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 → Short period

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…

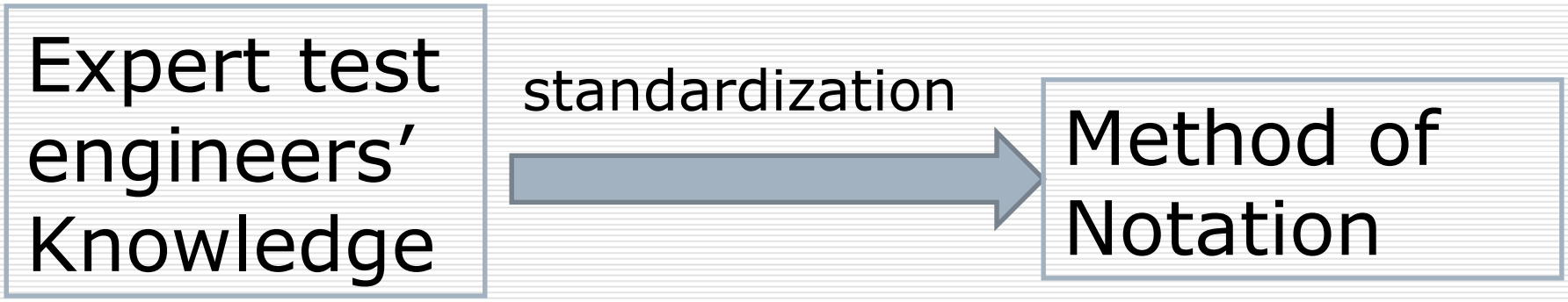
We can describe detail spec on XSTD.

Resolving the first problem

What's XSTD? part2

Define notation:

All test engineers can develop XSTT by the notation.



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

Resolving the second problem

Why we develop XSTD･･･1

expected effect 1

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

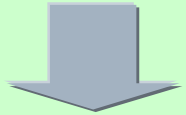
Enable to

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･･･3

expected effects
other ……

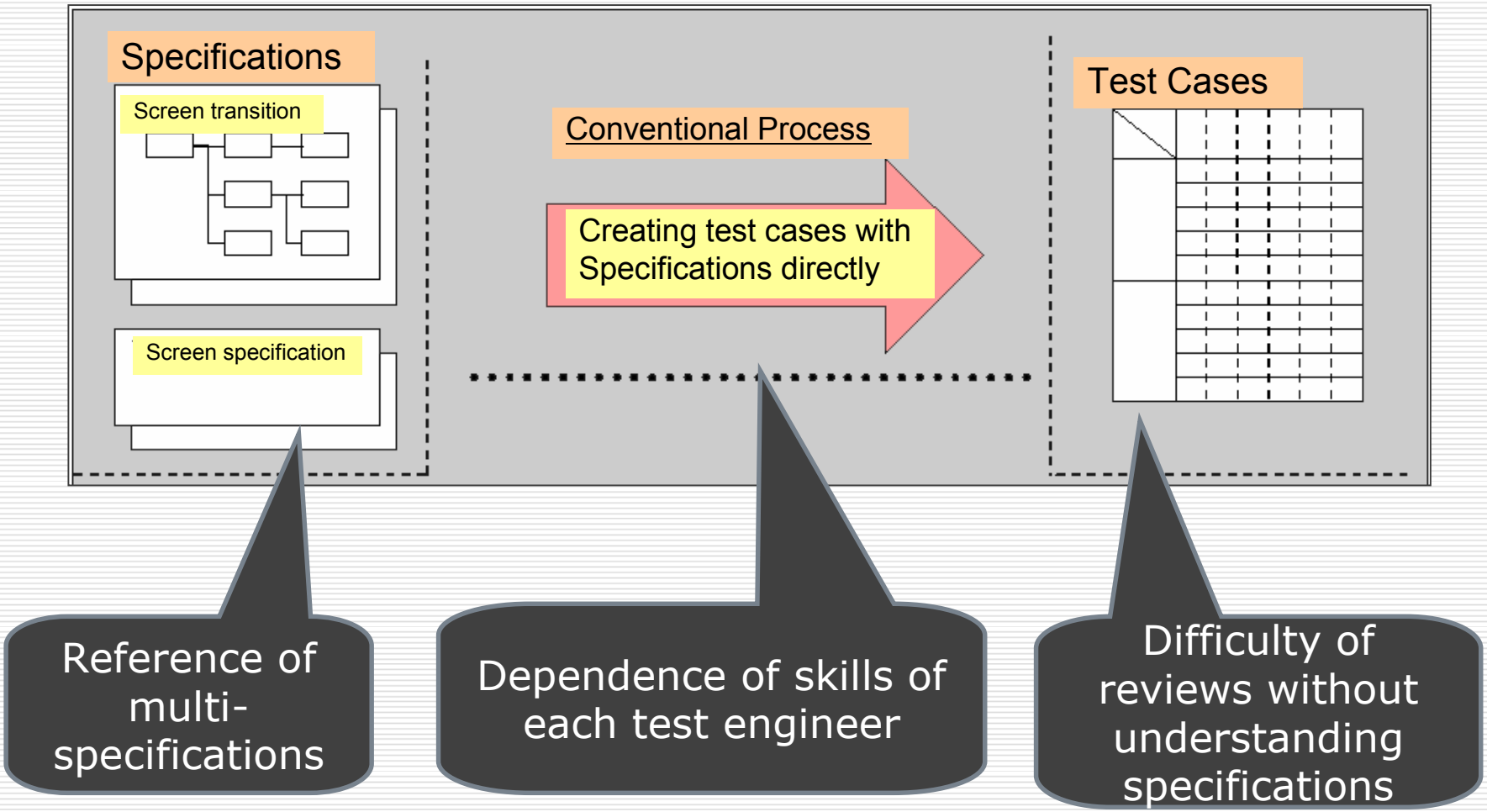
1 **We can integrate technical information which is scattered in specifications.**

