3rd UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

Sophia Antipolis, French Riviera
20-22 October 2015

RTCM: A NATURAL LANGUAGE BASED, AUTOMATED AND PRACTICAL TEST CASE GENERATION FRAMEWORK

Tao Yue, PhD

Senior Research Scientist, Simula Research Laboratory, Oslo, Norway

tao@simula.no

http://zen-tools.com/people/taoy.html

Outline

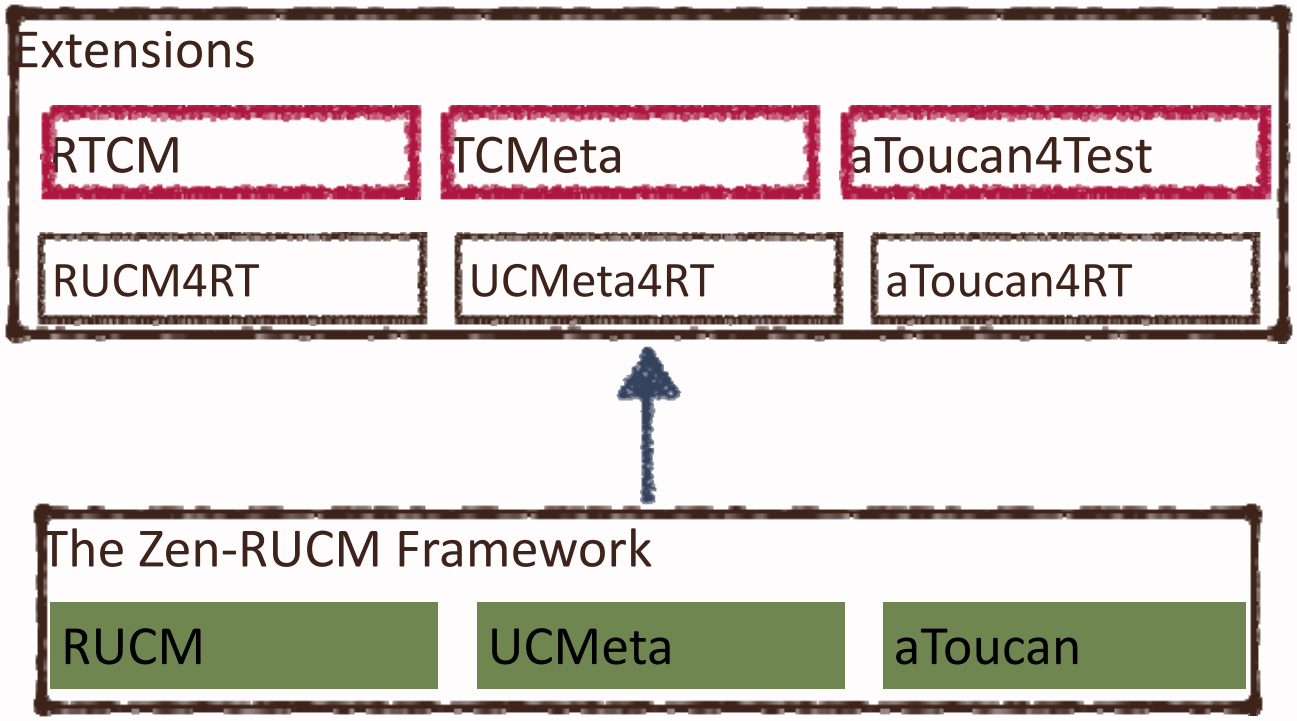
- Background
  - The Zen-RUCM Framework
  - Restricted Use Case Modeling (RUCM)
- RTCM: A Natural Language Based, Automated and Practical Test Case Generation Framework
  - RTCM Specification
  - Test Generator: aToucan4RTCM
  - Evaluation

