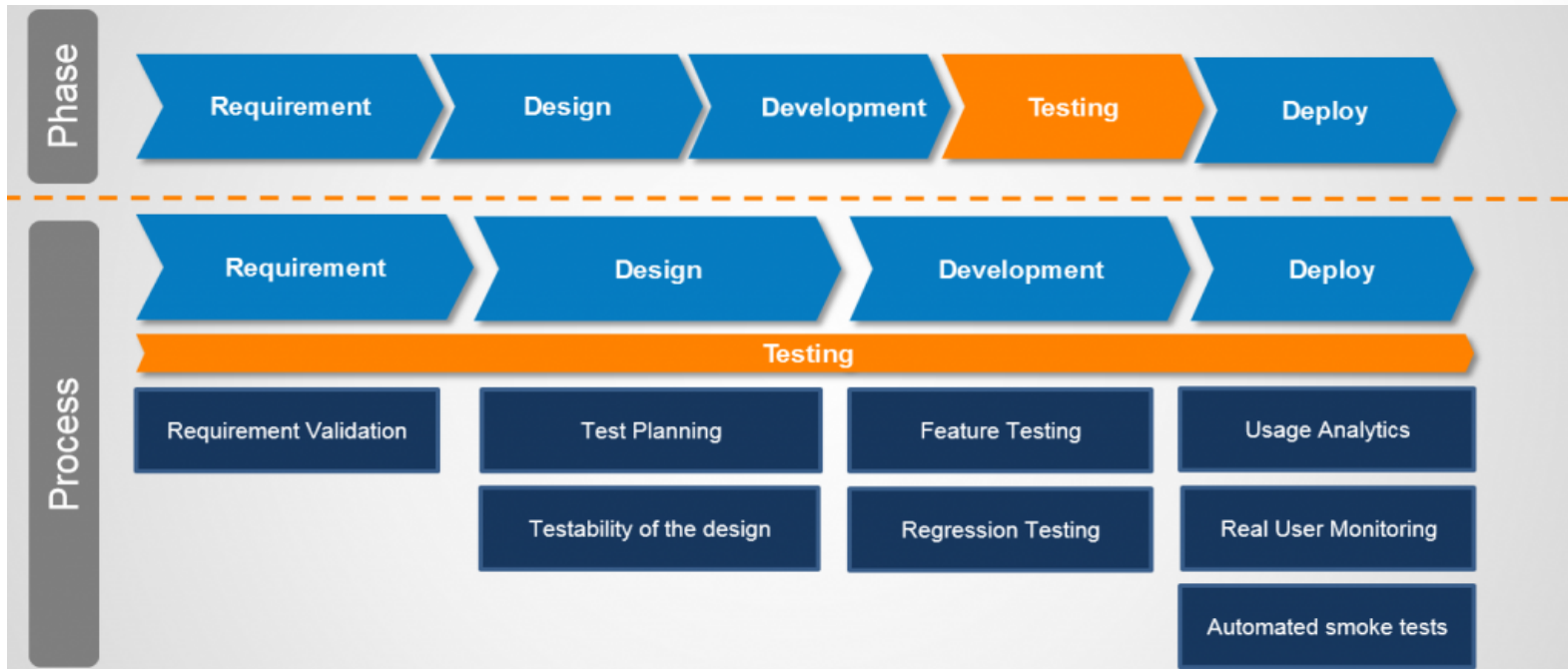


Testing in Research and Industry

ME Work (University project + Industry Internship)