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

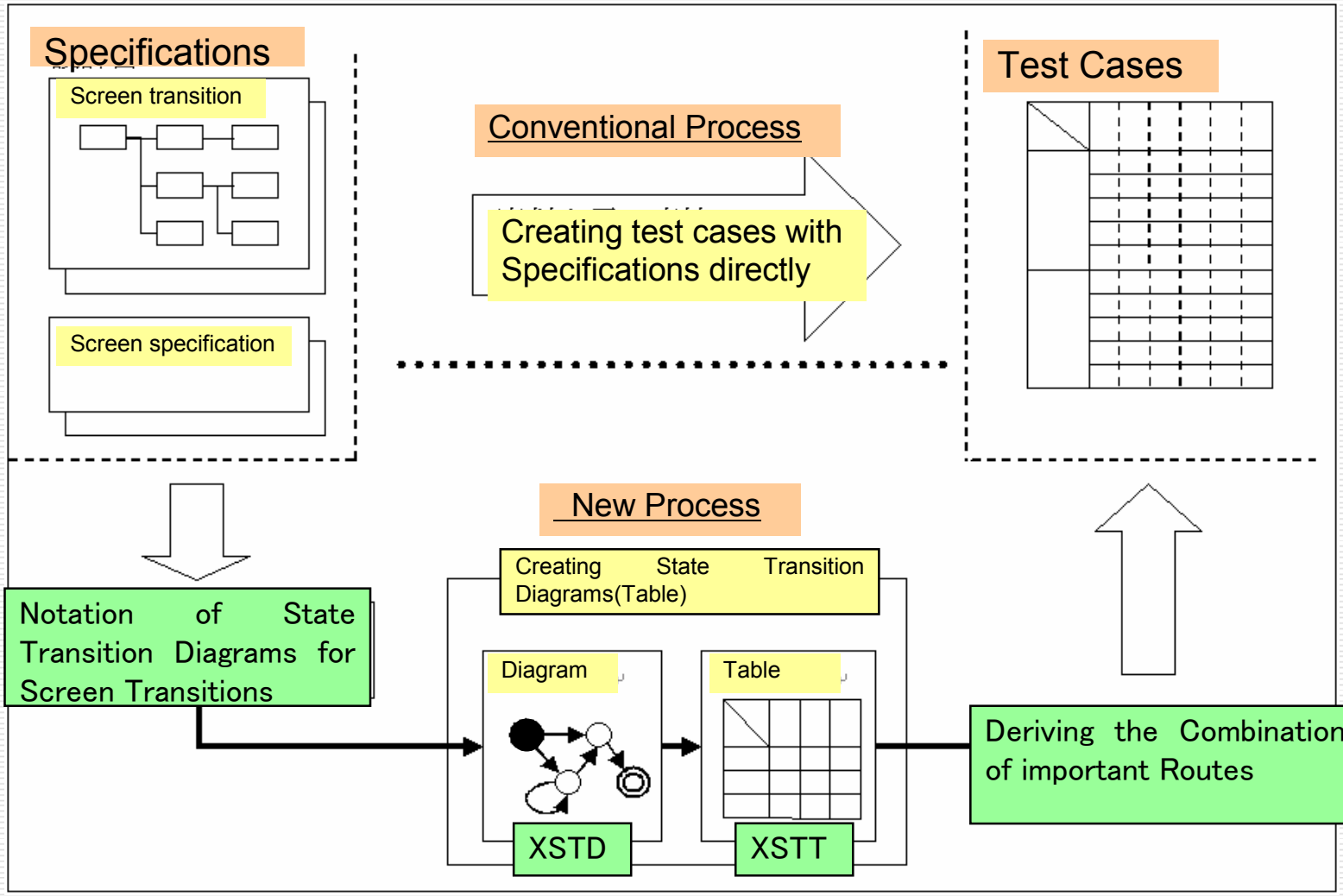
3 **A third person can evaluate validity and completeness of test cases without understanding specifications deeply.**

2. Notation of Extensible Screen Transition Diagram (XSTD)

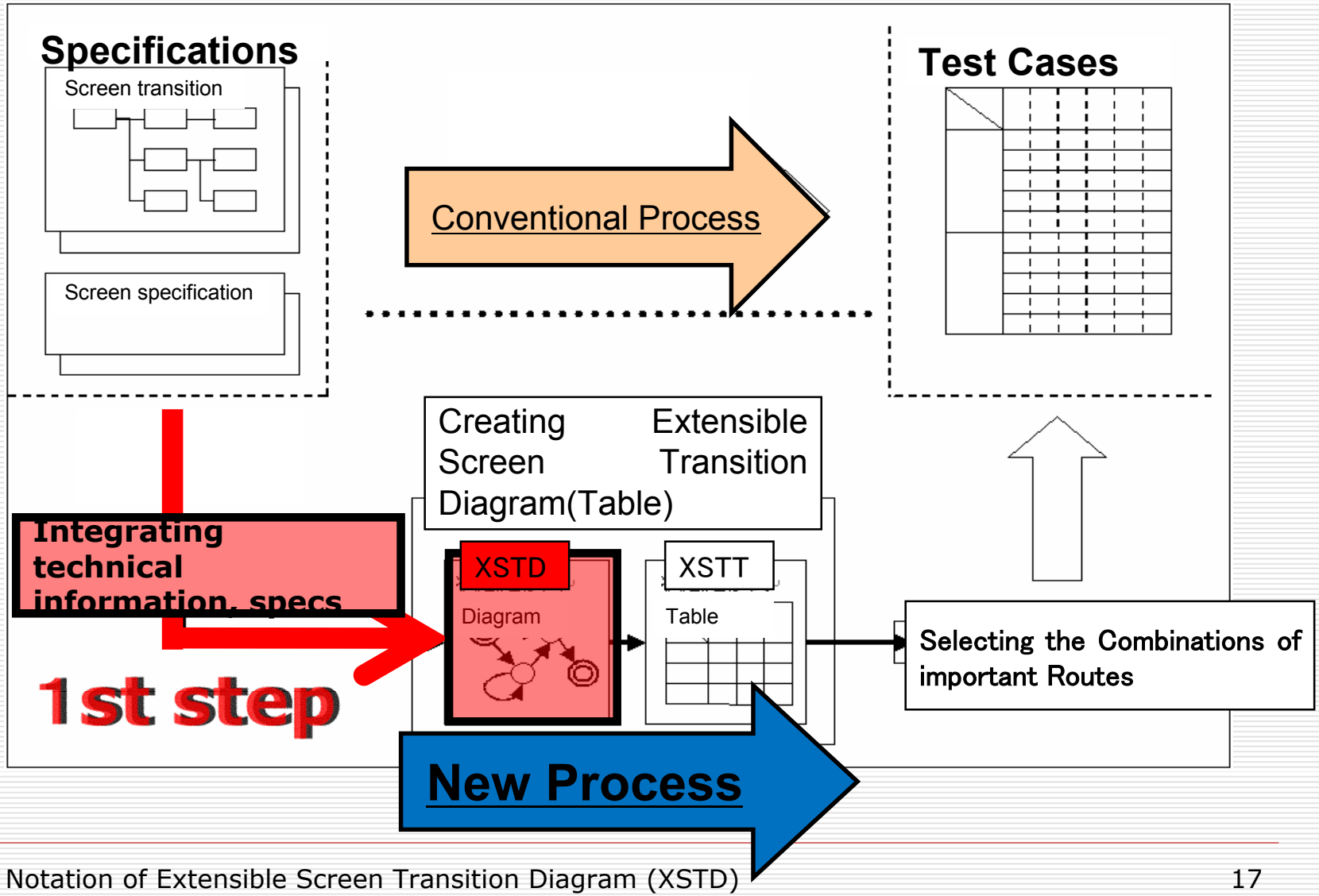
Conventional Test Design Process for Screen Transitions



