# The UML 2.0 Testing Profile

Ina Schieferdecker,
Technical University Berlin/FOKUS

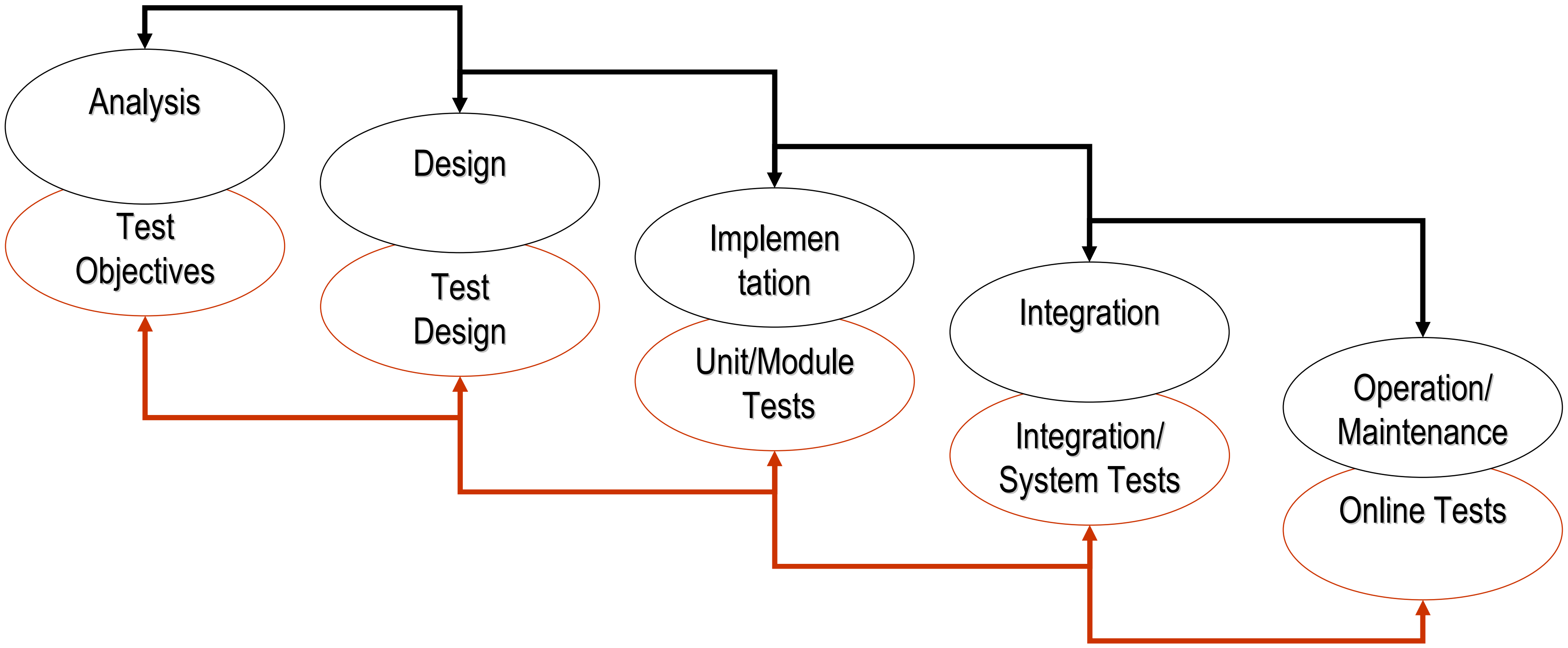
Jens Grabowski, Uni Göttingen

## Outline

- Introduction
- The Testing Profile
- Its Relation to TTCN-3

## Motivation

- Integrated Development and Testing



➢ Early and continuous consideration of test aspects

## Motivation

- Model Driven Architecture as new OMG strategy

  - One objective of UML 2.0 is executable UML meaning
    - code generation
    - simulation
    - validation
    - test generation

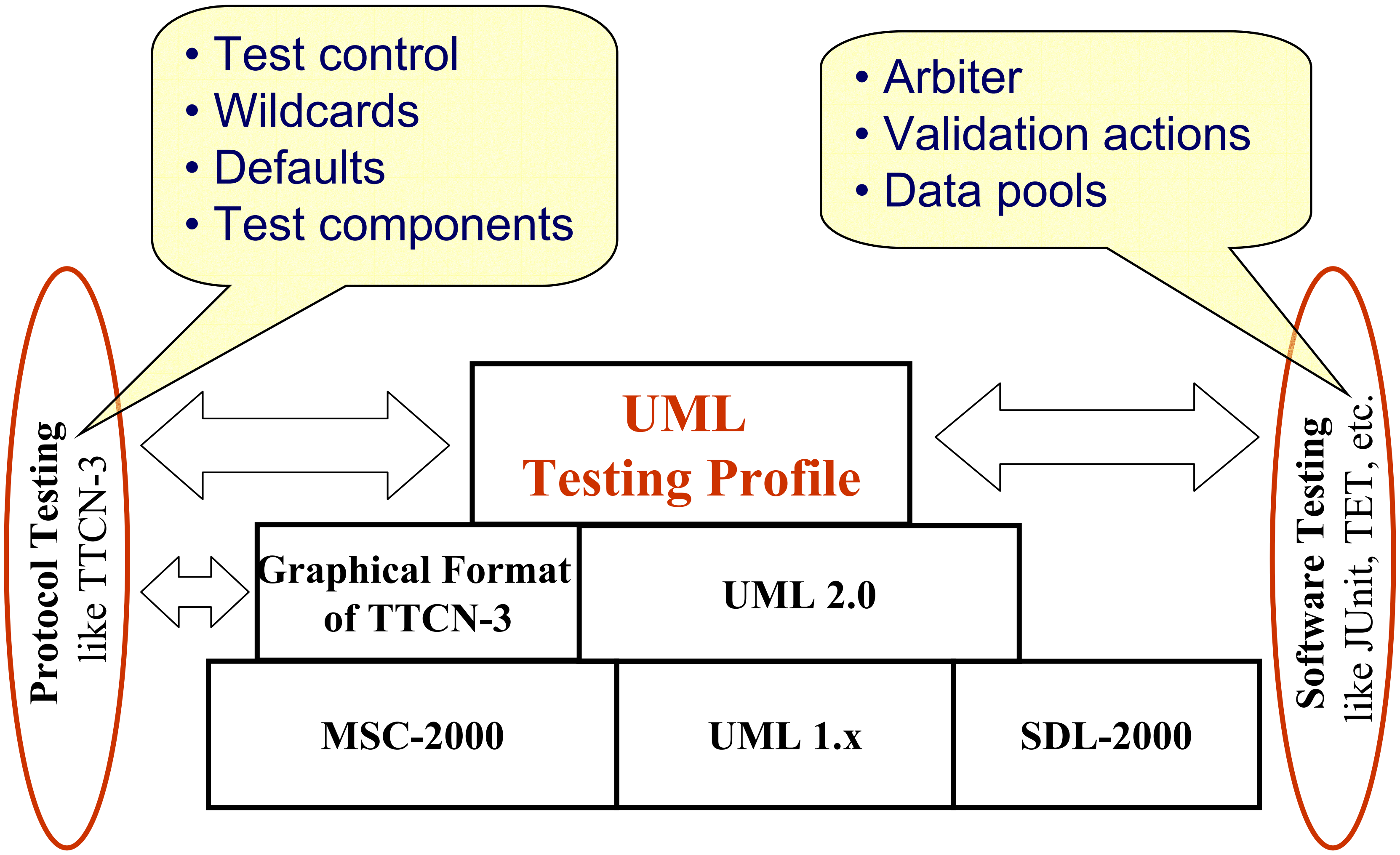
  - "..., the expanded role of the OMG must be built on rock-solid testing, certification and branding. …"

