

Release Presentation

ASAM AE ATX (Automotive Test Exchange Format)

2012-07 (ASAM TSC Meeting)



Deliverables

Deliverable	Acceptance Criteria	Responsible
XML Schema(Maybe this can be generated by model2schema convert, e.g. autosar schema generator.)	Formal Acceptance by Project Team	Supplier
Several XML Example Test Cases	Formal Acceptance by Project Team	Project Team
Schema Reference Guide	Formal Acceptance by Project Team	Supplier
ATX User's Guide	Formal Acceptance by Project Team	Supplier

Introduction

ATX = Automotive Test Exchange Format

ATX is a standardized test exchange format, which allows reusing existing test cases in different test automation software systems.

The main benefits are:

- Development of test system is independent from test libraries (efficiency increase on end user side)
- Reduction of compatibility problems of the test automation software. Instead of n connectors for n systems to interface only one implementation is needed (efficiency increase on tool supplier side)

Introduction

Main Activities Supported by ATX

Automotive Test Exchange Format

- **Test Specification**

ATX supports the creation of manual and automated test specifications.

It enables the specification of high level(logical) and low level(concrete) test cases.

- **Test Preparation**

ATX supports the definition test data(parameter values) and test execution plans.

It support the definition of test data for ECUs like curves or maps as used in ASAM 2MC A2L, ASAM AE MDX or AUTOSAR.

