Automatic Generation of an Operation Procedure Presentation System Reusing User's Input Data

Shimon Nakamura[1], Hajime Iwata[2], Junko Shirogane[3] and Yoshiaki Fukazawa[1]
[1]*Department of Computer Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan*
[2]*Department of Network and Communication, Kanagawa Institute of Technology, 1030 Shimoogino, Atsugi-shi, Kanagawa, 243-0292, Japan*
[3]*Department of Communication, Tokyo Woman's Christian University, 2-6-1 Zenfukuji, Suginami-ku, Tokyo, 167-8585, Japan*

Keywords: Operation Support, Input Reuse, Operation Procedure, Ontology, Activity Diagram.

Abstract: Users use software applications to achieve a goal. Occasionally they make mistakes in the operation path due to the complexity of large-scale applications, which requires them to back track to the appropriate operation step and reenter previously input data. This is burdensome for users. Herein a method is proposed to generate an operation support system that reuses previously input data in an inappropriate operation path as much as possible by navigating users to the appropriate operation path. Specifically, our method has an input reuse function for copying previously input data to similar input items as well as an operation procedure presentation function to highlight the operation procedure from the current step to the goal. Our integrated operation support can minimize users' rework. To generate our system, developers must create an ontology, including concepts of label names of input items, correspondence between input items and label names, an activity diagram of the target application, and the operation procedure. Our system uses this information to compute the similarity of label names between input items, copy input data for similar input items, and present operation procedures to users.

1 INTRODUCTION

Diverse software applications and web services are important in business and everyday life. Some of these applications are complicated and large-scale. Their complexity often requires users to complete many tasks to achieve the intended purpose, which may cause users to make mistakes in the operation path. When a mistake is made, a user must retrace his or her steps and then proceed down the appropriate path. The data input in the inappropriate path must be sometimes manually reentered in the appropriate operation path.

Operation support systems have been developed to aid users to the appropriate operation path. Examples include online help or instruction manuals. However, many support systems present instructions from the beginning to the end (Shneiderman et al., 2013; Iwata et al., 2010). In this case, inappropriate and appropriate paths may have common input items. Existing operation support systems do not consider this fact. Hence, users must reenter data when a mistake occurs.

Inputting common data is a usability issue. Some studies aim to reduce user's effort by reusing data (Constantine and Lockwood, 1999) and minimizing number of actions (Seffah et al., 2006). Thus, it is desirable to reuse data input in an inappropriate path in the appropriate path automatically.

We propose a method to generate an operation support system that reuse a user's input data in an inappropriate operation path as much as possible by navigating a user to the appropriate operation path. Specifically, our system has two functions: an input reuse function to reuse previously input data for other input items whose labels are similar to those of the previously input data and a presentation operation procedure function to highlight the procedure from the current operation step to the appropriate operation step.

These two functions reduce a user's operation burden by limiting the amount of data that has to be reentered. The input reuse function employs a user's previously input data in an inappropriate operation

Nakamura, S., Iwata, H., Shirogane, J. and Fukazawa, Y.
Automatic Generation of an Operation Procedure Presentation System Reusing User's Input Data.
DOI: 10.5220/0006628001240131
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 2: HUCAPP*, pages 124-131
ISBN: 978-989-758-288-2

step, while the presentation operation procedure function demonstrates how to return to an appropriate operation path.

This paper is composed of eight sections. Section 2 describes related works. Section 3 highlights the contributions of this research. Section 4 overviews the functions of the proposed operation support system. Section 5 details the procedures to generate the operation support system. Section 6 evaluates the proposed system. Section 7 provides a discussion, and Section 8 summarizes this study.

2 RELATED WORK

Numerous types of operation support systems have been proposed for user operation assistance. Below we describe related work.

Some studies on input prediction provide automatic complements based on the user's input situation (James and Reischel, 2001). Input prediction is not performed without inputting characters. Our proposed operation support system uses similarilities between input items to automatically reuse appropriate input items. Thus, a user does not have to reenter data.

Some systems research focuses on active support by sensing user's mistakes and inefficient operations (Breuker et al., 1987). However, users are unlikely to accept an active operation support system because it is possible that users receive undesired support (Shneiderman et al., 2013). Users tend to prefer control of the user's side rather than control of the system side (Shneiderman et al., 2013). In our proposed operation support system, a user initiates the support system. Consequently, undesired support does not occur.