## Test Artefacts in UML

- Possible
  - Internal structure for test configuration
  - Class diagrams for test structuring
  - sequence diagrams for test purpose definition
- However, test case specifications ?
  i.e. test specific concepts which have evolved over time
  - Points of control and/or observation and coordination points
  - Test components and SUT
  - Default behaviors
  - Verdict handling

➢UML is not yet ready for testing

## The Testing Profile Roots



## Concepts of the Testing Profile

- Test architecture
  - Test structure, test components and test configuration
- Test data
  - Test data and templates used in test procedures
- Test behavior
  - Dynamic aspects of test procedures
- Test time
  - Time quantified definition of test procedures

# Testing Profile Concepts

| Architecture concepts | Behavior concepts | Data concepts | Time concepts |
|---|---|---|---|
| SUT | Test objective | Wildcards | Timer |
| Test components | Test case | Data pools | Time zone |
| Test suite | Defaults | Data partitions | |
| Test configuration | Verdicts | Data selectors | |
| Test control | | Coding rules | |
| Arbiter | | | |
| Scheduler | | | |
| Utility part | | | |

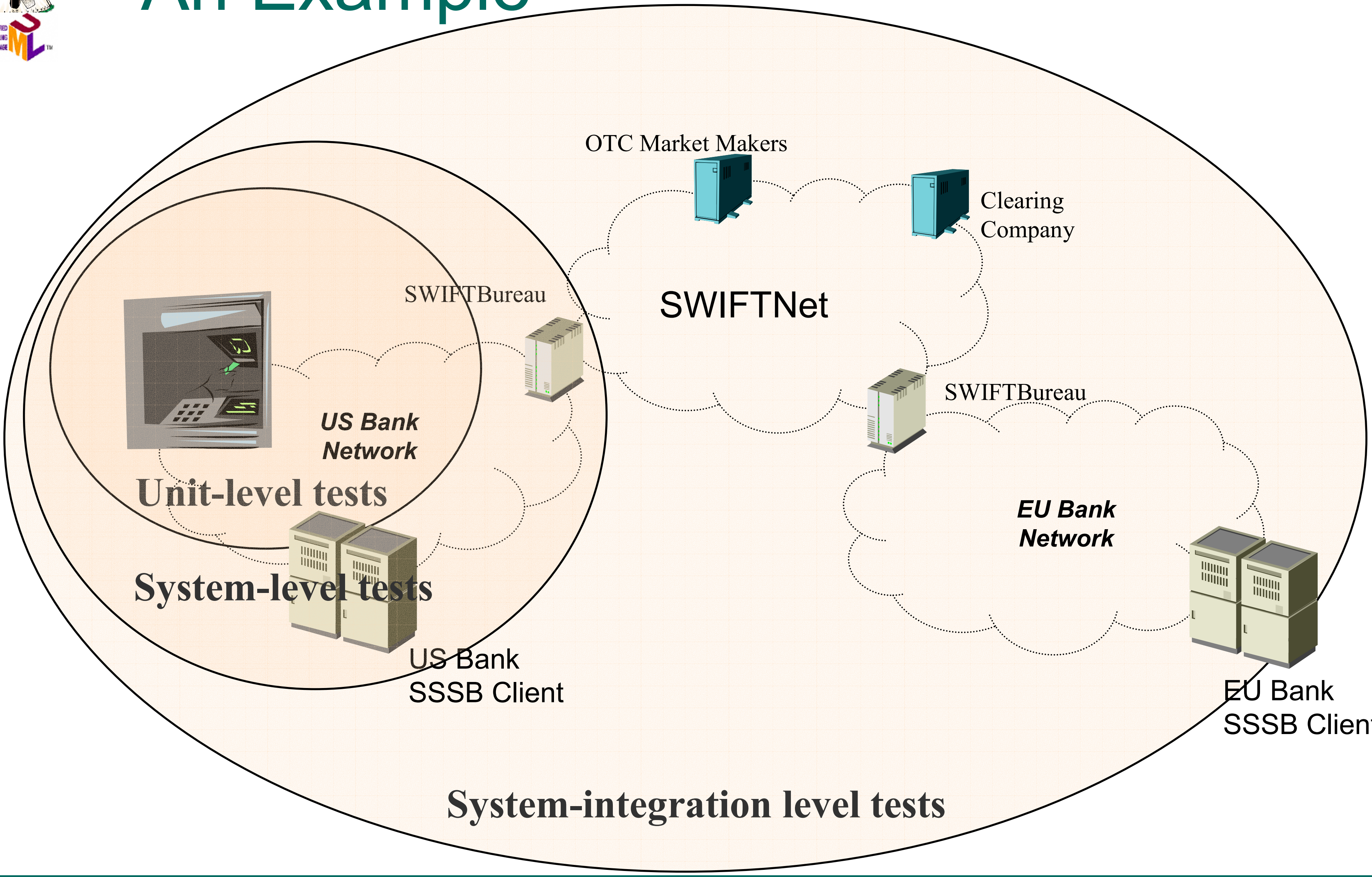
# Concepts beyond UML

- Defaults within test behavior
  - Concentration on main flow of test behavior
  - Default hierarchy to handle different concerns
- Wildcards within test data
  - Flexible definition of value sets
- Timers and time constraints
  - Time controlled test behavior
- Arbitration and verdicts
  - Assessment of test behavior

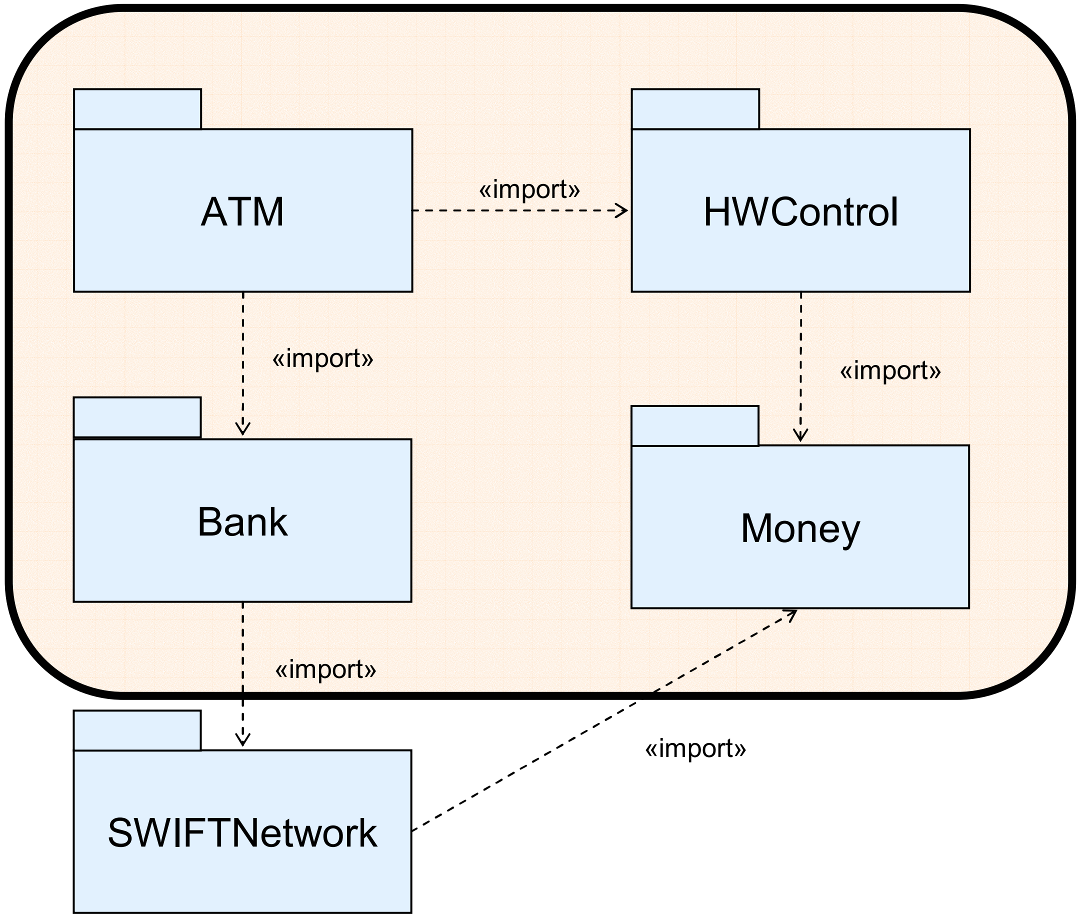
## Concepts beyond TTCN-3

- Unification of test cases:
  - Test case as a composition of test cases
  - Test behavior defines the execution of a test case
- Separation of test behavior and verdict handling
  - Arbiter is a special component to evaluate the verdict
  - Validation actions are used to set the verdict
- Abstract test cases which can use a set of stimulus data
  - Data partitions to describe value ranges for observations and stimuli
- Test architecture with test deployment support
  - Part of the test specification is the definition of deployment requirements for a test case