Pooja Rani

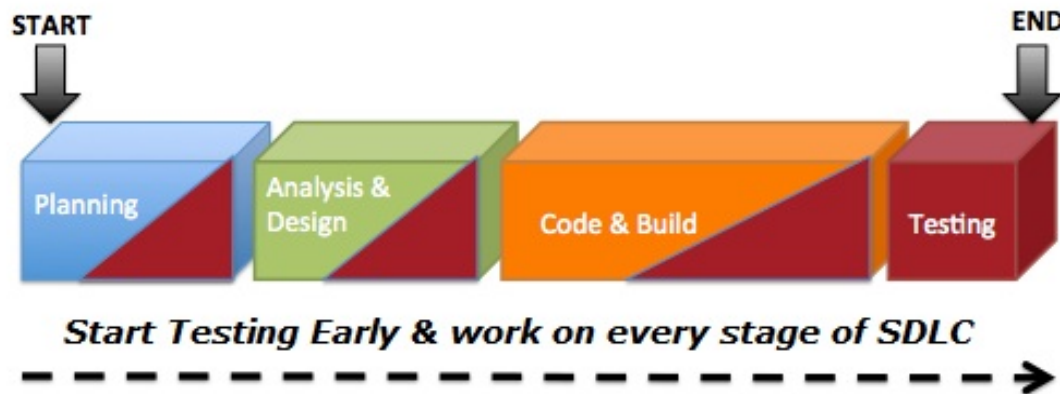
Testing is a process



- Time , Efforts
- Cost of bugs during Implementation phase
- Automating the testing process

Objective

- Analysis of software – From **testing** point of view



- Choose the **level** of analysis- class, method, module, package
- Perform analysis and prepare test reports

Approach

- Software failure mode and effect analysis(SFMEA)
- SFMEA used to discover design defects
- Level of analysis – Method
- Modeling language – UML 2.0

SFMEA

- SFMEA present ways to analysis item. An item can be class, method, module or package.

<i>Item</i>	<i>Failure mode</i>	<i>Causes</i>	<i>Effects</i>	<i>Probability of occurrence</i>
Item to be analysed for failure analysis	The failure modes of the selected item	Cause of the Failure	Effects that can be caused because of the failure	What is the probability of occurrence of this failure

- Considered Method as an item to analysis

Method-level Analysis

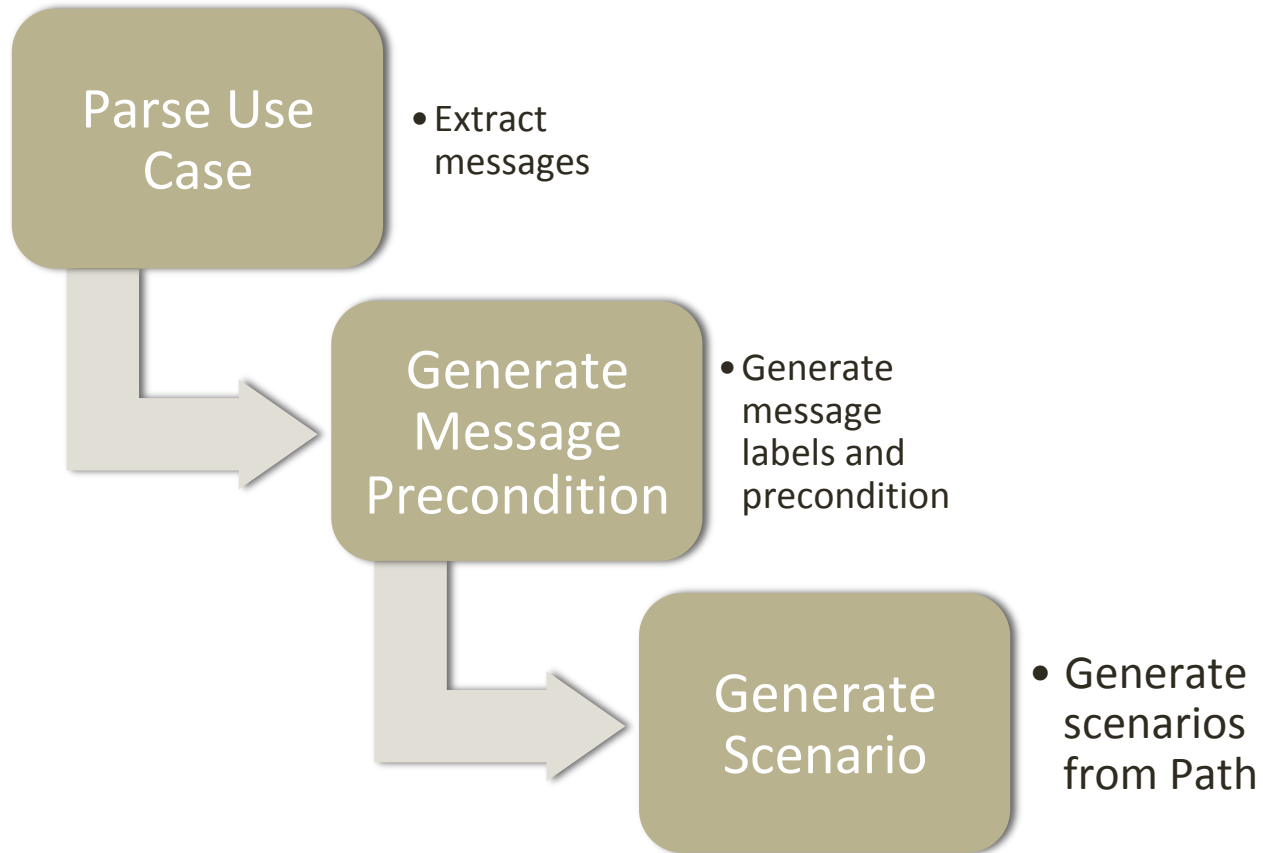
- Precondition Violation failure Modes of method
-
- Parametric Failure Mode of method
- Method Invocation order
- Post condition failure mode

