

Measurable Interactive Application to Find Out User Recognition and Strategy when Problem Solving

Taku Yamaguchi^{1*}, Michiko Oba²

¹ Independent Researcher

² Future University Hakodate, 116-2 Kamedanakano-cho, Hakodate, Hokkaido, Japan 041-8655

* Corresponding author. email: study.yamahige@gmail.com

Manuscript submitted November 7, 2019; accepted January 18, 2020.

doi: 10.17706/jsw.15.1.12-22

Abstract: This paper describes how to determine user recognition or problem-solving strategies by measuring and analyzing their interactions with computer applications. In recording interactions, each action should be associated with “a term” that is used to describe user recognition or strategy. Recorded actions can be analyzed through their temporal co-occurrence. When two actions occur a significant number of times, it indicates that there is a close relationship between those actions for the user. That relationship can be described with the terms associated with those actions and can be interpreted as the user's recognition or strategies. As a proof of concept, we evaluated interactions in rearranging puzzles, i.e., Jigsaw Text and Jigsaw Code.

Key words: measurement and analysis of the problem-solving process, problem-solving strategy, temporal co-occurrence, thinking process

1. Introduction

Fostering the ability to think is an important aspect of education. The Central Council for Education reported that “cognitive capabilities, including critical and rational thinking, to identify solutions to 'unsolvable' issues” is “required in a matured society” [1]. The question we ask here is how can we detect and evaluate the process to solve “unsolvable” problems. It is reasonable to assume that the solution involves trial-and-error; however, that may not be the shortest route to the answer.

Recent studies analyze people's human-computer interaction (HCI) activities to evaluate the usability of the system or to identify the user [2]. When we extract some features from the activity records, we want to describe them in a natural language that people can understand and let the people use them as a strategy.

Other studies have attempted to clarify the process of thinking. In studies that measure and analyze keystrokes or pen-strokes, the degree of proficiency in writing can be clarified; however, the decisions and strategies that the writers made cannot be determined [3]-[4]. Approaches that use eye-tracking can reveal that, when reading Mind Maps, experts browse the main stems first while novices thoroughly scan entire branches attached to every stem [6].

In programming exercises, some strategies are observed by watching video recordings of students solving Parson's problems, such as looking for variable declarations [6]. Parson's problems are puzzles that require the players to rearrange randomized lines of program code in the correct order. Trial-and-error strategies were also observed. Typically, such strategies are employed when the player is struggling with the puzzle. A way to evaluate the degree to which the player's operations conform to the desired strategy to solve the

puzzle has been proposed; however, it loses the trial-and-error activities [8].

In another study that uses a system called “Card operation-based programming learning system,” which is similar to Parson's Problem, student operations are analyzed using the time series of the “distance” of the current order compared to the correct order [9]. That study observed that a student who gets close to the correct order sometimes then deviates from the correct order. However, the data analysis alone did not show the student's strategy to close to and deviate from the correct order.

It is essential to consider that problems may be unsolvable, and the strategy includes trial-and-error. A method that is used to measure and analyze student operations requires equipment that is common, scalable, and easy to apply.

2. Method

Watching games, such as board games, card games, and baseball, we can determine the player's strategy based on their actions/operations in that game. Therefore, if an interactive computer application that assists the user in solving problems is designed to act close to the user's thoughts and judgments, that thought can be inferred by measuring and analyzing the user operations to the objects on that application. In this paper, we refer to such an application as a “puzzle,” a user of the application is a “player,” the object, such as “line of code” or “sentence,” are a “piece” of the puzzle, and to solve the puzzle is to “play.”

In this paper, we take rearranging puzzles as examples of such puzzles. When a game starts, pieces are placed in random order. Players move the pieces so that they are arranged in the correct order. The puzzles are web applications that can be played using browsers on PCs and smartphones.

2.1. Puzzle Applications

We propose to design an application such that the objects that make up the user interface (UI) are associated with terms that are used to describe the user's recognition or strategies.