In addition, Parmit et al. has proposed a help system called *LemonAid* (Chilana et al., 2013). This is integrated into the target application. A user selects the interface element of the application, and gets related question and answer. All questions and answers are from actual users. The provided help information depends on the users. Hence, in applications with limited users, it is highly probable that the amount of help information is small. Since our proposed operation support system covers all the functions available in the target application, the help information supports all areas.

3 CONTRIBUTIONS

Our system's contributions are described below.

3.1 Input Reuse

If a user proceeds from an inappropriate operation path to an appropriate operation path, the input data in the inappropriate path should be reused in the appropriate path. In our system, previously input data can be reused for other input items with similar label names. Additionally, a user can set the degree of reuse for the input data. The degree indicates the similarity level of label names of the input items. In response to the degree, accurate reuse or wide reuse is possible.

Fig. 1 shows an example of a window transition in an application. Nodes represent windows and edges represent window switching. WindowA is the first window of this application (the application is initiated from this window). In WindowB, a user inputs some data and the operation path is branched into WindowD and F by the selection in WindowC.

Consider the case where a user's goal is the operation path to WindowE, and the current step is WindowG. In addition, some of input items in WindowF are common to WindowD. In this case, the input data in WindowF are reused for the common input items in WindowD. Due to this, a user's input operation in WindowF can be reduced.

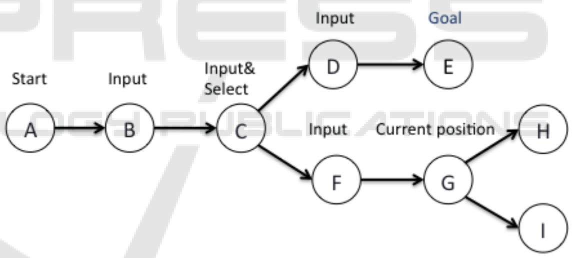

Figure 1: Window transitions of an application.

3.2 Instructions Considering User's Context

Many conventional operation support systems only present instructions from the beginning to the end of a function. When both inappropriate and appropriate operation paths contain common input items, existing operation support systems do not consider these common items. Consequently, a user must reenter the input item in the appropriate operation path. Our proposed method guides a user from the current step to the originally intended operation step. In this way, proceeding back to the intended operation, a user does not have to repeat operations that are part of the target function.

Considering the example in Fig. 1, the common parts of the path from WindowA to the goal

(WindowE) and the path from WindowA to current position (WindowG) (flow of the window transitions) are WindowA, WindowB, and WindowC. Thus, when a user starts from WindowA, again, the inputs in WindowB and WindowC are discarded. To achieve goal (WindowE), the same inputs must be reentered in WindowB or WindowC. Returning the procedure from current position (WindowG) to branch point (WindowC) to realize goal (WindowE), the data does not need to be reentered in WindowB.

3.3 Learning Effect

Our operation support system can produce a learning effect when a user make mistake. Thus, the next user operation should be smoother. Our operation support system provides instructions to return to the previous steps. Thus, a user can confirm why a mistake occurred. In a conventional operation support, a user may not understand why a mistake was made because the instructions always start at the beginning. Humans can learn a lot by failing (Sitkin, 1992). Our operation support system realizes support to learn from failure.

4 FUNCTIONS OF THE OPERATION SUPPORT SYSTEM

4.1 Usage of the Operation Support System

This section describes a user's view and internal processes of both the input reuse function and the operation procedure presentation function in the present system.

4.1.1 Example

Fig. 2 shows the window transitions of the fictional application in Fig. 1 using an activity diagram. The activities are all window displays. The example here uses the same situation described in Section 3.1, where the current step is WindowG, but the goal is WindowE. In this application, three functions are prepared. The flows of these functions are blue arrows.

In this example, a user is in the middle of the procedure for Function 2 or 3, but desires to execute Function 1. The red value shows the degree of coincidence among the input items prepared for each function application. The degree of coincidence is obtained by calculating the similarity between ontological concepts. The calculation of the similarity is described in Section 4.2.2. It is assumed that input item 1 is in WindowF, input item 2 is in WindowD, and input item 3 is in WindowE.