SFMEA using Use Case Diagram

Requirement Analysis phase and Design phase

-University Project

Methodology



Input Use Case

Software

File Edit

Software Failure Modes & Effect Analysis For Use-Cases

Home Step 1 Step 2 Step 3

Use Case Name

Actor

Precondition

Action Details

```
requestWithdraw
InsertATM
InputPIN
IF Condition1 THEN
InputBalance
WHILE Condition2
selectOption
performTransaction
getStatus
ENDWHILE
checkStatus
ELSE
changePIN
IF Condition3 THEN
InputNewPIN
ELSE
selectOtherPIN
ENDIF
CASE
```

Save

Check Format

Proceed >>

Home tab where various information is provided

Generate Message Precondition

Home

Step 1

Step 2

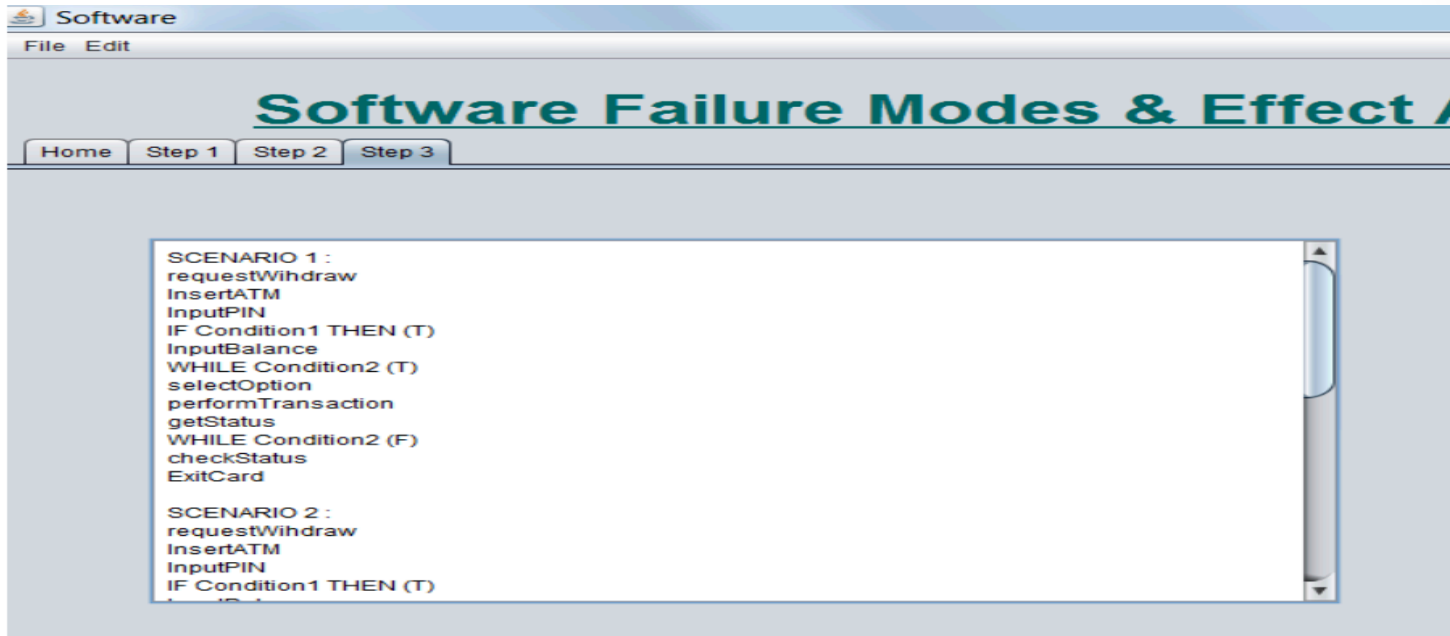
Step 3

Message Precondition Table

