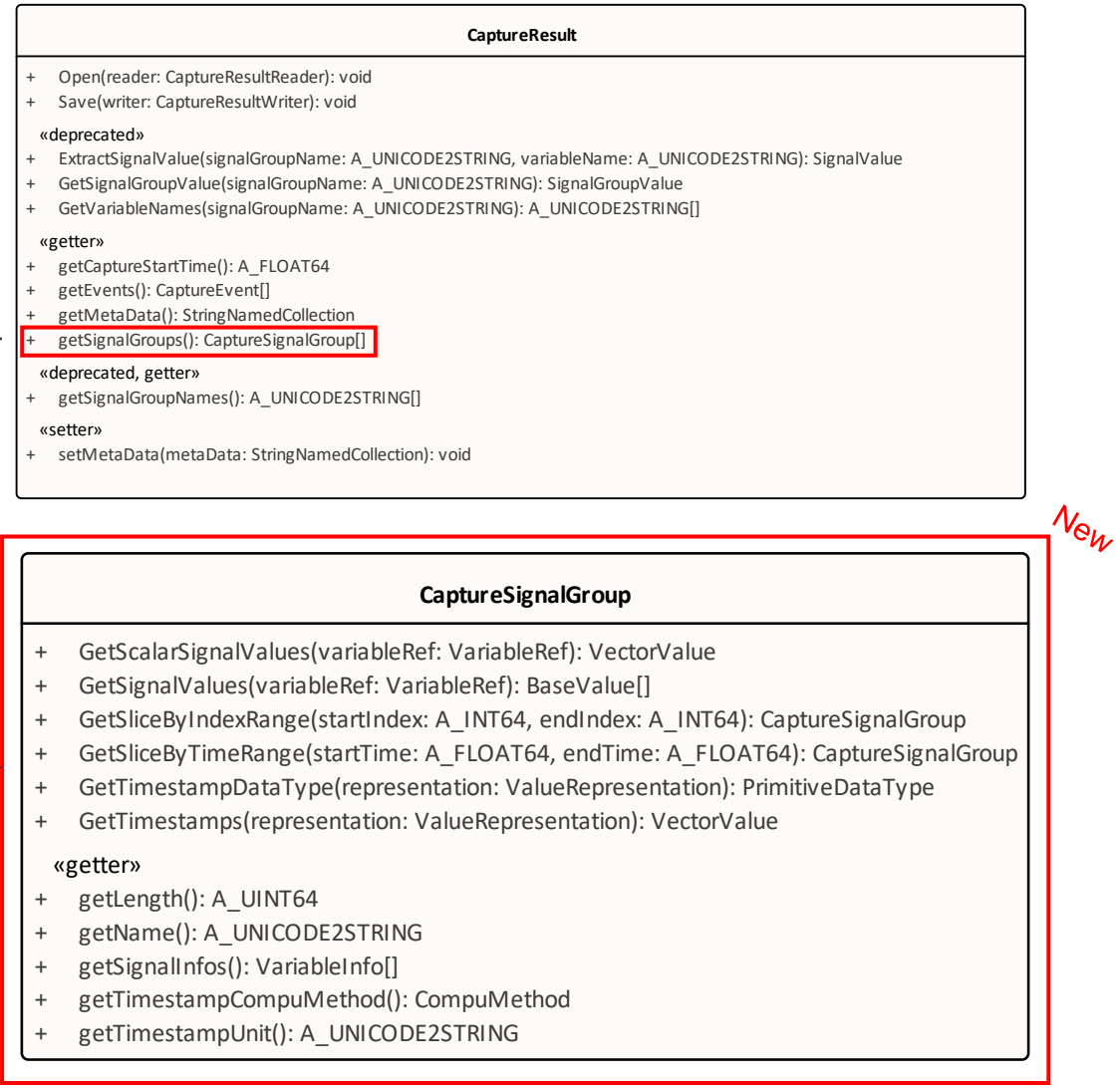


Goal

Support the capturing of complex data types

Capture Support For Complex Data Types 2/2

New



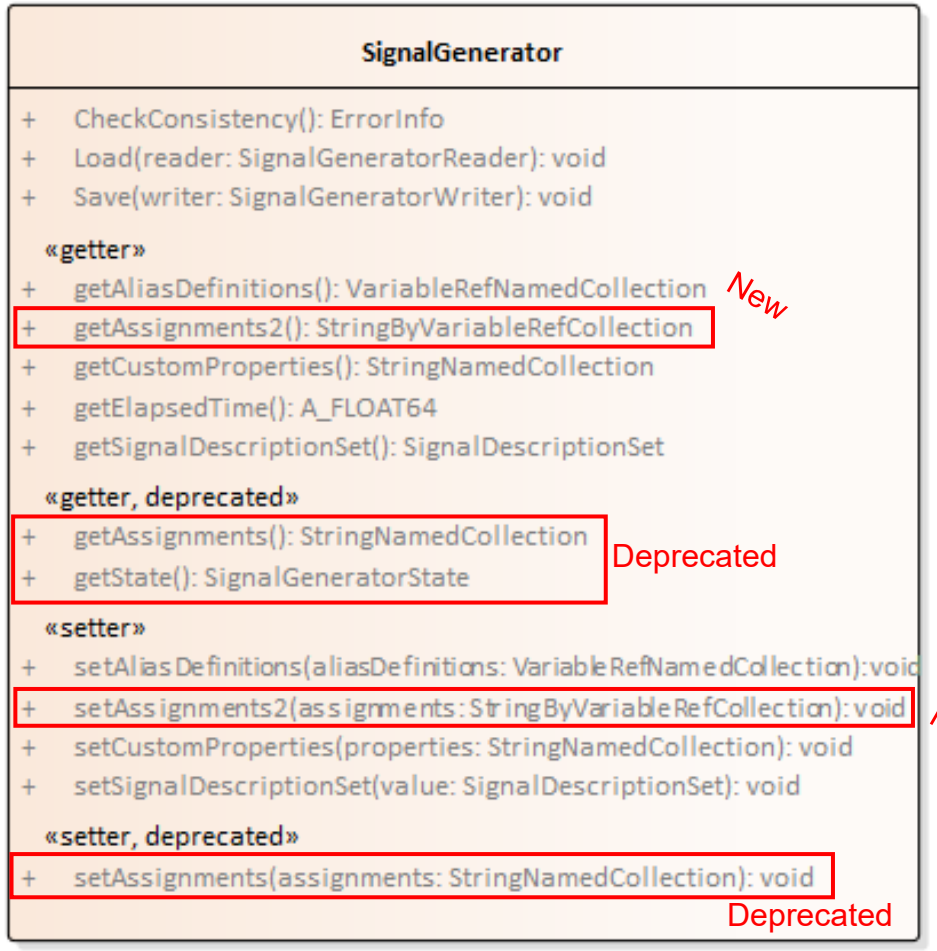
Goal

Support the capturing of complex data types

- The new `CaptureSignalGroup` interface is now the interface to retrieve data from `CaptureResult`
- The new interface can handle scalar values as well as complex structures like **vector** and **matrix**
- The `getSignalInfos` function in the `CaptureSignalGroup()` interface enables the user also to query metadata like data type, unit, CompuMethod and more
- The CaptureResult can return raw values as well as physical values

Signal Generation Support For Complex Data Types 1/2

Improvement



- The `SignalGenerator` interface is now able to assign a SignalDescription of a SignalDescriptionSet to more than one variable

- **ASAM XIL 2.1**

  SignalDescriptorSet1
  - SignalDescription_1 → Variable_1
  - SignalDescription_2
  - SignalDescription_3 → Variable_3

- **ASAM XIL 2.2**

  SignalDescriptorSet1
  - SignalDescription_1 → Variable_1
  - SignalDescription_2 → Variable_2
  - SignalDescription_3 → Variable_3, Variable_4, Variable_5 (New)

ASAM

Signal Generation Support For Complex Data Types 2/2

New

SignalGenerator

+ CheckConsistency(): ErrorInfo
+ Load(reader: SignalGeneratorReader): void
+ Save(writer: SignalGeneratorWriter): void

«getter»

+ getAliasDefinitions(): VariableRefNamedCollection — New
+ getAssignments2(): StringByVariableRefCollection
+ getCustomProperties(): StringNamedCollection
+ getElapsedTime(): A_FLOAT64
+ getSignalDescriptionSet(): SignalDescriptionSet

«getter, deprecated»

+ getAssignments(): StringNamedCollection — Deprecated
+ getState(): SignalGeneratorState