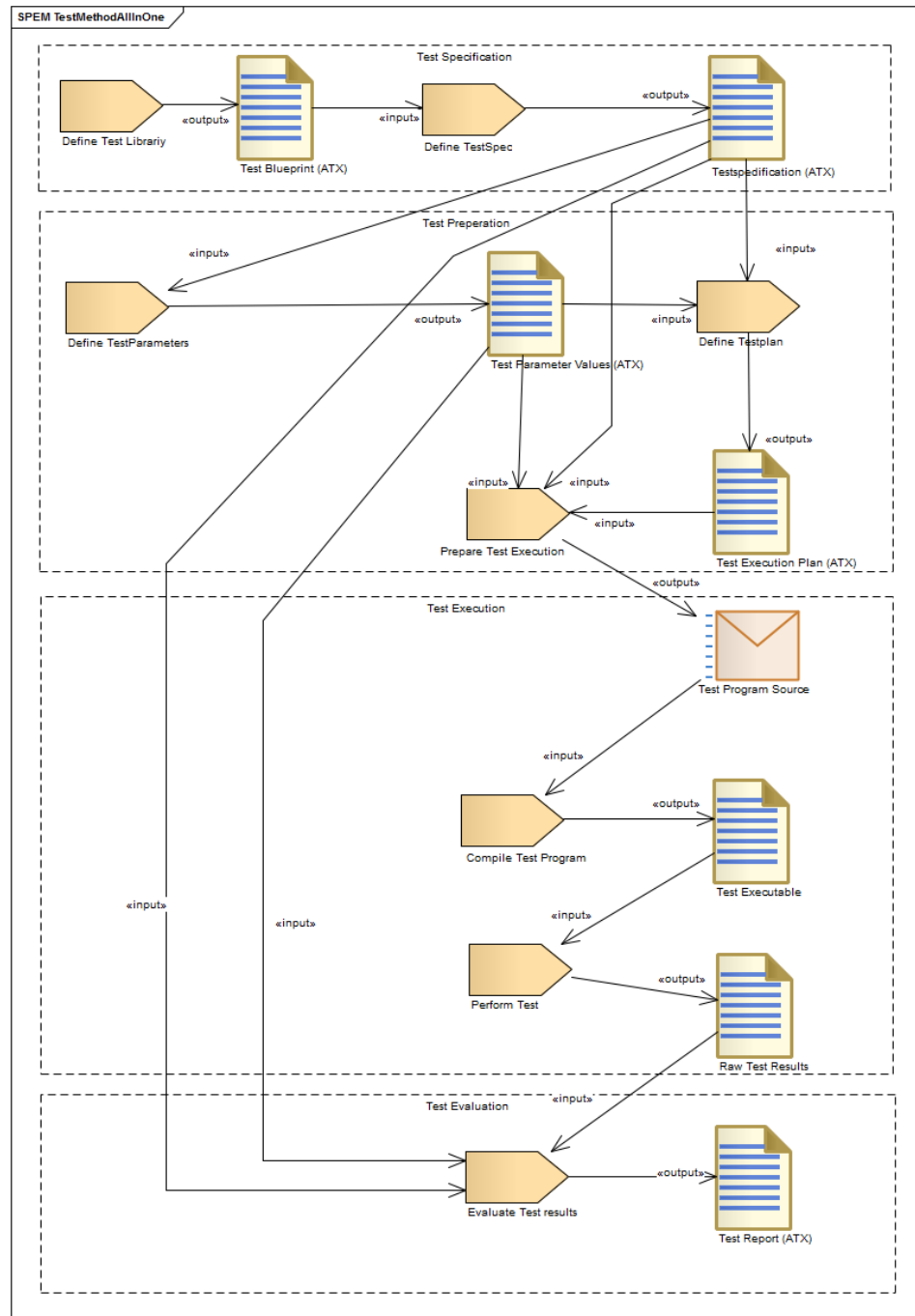
- **Test Execution**

ATX can define data needed to generate test program skeletons or fully automated test programs.

- **Test Analysis and Test Report**

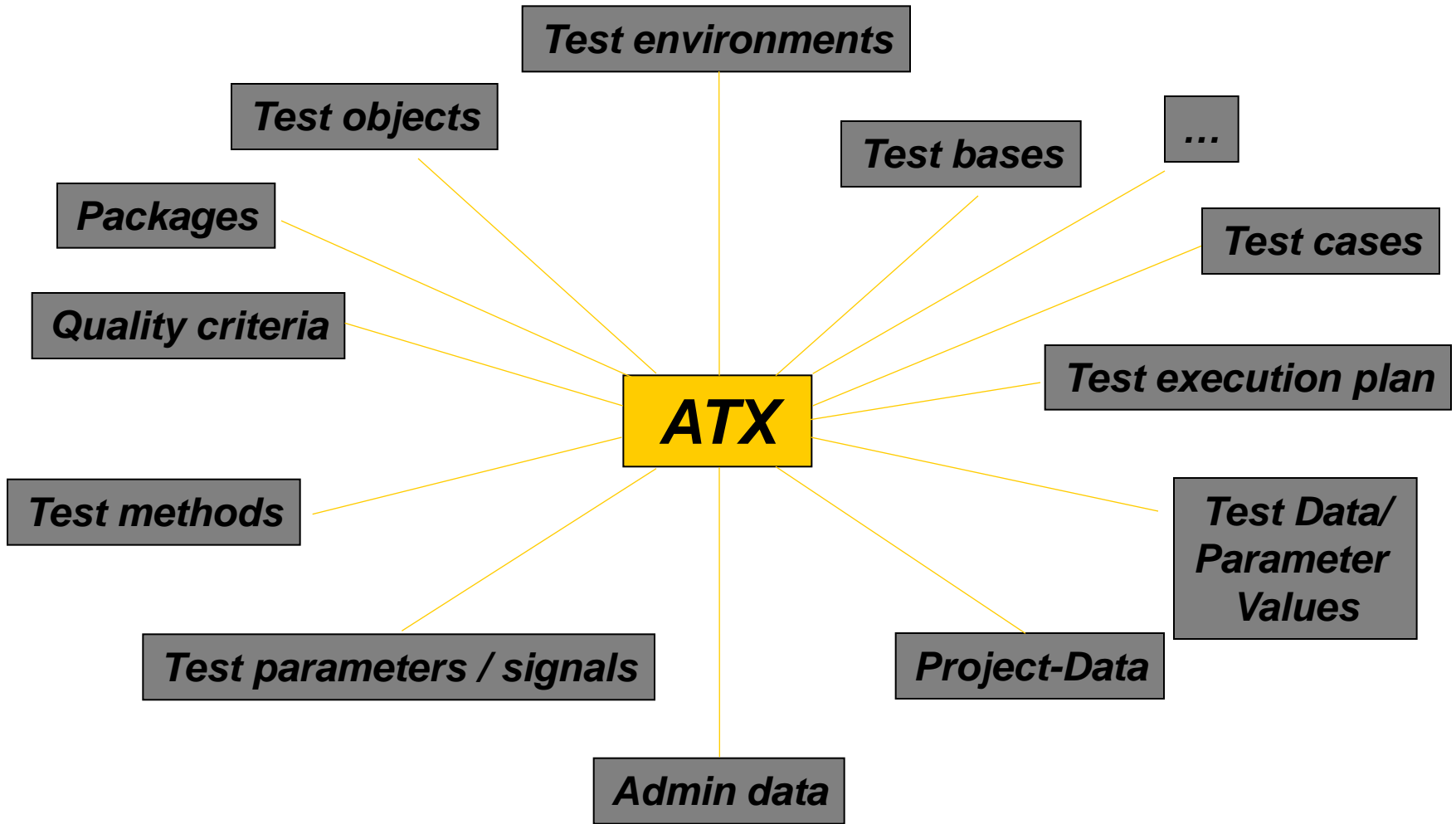
ATX has enables the automated verdict evaluation and test report creation.