Action No	Action Label	Action Label Precondition
A1	requestWithdraw	
A2	InsertATM	A1,
A3	InputPIN	A1, A2,
A4	IF Condition1 THEN	A1, A2,

Generate Scenario

Scenario Generation



- Once we have scenario generate we can treat each message as item to be analyzed using SFMEA
- Perform different method-based analysis on each message

Limitation of use-case approach

- Assumption about input format
- Capture static information
- No sequence information about messages flow
- No information about message coverage

Automatic Test Case Generation using Sequence Diagram

Design phase and partially Development phase

-University Project

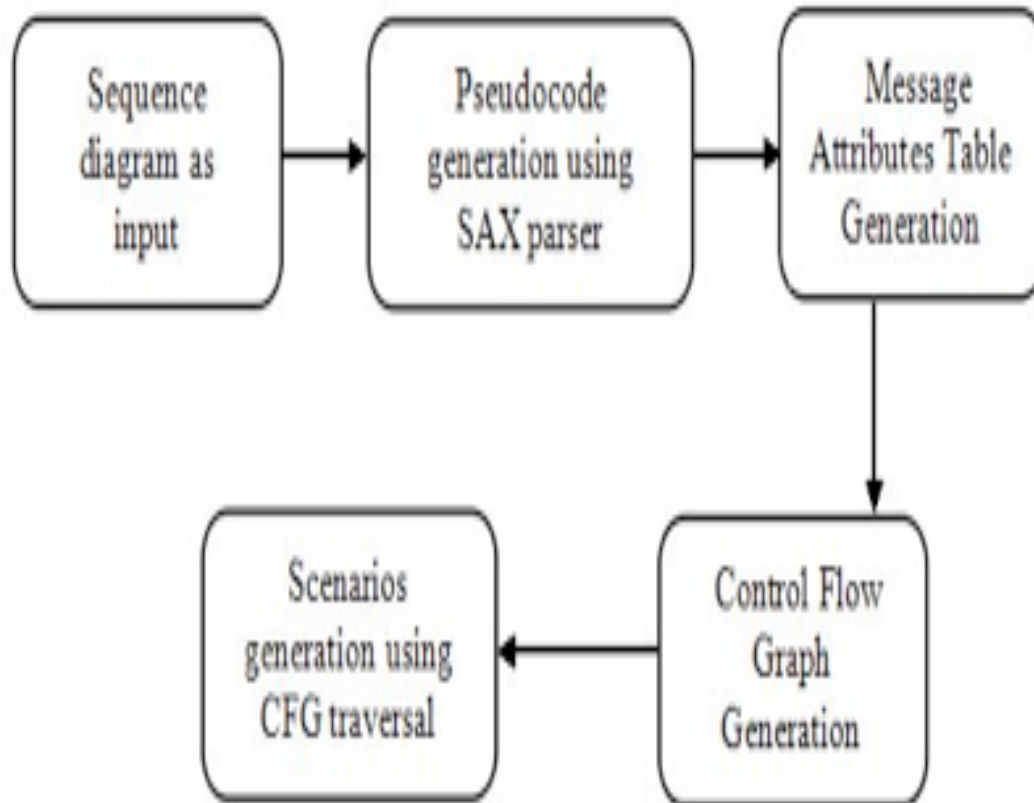
Why sequence diagram?