When objects are “pages” (page 1, page 2, and so on), it may be found that “Players tend to read around page 2 first and then move to around page 4” as a result of an analysis. Based on the page-turning frequencies, we may evaluate the players' engagement with a class. When objects are related to “introduction,” “methods,” “results,” etc. it may be found that “Players tend to read methods first, then move to results and exit.” Based on the movement related to the organizational structure, it may be found that the players' strategy is to grasp articles concisely.

2.1.1. Topic writer

Topic Writer is a text editor that guides a player to write along a simple logic tree (Fig. 1). Logic, i.e., the text's structure, is presented in one or more axes of a matrix that we call a “worksheet.” A player edits cells in the matrix. Players in a class may produce different texts; however, they use the same worksheet. Thus, we may find “They tend to edit introduction and discussion alternately” even if we do not have the produced texts.

Topic Writer has been used in technical/academic writing and other classes for eight semesters in Future University Hakodate, etc.

Glossary for specific readers 4 paragraphs 4 sentences 163 characters	
Readers	ライティングの教師や教育分野の研究者
Characteristics of the readers	(click to edit)
Concept: description of what it is, in one sentence	ジグソー・テキストは、文章の並べ替え作文アプリである。
Description in detail	文章を分割してランダムに並べられたピースを、並べ替え 操作 完成させる文章のジグソー・パズルである。並べ替え操作を記録していて、
Benefits or the usage	Webアプリケーションで、スマホで手軽に操作できる。記録したデータからプレイヤーの操作の傾向や考え方を分析できる。

Fig. 1. Topic Writer: Translated from the original Japanese version.

2.1.2. Jigsaw text

In Jigsaw Text, the pieces are fragments of a text. Players rearrange the fragments using a drag & drop function, to create correctly ordered text (Fig. 2). In Fig. 3, Jigsaw Text has two areas: select a piece from the left area and move it into the right area, or delete a piece from the right area.

Jigsaw Text is used in technical/academic writing classes and logical thinking classes.

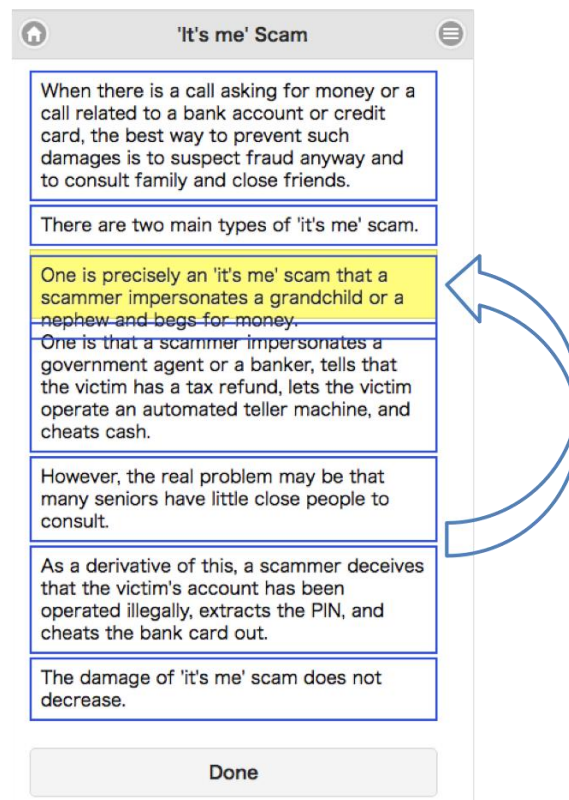


Fig. 2. Jigsaw Text puzzle “‘It's me' Scam” translated from the original Japanese version “オレオレ詐欺”

