EXPECTATIONS
MATTER.

WE'RE COMMITTED TO EXCEED YOURS

USING FORMAL SPECIFICATIONS TO SUPPORT MODEL BASED TESTING

ASDSPEC: A TOOL COMBINING THE BEST OF TWO TECHNIQUES

Rachid Kherrazi

6-4-2014 Grenoble

ETAPS
EUROPEAN JOINT CONFERENCE ON
THEORY & PRACTICE OF SOFTWARE

NSPYRE

PEOPLE INVOLVED

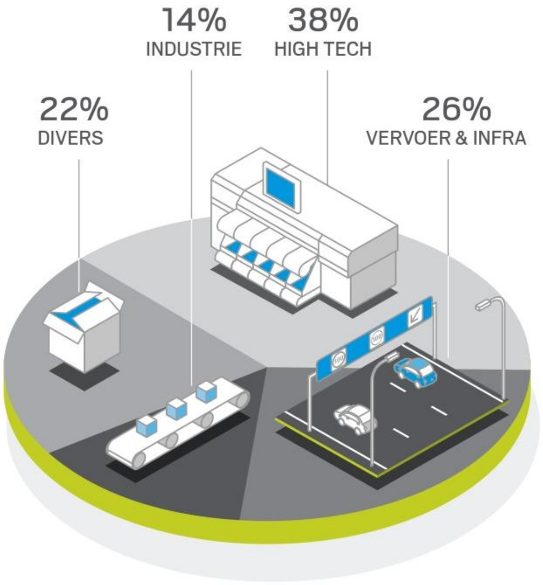
- Rachid Kherrazi
  - Senior Consultant @ Nspyre
  - Domain
    - Process and Product Improvement
    - RAMS (Reliability Availability Maintenance and Safety)
    - Model Based Testing and Model Driven Engineering
- Also : Arjan van Der Meer & Marc Hamilton

ABOUT NSPYRE

NSPYRE

MARKET SEGMENTS

- High Tech
- Traffic & Infra
- Industry



AREA’S OF EXCELLENCE

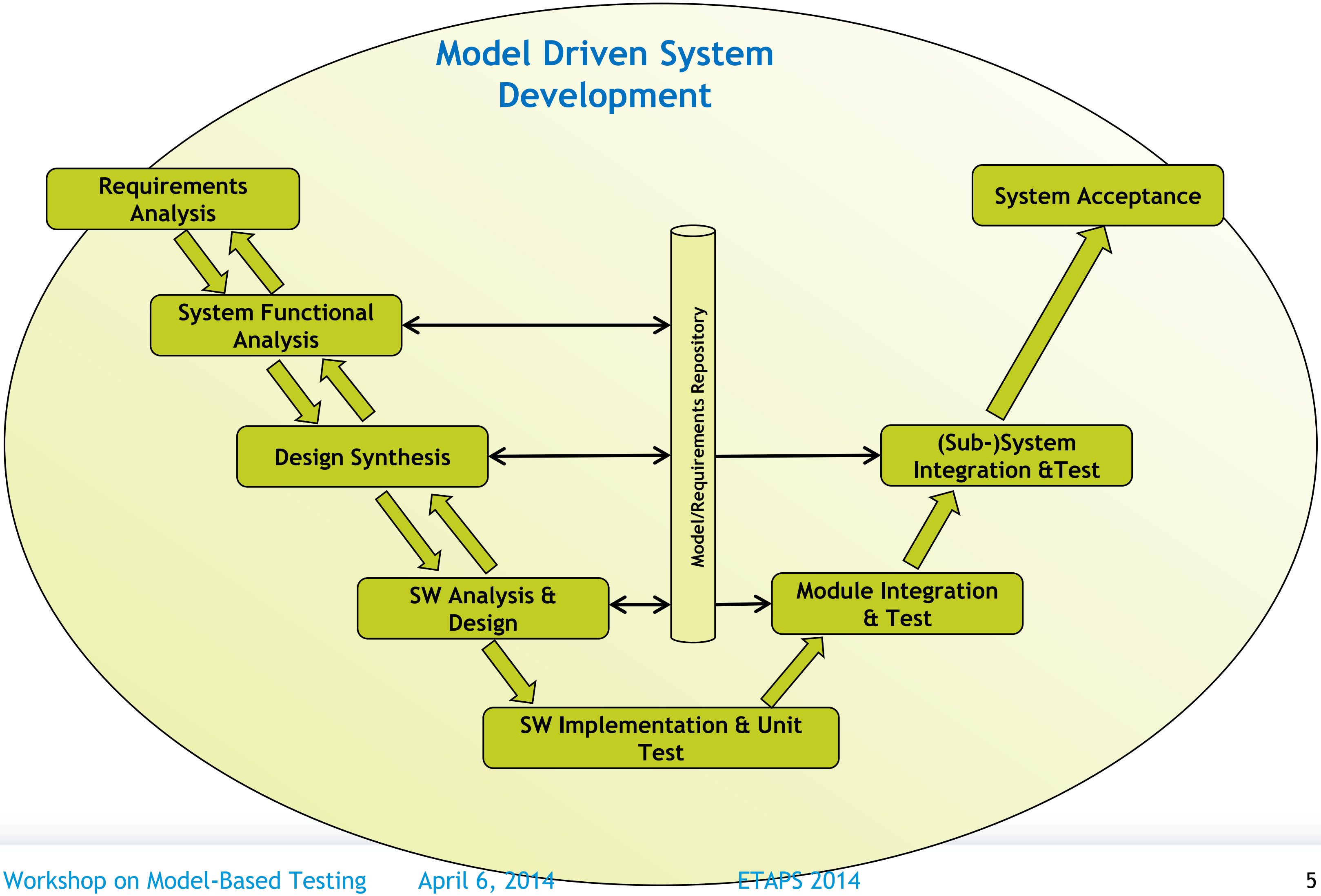
(Model Based) Systems Engineering /Model Driven Engineering / Model Based Testing /Industrial Automation / Simulation / Big Data / Mobile Solutions

CONTENTS

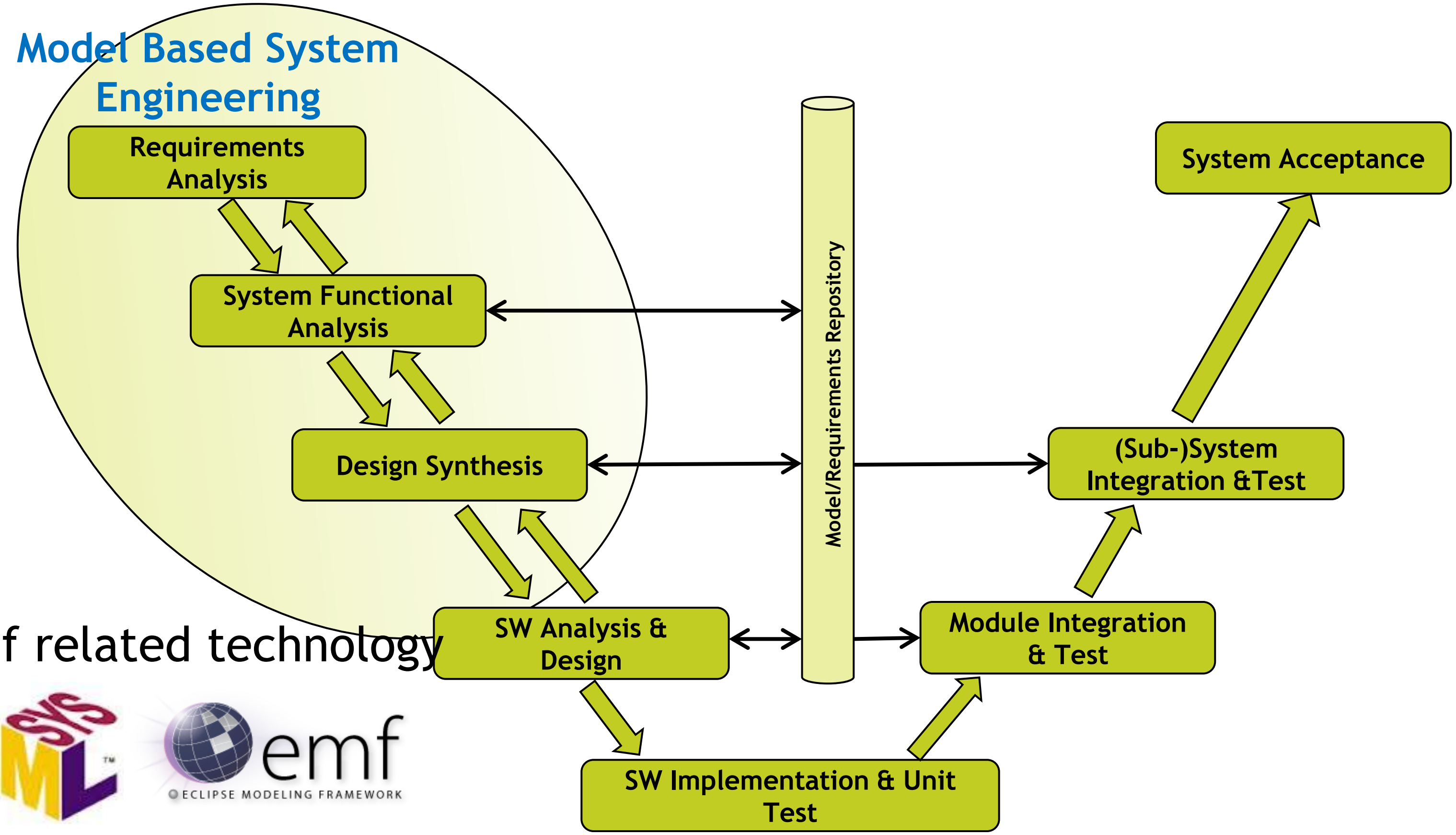
1) MDSD Introduction
2) MDE with ASD:Suite
3) MBT with MS Spec Explorer
4) ASDSpec

V-MODEL AND MODEL DRIVEN SYSTEM DEVELOPMENT

NSPYRE


Model Driven System Development
Requirements Analysis
System Functional Analysis
Design Synthesis
SW Analysis & Design
SW Implementation & Unit Test
Module Integration & Test
(Sub-)System Integration &Test
System Acceptance
Model/Requirements Repository

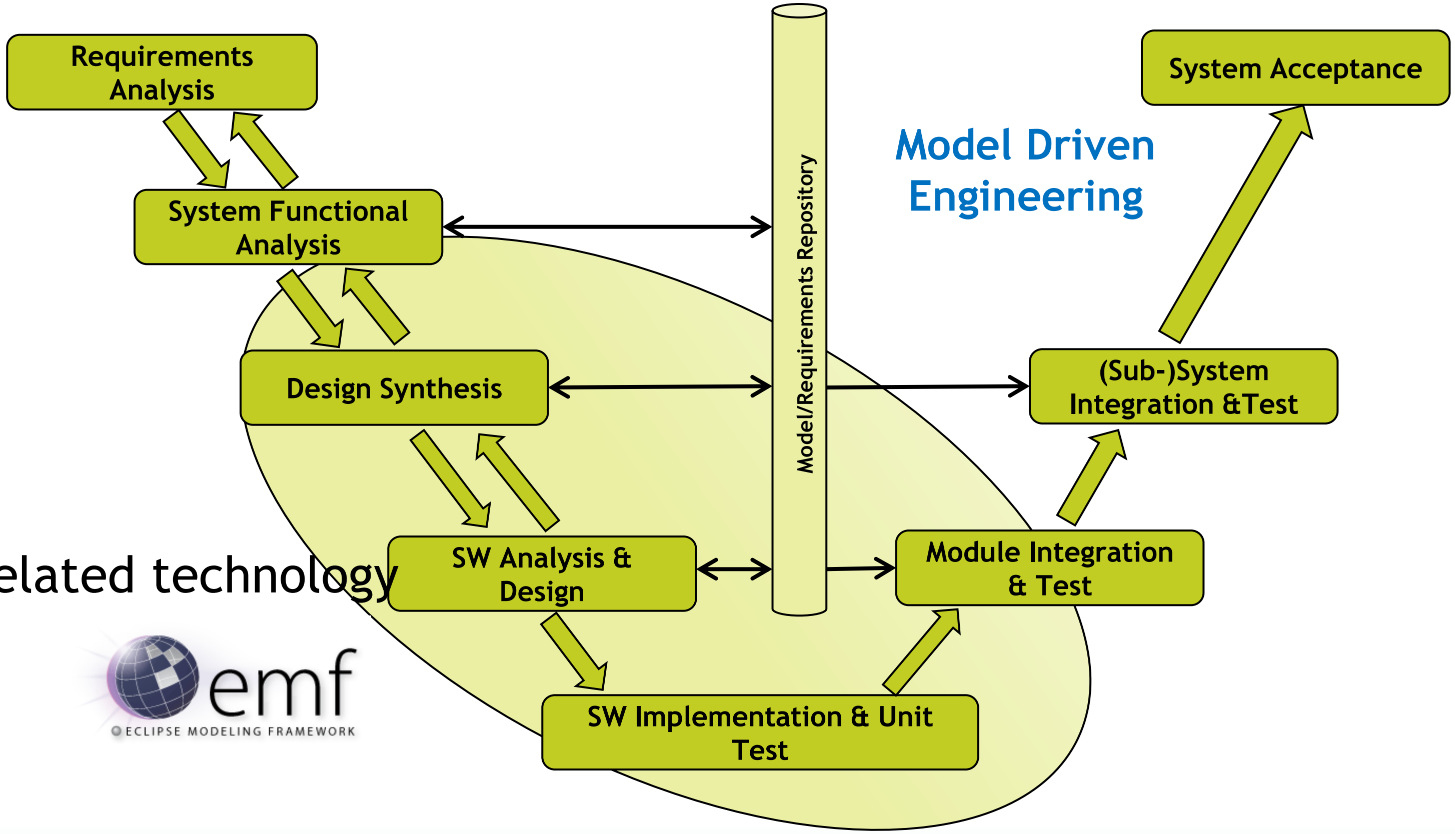
V-MODEL AND MODEL DRIVEN SYSTEM DEVELOPMENT


Model Based System Engineering
Requirements Analysis
System Functional Analysis
Design Synthesis
SW Analysis & Design
SW Implementation & Unit Test
Module Integration & Test
(Sub-)System Integration &Test
System Acceptance
Model/Requirements Repository