Zen-RUCM

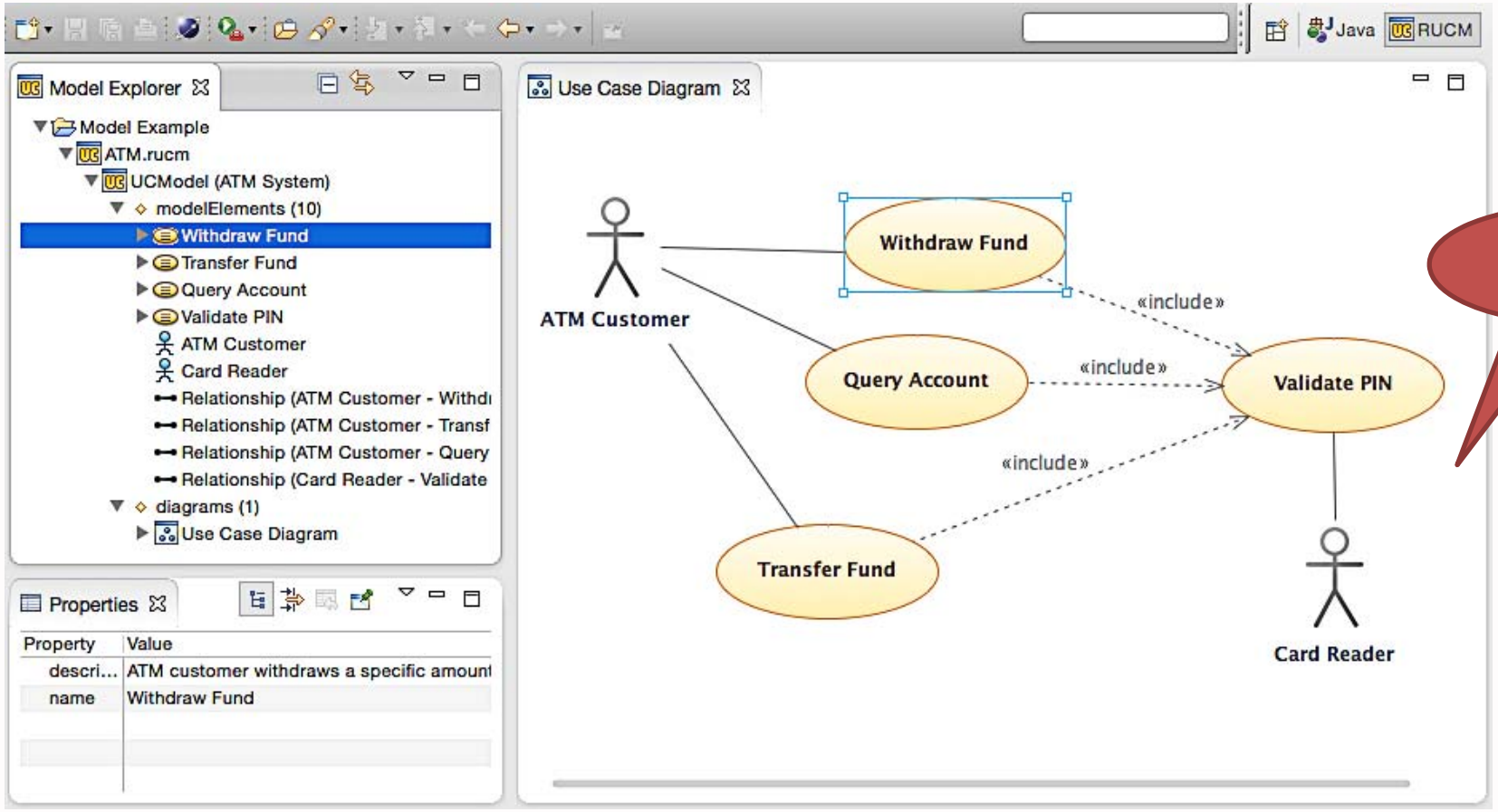
What is Zen-RUCM?

Zen-RUCM is a **restricted NL-**based, **model-based**, **scenario specification**, automated **model generation**, **verification** and **validation** platform, with **tool support** for software and system engineering.