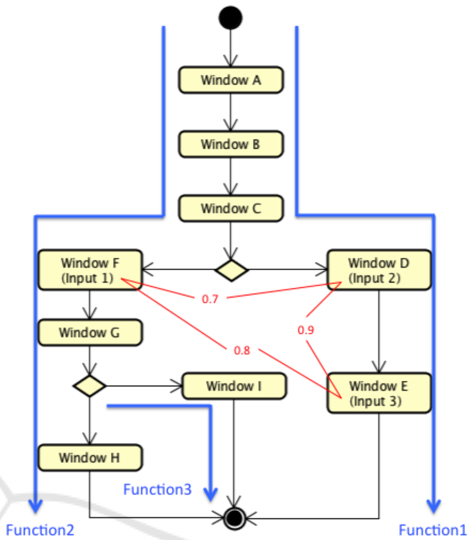

Figure 2: Activity diagram of the application, flows of the three functions (blue arrows), and the degree of coincidence between each input item (red values).

4.1.2 Usage of the Input Reuse Function

Initiating the application also starts the operation support system. A user must specify the threshold value to determine whether to reuse the input data.

The threshold value, which ranges from 0 to 1, represents the similarity between label names. Because the meaning of each input item is transmitted to a user by the label name, the input item is associated with the label name. As a result, the similarity of each label name is the similarity of the corresponding input item. The closer the threshold value is to 1, the higher the similarity of the reusable input data. However, the range where reuse is performed becomes narrower. On the contrary, the closer the threshold value is to 0, input data with a low similarity is reused. However, the range of reuse is wide. When a user transitions to a window that includes an input item with a label name similar to input data entered by a user in a previous path and a user has specified the threshold value, reuse of a similar input is automatically performed based on the threshold value.

As an example, consider the case in Section 4.1.1. For a user to use Function 1, it is necessary to return to WindowC, proceed to WindowE, and to enter input items 2 and 3 in the respective windows. Since input item 1 is already entered on WindowF, it is possible to reuse input items 2 and 3. Table 1 shows the value of each input item when the threshold value is set to 0.6 or 0.8.

Table 1: Results when the threshold is 0.6 or 0.8.

	Input data (threshold is 0.6)	Input data (threshold is 0.8)
Input item1	Inputted before using input reuse	Inputted before using input reuse
Input item2	Data of input item1 is reused	Input reuse doe's not occur
Input item3	Data of input item2 is reused	Data of input item2 is reused

4.1.3 Usage of the Operation Procedure Presentation Function

When a user wants to know the operation procedure to achieve user's goal, he or she accesses the operation support system, which is simultaneously activated with the application. The operation support system presents the function list available for a given application. Furthermore, searching in the operation support system can narrow the goals to those related to keywords. When a user selects a function from the function list, instructions from the current step to the goal of the function (information on how to use the function) are presented. These instructions are given as a bulleted list. The operations are performed in order beginning from the top. Each bullet shows operations in a window.

4.2 Mechanism of the Input Reuse

This section details the internal process of the input reuse function. The input reuse function realizes reuse of a user's previously input data for the input items of other windows. Therefore, the following processes are required to realize input reuse:

(1) Retain input data

(2) Reuse input data

4.2.1 Retain Input Data

When saving the input data, the operation support system associates input data with a label name. The association mechanism requires that the developer associate input items with label names. This process is discussed in Section 5.2.

As an example, suppose that there is an input item "Last Name" and a user enters and commits "Nakamura". The input reuse function retains the label name "Last Name" and the value "Nakamura" as a set like "Last Name = Nakamura". When a user commits, the input data is probably most correct. Hence, our proposed method saves the input data when a user commits the input data.

4.2.2 Reuse Input Data

The input reuse function reuses the previously input data for appropriate input items in other windows. After a window transition, to judge whether the input data can be reused, ontology with basic concepts of labels for each input item is used. A basic concept is one determined only by its own properties and is independent of other concepts. After a window transition, how similar the label names of the input items included in the transition destination window to the stored input data is evaluated. If the degree of similarity exceeds the threshold set by a user, the input data can be reused automatically.

4.3 Mechanism of the Operation Procedure Presentation

The proposed operation procedure presentation function represents the procedure from a user's current step in a target application to a user's goal. Therefore, the operation procedure presentation function involves the following steps:

(1) Derive paths from the start to all goals

(2) Monitor the current step and retain the previous path

4.3.1 Derive Paths from the Start to Goals

When the operation support system is initiated (i.e., the application is activated), the paths from the start to all goals is initially derived based on the data obtained by analyzing the activity diagram. After deriving all paths, the operation support system retains the path information. The activity diagram in Fig. 2 indicates that there are the paths of functions.

4.3.2 Calculate the Path from the Current Step to the Goal

Our system monitors the current step and the window transition history. When our system senses a newly active window for each window transition, our system updates the current step and transition history. Based on them, the operation support system calculates the common path between the history of the window transitions and the path from start to the goal of the function selected. When confirming common paths, the following two patterns are conceivable:

(1) The current step exists in the path from the start to a user's goal.