Some of related technology

V-MODEL AND MODEL DRIVEN SYSTEM DEVELOPMENT

NSPYRE

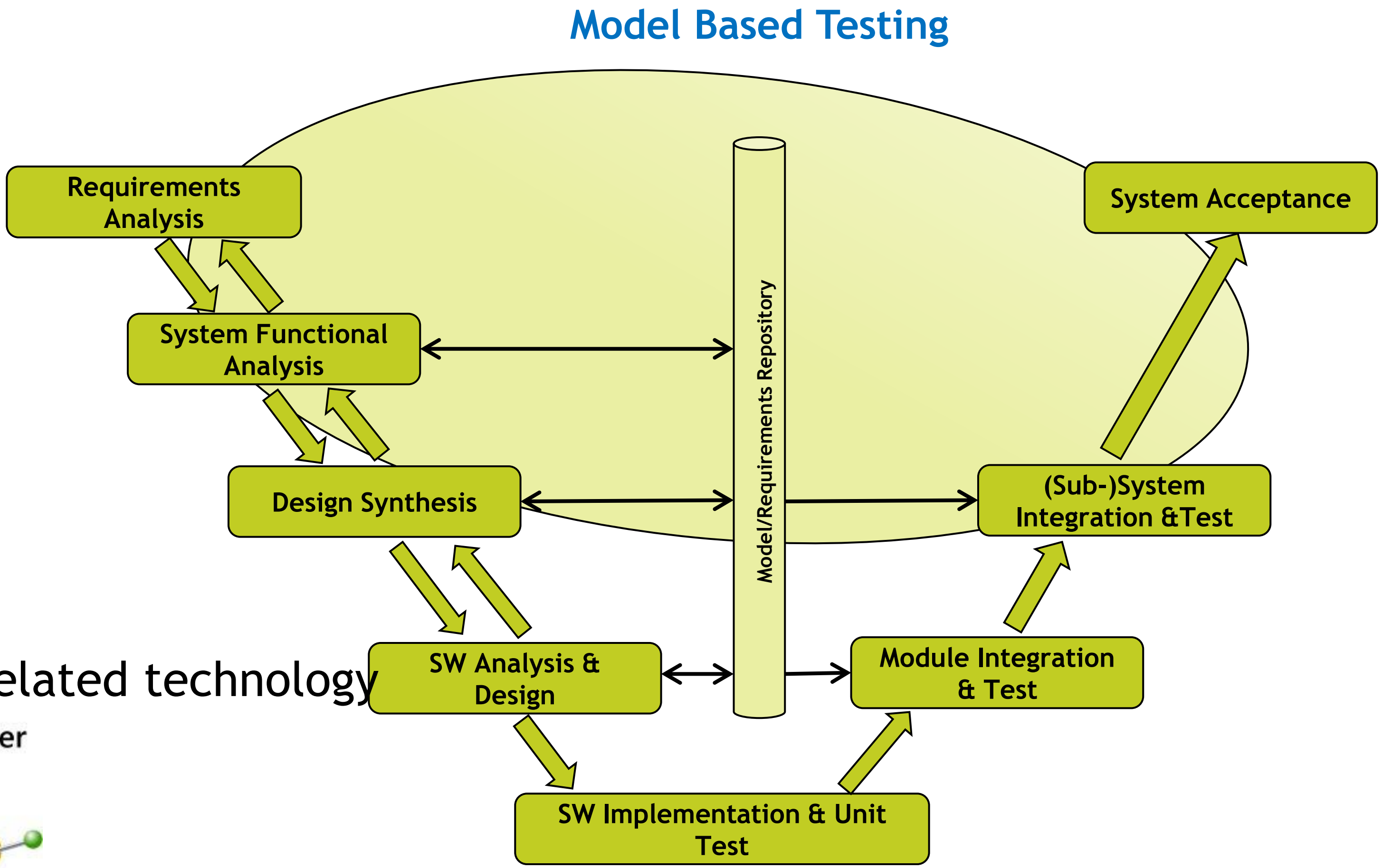

Requirements Analysis
System Functional Analysis
Design Synthesis
SW Analysis & Design
SW Implementation & Unit Test
Module Integration & Test
(Sub-)System Integration &Test
System Acceptance
Model/Requirements Repository
Model Driven Engineering

Some of related technology

ASD

emf ECLIPSE MODELING FRAMEWORK

V-MODEL AND MODEL DRIVEN SYSTEM DEVELOPMENT

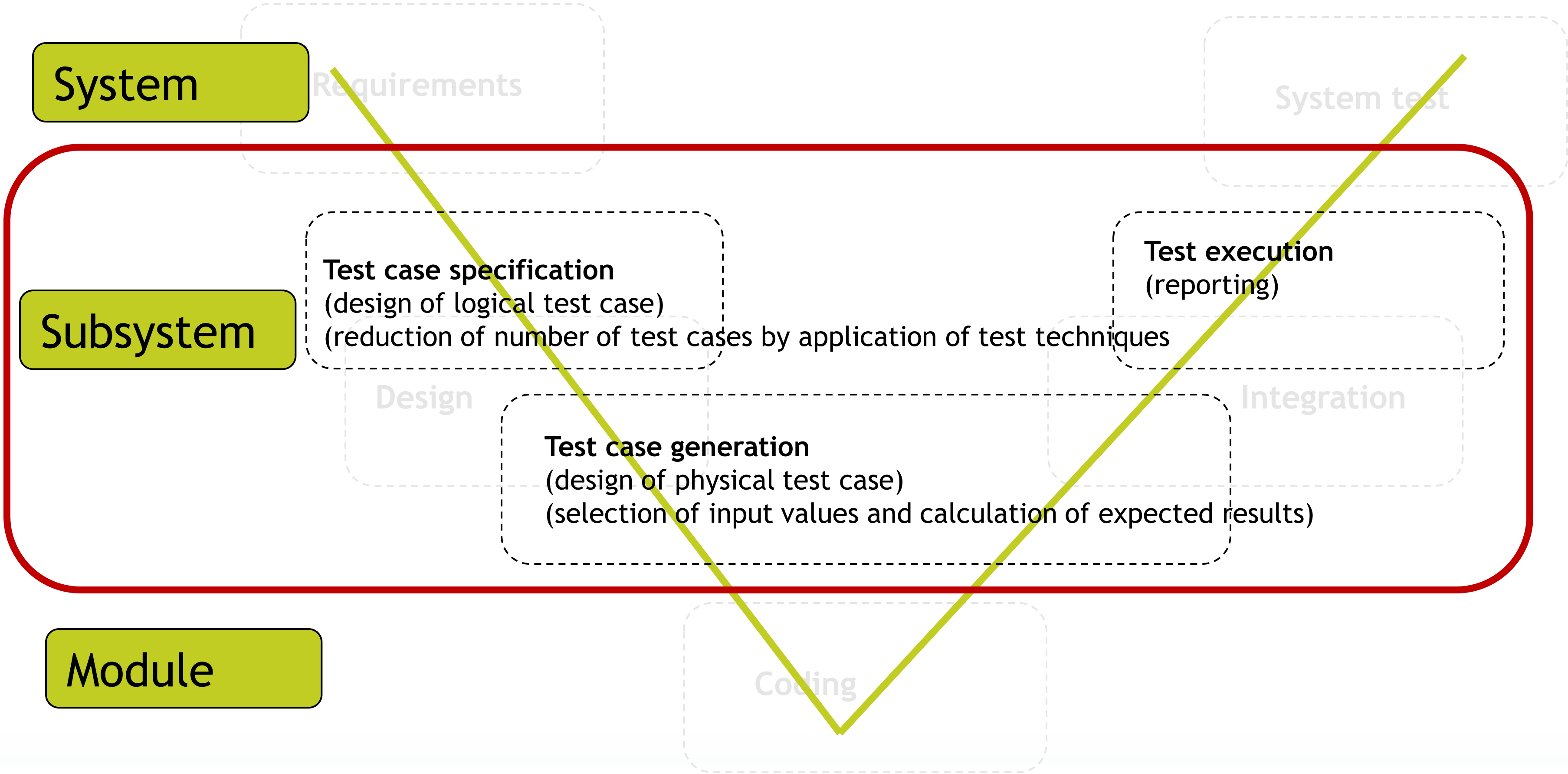

Model Based Testing
Requirements Analysis
System Functional Analysis
Design Synthesis
SW Analysis & Design
SW Implementation & Unit Test
Module Integration & Test
(Sub-)System Integration &Test
System Acceptance
Model/Requirements Repository

Some of related technology

Spec Explorer

V-MODEL FOR TESTING

3 main steps in test process

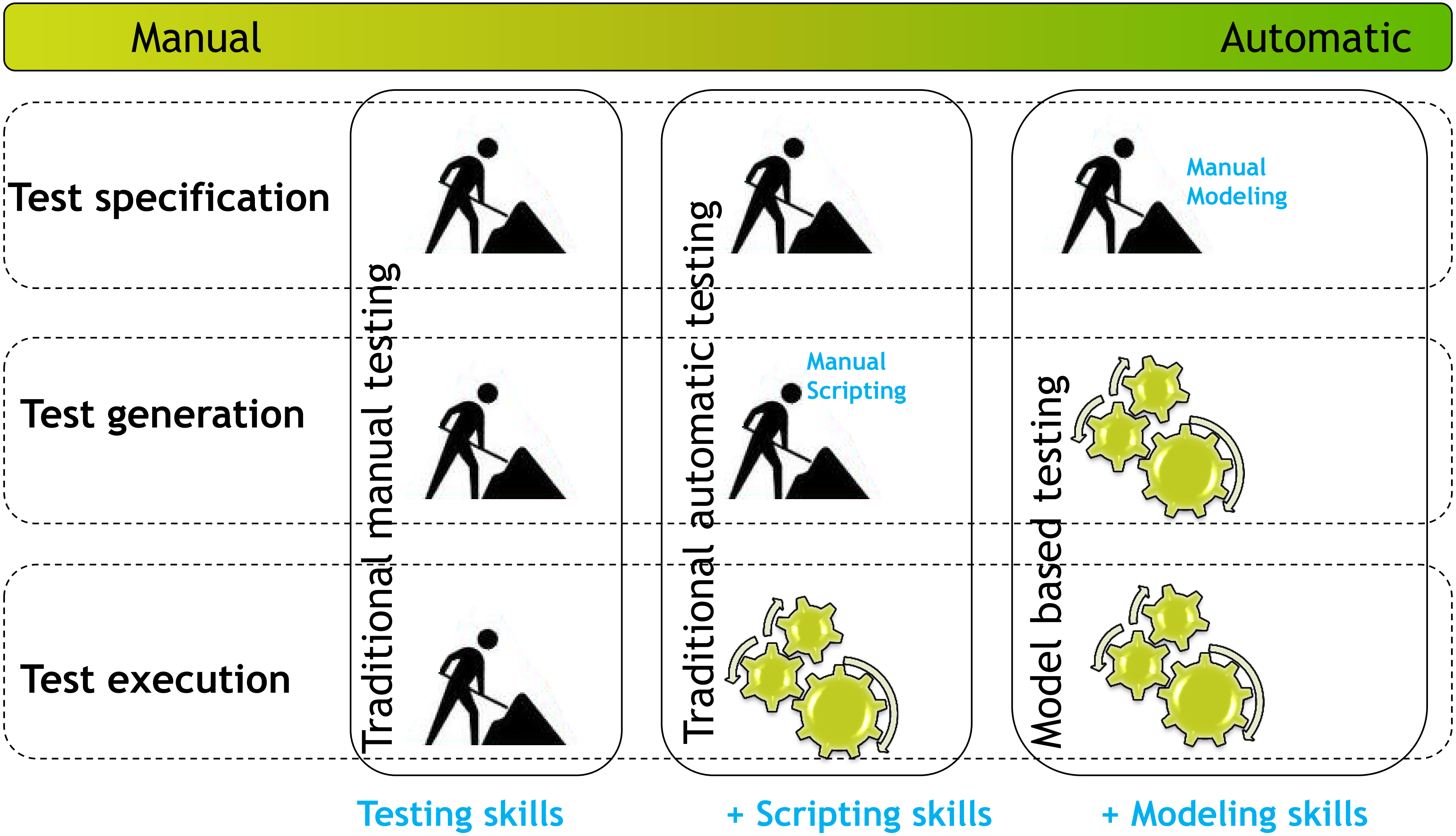


Test case specification
(design of logical test case)
(reduction of number of test cases by application of test techniques

Test execution
(reporting)

Test case generation
(design of physical test case)
(selection of input values and calculation of expected results)

MBT IS THE AUTOMATION OF TEST CASE GENERATION



Manual

Automatic

Test specification

Test generation

Test execution

Traditional manual testing

Traditional automatic testing

Manual Scripting

Model based testing

Manual Modeling

Testing skills

+ Scripting skills

+ Modeling skills

PERCEIVED BENEFITS

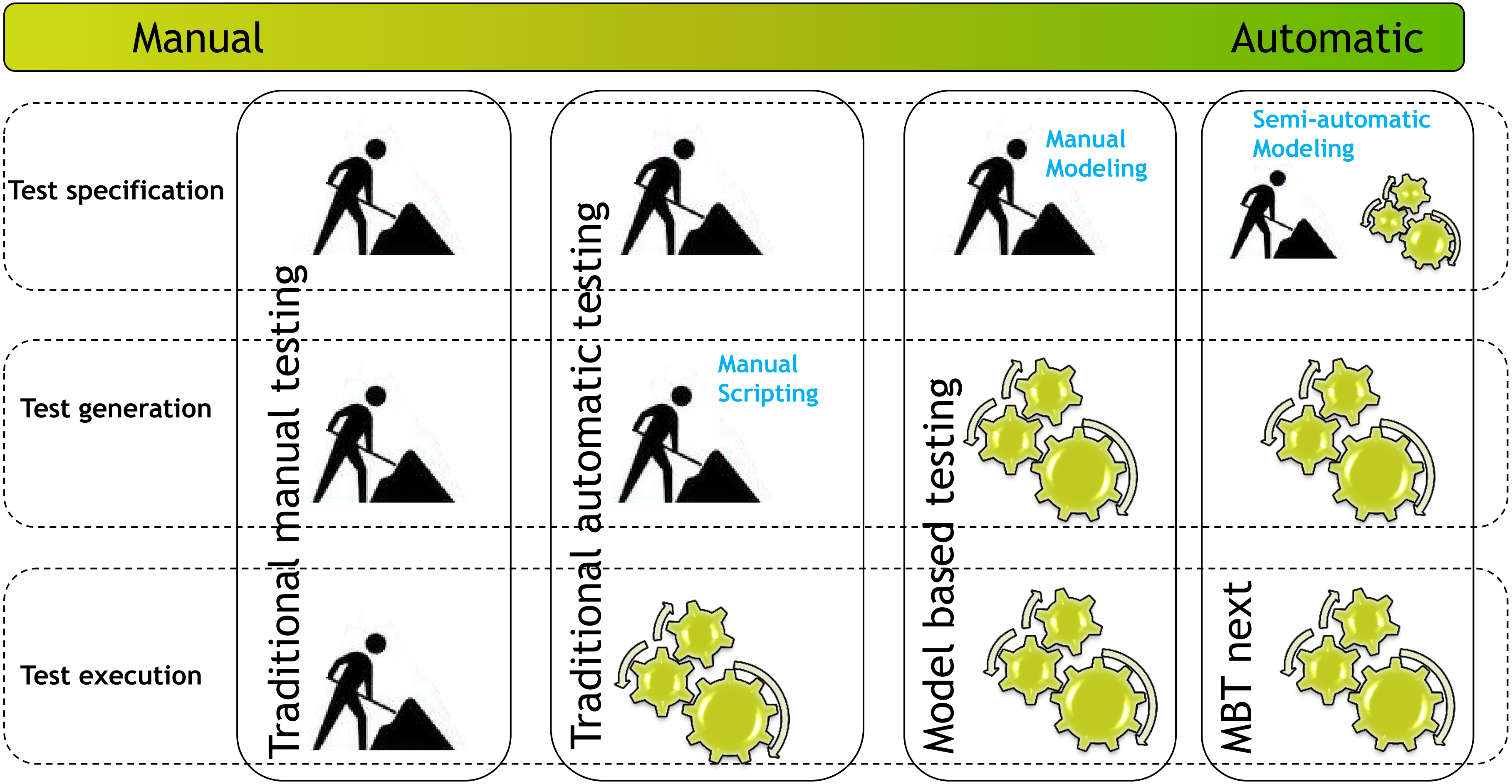
NSPYRE

- Increased productivity (increased automation)
- Better test script maintenance
- Improved product reliability (new type of bugs, Increased test coverage)
- Agility (Easily react to new feature changes, Reusability of test semantics, Early test engagement, Drive quality upstream)
- Increased employee satisfaction (challenging, new horizon, fun)

- Question: How can we improve MBT e.g. increase further the productivity?

