USING TASK MODELS IN MODEL-BASED TESTING

by Rachid Kherrazi
CONTENT

• Introduction to Model Based Testing
  • Why MBT?
  • What is MBT?
  • What are benefits and challenges?

• User Experience and Task Models

• From Task model to Test Model using UXSpec tool

• Case study and results

Rachid Kherrazi
  Sr. Consultant @ Nspyre
  Domain
  Process and Product Improvement
  Model Based Testing and Model Driven Engineering
  iSQI Accredited Certified Model Based Tester Trainer

Also involved in this project
  Anita Wierda  User Experience specialist
  Neda Noroozi MBT specialist
  Arjan van der Meer MDE specialist
Typical issues with traditional testing

- Many problems discovered at a late stage of development process, often costly to resolve.

- In addition, they affect the planning of development processes in that they must be resolved 'In between'

  ➔ Several pre and post release defects not found with traditional testing, which results in
  ➔ Several delays in planning
  ➔ With related quality/cost impact

Typical what we hear from our customers
Root Cause

- Unit testing is insufficient
  - Single unit may work properly in isolation
  - Incorrect interaction between units may cause serious reliability failures

- System-level testing
  - Requires **model** of system behavior
  - Behavior is often reactive/nondeterministic
    - Implementation is multi-threaded or distributed
    - Thread scheduling hard to control
  - State space is typically infinite
  - Traditional FSM-based testing does not scale
MBT helps to avoid integration issues and post release defects

Manual testing
- happy flows
- alternative flows
- bad weather flows

Review
- Static testing
- Unit testing

Typical uncovered
- Post release defects
- Integration issues

Typical uncovered
- Happy flows
- Alternative flows
- Bad weather flows

Manual exhaustive exploratory testing

Increased complexity and effort of testing

You can try to avoid integration issues and post release defects by adding manual test case, but it will be complex will take lot of effort and finally costly.

This is typical domain where Model Based Testing helps.
Other Drivers for MBT

Our world evolves....and becomes complex

- Cyber-Physical Systems/EoT
- Machine2Machine
- Digital Transformation

Increased need to manage complexity

Increased need to technology enablers

Innovative company’s need innovative tools and approaches to develop innovative products and services

- Model Based System Engineering
- Model Driven Engineering
- Model Based Testing
Test process

3 main steps in test process

- **System**
  - Test case specification
    - (design of logical test case)
    - (reduction of number of test cases by application of test techniques)
  - Test execution
    - (reporting)

- **Subsystem**
  - Test case generation
    - (design of physical test case)
    - (selection of input values and calculation of expected results)

- **Module**

Rachid Kherrazi
UCAAT 2015
MBT is the automation of test case generation

### Manual vs Automatic

**Test case specification**
- Manual: Traditional manual testing
- Automatic: Modeling Lightweight Modeling languages

**Test case generation**
- Manual: Traditional manual testing, Manual C# Scripting
- Automatic: Model based testing, Generate Scripting

**Test case execution**
- Manual: Traditional manual testing
- Automatic: Model based testing

**Skills Required**
- Testing skills
- + Scripting skills
- + Modeling skills
Model Based Testing

+ better communication using models
+ abstract tests
+ auto design of tests, variety of test suites
+ model checking
+ early exposure of ambiguities in spec/design
+ ease of updating of test suites
+ systematic coverage

- automatic execution
- auto regression testing

- modelling overhead
- complexity of the models
Example Spec Explorer and possible extensions

UML Extension
Sequence Diagrams

UXSpec
(converts Task Models to SE test models)

Model
Explore & Analyze
Remodel
Generate

Test Suite

Model Graph

Note: this is not implementation code, this is the textual Test model (describing the interface behavior)

Test Execute Tool

MBT extension to overcome modeling/scripting skills and reuse available models

Rachid Kherrazi
UCAAT 2015
**MBT: workflow**

Tester reuses existing UX Task models to generate Spec Explorer MBT models
This figure shows the interdependence of human-centered design activities according to the ISO 9241-210 standard.

Blue rectangles show the 5 key design activities in an iterative, human-centered design process. "UX deliverables" are key deliverables from the corresponding design activity.

During Analysis phase, Task Model is one of the deliverables

Task Model: A description of the subtasks that have to be carried out in order to reach the user’s goals
What are task models, anyway?

