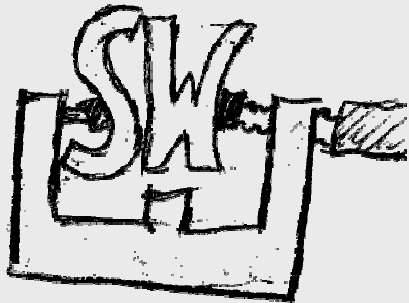


Model-based Testing with UML applied to a Roaming Algorithm for Bluetooth Devices

Jens Grabowski

- Institute for Informatics -
Software-Engineering for
Distributed Systems Group
University of Göttingen



Contents

- Introduction: UML and Testing
- The UML 2.0 Testing Profile
- Bluetooth Case Study
- Implementation of Test Cases
- Summary and Outlook

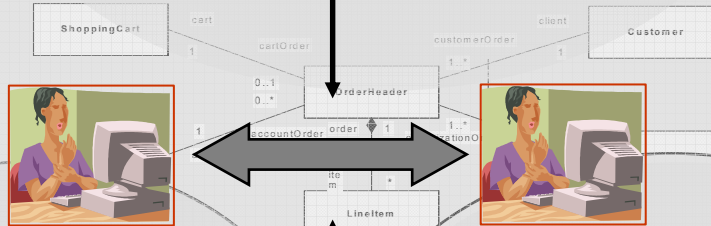
UML ...

Developers

use case diagrams

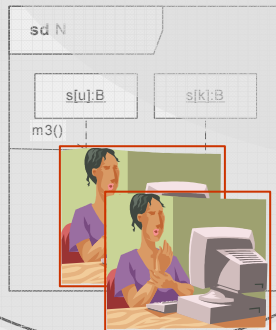


class diagrams

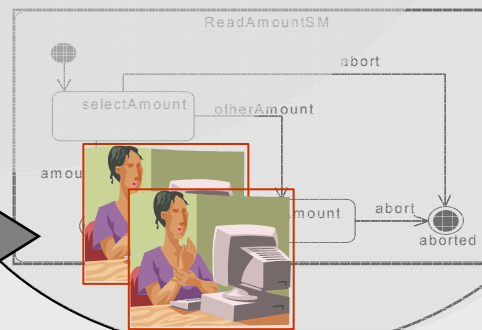


Integrators

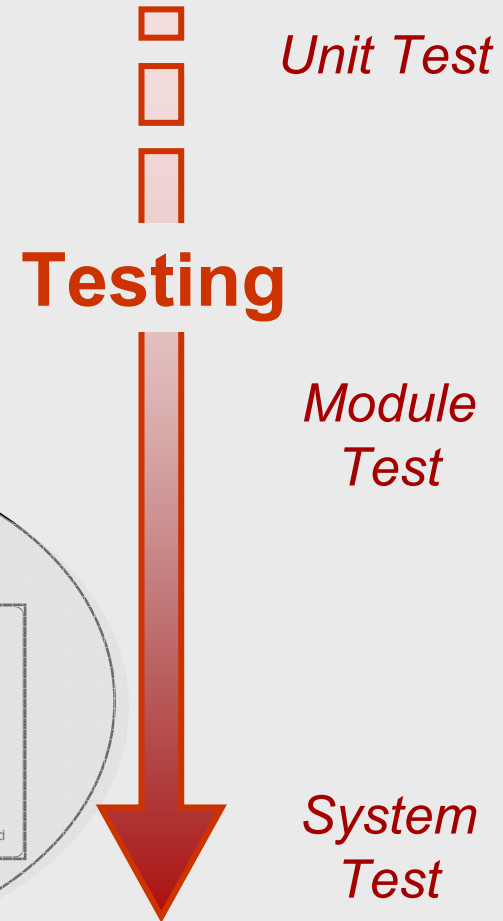
interactions



state machines



System Integrators



Unit Test

Module Test

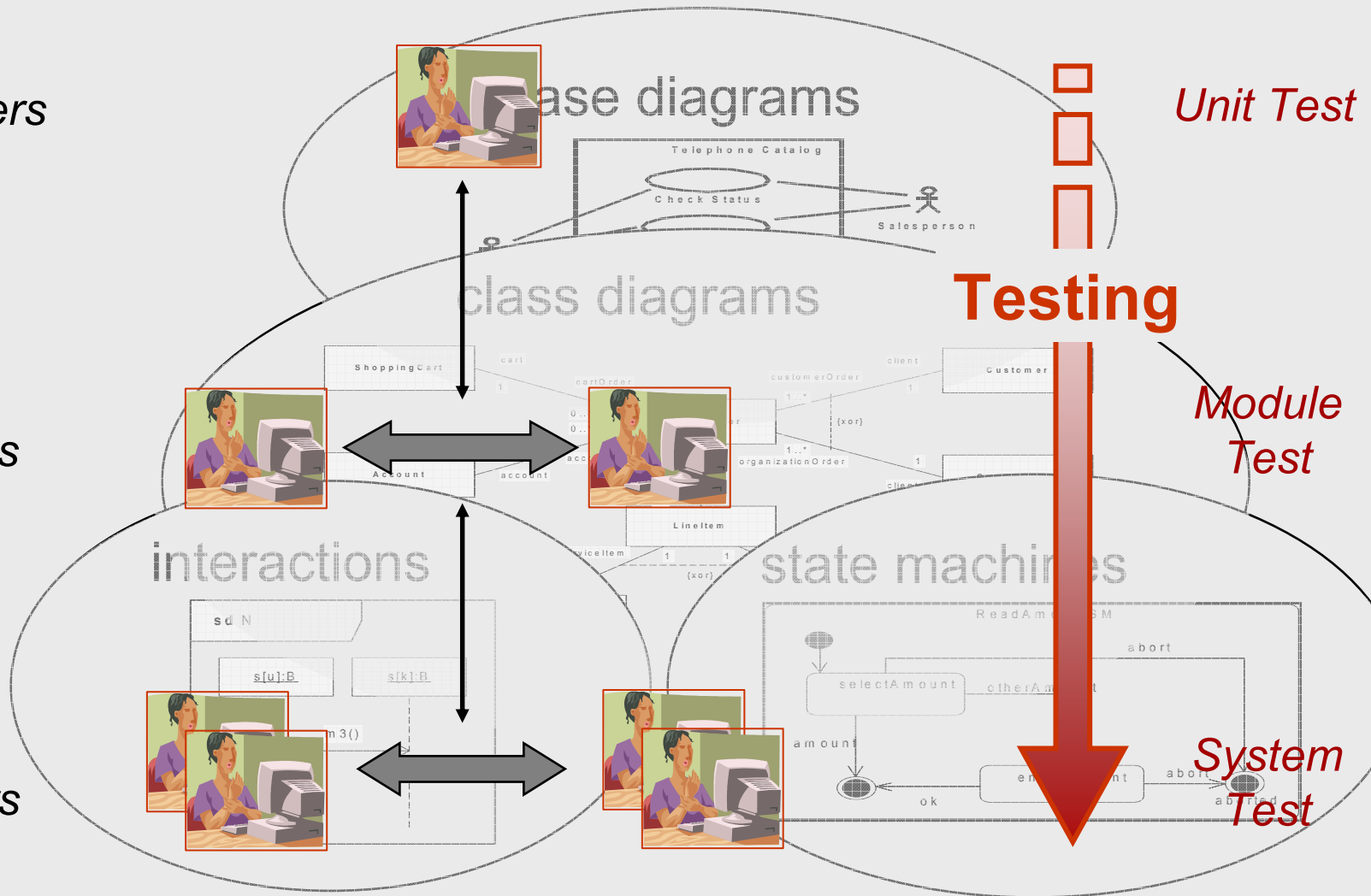
System Test

... and Testing

Developers

Integrators

System Integrators



Projects on UML and Testing

- UML-based test case generation, e.g.,
 - Autolink (SDL/MSD), Telelogic & Uni Lübeck, until 2001
 - Integrationtesting, Siemens 2000
 - Framework for Component Test (TINA), FOKUS 2000
 - Component Testing (COTE), INRIA, 2001
- UML-based notations for testspecification, e.g.,
 - Agedis, EC IST Project
 - TeLa, COTE Project
 - UML 2.0 Testing Profile, OMG