- Dynamic view of the system
- Top to bottom time sequence
- Captures object interaction
- Based on Model-based testing
- Test coverage

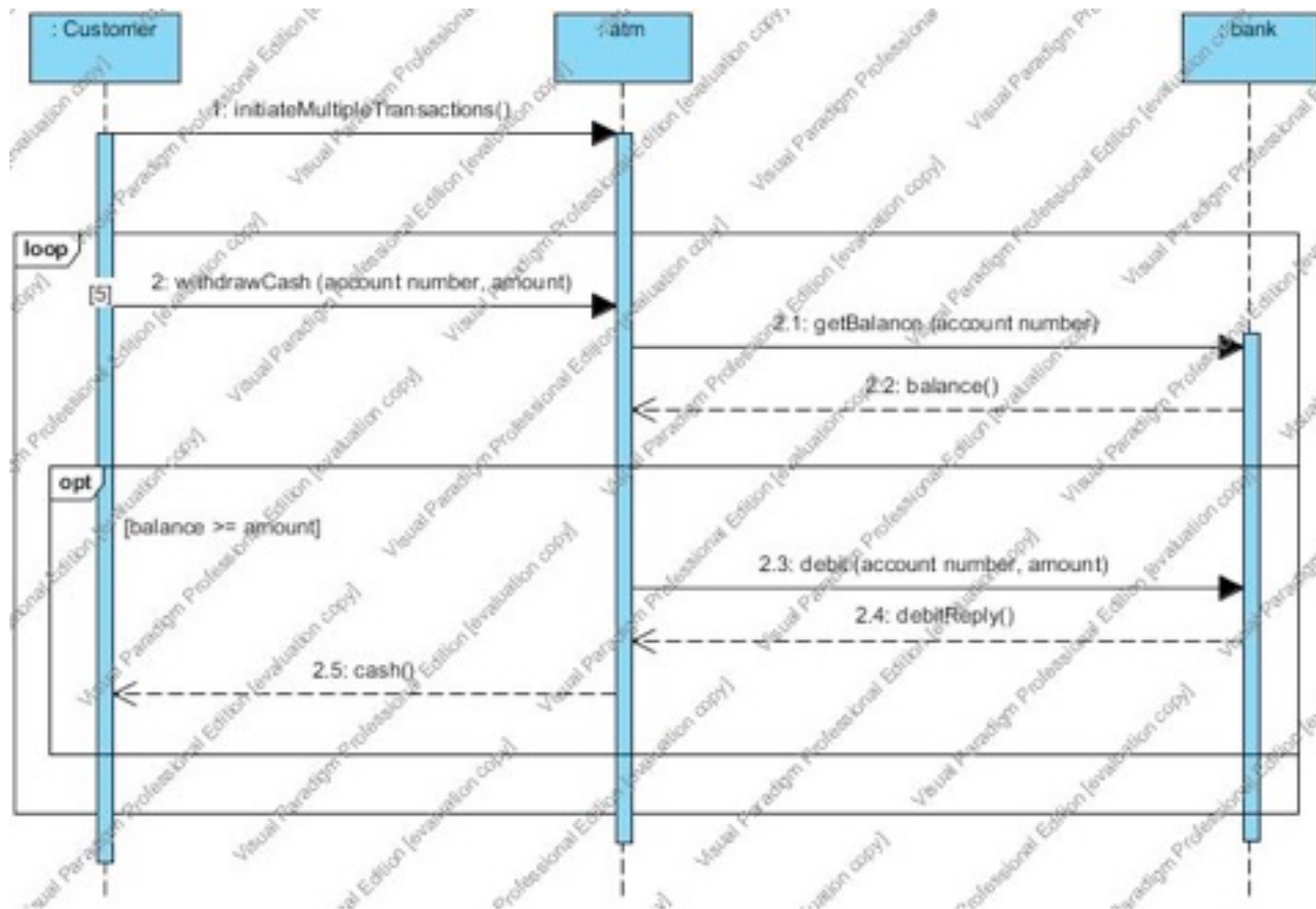
ATCG using Sequence Diagram

- **Methodology Followed**
- **Sequence Diagram as input**
- **Pseudo code generation using SAX Parser**
- **Message Attributes Table generation**
- **Control Flow Graph generation**
- **Scenarios generation using CFG traversal**
- **Tools and Technologies**