- Description of the process a user takes to reach a goal in a specific domain
- Typically have hierarchical structure
- Number of different task modeling languages
  - GOMS
  - UAN
  - ConcurTaskTrees (CTT)

<table>
<thead>
<tr>
<th></th>
<th>GOMS</th>
<th>UAN</th>
<th>CTT</th>
<th>MAD</th>
<th>GTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Order independence</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interruption</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concurrency</td>
<td>Only CPM-GOMS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Optionality</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Objects</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Performance</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-post condition</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

ConcurTaskTrees

Developed by Fabio Paterno et al. for the design of user interfaces

- Goals
  - Graphical for easy interpretation
  - Concurrent model for representing UI tasks
  - Different task types
    - Represent all tasks, including those performed by the system

<table>
<thead>
<tr>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction Task</td>
<td>Represents user interaction with the system.</td>
</tr>
<tr>
<td>Application Task</td>
<td>Represents tasks that must be performed by the system.</td>
</tr>
<tr>
<td>User Task</td>
<td>Represents user decision points.</td>
</tr>
<tr>
<td>Abstraction Task</td>
<td>Represents abstract tasks (i.e. the combination of subtasks into a higher level task)</td>
</tr>
</tbody>
</table>
Task Building Process

Three phases

- Hierarchically decompose the tasks
- Identify the temporal relationships among tasks at the same level
- Identify what objects are manipulated and what actions can be performed on them, and assign these to the tasks as appropriate.

Temporal Relationships

- $T_1 [] T_2$ - Choice
- $T_1 ||| T_2$ - Interleaving
- $T_1 [||] T_2$ - Synchronization
- $T_1 >> T_2$ - Enabling
- $T_1 [>>] T_2$ - Enabling with Information Passing
- $T_1 [>] T_2$ - Deactivation
- $T_1^*$ - Iteration
- $T_1(n)$ - Finite Iteration
- $[T_1]$ - Optional
- $T$ – Recursion
Example CTT

Show cpHomePage  Enter search information  Press confirmation  Show results

Enter origin  Enter destination  Enter date

Ref: Paulo Filipe de Jesus Cruz

Figure 5 - Searching Alpha and Intercity trains in CP web page

Rachid Kherrazi UCAAT 2015
Case study:
Car reservation system
MBT + UXspec: generated Model

13 Test cases
Complete Test cases coverage
Next Steps

» Run project based on real product from industry
» Tool improvements
   » Support data handling/configuration aspects in UXSpec, instead of relying on manual additions in Spec Explorer,
   » Supporting other MBT tools.
Questions?

MBT is.....

Strategies, tools and artifacts

Manage complexity

Reduces the need of manual or human involvement of interaction

Avoids spending time in unskilled repetitive error prone or redundant tasks

Provides bandwidth to Innovate!!
Benefits of (CTT) Task Modeling

Translation of design to Concur Task Tree seems straightforward

Clear visual way of specifying all interaction details (e.g. all input fields in a screen) and user interaction flows

Changes in requirements: Only keep this model up to date, not have to change all design mockups

Model can be maintained by others than UX (free tooling)
## Results

<table>
<thead>
<tr>
<th></th>
<th>Manual test design and manual scripting (automatic test execution)</th>
<th>MBT+ UXspec</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SUT</strong></td>
<td>Car reservation system (2 requirements)</td>
<td>Car reservation system (2 requirements)</td>
</tr>
<tr>
<td><strong>Test Design</strong> (logical test case)</td>
<td>0.5 h for 2 test specification</td>
<td>½ h for creating task model</td>
</tr>
<tr>
<td><strong>Test Generation</strong> (physical test case)</td>
<td>0.5 h for 2 scripting the test cases</td>
<td>Automatic generation of 13 test cases</td>
</tr>
<tr>
<td><strong>Test Execution</strong></td>
<td>Automatic execution (MS UI automation)</td>
<td>Automatic execution (MS UI automation)</td>
</tr>
<tr>
<td><strong>Number of Test Cases</strong></td>
<td>2 Test cases (2 test case/hrs)</td>
<td>13 Test cases (26 test case/hrs)</td>
</tr>
<tr>
<td><strong>Coverage</strong></td>
<td>2 scenarios 2 req</td>
<td>All scenarios 2 req</td>
</tr>
<tr>
<td><strong>Maintenance Effort</strong> (change/new requirements)</td>
<td>1 h for adding test scenario (new requirement)</td>
<td>½ h for updating task model</td>
</tr>
</tbody>
</table>