Zen-RUCM - Architecture



Restricted Use Case Modeling (RUCM)



RUCM Use Case Template

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 (Untitled) ▼	Steps	
	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.

Typical Fields

Basic Flow

RUCM Use Case Template

Alternative Flows

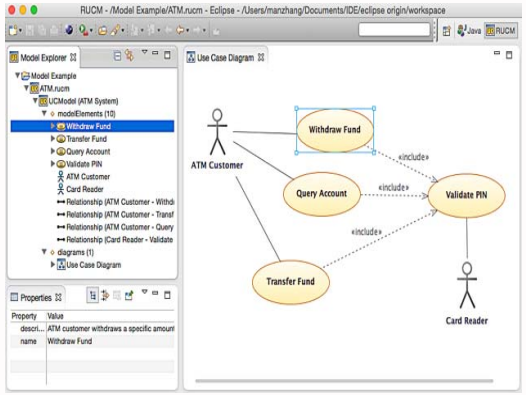
Specific Alternative Flow "alt1"▼	**RFS** 8	
	1	The system displays an apology message **MEANWHILE** the system ejects the ATM card.
	2	The system shuts down.
	3	**ABORT.**
	Postcondition	ATM customer funds have not been withdrawn. The system is shut down.
Bounded Alternative Flow "alt2"▼	**RFS** 5-7	
	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.

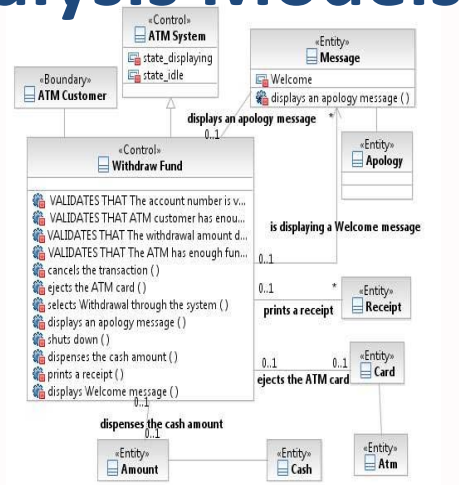
RUCM Use Case Template

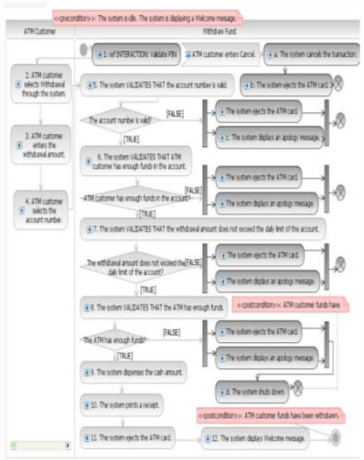
Basic Flow	Steps	
(Untitled) ▼	1	**INCLUDE USE CASE** Validate PIN.
	2	ATM customer selects Withdrawal through the system.
	3	ATM customer enters the withdrawal amount through the system.
	4	ATM customer selects the account number through the system.
	5	The system **VALIDATES THAT** the account number 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.

Specific Alternative Flow	RFS 8	
"alt1" ▼	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.