Introduction



Introduction

ATX main content objects



Introduction

ATX organizes its contents in encapsulated packages

- ▲ <> Package: MyComp
 - ▲ <> Package: MyProj
 - ▲ <> Package: AirCondition
 - ▷ <> Package: TestBases
 - ▷ <> Package: TestEnvs
 - ▷ <> Package: TestObjects
 - ▷ <> Package: TestSpecs
 - ▷ <> Package: TestExecutionPlans
 - ▷ <> Package: TestData
 - ▷ <> Package: TestReports
 - ▷ <> Package: ApplicationDataTypes
 - ▲ <> Package: ImplementationDataTypes
 - ▷ <> Package: ImplementationDataTypes_CSharp
 - ▷ <> Package: ImplementationDataTypes_Java
 - ▷ <> Package: ImplementationDataTypes_OTX
 - ▲ <> Package: SwBaseTypes
 - ▷ <> Package: SwBaseTypes_CSharp
 - ▷ <> Package: SwBaseTypes_Java
 - ▷ <> Package: SwBaseTypes_OTX
 - ▷ <> Package: DataTypeMappings
 - ▷ <> Package: TestActionImpls
 - ▲ <> Package: ASAM
 - ▷ <> Package: ISO_9126_1_QualityAttributes
 - ▷ <> Package: ISTQB_TestTechniques
 - ▷ <> Package: AsamDataTypes

Introduction

ATX supports the grouping of test cases ...

- ▲ <> TEST-SPEC: AirConditionTestSpec
 - ▷ <> TEST-CASE: AirConditionLoadTorque
 - ▷ <> TEST-CASE: AirConditionLoadTorquePrg
 - ▲ <> TEST-CASE-FOLDER: IntegrationTests
 - ▲ <> TEST-CASE-FOLDER: SpecificationBasedTest
 - <> TEST-CASE: SpecificationBasedTest1
 - <> TEST-CASE: SpecificationBasedTest2
 - <> TEST-CASE: SpecificationBasedTest3
 - <> TEST-CASE: SpecificationBasedTest4
 - ▲ <> TEST-CASE-FOLDER: AnalysisBasedTest
 - <> TEST-CASE: AnalysisBasedTest1
 - <> TEST-CASE: AnalysisBasedTest2
 - <> TEST-CASE: AnalysisBasedTest3
 - <> TEST-CASE: AnalysisBasedTest4