If the current step exists in the path from the start to the goal, the current path is a part of the appropriate path. Therefore, the path presented to a user is the part of the appropriate path, which does not contain common parts. If the desired function is Function 2 in Fig. 2 and the current step is WindowG, the path presented to user is from WindowG

(2) The current step does not exist in the path from the start to the goal.

If the current step does not exist in the path from the start to the goal, a branch is created from the last window in the common path between the two paths. Therefore, the path from the current step to the goal is a combination of a back path from the current step to the branch step and the path from the branch step to the goal. Our system presents this connected path. When the intended function is Function 1 in Fig. 2, the path presented to user is one combined back path to WindowC and path from WindowC to Window E.

5 GENERATION PROCEDURE

In this section, we describe the generation procedure of our system. There are five steps:

(1) Create the ontology
(2) Associate input items with label names
(3) Describe the activity diagram
(4) Analyze the activity diagram
(5) Add the function name and instructions

The source programs to realize our system are generated based on these steps. Also, our system is implemented for web applications that work on the client side with HTML and JavaScript and can handle Java on the server side. We calculate the similarity using the ontology editor of Hozo (Hozo-Ontology Editor). However, if the ontology data can be handled and the similarity between the input items can be calculated, and if associating a label name with a text field is possible, the programming language and environment of the target application are not limited.

5.1 Create the Ontology

As mentioned above, we use an ontology in the input reuse function. Developers must create the ontologies beforehand. When creating an ontology, it is necessary to include all basic concepts with the label names of the input items. Setting the data type of each input item allows the similarity between input items to be calculated more accurately. If relationships can be made between input items, they should be related.

5.2 Associate Input Items with Label Names

As described in Section 4.2.1, for the input reuse function to retain the value entered in the original input item associated with the label name, the developer must create a correspondence between them beforehand. Because we assume that our system is incorporated into the target application, it is necessary to make correspondences within the source program of the target application so that the input reuse function of our system can understand them programmatically. In addition, it is necessary that the label name in this correspondence is the same as the label name of the basic concept used in the ontology. Through the unique data attribute of HTML, the input items are associated with label names in our system.

5.3 Create the Activity Diagram

The operation procedure presentation function analyzes the activity diagram to obtain information about the window transition of the target application. In addition to the basic components of the activity diagram, window object names, which are the names declared in the source programs of the application, are additionally described in the activity diagram. This additional description allows the activity added by window name to be regarded as an activity of the window display. Hence, the activity diagram can be treated as a window transition diagram.

Associating the window object name with the activity in the activity diagram indicates that the associated activity is a window display activity. Additionally, the associated window object name serves as an identifier of the activity. As a method of association, an asterisk "*" is added to the original description of the associated activity, such as "* Start_Window *".

5.4 Analyze the Activity Diagram

Based on the activity diagram, the information necessary to automatically generate the code of the operation support system is extracted. The following information can be extracted.

- Window object name

The window object name is described by surrounding it with "*" for the corresponding activity. Therefore, the original description and the window object name can be extracted separately.

- Path information

By narrowing to only the activity with the associated window object name, the activity diagram can be treated as a window transition diagram. From this activity diagram, all paths of the application are derived by a depth-first search.

5.5 Add the Function Name and Instruction

Our system requires the function names and contents of the instructions to be used. Therefore, all the path information obtained by analyzing the activity diagram is presented to the developer. Then the developer creates the following descriptions.

- Function Name

For all the path information obtained in the analysis in Section 5.4, the developer must describe the purpose of each path. This written description is used as a list of functions displayed when a user uses our system.

- Instructions

Further, the developer describes the operation procedure to be performed in each window in order to trace the path. The contents described here are used as instructions to be displayed when a user uses our system.

After describing these, operation support system is generated.

6 EVALUATION

In the evaluation, we adopt open-source software of an expense report submission system. This system is web-based software to create expense reports and has many input items. To use this system, administrator and general user roles were provided. Administrators could audit the expense reports, add user's information, etc. General users could create expense reports. This system was implemented by several types of programming languages, such as html, jsp, javascript, java, etc, and had 929 files and 166803 lines of code, totally.

