

A Systematic Approach to Automatically
Derive Test Cases from Use Cases
Specified in Restricted Natural
Languages

Man Zhang, Tao Yue, Shaukat Ali, Huihui Zhang and Ji Wu
tao@simula.no

Simula Research Laboratory, Oslo, Norway
Beihang University, Beijing, China
SAM 2014

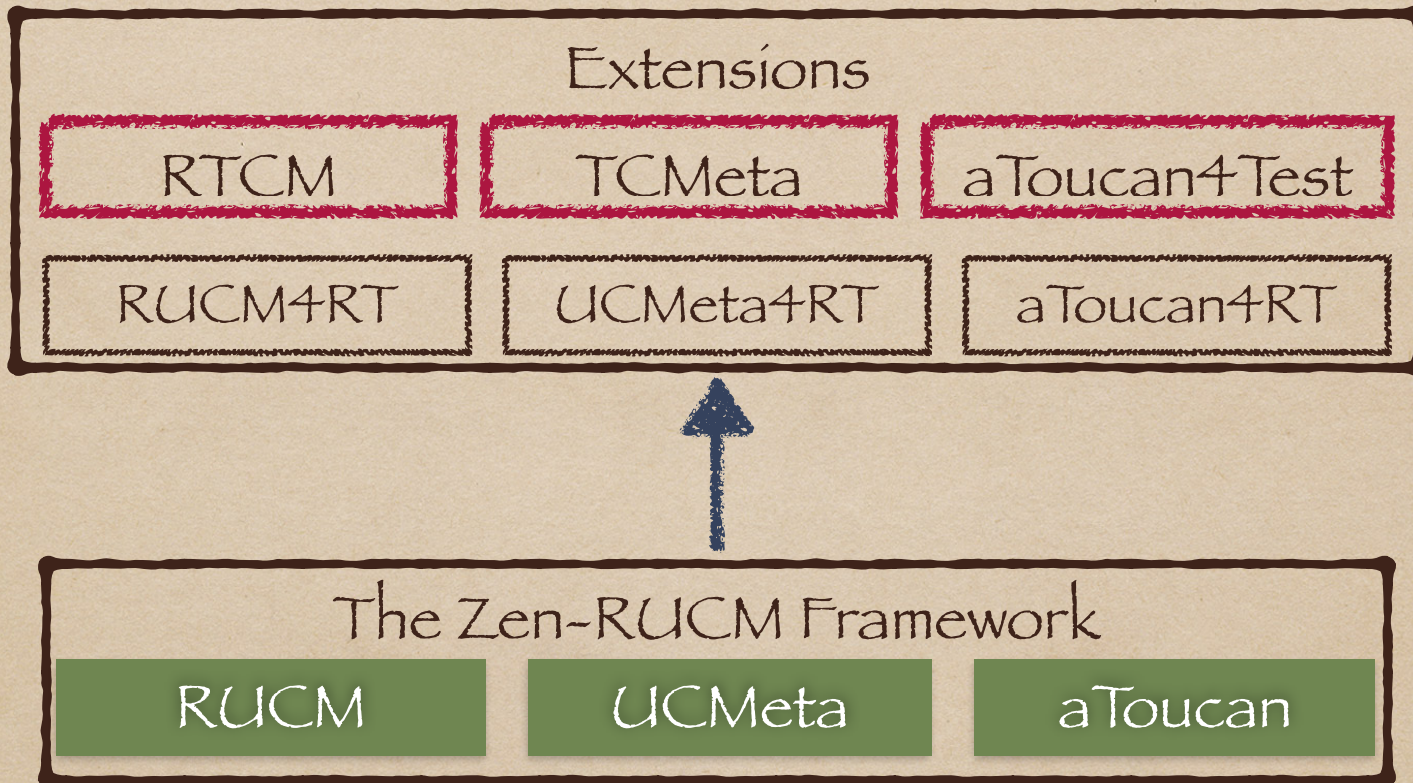
Motivation

- ◆ Current Industrial Practice
 - ◆ Test cases in many domains are derived and executed manually.
- ◆ Drawbacks
 - ◆ The overall process is largely dependent on domain knowledge;
 - ◆ Test cases written in NL are often ambiguous and therefore interpreted differently;
 - ◆ Deriving test cases is not systematic; and
 - ◆ Traceability from requirements to test cases and vice versa is not systematically and automatically established.

