Co-developing Model for User Participation in Web Application Development

Tae Yoneda, Kouhei Mitsui, Jun Sasaki, Yutaka Funyu

Iwate Prefectural University, Japan

2005.9.30
Contents

• Background and Object of our research
• Co-developing Model
• Logic to realize the Co-developing
• Prototype System of Co-developing Environment
• Conclusion
Background [1]

Failure of the software development projects

Project Challenged Factors Top3
1. Lack of User Input
2. Incomplete Requirements
3. Changing Requirements

How to get and reflect the user requirements precisely and flexibly are important issue.
Object of our research

To propose a new software development framework to realize that user requirements can be gotten and reflected precisely and flexibly.

Scale of System: Small and medium-sized scale system
Scale of System:

• Information systems have increasingly been introduced in a variety of fields.
  – From large size system of big company to small size system nearing our life

• Iwate Prefecture needs
  • to promote agriculture and fishery
  • to promote social services
  • to develop small businesses etc.
  – Example) safety confirming system
Object of our research

To consider and propose a new software development methodology to realize that user requirements can be gotten and reflected precisely and flexibly.

System Size: Small and medium-sized scale system
System Type: a web application

is developed by using web application

User: who will use a developed system

who has the basic handling skills to use a computer
who has not the skills to develop a web application
Developing Style of our target

End User Computing

System Engineer Developing (Traditional Style)

User Oriented

Engineer Oriented
End User Computing

- “Cyber Framework” developed by Cyber Laboratory Inc.

It is still difficult to realize the style and apply it widely in practice.
Developing Style of our target

User Oriented

End User Computing
(Studying, Difficult to realize)

On-site Customer Style in XP
(Practical New Style)

System Engineer Developing
(Traditional Style)

Engineer Oriented
On-site Customer Style on XP

- The user requirements can be comparatively more satisfied.
- A user doesn't develop the system subjectively.

Difficult to develop a system that flexibly reflects the user requirements
### Developing Style of our target

- **User Oriented**
  - End User Computing
    - (Studying, Difficult to realize)
  - User and System Engineer Co-developing

- **Engineer Oriented**
  - On-site Customer Style in XP
    - (Practical New Style)
  - System Engineer Developing
    - (Traditional Style)

**Our target**
User and System Engineer Co-developing

- User can participate to system development positively.
- System Engineer support to develop the system.
Co-development Model
Co-development Model

For user, requirements can be described easily. The developing system can be confirmed by executing. Modifying the developing system can be easily.

User requirements can be understood easily and in detail. User requirements can be coded easily.
Co-development Flow [1]

Requirements can be described by natural language.

User requirements can be understood.
Co-development Flow [2]

The developing system can be confirmed by executing.

User requirements can be coded part by part.
Co-development Flow [3]

Modifying the developing system by natural language.

User

User requirements can be understood.

System Engineer

・システムの修正, 追加を提示

© Funyu Lab.
Co-development Flow [4]

The developing system can be confirmed by executing.

User

User requirements for modifying are coded.

System Engineer
Logic to realize the co-developing

Web application by LAMP
point of view

<table>
<thead>
<tr>
<th>Screen</th>
<th>Screen Composition (natural language)</th>
<th>Screen name</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screen Transition</th>
<th>Pre-screen name</th>
<th>Post-screen name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition, Transaction (natural language)</td>
<td>(program code)</td>
<td>(program code)</td>
</tr>
</tbody>
</table>

© Funyu Lab.
Actions in a web application

Step 1) decisions in respect of the screen displayed (screen transition)

Step 2) display of the screen (screen display)
Operation Model

Web Application

<Screen Transition>

(b) Evaluation and Transaction
(a) Selection

<Screen Display>

(c) Data Extraction
(e) Screen Output
(d) Data Synthesis

User request
output
Co-development procedures

Definitions of screens
Definitions of screen transitions
Library

Step 1)
Step 2)
Step 3)
Step 4)

Whole Control
Evaluation and Transaction
Selection
Data Extraction
Screen Output
Data Synthesis

User
System Engineer
Prototype System of
Co-developing Environment
select a system

create a system

modify

select a system and push "compilation" button

modify  delete

SE mode
### Input a system name

<table>
<thead>
<tr>
<th>System Name (2 byte character permitted)</th>
<th>SMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Name (2 byte character not permitted)</td>
<td>SafetyMonitoringSys</td>
</tr>
</tbody>
</table>

Click `add` to add the system names.
select a system

create a system

modify

select a system and push "compilation" button

SafetyMonitoringSystem

modify  delete

SE mode
User describe a screen by using natural language.