MBT IS THE AUTOMATION OF TEST CASE GENERATION

NSPYRE



Manual

Automatic

Test specification

Test generation

Test execution

Traditional manual testing

Traditional automatic testing

Manual Scripting

Model based testing

Manual Modeling

MBT next

Semi-automatic Modeling

Testing skills

+ Scripting skills

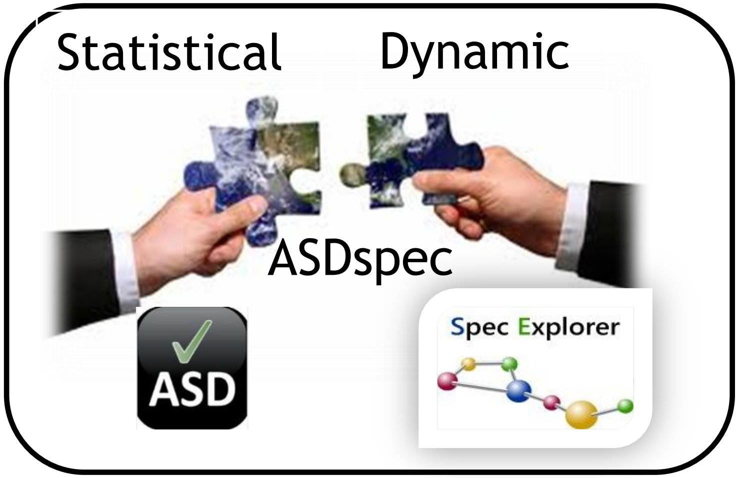
Modeling skills

IDEA: USING BEST OF BOTH WORLDS

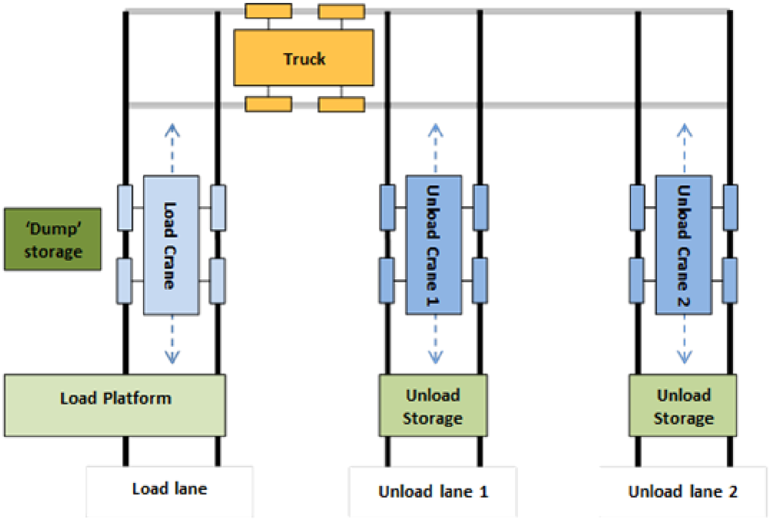
- Automatic generation of (partial) Spec Explorer test model from existing ASD interface model

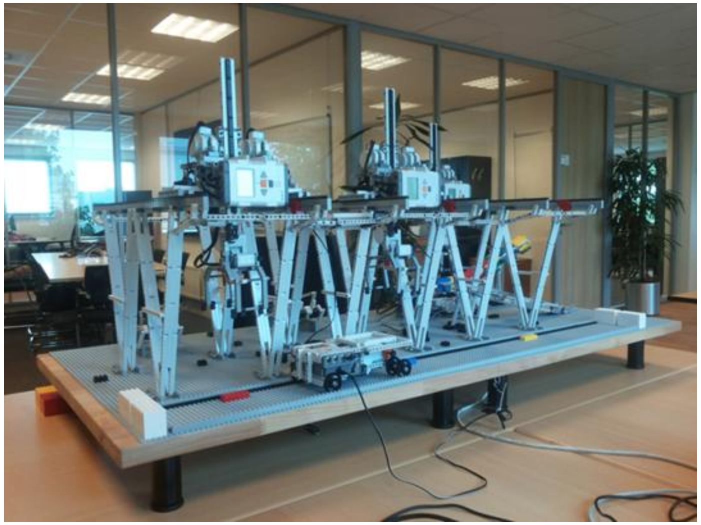
Benefits

- Less effort for model creation
  - reuse of existing work
- Testing of complete system including
  - Legacy code
  - External components
  - Data combination testing
  - Interaction testing
- Results: High Quality, Reduced cost



USE CASE: CONTAINER TERMINAL





- Multiple components
- Components need to interact to function
- Controller needed to coordinate interaction

OVERVIEW OF PROJECT STAGES

	Stage 1	Stage 2	Stage 3
Use case /SUT	Container Terminal (CT)	Container Terminal (CT)	Container Terminal (CT)
Goal	Development of control software for the CT	Verification of the developed control software of the CT (generated + Hand written)	Verification of the developed control software of the CT (generated + Hand written)
Technique	Model Driven Engineering + hand written sw	Model based Testing	Model based Testing
Tool	ASD:Suite	Spec Explorer	ASDspec & Spec Explorer
Method	Create design Models + interface models in ASD Generate code	Create Test Model in Spec Explorer, generate test suite	Generate Test Model from existing ASD interface Model , complete Test Model , generate test suite
Results	✓ Productivity (code generation) c.t. trad. dev ✗ Testing complete system, Interaction, data, external code	✓ Test Productivity ✗ Modeling skills, complexity costs	✓ Productivity (code generation, partial test generation) Testing complete system, Interaction, data, external code ! (benefits only in case of existing of ASD models)

CONTENTS

1) MDSD Introduction

2) MDE with ASD:Suite

3) MBT with MS Spec Explorer

4) ASDSpec

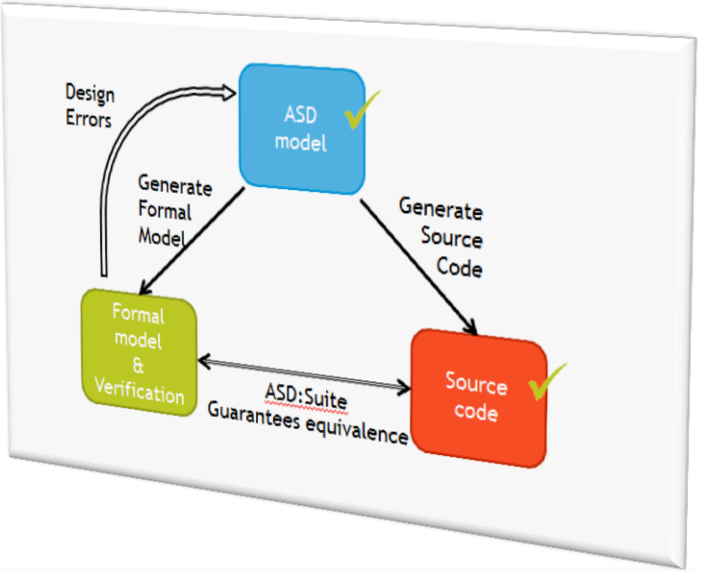
Company

Tool

ASD (ANALYTICAL SOFTWARE DESIGN)

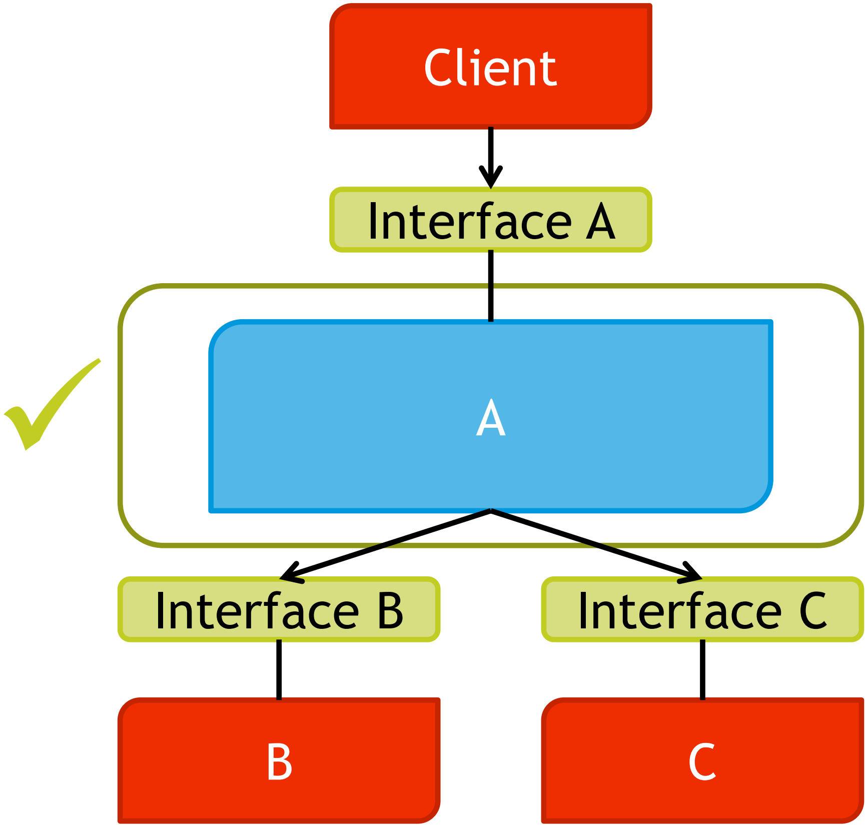
NSPYRE

- Model Driven Engineering (code generation)
- Component Based Development
- Models are verified mathematically at design time (formal methods)



COMPONENT BASED DEVELOPMENT ASD-STYLE

NSPYRE



Client component

Implemented service

ASD component

Used service

Used component

Component:
Decomposition
Specification
Design
Verification
Generation

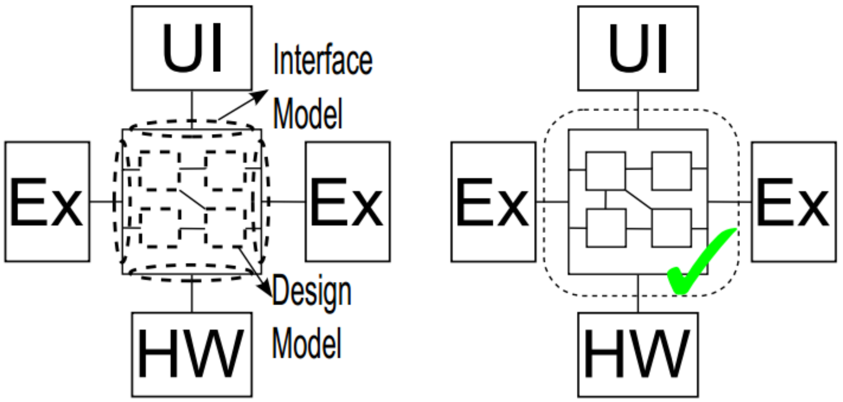
Interface model *specifies* behaviour

Design model *implements* behaviour

For each component a interface mode and design model are created in ASD

ASD: WORKFLOW

- Designer defines behavior in component models
- ASD:suite verifies models using model checking
- ASD:suite generates implementation code



(a) ASD specifications consist of design and interface models

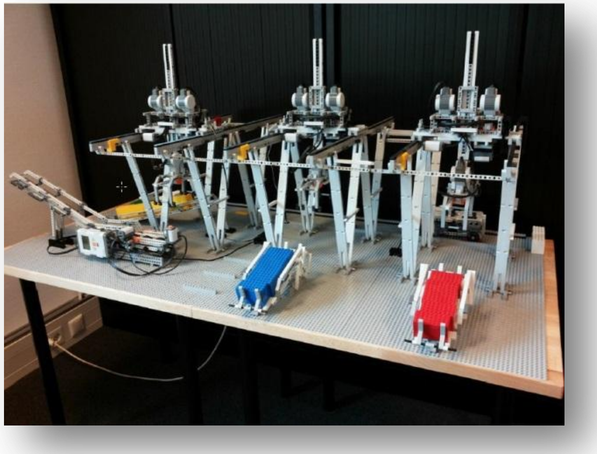
(b) ASD guarantees correctness o generated code using static verification

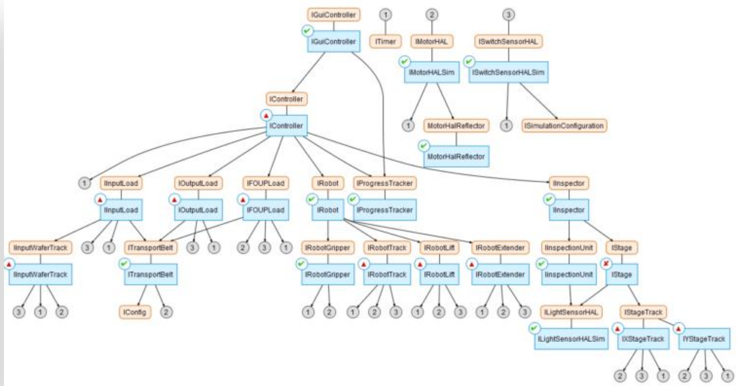
But some shortcoming

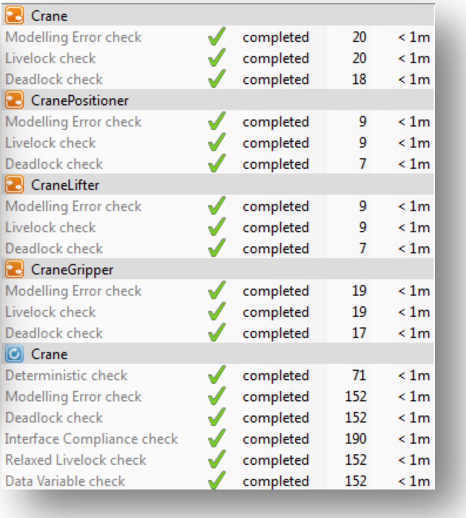
- No support for Legacy/External code
- No support for indirect component interaction
- Limited data interaction

➔ Testing is needed.

RESULTS: ASD-CONTAINER TERMINAL







Crane				
Modelling Error check	✓	completed	20	< 1m
Livelock check	✓	completed	20	< 1m
Deadlock check	✓	completed	18	< 1m
CranePositioner				
Modelling Error check	✓	completed	9	< 1m
Livelock check	✓	completed	9	< 1m
Deadlock check	✓	completed	7	< 1m
CraneLifter				
Modelling Error check	✓	completed	9	< 1m
Livelock check	✓	completed	9	< 1m
Deadlock check	✓	completed	7	< 1m
CraneGripper				
Modelling Error check	✓	completed	19	< 1m
Livelock check	✓	completed	19	< 1m
Deadlock check	✓	completed	17	< 1m
Crane				
Deterministic check	✓	completed	71	< 1m
Modelling Error check	✓	completed	152	< 1m
Deadlock check	✓	completed	152	< 1m
Interface Compliance check	✓	completed	190	< 1m
Relaxed Livelock check	✓	completed	152	< 1m
Data Variable check	✓	completed	152	< 1m

Higher overall productivity compared to traditional development.