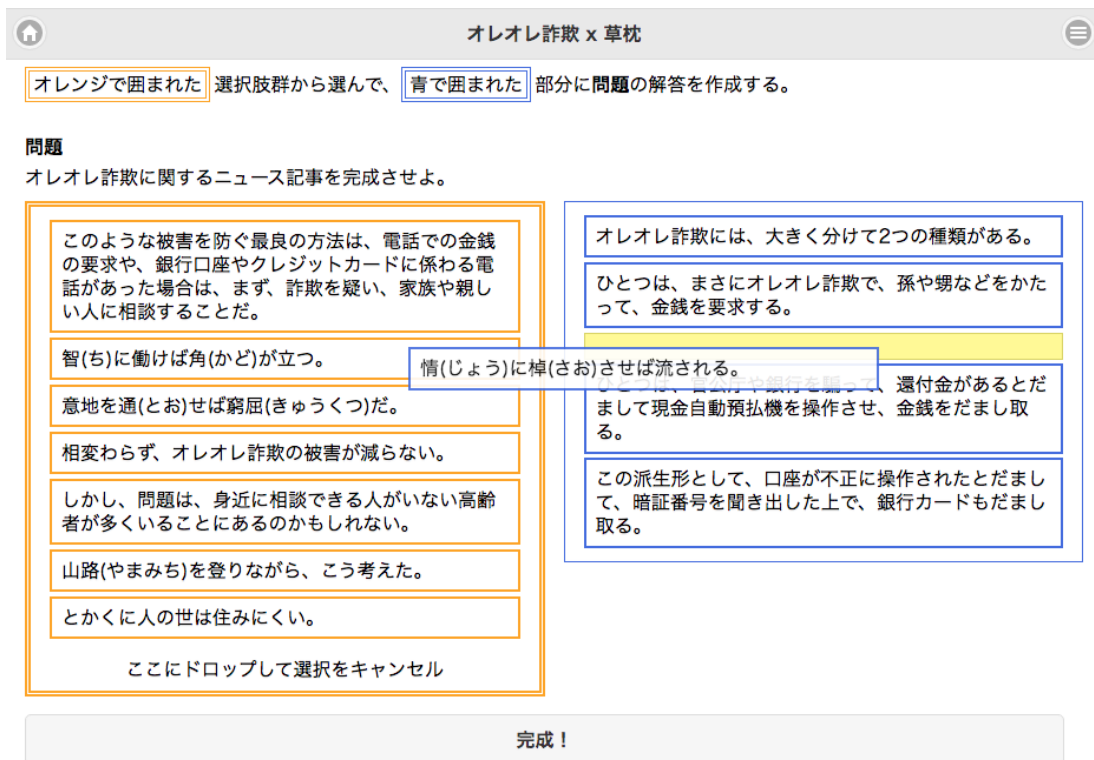


Fig. 3. Jigsaw Text with two areas: select and reorder.

2.1.3. Jigsaw code

In Jigsaw Code, the pieces are the lines of code of a computer program (Fig. 4, Fig. 5). Players rearrange and create differently ordered pieces; however, they use the same pieces.

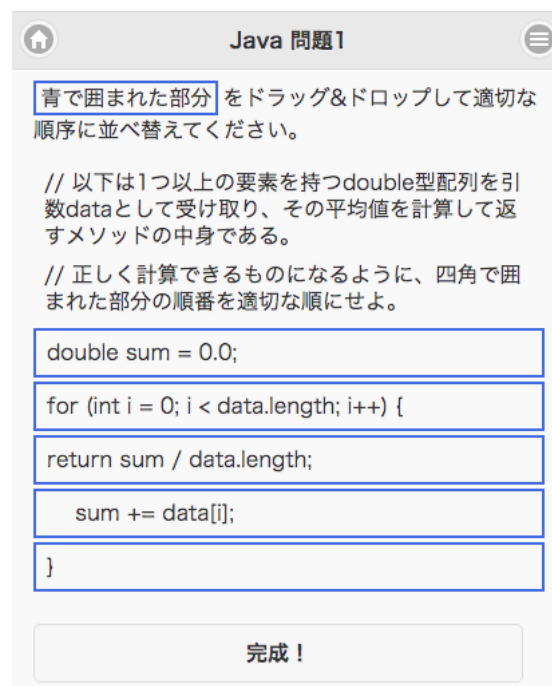


Fig. 4. Jigsaw code.

