Model driven workflow

applied to an

IMS Application server

Antal Wu-Hen-Chang
antal.wu-hen-chang@ericsson.com
Tibor Csöndes
tibor.csondes@ericsson.com
Agenda

› Introduction
  – Model-driven Design (MDD) and Model-based Testing (MBT) synergy
  – IMS Application Server

› Model-driven Workflow
  – Workflow overview
  – Service implementation example

› Experiences
  – Challenges and solutions
  – Benefits
introduction
MDD and MBT synergy

xtUML Model

Model Compiler

C++ Source

Executable

Model Verifier

Build Environment

Test Model

TTCN-3 Source

Executable Test Suite

Test Generator

TTCN-3 Compiler
Mission

Tool Chain
Modeling expertise
Methodology

MDW

Domain specific knowledge
Large scale development competence
Established WOW

Receiver org

Polished tool chain
Proper methods
Wide scale deployment WOW
IMS application server

- HSS
- Subscriber Data
- MMT XDMS
- TISPAN Service 1
- TISPAN Service 2
- Charging Ro/Rf
- Built in MRFC (H.248)
- Diameter
- Ro/Rf
- Mt
- ISG
- SIP
- SIP

Charging system

Subscriber Data

Sh

Ut

Web portal

MTAS

MMT

XDMS

TISPAN

Service 1

TISPAN

Service 2

Charging

Ro/Rf

Built in MRFC

(H.248)

Charging

system
Workflow Introduction challenges

- Several teams
- Seamless gradual introduction
- Agile ways of working
- Huge handwritten legacy code
- Teams have no modeling knowledge
Model driven workflow
Service specification (original)

• New service requirement
• System architect designs the service logic in a natural language document
• Specification is given to the work package teams
• Parallel development of implementation and automated tests
• In the end of the sprints they can test their systems against each other
Service specification (modeled)

- New service requirement
- System architect designs the service logic in a **formal language**
- Specification is given to the work package teams
- Parallel development of implementation and automated tests
- **Early in the process** they can test their systems against each other
Agile way of working

- Starting point: high level black box model
- Test driven development
- Continuous integration: nightly builds

CSCF
- 3 designers
- 3 testers
- 1 system

MDW
- 4 designers
- 1 tester
- 2 MC

3..4 weeks
Original TEST workflow

- WPS
- REQs
- Test analysis
  - Test case headers
    - Design TCs in TTCN-3
      - TTCN-3
      - Test Execution
        - Test results
    - Describe TCs in FTD
      - Test documents
      - Write Test Report
      - Test Report
Model based Test workflow

Specification

Modeling & Implementation

Execution & Reports

High level modeling

High Level Model

Adaptation layer implementation

Model extension

TTCN-3

Test documents

Test Execution

Test results

Generate Test Report

Test Report

WPS

REQs

Modeling & Implementation

Execution & Reports

Specification

High level modeling

High Level Model

Adaptation layer implementation

Model extension

TTCN-3

Test documents

Test Execution

Test results

Generate Test Report

Test Report

UCAAT 2013 | 22-24 October 2013 | Paris, France | Page 13
Model in the spec

Service Logic

- Top down modeling
- Some of the signaling is pushed down
Model in the spec

Service Logic

DNM Service Logic

Geo-Location Server

Common Components

SIP Originating

Media Player

SIP Terminating

• Upper layer is evolved continuously

• No structural changes in the state machine

• Common parts are organized in a separate reusable libraries
Function Testing with MBT

Test Model
  - Test goal specific
  - High level

Test Harnesses
  - Provides adaptation towards the SUT

QML model

TTCN-3 TCs + TTCN-3 Test Harness

xtUML TCs + Bridge Point Test Harness

ONE model!

Same TCs in different Languages!

› TTCN-3 scripter

› xtUML scripter

› Scripters
  - Used to generate the code that realize the tests
  - TTCN-3 scripter, OAL (xtUML) scripter
Tool chain Synergy

BridgePoint (Design model)

xtUML tests

Test Message sequences

xtUML tests

xtUML scripter

QML modeler (test model)

Model level test

Function test

Legacy infrastructure

MC C++

MC JAVA

tsp

JVM

binary

binary

ttcn testcases

ttcn scripter

MC JAVA

binary

UCAAT 2013 | 22-24 October 2013 | Paris, France | Page 17
Challenges and solutions

› Integration
  – Deciding the model boundary
  – Adaptation to legacy frameworks’ APIs

› Version control and collaborative development
  – Model merging problems
    › Graphical conflicts
    › Not resolvable conflicts: e.g. structural changes
  – Solutions
    › Tools for automated merge
    › Work separation
Challenges and solutions

› Resistance at the receiver organization
  - Remain enthusiastic
  - Working examples are the best evidences
  - Be patient, let them do it themselves
  - If one turns, the rest will follow
  - Expect slow start, they have to learn

› Consultancy
  - Orthogonal knowledge areas
  - Tailoring for the different needs
  - Providing guidelines (e.g. Do not touch the generated code)

› Tools are expensive
  - Work separation
benefits

› Modeling
  – Raising the abstraction level
  – Thinking before coding
    › Use-cases -> Test strategy
    › Fixing it on the drawing board or fixing it on-site
  – Early testing
  – Enables automation

› Test driven development
  – Builds confidence
  – Facilitates discussion