Contents

- Introduction: UML and Testing
- **The UML 2.0 Testing Profile**
- Bluetooth Case Study
- Implementation of Test Cases
- Summary and Outlook

UML 2.0 Testing Profile (U2TP)

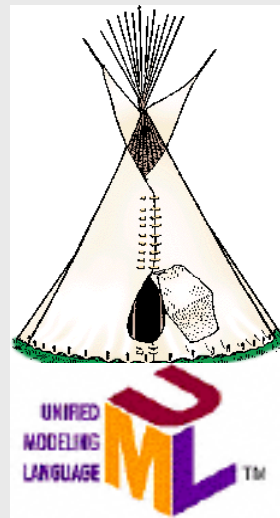
- **Profile:** Standard UML extension mechanism
- **Start:** End of 2001
- **Goal:** Making UML applicable for Software-
and System-Testing
- **Adopted:** June 2003
- **Approved:** April 2004

UML 2.0 Testing Profile Consortia

Software Developers
(UML, JUnit)
• Ericsson
• IBM

UML Tool Manufacturers
• Rational (meanwhile IBM)
• Telelogic
• Softeam

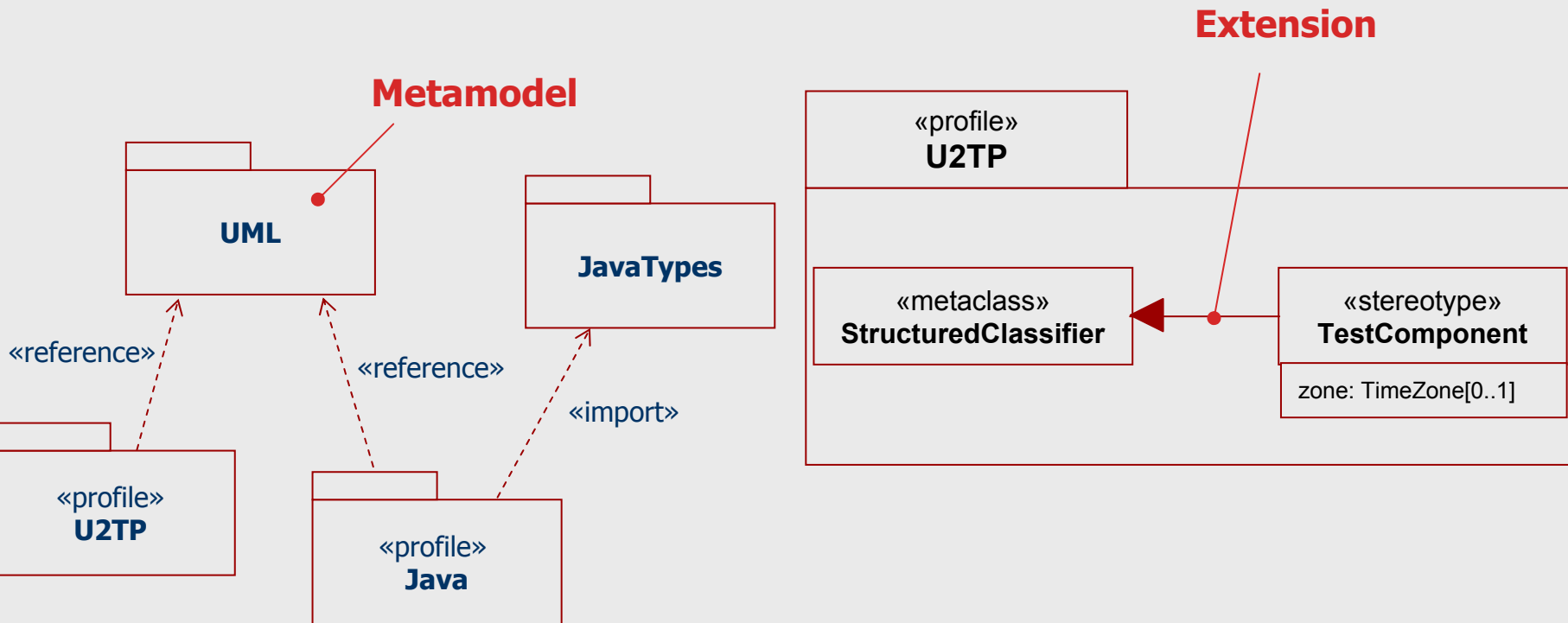
Telecommunications
(TTCN-3, ISO 9646)
• Fraunhofer FOKUS
• **Uni Göttingen/Lübeck**
• Motorola



U2TP

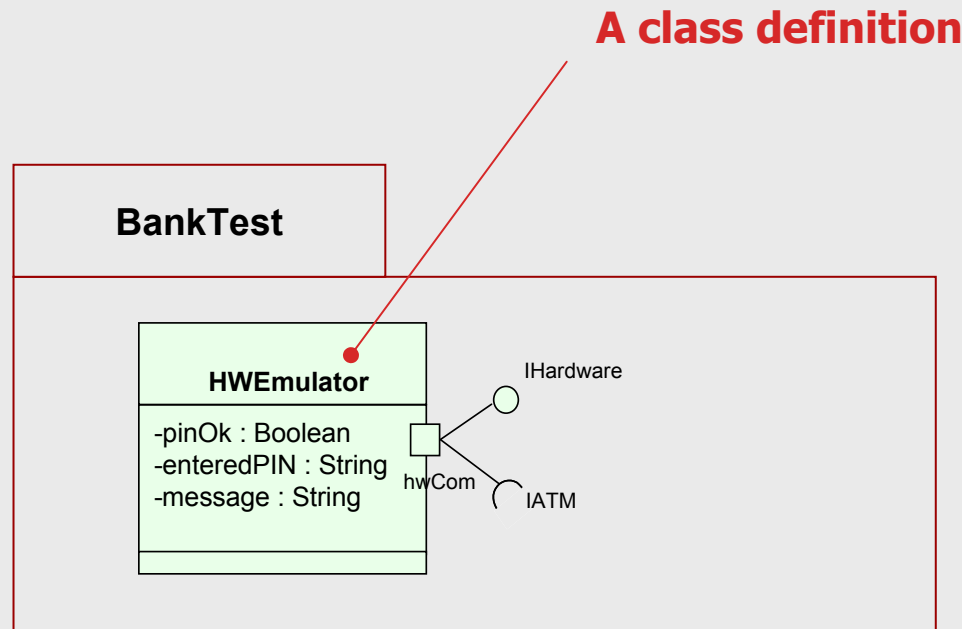
UML 2.0 Profile Walkthrough (1)

- Define profile(s)
 - based on reference metamodel
 - may use other packages for its definition