[Java問題]5-3 直方体

オレンジで囲まれた 選択肢群から選んで、青で囲まれた 部分に問題の解答を作成する。

問題
 以下はx, y, zに直方体のそれぞれ辺の長さを持ち、volumeに体積の値を持つRectangularクラスです。
 適切なコードを適切な順に並べ、インスタンス生成時に各変数の初期値を設定するコンストラクタを完成させてください。
 ただし、宣言された変数の順番(x, y, z, volume)の通りに並べるとします。
 参考：直方体の体積 = 底面の面積 * 高さ

this.volume = volume;

return volume;

public Rectangular(double x, double y, double z) {

public Rectangular() {

ここにドロップして選択をキャンセル

```
public class Rectangular {
    double x;
    double y;
    double z;
    double volume;

    this.y = y;
    }
    this.z = z;
    this.x = x;
    this.volume = this.x * this.y * this.z;
}
```

完成！

Fig. 5. Jigsaw Code with two areas.

2.1.4. Jigsaw event

Jigsaw Event is a chronological puzzle where the player must order past events (Fig. 6) or tasks to be performed (Fig. 7).

ジグソーできごと

ドラッグ&ドロップして適切な順序に並べ替えてください。

社会連携センターがスタート

函館新外環 函館IC-赤川IC間が開通

公立はこだて未来大学開学

函館市国際水産・海洋総合研究センター開設

「プロジェクト」学習スタート

完成！

Fig. 6. Jigsaw events puzzle “history of future university Hakodate”.

Puzzles like Fig. 7 are designed not only to ask the correct answer but also to determine what is important for players from the order in which pieces are moved.

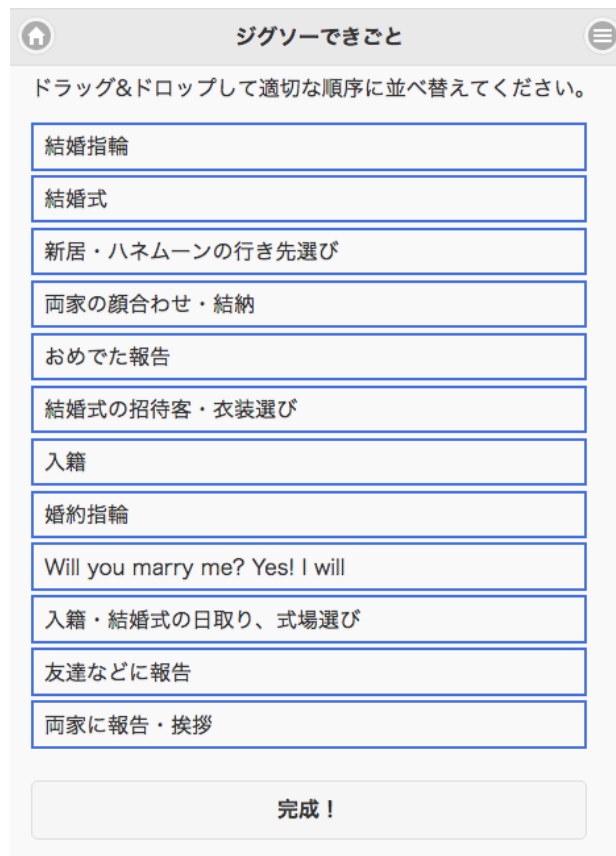


Fig. 7. Jigsaw events puzzle: “Wedding procedure”.

