Combinatorial Test Architecture Design
Using Viewpoint diagram

IWCT 2013
(International workshop on Combinatorial Testing)
in conjunction with ICST 2013 in Luxembourg

NISHI, Yasuharu
The university of Electro-Communications, Tokyo

© NISHI, Yasuharu
Today's presentation

• In this presentation we'd like to share our experience of design CT
  - It's just abstract represents such as design patterns because NDA obstruct some detail explanations :-(

• This presentation is mainly on parameter-level “design patterns” of CT
  - For a large-scale and complicated software we also introduce our methodology which consists of:
    » notation (NGT)
    » process model (VSTeP)
    » techniques (design patterns, quality characteristics etc.)
Term: Parameters, Values and Test cases

<table>
<thead>
<tr>
<th>Parameter</th>
<th>OS</th>
<th>Web browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Windows Vista</td>
<td>Internet Explorer</td>
</tr>
<tr>
<td></td>
<td>Windows 7</td>
<td>Firefox</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chrome</td>
</tr>
</tbody>
</table>

Combinatorial Test Design

<table>
<thead>
<tr>
<th>ID</th>
<th>Test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vista - IE</td>
</tr>
<tr>
<td>2</td>
<td>Vista - Firefox</td>
</tr>
<tr>
<td>3</td>
<td>Windows Vista - Chrome</td>
</tr>
<tr>
<td>4</td>
<td>7 - IE</td>
</tr>
<tr>
<td>5</td>
<td>7 - Firefox</td>
</tr>
<tr>
<td>6</td>
<td>7 - Chrome</td>
</tr>
</tbody>
</table>
Our idea: modeling only parameters and combinations

<table>
<thead>
<tr>
<th>ID</th>
<th>Test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vista - IE</td>
</tr>
<tr>
<td>2</td>
<td>Vista - Firefox</td>
</tr>
<tr>
<td>3</td>
<td>Windows Vista - Chrome</td>
</tr>
<tr>
<td>4</td>
<td>7 - IE</td>
</tr>
<tr>
<td>5</td>
<td>7 - Firefox</td>
</tr>
<tr>
<td>6</td>
<td>7 - Chrome</td>
</tr>
</tbody>
</table>
Complicated and large numbers of combination

• Test engineers tend to design CT cases without deep consideration, sometimes with fear
  - They’re often troubled with combinatorial bugs and have to design CT
  - They have useful tools e.g. PICT, AETG, Allpairs
  - They consider “Is this parameter necessary for CT?”
    » Boss says “Do you take the responsibility when bugs appear on the market if you reduce CT cases?” :-(
  - They throw all sort of parameters into their CT tool without deep consideration and get huge CT cases actually
    » Ex) Sometimes I hear a question “how to make L256 OA?”

• Test engineers can feel relaxed with huge CT cases but will lose their sense or purpose of test design
  - Whether each parameter is necessary for CT?
  - Whether each combination is necessary to test?
  - They don’t have any technique for deep consideration
As a result... :-(
Three strategies for reduction of CT design

- Reduction of CT cases is one of important research theme
  1) Test cases reduction
     » To reduce test cases with fixed parameters and fixed values by all-pairs, orthogonal arrays etc.
     » Ex) from 256 ($2^8$) cases to 8 cases with an L8 orthogonal array
     » Actively researched by researchers!
  2) Test values reduction
     » To reduce values with fixed parameters by equivalence partitioning etc.
     » Ex) from 12 months to 3 types of months, i.e. long, short and Feb.
  3) Test parameters and combinations reduction: our research
     » To reduce parameters or combinations by re-consideration ;-) 
     » EX) Decide not to test combinatorially between OSs and Web browsers
Technique for deep consideration of CT design

• Test engineers should have technologies to deeply consider necessities of parameters and combinations for CT
  - Model is a technology for deep consideration in software engineering
  - We proposed NGT, Notation for Generic Testing, for test modeling

