Suggestions on Screen Transition Diagram Development for Web System and How to Create Effective Test Cases

Canon IT Solutions Inc.
Takashi Otomo
Table of Contents

1. Introductions
2. Notation of Extensible Screen Transition Diagram (XSTD)
3. Notation of Extensible Screen Transition Table (XSTT)
4. Creating Test Cases from XSTD Test Cases
5. Experiments
6. Consideration
7. Conclusions
1. Introductions

- Problem And Solution Of Testing A Web System -
Current Web System

- A web system has become complex by advanced technology and diversification of user request
  - Security Authentication
  - Session Status
  - SSL
  - Browser button (back-button, update-button etc)
  - Connections with other systems
  - Increase in the number of screens ............ Etc.

But......

- A period of system development decreased

Creating test cases  \[\rightarrow\]  Short period

1. Introductions
Current Problems

Under the circumstances........

Testing a current web system has two problems.
Problems in Testing a Web System (1)

Too many technical information
browser button (back, update), SSL, cookie, session information, etc...

All detail specs are not described in specifications

It is difficult to create complete test cases.
Problems in Testing a Web System (2)

A period of system development was decreased.

Know-how of testing screen transition is left as implicit knowledge of expert test engineers, and tools are seldom used.

**Test quality depends on a skill level of each test engineer.**
Suggestion to Solution

- Suggestion:
  A method to describe screen transition diagrams, under consideration of characteristics of web systems

We call

“Extensible Screen Transition Diagram(XSTD)”
What’s XSTD? part1

Focus attention on:
Tests for screen transitions in web systems

Rich technical information:
browser button, SSL, cookie, session information, etc...

We can describe detail spec on XSTD.

Resolving the first problem

1. Introductions
What’s XSTD? part2

**Define notation:**
All test engineers can develop XSTT by the notation.

- Expert test engineers’ Knowledge
- Method of Notation
- standardization

Test quality dose not depend on a skill level of each test engineer.

Resolving the second problem

1. Introductions
Why we develop XSTD

expected effect 1

Information regarding screen transitions in each specification document can be organized.

Enable to prevent omission of test cases
Why we develop XSTD

...2

expected effect 2
By developing XSTD, important transition routes can be discovered by weighing each transition with a priority.

Enable to

select more important test cases effectively
Why we develop XSTD

---

1. We can integrate technical information which is scattered in specifications.

2. We can create test cases from a number of factors that is not described in specifications.

3. A third person can evaluate validity and completeness of test cases without understanding specifications deeply.

---

1. Introductions
2. Notation of Extensible Screen Transition Diagram (XSTD)
Conventional Test Design Process for Screen Transitions

- Reference of multi-specifications
- Dependence of skills of each test engineer
- Difficulty of reviews without understanding specifications

2. Notation of Extensible Screen Transition Diagram (XSTD)
Suggested Test Design Process for Screen Transitions

1. Specifications
   - Screen transition
   - Screen specification

2. Conventional Process
   - Creating test cases with Specifications directly

3. New Process
   - Notation of State Transition Diagrams for Screen Transitions
   - Creating State Transition Diagrams (Table)

4. Test Cases
   - Deriving the Combinations of important Routes

2. Notation of Extensible Screen Transition Diagram (XSTD)
First Step for Creating Test Cases

Specifications
- Conventional Process
  - Selecting the Combinations of important Routes
  - Creating Extensible Screen Transition Diagram (Table)

Test Cases

New Process

2. Notation of Extensible Screen Transition Diagram (XSTD)
Extensible Screen Transition Diagram (overall view)

2. Notation of Extensible Screen Transition Diagram (XSTD)
Extensible Screen Transition Diagram (close-up)

2. Notation of Extensible Screen Transition Diagram (XSTD)
Notation of XSTD

2. Notation of Extensible Screen Transition Diagram (XSTD)

- differentiating transition arrow-lines
- when transition result is different, arrow-lines are described to the number of result.

Add operations and data

<table>
<thead>
<tr>
<th>code</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
</tr>
<tr>
<td>no symbol</td>
<td>0</td>
</tr>
</tbody>
</table>
3. Notation of Extensible Screen Transition Table (XSTT)
Second Step for Creating Test Cases

Conventional Process
- Screen transition
- Selecting the Combinations of important Routes
- Integrating technical information, specs

New Process
- Creating Screen Transition Diagram
- Extensible Screen Transition Table

3. Notation of Extensible Screen Transition Table (XSTT)
Extensible Screen Transition Table (XSTT)

Second step:

We need develop a table from XSTD before creating test cases.