## An Example

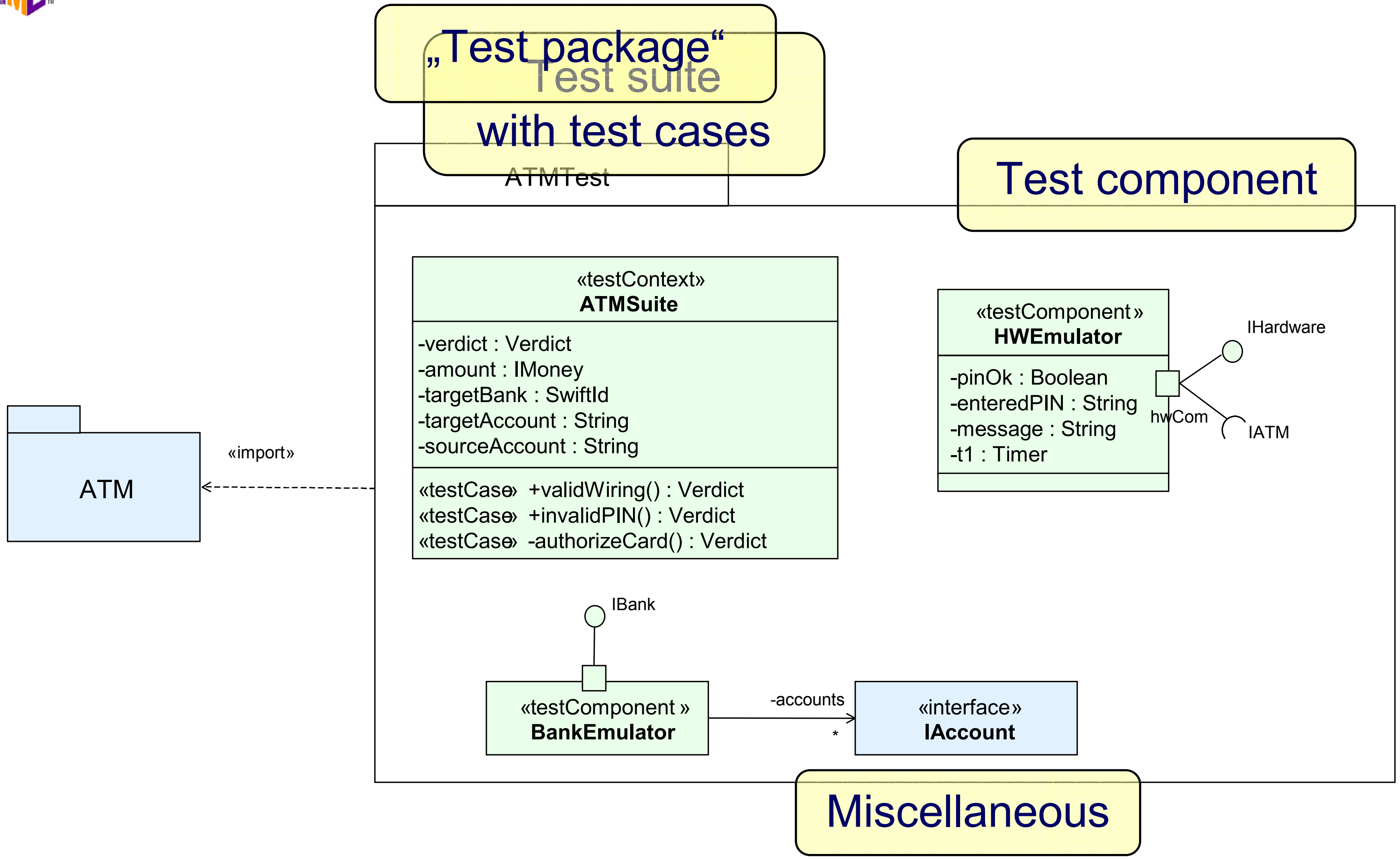

OTC Market Makers
Clearing Company
SWIFTBureau
SWIFTNet
SWIFTBureau
US Bank Network
EU Bank Network
Unit-level tests
System-level tests
US Bank SSSB Client
EU Bank SSSB Client
System-integration level tests