aToucan: Automated Transformation from Use Cases to UML Analysis Models

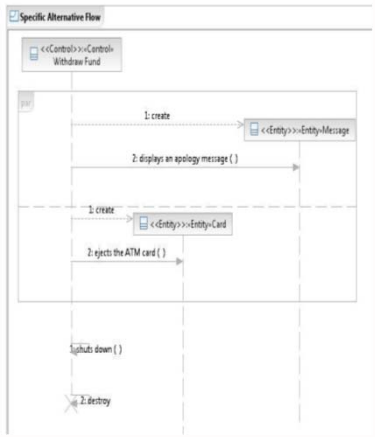


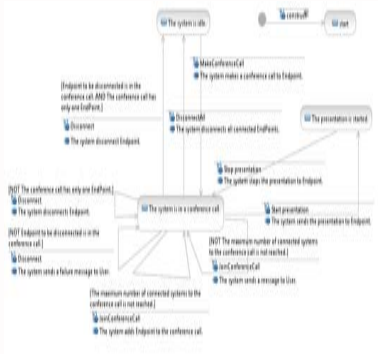




Use Case Specification	
Use Case Name	Withdraw Fund
Brief Description	ATM customer withdraws a specific amount of funds from a valid bank account.
Precondition	The system is idle. The system is displaying a Welcome message.
Primary Actor	ATM Customer
Secondary Actors	Card Reader
Dependency	**INCLUDE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

Basic Flow (Untitled) ▼	Steps
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.





Restricted Test Case Modeling (RTCM)

Tao Yue, Shaukat Ali, and Man Zhang. **Applying A Restricted Natural Language Based Test Case Generation Approach in An Industrial Context**, In International Symposium on Software Testing and Analysis (ISSTA), 2015.

Challenges

- **Specification/Modeling**
  - NL-based specifications are inherently ambiguous;
  - Supporting MBT often relies on more formal specifications using e.g., UML state machines.
- **Automation**
  - Deriving tests from NL-based specifications is often not systematic and automated;
  - Traceability between requirements and tests can be established in a cost-effective manner.

What is RTCM?

RTCM, is built on RUCM, as part of the Zen-RUCM framework, is **a lightweight MBT methodology** that automatically **generates test case specifications and executable test cases** with the aim to reduce modeling effort required for enabling MBT.

RTCM – Specifying Test Case Specification 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)	Steps	
(Untitled) ▼	1	**INCLUDE TC SPEC** Test_Validate PIN
	2	Test System or Tester selects Withdrawal through the system
	3	Test System or Tester enters the withdrawal amount through the system
	4	Test System or Tester 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 Editor

RTCM – Specifying Test Case Specification without API Information

Specific Alt. Flow (Test Sequence) "alt1" ▼	**RFS** 8	
	1	Test Sysetem or Tester **VERIFIES THAT** the system displays an apology message **MEANWHILE** Test System or Tester **VERIFIES THAT** the system ejects the ATM card.
	2	Test System or Tester **VERIFIES THAT** the system shuts down.
	3	**ABORT.**
	Postcondition (Test Oracle)	ATM customer funds have not been withdrawn. The system[sut] is shut down.

Bounded Alt. Flow (Test Sequence) "alt2" ▼	**RFS** 5–7	
	1	Test Sysetem or Tester **VERIFIES THAT** the system displays an apology message **MEANWHILE** Test System or Tester **VERIFIES THAT** the system ejects the ATM card.
	2	**ABORT.**
	Postcondition (Test Oracle)	ATM customer funds have not been withdrawn. The system[sut] is idle. The system[sut] is displaying a Welcome message.

Global Alt. Flow (Test Sequence) "alt3" ▼	Test System or Tester enters Cancel.	
	1	<- Internal transaction ->
	2	<- Internal transaction ->
	3	**ABORT.**
	Postcondition (Test Oracle)	ATM customer PIN number has not been withdrawn. The system[sut] is idle. The system[sut] is displaying a Welcome message.

RTCM – Specifying Test Case Specification with API Information

RTCM Editor

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	Create Device
	Description	This test setup aims to create an device to start a call behavior.

Basic Flow (Test Setup) (Untitled) ▼	Steps	
	1	The test system creates a device as A (IP=192.168.0.1,name=epa,username=admin,password=password, rootpassword=password).
	2	The test system creates a device as B (IP=192.168.0.2,name=epa,username=admin,password=password, rootpassword=password).
	Postcondition (Test Oracle)	The device A has been created.The device B has been created.