Productivity

Main remaining problems:

- Testing interaction and complete system.
- Debugging of third party library (some bugs found in manual written code, legacy code)

CONTENTS

NSPYRE

1) MDSD Introduction

2) MDE with ASD:Suite

3) MBT with MS Spec Explorer

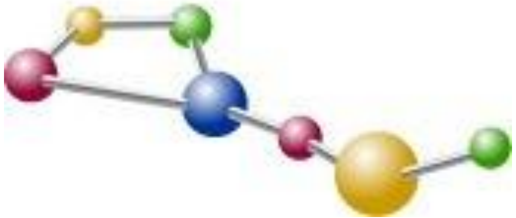
4) ASDSpec

Company

Microsoft®

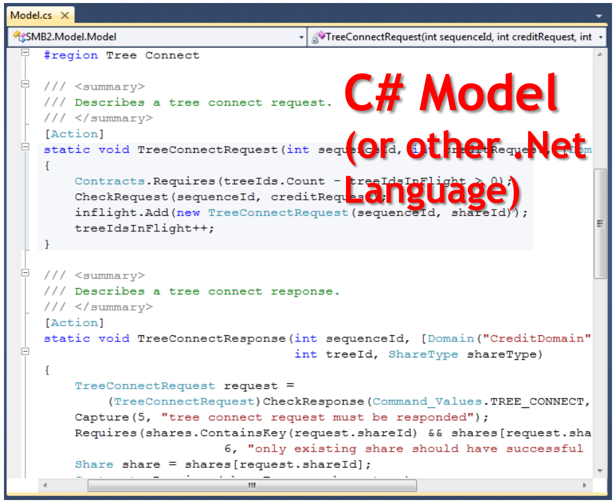
Tool

Spec Explorer



MBT WITH SPEC EXPLORER

NSPYRE

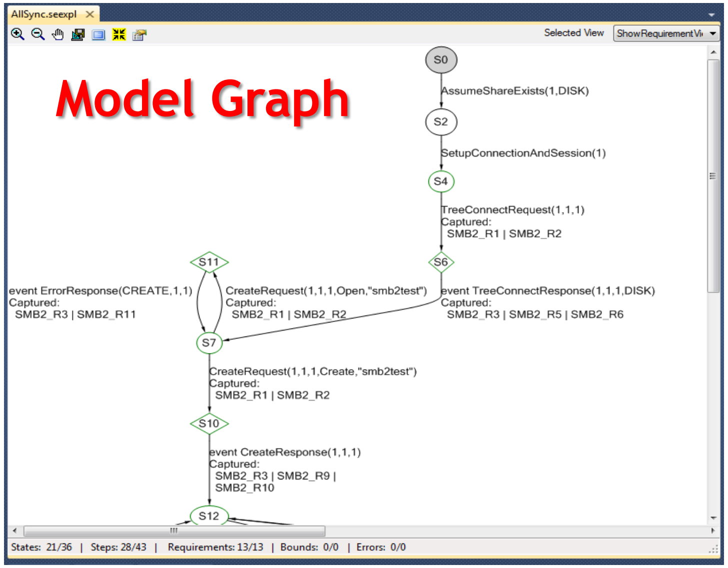


```
Model.cs
SMB2.Model.Model        TreeConnectRequest(int sequenceId, int creditRequest, int

#region Tree Connect

/// <summary>
/// Describes a tree connect request.
/// </summary>
[Action]
static void TreeConnectRequest(int sequenceId, int creditRequest, [Dom
{
    Contracts.Requires(treeIds.Count - treeIdsInFlight > 0);
    CheckRequest(sequenceId, creditRequest);
    inflight.Add(new TreeConnectRequest(sequenceId, shareId));
    treeIdsInFlight++;
}

/// <summary>
/// Describes a tree connect response.
/// </summary>
[Action]
static void TreeConnectResponse(int sequenceId, [Domain("CreditDomain"
                                int treeId, ShareType shareType)
{
    TreeConnectRequest request =
        (TreeConnectRequest)CheckResponse(Command_Values.TREE_CONNECT,
    Capture(5, "tree connect request must be responded");
    Requires(shares.ContainsKey(request.shareId) && shares[request.sha
                      6, "only existing share should have successful
    Share share = shares[request.shareId];
```

C# Model (or other .Net Language)

Explore & Analyze

Remodel

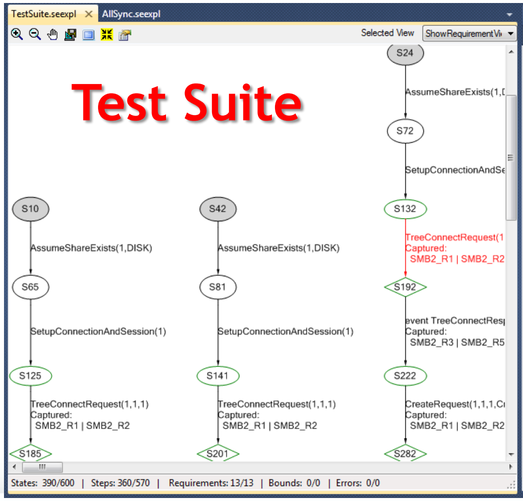


AllSync.seexpl
Selected View
ShowRequirementVi
S0
AssumeShareExists(1,DISK)
S2
SetupConnectionAndSession(1)
S4
TreeConnectRequest(1,1,1)
Captured:
SMB2_R1 | SMB2_R2
S6
S11
event ErrorResponse(CREATE,1,1)
Captured:
SMB2_R3 | SMB2_R11
CreateRequest(1,1,1,Open,"smb2test")
Captured:
SMB2_R1 | SMB2_R2
event TreeConnectResponse(1,1,1,DISK)
Captured:
SMB2_R3 | SMB2_R5 | SMB2_R6
S7
CreateRequest(1,1,1,Create,"smb2test")
Captured:
SMB2_R1 | SMB2_R2
S10
event CreateResponse(1,1,1)
Captured:
SMB2_R3 | SMB2_R9 |
SMB2_R10
S12

States: 21/36 | Steps: 28/43 | Requirements: 13/13 | Bounds: 0/0 | Errors: 0/0

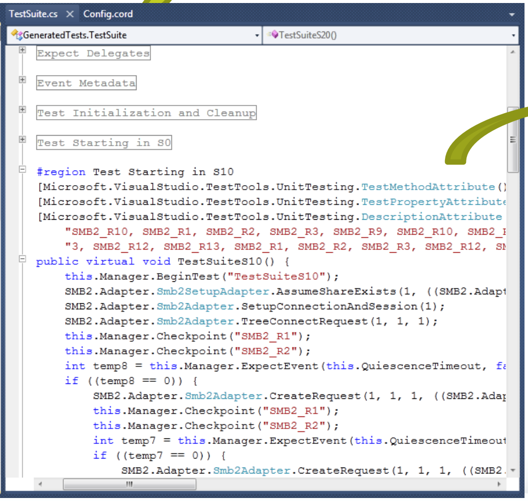
Model Graph

Generate



States: 390/600 | Steps: 360/570 | Requirements: 13/13 | Bounds: 0/0 | Errors: 0/0

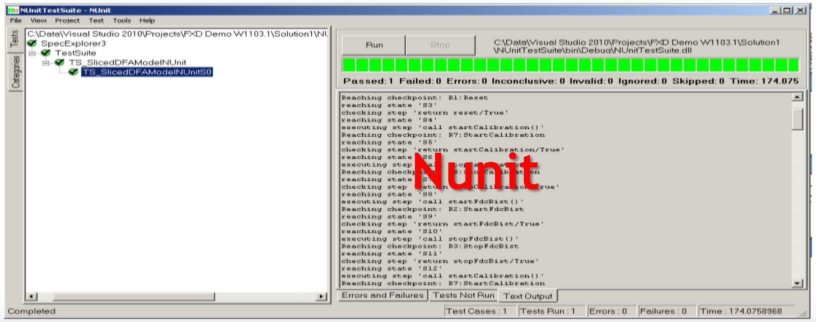
Test Suite



```
TestSuite.cs    Config.cord
GeneratedTests.TestSuite        TestSuiteS20()

Expect Delegates

Event Metadata

Test Initialization and Cleanup

Test Starting in S0

#region Test Starting in S10
[Microsoft.VisualStudio.TestTools.UnitTesting.TestMethodAttribute()
[Microsoft.VisualStudio.TestTools.UnitTesting.TestPropertyAttribute
[Microsoft.VisualStudio.TestTools.UnitTesting.DescriptionAttribute
    "SMB2_R10, SMB2_R1, SMB2_R2, SMB2_R3, SMB2_R9, SMB2_R10, SMB2_I
    "3, SMB2_R12, SMB2_R13, SMB2_R1, SMB2_R2, SMB2_R3, SMB2_R12, SM
public virtual void TestSuiteS10() {
    this.Manager.BeginTest("TestSuiteS10");
    SMB2.Adapter.Smb2SetupAdapter.AssumeShareExists(1, ((SMB2.Adapt
    SMB2.Adapter.Smb2Adapter.SetupConnectionAndSession(1);
    SMB2.Adapter.Smb2Adapter.TreeConnectRequest(1, 1, 1);
    this.Manager.Checkpoint("SMB2_R1");
    this.Manager.Checkpoint("SMB2_R2");
    int temp8 = this.Manager.ExpectEvent(this.QuiescenceTimeout, fa
    if ((temp8 == 0)) {
        SMB2.Adapter.Smb2Adapter.CreateRequest(1, 1, 1, ((SMB2.Adap
        this.Manager.Checkpoint("SMB2_R1");
        this.Manager.Checkpoint("SMB2_R2");
        int temp7 = this.Manager.ExpectEvent(this.QuiescenceTimeout
        if ((temp7 == 0)) {
            SMB2.Adapter.Smb2Adapter.CreateRequest(1, 1, 1, ((SMB2.
```

Execute

http://www.nunit.com/



```
C:\Data\Visual Studio 2010\Projects\FXD Demo W1103.1\Solution1\NU
SpecExplorer3
  TestSuite
    TS_SlicedDFAModelNUnit
      TS_SlicedDFAModelNUnitS0
```

```
C:\Data\Visual Studio 2010\Projects\FXD Demo W1103.1\Solution1
\NUnitTestSuite\bin\Debug\NUnitTestSuite.dll
Passed: 1  Failed: 0  Errors: 0  Inconclusive: 0  Invalid: 0  Ignored: 0  Skipped: 0  Time: 174.075
```

```
Reaching checkpoint: R1:Reset
reaching state 'S3'
checking step 'return reset/True'
reaching state 'S4'
executing step 'call startCalibration()'
Reaching checkpoint: R7:StartCalibration
reaching state 'S5'
checking step 'return startCalibration/True'
reaching state 'S6'
executing step 'call stopCalibration()'
Reaching checkpoint: R8:StopCalibration
reaching state 'S7'
checking step 'return stopCalibration/True'
reaching state 'S8'
executing step 'call startFdcBist()'
Reaching checkpoint: R2:StartFdcBist
reaching state 'S9'
checking step 'return startFdcBist/True'
reaching state 'S10'
executing step 'call stopFdcBist()'
Reaching checkpoint: R3:StopFdcBist
reaching state 'S11'
checking step 'return stopFdcBist/True'
reaching state 'S12'
executing step 'call startCalibration()'
Reaching checkpoint: R7:StartCalibration
```

```
Completed    Test Cases : 1    Tests Run : 1    Errors : 0    Failures : 0    Time : 174.0758968
```

Nunit

MODELING IN SPEC EXPLORER

NSPYRE

```
[TypeBinding("CraneComponent")]
    class Crane
    {
        public Cranestates Cranestatevar;
        internal ICrane_NIModel ICrane_NIimpl;
        internal ICrane ICraneimpl;
        [Rule(Action = "new CraneComponent(craneNr)",
                                    ModeTransition = "crane->crane0")]
        Crane(int craneNr)
            {
                ICraneimpl = new ICraneCranecomponent(this);
                craneList.Add(this);
            }
```

State declaration

Used components

Initialization

Spec Explorer represents models as annotated C# code

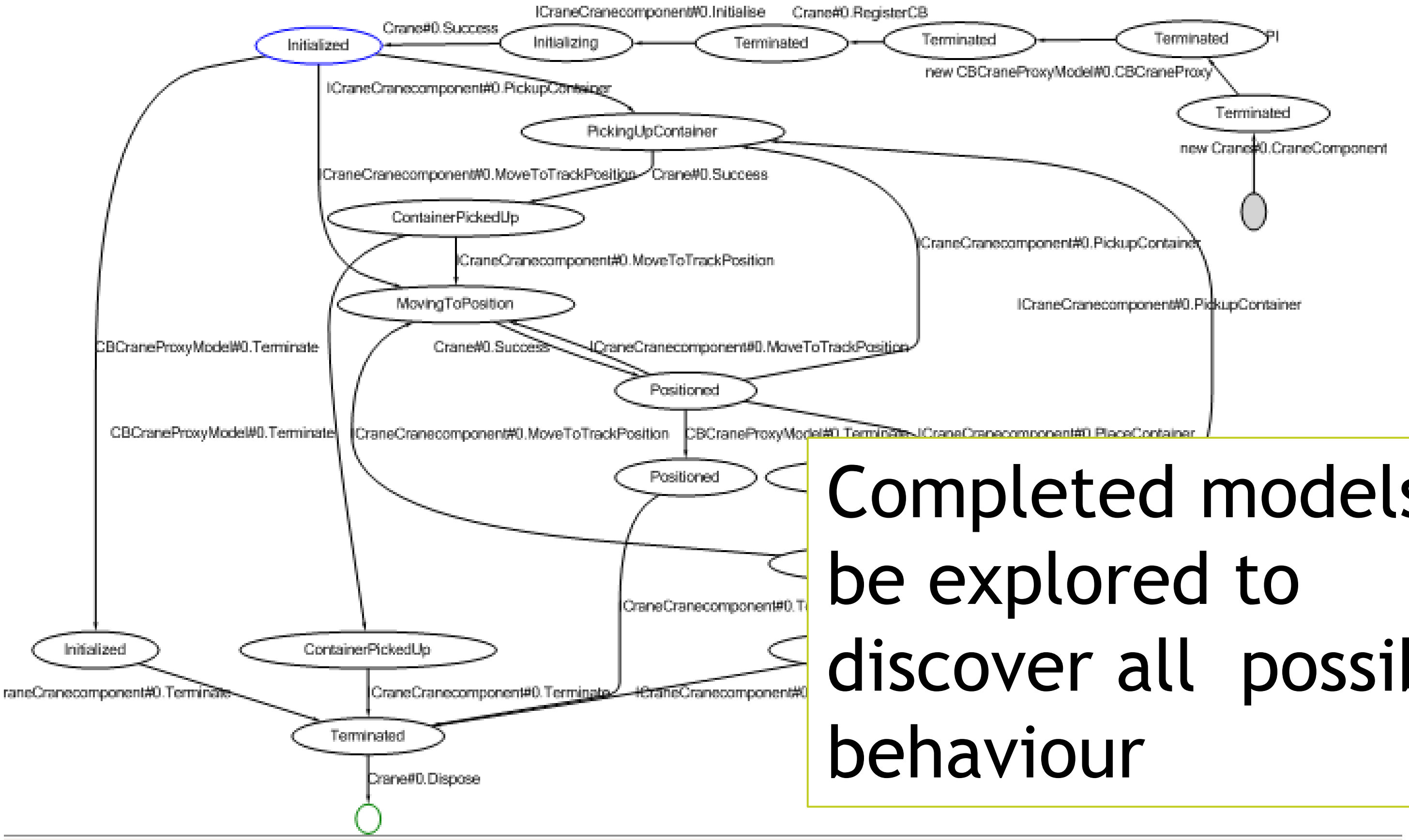
MODELING IN SPEC EXPLORER

```
[Rule(Action = "this.PlaceContainer(height)", ModeTransition = "crane4->crane4")]
      public void ICrane_PlaceContainer(GripperHeightEnum height)
      {
          switch (basecomponent.Cranestatevar) {
              case Cranestates.Positioned: {
                      basecomponent.Cranestatevar
                      = Cranestates.PlacingContainer;
                      basecomponent.ICrane_NIimpl.PlaceContainer(height);
                      return;
                  }
              default:{
                      Condition.IsTru
                      throw new Inval
                  }
          };
      }
```

Every model method represents an action, its conditions and its effects.