Objective

- ◆ Deriving easy-to-understand and manually-executable test cases from textual and informal requirements automatically
- ◆ Establishing traceability links between requirements and tests
- ◆ Implementing coverage criteria to generate effective test cases

Zen-RUCM



Zen-RUCM

<http://www.zen-tools.com/Zen-RUCM.html>

Restricted Use Case Modeling (RUCM)

- ◆ Requirements Specification Language
 - ◆ UML Use Case Diagram
 - ◆ Textual Use Case Specifications
 - ◆ Unique use case template
 - ◆ Carefully selected a set of restrictions including keywords
- ◆ Tool Support: The RUCM Editor

Generic RUCM

Use Case Specification

Use Case Name	Withdraw Fund
Brief Description	ATM customer withdraws a specific amount of funds from a valid bank account.
Precondition	The system is idle. The system is displaying a Welcome message.
Primary Actor	ATM Customer
Secondary Actors	Card Reader
Dependency	INCLUDE USE CASE Validate PIN
Generalization	None

Basic Flow	Steps
(Untitled) ▼	<ol style="list-style-type: none"> 1 INCLUDE USE CASE Validate PIN. 2 ATM customer selects Withdrawal through the system. 3 ATM customer enters the withdrawal amount through the system. 4 ATM customer selects the account number through the system. 5 The system VALIDATES THAT the account number is valid. 6 The system VALIDATES THAT ATM customer has enough funds in the account. 7 The system VALIDATES THAT the withdrawal amount does not exceed the daily limit of the account. 8 The system VALIDATES THAT the ATM has enough funds. 9 The system dispenses the cash amount. 10 The system prints a receipt. 11 The system ejects the ATM card. 12 The system displays Welcome message.
	Postcondition ATM customer funds have been withdrawn.

Generic RUCM

Specific Alternative Flow "alt1" ▼	RFS 8
	1 The system displays an apology message MEANWHILE the system ejects the ATM card.
	2 The system shuts down.
	3 ABORT.
Postcondition	ATM customer funds have not been withdrawn. The system is shut down.
Bounded Alternative Flow "alt2" ▼	RFS 5-7
	1 The system displays an apology message MEANWHILE the system ejects the ATM card.
	2 ABORT.
Postcondition	ATM customer funds have not been withdrawn. The system is idle. The system is displaying a Welcome message.
Global Alternative Flow "alt3" ▼	ATM customer enters Cancel.
	1 The system cancels the transaction.
	2 The system ejects the ATM card.
	3 ABORT.
Postcondition	ATM customer PIN number has not been withdrawn. The system is idle. The system is displaying a Welcome message.

RUCM for Real Time (RUCM4RT)

Use Case Specification

Use Case Name	Synchronize with AutopilotSystemB
Brief Description	the system synchronizes with AutopilotSystemB
Precondition	the system starts successfully
Dependency	EXTENDED BY USE CASE autoPilot::Handle Faults
Generalization	None
Primary Actor	<<Timer>> MainTimer
Secondary Actors	<<ExternalSystem>> AutopilotSystemB
Resources	<<CommunicationMedia>> autoPilot::CCDL
Period	20.0 ms
Time Cost	<=[120.0, us] XOR <=[360.0, us] XOR <=(560.0 us)

RUCM for Real Time (RUCM4RT)

Basic Flow	Steps
"system synchronization" ▼	1 MainTimer sends a pulse to the system
	2 the system closes it's interrupt
	3 the system sends a synchronization signal to AutopilotSystemB
	4 the system gets response from AutopilotSystemB
	5 the system VALIDATES THAT CONSTRAINT rtDConstraint_1 IS SATISFIED
	6 the system VALIDATES THAT the feedback is correct
	7 the system synchronizes with AutopilotSystemB VIA CCDL COMMUNICATION MEDIA
	8 the system sets synchronized state
	9 the system open it's interrupt
Postcondition	the system synchronizes with AutopilotSystemB successfully