Methodology



Sequence Diagram as Input



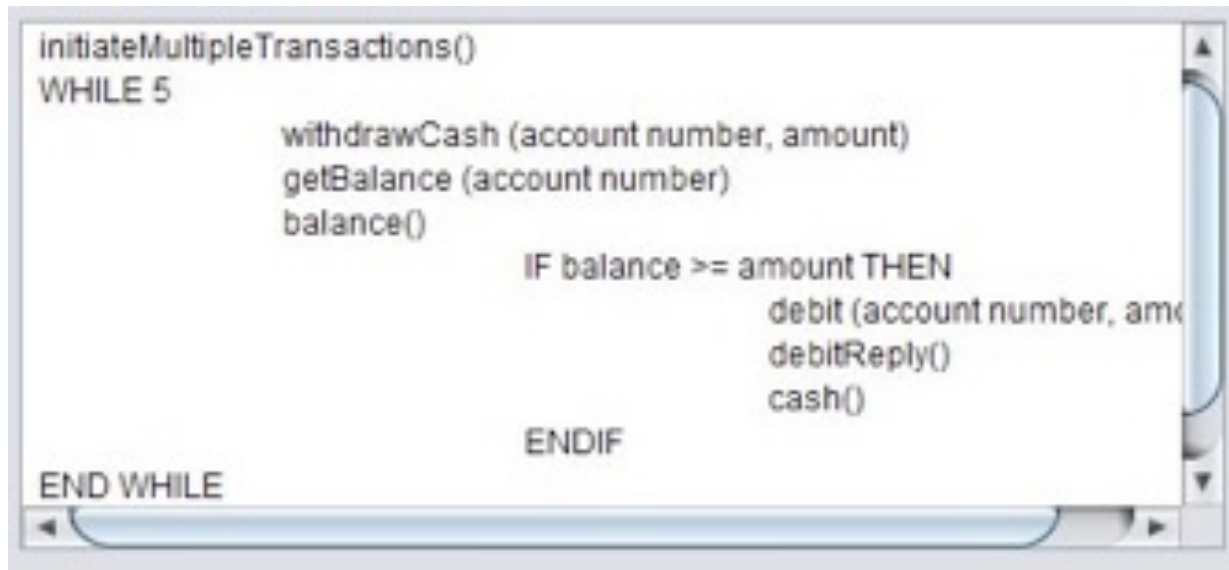
Sample XML

- Parse XML, extract tags

```
<ownedMember xmi:type="uml:Collaboration">
  <xmi:Extension extender="Visual Paradigm">
    <dummyModel value="true"/>
  </xmi:Extension>
  <ownedBehavior name="ATM Sequence Diagram" xmi:id="Fq2hoQqFTYQyWQOL" xmi:type="uml:Interaction">
    <lifeline name="" represents="FK0hoQqFTYQyWQOL_represents" xmi:id="FK0hoQqFTYQyWQOL" xmi:type="uml:Lifeline">
      <xmi:Extension extender="Visual Paradigm">
        <active xmi:value="false"/>
        <stopped xmi:value="false"/>
        <multiObject xmi:value="false"/>
        <activation name="Activation" type="false" xmi:id=".1cP0wqFTYgCXQJf" xmi:type="activation">
          <xmi:Extension extender="Visual Paradigm">
            <qualityScore value="-1"/>
          </xmi:Extension>
        </activation>
      </lifeline>
      <qualityScore value="-1"/>
    </xmi:Extension>
    <coveredBy xmi:ldref="#1cP0wqFTYgCXQJf"/>
  </lifeline>
  <message name="initiateMultipleTransactions()" receiveEvent="#1cP0wqFTYgCXQJf" sendEvent="#1cP0wqFTYgCXQJf" xmi:id="#1cP0wqFTYgCXQJf" xmi:type="ua">
    <signature xmi:id="#1cP0wqFTYgCXQJf_type" xmi:type="uml:Action"/>
    <xmi:Extension extender="Visual Paradigm">
      <number xmi:value="1"/>
      <asynchronous xmi:value="false"/>
      <fromActivation>
        <activation xmi:value=".1cP0wqFTYgCXQJf"/>
      </fromActivation>
      <toActivation>
        <activation xmi:value=".1cP0wqFTYgCXQJf"/>
      </toActivation>
      <qualityScore value="-1"/>
    </xmi:Extension>
  </message>
  <message name="withdrawCash (account number, amount)" receiveEvent="#1cP0wqFTYgCXQJf" sendEvent="#1cP0wqFTYgCXQJf" xmi:id="#1cP0wqFTYgCXQJf" xmi:type="ua">
    <signature xmi:id="#1cP0wqFTYgCXQJf_type" xmi:type="uml:Action"/>
  </message>
</ownedMember>
```