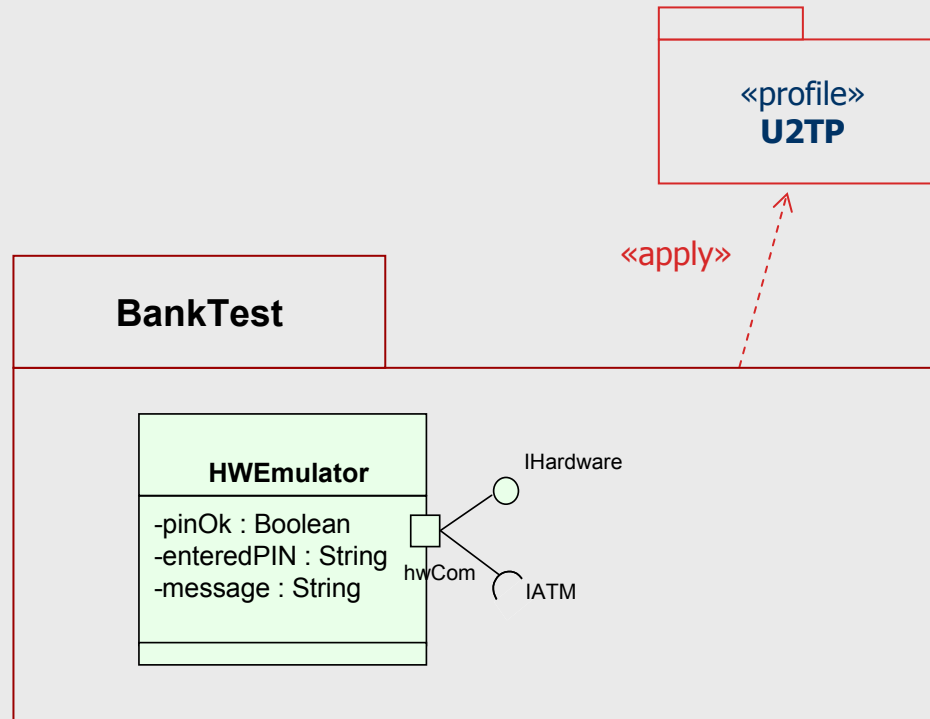
UML 2.0 Profile Walkthrough (2)

- Specify model
 - based on UML metamodel



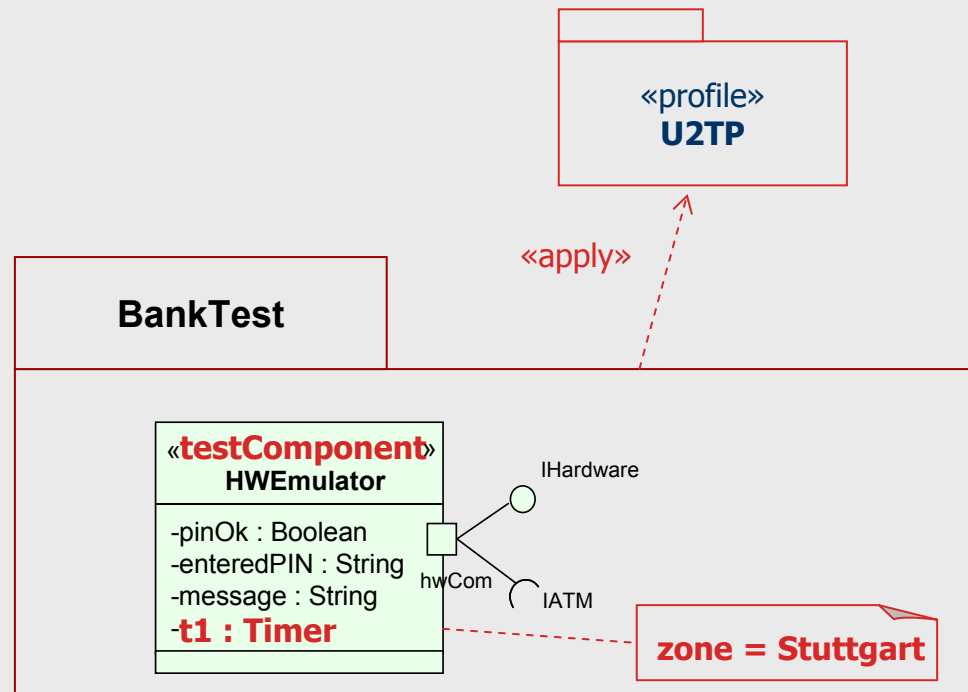
UML 2.0 Profile Walkthrough (3)

- Apply profile(s) to model
 - make it possible to apply stereotypes of the profile to the model elements



UML 2.0 Profile Walkthrough (4)

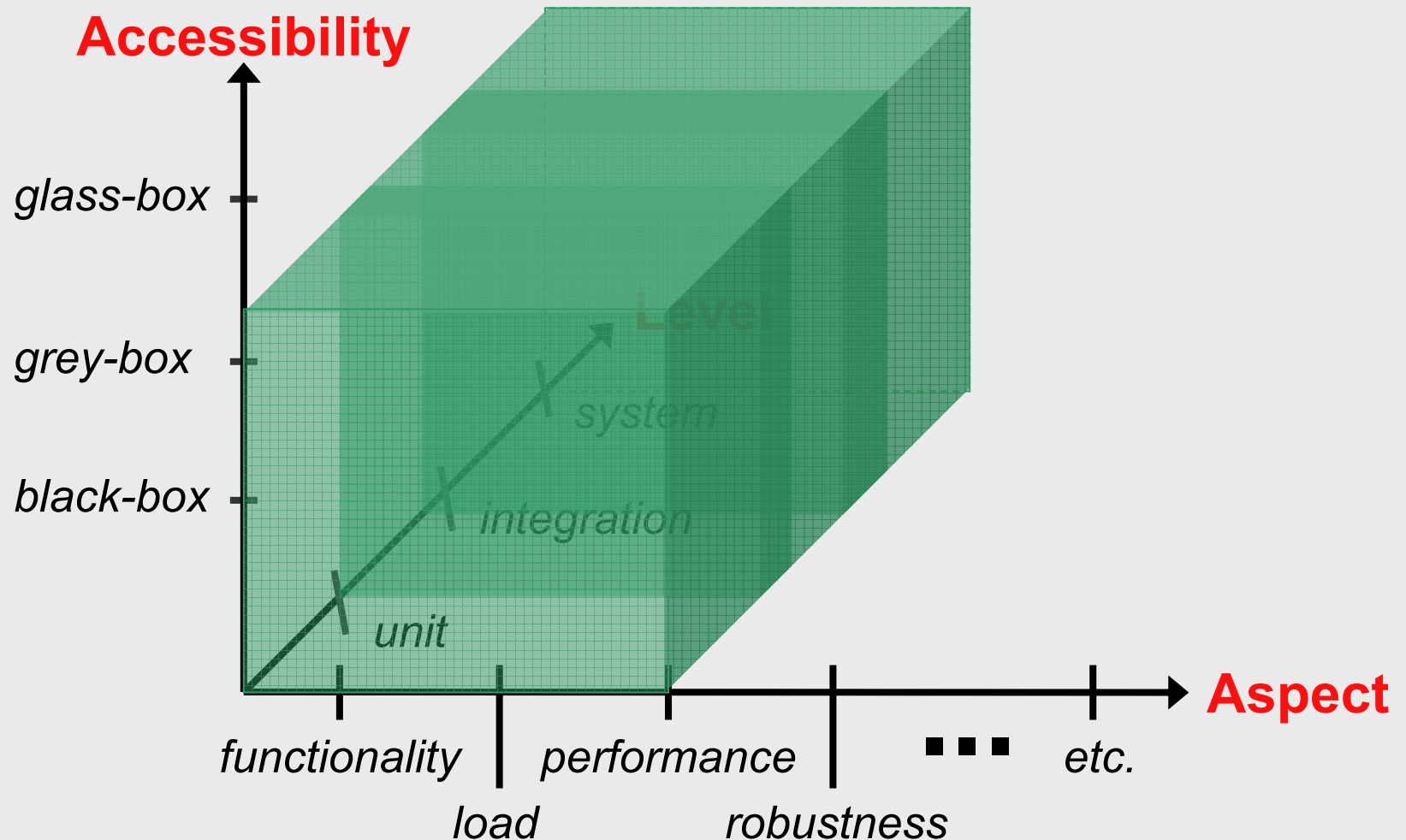
- Apply stereotypes to model elements as desired



Testing Profile Concepts

Architectural concepts	Behavioral concepts	Data concepts	Time concepts
SUT	Test objective	Wildcards	Timer
Test components	Test case	Data partition	Time zone
Test suite	Defaults	Coding rules	
Test configuration	Verdicts		
Test control			
Arbiter			
Utility part			