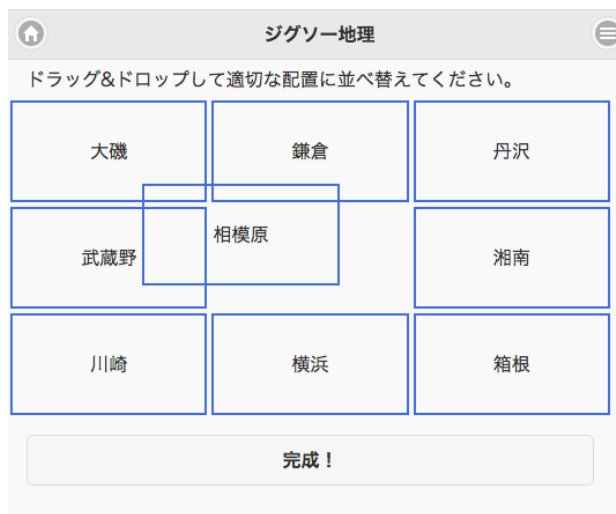


Fig. 8. Jigsaw geography puzzle “location of Shounan”.

2.1.5. Jigsaw geography

Jigsaw geography is a two-dimensional rearranging puzzle that involves a spatial arrangement of buildings, cities, and areas. It does not necessarily require players to determine the exact position. It is also designed to find the mental model of those objects by being able to freely arrange from the latitude and longitude. Fig. 8 is a puzzle that asks the range and position of a famous seashore area 湘南(Shounan). Its range and position vary from person to person. In Fig. 9, the pieces are places, such as the place of birth, school, port, and mountain. It is expected that regional landmarks will be found through the patterns of the

player's operations.

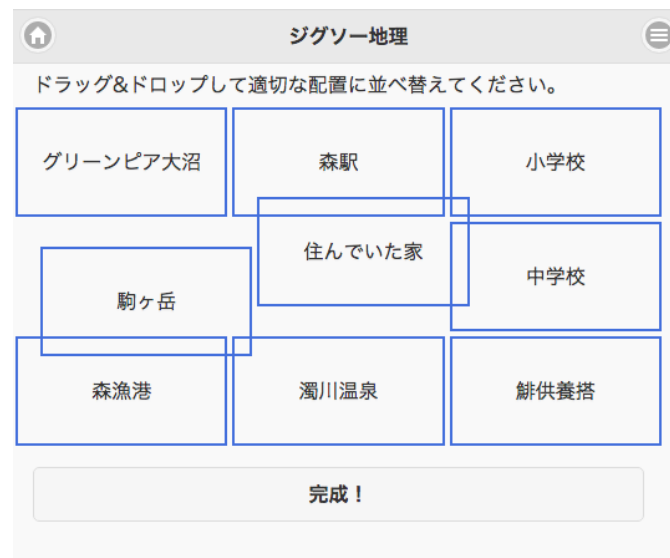


Fig. 9. Jigsaw geography puzzle “Your hometown”.

2.2. Measurement

We propose recording all user operations with “context,” which, as well as UI objects, are related to a term used to describe user activity [10]-[16].