Pseudo code generation

- Based on various tags, generate pseudo-code



```
initiateMultipleTransactions()  
WHILE 5  
    withdrawCash (account number, amount)  
    getBalance (account number)  
    balance()  
        IF balance >= amount THEN  
            debit (account number, am  
            debitReply()  
            cash()  
        ENDIF  
    END WHILE
```

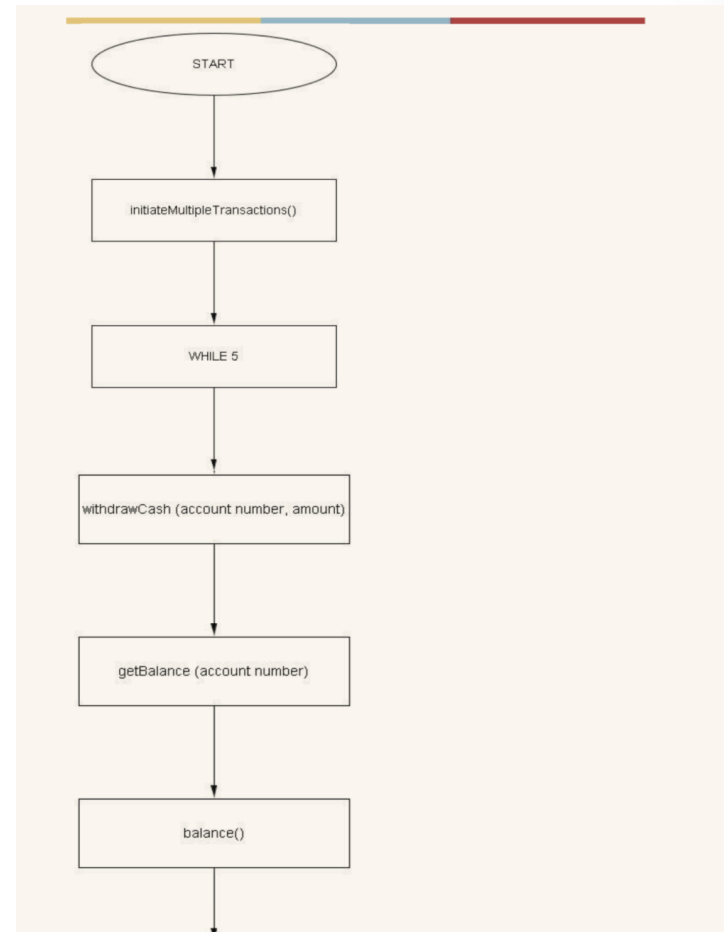
Message attributes table

- Extract Message attributes from SD
- Like Message Sequence No, Name, Type, Tag etc.

Message Sequence No	Message Name	Message Type	isReply	Message Tag	From	To	Dependent Message List	Independent Message List	Send Event	Receive Event
M1	cashCheque	Synchronous	False	1	Customer:customer	Bank	1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9		E11	E12
M2	getAmount	Synchronous	False	1.1	Bank	Cheque		1.3, 1.5, 1.6, 1.7, 1.8	E21	E22
M3	amount	Synchronous	True	1.2	Cheque	Bank			E31	E32
M4	getBalance	Synchronous	False	1.3	Bank	Account		1.5, 1.6, 1.7, 1.8	E41	E42
M5	balance	Synchronous	True	1.4	Account	Bank			E51	E52
M6	addInsufficientFundFee	Synchronous	False	1.5	Bank	Account		1.6, 1.7, 1.8	E61	E62
M7	noteReturnedCheque	Synchronous	False	1.6	Bank	Account		1.7, 1.8	E71	E72
M8	addDebitTransaction	Synchronous	False	1.7	Bank	Account		1.8	E81	E82
M9	storePhotoOfCheque	Synchronous	False	1.8	Bank	Account			E91	E92
M10	cash	Synchronous	True	1.9	Bank	Customer:customer			E101	E102

Control Flow Graph (CFG) generation

- Create control graph based on pseudo code tags
- Traverse Graph using DFS
- Generate scenarios