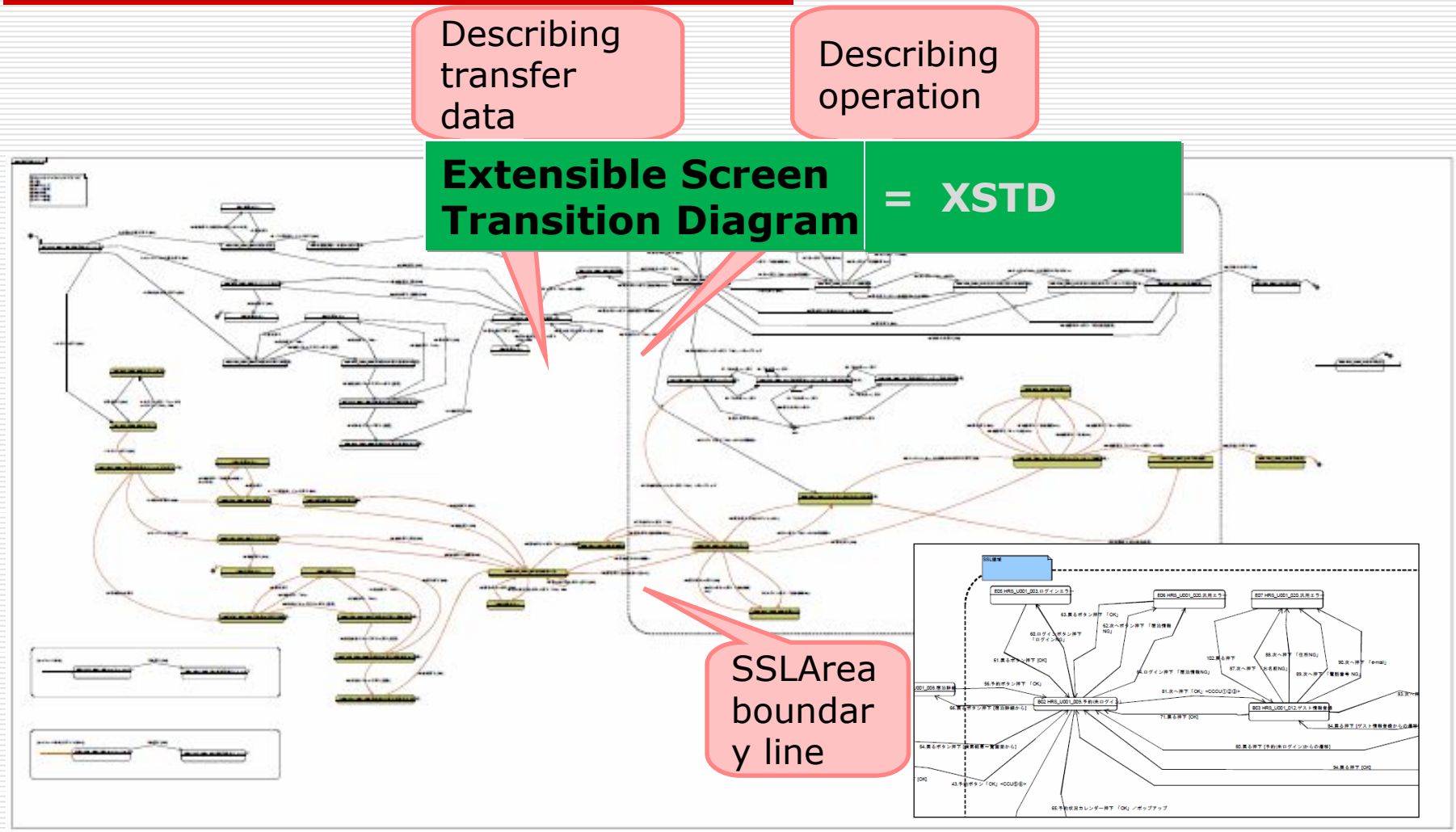
Suggested Test Design Process for Screen Transitions



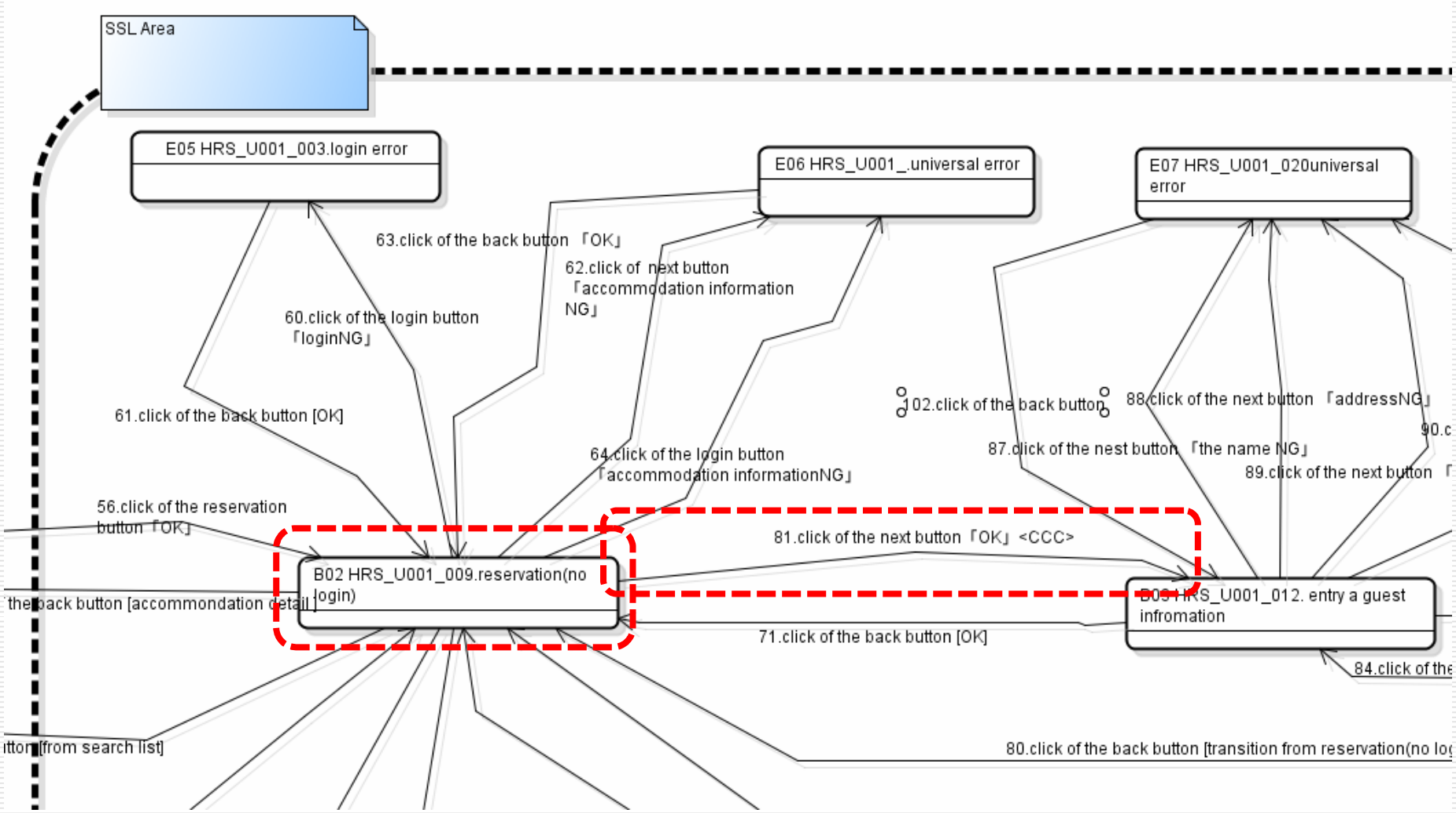
First Step for Creating Test Cases



Extensible Screen Transition Diagram(overall view)