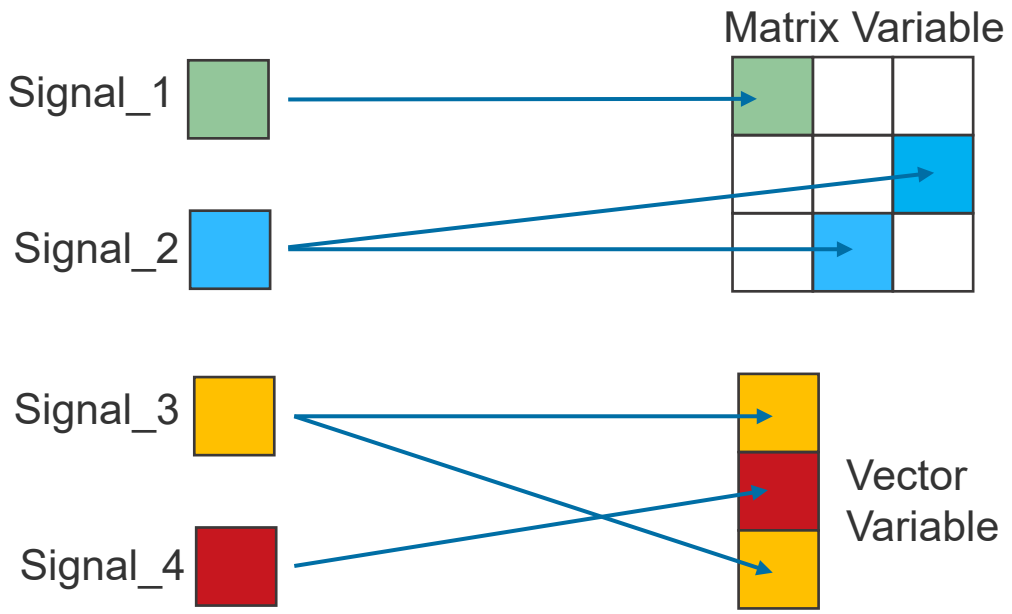
«setter»

+ setAliasDefinitions(aliasDefinitions: VariableRefNamedCollection):void
+ setAssignments2(assignments: StringByVariableRefCollection): void — New
+ setCustomProperties(properties: StringNamedCollection): void
+ setSignalDescriptionSet(value: SignalDescriptionSet): void

«setter, deprecated»

+ setAssignments(assignments: StringNamedCollection): void — Deprecated

- It‘s now also possible to stimulate single vector/matrix elements by signals in a `SignalGenerator`



- The `SignalGenerator` interface can now also stimulate variables with raw values or physical values

Extension Of The Framework Configuration File 1/2

Improvement

PortDefinition type	Permitted target states	Parameters
MAPortDefinition	eDISCONNECTED	---
	eSIMULATION_STOPPED	New PortConfigurationFile
	eSIMULATION_RUNNING	New ForceConfig
DiagPortDefinition	eDISCONNECTED	---
	eCONNECTED	PortConfigurationFile
EESPortDefinition	eDISCONNECTED	---
	eCONNECTED	PortConfigurationFile
	eDOWNLOADED	New PortConfigurationFile
	eACTIVATED	New ErrorConfigurationFile
ECUCPortDefinition	eDISCONNECTED	---
	eOFFLINE	PortConfigurationFile
	eONLINE	New PortConfigurationFile
		New LoadingType
ECUMPortDefinition	eDISCONNECTED	---
	eMEASUREMENT_STOPPED	PortConfigurationFile
	eMEASUREMENT_RUNNING	
NetworkPortDefinition	eDISCONNECTED	---
	eSTOPPED	PortConfigurationFile
	eRUNNING	

- Modification of the framework configuration file to add new parameters
- New parameters enable the framework to drive all XIL ports to all states without user interaction or default values
- New Parameters
  - ForceConfig
  - ErrorConfigurationFile
  - LoadingType

Extension Of The Framework Configuration File 2/2

Improvement

ASAM XIL 2.1

```
<MAPortDefinition InstanceName="MaPortRunning" InitOrder="2" ShutdownOrder="2" TargetState="eSIMULATION_RUNNING">
    <VendorName>VendorXy</VendorName>
    <ProductName>ProductAb</ProductName>
    <ProductVersion>20198-B.2</ProductVersion>
    <PortConfigurationFile>MAPortConfig.xml</PortConfigurationFile>
</MAPortDefinition>
```