Event Observations

@t1	RFS BasicFlow 3	TimeInstantObservation of sending synchronization signal
@t2	RFS BasicFlow 4	TimeInstantObservation of receiving response
&d1	RFS response error 2	TimeDurationObservation of response error
&d2	RFS synError 3	TimeDurationObservation of synchronization error

Timing Constraints

rtDConstraint_1	RFS BasicFlow_{3,4}	$(0.0, \text{us}) < (t2 - t1) \leq (120, \text{us})$
rtDConstraint_2	RFS response error_{2}	$(120, \text{us}) \leq d1 \leq (240, \text{us})$
rtDConstraint_3	RFS synError_{3}	$(0, \text{us}) < d2 \leq (200, \text{us})$

RTCM - Specifying TCS without API Information

Test Case Specification																	
Name	Test_Withdraw Fund																
Brief Description	This test case specification is for testing use case specification that ATM customer withdraws a specific amount of funds from a valid bank account.																
Precondition (Test Data Specification)	The system is idle. The system is displaying a Welcome message.																
Basic Flow (Test Sequence) (Untitled) ▼	<table border="1"> <thead> <tr> <th>Steps</th> <th></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>INCLUDE TEST CASE SPECIFICATION Test_Validate PIN</td> </tr> <tr> <td>2</td> <td>Test System selects Withdrawal through the system</td> </tr> <tr> <td>3</td> <td>Test System enters the withdrawal amount through the system</td> </tr> <tr> <td>4</td> <td>Test System selects the account number through the system</td> </tr> <tr> <td>5</td> <td>Test System or Tester VERIFIES THAT the validation that the account number is valid is <True or False>.</td> </tr> <tr> <td>6</td> <td>Test System or Tester VERIFIES THAT the validation that ATM customer has enough funds in the account is <True or False>.</td> </tr> <tr> <td>7</td> <td>Test System or Tester VERIFIES THAT the validation that the withdrawal amount does not exceed the daily limit of the account is <True or False>.</td> </tr> </tbody> </table>	Steps		1	INCLUDE TEST CASE SPECIFICATION Test_Validate PIN	2	Test System selects Withdrawal through the system	3	Test System enters the withdrawal amount through the system	4	Test System selects the account number through the system	5	Test System or Tester VERIFIES THAT the validation that the account number is valid is <True or False>.	6	Test System or Tester VERIFIES THAT the validation that ATM customer has enough funds in the account is <True or False>.	7	Test System or Tester VERIFIES THAT the validation that the withdrawal amount does not exceed the daily limit of the account is <True or False>.
Steps																	
1	INCLUDE TEST CASE SPECIFICATION Test_Validate PIN																
2	Test System selects Withdrawal through the system																
3	Test System enters the withdrawal amount through the system																
4	Test System selects the account number through the system																
5	Test System or Tester VERIFIES THAT the validation that the account number is valid is <True or False>.																
6	Test System or Tester VERIFIES THAT the validation that ATM customer has enough funds in the account is <True or False>.																
7	Test System or Tester VERIFIES THAT the validation that the withdrawal amount does not exceed the daily limit of the account is <True or False>.																

RTCM - Specifying TCS without API Information

<p>Specific Alt. Flow (Test Sequence) "alt1" ▼</p>	<p>RFS 8</p> <p>1 Test System or Tester VERIFIES THAT The system displays an apology message MEANWHILE Test System or Tester VERIFIES THAT the system ejects the ATM card</p> <p>2 Test System or Tester VERIFIES THAT The system shuts down.</p> <p>3 ABORT.</p> <p>Postcondition (Test Oracle) ATM customer funds have not been withdrawn. The system is shut down.</p>
<p>Bounded Alt. Flow (Test Sequence) "alt2" ▼</p>	<p>RFS 5-7</p> <p>1 Test System or Tester VERIFIES THAT The system displays an apology message MEANWHILE Test System or Tester VERIFIES THAT the system ejects the ATM card</p> <p>2 ABORT.</p> <p>Postcondition (Test Oracle) ATM customer funds have not been withdrawn. The system is idle. The system is displaying a Welcome message.</p>
<p>Global Alt. Flow (Test Sequence) "alt3" ▼</p>	<p>ATM customer enters Cancel.</p> <p>1 <- internal transaction -></p> <p>2 <- internal transaction -></p> <p>3 ABORT.</p> <p>Postcondition (Test Oracle) ATM customer PIN number has not been withdrawn. The system is idle. The system is displaying a Welcome message.</p>