Types of Testing



A Methodology on the Testing Profile

Test Architecture Concepts	Test Behavior Concepts	Test Data Concepts	Time Concepts
SUT	Test objective	Wildcards	Timer
Test components	Test case	Logical partition	Time zone
Test suite	Defaults	Coding rules	
Test configuration	Verdicts		
Test control			
Arbiter			
Utility part			

Mandatories

Optionals

Derivables

Contents

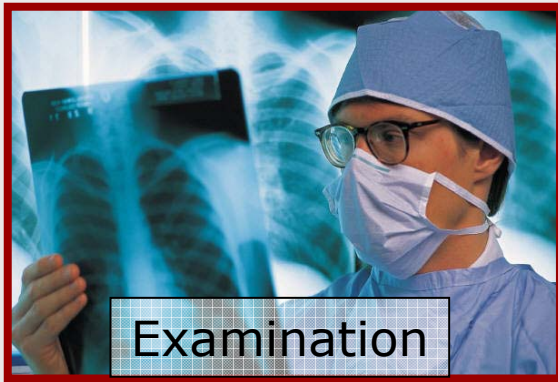
- Introduction: UML and Testing
- The UML 2.0 Testing Profile
- **Bluetooth Case Study**
- Implementation of Test Cases
- Summary and Outlook



The Bluetooth Application

- Short-range (10 m) radio transmission standard
- Bluetooth based transmission of medical sensor data in hospitals (ECG, SPO₂ etc.)
- Continuous monitoring
 - examination → preparation → surgery
→ intensive care

The Bluetooth Case Study



Movement of
the patient →

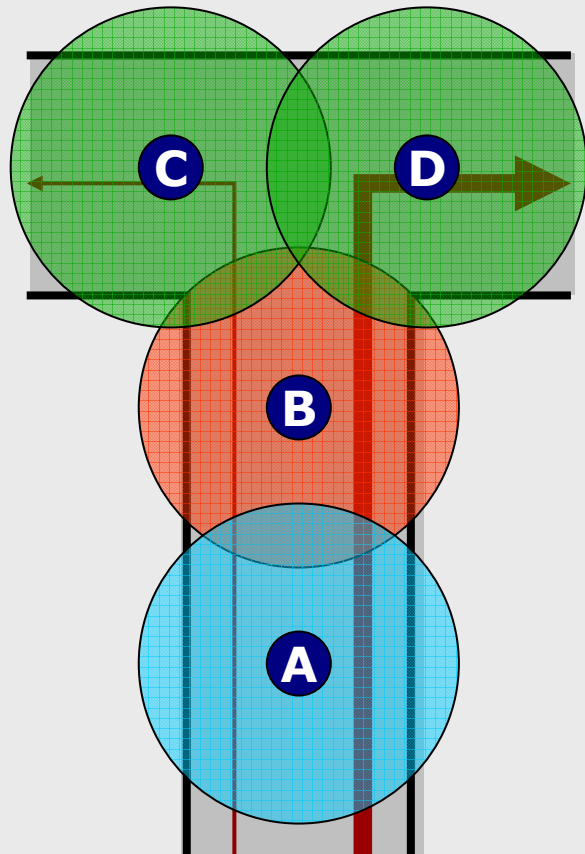


- Patient is moving, mobile sender fixed at patient (**Slave**).
- Area covered by multiple receivers (**Masters**) connected by the fixed network.
- **Roaming** from one master to the next is needed.
- **But:** No roaming functionality defined in the current Bluetooth standard!

The Roaming Approach

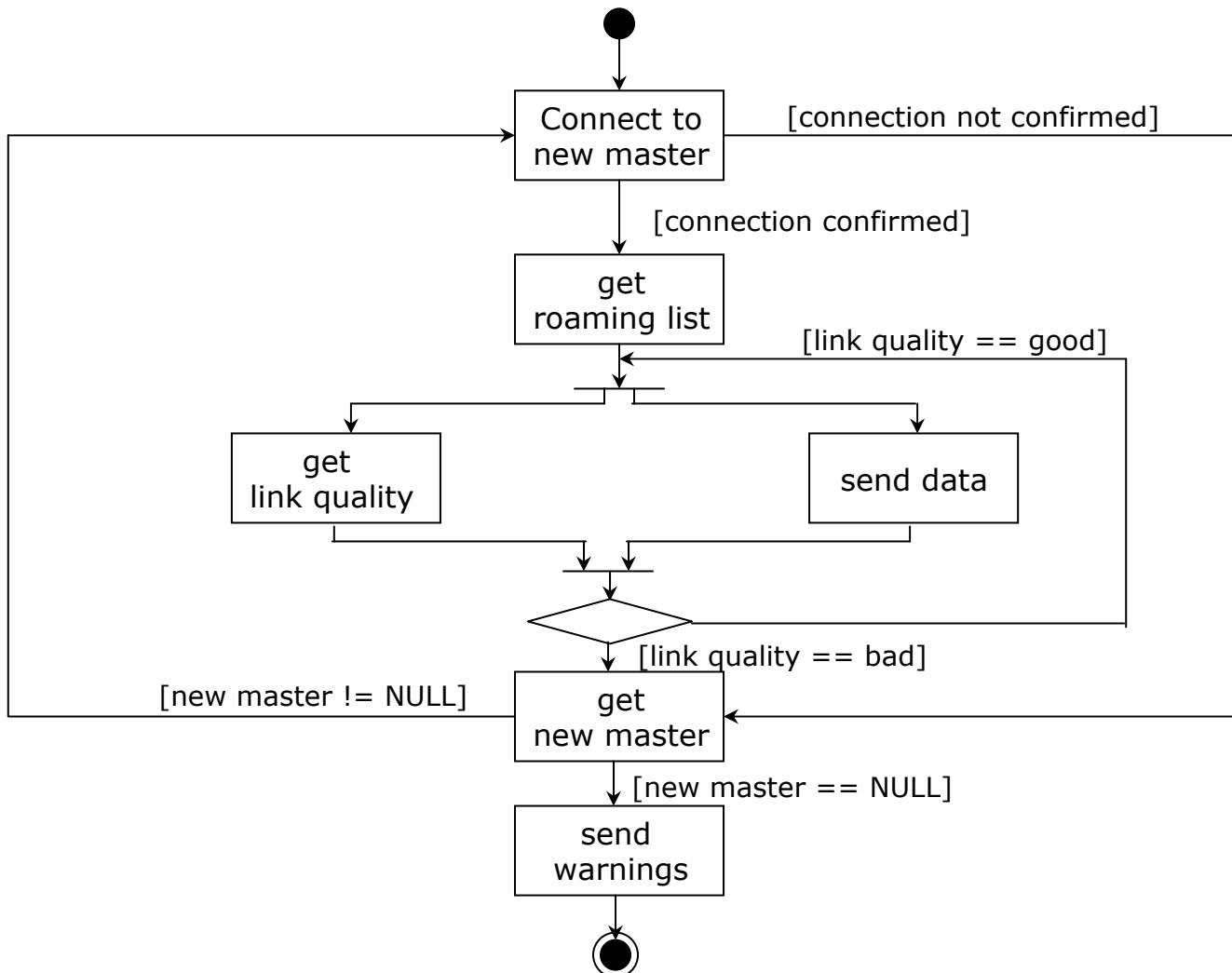
- Need for roaming determined by **quality of link** between master and slave.
- Every slave owns a **roaming list**.
 - Potential new masters defined by roaming lists.
- Roaming lists are calculated by a **Location Server**.
 - Location Server knows about network topology and the positions of slaves.