## The Example Packages

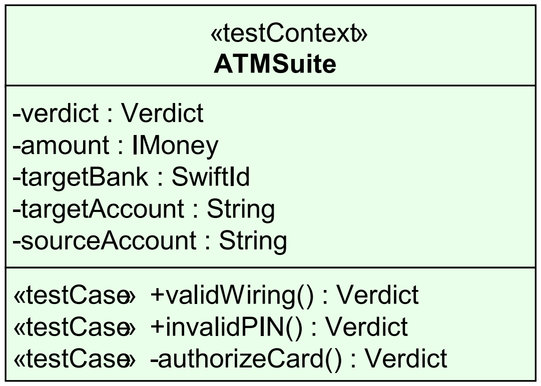


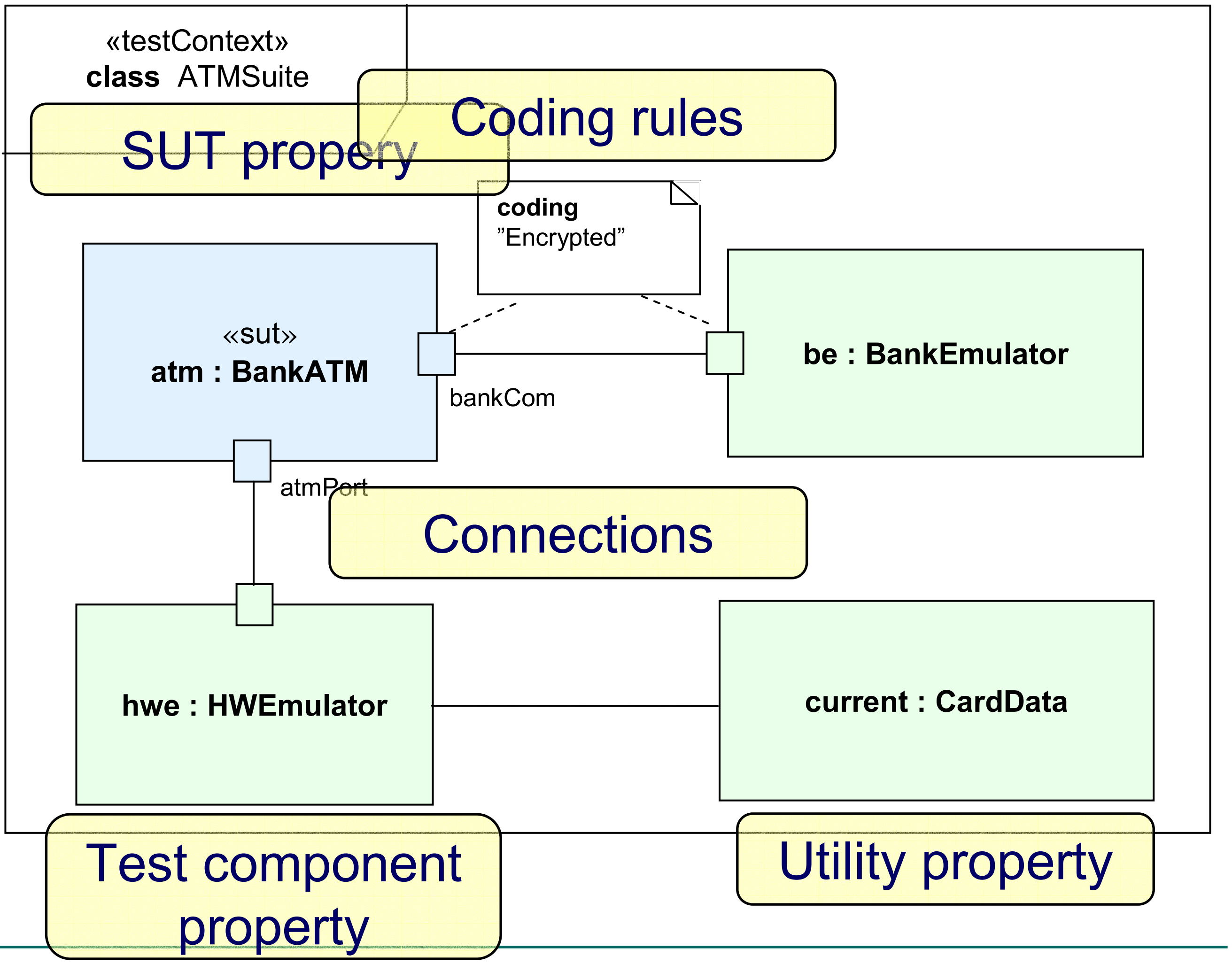
**System Test**

# System Level Test



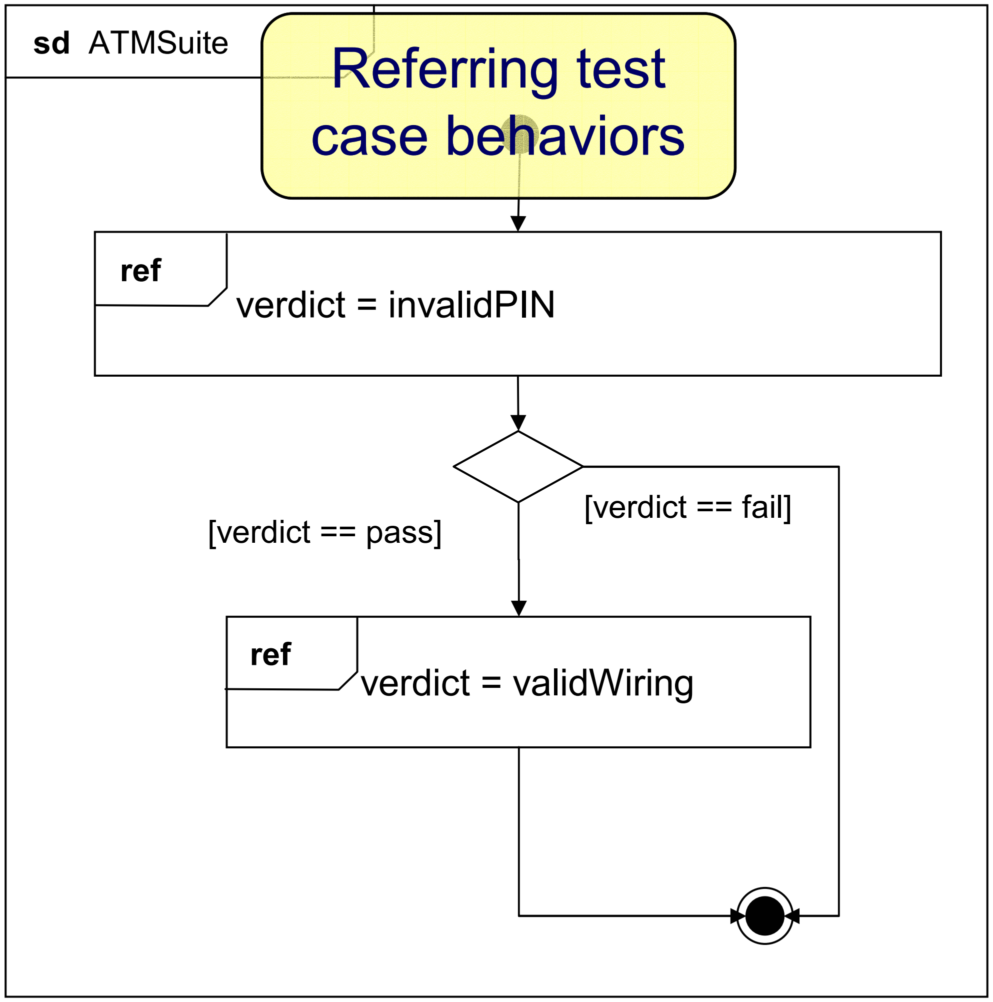