6.1 Participants

8 (male) students were recruited from a local university to participate in the evaluation. All subjects were computer science majors, and their average age was 22.6 (21 to 24). Although the subjects were familiar with operating a personal computer, they were unfamiliar with the expense report submission system used in the evaluation. The subjects were divided into two groups. In Group A (4 students), the first task was performed using the proposed operation support system. Then the same task was performed using the operation manual attached to the expense report submission system. Group B (4 students) used the opposite task flow as Group A.

6.2 Procedure and Apparatus

Both groups were given overviews of the operation support system and the expense report submission system prior to performing the tasks. When completing the tasks using the operation support system, we also explained the threshold (0 to 1) for input reuse, and asked the subjects to enter the thresholds before performing the tasks. Because the subjects were unfamiliar with the expense report submission system, they were asked to assume the context, such as a part-time manager, and to perform tasks that require less business knowledge. The task flow was to log in as the administrator of the expense report submission system where departments, users, and email addresses (so that the image of the receipt can be sent via e-mail) can be added. After completing the second task, the participants answered the questionnaire. The question items of the questionnaire were as follows:

Q1. Is the operation procedure presented properly?
Q2. Which is more useful, the conventional operation support or the operation support that presents procedures from the current step?
Q3. What is the reason for your answer in Q2?
Q4. In what situations do you think that support for presentation operation procedures is useful?
Q5. Is reuse necessary for each input field?
Q6. Is appropriate data reused?
Q7. Is setting the threshold appropriate?
Q8. Are the reuse items appropriate?
Q9. What input items did you want to reuse?
Q10. Is input items inappropriately reused?
Q11. Do you have additional comments?

For Q1, Q5 to Q7, we adopted a five-point Likert-scale, where 1 equals "Strongly disagree" and 5 equals "Strongly agree". Q2 also adopted a five-point Likert-scale, where 1 equals "Strongly think conventional operation support" and 5 equals "Strongly think operation support system by

Table 2: The relationship between the threshold set by the subjects and the input reuse.

Threshold	Number of inputs	Number of reuses	Number of inappropriate reuses	Number of appropriate reuses	Number of times that reuse does not occur
0.2	17	9	6	3	1
0.3	21	11	5	6	0
0.3	20	11	6	5	0
0.4	13	7	4	3	0
0.5	21	4	2	2	0
0.5	21	4	2	2	3
0.6	21	3	2	1	4
0.7	17	3	2	1	4

presentation of operation procedure from the current step”. Other questions were free response.

The instructions presented in the operation procedure presentation function of the operation support system were described in English and include contents that equivalent to the operation manual provided in the target application to prevent significant differences due to the language and content.

The evaluation experiment was conducted using Apple’s Macbook Air (Mac OS 10.12.3) and a Firefox browser (54.0.1).

6.3 Results

Each subject completed the questionnaire after the experiment. Fig. 3 shows the results for Q5–Q7, which focus on input reuse. The x-axis is the threshold value selected by each user. A threshold value of 0.2 provided the lowest results for Q5–Q7. For Q6, the highest result was obtained when the threshold value was 0.3 and 0.4. For Q7, a threshold value of 0.4 and 0.7 gave the highest result.

Table 2 shows the relationship between the threshold set by the subjects and the input reuse. The number of input items differed between subjects because the input items of the tasks included unnecessary input items. According to Table 2, as the threshold value became higher, both the number of reused input items and the number of inappropriate reuses decreased. On the other hand, as the threshold value became lower, the overall reuse and the number of appropriate reuses decreased. However, the number of desired reuses that did not occur tended to increase.

According to the results of Q1 and Q2, operation support from the current step was more useful than the conventional operation support.

Some subjects did not refer to the operation support. They answered that they were neutral between the proposed and conventional support systems.

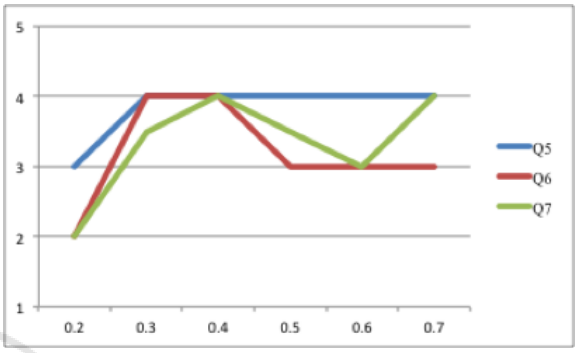

Figure 3: Results of Q5 to Q7 regarding input reuse.

7 DISSCUSION

Based on the evaluation results in Section 6, this section discusses operation support by presentation of operation procedures and input reuse.