RTCM - Specifying TCS with API Information

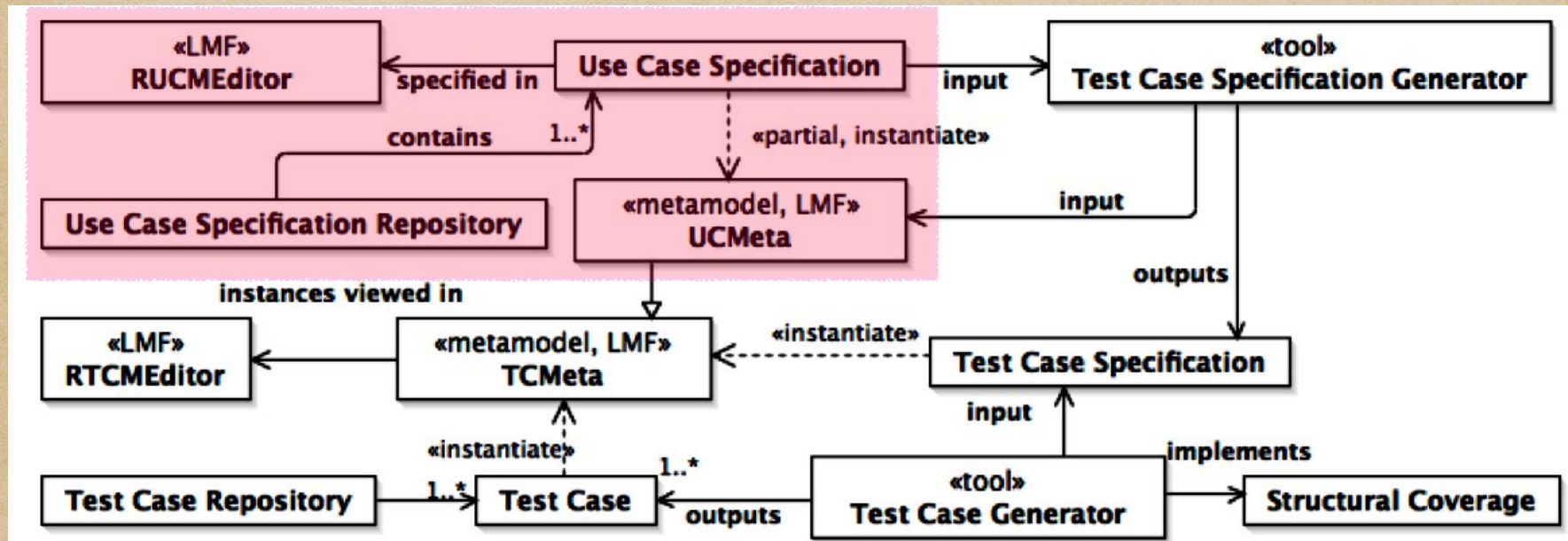
Test Case Specification

Name	CallBehavior	
Brief Description	The test specification aims to test the behavior of the system under test makes calls to other endpoint.	
Precondition (Test Data Specification)	The device A Configuration.NetworkServices.H323.Mode=On The device A Configuration.Conference.AutoAnswer.Mode=On The device B Configuration.NetworkServices.H323.Mode=On The device B Configuration.Conference.AutoAnswer.Mode=On	
Tester	None	
Dependency	None	
Test Setup		
▼	Name	CreateDevices
	Description	This test setup aims to create two devices in the test environment.
Basic Flow (Test Setup)		
(Untitled) ▼	Steps	
	1	The test system creates a device as A (IP=192.168.0.1,name=epa,username=user,password=password,rootpassword=rootpassword).
	2	The test system creates a device as B (IP=192.168.0.2,name=epb,username=user,password=password).
	Postcondition (Test Oracle)	The device A is created. The device B is created.