# Test Configuration



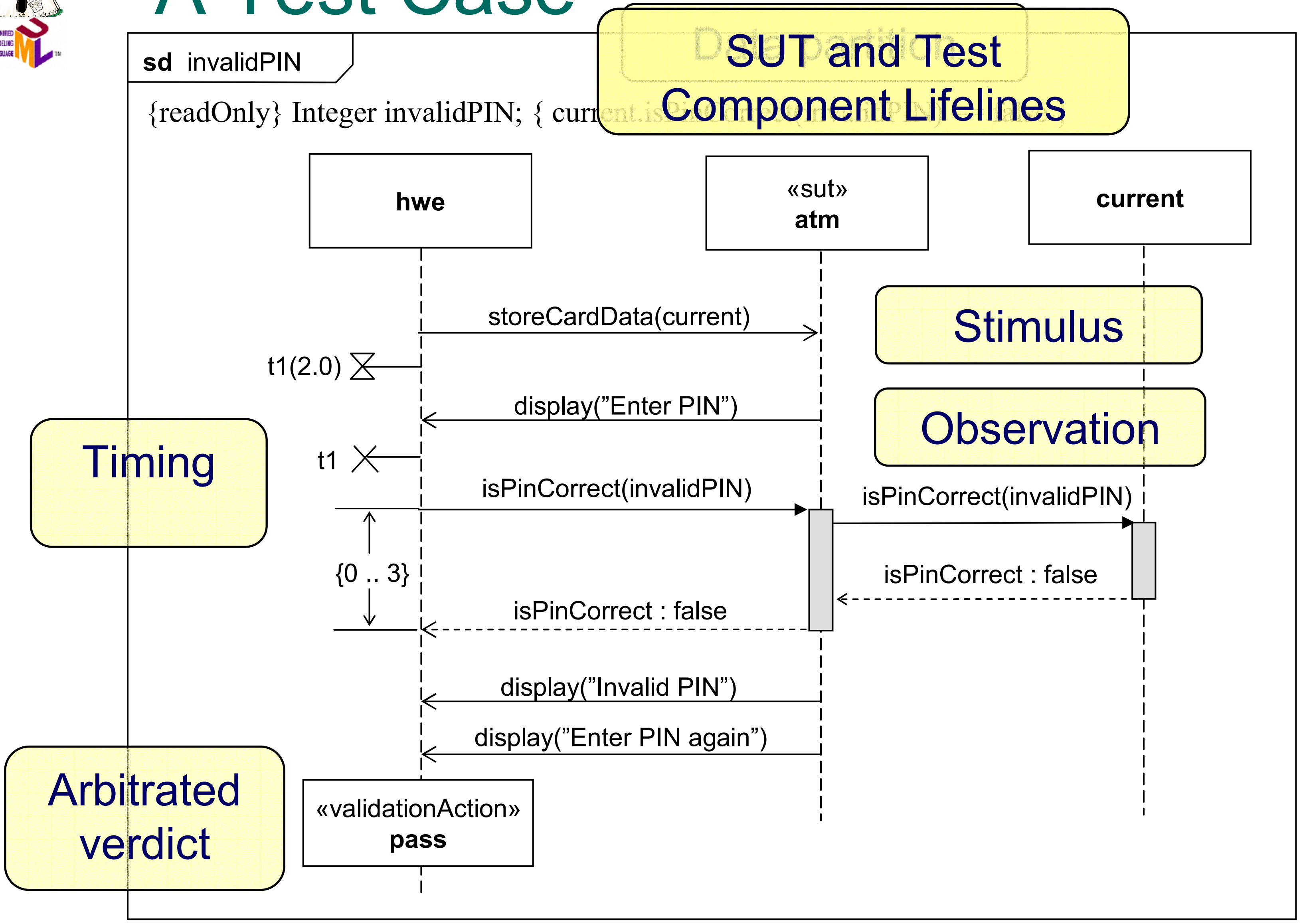


# Test Control

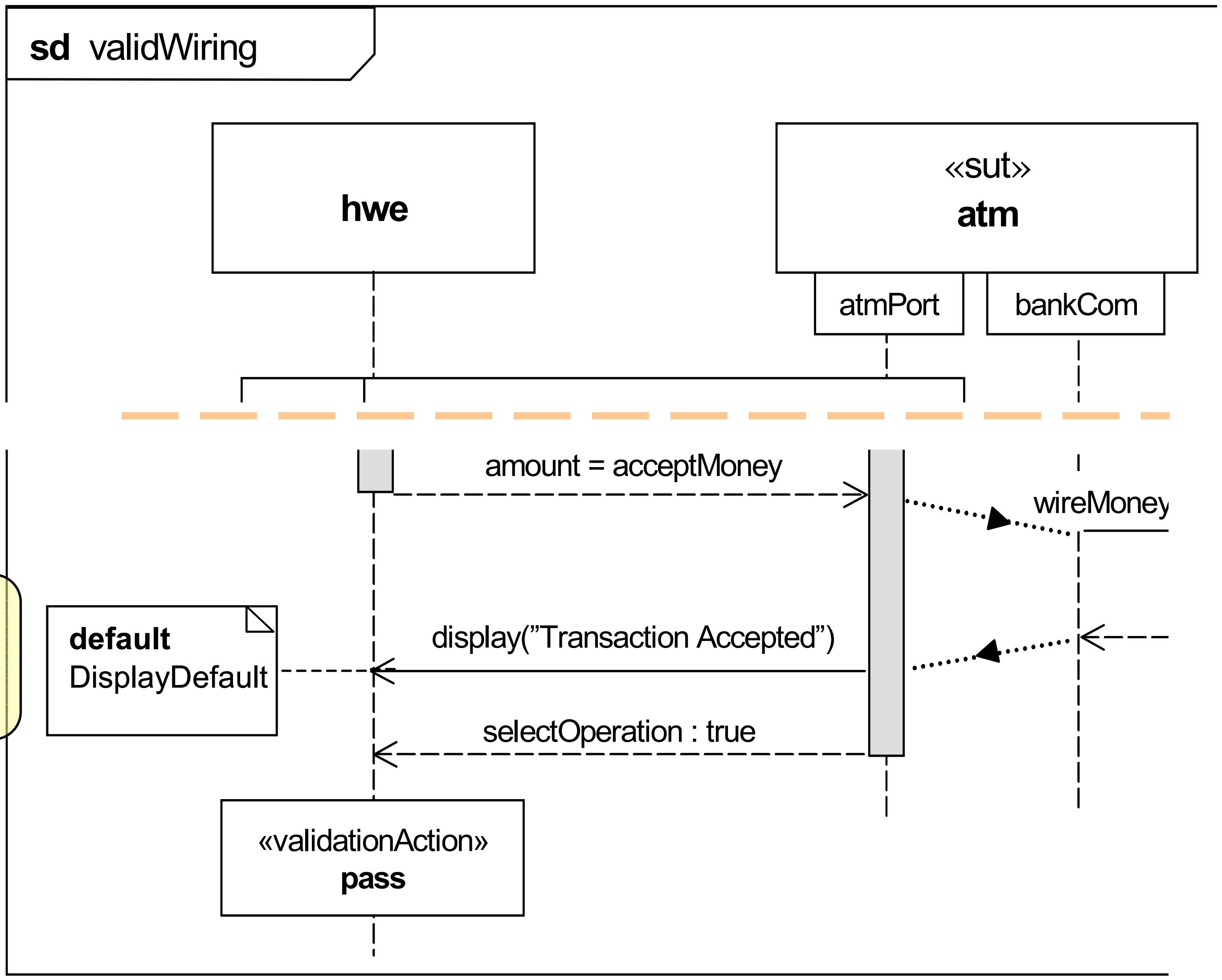
| «testContext» **ATMSuite** |
|---|
| -verdict : Verdict<br>-amount : IMoney<br>-targetBank : SwiftId<br>-targetAccount : String<br>-sourceAccount : String |
| «testCase» +validWiring() : Verdict<br>«testCase» +invalidPIN() : Verdict<br>«testCase» -authorizeCard() : Verdict |