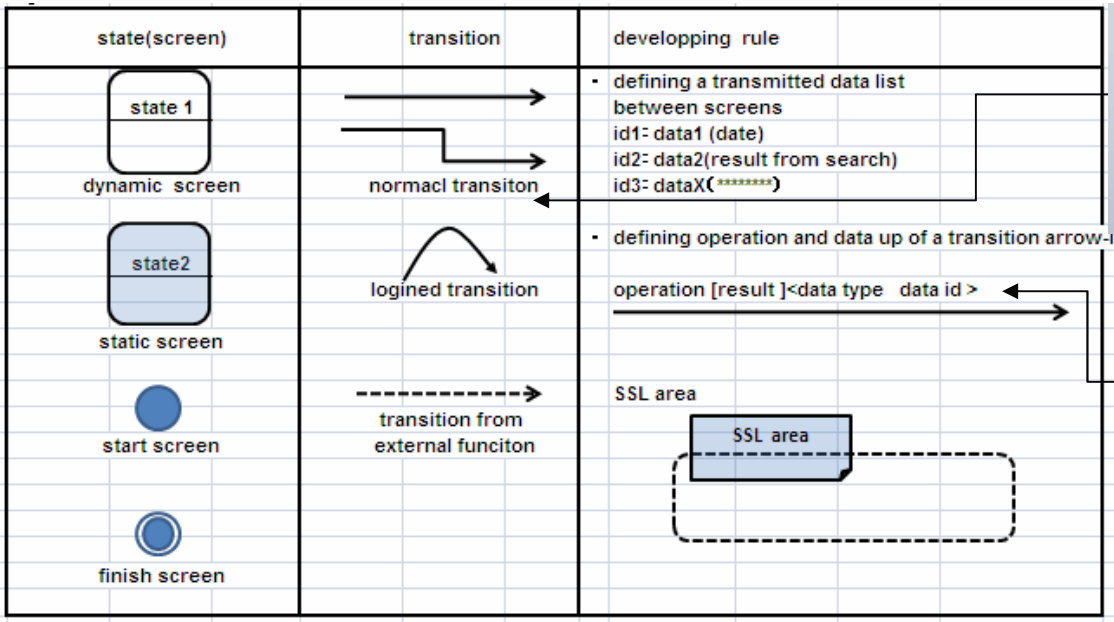


Extensible Screen Transition Diagram（close-up）



Notation of XSTD

- Extensible Screen Transition Diagram



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

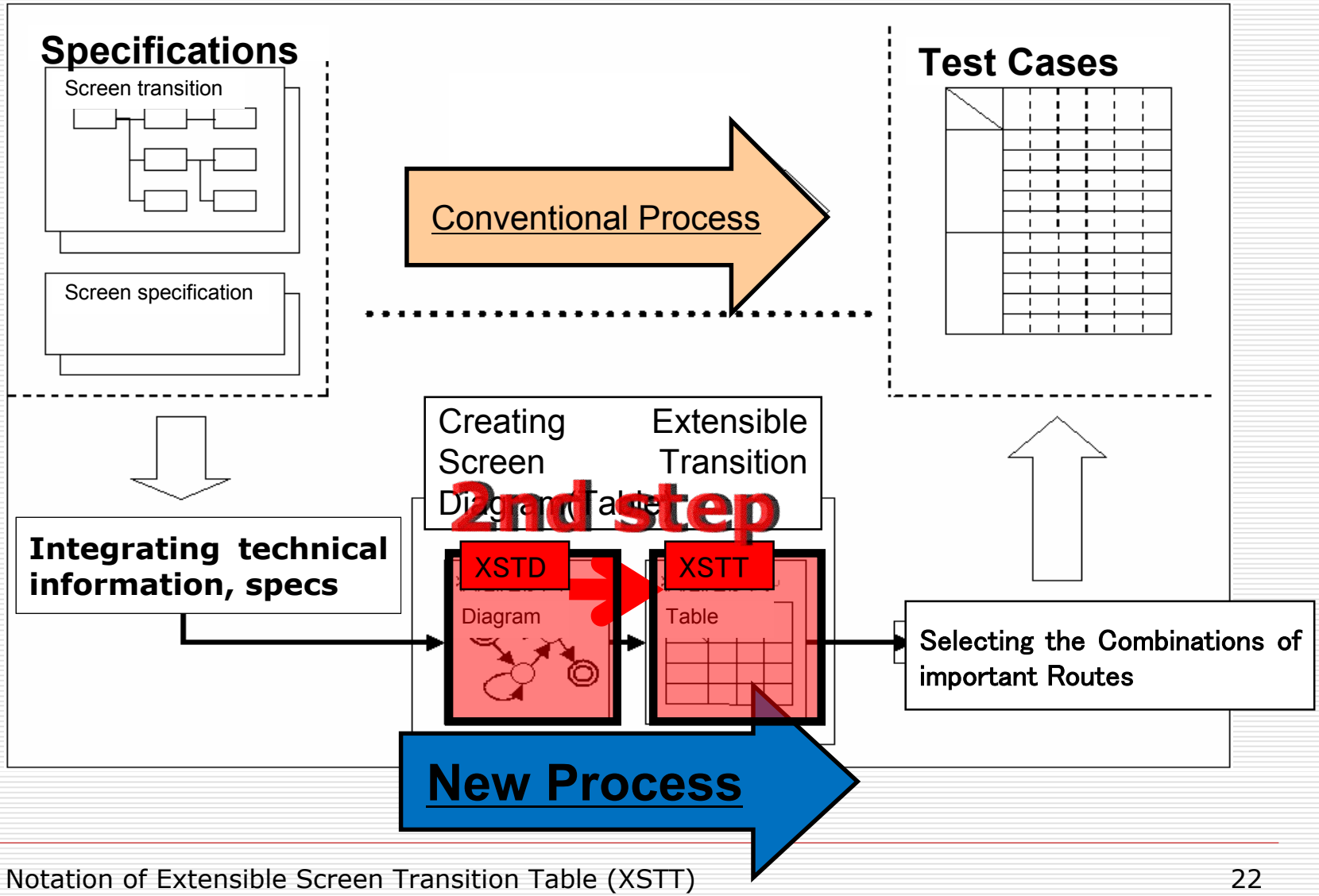
Add operations and data

Definition of transition weight

code	weight
S	1
B	1
C	1
D	3
E	1
U	1
no symbol	0

3. Notation of Extensible Screen Transition Table (XSTT)

Second Step for Creating Test Cases



Extensible Screen Transition Table(XSTT)

Second step:

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

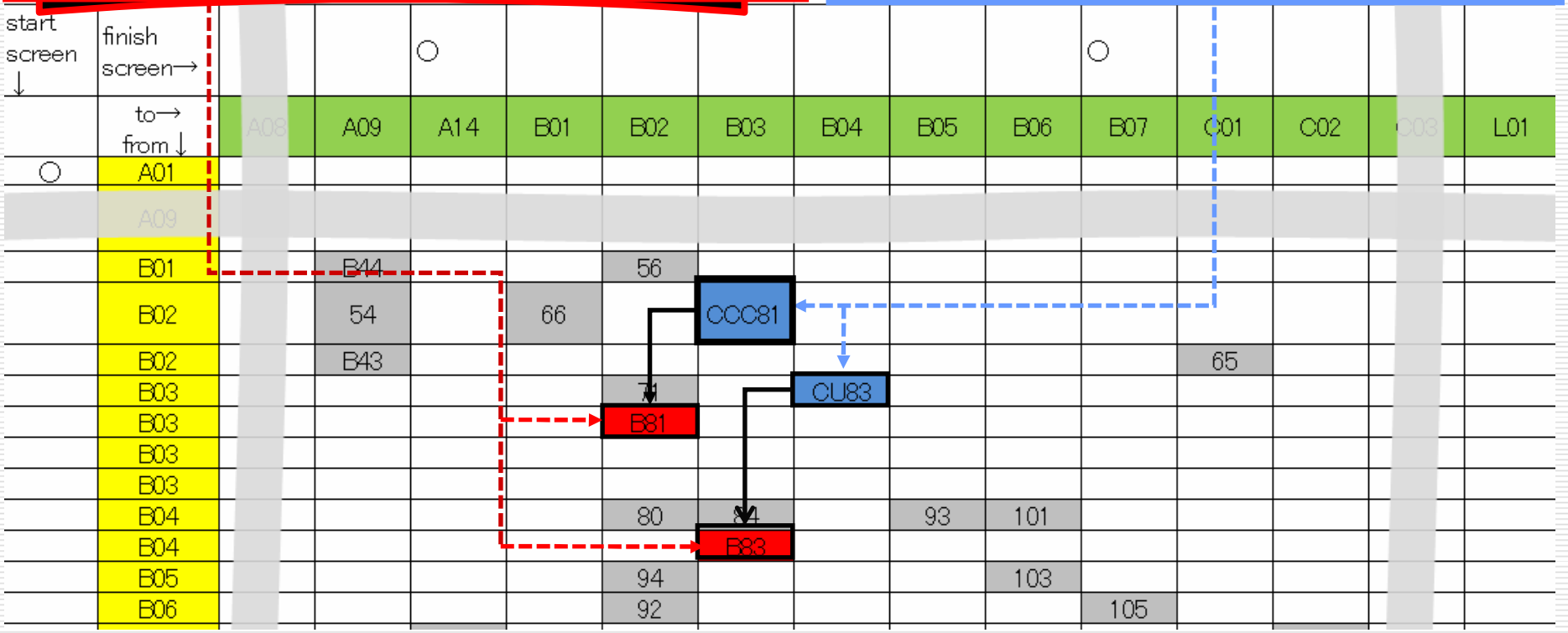
We called "Extensible Screen
Transition Table(XSTT)"

Notation of XSTT

Creating XSTT from XSTD

Addition of browser-back-button transitions
(retained data, high-weight)

Listing
・weight symbol(C,D,E,U,etc…)
・transitions No.(1,81,etc…)



What's weight

Data transfer transitions tend to have many hidden bugs .

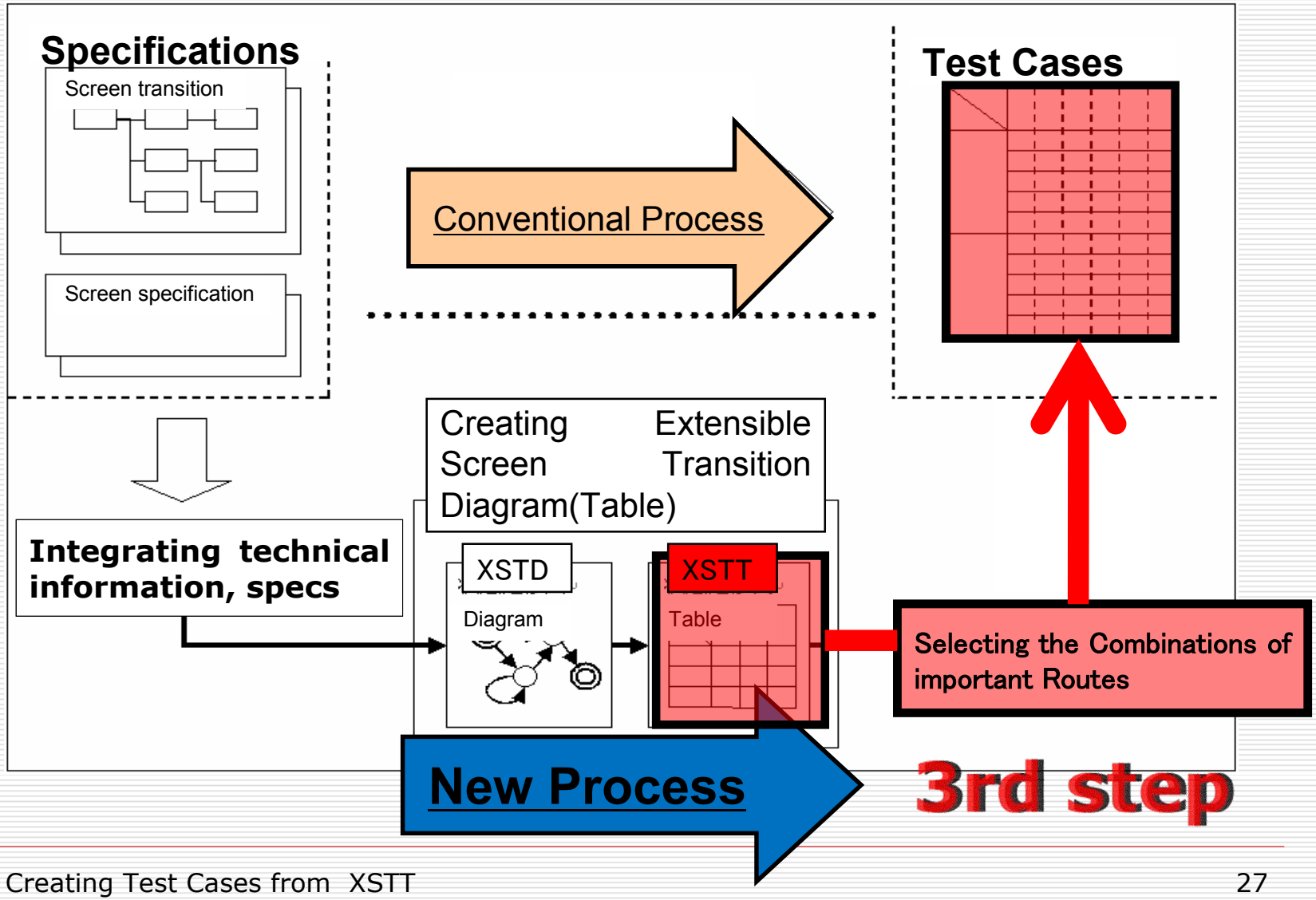
Therefore……

The transition is give increased priority by adding weight.

code	detail	weight
S	transiton in SSL area	1
B	transition by browser back button	1
C	transition with creating data	1
D	transition with deleting data	3
E	transition to finish	1
U	transition with update data	1
no symbol	transition with update data	0

4. Creating Test Cases from XSTT

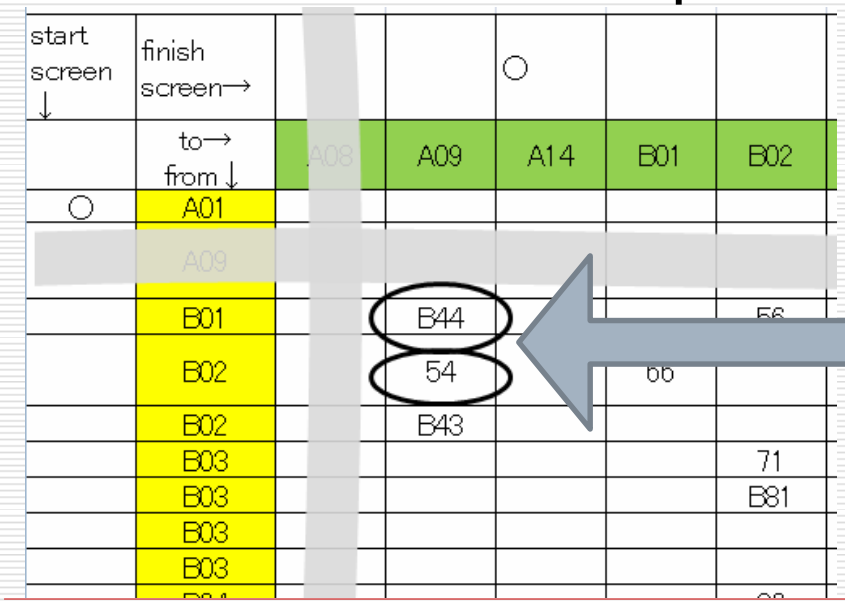
Third Step for Creating Test Cases



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.

