AIST-PS-2005-018

Efficiency Analysis of Model-based Review in Actual Software Design

Hitoshi FURUSAWA[1]
Eun-Hye CHOI[1]
Hiroshi WATANABE[1]

1: Research Center for Verification and Semantics, AIST

Efficiency Analysis of Model-based Review in Actual Software Design

Hitoshi FURUSAWA[1]
Eun-Hye CHOI[1]
Hiroshi WATANABE[1]

1: Research Center for Verification and Semantics, AIST

Efficiency Analysis of Model-based Review in Actual Software Design

Hitoshi FURUSAWA   Eun-Hye CHOI   Hiroshi WATANABE
Research Center for Verification and Semantics,
National Institute of Advanced Industrial Science and Technology (AIST),
E-mail: {hitoshi.furusawa, e.choi, hiroshi-watanabe}@aist.go.jp

November 30, 2005

Abstract

In this paper, we quantitatively analyze the efficiency of the Model-Based Review (MBR) method in an actual software design from the two points of view; one is cost and the other is reviewability. The MBR method is a modeling procedure for the purpose of reviewing preliminary design specifications of web-based applications. We previously have collected process data in applying both of the MBR method and an ordinary review to a preliminary design of a developing web-based library system. Analyzing the collected process data, we quantitatively compare the efficiency of the MBR method and that of the ordinary review. As a result of this comparative analysis, we show that the MBR method is superior to the ordinary review in terms of not only reviewability but also cost in the actual software design.

1 Introduction

Although it is said that the sooner find of defects decreases cost of software development and contributes increasing dependability and productivity [5], software developers (programmers) tend to depreciate to pursue accuracy and exactness of a preliminary design compared to a detailed design and an implementation. However, a preliminary design specification is not only for software developers but also for customers. Since, for almost customers who are not software experts, the preliminary specification is an important document to understand a system, it is necessary to prepare an accurate and exact preliminary design specification. An accurate preliminary design specification is also important for maintenance of the system.

The Model-Based Review (MBR) method that has been proposed in [8] is a modeling procedure for the purpose of reviewing preliminary design specifications of web-based applications. The MBR method models preliminary specifications, such as page-flow diagrams and functional specifications of web page items, into asynchronous processes with synchronous communication by input-output channels. The modeling procedure is not an abstract methodology but a specific and detailed operational procedure. Following the procedure, it is possible to find defects of the specifications in the modeling phase if they exist. Also, the constructed models can be used for more detailed inspection using model checking technique [1]. Such the specific procedure of MBR enables review of fine quality without depending on reviewer's knowledge, experience, or perception. The operational style of MBR also prevents a loose attention of reviewers.

Intuitively, the MBR method is much better way of review than ordinary ones by reading in a traditional way. But, there is no objective evidence. To overcome this problem, we quantitatively analyze the efficiency of the MBR method in this paper. Here, the efficiency has two points of view: one is cost and the other is reviewability. Cost indicates the time needed for the MBR method and reviewability indicates the number of defects of specifications detected by review.

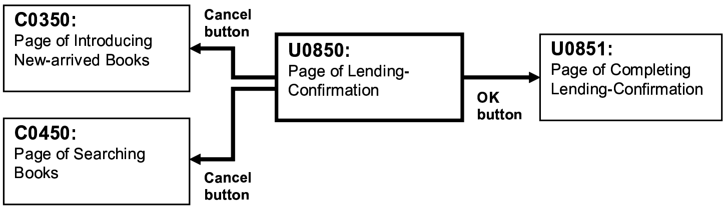
For the purpose of efficiency analysis, we have applied both the MBR method and the ordinary review in a preliminary design phase of an actual software development and collected process data of time and detected defects. The target software is a web-based library system which is being developed for Research Center of Verification and Semantics (CVS) in AIST. The experimental process data collection has been reported in [2].

In the process of data collection, we applied the MBR method to a part of the preliminary design specification of the library system. It is thus necessary to estimate cost for MBR in the case of to be applied to the whole preliminary design specification. We believe that such an estimation to fill up a lack of data is reasonable for usual cases of process data analysis because it is very difficult to spend a huge cost for data collection in an actual software development. The way of estimation that we introduce in this paper might also be effective to estimate modeling cost in general cases if its modeling procedure is clearly established.