and the grouping of test actions

- ▲ <> TEST-SETUP-STEPS
 - ▲ <> TEST-STEP: InitializationPrg
 - <> TEST-ACTION: Clamp15_On
 - <> TEST-ACTION: StartEnginePrg
 - ▲ <> TEST-ACTION-FOLDER: InitializeValves
 - <> TEST-ACTION: InitializeValve1
 - <> TEST-ACTION: InitializeValve2
 - <> TEST-ACTION: InitializeValve3
 - <> TEST-ACTION: InitializeValve4
 - <> TEST-ACTION: WaitIdleControllerSteadyStatePrg
 - ▲ <> TEST-EXECUTION-STEPS

Introduction

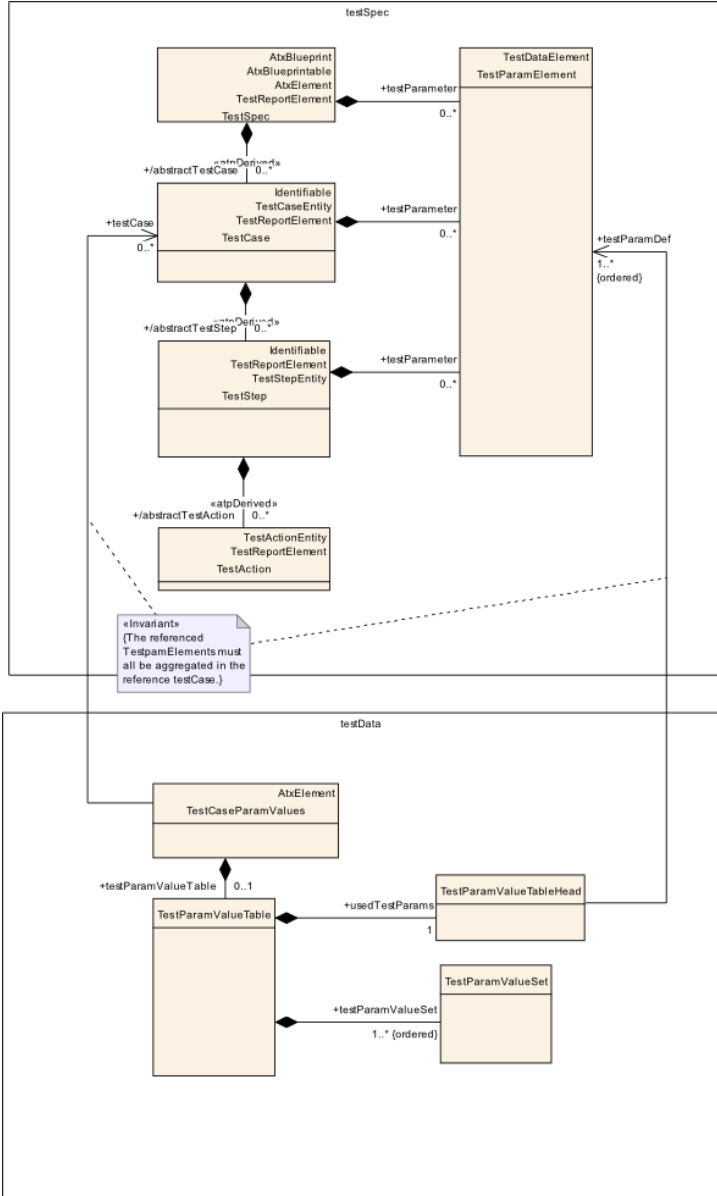
- ATX Test-Spec is a container for test cases. (It is not the execution plan!)
- Test cases have 3 execution phases: test setup, test execution, test teardown.
- Test steps consist of a sequence of test actions or test action folders.

```

  <> TEST-SPEC: AirConditionTestSpec
    <> TEST-CASE: AirConditionLoadTorque
      <> TEST-SETUP-STEPS
        <> TEST-STEP: Initialization
      <> TEST-EXECUTION-STEPS
        <> TEST-STEP: AirConditionMaxLoadTorqueMinus40Nm
          <> TEST-ACTION: SetTorqueMinus40Nm
          <> TEST-ACTION: SwitchOnAirCondition
          <> TEST-ACTION: wait10Sec
          <> TEST-ACTION: CheckIdleController800Rpm
        <> TEST-STEP: SwitchOff
          <> TEST-ACTION: SwitchOffAirCondition
          <> TEST-ACTION: WaitIdleControllerSteadyState2
        <> TEST-STEP: IncreaseLoadTorqueStepwise
          <> TEST-ACTION: SetTorqueMinus0Nm
          <> TEST-ACTION: SwitchOnAirCondition2
          <> TEST-ACTION: wait10Sec2
          <> TEST-ACTION-FOLDER: IncreaseLoadTorqueMinus10NmEvery2Sec
            <> TEST-ACTION: IncreaseLoadTorqueMinus10Nm
            <> TEST-ACTION: wait2Sec
          <> TEST-ACTION: wait10Sec3
          <> TEST-ACTION: CheckIdleController800Rpm2
      <> TEST-TEARDOWN-STEPS
        <> TEST-STEP: Finalize
          <> TEST-ACTION: SwitchOffAirCondition
          <> TEST-ACTION: StopEngine
    <> TEST-CASE: AirConditionLoadTorquePrg
    <> TEST-CASE-FOLDER: IntegrationTests
  