Tools & Technologies used

- Visual Paradigm for Sequence diagram
- SAX parser for parsing XML file
- Grappa library for CFG
- Netbeans as IDE
- Java

Future Work

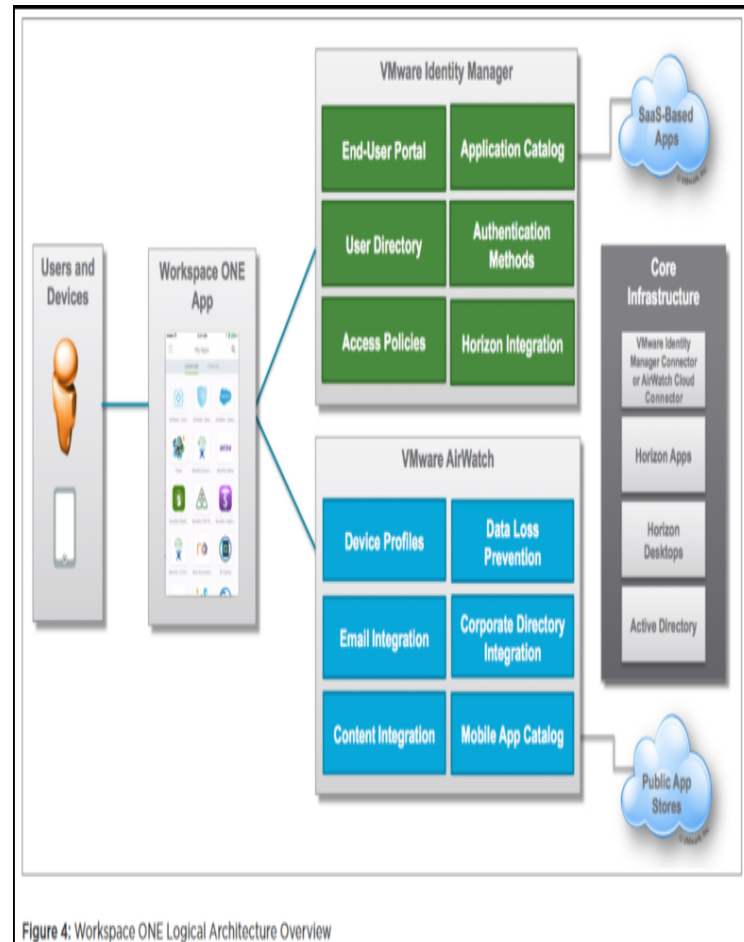
- Identifying valid and invalid scenario
- Elimination of redundant test case
- Generate Java test case automatically
- Execute test case automatically and generate test report
- Successfully generated java code for Finite State Machine(ModelJUnit)

Testing in Industry

VMware Internship – Testing in Development phase

Workspace ONE - Product

- Digital Workspace for enterprise application
- Giving IT a central place to manage user provisioning and access policy for applications.



VMware Identity Manager (VIDM)

- Provide identity and access management to users for Workspace ONE
- Secured access to corporate applications
- Across all devices and platforms
- Single portal access for employee work applications
- Conditional access control to apps based on device, network, and or user.
- Different Authentication Methods like RSA, Radius, certificates.

Challenges in testing VIDM

- Why Automation testing is difficult?
- Time to setup cluster environment- Short release cycles
- Identifying common scenario e.g Login test – User belonging to one domain should not be allowed to login with another domain and within one domain, group of users are there.
- Basic login test complexity
- Test coverage in Web and Mobile testing
- Lack of single testing environment
- Tools used – Selenium Grid, TestNG, Spring Framework

My Task

- Developing test libraries and extending test framework
- Identifying common scenario to automate
- Automate VIDM test cases like login test case
- Prepare regression test suite
- Integrate test cases in Continuous integration (CI) pipeline

Future Work

- Reducing test suite complexity
- Re-using test case- with every change in product needs modification in test cases
- Less GUI oriented automation test
- Increase test coverage
- Not enough coverage for mobile automation testing
- Better tools for Cloud Testing

Research Gap

- From the past experience of Samsung and Vmware
- Mobile testing automation process is very slow
- Preparing test suite is still manual
- Cloud testing facing lots of challenges
- Model based testing is not used much in Industry
- Industries are skeptical about automation tools

No question? Thanks 😊