The rest of this paper is organized as follows: Section 2.1 provides the MBR method and Section 2.2 gives a short report of applying the MBR method and collecting process data in a preliminary design of a web-based library system. Efficiency analysis of the MBR method using collected process data is then described in Sections 3–4; Cost analysis in Section 3 and reviewability analysis in Section 4. Finally, we conclude in Section 5.

2 Preliminaries

Here we first provide the Model-Based Review (MBR) method proposed in [8] in Section 2.1. We next give a short report of applying the MBR method to a preliminary design of an actual



No	Item	Specification
1	**OK Button**	When the button is clicked, 1. Mail to the administrator. 2. Transfer to U0851(Page of Completing Lending-Confirmation).
2	**Cancel Button**	When the button is clicked, 1. Do a Page Transfer. 1.1 If the previous page is C0350(Page of Introducing New-arrived Books), cancel the lending process and transfer to C0350. 1.2 If the previous page is C0450(Page of Searching Books), cancel the lending process and transfer to C0450.

Figure 1: Design Specification for Page U0850 of the Library System

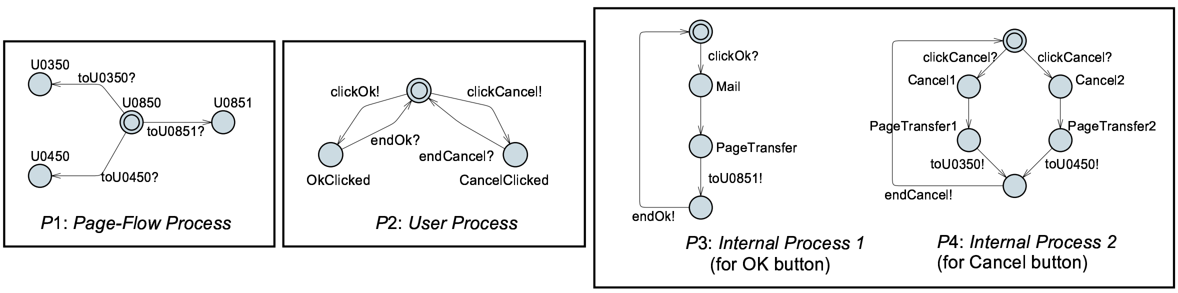


Figure 2: MBR Model for Page U0850

software that is a web-based library system being developed in Section 2.2. In Section 2.2, the outline of the experimental environment and the process data collected in the preliminary design are described.

2.1 Model-based Review Method

In this paper, we say *Model-based Review (MBR) Method* to mean a modeling method for the purpose of reviewing specifications, or a modeling procedure which accompanies side-effects of reviewing specifications. We require the following features for MBR methods:

- The modeling has explicit application target and domain in the design specifications, e.g. documents in the preliminary design or detailed design.
- The modeling provides explicit and concrete set of instructions or working procedures for the reviewers to translate the specifications into some mathematical formalisms, that is different from only providing some formalisms. The reviewer who makes models does not need to invent or elaborate complicated models by himself.
- The modeling procedure is lightweight which accepts semi-formal. We never assume heavy work load like formalizing models for theorem prover.

We summarize our MBR method for preliminary design of web-based applications as follows:

2.1.1 Target of MBR Method

The target of the proposed reviewing method is preliminary specifications of several pages or a module consisting of

1. page-flow diagram,
2. specifications of pages including of lists of items, e.g. buttons, text-fields etc. in every page, and
3. specifications of functions of the application.

Example 1 Figure 1 shows an example of target specifications which our method translate into a model. The specification consists of a page-flow diagram (the top diagram), a list of items in Page U0850 (the middle array in the bottom column) and functional explanations (the right array in the bottom column) around Page U0850 in a library system. The specification of pages and functions are not separated in the example. □

2.1.2 Our MBR Model

We apply our modeling for module or parts consisting of several pages in a system and translate above specifications into asynchronous processes with synchronous communication by input-output channels, which is a fragment of modeling language used in [7].

First, we give an informal explanation of our model and modeling procedure.

Our model consists of three kinds of processes:

page-flow process, This models the page-flow behavior of the application.

user process, This models the interaction between the user and GUI's, capturing the "pushing buttons sequentially".

internal processes, These model the behavior of programs or softwares, capturing the functions that occur after the events like "pushing buttons".

These models are extracted by the instructions 1, 2 and 3 respectively, in the modeling procedures explained in the next section.