RTCM – Specifying Test Case Specification with API Information

Status.Conference.Presentation.Mode == Off

Basic Flow (Test Sequence) (Untitled) ▼	Steps	
	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 a 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)	The device A Status.Conference.Presentation.Mode == Off The device A Status.SystemUnit.State.NumberOfActiveCalls == 0

INVOKES API Command.Dial(the Device B)

Test Generator: aToucan4Test

aToucan4Test: Transformation from RTCM to Test Cases - Example

RTCM Model Explorer

- CallBehavior
  - basicFlow
    - BasicFlow4TestSequence
      - steps (10)
        - "The test system VALIDATES THAT the device A
        - "The test system VALIDATES THAT the device A
        - "DO"
        - "The device A INVOKES API Command.Dial(the
          - natures (1)
            - [InvokeAPINature]
        - "UNTIL the device A Status.SystemUnit.State.N
        - "The device A INVOKES API Command.Presen
        - "The test system VALIDATES THAT the device A
        - "The device A INVOKES API Command.Presen
        - "The test system VALIDATES THAT the device A
        - "The device A INVOKES API Command.Call.Dis
      - postCondition
    - preCondition
    - briefDescription
    - extendSpecs (0)
    - includeSpecs (0)
    - testSetupLink
    - alternativeFlows (6)
      - SpecificFlow4TestSequence (TS1)
      - SpecificFlow4TestSequence (TS2)
      - OracleVerificationFlow (O1)
        - rfsSentence
          - "RFS 4"
        - steps (1)
          - "The test system VALIDATES THAT the device A
            - natures (1)
              - [ValidateThatNature]
          - postCondition (null)
      - OracleVerificationFlow (O2)

CallBehavior

Basic Flow (Test Sequence) (Untitled)	Steps	
	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 a 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)	The device A Status.Conference.Presentation.Mode == Off The device A Status.SystemUnit.State.NumberOfActiveCalls == 0

Specific Alt. Flow (Test Sequence) "TS1"	RFS 1	
	1	The device A INVOKES API Command.Presentation.Stop() to stop the active presentation.
	2	RESUME STEP 2
	Postcondition (Test Oracle)	The device A Status.Conference.Presentation.Mode==Off

Properties

Property	Value
api	Command.Dial(the device B) to make a call
subject	The device A

aToucan4Test: Transformation from RTCM to Test Cases - Example


```
def test_case_26(self):
    from utility import Translate
    from lxml.etree import tostring

    self.assertTrue(self.A.cuil.status.Conference.Presentation.Mode.text == "Off")
    self.assertFalse(int(self.A.cuil.status.SystemUnit.State.NumberOfActiveCalls.text) == 0)
    old_status = None
    old_status = copy.copy(self.A.cuil.get_xml("/Status"))
    Translate.TranslateA("Command.Call.DisconnectAll()",self.A.cuil).translate()
    # Resume 3
    self.assertTrue(int(self.A.cuil.status.SystemUnit.State.NumberOfActiveCalls.text) == 0)
    while True:
        old_status = None
        old_status = copy.copy(self.A.cuil.get_xml("/Status"))
        self.A.dial(self.B)
        #RFS of Oracle verification flow
        self.assertTrue(int(self.A.cuil.status.SystemUnit.State.NumberOfActiveCalls.text)
                        == int((old_status[0].xpath("SystemUnit/State/NumberOfActiveCalls"))[0].text) + 1)
        if (int(self.A.cuil.status.SystemUnit.State.NumberOfActiveCalls.text) > 0): break
    old_status = None
    old_status = copy.copy(self.A.cuil.get_xml("/Status"))
    Translate.TranslateA("Command.Presentation.Start()",self.A.cuil).translate()
    self.assertTrue(self.A.cuil.status.Conference.Presentation.Mode.text == "Sending")
    old_status = None
    old_status = copy.copy(self.A.cuil.get_xml("/Status"))
    Translate.TranslateA("Command.Presentation.Stop()",self.A.cuil).translate()
    self.assertTrue(self.A.cuil.status.Conference.Presentation.Mode.text == "Off")
    old_status = None
    old_status = copy.copy(self.A.cuil.get_xml("/Status"))
    Translate.TranslateA("Command.Call.DisconnectAll()",self.A.cuil).translate()
    self.assertTrue(self.A.cuil.status.Conference.Presentation.Mode.text == "Off")
    self.assertTrue(int(self.A.cuil.status.SystemUnit.State.NumberOfActiveCalls.text) == 0)
```