```

Introduction: Test Parameter, Test Parameter Values

Test parameters can be defined for test specs, test cases and test steps.



ATX allows the definition of test parameter value tables which fit to a test case.

Introduction: Test Execution Plan

Test execution plan identifies the test cases to be executed, associates the parameter value sets and defines the execution repetitions.

- ▲ <> 5 TEST-EXECUTION-PLAN: (AirConditionTestExecutionPlan1)
 - ▲ <> 5.1 PLANNED-TEST-CASE: (TC001_AirConditionLoadTorque)
 - <> 5.1.1 TEST-CASE-REF: (AirConditionLoadTorque)
 - ▲ <> 5.2 PLANNED-TEST-CASE: (TC002_AirConditionLoadTorquePrg)
 - <> 5.2.1 TEST-CASE-REF: (AirConditionLoadTorquePrg)
 - ▲ <> 5.2.2 PLANNED-TEST-CASE-DATAS: ()
 - <> 5.2.2.1 TEST-VALUE-SET-REF: (TestParamValues/LoadTorqueMaxMinus40)
 - <> 5.2.2.2 TEST-VALUE-SET-REF: (TestParamValues/LoadTorqueMaxMinus60)

The test report of the example might list the following records:

TC001_001 Test Case: AirConditionLoadTorque, Test Data Value Set: -, RunNo:1

TC002_001 Test Case: AirConditionLoadTorquePrg, Test Data Value Set: LoadTorqueMaxMinus40, RunNo:1

TC002_002 Test Case: AirConditionLoadTorquePrg, Test Data Value Set: LoadTorqueMaxMinus40, RunNo:2

TC002_003 Test Case: AirConditionLoadTorquePrg, Test Data Value Set: LoadTorqueMaxMinus60, RunNo:1

Compatibility

ATX uses the terminology defined by ISTQB.