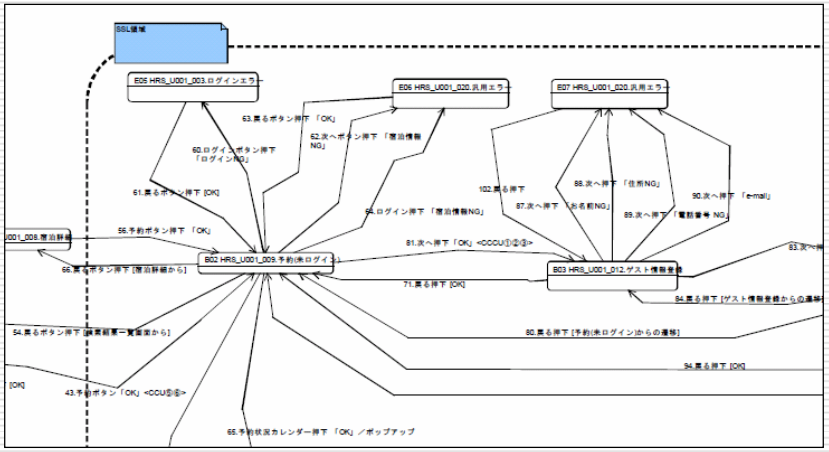
Creating Test cases from XSTT-1

Testing sequence of transitions-1

Selecting for high priority test cases among routes

- 2.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.

weight / start screen	0	1	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	total
A01	3		1	740	791	1490	1109	430	143	108	90	39						13586
A02			2	1101	1476	2727	2051	936	490	483	378	179	41	5				19916
A03			1	816	1184	2180	1649	784	441	447	348	166	41	5				14795
A04	1		3	1632	2368	4360	3298	1568	882	894	696	332	82	10				29591
A05				21	6	14	10											575
A06				63	44	80	53	13	2									1261
A07				232	251	474	354	139	47	36	30	13						4155
A08			1	816	1184	2180	1649	784	441	447	348	166	41	5				14795
A09			5	2771	4815	9014	7226	3974	2858	3308	2788	1540	589	276	188	110	34	59161
B01	47		7	5413	3169	2995	4229	3465	1879	903	677	506	263	75	5			43223
B02	95		14	12189	8300	8420	10173	8904	5621	3246	2335	1799	1084	450	123	13		104269
B03	91	2	12	13238	10848	7677	6017	5150	3736	2061	944	678	278	131	34			112710
B04	91	2	12	11640	8491	6016	4961	3900	2454	1155	490	220	102	34				97393
B05	58		5	4713	2829	2122	2358	2061	1222	501	235	158	97	34				35014
B06	47		4	3818	2353	1811	2049	1816	1095	462	230	158	97	34				28369
LB05	77		2	2384	1162	948	1107	959	563	217	112	88	50	21				20707
LB06				701	531	212	148	158	129	66	19	3						6111

tast 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.

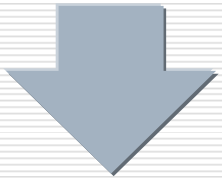
The process is able to select automatically with Excel macro!

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



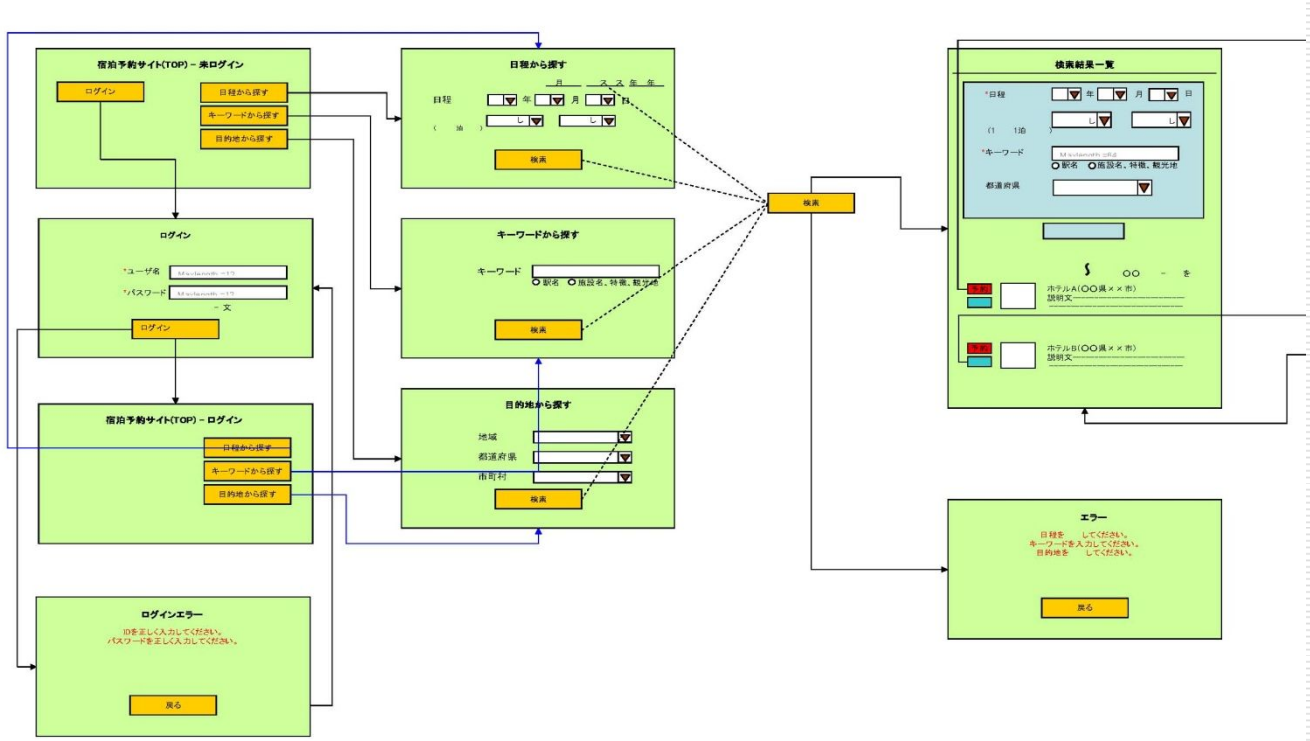
Describing XSTD and XSTT
and
Creating test cases

Validation（Preparation of the Test Candidate1）

We developed design Accommodations site, and developed specifications.