RTCM - Specifying TCS with API Information

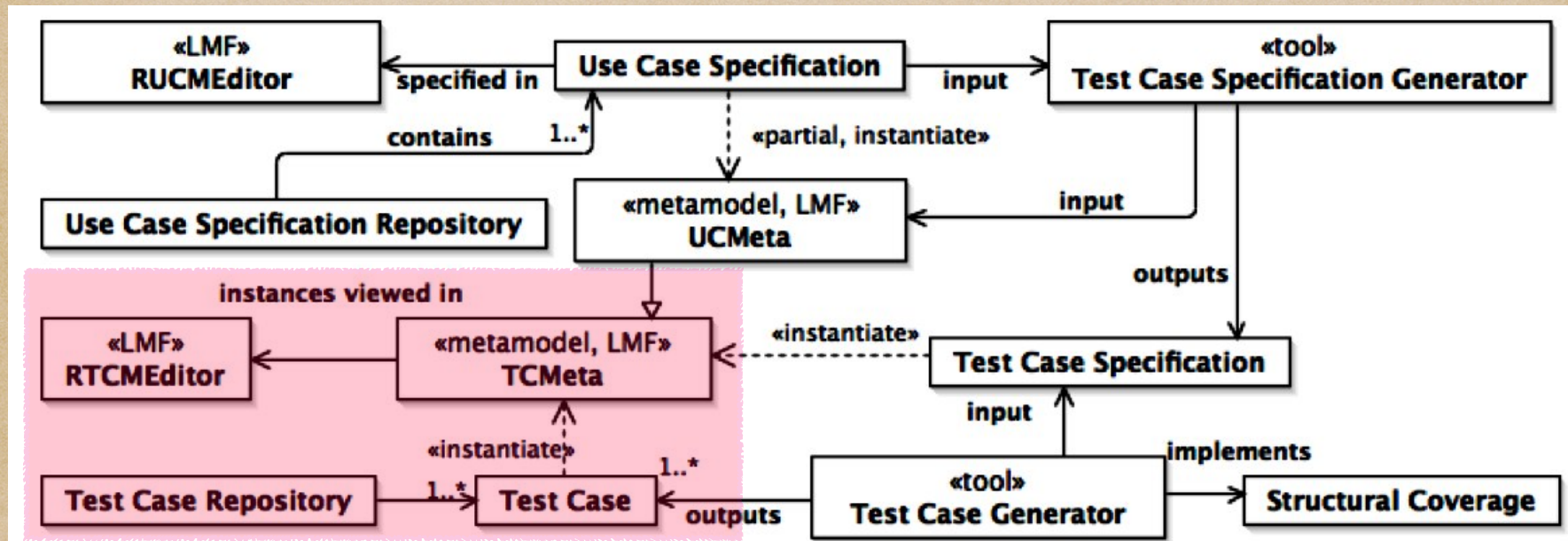
Basic Flow (Test Sequence)	Steps
(Untitled) ▼	1 The test system VALIDATES THAT the device A Status.Conference.Presentation.Mode==Off
	2 The test system VALIDATES THAT the device A Status.SystemUnit.State.NumberOfActiveCalls == 0
	3 DO
	4 The device A INVOKES API Command.Dial(the device B) to make a call.
	5 UNTIL the device A Status.SystemUnit.State.NumberOfActiveCalls > 0
	6 The device A INVOKES API Command.Presentation.Start() to start a presentation.
	7 The test system VALIDATES THAT the device A Status.Conference.Presentation.Mode == Sending
	8 The device A INVOKES API Command.Presentation.Stop() to stop the presentation.
	9 The test system VALIDATES THAT the device A Status.Conference.Presentation.Mode == Off
	10 The device A INVOKES API Command.Call.DisconnectAll() to disconnect all endpoints.
	Postcondition (Test Oracle)