Roaming: Example



- Current master: B
- Last visited master: A
- Roaming list:
 1. Master D
 2. Master C
 3. Master A

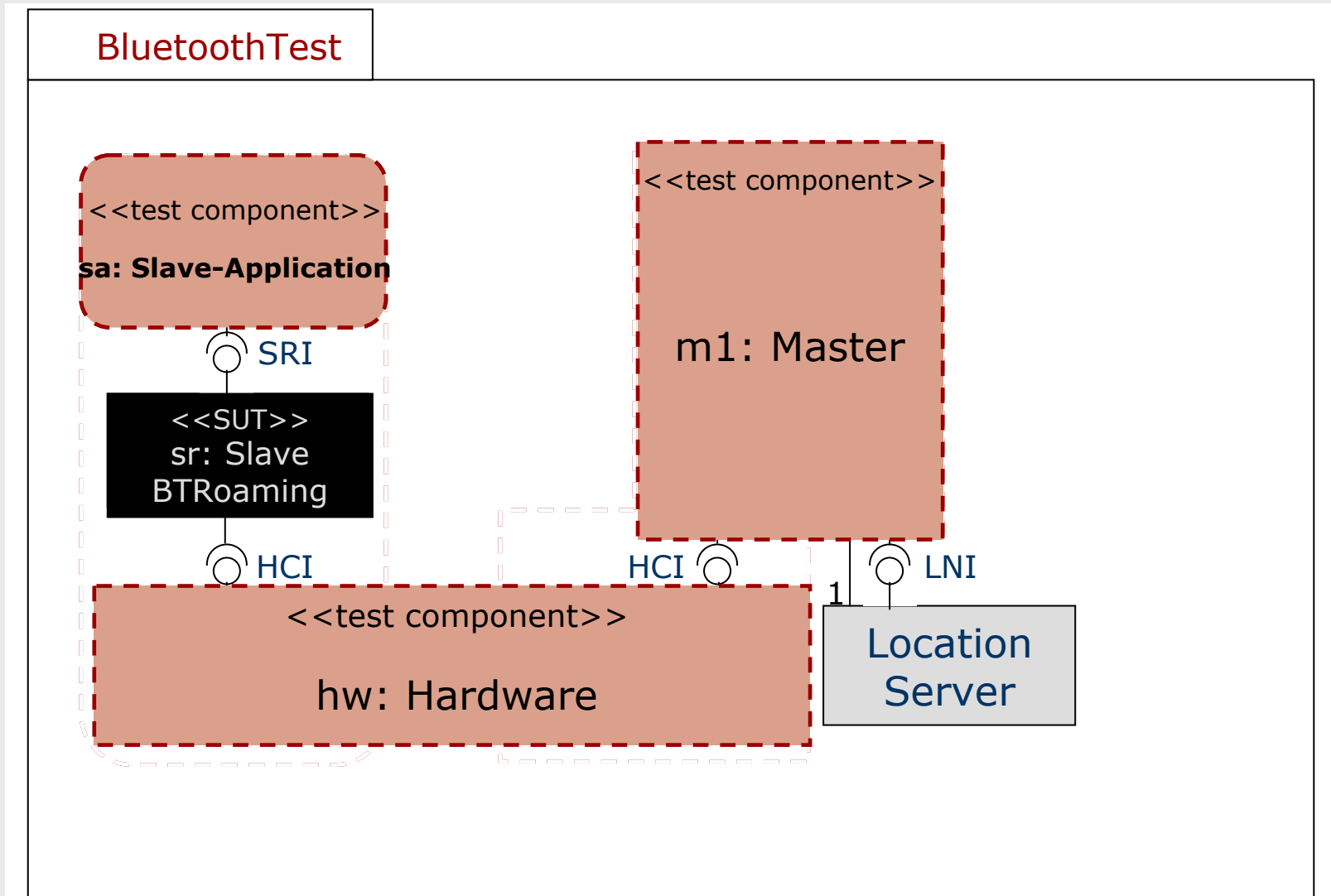
Roaming Algorithm for Slave



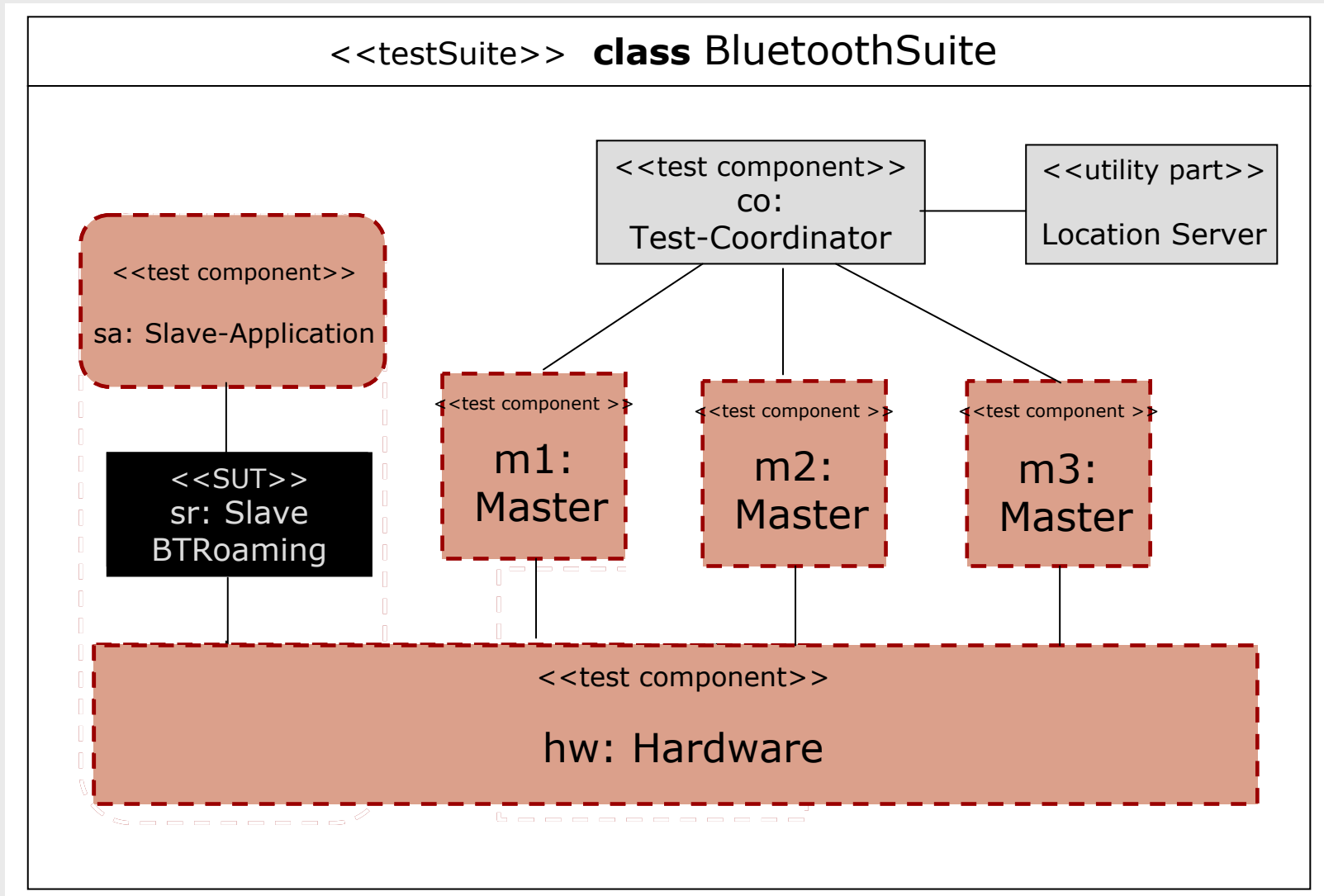
Applying Methodology for Test Specification