RTCM and aToucan4Test - Summary

RUCM - Use Case Specifications

- UCModel (ATM System)
  - modelElements (10)
    - Withdraw Fund
      - specification
        - Specification
          - basicFlow
          - preCondition
          - briefDescription
          - alternativeFlows (3)
            - SpecificAlternative (alt1)
            - BoundedAlternative (alt2)
            - GlobalAlternative (alt3)
          - branchSpecs (0)
        - extensionPoint (0)
        - extend (0)
        - include (1)
    - 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

- TCS_ATM System
  - modelElements (5)
    - Default Test Setup
    - Test_Withdraw Fund
    - Test_Transfer Fund
    - Test_Query Account
    - Test_Validate PIN
      - basicFlow
      - preCondition
      - briefDescription
      - extendSpecs (0)
      - includeSpecs (0)
      - testSetupLink
      - alternativeFlows (4)
        - SpecificFlow4TestSequence (alt1)
        - BoundedFlow4TestSequence (alt2)
        - SpecificFlow4TestSequence (alt3)
        - GlobalFlow4TestSequence (alt4)

RTCM - Test Cases

- ATCs_ATM System
  - modelElements (5)
    - Default Test Setup
    - Test_Withdraw Fund
      - modelElements (60)
    - Test_Transfer Fund
      - modelElements (50)
    - Test_Query Account
      - modelElements (20)
    - Test_Validate PIN
      - modelElements (10)
        - Test_Validate PIN_1
        - Test_Validate PIN_2
        - Test_Validate PIN_3
        - Test_Validate PIN_4
        - Test_Validate PIN_5
        - Test_Validate PIN_6
        - Test_Validate PIN_7
        - Test_Validate PIN_8
        - Test_Validate PIN_9
        - Test_Validate PIN_10

Evaluation

Case Studies

Case Study		Subsystem	# TCS	# Flow of events	# Oracle Verification Flow	# Status Variable	# API Invocation
Manually Derived	VCS	Call	3	9	3	6	9
		Presentation	2	3	0	2	3
		Camera	4	7	1	5	18
		Total	**9**	**19**	**4**	**13**	**30**
	SOGPS	Subsea	8	3	6	N/A	N/A
		Surface	1	2	3	N/A	N/A
		Total	**9**	**5**	**9**	**N/A**	**N/A**
Automatically Generated	Autopilot	N/A	11	34	N/A		
	CMS		1	8			
	ATM		4	14			
	SPS		14	28			
Grand Total			**48**	**108**	**13**	**13**	**30**

Effort for Applying RTCM and aToucan4Test

- Average time (in minutes) for deriving an automatically executable test case:
  - Time required to specify TCSs/Total number of test cases generated= 120 mins/246≈0.5 min
- Comparison with an existing MBT approach [1]
  - UML State machines, OCL constraints
  - Cost per test case is roughly 6 minutes
- RTCM takes on average takes 0.5 minutes, whereas MBT approach takes roughly 6 minutes

[1] Ali, S. and Hemmati, H., 2014. Model-based Testing of Video Conferencing Systems: Challenges, Lessons Learnt, and Results. In *Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on* IEEE, 353-362.