Moreover, we consider combining them into a process. Since these processes are models of a web-application, it is possible to combine them. We assume these processes give an asynchronous process, and we use communications by channels to synchronize transitions that correspond to same events in different processes. Instructions 4-6 give what we should synchronize in these processes: Instruction 4(a) synchronizes transitions the event "the button is clicked" in the user process and "the beginning of functions" after the event in the internal processes. Instruction 4(b) synchronizes the end of "functions" in the internal process and "releasing the button" for next event in the user process. Instruction 5 synchronizes the page-transfer in the page-flow process and that in the internal processes.

2.1.3 MBR Modeling Procedures

The instruction is given by the following six steps:

1. **Extract a page-flow process** from the page-flow diagram: Just use the page-flow diagram as page-flow process, by regarding every page and page-flow as a state and a transition respectively.
2. **Extract a user process** from specifications of pages by writing a process in the following manner; Put a fresh initial state; Put states as many as buttons in the lists of items; Put two-way transitions between those states and the initial state.
3. **Extract internal processes** from the specifications of functions. For every button, we represent the functions by a process as follows: First represent the functions by processes; Combine them into a process by adding a fresh initial and a fresh terminal state called "default states", and transitions from the fresh initial state to initial states of the processes, from terminal states of the processes to the fresh terminal state, and at last, from the fresh terminal state to the fresh initial state.
4. **Put input and output channels on transitions in the user and the internal processes** by comparing the specifications of pages and that of functions: For every button,
   (a) Synchronize both outgoing transitions from the initial states in the user process and in the internal process which corresponds to the button.
   (b) Synchronize both incoming transitions to the initial states in the user process and in the corresponding internal process.

5. **Put input and output channels on transitions in the page-flow and internal processes** by comparing the page-flow diagram and the specifications of functions: For every page-flow, synchronize the transition in the page-flow process and the transitions occurring the page-transfer in the internal processes.

6. **Add appropriate guards to the beginning of internal processes.** (This is optional when we have specifications of many pages.) For every outgoing transitions from initial states in the internal processes, add guards "The state of the page-flow process is in B" on the transition when the button triggering the internal process belongs to Page B.

Example 2 Figure 2 shows an example of MBR model extracted, according to the above instruction, from the specification given in Figure 1. We give work descriptions with respect to the procedures as follows:

1. We extract Process $P1$ (without input-output channels) from page-flow diagram.
2. We extract Process $P2$ (without input-output channels) from specifications of pages.
3. We extract two processes $P3$ and $P4$ (without input-output channels) in the bottom. There are two functions specified in the document, so we extract two internal processes.
4. (a) We put labels `clickOK!` and `clickCancel!` in the User Process and `clickOK?` and `clickCancel!` in the internal processes.
   (b) We put labels `endOK?` and `endCancel?` in the User Process and labels `endOK!` and `endCancel!` in the internal processes.
5. We put labels `toU0350?`, `toU0851?` and `toU0450?` in the Page-Flow Process, and labels `toU0350!`, `toU0851!` and `toU0450!` in the internal processes.

□

Note that optional step 6 of the modeling procedure was not applied in the above example. Step 6 was also not needed for the MBR modeling of the specification of a library system explained in Section 2.2.

2.1.4 Effects and Scopes of MBR Modeling

By applying this modeling, we can detect

- missing or redundant flows in the page-flow specification,
- missing or redundant buttons in the list of items, and
- missing or redundant explanation of functions in the functional explanation.

This modeling however is not directly help to detect

- unspecified requirements,
- bugs or inconsistencies in requirement specifications, and
- bugs in implementations.

2.1.5 Yet another application of MBR models

We can feed our MBR models into model checkers, for example UPPAAL [7], and check more detailed properties in the specifications. For example, we can propose to check reachability to pages, some specific function and so on.

2.2 Applying MBR to Actual Software Design

We have applied both the MBR method and the ordinary review to a preliminary design of a web-based library system and collected process data to use for the quantitate efficiency analysis. Here, we sketch the application and the data collection. Details of them have been reported in [2].

2.2.1 Experimental Environment

Here we explain the target library system, a development team for designing the library system, and the environment for collecting process data.

Target Library System The target system is a web-based library system that will be used for a library of CVS, AIST. The system is required to fit for a library at CVS whose scale is about 100 users, 1 librarian, and about 3000 books.

Development Team The development team consists of a design engineer and two customers. The design engineer who is an examinee of our data collection has more than ten years business experience as a software engineer. The engineer has experience of model checking a few cases at the similar scale with examples in a standard text book (for example, [3]) using model checking tool SMV [4], but, is not an expert of formal methods including model checking. One of the customer is the librarian at CVS, and the other is a researcher at CVS.