- Test preparations
- Test configuration
- Test cases
- Defaults

Test preparations



Test configuration



Test Behavior

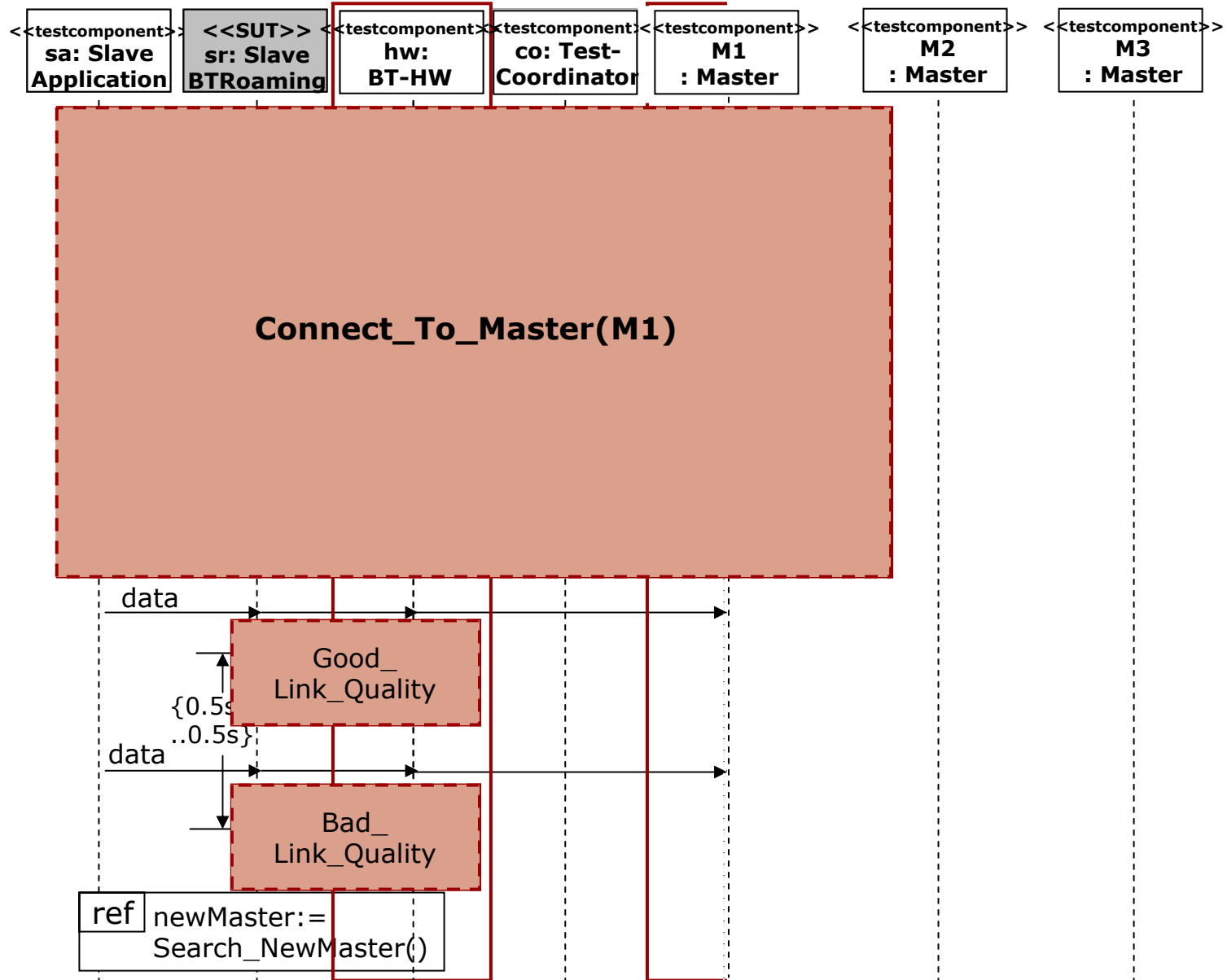
■ Test Objective:

After the exchange of two data packages, the link quality between Slave and its current master M1 becomes bad.

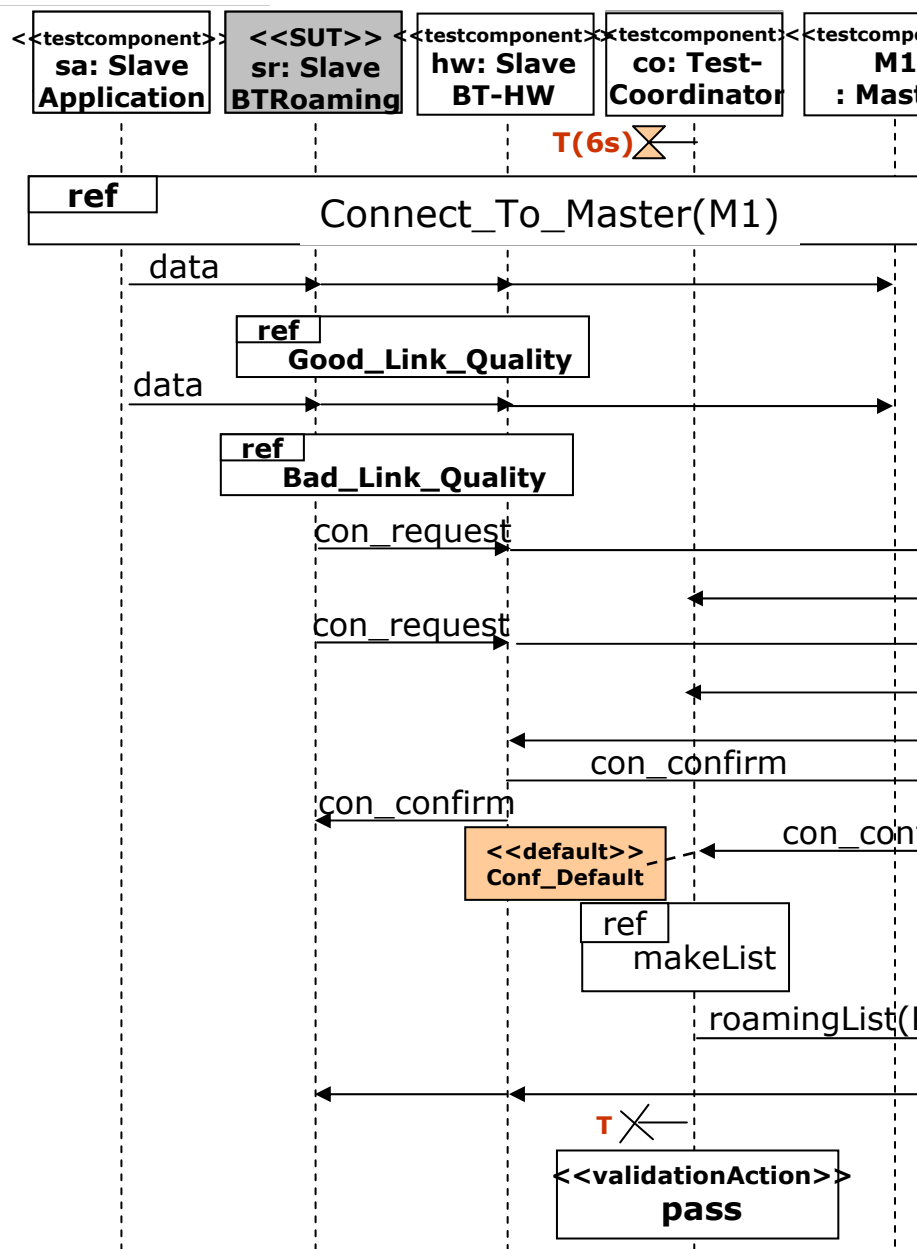
The first of the masters in the roaming list M2 cannot be reached.

Thus, the next master in the roaming list M3 is chosen and the connection is established successfully.