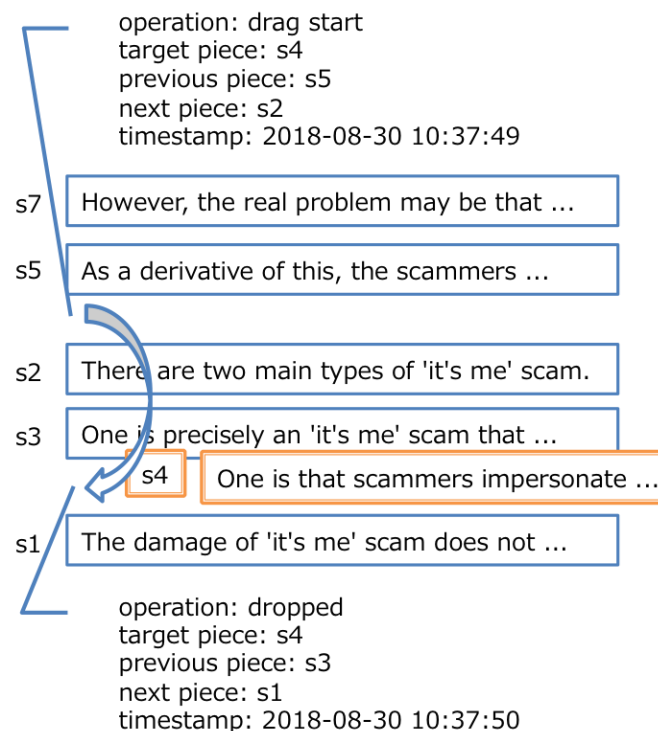


Fig. 10. Measuring solving operations for Jigsaw text.

In Jigsaw Text, each operation is recorded with the IDs of the objects that are placed before and after the dragged/dropped object as well as the object ID and time stamp. In Fig. 10, “s4” is the ID of the object that is dragged and dropped. “s4” is dragged from between “s5” and “s2” and is dropped into between “s3” and

“s1.”

2.3. Analysis: Temporal Co-occurrence

We propose finding temporal/sequential co-occurrence patterns in user operations. When two operations frequently occur at close to the same time, it indicates that, for the user, there is some relationship between those operations [10]-[16].

n \ n+1	s1	s2	s3	s4	s5	s6	s7
s1	4	3	5	1	1	3	0
s2	2	3	3	3	0	1	2
s3	1	0	1	6	1	2	2
s4	1	1	2	3	6	4	1
s5	0	0	1	1	2	4	1
s6	2	2	1	4	3	3	5
s7	2	0	2	1	1	3	1

Fig. 11. Temporal co-occurrence matrix for Jigsaw Text (n = 15).

Co-occurrence among the pieces is arranged in a matrix. The occurrence frequency can be evaluated by the mean (μ) and standard deviation (σ). Fig. 11 shows an example of a co-occurrence matrix. In Fig. 11, the “s4” operation occurred after the “s3” operation 6 times, and 6 times is greater than $\mu+2\sigma$.

3. Proof of Concept

We present the results of a proof of concept for Jigsaw Code and Jigsaw Text.

3.1. Jigsaw Code

Thirty-two undergraduate and master's students solved the puzzle in Fig. 4 [17]. Fig. 12 shows the Jigsaw Code puzzle presented in Fig. 4 with the IDs of the lines of codes. The co-occurrence matrix for the puzzle is shown in Fig. 13. Here, we exclude the diagonal cells because they do not correspond to the relation of different pieces. The average value (μ) of the cells is 3.4, and the standard deviation value (σ) is 2.7. Here, “s6” is dragged after “s4” 10 times, and “s3” is dragged after “s6” 9 times. Both cases occurred more times than $\mu+2\sigma = 8.8$. The number of co-occurrence is significantly large (2σ). This indicates that there is some relationship between corresponding lines of codes. “s4” is the start of a for-loop and includes the opening brace, and “s6” is the end of the for-loop and includes the closing brace. The co-occurrence indicates that players tend to arrange the for-loop skeleton before the looped content

Fig. 14 shows the times and percentages of lines that were moved first in each play. Line “s4” was dragged most frequently.

In Jigsaw Code, when two lines of code are dragged frequently at close to the same time, it indicates that, for the user, there is some relationship between those lines. Fig. 13 and Fig. 14 suggest that players tend to focus on for-loops and to arrange the for-loop skeleton first. We found their strategy without watching video recordings.