Data Collecting Environment The engineer worked in a office at CVS shared with the two customers and used PC prepared by CVS for the development. The working directory of the PC is back-upped twice a day.

2.2.2 Work Order and Tasks

Here we explain the tasks performed in the preliminary design with data collection.

Work Order The design engineer has been worked in the following order:

1. Complete a preliminary design specification in a usual design procedure and an ordinary review procedure,
2. Learn the MBR method from one of the authors of [8],
3. Apply the MBR method to the design specification completed at the first and revise it if necessary.

Until the ending up the first, the design engineer had not known anything about the MBR method at all.

Tasks	Activities	Times (h)
Design Task	Investigation	77.00
	Drawing Up	248.75
	SUM	325.75
MBR Task	Learning	6.00
	Modeling	32.50
	Checking	7.50
	SUM	46.00

Table 1: Time Required for Design and MBR Tasks

Tasks The design engineer was given two tasks; one is design task and the other is MBR task. Details of these two tasks are as follows.

- Design Task

  Design task requires investigating requirement and specification and drawing up preliminary design specification to a design engineer. This task is classified into an investigation activity and a drawing up activity.

  In this task, the design engineer produced

  - List of requirements,
  - Preliminary design specification, and
  - List of failures (containing defects found by an ordinary review and a record of revisions).

  Note that the design engineer was not ordered any special requests on features of list of these three products.

- MBR Task

  MBR task requires applying the MBR method to the preliminary design specification produced in the design task, after learning the MBR method. This task is classified into a learning activity, a modeling activity and a checking activity.

  The design engineer learned the MBR method in the following order: 1. Learn the procedure of the MBR method, 2. Learn how to use model checking tool UPPAAL, and 3. Exercise modeling and checking of MBR using a toy example and UPPAAL. After this learning, the design engineer does modeling and checking including failure analysis of the target specification based on MBR without anyone's help.

  In this task, the design engineer produced

  - Models for a part of the preliminary design specification, and
  - List of failures (containing defects found by MBR and record of revisions).

The design engineer recorded the time required for each activity of the tasks everyday.

2.2.3 Collected Data

Here we show the collected data obtained as a result of the application from the two points of view; the required time and detected failures.

Time Table 1 shows the time required for each activity of the design task and the MBR task. As to the design task, it took 77 hours for the investigation activity including a requirement analysis and a specification investigation by the ordinary review and 248.75 hours for drawing up the specification.

As to the MBR task, due to the time limitation, the MBR method was applied to 4 page designs among total 55 page designs of the preliminary specifications. It took 6 hours for learning the MBR method (3 hours for a lecture and 3 hours for an exercise) and, for the 4 page designs, it took 32.5 hours for modeling and 7.5 hours for checking including a failure analysis.

A detail of the time required for applying the MBR method to the 4 page designs is shown in Table 2. As described in Section 2.1, the modeling phase of MBR consists of 5 steps: 1) extraction of a page-flow process, 2) extraction of a user process, 3) extraction of internal processes, 4) synchronization for page-flow and internal processes, and 5) synchronization for user and internal processes. Table 2 shows the time required for each step of the modeling phase and the time required for the checking phase for each page design. From a result, it took much longer for Page C0010 than other 3 pages. The reason is that the Page C0010 was the first case of the application for the design engineer and thus an extra time is needed for practicing.

Detected Failures During the design task, 5 defects of the specification were found by ordinary review. Among them, two are unspecified requirements, the other one is a lack of a data table, and another two are missing explanation of functions in page designs. After revising page designs related to these 5 defects, the MBR method was applied in the MBR task. Then, more 2 defects were found. One is a missing in the page-flow specification, and the other is a missing function of a button. Totally, 7 defects were found.

3 Cost Analysis

In this section, we show and compare the cost of ordinary review and the cost of MBR in the case of the library system design. Here cost stands for the time required for review.

3.1 Cost of Ordinary Review

As Table 1, the whole design task took 325.75 hours. The design task was divided into an investigation activity and a drawing up activity by a design engineer, and both of a requirement analysis and an ordinary review are classified as the investigation activity.

Though the data we obtained is the mixed time for these two activities, it is not very difficult to distinguish the time for the ordinary review from the time for the requirement analysis if we pay attention to the time series data collected. Recall that we collected the time series data where the time required for each activity was recorded everyday.