We called “Extensible Screen Transition Table (XSTT)”
3. Notation of Extensible Screen Transition Table (XSTT)
What’s weight

Data transfer transitions tend to have many hidden bugs. Therefore......

The transition is give increased priority by adding weight.

<table>
<thead>
<tr>
<th>code</th>
<th>detail</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>transition in SSL area</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>transition by browser back button</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>transition with creating data</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>transition with deleting data</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>transition to finish</td>
<td>1</td>
</tr>
<tr>
<td>U</td>
<td>transition with update data</td>
<td>1</td>
</tr>
<tr>
<td>no symbol</td>
<td>transition with update data</td>
<td>0</td>
</tr>
</tbody>
</table>
4. Creating Test Cases from XSTT
Third Step for Creating Test Cases

Specifications
- Screen transition
- Screen specification

Integrating technical information, specs

Conventional Process

Creating Extensible Screen Transition Diagram (Table)

New Process

Test Cases
- Selecting the Combinations of important Routes

4. Creating Test Cases from XSTT
Creating Test cases from XSTT-1

Testing single transitions
- Creating test cases between two screens from XSTT
  - 1. Coverage of the test cases between two screens was improved
    - comprehensive transition routes are automatically derived.

4. Creating Test Cases from XSTT
Creating Test cases from XSTT-1

Testing sequence of transitions-1

Selecting for high priority test cases among routes

- High priority routes is selected for the composite test cases in XSTT

Operation1: The calculation of the total weight

- The decision of the maximum number of transitions
- The decision of the repeat count
- The calculation of routes from each screen as a start position
- The calculation of the total weight
Creating Test cases from XSTT-1

Testing sequence of transitions-2
Operation2: Selecting of test cases among with the routes in all screen have top weight.

| start screen | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | total |
|--------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|     |
| A01          | 3 | 1 | 0 | 740 | 791 | 1460 | 1109 | 490 | 143 | 108 | 96  | 13598 |
| A02          | 5 | 1 | 1101 | 1476 | 2727 | 2651 | 838 | 490 | 483 | 378 | 178 | 41  | 19016 |
| A03          | 15 | 5 | 816 | 1154 | 2190 | 1649 | 784 | 441 | 447 | 348 | 113 | 41  | 14795 |
| A04          | 1 | 3 | 0 | 1632 | 2388 | 4380 | 3268 | 1568 | 882 | 694 | 896 | 350 | 220 | 10  | 29691 |
| A05          | 5 | 1 | 63 | 44 | 80 | 53 | 6 | 2 | 36 | 30 | 51 | 4155 |
| A06          | 1 | 1 | 63 | 44 | 80 | 53 | 6 | 2 | 36 | 30 | 51 | 1261 |
| A07          | 1 | 6 | 232 | 251 | 475 | 354 | 138 | 47 | 36 | 30 | 51 | 4155 |
| A08          | 1 | 15 | 816 | 1154 | 2190 | 1649 | 784 | 441 | 447 | 348 | 113 | 41  | 14795 |
| A09          | 5 | 6 | 2771 | 4815 | 9014 | 7266 | 9974 | 2882 | 3398 | 2798 | 1540 | 580 | 279 | 188 | 1104 | 34 | 59161 |
| B01          | 47 | 10 | 7 | 10 | 5413 | 3190 | 2865 | 4229 | 5445 | 1879 | 503 | 677 | 506 | 292 | 78 | 5 | 43225 |
| B02          | 95 | 10 | 7 | 10 | 12188 | 8300 | 8490 | 10173 | 8843 | 5621 | 2246 | 2305 | 2899 | 199 | 456 | 179 | 10 | 10428 |
| B03          | 39 | 2 | 12 | 15 | 13028 | 10048 | 7677 | 8017 | 9150 | 7738 | 2201 | 344 | 178 | 278 | 1137 | 34 | 59161 |
| B04          | 91 | 2 | 12 | 15 | 11640 | 8491 | 6016 | 4991 | 3990 | 2494 | 1150 | 480 | 210 | 153 | 10 | 10428 |
| B05          | 58 | 2 | 5 | 13 | 4713 | 2359 | 2122 | 2355 | 2061 | 1222 | 501 | 185 | 151 | 29 | 256 | 456 | 34 | 35614 |
| B06          | 47 | 10 | 7 | 10 | 3818 | 2363 | 1811 | 2048 | 1816 | 1065 | 624 | 288 | 158 | 29 | 256 | 456 | 34 | 2868 |
| L01          | 39 | 2 | 12 | 15 | 13028 | 10048 | 7677 | 8017 | 9150 | 7738 | 2201 | 344 | 178 | 278 | 1137 | 34 | 59161 |
| LB05         | 77 | 0 | 3 | 7 | 2384 | 1182 | 954 | 1107 | 959 | 563 | 213 | 112 | 517 | 277 | 21 | 6111 |
| LB06         | 4 | 17 | 7 | 10 | 701 | 531 | 212 | 148 | 158 | 129 | 85 | 40 | 256 | 456 | 34 | 2868 |