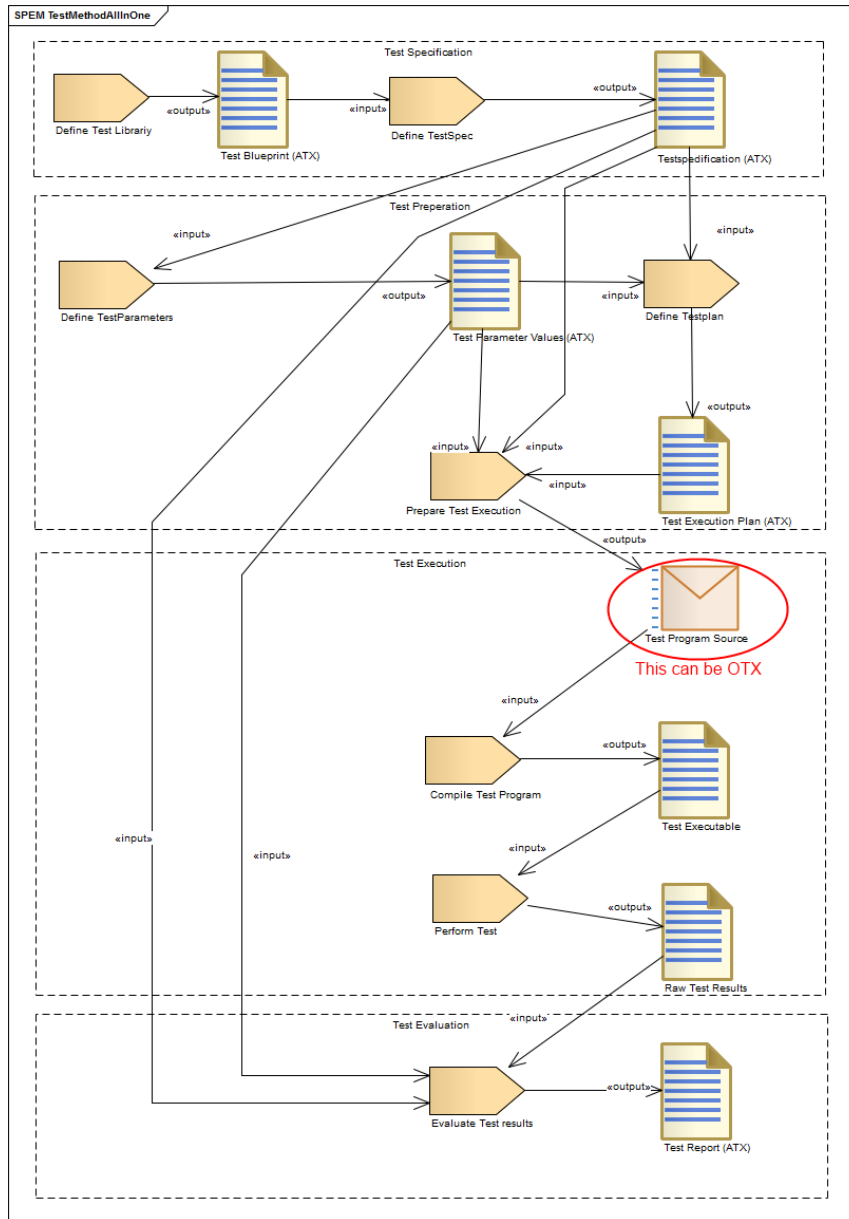
ATX reuses patterns and xml structures defined by the following standards:

- ASAM Harmonized Data Objects (ASAM-HDO)
- ASAM AE Functional Specification Exchange Format (ASAM AE-FSX)
- AUTOSAR V4.0

Relation to ISO

No direct relation to ISO Standards.

ATX and OTX



ATX and OTX are on different levels of the same process.

OTX is a programming language and can be used to implement the test cases specified in ATX.

Comparison of typical data objects of ATX and OTX

ATX

Test environments

Test objects

Quality criteria

Test bases

Test cases

Test execution plan

Test methods

Packages

Test parameters / signals

Test Data/Parameter Values

Admin data

Project-Data

OTX

procedure

action

if

else

loop

flow

throw

specification

realisation

variable

parameter