・・・・screen transition specifications, screen specifications・・・・

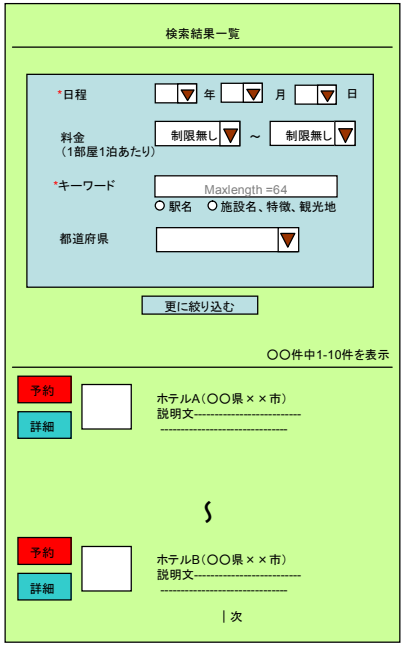
Screen Transition specifications ・・・※Sample



Validation（Preparation of the Test Candidate2）

Screen specifications ・・・※Sample

HRS_U001_007. 検索結果一覧



○概要
検索結果一覧ページ

○URL
未定

○表示内容
絞込み検索
各検索画面で設定した検索条件を該当エリアに表示。
再表示の場合、入力内容をデフォルト表示する。（セッションの情報を表示）
・日程：HRS_U001_004. 日程から探す　を参照
・キーワード：HRS_U001_005. キーワードから探す　を参照
・都道府県：「47都道府県」＋「海外」

検索結果
・検索条件と一致したデータを以下の通り表示する。
表示件数：1ページ10件。11件以上は改ページ処理を行う。
表示順：（管理）ホテル登録日の降順

○入力チェック
入力内容のチェックを行う。
入力エラー時は汎用エラー画面を表示。

○画面遷移
・「予約」：
ログイン　⇒　HRS_U001_010.予約
未ログイン　⇒　HRS_U001_009.予約
・「詳細」：HRS_U001_008.宿泊詳細
・「次へ」リンク：次ページ
・「前へ」リンク：前ページ
・「更に絞り込む」：設定条件で検索を行い、検索結果を再表示。

Validation（Bugs to be detected）

＜ assumption of 3 bugs＞

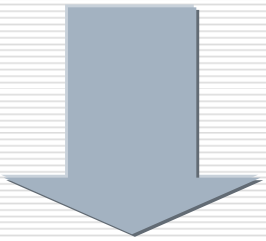
1. Inconsistent Data

   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”

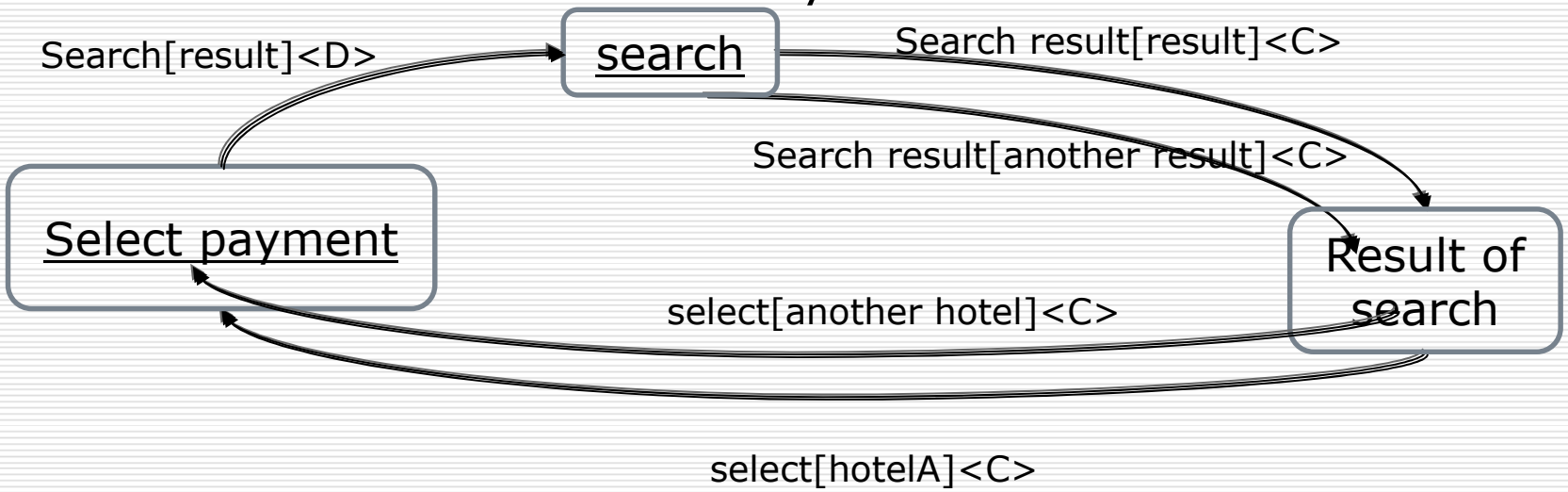


We assumed difficult-to-find bugs that are resulted from screen transition combinations

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”



Result of Validation(1st bug)-2

A bug occurs

Displayed old hotel!!

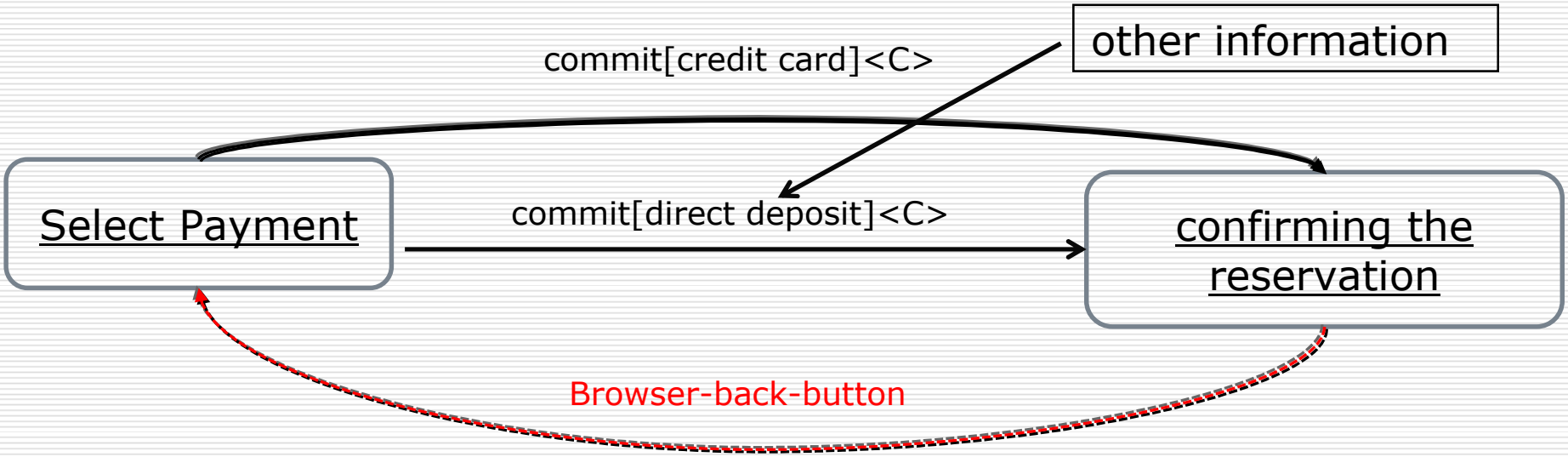
The bug occurs in a route that includes data transfer transition and delete data transition.