• “Viewpoint diagram” can be a technology for modeling of parameters and combinations for CT
  - Viewpoint diagram is one of diagrams of NGT
  - Viewpoint diagram fundamentally consists of boxes and lines
  - Boxes represent parameters, called “viewpoints”
  - Lines represent combinations, called “relationships”
  - Viewpoint diagram hides values and can make test engineers concentrate their attention to parameters and combinations
Viewpoint diagram and test cases

<table>
<thead>
<tr>
<th>ID</th>
<th>Test cases</th>
<th>OS</th>
<th>Web browser</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vista - IE</td>
<td>Windows Vista</td>
<td>Internet Explorer</td>
</tr>
<tr>
<td>2</td>
<td>Vista - Firefox</td>
<td>Windows Vista</td>
<td>Firefox</td>
</tr>
<tr>
<td>3</td>
<td>Vista - Chrome</td>
<td>Windows Vista</td>
<td>Chrome</td>
</tr>
<tr>
<td>4</td>
<td>7 - IE</td>
<td>Windows 7</td>
<td>Internet Explorer</td>
</tr>
<tr>
<td>5</td>
<td>7 - Firefox</td>
<td>Windows 7</td>
<td>Firefox</td>
</tr>
<tr>
<td>6</td>
<td>7 - Chrome</td>
<td>Windows 7</td>
<td>Chrome</td>
</tr>
</tbody>
</table>
Hierarchical relationships of viewpoints

- For convenience of modeling, viewpoints can be in hierarchy
  - Arrows with a closed head represent “hierarchical relationships”

```
  Environment
    ↓
  Platform
    ↓
  OS
    ↓
  Kind of OS
    { Win7, WinXP, Win2000 }
```

... Viewpoint
... Viewpoint
... Viewpoint
... Bottom viewpoint (parameter)
... Values (not viewpoints)
Types of Hierarchical relationship

- Hierarchical relationships can bear several meanings
  - is-a relationship: inheritance
  - has-a relationship: possession
  - There may be other hierarchical relationships
  - Relationships can represent their meanings with <<stereotype>>

```
OS
   <<is-a>>
   <<has-a>>
  /     \
Windows Memory
     Management Subsystem
```
Interactive relationships of viewpoints

- Viewpoints can relate each other with interactive relationships
  - Non-hierarchical relationships are necessary: Interactive relationships
  - They can also bear several meanings: combination, sequential etc.
  - Lines without arrowhead represent “combinatorial relationships”
  - Arrows with an open head represent “sequential relationships”
  - Relationships can represent their meanings with <<stereotype>>
  - In this presentation interactive relationships without stereotypes represent combinatorial relationship

```
OS          Web browser
           <<combination>>
```

```
Function    Configuration
           <<sequence>>
```
Notation of viewpoint diagram in NGT

Viewpoint

... Viewpoint

Hierarchical Relationship

Interactive Relationship

<<stereotype>>

... Stereotype

Combinatorial Relationship

<<combination>>

... Sequential Relationship

<<sequence>>

Drawing tools for mindmaps are useful
Example of part of viewpoint diagram

Test Item / SUT

E-mail client

GUI

Functions

Environment

Data

Platform

Network

OS

Hardware

Kind of OS

Version of OS

Internet Explorer
Why “viewpoint”?

• The word “viewpoint” is independent of roles

- Requirement!
- Parameter
- Test purpose
- Test condition
- Test Environment

- Analyst
- CT Researcher
- Test Manager
- Test Operator
- Test Engineer
Test architecture design

• CT consists of two layers of design:
  – Value/case-level design (Test detail design)
    » To extract values with fixed parameters by equivalence partitioning etc.
    » To design test cases with fixed parameters and fixed values by all-pairs, orthogonal arrays etc.
    » Using CT design technique/method
  – Parameter/combinations-level design (Test architecture design)
    » To find out appropriate parameters and combinations
    » To consider which parameters and combinations are suitable for the current test project
    » Making CT design modeling

• They are similar to detail design and architecture design in software development
  – They are both important!
CT engineering process