# A Test Case


sd invalidPIN
{readOnly} Integer invalidPIN; { current.isPinCorrect(invalidPIN) == false }
Data partition
SUT and Test Component Lifelines
hwe
«sut» atm
current
storeCardData(current)
t1(2.0)
display("Enter PIN")
t1
isPinCorrect(invalidPIN)
isPinCorrect(invalidPIN)
{0 .. 3}
isPinCorrect : false
isPinCorrect : false
display("Invalid PIN")
display("Enter PIN again")
«validationAction» pass
Stimulus
Observation
Timing
Arbitrated verdict

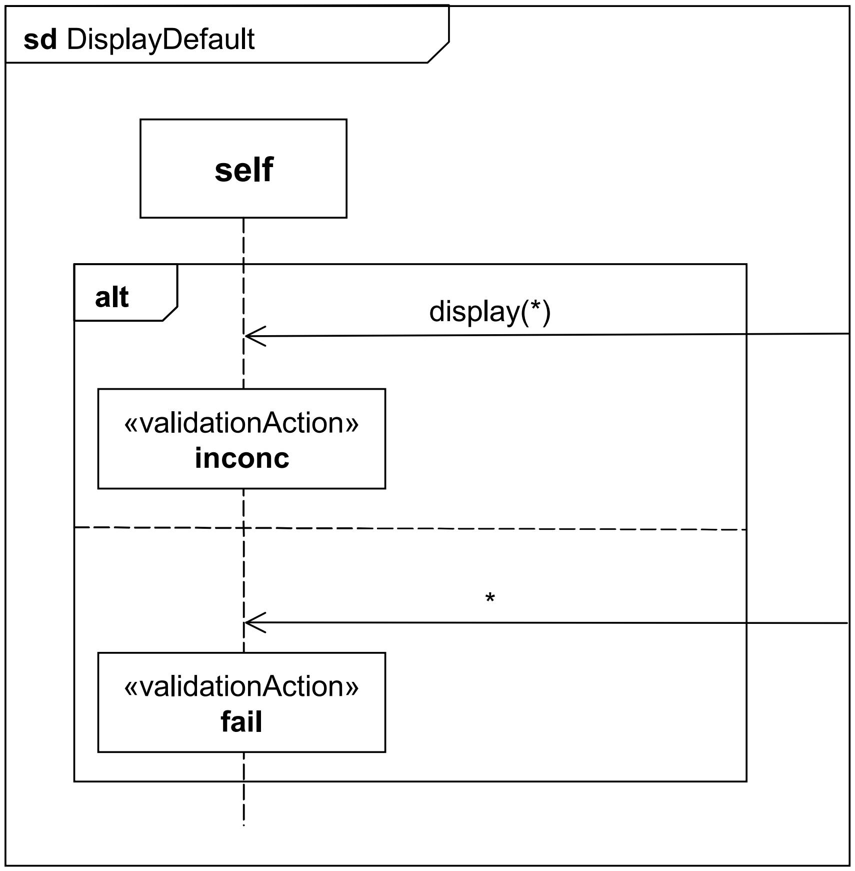
# A Test Case with Default (Extract)