Screen modify

- **Screen name**: notice_sent
- **Screen outline**: Text: "Your today's condition "CONDITION" was sent (^-^)"
  
  button: "top"
<table>
<thead>
<tr>
<th>edit screen</th>
<th>edit screen transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>add screen</td>
<td>add screen transition</td>
</tr>
<tr>
<td></td>
<td>add screen transition</td>
</tr>
<tr>
<td>edit/delete a screen</td>
<td>edit/delete a screen transition</td>
</tr>
</tbody>
</table>

**Set initial screen**

- top
- set
User describe a screen transition by using natural language.

From:
- `send_safety_conf`

To:
- `notice_sent`

Transition condition:
- Push button: "fine" or "unwell" or "bad"
- Store date and state

Modify | Back

© Funyu Lab.
Engineer code for each part described by user.

<table>
<thead>
<tr>
<th>screen name</th>
<th>notice_sent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>screen outline</strong></td>
<td></td>
</tr>
<tr>
<td>Text: “Your today’s condition &quot;CONDITION&quot; was sent (^-^)”</td>
<td></td>
</tr>
<tr>
<td>button: &quot;top&quot;</td>
<td></td>
</tr>
<tr>
<td><strong>screen template</strong></td>
<td></td>
</tr>
<tr>
<td><code>&lt;html&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;body&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;form method=&quot;post action=&quot;&quot;&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;hidden&quot; name=&quot;SCREEN_ID&quot; value=&quot;${TPL_SCREEN_ID}&quot;&gt;</code></td>
<td></td>
</tr>
<tr>
<td>Your today’s condition <code>${TPL_CONDITION}</code> was sent (^-^-)&lt;br&gt;`</td>
<td></td>
</tr>
<tr>
<td><code>&lt;input type=&quot;submit&quot; name=&quot;top&quot; value=&quot;top&quot;&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;/form&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;/body&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;/html&gt;</code></td>
<td></td>
</tr>
<tr>
<td><strong>data extract</strong></td>
<td></td>
</tr>
<tr>
<td><code>$this-&gt;hash[&quot;TPL_CONDITION&quot;] = str_replace(&quot;￥￥&quot;, &quot;,&quot;, $_POST [&quot;condition&quot;]);</code></td>
<td></td>
</tr>
</tbody>
</table>
### screen transition modify

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>send_safety_conf</td>
<td>notice_sent</td>
</tr>
</tbody>
</table>

#### transition condition

- Push button: "fine" or "unwell" or "bad"
- Store date and state

#### transition condition (program)

```php
return true;
```

#### transaction

```php
if ($_POST["condition"] == "fine") {
    $state = 1;
} else if ($_POST["condition"] == "unwell"){
    $state = 2;
} else if ($_POST["condition"] == "bad") {
    $state = 3;
}
$db = new db("localhost", "himekami", "himekami", "safety_monitoring");
$todat = date("Y-m-d");
$ssql = "select dss_state from dt_safe_state where dss_date = ´{$today}´ and dss_uid = 1";
$db_stat =т$db->ExecSQL($ssql);
$db_rows = mysql_num_rows($db_stat);
if ($db_rows > 0) {
    $ssql = "update dt_safe_state set dss_state = {$state} where dss_date = ´{$today}´ and dss_uid = 1";
} else {
    $ssql = "insert into dt_safe_state (dss_date, dss_uid, dss_state) values (´{$today}´, 1, {$state})";
}
$db->ExecSQL($ssql);
```
### Edit Screen

<table>
<thead>
<tr>
<th>add screen</th>
<th>add screen transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>check_safety_conf_error</td>
<td>safety_conf_cal → check_safety_conf</td>
</tr>
<tr>
<td>check_safety_conf</td>
<td>check_safety_conf → safety_conf_cal</td>
</tr>
<tr>
<td>top</td>
<td>check_safety_conf → top</td>
</tr>
<tr>
<td>send_safety_conf</td>
<td>notice_sent → top</td>
</tr>
<tr>
<td>notice_sent</td>
<td>top → send_safety_conf</td>
</tr>
<tr>
<td>safety_conf_cal</td>
<td>top → safety_conf_cal</td>
</tr>
</tbody>
</table>

### Edit/Delete a Screen

<table>
<thead>
<tr>
<th>edit</th>
<th>delete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Set Initial Screen

<table>
<thead>
<tr>
<th>top</th>
<th>set</th>
</tr>
</thead>
</table>

© Funyu Lab.
Conclusion [1]

- Described the Co-developing Model and showed the prototype system of Co-developing environment
  - User oriented development model
  - There is a generator to generate program.
  - The developing system can be confirmed by executing.
  - Suit for Agile development

User requirements can be gotten and reflected precisely and flexibly.
Conclution [2]

Considerations;

• There is not enough support to create dynamic pages in the web application.
• There is not enough support to set up a database.

Future works;

• To evaluate this model; thus, it will be necessary to apply it to real user system developments.
• A visual interface for the user will be required.
• The creation of a mechanism to reuse the defined and programmed components.
Thank you