3.2. Jigsaw Text

Fifteen master's students solved the puzzle in Fig. 2 in a technical writing class for two semesters. Fig. 11 shows the co-occurrence matrix of their plays. In the matrix, excluding diagonal cell values, “s4” is

frequently (2σ) dragged after “s3,” and “s5” is frequently (2σ) dragged after “s4.”

s1	// 以下は1つ以上の要素を持つdouble 型配列を引数dataとして受け取り、そ の平均値を計算して返すメソッドの中身 である。
s2	// 正しく計算できるものになるよう に、四角で囲まれた部分の順番を適切な 順にせよ。
s3	double sum = 0.0;
s4	for (int i = 0; i < data.length; i++) {
s5	sum += data[i];
s6	}
s7	return sum / data.length;

Fig. 12. IDs of each line of code in Jigsaw code.

n \ n+1	s3	s4	s5	s6	s7
s3	7	1	4	1	5
s4	4	5	3	10	5
s5	2	1	3	0	3
s6	9	0	2	4	4
s7	5	3	1	4	3

Fig. 13. Co-occurrence matrix for Jigsaw code ($n = 32$).

ID	times	percentage	
s3	4	12%	##
s4	18	56%	#####
s5	1	3%	#
s6	3	9%	##
s7	6	18%	####
合計	32		

Fig. 14. Percentage of lines moved first.

In Jigsaw Text, when two text pieces are dragged frequently at close to the same time, it indicates that, for the user, there is some relationship between those text pieces. Fig. 11 suggests that there are relationships between “s3 One is exactly...” and “s4 One is that...,” and between “s4 One is that...” and “s5 As a derivative...”

It follows from this is that players tend to focus on the somewhat syntactic relation of the three pieces: “s2 There are two...,” “s3 One is exactly...,” and “s4 One is that...,” This is similar to solving a jigsaw puzzle using the pieces' shapes as a clue rather than the pictures on the pieces.

4. Conclusion

We have proposed (a) to design an interactive application such that the UI objects are related to terms that are used to describe user's strategy to solve the problem, (b) to record user operations through those UI objects, and (c) to evaluate the temporal co-occurrence of the operations. In addition, instead of watching videos, we were able to employ data analysis to describe the strategy used to solve Jigsaw Code and Jigsaw Text puzzles regardless of whether the answers were correct.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 17K01085.

We would like to thank Tatsuo Kobayashi for the puzzles of Jigsaw Text. We also would like to thank Shigeiko Takahashi for the worksheets of Topic Writer. Finally, we would like to thank Hiromasa Kawakita at Future University Hakodate for sharing his data with us.