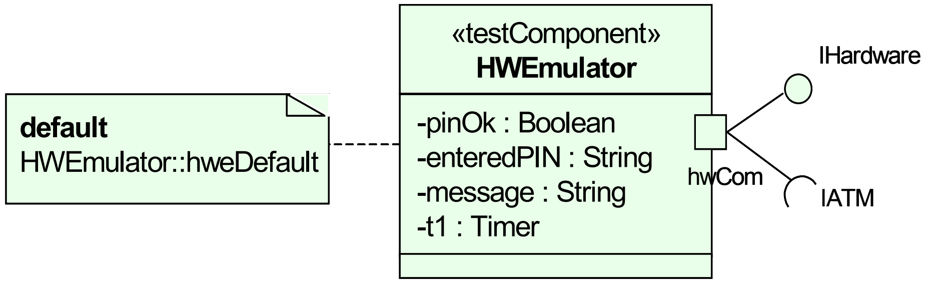
Default application

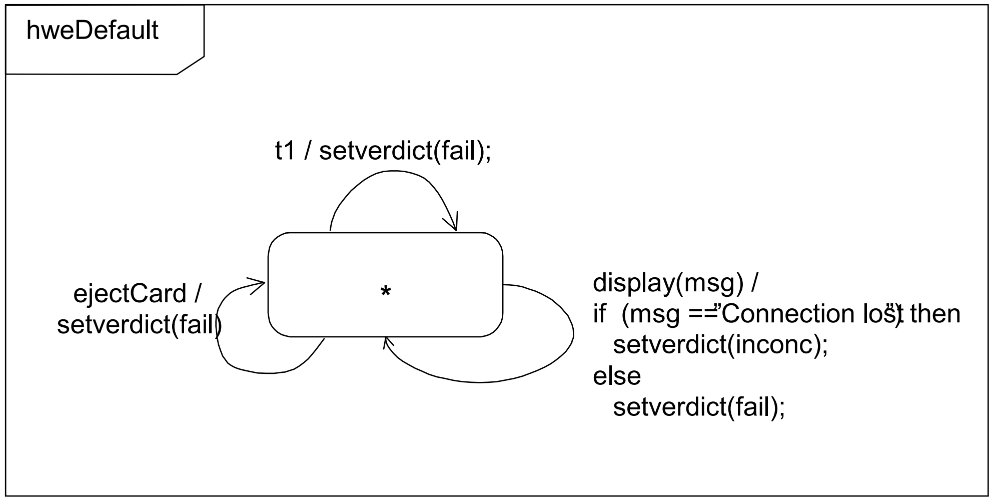
## Defaults

Defining an event-specific default



Applying a component-specific default



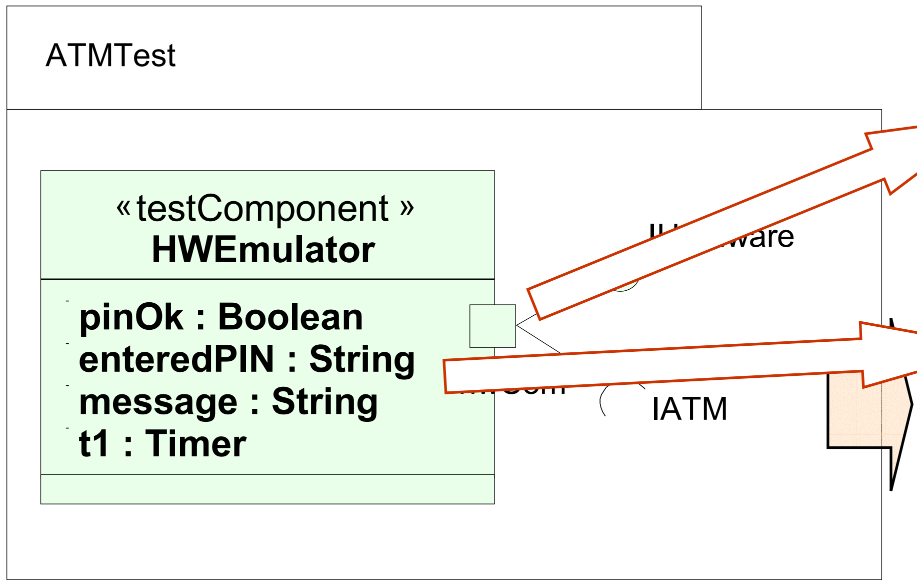


Defining a component-specific default

## The Mappings

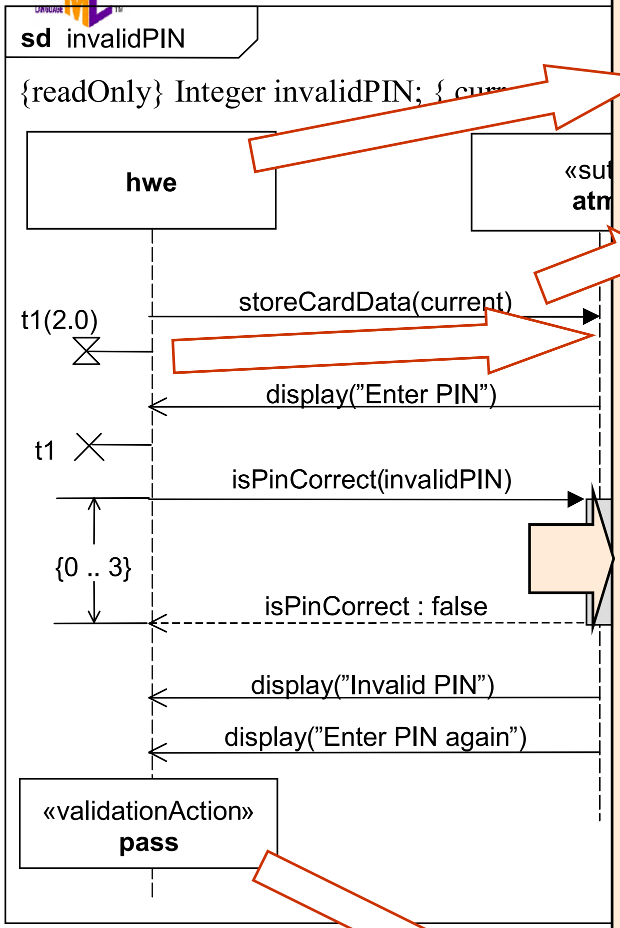
- To enable the direct execution of U2TP specifications by reusing existing test infrastructures
- Mappings to
  - The JUnit test framework
    - An open source test technology for Java
    - Black-box tests on unit level
    - ➢Only selected concepts of U2TP can be mapped
  - The Testing and Test Control Notation TTCN-3
    - A generic test technology by ETSI/ITU-T
    - Black-box/grey-box tests on unit, component, integration and system level
    - ➢Almost all concepts can be mapped

# Example for Mapping to TTCN-3



```
...
type port hwCom_PType
  procedure {...}
...
type component
HWEmulator_CType{
  port atmPort_PType hwCom;
  var boolean pinOk;
  var charstring enteredPIN;
  var charstring message_;
  timer t1;
}
```

# Example for Mapping to TTCN-3



```
function invalidPIN_hwe ... {
 ...
 hwCom.call(
   storeCardData:{current},nowait);
 t1.start(2.0);
 hwCom.getreply(
   display_:{"Enter PIN"});
 t1.stop;
 hwCom.call(
   isPinCorrect:{invalidPIN},3.0) {
 [] hwCom.getreply(
      isPinCorrect:{?} value false) {}
 }
 hwCom.getreply(
   display_:{"Invalid PIN"});
 hwCom.getreply(
   display_:{"Enter PIN again"});
 setverdict(pass); }
```

## Summary

- UML Testing Profile provides specification means for test artifacts of systems from various domains
- Enhances UML with concepts like test configuration, test components, SUT, verdict and default
- Seamlessly integrates into UML: being based on UML metamodel, using UML syntax

➢Direct support for test design

➢Integration with the system development process

➢Finalized at OMG TM in St. Louis, Apr. 2004

## Implementations under Development

- Eclipse Project Hyades on an Open Source Trace and Test Framework
  - The test part is based on the U2TP specification
- Microsoft Visual Studio
- Telelogic Tau G2
- ITEA Project on Advanced Test Methods and Tools TTmedal

Thank you
for your attention!

→ www.fokus.fraunhofer.de/u2tp

*Questions?*