a Toucan4 Test: Transformation from RUCM to RTCM

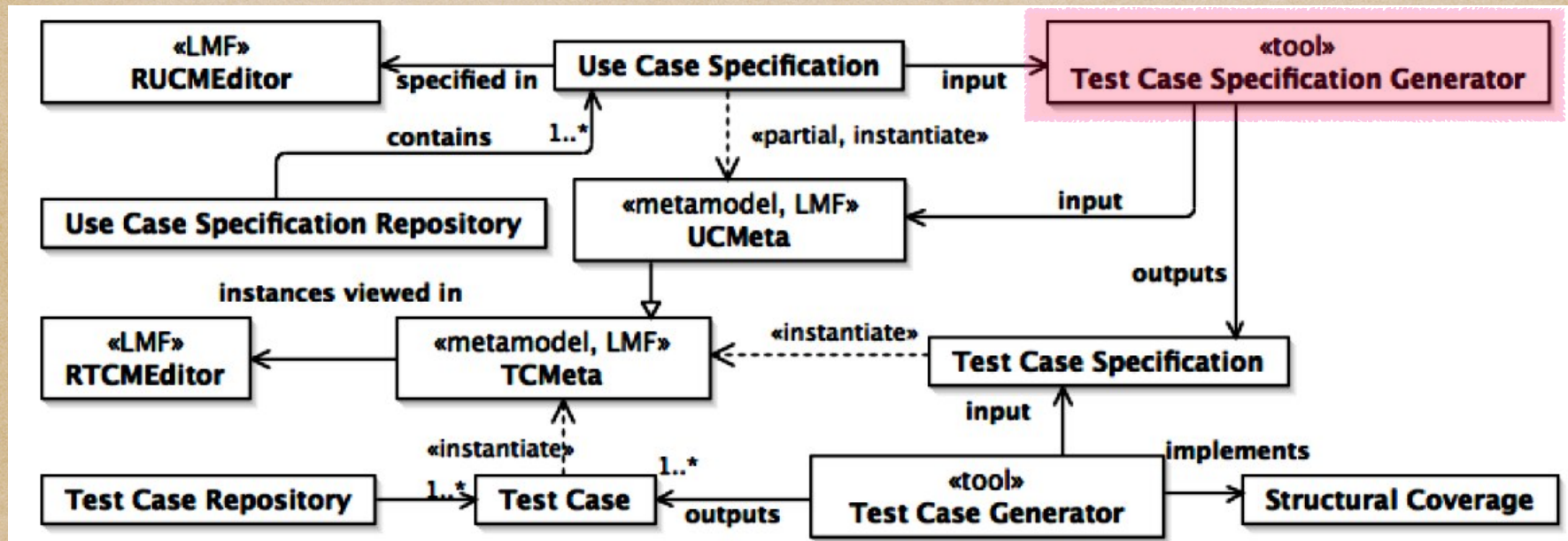


- ◆ LMF implements similar kinds of functionalities as EMF but with a lightweight design
- ◆ LMF aims to reduce tight coupling with Eclipse to facilitate easier transformations to other platforms.

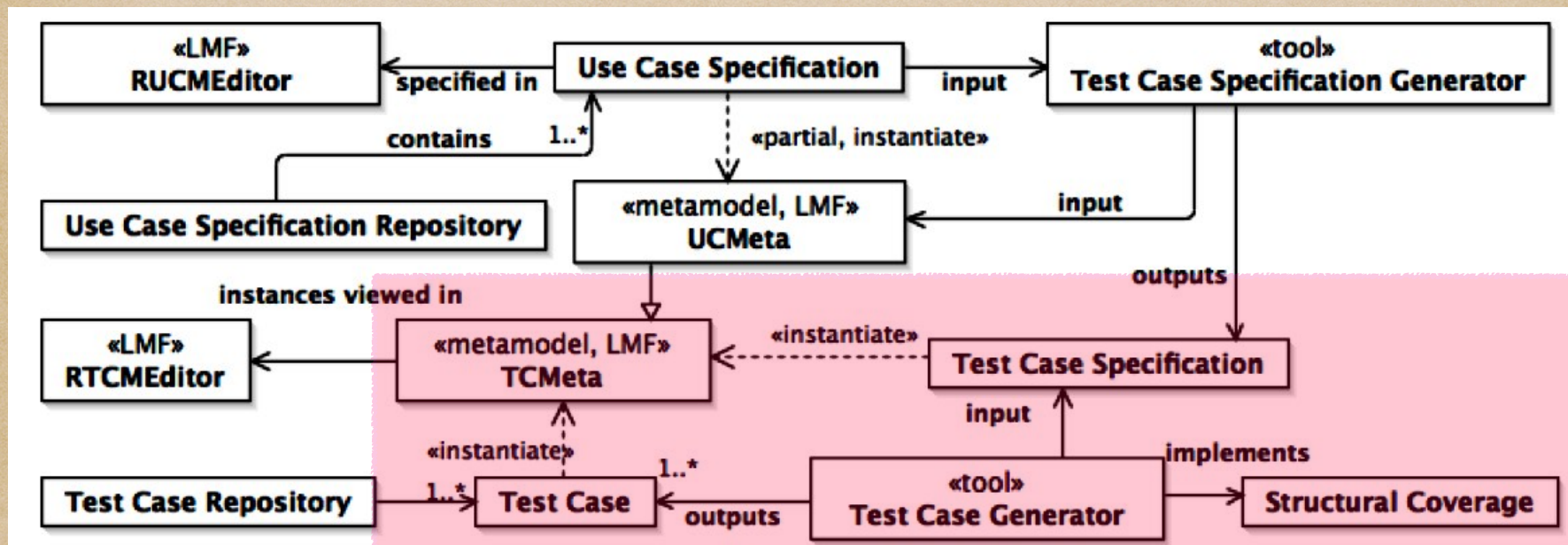
a Toucan4 Test: Transformation from RUCM to RTCM