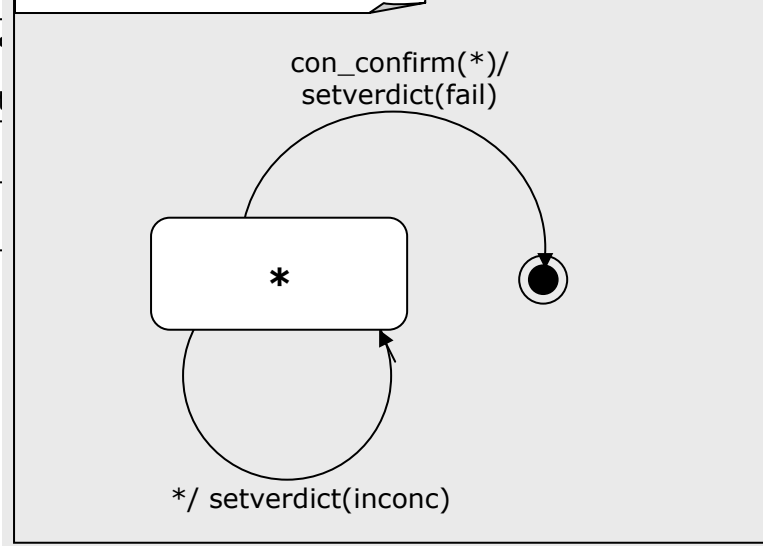
sd TestRoaming_noWarning():Verdict



sd TestRoaming_noWarning():Verdict



statemachine Conf_Default



Methodology Wrap-Up

- Test preparations
 - Define a test package with a test suite.
 - Determine SUT and import its implementation.
 - Determine test objectives.
- Test configuration
 - Define concrete test configuration
 - Assign roles/attach stereotypes
 - SUT, test components, utility part.
- Test cases
 - Group instances.
 - Attach stereotypes.
 - Identify and define re-usable test fragments.
 - Assign verdicts.
 - Specify timers.
 - Specify default behaviors

Contents

- Introduction: UML and Testing
- The UML 2.0 Testing Profile
- Bluetooth Case Study
- **Implementation of Test Cases**
- Summary and Outlook

Implementation of Test Cases

Developers

use case diagrams



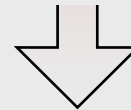
class diagram

Testing

Testing tight to Development



e.g. JUnit



Testing tight to Specification

e.g. TTCN-3

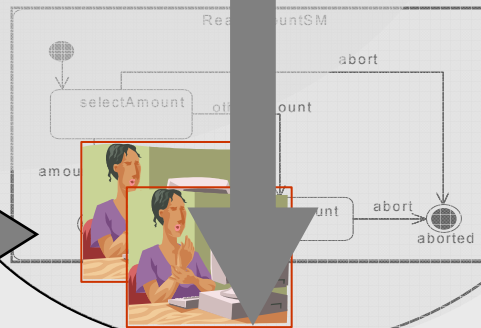


Integrators

interactions

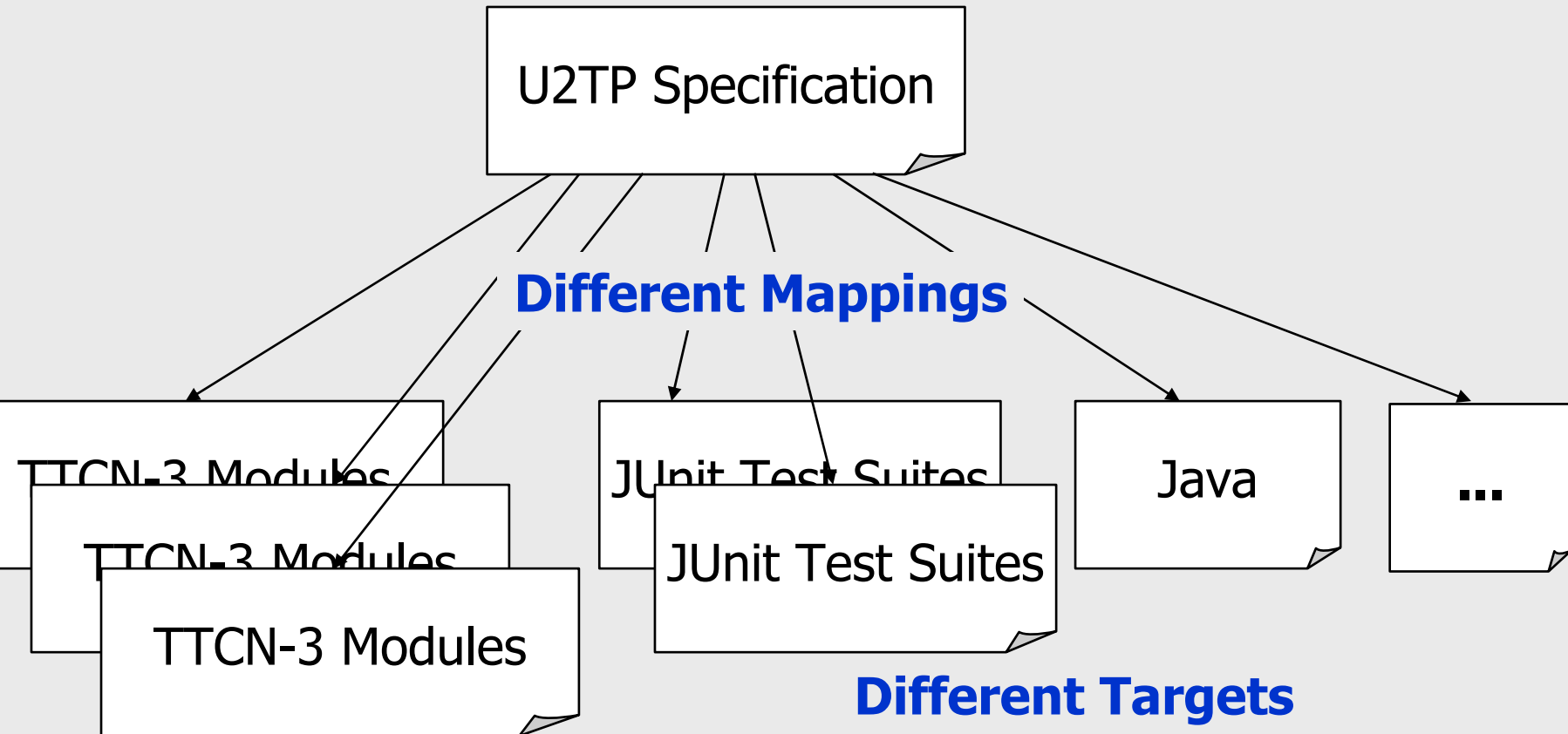


state machines



System Integrators

Mappings



- The mappings define possible, but not the only mappings

Principles of Mapping to TTCN-3

Direct mapping	
Test control	<code>control of a test module</code>
Test case	<code>testcase</code>
Stimulus, observation, coordination	<code>communication to SUT and to test components</code>
Defaults	<code>altsteps and activate/deactivate operations</code>
Wildcards and matching	<code>template and pattern mechanisms</code>
Test components	<code>component types and instances</code>
Test configuration	<code>create, connect, map ... operations</code>

Principles of Mapping to TTCN-3

Direct mapping

Arbiter	<code>user defined verdict handling (and user defined verdict type)</code>
Validation action	<code>(external) data function</code>
Utility part and data pool	<code>(external) constants or module parameters</code>
Coding rule	<code>encoding attribute</code>
Timer	<code>timer</code>
Logging	<code>log operation</code>