From the time series data, the design task began with the investigation activity, then, after 41 hours investigation, the drawing up activity began. During the drawing up activity, 12 times and total 33 hours investigation activity was done.

Page ID	Modeling Time (min.)						Checking Time (min.)	SUM (min.)
	Step 1	Step 2	Step 3	Step 4	Step 5	SUM		
C0010	30	40	90	30	30	220	15	235
U0050	15	20	20	5	10	70	5	75
M0070	10	15	20	5	5	55	5	60
M0400	10	15	20	5	5	55	5	60

Table 2: Time Required for MBR

Since, clearly, it is able to consider that the interwoven investigation activity with the drawing up activity corresponds to an ordinary review, we conclude that the ordinary review took 33 hours. The time required for the ordinary review is then about 10% of the whole design task. It coincides with the recommended average time required for review [6].

3.2 Cost of MBR

The MBR task is applied to only a part of the preliminary design specification of the library system by a time limitation, as explained in Section 2.2. Thus, it is necessary to estimate the time required for MBR in the case of to be applied to whole preliminary design specification.

Actually, MBR was applied to 4 page designs among total 55 page designs. Here we estimate the time required for the rest 51 page designs to which the MBR method was not applied. Then the time required for MBR is obtained by adding the estimated time for the 51 page designs and the measured time for the 4 page designs.

The MBR task consists of the modeling phase and the checking phase containing failure analysis, as mentioned in Section 2.2. For each phase, we first build cost formulas to represent the required times and next calculate the times required for all the rest 51 page designs based on the cost formulas. As a base of the estimation, we use the data of 3 page designs of U0050, M0070, and M0400, to which MBR was applied. However, we do not use the data of page design of C0010, since an extra practice time is included in the measured time for C0010 as explained before.

We show the way and results of the estimation for the modeling phase and the checking phase in Section 3.2.1 and Section 3.2.2, respectively.

3.2.1 Cost Estimation for Modeling Phase

The modeling phase of MBR consists of the following 5 steps: 1) extraction of a page-flow process, 2) extraction of a user process, 3) extraction of internal processes, 4) synchronization for page-flow and internal processes, and 5) synchronization for user and internal processes. The time required for the modeling phase is thus the sum of the times needed for the above 5 steps.

The time needed for each of Steps 1–3 is the time for describing processes and it then depends on the complexity of the processes. In this case, the complexity of processes can be considered to be propositional to the number of transitions of the processes. On the other hand, the time needed for each of Steps 4–5 is the time for adding communication channels to the processes. The time then depends on the number of described channels.

We therefore assume that the time required for each of Steps 1–3 is propositional to the number of described transitions of the processes and the time required for each of Steps 4–5 is propositional to the number of described input-output channels.

Under the assumptions, we first define cost formulas T_1–T_5 to respectively represent the times for Steps 1–5 of the modeling phase as follows:

1. Time for the Extraction of Page-Flow Process

   For each page, a page-flow process is given by putting every reachable page in a page-flow diagram as a state and drawing a transition from the initial state to the state. The complexity to extract a page-flow process is clearly considered to be proportional to the number of reachable pages. We thus define T_1 as follows:

   $$T_1 = c_1 \times X$$

   with coefficient c_1 and explanation variable X that denotes the number of reachable pages in a page-flow diagram.

2. Time for the Extraction of User Process

   For each page, a user process is given by putting every button as a state and drawing transitions between the initial state and the state. The complexity to extract a user process is thus proportional to the number of buttons. We thus define T_2 as follows:

   $$T_2 = c_2 \times Y$$

   with coefficient c_2 and explanation variable Y that denotes the number of buttons.

3. Time for the Extraction of Internal Processes

   For each button $i (1 \leq i \leq Y)$ in a page, an internal process is given by putting states and transitions representing a procedure of the button between default states. Let W_i denote the number of added transitions to represent a procedure of button i plus 1, where 1 represents the complexity to describe default transitions. The complexity to extract an internal process for button i can be proportional to W_i. We thus define T_3 as follows:

   $$T_3 = c_3 \times \sum_{1 \leq i \leq Y} W_i$$

   with coefficient c_3.