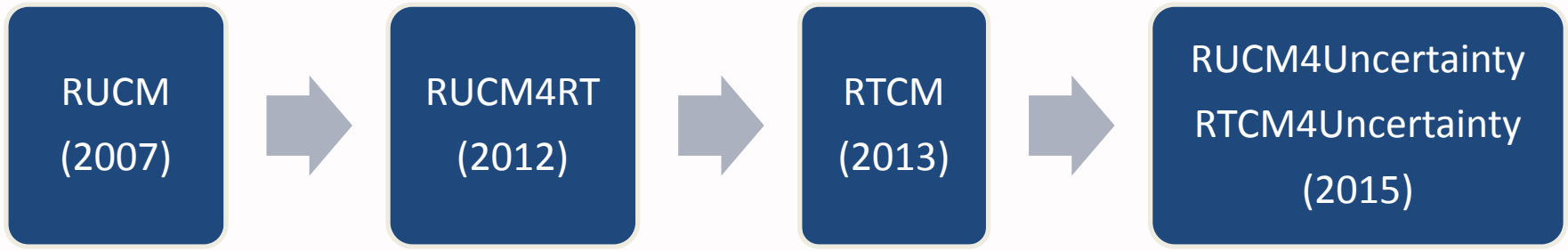
Experience and Lessons Learnt

- Highlights
  - Easy to learn
  - Lightweight
  - “Free” automation
  - Easy to maintain test cases
- Drawbacks
  - Limited application scopes
  - No support on dictionary/glossary of domain concepts

Experience and Lessons Learnt

- Recommend Practice
  - Carefully select contexts to apply
  - Tailored methodologies and tool supports are needed in certain contexts
  - Tool support is the key to ease the adoption of such a methodology
  - Integrate with test management solutions used in a particular company

Roadmap of Zen-RUCM



References – Zen-RUCM

- Tao Yue, Lionel Briand, and Yvan Labiche. **aToucan: An Automated Framework to Derive UML Analysis Models from Use Case Models**, in ACM Transactions on Software Engineering and Methodology (TOSEM), 24, no. 3, 2015.
- Tao Yue, Lionel Briand, and Yvan Labiche. **Facilitating the Transition from Use Case Models to Analysis Models: Approach and Experiments**, Transactions on Software Engineering and Methodology (TOSEM) 22(1), 2013.
- Tao Yue, Shaukat Ali, and Man Zhang. **Applying A Restricted Natural Language Based Test Case Generation Approach in An Industrial Context**, In International Symposium on Software Testing and Analysis (ISSTA), 2015.
- Man Zhang, Tao Yue, Shaukat Ali, Huihui Zhang, and Ji Wu. **A Systematic Approach to Automatically Derive Test Cases from Use Cases Specified in Restricted Natural Languages**, In: 8th System Analysis and Modelling Conference (SAM'14).
- Gong Zhang, Tao Yue, Ji Wu, and Shaukat Ali. **Zen-RUCM: A Tool for Supporting a Comprehensive and Extensible Use Case Modeling Framework**, In: ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems (MODELS), Tools Demonstration, 2013.
- Tao Yue and Shaukat Ali, **Bridging the Gap between Requirements and Aspect State Machines to Support Non-Functional Testing: Industrial Case Studies**, in: 8th European Conference on Modelling Foundations and Applications (ECMFA) 2012

References – Zen-RUCM

- Tao Yue, Shaukat Ali, and Lionel Briand, **Automated Transition from Use Cases to UML State Machines to Support State-based Testing**, In: 7th European Conference on Modelling Foundations and Applications (ECMFA) 2011
- Tao Yue, Lionel Briand, and Yvan Labiche, **A Use Case Modeling Approach to Facilitate the Transition Towards Analysis Models: Concepts and Empirical Evaluation**, In: ACM/IEEE International Conference on Model Driven Engineering Languages and Systems (MODELS), 2009
- Tao Yue, Lionel Briand, and Yvan Labiche. **An Automated Approach to Transform Use Cases into Activity Diagrams**, In: 6th European Conference, on Modelling Foundations and Applications(ECMFA) 2010.
- Man Zhang, Bran Selic, Shaukat Ali, and Tao Yue. **An Uncertainty Taxonomy to Support Model-Based Uncertainty Testing of Cyber-Physical Systems**. Simula Research Laboratory, 2015.

http://www.zen-tools.com

Thank you! Questions?