MBT-VISUALIZATION

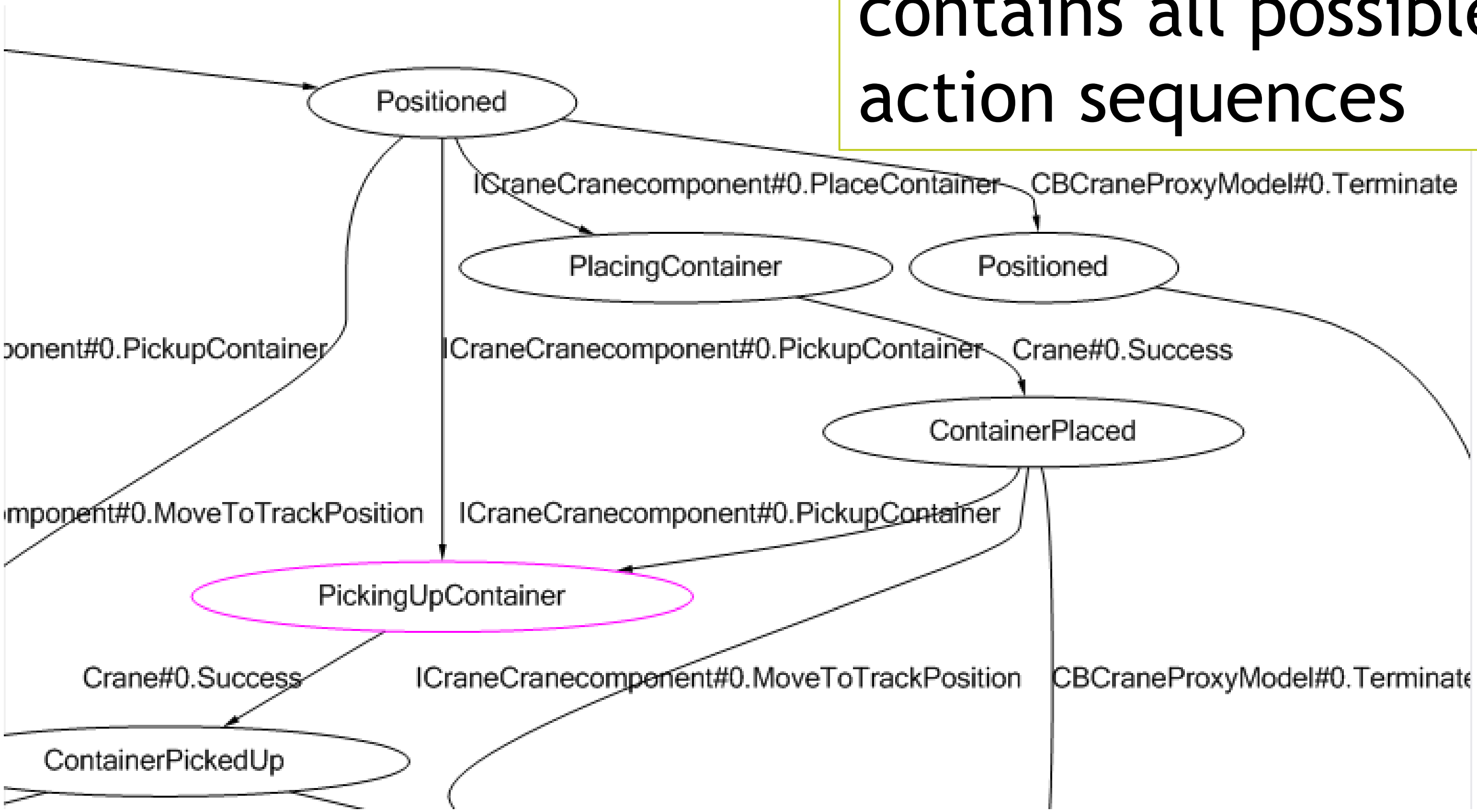
NSPYRE


ICraneCranecomponent#0.Initialise
Crane#0.RegisterCB
Crane#0.Success
Initialized
Initializing
Terminated
Terminated
Terminated
PI
new CBCraneProxyModel#0.CBCraneProxy
Terminated
new Crane#0.CraneComponent
ICraneCranecomponent#0.PickupContainer
PickingUpContainer
ICraneCranecomponent#0.MoveToTrackPosition
Crane#0.Success
ContainerPickedUp
ICraneCranecomponent#0.MoveToTrackPosition
ICraneCranecomponent#0.PickupContainer
MovingToPosition
ICraneCranecomponent#0.PickupContainer
CBCraneProxyModel#0.Terminate
Crane#0.Success
ICraneCranecomponent#0.MoveToTrackPosition
Positioned
CBCraneProxyModel#0.Terminate
ICraneCranecomponent#0.MoveToTrackPosition
CBCraneProxyModel#0.Terminate
ICraneCranecomponent#0.PlaceContainer
Positioned
ICraneCranecomponent#0.T
Initialized
ContainerPickedUp
ICraneCranecomponent#0.Terminate
ICraneCranecomponent#0.Terminate
ICraneCranecomponent#0
Terminated
Crane#0.Dispose

Completed models can be explored to discover all possible behaviour

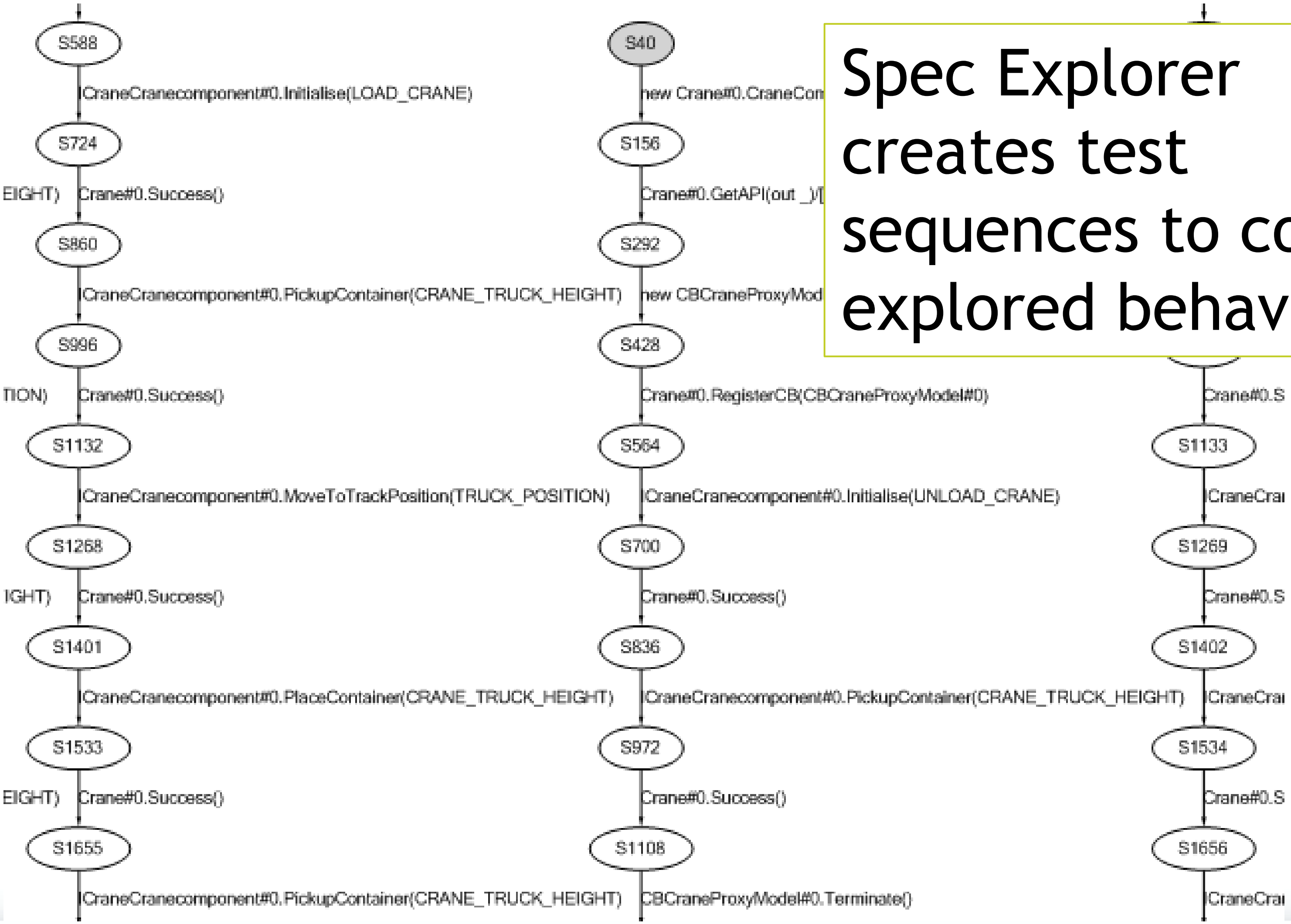
MBT- VISUALIZATION

Explored model contains all possible action sequences


Positioned
ICraneCranecomponent#0.PlaceContainer
CBCraneProxyModel#0.Terminate
PlacingContainer
Positioned
ponent#0.PickupContainer
ICraneCranecomponent#0.PickupContainer
Crane#0.Success
ContainerPlaced
mponent#0.MoveToTrackPosition
ICraneCranecomponent#0.PickupContainer
PickingUpContainer
Crane#0.Success
ICraneCranecomponent#0.MoveToTrackPosition
CBCraneProxyModel#0.Terminate
ContainerPickedUp

TEST GENERATION

NSPYRE


S588
CraneCranecomponent#0.Initialise(LOAD_CRANE)
S724
EIGHT)
Crane#0.Success()
S860
CraneCranecomponent#0.PickupContainer(CRANE_TRUCK_HEIGHT)
S996
TION)
Crane#0.Success()
S1132
CraneCranecomponent#0.MoveToTrackPosition(TRUCK_POSITION)
S1268
IGHT)
Crane#0.Success()
S1401
CraneCranecomponent#0.PlaceContainer(CRANE_TRUCK_HEIGHT)
S1533
EIGHT)
Crane#0.Success()
S1655
CraneCranecomponent#0.PickupContainer(CRANE_TRUCK_HEIGHT)
S40
new Crane#0.CraneCom
S156
Crane#0.GetAPI(out _)/[
S292
new CBCraneProxyMod
S428
Crane#0.RegisterCB(CBCraneProxyModel#0)
S564
CraneCranecomponent#0.Initialise(UNLOAD_CRANE)
S700
Crane#0.Success()
S836
CraneCranecomponent#0.PickupContainer(CRANE_TRUCK_HEIGHT)
S972
Crane#0.Success()
S1108
CBCraneProxyModel#0.Terminate()
Crane#0.S
S1133
CraneCra
S1269
Crane#0.S
S1402
CraneCra
S1534
Crane#0.S
S1656
CraneCra

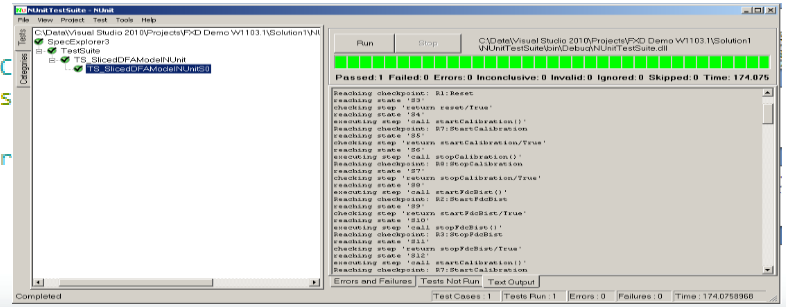
Spec Explorer creates test sequences to cover explored behaviour

GENERATED TEST SCRIPTS

NSPYRE

Nunit test scripts are used to implement and execute tests

```
#region Test Starting in S100
[Microsoft.VisualStudio.TestTools.UnitTesting.TestMethodAttribute()]
public void CraneTestCasesS100() {
    this.Manager.BeginTest("CraneTestCasesS100");
    this.Manager.Comment("reaching state \'S100\'");
    CraneComponent temp6;
    this.Manager.Comment("executing step \'call new Crane#0.CraneCompone
    temp6 = new CraneComponent(0);
    this.Manager.Comment("reaching state \'S101\'");
    this.Manager.Comment("checking step \'return new Crane#0.CraneComponent\'");
    TestManagerHelpers.AssertBind<CraneComponent>(this.Manager, this.o, temp6, "this
    this.Manager.Comment("reaching state \'S186\'");
    ICrane temp7;
    this.Manager.Comment("executing step \'call Crane#0.GetAPI(out _)\'");
    this.o.Value.GetAPI(out temp7);
    this.Manager.Comment("reaching state \'S254\'");
    this.Manager.Comment("checking step \'return Crane#0.GetAPI/[out ICraneCranecompo
    TestManagerHelpers.AssertBind<CraneImplScope.ICraneProxy>(this.Manager, this.o1,
    this.Manager.Comment("reaching state \'S322\'");
    CBProxies.CBCraneProxy temp8;
    this.Manager.Comment("executing step \'call new CBCraneProxyModel#0.CBCraneProxy(
    temp8 = new CBProxies.CBCraneProxy();
    this.Manager.Comment("reaching state \'S390\'");
    this.Manager.Comment("checking step \'return new CBCraneProxyModel#0.CBC
    TestManagerHelpers.AssertBind<CBProxies.CBCraneProxy>(this.Manager, this
    this.Manager.Comment("reaching state \'S458\'");
    this.Manager.Comment("executing step \'call Crane#0.RegisterCB(CBCranePr
    this.o.Value.RegisterCB(((ICrane_NI)(this.o2.Value)));
    this.Manager.Comment("reaching state \'S526\'");
    this.Manager.Comment("checking step \'return Crane#0.RegisterCB\'");
```



C:\Data\Visual Studio 2010\Projects\FXD Demo W1103.1\Solution1\NU

- SpecExplorer3
  - TestSuite
    - TS_SlicedDFAModelNUnit
      - TS_SlicedDFAModelNUnitS0

C:\Data\Visual Studio 2010\Projects\FXD Demo W1103.1\Solution1
\NUnitTestSuite\bin\Debug\NUnitTestSuite.dll

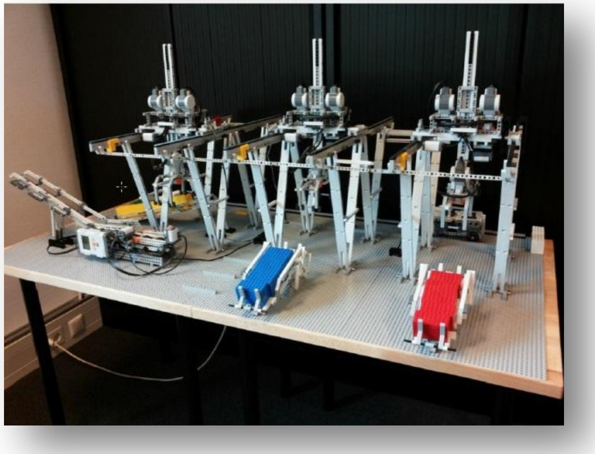
Passed: 1 Failed: 0 Errors: 0 Inconclusive: 0 Invalid: 0 Ignored: 0 Skipped: 0 Time: 174.075