When the threshold value is 0.6, the results of Q6 and Q7 are low (Fig. 3). Although the threshold value is subjective, it is necessary to examine criteria of threshold in order to prevent inappropriate reuse because it may induce stress in users.

Although the same threshold value and the same number of reuses are used, the number of inappropriate reuse items and the appropriate number of reuse items differ by subject (Table 2). This is because the appropriateness of reuse may depend on the context. In this evaluation, some subjects used the reused data, while others deleted the reused data and entered different data. These results demonstrate that it is necessary to verify the reuse according to the context. In this evaluation, reuse was forcibly performed without consideration of the user’s context. In the future, a method to automatically sense the user’s context or one that allows users to set the context of the operation support system will be studied.

As the threshold value rises, the number of items that the subjects wished to reuse also increases. This is because the similarity between the ontology concepts created in this evaluation is generally low.

Thus, the threshold value is set higher than the similarity, and reuse is not performed. In this evaluation, only data types were set in some concepts. If the setting threshold is low, reuse may occur between input items with the same data type although they are not similar. To measure the similarity more precisely, it is necessary to define the concept more strictly within the ontology.

With respect to operation support by presentation of operation procedure, negative results were not reported. However, none of the subjects indicated that the presentation of the operation procedure is appropriate or more useful than the conventional one. This is because the window transition hierarchy of the target application in this evaluation is wide and shallow. Hence, it seems that the support by presenting the operation procedure is not perceived as very useful. One comment indicated that it would be more useful for a more complicated procedure.

In addition, 75% subjects in Group A and 50% subjects in Group B performed the first task again, because they couldn’t understand what data should be input. Instructions by our system were similar to the manual of the application. Thus, this did not always mean that our system was more effective than the manual. Not only operation procedures but meanings of input items could be difficult for users, so supporting users to understand the meanings of input items should be considered.

Meanwhile, some tasks generally can’t be cancelled, such as file saving and printing. In this experiment, the task was data registration and could not be cancelled just by return operations. In these cases, it is necessary to present completion of operations (impossible return operations) or how to cancel the operations.

Thus, although there were some issues that should be considered, we confirmed that our system could supported user operations effectively.

8 CONCLUSIONS

Herein we propose automatic generation of an operation procedure presentation system that reuses user’s previously input data. The proposed operation support system contains operation support that presents the procedure from the current step to the end step of the intended function, and reuses the previously input data entered in an inappropriate operation path by a user in the appropriate operation path. The input reuse function of this system is generated by creating an ontology and associating a label name and an input item. The operation procedure presentation function is generated based on the activity diagram describing additional information. Future tasks of this research include:

- Improving our system
- Reconsidering target applications for evaluation

REFERENCES

Breuker, J., Winkels, R., and Sandberg, J. 1987. Coaching strategies for help systems: EUROHELP. *Proceedings of the 4th Annual ESPRIT Conference*.

Chilana, P.K., Ko, A.J., Wobbrock, J.O., and Grossman, T. 2013. A multi-site field study of crowdsourced contextual help: usage and perspectives of end users and software teams. *Proceedings of Human Factors in Computing Systems*, ACM.

Constantine L., and Lockwood L. 1999. Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. Boston: Addison-Wesley Professional.

Expense Submital System, https://sourceforge.net/projects/expense-ss/ (accessed on 12 September, 2017).

Hozo-Ontology Editor, http://www.hozo.jp/hozo/ (accessed on 12 September, 2017).

Iwata, H., Fukazawa, Y., and Shirogane, J. 2010. Generation of an operation learning support system by log analysis. *Proceedings of 2nd International Conference on Software Engineering and Data Mining*. IEEE.

James, C. L., and Reischel, K. M. 2001. Text input for mobile devices: comparing model prediction to actual performance. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM.

Kaiya, H., and Sekii, M. 2006. Using Domain Ontology as Domain Knowledge for Requirements Elicitation. *Proceedings of the 14th IEEE International Conference on Requirements Engineering.* IEEE.

Seffah, A., Donyaee, M., Kline, R.B., and Padda, H.K. 2006. Usability measurement and metrics: A consolidated model. *Software Quality Journal. 14(2)*, pp. 159-178.

Shneiderman,B., Plaisant,C., Cohen,M., and Jacobs,S., 2013. Designing the User Interface: Strategies for Effective Human-Computer Interaction. Pearson.

Sitkin,S.B. 1992. Learning Through Failuer: The Strategy Of Small Losses. Research in organizational behavior, 14, pp. 231-266.