ASAM XIL 2.2

```
<MAPortDefinition InstanceName="MaPortRunning" InitOrder="2" ShutdownOrder="2">
    <VendorName>VendorXy</VendorName>
    <ProductName>ProductAb</ProductName>
    <ProductVersion>2019-B.2</ProductVersion>
    <TargetState>
        <eSIMULATION_RUNNING>
            <PortConfigurationFile>MAPortConfig.xml</PortConfigurationFile>
            <ForceConfig>false</ForceConfig>
        </eSIMULATION_RUNNING>
    </TargetState>
</MAPortDefinition>
```

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	**Other Changes**
5	Backward-Compatibility
6	Deliverables

Deprecation Marking In Source Code Files

New

ASAM XIL 2.1

```
/// <summary>
/// Creates a MAPortBreakpoint object.
/// </summary>
/// <param name="watcher">Breakpoint condition (regarding elapsed simulation time
/// or certain values of simulation variables) in form of a Watcher.</param>
/// <param name="action">Action to be automatically performed when simulation
/// reaches the breakpoint, i.e when the breakpoint condition is met.</param>
int CreateMAPortBreakpoint(IWatcher watcher, BreakpointAction action);
```

ASAM XIL 2.2

```
/// <summary>
/// Creates a MAPortBreakpoint object.
/// </summary>
/// <param name="watcher">Breakpoint condition (regarding elapsed simulation time
/// or certain values of simulation variables) in form of a Watcher.</param>
/// <param name="action">Action to be automatically performed when simulation
/// reaches the breakpoint, i.e when the breakpoint condition is met.</param>
[Obsolete("This method is deprecated and might be removed in a future version of
the standard. So XIL clients are strongly recommended to use the replacement
specified in the API documentation and the Programmer's Guide Appendix.", false)]
int CreateMAPortBreakpoint(IWatcher watcher, BreakpointAction action);
```

- Deprecated functions are now also marked in the source code files as deprecated

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	**Backward Compatibility**
6	Deliverables

Backward Compatibility

Affected Component	Deprecated Element	Replacement
Attributes (Package Testbench.Common.ValueContainer)	Complete interface	Interface **VariableInfo** and its Port specific derivations (e.g. **MAPortVariableInfo** from package Testbench.MAPort) for predefined Attributes “Name”, “Description” and “Unit”. No replacement for user defined Attributes.
BaseValue (Package Testbench.Common.ValueContainer)	Property **Attributes**	Method **GetVariableInfo** of the Port interfaces for predefined Attributes “Name”, “Description” and “Unit”. No replacement for user defined Attributes.
	Property **Type**	Properties **ContainerType** and **ElementType**
Capture (Package Testbench.Common.Capturing)	Property **DurationUnit**	Implicitly set by passing a **TimeSpanDuration** or **CycleNumberDuration** object (from package Testbench.Common.Duration) to SetStartTrigger, SetStopTrigger and the DurationWatcher factory methods.
	Method **SetStartTriggerCondition**	Method **SetStartTrigger**
	Method **SetStopTriggerCondition**	Method **SetStopTrigger**
	Property **Variables**	Property **Variables2**
CaptureResult (Package Testbench.Common.CaptureResult)	Method **ExtractSignalValue**	Methods **GetTimestamps**, **GetSignalValues** and **GetScalarSignalValues** of the **CaptureSignalGroup** interface obtainable via the **SignalGroups** property
	Method **GetSignalGroupValue**	Property **SignalGroups**

- Deprecated elements are listed in Appendix F. – Deprecated Elements of the ASAM XIL 2.2 Programmers Guide
- The API elements listed in the second column of the table are deprecated and might be removed in a future version of the standard.
- So it is recommended to use the replacement depicted in the third column.

Agenda

1	Introduction
2	Motivation for New Release
3	New Features
4	Other Changes
5	Backward-Compatibility
6	**Deliverables**

Deliverable Content

Documents

- Content Overview
- Generic UML Model (Enterprise Architect)
- Programmers Guide

Supplementary Files

- Schemas (incl. examples)
  - EES_Configuration
  - FrameworkConfiguration
  - ImplementationManifest
  - Mapping
  - StimulusSignalDescription
- Technology Reference for C#
  - C# Interfaces
  - C# Examples
  - C# Test Suite
- Technology Reference for Python
  - Python Interfaces
- XIL Standard Assemblies (MSI Installer)