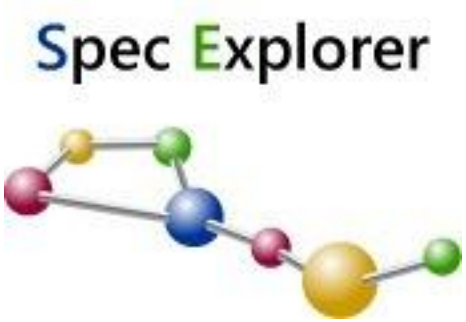
```
Reaching checkpoint: R1:Reset
reaching state 'S3'
checking step 'return reset/True'
reaching state 'S4'
executing step 'call startCalibration()'
Reaching checkpoint: R7:StartCalibration
reaching state 'S5'
checking step 'return startCalibration/True'
reaching state 'S6'
executing step 'call stopCalibration()'
Reaching checkpoint: R8:StopCalibration
reaching state 'S7'
checking step 'return stopCalibration/True'
reaching state 'S8'
executing step 'call startFdcBist()'
Reaching checkpoint: R2:StartFdcBist
reaching state 'S9'
checking step 'return startFdcBist/True'
reaching state 'S10'
executing step 'call stopFdcBist()'
Reaching checkpoint: R3:StopFdcBist
reaching state 'S11'
checking step 'return stopFdcBist/True'
reaching state 'S12'
executing step 'call startCalibration()'
Reaching checkpoint: R7:StartCalibration
```

Completed | Test Cases : 1 | Tests Run : 1 | Errors : 0 | Failures : 0 | Time : 174.0758968

RESULTS:
SPEC EXPLORER-CONTAINER TERMINAL

NSPYRE





Significantly less effort than traditional automated testing

- Model Based Testing + Software Analysis
- Support of data combination testing
- Support of Model composition, incremental

However some remaining problems:

- Modeling still comparatively expensive
- Modeling effort, complexity and skills (experienced tester needed)

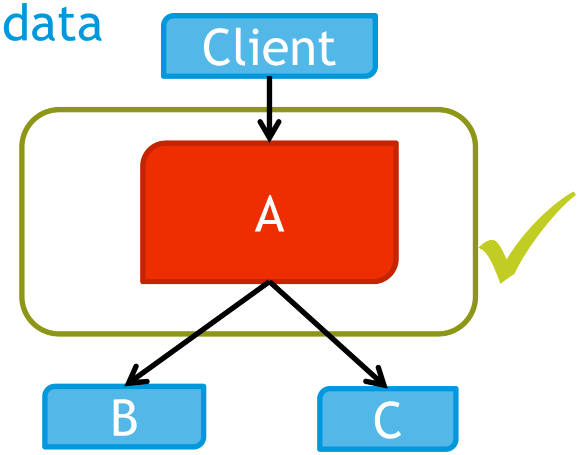
HOW TO COMBINE BOTH BENEFITS

NSPYRE

ASD generates verified software components

Complications

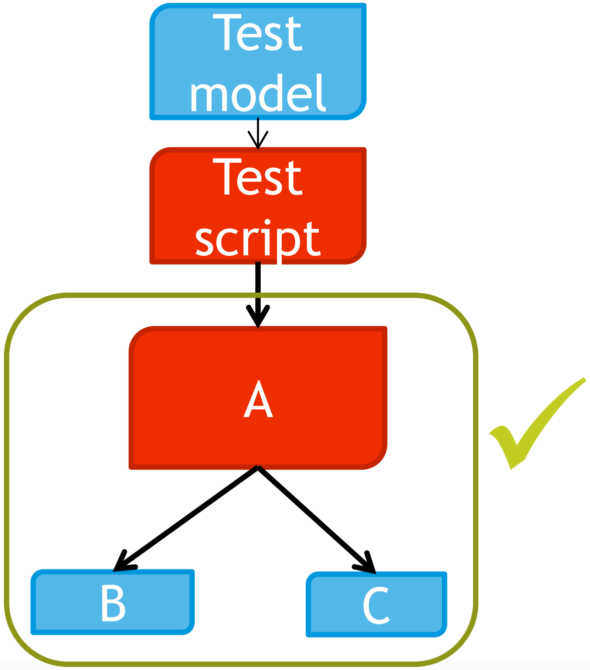
- Legacy code
- External code
- Component Interaction
- Limited support of data



Spec Explorer generates automated software tests

Complications

- Modeling needed
- MBT skills needed



CONTENTS

NSPYRE

1) MDSD Introduction

2) MDE with ASD:Suite

3) MBT with MS Spec Explorer

4) ASDSpec

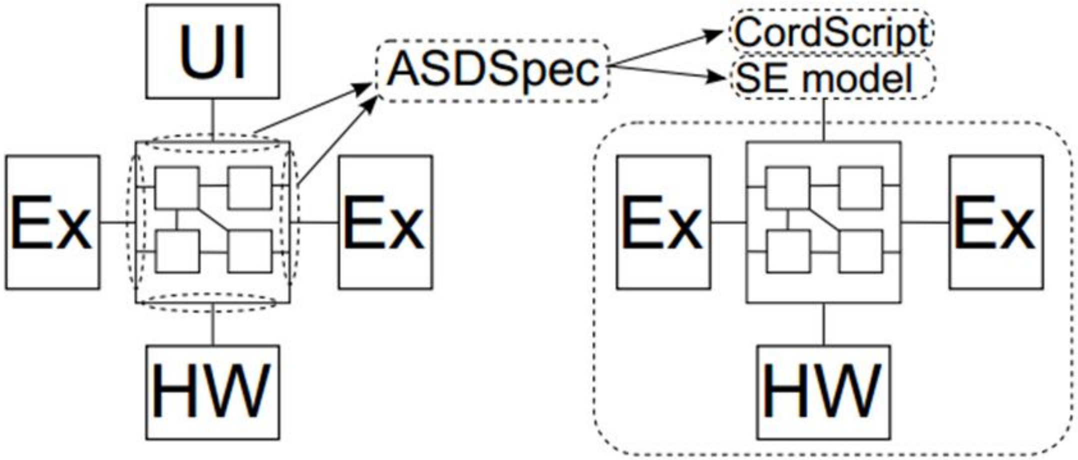
Company

NSPYRE

Tool

OUR SOLUTION: ASDSPEC

NSPYRE



ASDSpec converts ASD Interface Models to Spec Explorer Models and Cord Scripts

ASDSpec generates Test Model (push button) from existing ASD interface models
Tester need to refine manually the generated model (add behavior, data, slicing,…)

COMBINING THE BEST OF BOTH WORLDS

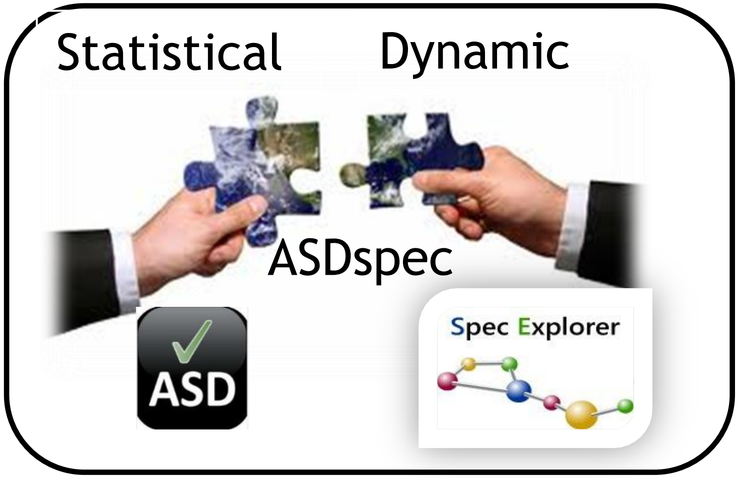
Automatic generation of partial Spec Explorer test model from existing ASD model (push button)

Benefits

- Effortless model creation
  - reuse of existing work
- Testing of complete system including
  - Legacy code
  - External components
  - Data combination testing
  - Interaction testing

Results:

- High Quality, Reduced cost
- Adds dynamic testing to ASD



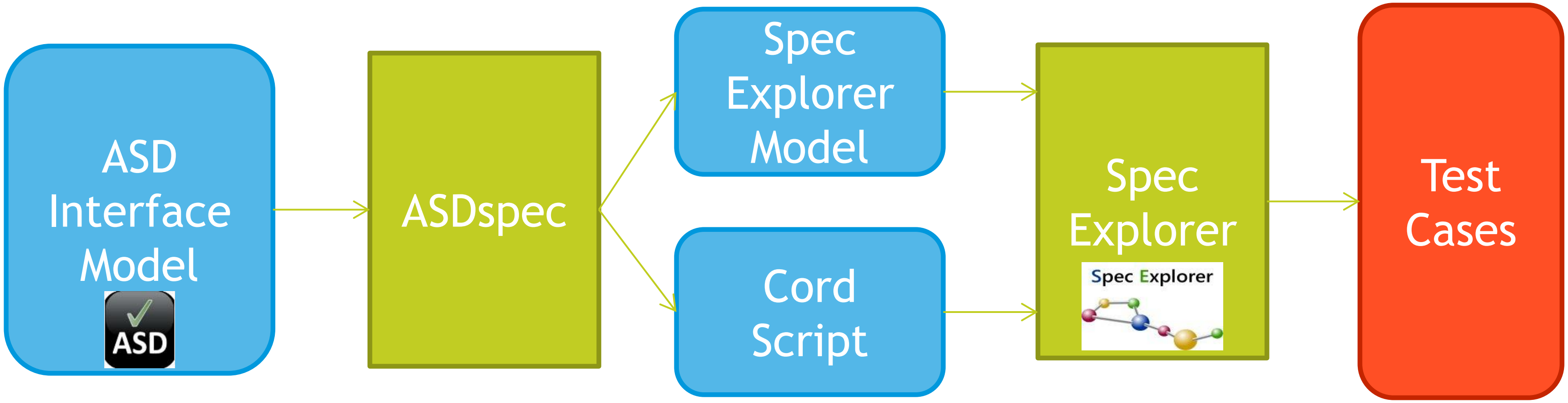
Statistical

Dynamic

ASDspec

ASDSPEC: WORKFLOW

NSPYRE

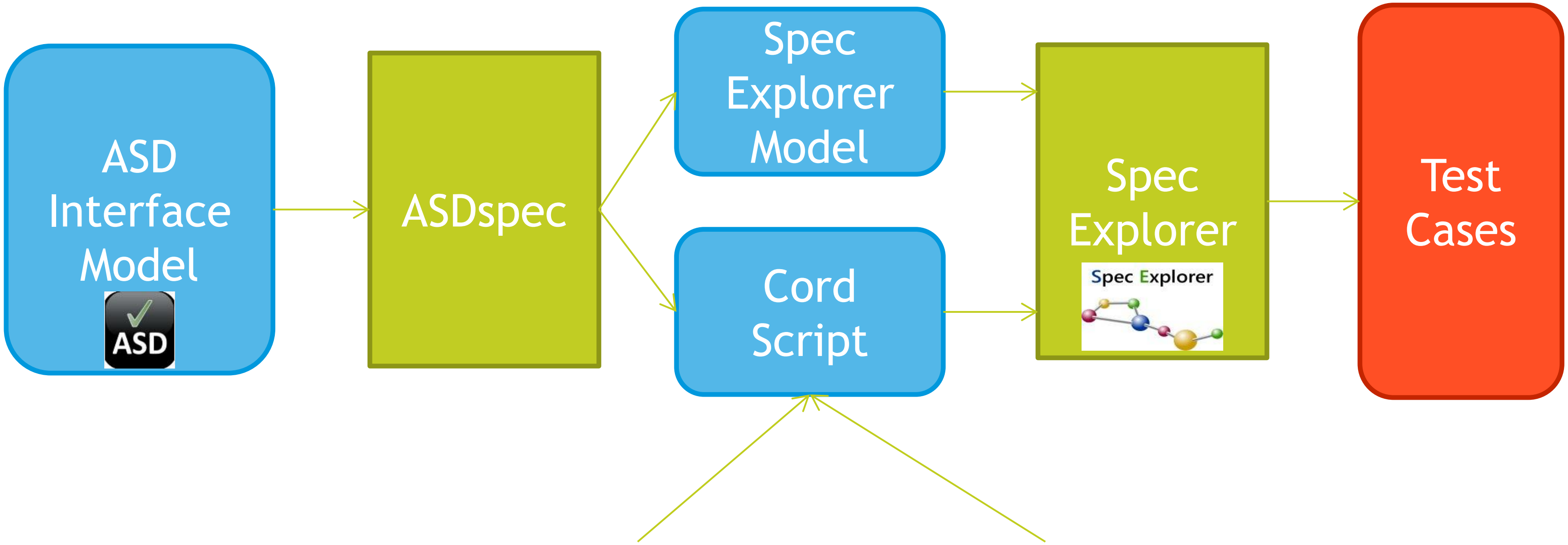


Reuse existing ASD interface models to generate Spec Explorer MBT models automatically