4. Time for the Synchronization for User and Internal Processes

   In this step, input-output channels are added to transitions outgoing from and incoming to the initial states of the

	Explanation Variables					Times for Modeling (min.)				
	X	Y	$\sum_i W_i$	Z	Z'	T_1	T_2	T_3	T_4	T_5
U0050	11	10	20	10	10	15	20	20	5	10
M0070	13	12	24	12	12	10	15	20	5	5
M0400	6	6	12	6	6	10	15	20	5	5
SUM	30	28	56	28	28	35	50	60	15	20

(a) Explanation Variables and Measured Times for 3 Page Designs.

$\Downarrow$

Coefficients	Obtained Values
$c_1\ (= T_1/X)$	35/30
$c_2\ (= T_2/Y)$	50/28
$c_3\ (= T_3/\sum_{1 \le i \le Y} W_i)$	60/56
$c_4\ (= T_4/Y)$	15/28
$c_5\ (= T_5/(Z + Z'))$	20/56

(b) Calculated Coefficients.

Figure 3: Coefficients of Cost Formulas obtained from Data for 3 Page Designs.

No	Page ID	X	Y	$\sum_i W_i$	Z	Z'
1	C0250	5	4	11	4	4
2	C0350	4	6	15	4	4
3	C0450	4	4	12	4	4
4	C0600	3	4	24	3	3
5	C0601	1	2	6	1	1
6	U0210	5	2	14	5	5
7	U0211	2	2	6	2	2
8	U0212	1	1	2	1	1
9	U0220	2	1	5	2	2
10	U0221	3	1	7	3	3
11	U0222	2	2	5	2	2
12	U0223	1	1	2	1	1
13	U0550	2	4	10	2	3
14	U0551	2	2	4	2	2
15	U0552	1	1	2	1	1
16	U0850	3	2	10	2	2
	···	···	···	···	···	···
51	M1151	1	2	5	1	1
	SUM	101	133	412	99	109

(a) Explanation Variables for the 51 Page Designs.

$\Rightarrow$

	Estimated Time (min.)
$T_1\ (= c_1 \times X)$	117.833
$T_2\ (= c_2 \times Y)$	237.500
$T_3\ (= c_3 \times \sum_{1 \le i \le Y} W_i)$	441.429
$T_4\ (= c_4 \times Y)$	71.250
$T_5\ (= c_5 \times (Z + Z'))$	74.286
SUM	942.297

(b) Estimated Modeling Times for the 51 Page Designs.

Figure 4: Cost Estimation for the Rest 51 Page Designs.

user process and internal processes. The total number of added channels equals to 4 times of the number of buttons. The required time is then clearly proportional to the number of buttons. We thus define T_4 as follows:

$$T_4 = c_4 \times Y$$

with coefficient c_4 and explanation variable Y that denotes the number of buttons.

5. Time for the Synchronization for Page-Flow and Internal Processes

   In this step, input-output channels are added to transitions occuring a page-transfer in the page-flow process and internal processes. The number of channels added to internal processes equals to the number of page-transfers described in the specification of buttons. Let Z denote this number. On the other hand, the number of channels added to the page-flow process is the number of reachable pages by the specification of buttons. Let Z' denote this number. The required time is then clearly proportional to $(Z + Z')$. We then define T_5 as follows:

$$T_5 = c_5 \times (Z + Z')$$

   with coefficient c_5.

Example 3 Consider the page design and MBR model of U0850 shown in Figure 1 and Figure 2. Values of explanation variables X, Y, Z, Z' and W_i for U0850 are obtained as follows: From the page-flow diagram, the number of reachable pages from U0850 is 3, that is, $X = 3$. The number of buttons is 2, that is, $Y = 2$. $Z = 3$ since 3 page-transfers, 'Transfer to U0851', 'Transfer to C0350', and 'Transfer to C0450', exist in the specification of buttons. $Z' = 3$ since the number of reachable pages in the specification of buttons is 3. For the internal processes of Button 1 (OK button) and Button 2 (Cancel button), 2 transitions and 6 transitions are respectively described from the specification of buttons. Hence $W_1 = 1+2$ and $W_2 = 1 + 7$. For every page design, values of explanation variables are obtained in this way. □

Next, coefficients c_1–c_5 in the cost formulas are then obtained from the given times and the value of explanation variables for 3 page designs of U0050, M0070, and M0400, as shown in Figure 3. For example, coefficient $c_1 (= 35/30)$ is calculated by dividing the total time for Step 1 by the total number of reachable pages for U0050, M0070, and M0400.