a Toucan4 Test: Transformation from RUCM to RTCM



a Toucan4 Test: Transformation from RUCM to RTCM



- ◆ Structural coverage criteria
 - ◆ Branch coverage: All Branch and All Condition
 - ◆ Loop coverage: each loop (DO UNTIL) is exercised exactly one, none, and x number of times.
- ◆ Other coverage criteria
 - ◆ All Sentence and All FlowOfEvents

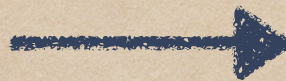
RUCM - Use Cases

- modelElements (10)
 - Withdraw Fund
 - Transfer Fund
 - Query Account
 - Validate PIN
 - ATM Customer
 - Card Reader
 - Relationship (ATM Customer - Withdraw Fund)
 - Relationship (ATM Customer - Transfer Fund)
 - Relationship (ATM Customer - Query Account)
 - Relationship (Card Reader - Validate PIN)



RTCM - Test Case Specifications

- modelElements (4)
 - Withdraw Fund
 - composedTestCaseSpecification (1)
 - composedTestSetups (1)
 - Transfer Fund
 - Query Account
 - Validate PIN
 - composedTestCaseSpecification (1)
 - composedTestSetups (1)



RTCM - Test Cases

- modelElements (4)
 - Withdraw Fund
 - composedTestCaseSpecification (66)
 - composedTestSetups (1)
 - Transfer Fund
 - composedTestCaseSpecification (55)
 - composedTestSetups (1)
 - Query Account
 - composedTestCaseSpecification (22)
 - composedTestSetups (1)
 - Validate PIN
 - composedTestCaseSpecification (11)
 - Test_Validate PIN_test0
 - Test_Validate PIN_test1
 - Test_Validate PIN_test2
 - Test_Validate PIN_test3
 - Test_Validate PIN_test4
 - Test_Validate PIN_test5
 - Test_Validate PIN_test6
 - Test_Validate PIN_test7
 - Test_Validate PIN_test8
 - Test_Validate PIN_test9
 - Test_Validate PIN_test10
 - composedTestSetups (1)

Evaluation

Case Studies	# UCSs	# Dependencies	# Flows	# Flow steps	# Condition sen.	# Validation sen.	# Resume
ATM	4	3	14	70	3	10	1
CMS	1	0	8	60	13	1	5
AS	11	5	34	212	11	16	3
SPS	14	6	28	237	28	15	2
Total	30	14	84	579	55	42	11

Case Studies	TCS (30 in total)			# Test Cases
	# Condition sentences	# Validation sentences	# Resume sentences	
ATM	3	10	1	154
CMS	7	1	5	60
AS	11	16	3	69
SPS	28	15	2	106
Total	49	42	11	389

Conclusion

- ◆ Systematic and automated
- ◆ Precise and easy to understand specifications
- ◆ Easier than modelling behavioural models for testing
- ◆ Reducing reliance on domain experts
- ◆ Traceability
- ◆ Easier test case maintenance
- ◆ Separation of concerns
- ◆ Conformance to existing standards

Questions?