ASDSPEC: DATA AND MODEL COMPOSITION

NSPYRE

Generated code can be extended with any Spec Explorer feature

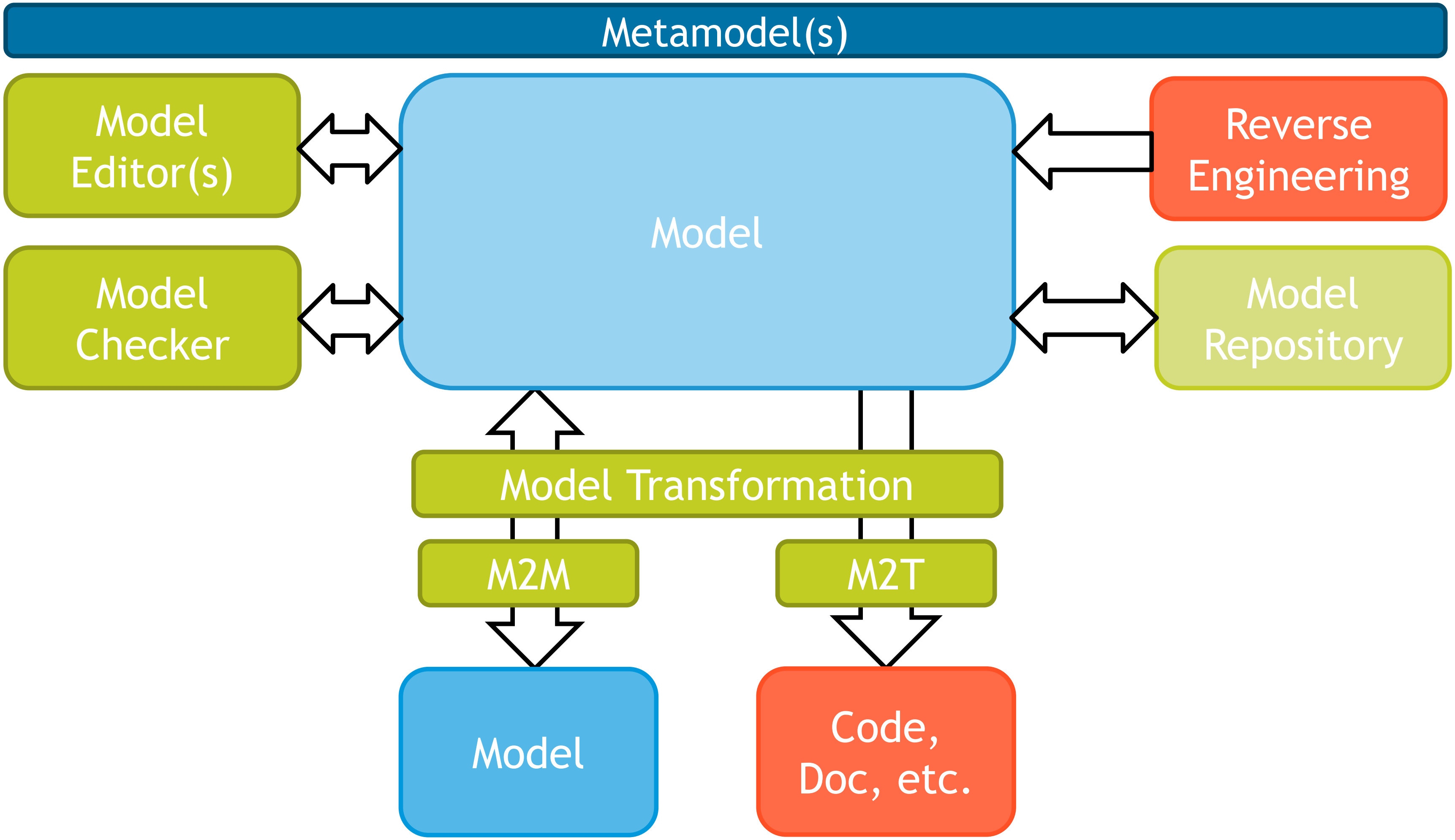


Add model composition

Add data interaction

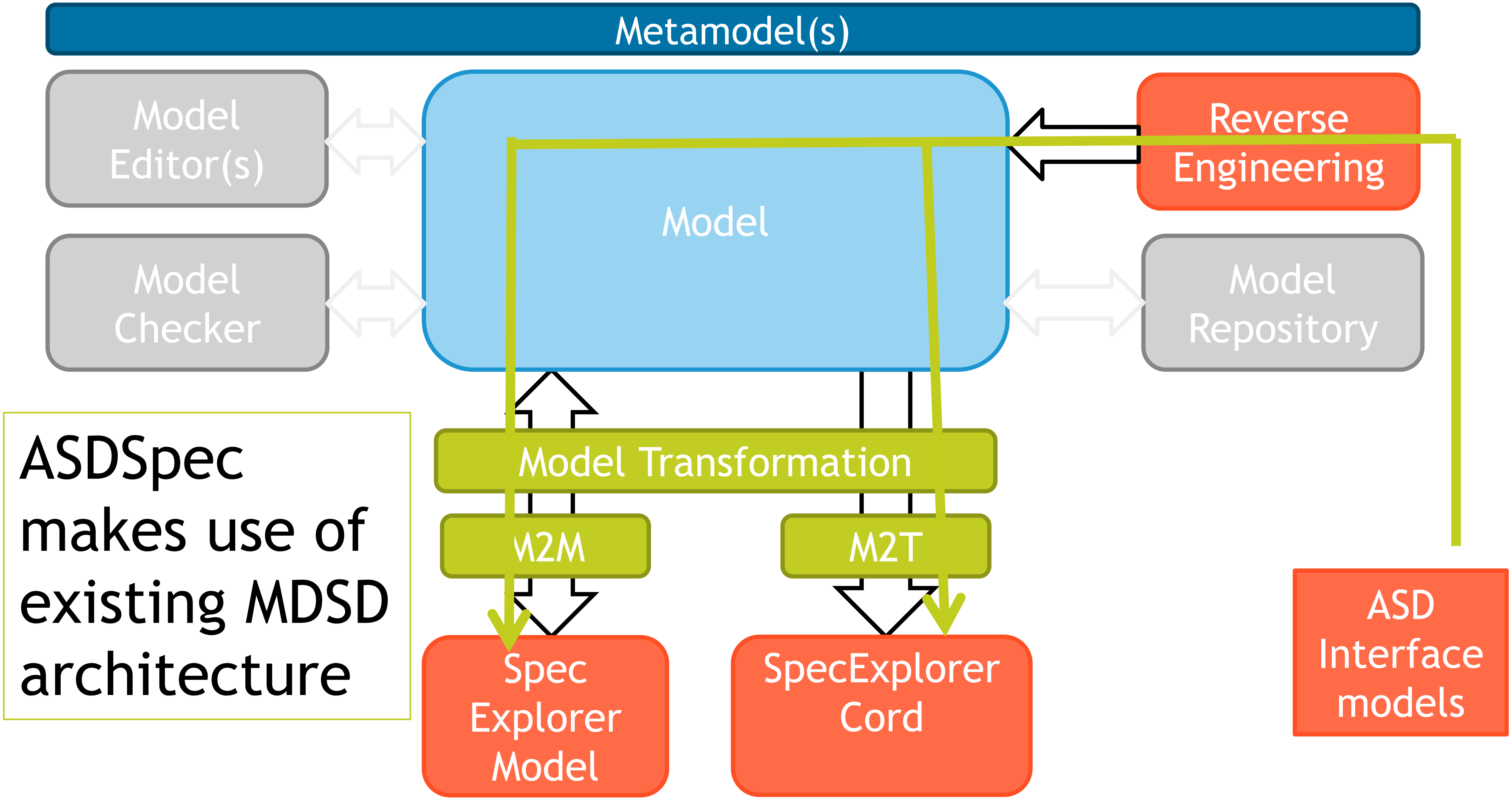
GENERAL MDSD ARCHITECTURE

NSPYRE



ASDSPEC ARCHITECTURE

NSPYRE

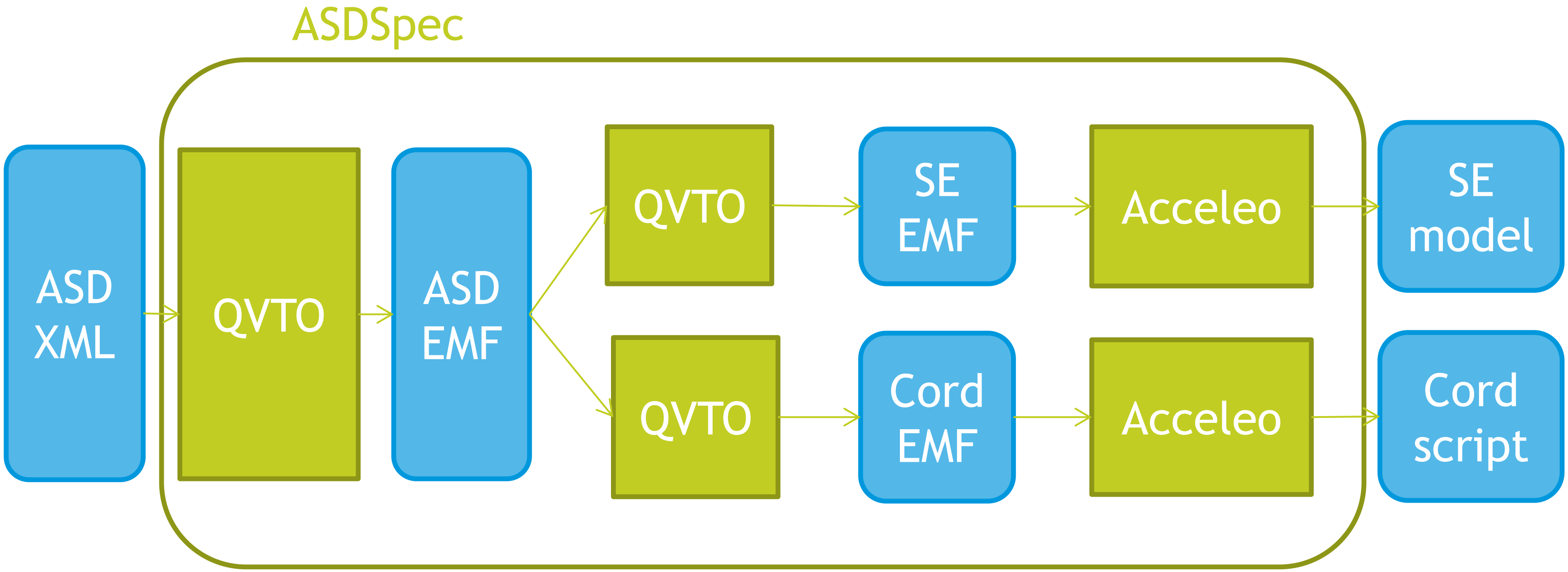


ASDSpec makes use of existing MDSD architecture

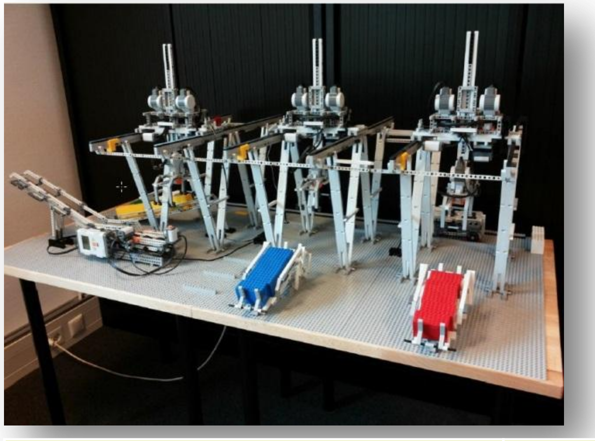
ASDSPEC: TECHNOLOGIES

NSPYRE

Model transformations bridge gap between ASD and Spec Explorer



RESULTS:
ASDSPEC- CONTAINER TERMINAL

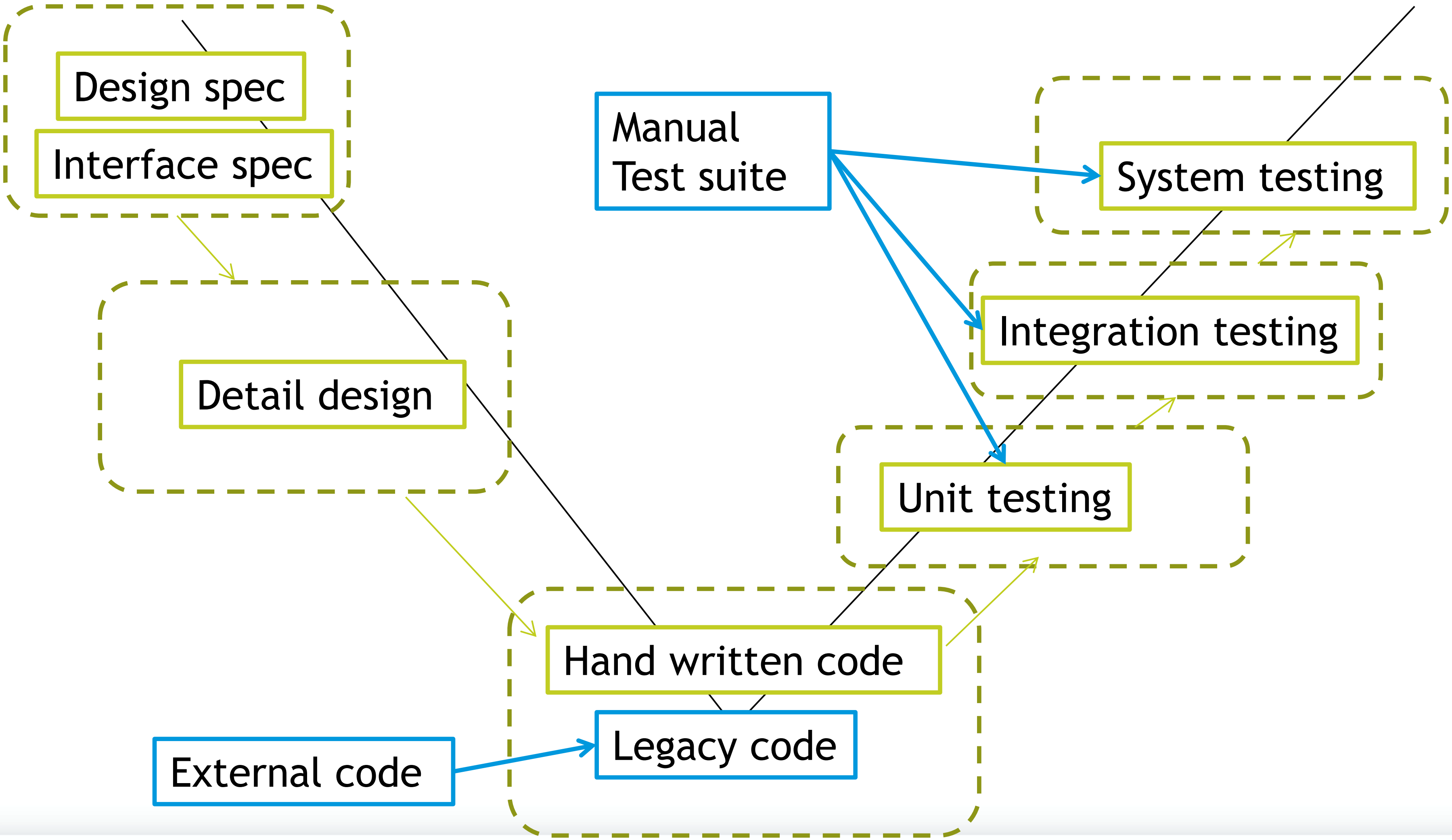


- Low effort model creation
- Easy reuse of existing work
- Testing of complete system
- Legacy code, data interaction
- High Quality, Cost Reduction

	Spec Explorer	ASD + ASDSpec
Approach	Generate test suite	Generate test model
Techniques	MBT	MBT
Effort/complexity	Medium	Low
Test cases	89	93
Perceived effectiveness	Medium	High
Bugs	Some bugs in hand written code /HAL	All known bus found