Found the Bug in a High-Priority Test Case!!

Result of Validation(2nd bug)-1

<procedure introducing a 2nd bug>

1. Hold down back-button(browser) from Screen "confirming the reservation"
2. Back to Screen "Select Payment ",and emend pay information.
3. Hold down button "confirm" again.



Result of Validation(2nd bug)-2

A bug occurs

Session Error!!

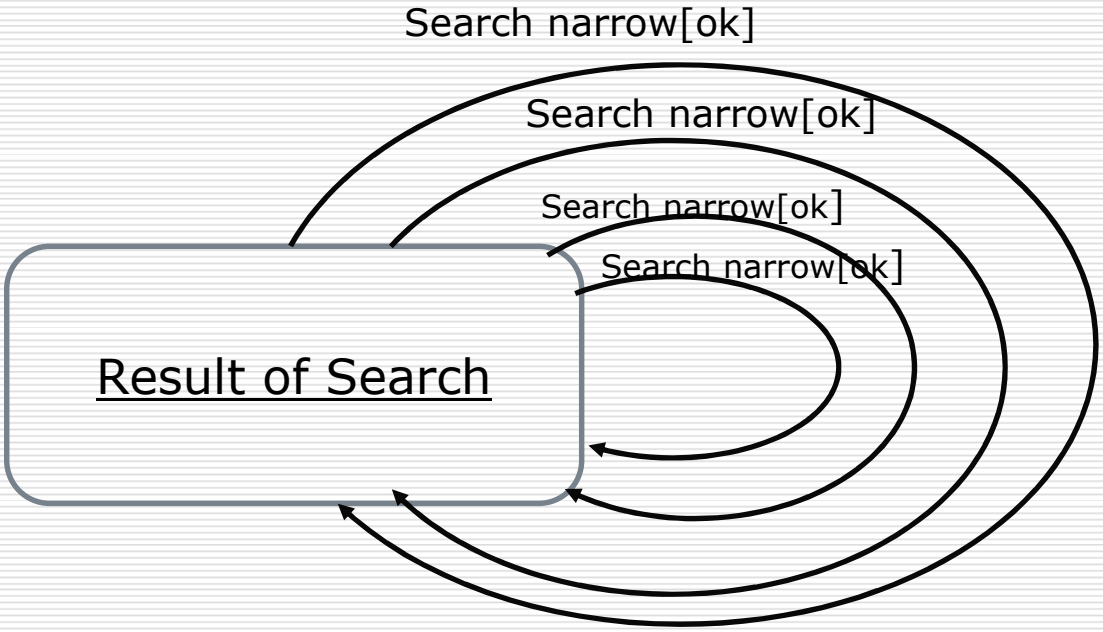
The bug occurs in a route that includes browser-back-button transition.

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



Result of Validation(3rd bug)-2

A bug occurs

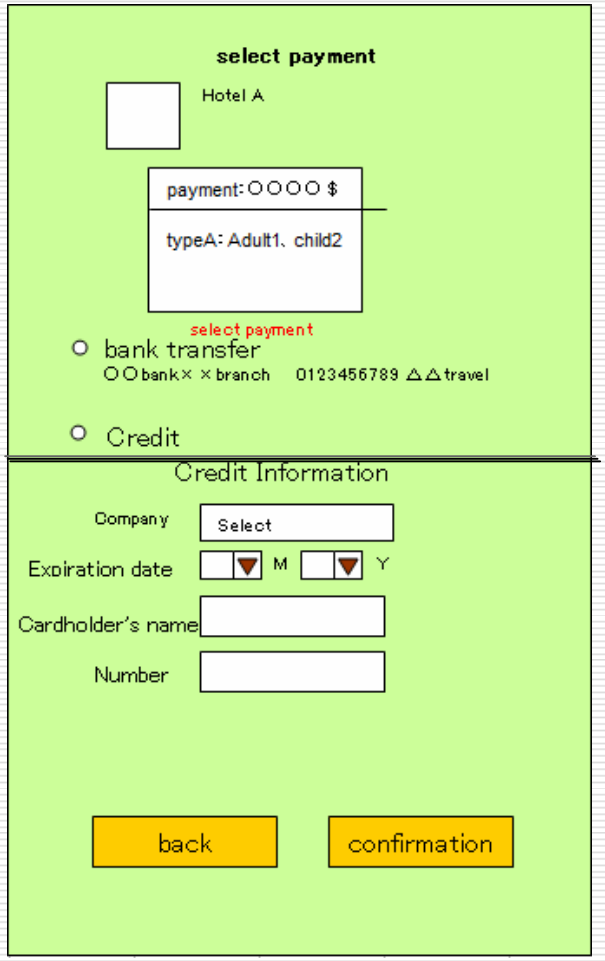
System Error!!

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

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

6. Consideration

Effectiveness1



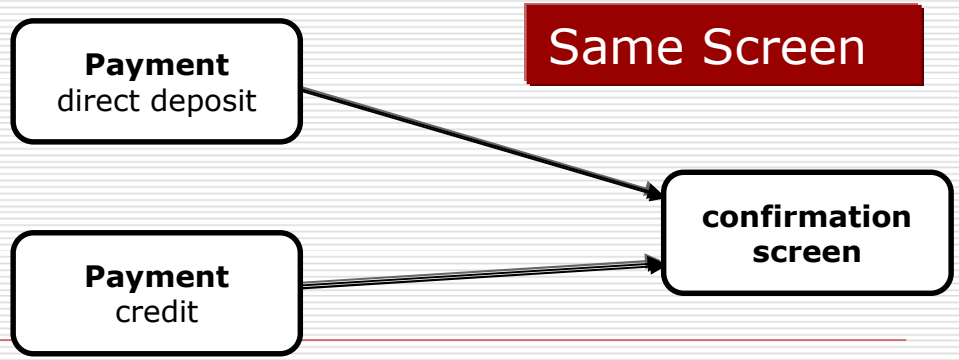
Only Selected credit

- Coverage of transition between 2 screens

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

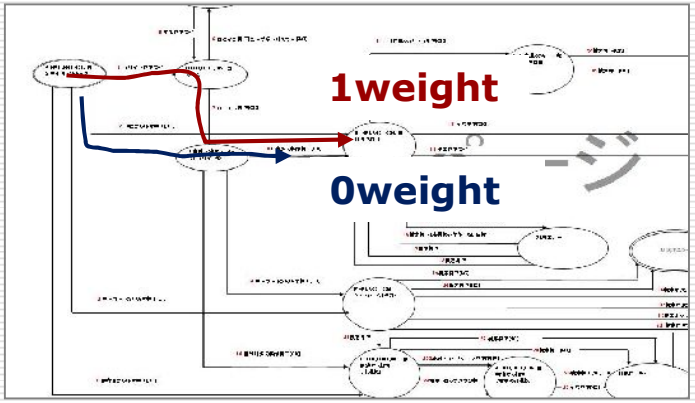
Not Omitted!

Same Screen



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



・conventional process
depends on the skills of each test designer.

7. Conclusions

Conclusions

The XSTD is 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 .