Finally, cost estimation is performed for the rest 51 page designs using the defined cost formulas and obtained coefficients, as shown in Figure 4. For example, since the sum of the value of X for the 51 pages is 101, the time required for the 51 pages in Step 1 is calculated by $c_1 \times 101 \doteq 117.833$. The estimated time of modeling for the 51 pages is 942.297 (min) as in Figure 4 and the given time of modeling for the 4 pages is 400 (min) as in Table 2. Hence, we conclude that the total time for the modeling phase of MBR is 1342.297(=942.297+400) minutes.

3.2.2 Cost Estimation for Checking Phase

In the checking phase of MBR, reachability checking was performed. For the reachability checking, formulas representing page-transfer in a page-flow diagram are generated and it is checked if the formulas hold for an MBR model using a model checker. The time required for the checking phase then can be considered to be propositional to the number of the generated formulas. The number of generated formulas equals to the number of reachable pages in a page-flow diagram.

We then define a cost formula T_6 to represent the time required for the checking phase as follows:

$$T_6 = c_6 \times X$$

where coefficient c_6 and explanation variable X which is the number of reachable pages in a page-flow diagram.

Coefficient c_6 is obtained from the given time for the checking phase and the value of explanation variables for 3 page designs of U0050, M0070, and M0400. The result is that $c_6 = 15/30 = 0.5$.

Since the sum of the value of X for the rest 51 pages is 101, the time required for the checking phase of the 51 page designs is estimated as $c_6 \times 101 = 50.5$ (min). The given time of checking for the 4 pages, C0010, U0050, M0070, and M0400, is 30 (min) as in Table 2. Hence, we conclude that the total time for the checking phase of MBR is 80.5 minutes.

3.3 Cost Comparison of MBR and Ordinary Review

By the estimation as explained in Section 3.2, the time required for the modeling phase and the checking phase of MBR in the case of to be applied to the whole preliminary design specification are respectively 1342.297 and 80.5 minutes. In addition, the time required for learning the MBR method was 6 hours as shown in Table 1. Hence, the total time required for MBR is estimated to about 29 hours and 42.8 minutes (≑1342.297+80.5+360 minutes).

On the other hand, the time required for the ordinary review was 33 hours as mentioned in Section 3.1.

As a result, one can see that the time required for MBR is shorter than the time for the ordinary review. We expect that the efficiency of MBR against the ordinary review with respect to cost is more increased when the size of a target design specification becomes larger.

4 Reviewability Analysis

In this section, we show and compare the reviewability of ordinary review and the reviewability of MBR in the case of the library system design. Here the reviewability is simply measured by the number of defects detected by review.

4.1 Reviewability of Ordinary Review

In the case of the library system, 5 defects were found by the ordinary review as shown in Section 2.2.

The 2 defects among them were missing explanation of functions. We certified that those defects can also be detected by applying the MBR method to the previous version of a preliminary design specification before revising those defects. The other 3 defects were unspecified requirements and a lack of a data table, and those could not be defected directly from

MBR since those are different kinds of defects from the scope of MBR. Such defects should be detected in a proper investigation of requirement and data table design.

4.2 Reviewability of MBR

As shown in Section 2.2, 2 defects were detected by MBR from the preliminary design specification revised after the ordinary review, even though the MBR method was applied to only 4 page designs among 55 page designs.

Moreover, we certified that more 8 defects can be detected by applying MBR to the rest 51 page designs. The 8 defects were found as a side-effect of the cost estimation for the rest 51 page designs.

In order to estimate the cost of MBR, we counted complexity points of models, that is, values of explanation variables of cost formulas. The complexity point is closely related to number of edges in models, as mentioned before. Thus, for the cost estimation, an imaginary (or rough) modeling of all page designs was performed following the modeling procedure of the MBR method explained in Section 2.1. Of course, since the modeling was just for counting edges, we just worked upto the third step of the modeling procedure using a pen and paper or without writing down the models instead of modeling carefully as the design engineer did using the UPPAAL tool. Then, still, we could find the 8 defects by the modeling for the cost estimation.

Half of the detected 8 defects are missing functional explanation, and the others are inconsistencies between page-flow diagrams and functional explanation, for example, functional explanation in the specification requires a page-transfer which is never appeared in a page-flow diagram.

It is surprising that we could found more defects than the design engineer's ordinary review by not serious application of the MBR method but just an imaginary modeling along the procedure of MBR.