test cases total 2713 total 824204

The routes with the top 40th were selected for each screen. The routes amount to 2,713 cases with weight ranging from 9 to 22. The process is able to select automatically with Excel macro!

4. Creating Test Cases from XSTT
5. Experiments
Experiments

Validation for availability of XSTD(XSTT)

1. Defining Virtual site as a test candidate
2. Creating specifications of Virtual site
3. Assumption of difficult-to-find bugs

Describing XSTD and XSTT and
Creating test cases
Validation (Preparation of the Test Candidate 1)

We developed design Accommodations site, and developed specifications.

... screen transition specifications, screen specifications ...

Screen Transition specifications ...

Sample

5. Experiments
Validation (Preparation of the Test Candidate2)

Screen specifications •••※Sample

<table>
<thead>
<tr>
<th>日程</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Experiments
Validation (Bugs to be detected)

< assumption of 3 bugs>

1. Inconsistent Data
   session of screen transition (display old data)
2. Inconsistent by using browser-back-button
   (session error)
3. memory leak by continuous holding down “refine button”

We assumed difficult-to-find bugs that are resulted from screen transition combinations
Result of Validation (1st bug)-1

<procedure introducing a 1st bug>
1. Search a hotel and transit Screen "Result of Search"
2. Transit to “Select Payment”.
2. Transit to Screen "Search”, and search again.
3. Select another hotel .
4. Transit to Screen “ Select Payment”

5. Experiments
Result of Validation (1st bug) - 2

A bug occurs

Displayed old hotel!!

The bug occurs in a route that includes data transfer transition and delete data transition.

Found the Bug in a High-Priority Test Case!!
Result of Validation (2\textsuperscript{nd} bug) - 1

\begin{enumerate}
\item Hold down back-button (browser) from Screen "confirming the reservation"
\item Back to Screen "Select Payment ", and emend pay information.
\item Hold down button "confirm" again.
\end{enumerate}
5. Experiments

Result of Validation (2\textsuperscript{nd} bug) - 2

A bug occurs

Session Error!!

The bug occurs in a route that includes browser-back-button transition.

Found the Bug in a High-Priority Test Case!!
Result of Validation (3rd bug) - 1

<procedure introducing a 3rd bug>

1. Hold down button "Search Narrow" in Screen "Result of Search" several times

![Diagram showing the process of introducing a 3rd bug with multiple iterations of "Search narrow[ok]" and "Result of Search".]
Result of Validation (3rd bug) - 2

A bug occurs

System Error!!

Consideration of weight when user action repeat transition in same screen → future task

Not Found the Bug in a High-Priority Test Case!!

5. Experiments
6. Consideration
Effectiveness

- Coverage of transition between 2 screens
  When a single screen has some operation-result, the transition between its screens is expressed other transition.

Only Selected credit

6. Consideration
Effectiveness2

- Priority Transitions
  - Selection of priority Test Cases
    - Testing all transitions are impossibility
  - High weight point transitions
    - Weight is configured in date transfer, SSLAreas, screens that is filled with bugs.
    - Weight can be configured by system scope or risks

- Consideration
  - conventional process depends on the skills of each test designer.
7. Conclusions
Conclusions

The XSTD is more effective than conventional methods.

- Enable to select more important test cases effectively.
- Test quality does not depend on a skill level of each test engineer.
Future Tasks

· Method of creating test cases to find bugs more effectively
  → more appropriate-priority
  To define the finer weight of transitions in accordance with the system characteristic.

· Automation
  → Creating Test Cases
  Use a variety of tools: Microsoft Visio, ChangeVision astah*, Microsoft Excel etc.
Thank you for your attention.