References

- [1] Central Council for Education. (2012). *Towards a Qualitative Transformation of University Education for Building a New Future - Universities Fostering Lifelong Learning and the Ability to Think Independently and Proactively (Report)*, Ministry of Education, Culture, Sports, Science and Technology. website: <http://www.mext.go.jp/en/publication/report/title01/detail01/1380275.htm>. website: http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/1325047.htm.
- [2] Zhang, T., Sun X., Chai Y., & Aghajan H. (2014). Human computer interaction activity-based user identification. *International Journal of Machine Learning and Computing*, 4(4), 354-358
- [3] Leijten, M., & Van, W. L. (2013). Keystroke logging in writing research: Using Inputlog to analyze and visualize writing processes. *Written Communication (WC)*, 30, 3, 358-392.
- [4] Tanaka, H., & Ishiguro, K. (2018). *The Writing Revision Process by Learners of Japanese: An Analysis of Position and Type of Revision by Chinese and Korean Learners*. NINJAL Research Papers, 14, 255-274.
- [5] Iiyama, M., Nakatsuka, C., Morimura, Y., Hashimoto, A., Murakami, M., & Minoh, M. (2017). Detecting answer stuck point using time intervals of pen strokes. *Transactions of Japanese Society for Information and Systems in Education*, 34(2), 166-171.
- [6] Yamauchi, H., Kobayashi, S., & Okanoya, K. (2012). Differences in eye movements between experienced and novice readers of a diagram-based notetaking method. *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society (JCSS)*, 19(4), 418-433.
- [7] Fabric, G., Mitrovic, A., & Neshatian, K. (2016). Investigating strategies used by novice and expert users to solve Parsons problems in a mobile Python tutor. *Proceedings of the 24th International Conference on Computers in Education (ICCE 2016)*.
- [8] Kumar, A. N. (2019). Representing and evaluating strategies for solving Parsons puzzles. *International Conference on Intelligent Tutoring Systems (ITS2019)*.
- [9] Morinaga, S., Matsumoto, S., Murakami, R., Hayashi, Y., & Tsukasa, H. (2019). Proposal of visualization method for card operation-based programming learning support system. *SIG-ALST, The Japanese Society for Artificial Intelligence (JSAI), SIG on Advanced Learning Science and Technology*.
- [10] Yamaguchi, M., & Oba, M. (2018). Measurement and analysis of reordering operations for events, procedures, codes and geographical objects. *Summer Symposium in front of Shiranui-sea (SSS) 2018*.
- [11] Yamaguchi, T., Kobayashi, T., Takahashi, S., & Oba, M. (2018). Inferring thinking: Measurement and analysis of puzzle solving operations. *Proceedings of the the 35th Annual Meeting of the Japanese Cognitive Science Society*.
- [12] Yamaguchi, T., Ito, K., & Oba, M. (2018). Measurement and analysis of programming puzzle solving operations. *SIG Technical Report*, Information Processing Society of Japan (IPSJ).
- [13] Oba, M., Yamaguchi, T., Takahashi, S., Kobayashi, T., & Fujiwara, R. (2018). Measurement and analysis of writhing and reviewing effects. *SIG Technical Report*, 2018-CE-144, 28, 1-7, Information Processing Society of Japan (IPSJ).
- [14] Oba, M., Yamaguchi, T., Takahashi, S., and Kobayashi, T. (2017). Analysis of relationship between text editing process and evaluation of written text in logical writing. *SIG Technical Report*, 2017-CE-141, 10, 1-9, Information Processing Society of Japan (IPSJ).

- [15] Yamaguchi, T., Oba M., Fujiwara, R., Takahashi, S., and Kobayashi, T. (2018). Reading and writing activity measurement and analysis. *SIG Technical Report*, 2018-CLE-24, 24, 1-8, Information Processing Society of Japan (IPSJ).
- [16] Yamaguchi, T., Oba, M., Takahashi, S., & Kobayashi, T. (2017). Reading and writing activity measurement and analysis by Jigsaw Text. *SIG Technical Report*, 2017-CE-142, 27, 1-6, Information Processing Society of Japan (IPSJ).
- [17] Oba, M., Yamaguchi, T., & Kawakita, H. (2019). Analysis of programming thinking process using puzzle. *Summer Symposium in Sasara (SSS) 2019*, 152-159, Information Processing Society of Japan (IPSJ).

Taku Yamaguchi obtained his M.S. in physics from Chiba University, Japan, in 1987.

He worked on document management in the Software Division of Hitachi Ltd., also worked on document editing and task management in Justsystems. He has been studying writing and reading processes. He has been a part-time lecturer in Future University Hakodate, Japan, and is teaching interoperable processing for character and date-time.

Mr. Yamaguchi is a member of the Information Processing Society of Japan (IPSJ) and the Japanese Cognitive Science Society (JCSS).

Michiko Oba obtained her B.S. in physics from Japan Women's University in 1982, and Ph.D. in engineering from Osaka University, Japan, in 2001.

She worked in the Systems Development Laboratory and the Software Division of Hitachi Ltd. Presently, she is a professor in the Department of Media Architecture, Future University Hakodate, Japan.

Prof. Oba is a member of the Information Processing Society of Japan (IPSJ), the Institute of Electrical Engineers of Japan (IEEE), the Institute of Electronics, Information and Communication Engineers (IEICE), the Japan Society for Software Science and Technology (JSSST), the Japanese Society for Artificial Intelligence (JSAT), and the Japan Society for Educational Technology (JSET).