4.3 Reviewability Comparison of MBR and Ordinary Review

For the target preliminary specification of the library system, the number of defects that can be detected by MBR was 10 as shown in Section 4.2, while the one by the ordinary review was 5 as shown in Section 4.1. As a result, one can see that the reviewability of MBR is higher than that of the ordinary review.

5 Conclusion

In this paper, we showed the efficiency of the MBR method in an actual process of preliminary design for the web-based library system by a comparative analysis against an ordinary review. For the efficiency analysis, we considered the two points of view; cost and reviewability. Cost was measured by the time required for the review and reviewability was measured by the number of defects detected.

A difficulty for the analysis was that the given process data for MBR was for a part of the preliminary design specification of the library system. To handle this difficulty, in the cost analysis, we constructed cost formulas for MBR and estimated the cost of MBR in the case of to be applied to the whole preliminary design specification. An estimation to fill up a lack of data is necessary in data analysis where the target is an actual software and it spends a huge cost for data collection. We believe that the way of our estimation can also be effective for general cost estimation of formal modeling.

As a result of the analysis, the cost of applying the MBR method including preliminary learning cost was less than the cost of an ordinary review. In addition, the reviewability of the MBR was higher than that of the ordinary review. Let's consider the efficiency of review using the following criterion:

$$\frac{\mathrm{Reviewability}(=\text{Number of defects detected by review})}{\mathrm{Cost}(=\text{Time required for review})}.$$

From the results of Sections 3.3 and 4.3, the efficiency of MBR is about 0.42($\doteq$ 10/23.7) and that of the ordinary review is about 0.15($\doteq$5/33). Hence, we can ascertain that the efficiency of MBR is much higher than that of the ordinary review.

The target specification of the library system to which the MBR method and the ordinary review were applied is a widely acknowledged usual preliminary design specification for a web-based applications. In the data collection, we did not order the design engineer any special requests on features of design specification. Therefore, we conclude that the MBR method can be a superior review method to ordinary review for general preliminary design of web-based applications.

6 Acknowledgments

A part of this work is supported by Japan Science and Technology Agency and also by MEXT Japan. Authors would like to thank Prof. Tohru Kikuno and Assistant Prof. Osamu Mizuno of Osaka University for their helpful suggestions and comments. Authors also would like to thank Akihiro Tokai, Atsuo Kishimoto, and Masahi Kamo of Reserch Center for Chemical Risk Management of AIST for their advice on collecting data, and thank to members of the development team, Yoshihito Wakano of KS Solutions and Kazuyo Murakami of CVS, AIST.

References

[1] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[2] H. Furusawa, H. Watanabe, T. Kawamoto, and E. Choi. Data collecting experiment to evaluate efficiency of a formal method - In preliminary design of a Web system (in Japanese). In *Proc. of the Second Symposium on Science and Technology for System Verification*, pages 196–204, 2005.

[3] M. R. A. Huth and M. D. Ryan. *Logic in Computer Science - Modelling and reasoning about systems*. Cambridge University Press, 2000.

[4] K. L. MacMillan. *Symbolic Model Checking*. Kluwer Academic, 1993.

[5] G. Myers. *The Art of Software Testing*. John Wiley and Sons, 1979.

[6] Y. Takagi, T. Tanaka, N. Niihara, K. Sakamoto, S. Kusumoto, and T. Kikuno. Analysis of review's effectiveness based on software metrics. In *Proc. of the 6th International Symposium on Software Reliability Engineering*.

[7] UPPAAL. http://www.uppaal.com/.

[8] H. Watanabe and T. Kawamoto. Model-based review method for preliminary specifications of web-applications (In Japanese). In *Proc. of the Second Symposium on Science and Technology for System Verification*, pages 122–133, 2005.

ソフトウェア設計仕様に対するモデルベースレビュー法の便益性評価
(in English)
(算譜科学研究速報)
発行日：2005 年 12 月 20 日
編集・発行：独立行政法人産業技術総合研究所関西センター尼崎事業所
システム検証研究センター
同連絡先：〒661-0974 兵庫県尼崎市若王寺 3-11-46
e-mail：informatics-inquiry@m.aist.go.jp

Efficiency Analysis of Model-based Review in Actual Software Design
(Programming Science Technical Report)
Dec. 20, 2005
Research Center for Verification and Semantics (CVS)
AIST Kansai, Amagasaki Site
National Institute of Advanced Industrial Science and Technology (AIST)
3-11-46 Nakouji, Amagasaki, Hyogo, 661-0974, Japan
e-mail: informatics- inquiry@m.aist.go.jp