TRADITIONAL SW DEVELOPMENT TRADITIONAL MANUAL TESTING

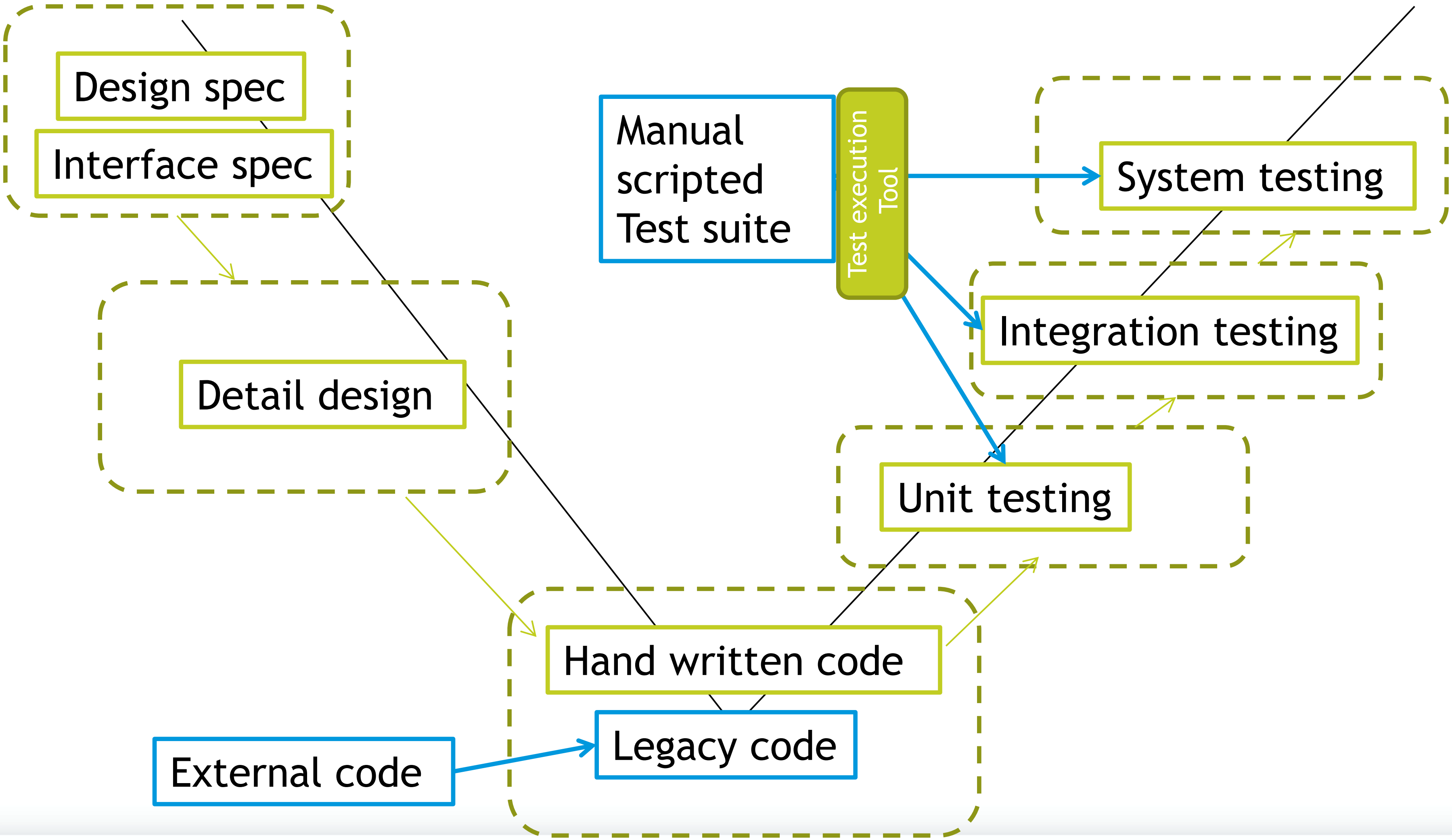
NSPYRE



Summary

TRADITIONAL SW DEVELOPMENT TRADITIONAL AUTOMATED TESTING

NSPYRE



Design spec

Interface spec

Detail design

Hand written code

Legacy code

External code

Manual scripted Test suite

Test execution Tool

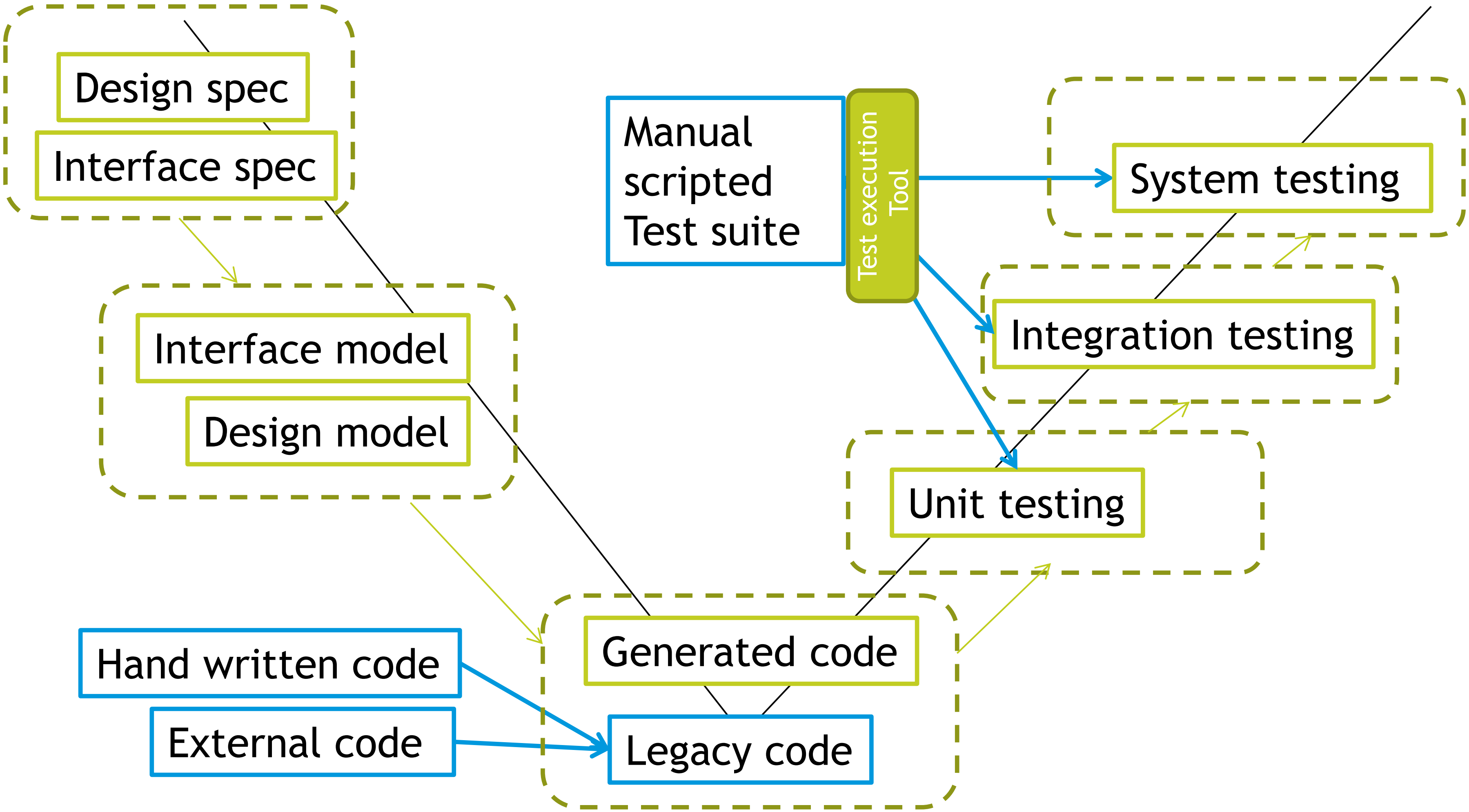
System testing

Integration testing

Unit testing

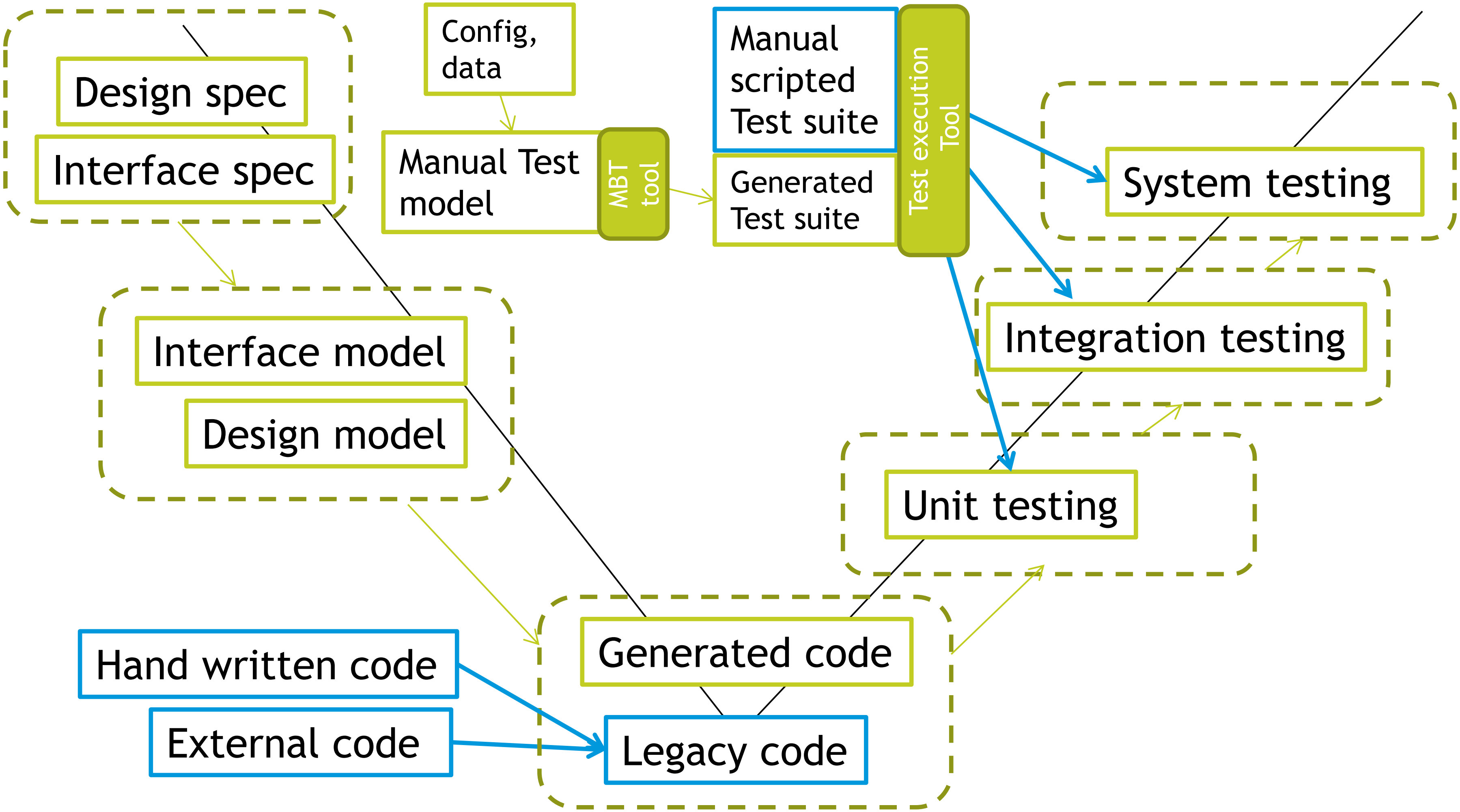
ASD BASED SW DEVELOPMENT TRADITIONAL AUTOMATED TESTING

NSPYRE



ASD BASED SW DEVELOPMENT MODEL BASED TESTING

NSPYRE



Design spec

Interface spec

Config, data

Manual Test model

MBT tool

Manual scripted Test suite

Generated Test suite

Test execution Tool

System testing

Interface model

Design model

Integration testing

Unit testing

Hand written code

External code

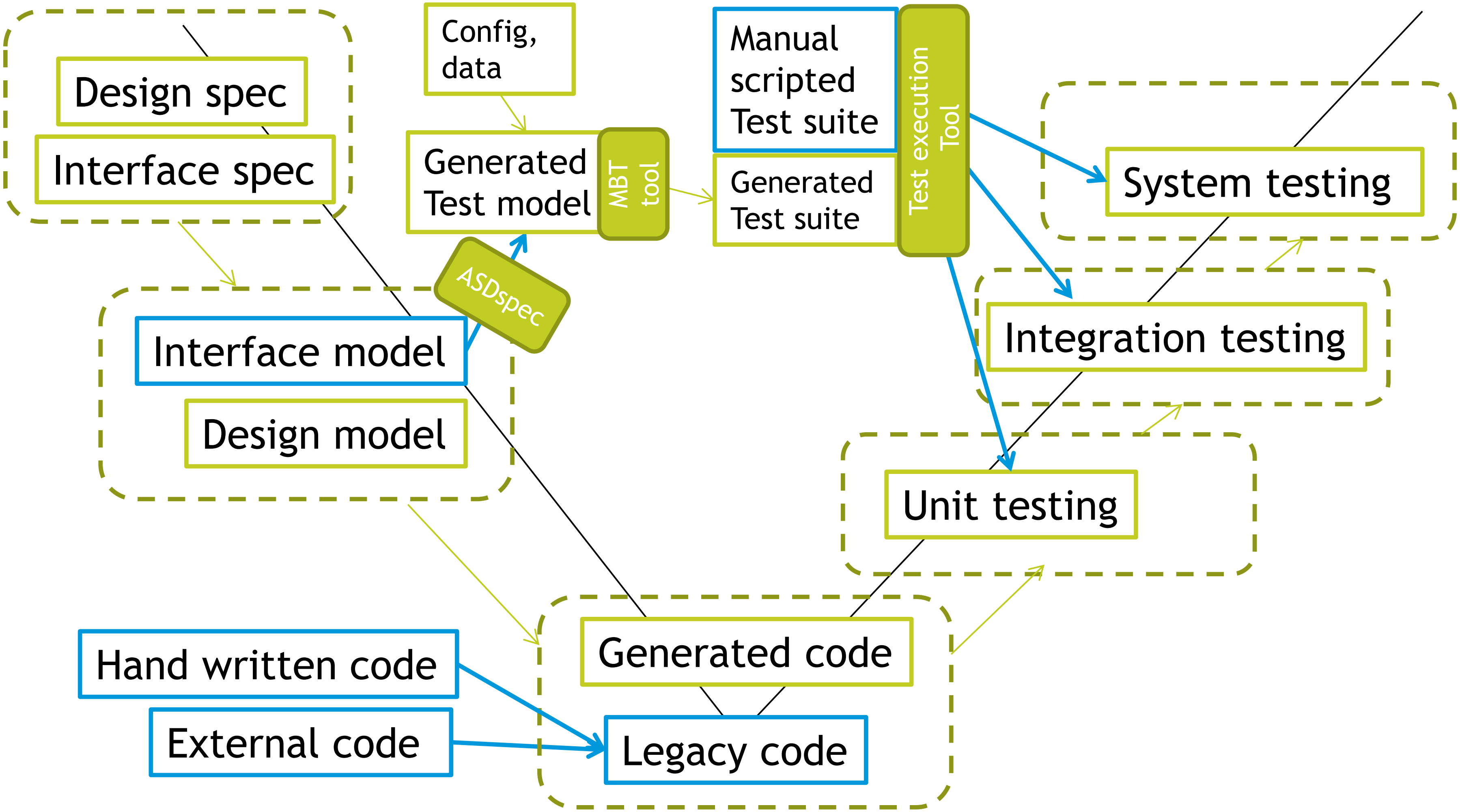
Generated code

Legacy code

Summary

ASD BASED SW DEVELOPMENT MODEL BASED TESTING+ASDSPEC

NSPYRE



TOOL STATUS

- Tool currently in prototype phase
- as an Eclipse plugin
- For now available on request

- Future steps
  - Support data handling/configuration aspects in ASDSpec, instead of relying on manual additions in Spec Explorer,
  - Supporting other MBT tools.

CONCLUSIONS

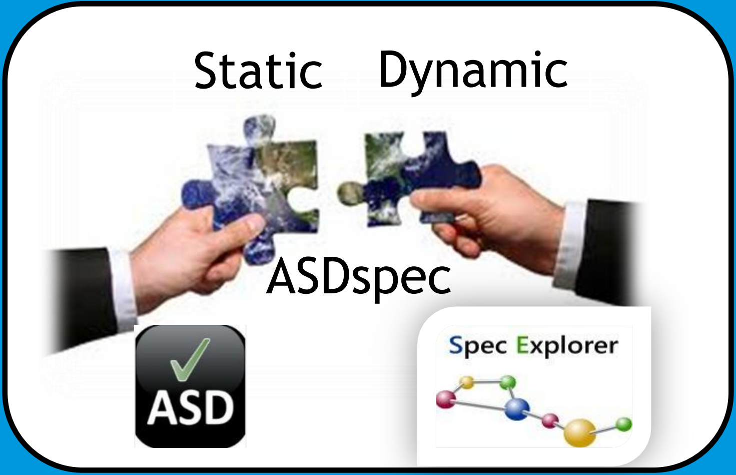
NSPYRE

- MDE and MBT technologies have matured a lot in the latest years
- It is a matter of time….. Evolution…..

QUESTIONS MATTER.

EUROPEAN JOINT CONFERENCE ON THEORY & PRACTICE OF SOFTWARE

SO YOU WON'T HAVE ANY LEFT



Using Formal Specifications to Support Model Based Testing

ASDSpec: A Tool Combining the Best of Two Techniques

NSPYRE