Principles of Mapping to TTCN-3

Implicit mapping

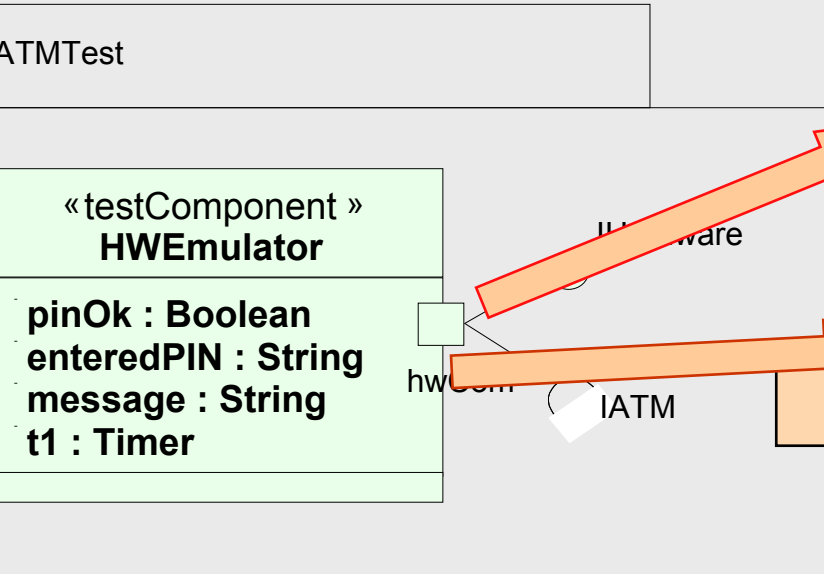
Test objective	<code>comment for a test case in a special attribute</code>
SUT	<code>characterized via the test system interface only</code>

Not supported

Timezone

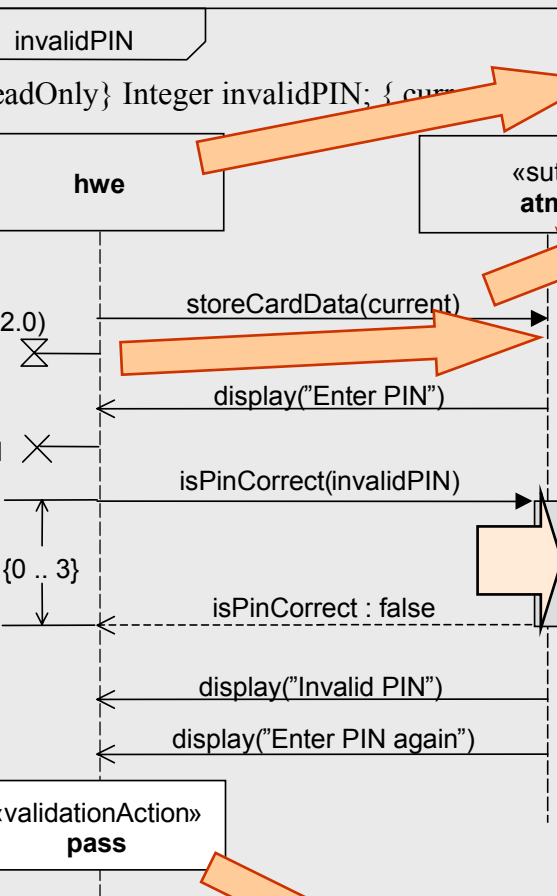
Test trace

Example for Mapping to TTCN-3



```
...
type port hwCom_PType
  procedure {...}
...
type component
  HWEulator_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); }  
}
```

Contents

- Introduction: UML and Testing
- The UML 2.0 Testing Profile
- Bluetooth Case Study
- Implementation of Test Cases
- **Summary and Outlook**

Summary

- UML Testing Profile enhances UML with test concepts.
- The methodology outlines how to transform a design model into a test model.
- High reusability of the design model.
- Different mappings provide an easy test implementation with well-established testing technologies.

Outlook

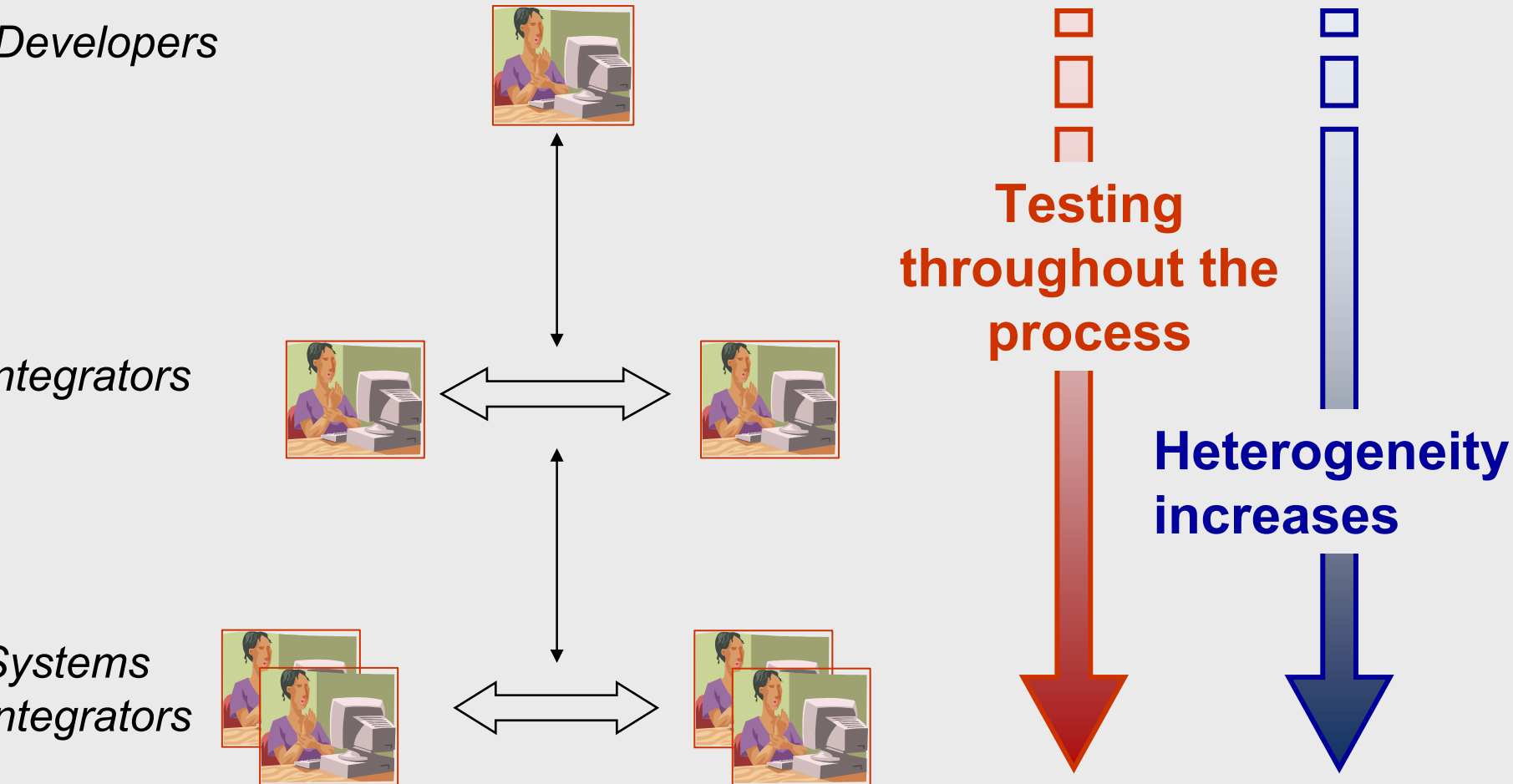
- U2TP tool support
 - IBM/Rational: Eclipse Project **Hyades**
 - Telelogic: **Tau G2**
 - Microsoft: **Visual Studio**
 - Daimler/FOKUS/Nokia/Testing Technologies:
ITEA-Project **TT-Medal**
- Refined Methodology
- Automatic Test Generation

Work on the Bluetooth case study has been done together with

- Zhen Ru Dai (Fraunhofer FOKUS, Berlin)
- Dr. Helmut Neukirchen (Uni Göttingen)
- Holger Pals (University of Lübeck)

Thank you
for your attention!

Testing in Software Development



The Testing Profile Roots