- CT should consist of engineering process similar to software development
  - CT-RA and CT-AD is traditionally called “test planning”
VSTeP: Viewpoint-based Test Process

• We proposed VSTeP, a generic process model for Viewpoint-based test architecture design not limited to CT
  – Especially for a large-scale and complicated software

Part of typical test process

VSTeP: Viewpoint-based Test Process
Viewpoint diagram is simple enough

- Viewpoint diagram is simple enough to make a test architecture design model
  - More simple than classification tree
Techniques for test architecture design

• Design patterns for test architecture design
  – Part of model which can simplify complicated design

• Styles of test architecture design
  – Typical set of top viewpoints
    » User, spec, design and bug
    » Functionality, reliability, usability and efficiency
    » etc.

• Quality characteristics of test suite
  – Maintainability of test cases
  – Automate-ability of test cases
  – etc.

• Product line engineering of test suite
  – Well-organized scheme of re-use of test cases
Overview of our methodology

Notation: NGT

Viewpoint-based Test Architecture Design

Process model: VSTeP

Techniques: Design patterns, Styles, etc.
CT Design patterns on viewpoint diagram

• This research shows four CT design patterns on viewpoint diagram
  - Interaction-Viewpoint Conversion Pattern
  - Interaction Cluster Partitioning Pattern
  - Interaction Demotion Pattern
  - Interaction Necessity Analysis pattern
Interaction-Viewpoint Conversion Pattern

• you can refine the viewpoint model and reduce test cases
  - if you can specify the source of combinatorial bugs is an overwritten shared DLL and accept risks of bugs from other sources

3 x 5
= 15 cases

3 + 5 + 2
= 10 cases
Interaction Cluster Partitioning Pattern

• you can refine the viewpoint model and reduce test cases
  – if you can specify the source of combinatorial bugs is e-mail protocols and accept risks of bugs from other sources

\[(3 \times 4 \times 2) \times (2 \times 3 \times 4) = 576 \text{ cases}\]

\[(3 \times 4 \times 2) + (2 \times 3 \times 4) + (4 \times 3) = 64 \text{ cases}\]
Interaction Demotion Pattern

- you can refine the viewpoint model and reduce test cases
  - if you can separate effects of sizes on trays and materials to duplex modes and accept risks of bugs from separation

\[
\text{Tray} \quad \text{Paper} \quad \text{Duplex}
\]

\[
3 \times 6 \times 2 = 30 \text{ cases}
\]

\[
\text{Tray} \quad \text{Size} \quad \text{Material} \quad \text{Duplex}
\]

\[
3 \times 4 + 2 \times 2 = 16 \text{ cases}
\]
Interaction Necessity Analysis pattern

- you can refine the viewpoint model and reduce test cases
  - if you’re certain that behavior of web browsers doesn’t depend on kinds of HDD and accept risks of bugs from dependency of them

```
  Web     HDD
  3 values 5 values
```

3 x 5 = 15 cases

```
  Web     HDD
  3 values 5 values
```

3 + 5 = 8 cases
We share four CT design patterns

- You can find more patterns in your CT design! :-)
Example of test architecture refinement

- Several patterns can refine test architecture
  - Both are the same meaning model of semi mission critical software
  - I refined the left model to the right model using several patterns
  - I’m sorry for insufficient explanation due to NDA :-(

Refine with several patterns
Conclusion

• Parameter-level / test architecture design for CT is important
  – especially for a large-scale and complicated software

• “Viewpoint diagram” is a technology
  for parameter-level / test architecture design for CT
  – We show NGT, notation for viewpoint diagram,
    and VSTeP, a process model focusing on test architecture design

• We share four design patterns
  on test architecture level for CT to reduce CT cases

• You can research CT technique on test architecture level
  – more design patterns for CT
  – high-maintainability CT parameter-level design
  – etc.
Thank you for your kind attention

NISHI, Yasuharu
http://blues.se.uec.ac.jp/
Yasuharu.Nishi@uec.ac.jp

© NISHI, Yasuharu