Syntax Structure Based Typing Tool for Source Code Writing

Takashi Yamaguchi[1], Shinya Iwasaki[1], Kazuma Mori[1], Kenneth J. Mackin[1], Masaki Hanada[1], Eiji Nunohiro[1]

[1]Tokyo University of Information Sciences, Japan
(Tel: 81-43-236-1232, Fax: 81-43-236-1232)

[1]tyamagu@rsch.tuis.ac.jp

Abstract: Typing is a fundamental skill for the computer works that has the tasks of text inputting by keyboard. For improving computer work efficiency, the training of fast typing is very important. In this paper, we proposed a typing tool that presents a source code for the subject word. Proposed application has following three aims: the learning of touch typing, the learning of the syntax structure of programming language, the learning of source code writing process, and the improvement of continuity in programming language practice. Our typing tool presents a part of source code for the subject words. The parts of source code are derived by blocks of syntax structure on programming language. These parts are presented by the sequences from super to sub nested blocks. From this source code presentation, the learner can feel the syntax structure and the writing process of expert programmers.

Keywords: e-learning, programming, typing tool, game

1 INTRODUCTION

Typing is a fundamental skill for the computer works that has the tasks of text inputting by keyboard. For improving computer work efficiency, the training of fast typing is very important. Therefore, in an elementary information technology education, students are often trained the touch typing that faster typing technique without using the sense of sight to find the keys. However, the long iterative practices are necessity because the finger placement of touch typing must be learned and unconsciously used.

For the training of touch typing, the typing tools are commonly used. The typical typing tool is software product that presents the subject words to make a learner types this subject words, and shows the result of typing errors for help the training of typing. Because the learning of touch typing is based on the iterative practice, the typical typing tools are designed like the computer games to improve the continuity of practice by make the attention and competitive spirit of learner [1].

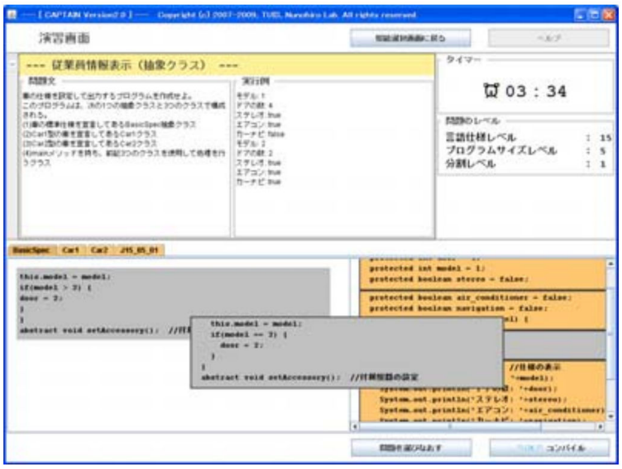
The improvement of learning continuity is very important characteristics for self-learning using e-learning systems because typical e-learning systems are used for self-learning. For improvement of learning continuity, John M. Keller proposed ARCS model that is an instructional design model by focusing on Attention, Relevance, Confidence, and Satisfaction [2]. These 4 ARCS model features are considered to be fundamental features to develop the design of products and package soft wares that include the typing tool or game.

In our past research, we develop an e-learning system for JAVA programming [3-8] that focus on the improvement continuity by ARCS model. Our e-learning system focuses on the programming learning with entertainment. In this paper, we investigate an application of typing tool for the elementary training of the programming language as another approach of entertainment e-learning system for programming.

2 PROGRAMMING LEARNING SYSTEM

2.1 puzzle-based programming learning system

In our past research, we develop CAPTAIN (Computer Aided Programming Training And INstruction) that is an e-learning system for JAVA programming. Fig.1 is screenshot of CAPTAIN client.



--- 従業員情報表示（抽象クラス） ---

問題文

車の仕様を設定して出力するプログラムを作成せよ。
このプログラムは、次の1つの抽象クラスと3つのクラスで構成される。
(1)車の標準仕様を宣言してあるBasicSpec抽象クラス
(2)Car1型の車を宣言してあるCar1クラス
(3)Car2型の車を宣言してあるCar2クラス
(4)mainメソッドを持ち、前記2つのクラスを使用して処理を行うクラス

実行例

モデル: 1
ドアの数: 4
ステレオ: true
エアコン: true
カーナビ: false
モデル: 2
ドアの数: 2
ステレオ: true
エアコン: true
カーナビ: true

03 : 34

言語仕様レベル : 15
プログラムサイズレベル : 5
分割レベル : 1

```
this.model = model;
if(model > 3) {
door = 2;
}
}
abstract void setAccessory(); //付
```

```
  this.model = model;
  if(model == 3) {
    door = 2;
  }
}
abstract void setAccessory(); //付属機器の設定
```

```
protected int model = 1;
protected boolean stereo = false;
```

```
protected boolean air_conditioner = false;
protected boolean navigation = false;
```

```
                                  //仕様の表示
                                  "+model);
                                  "+door);
  System.out.println("ステレオ: "+stereo);
  System.out.println("エアコン: "+air_conditioner);
```

Fig.1. Screenshot of CAPTAIN client

CAPTAIN focuses on the feature of entertainment for e-learning system. CAPTAIN has following three major functions; the puzzle-based programming, the score ranking, score visualization. CAPTAIN trains the ability to read the source code by puzzle-based programming function and improve the motivation by score ranking function. Moreover, the teachers can confirm the learner's understanding and individually support their understanding led by teacher.

The puzzle-based programming is a main function of CAPTAIN client for training JAVA programming language. The source code is separated into the multiple source code parts like the orange colored area in Figure 1. These parts are presented to the learner with randomly ordering. The learner drag and drop the source code parts to restore to original source code. From this system, the learner is trained the ability of source code reading. On the other hand, the ability of source code writing is not improve since the learner has nothing to write in programming training on CAPTAIN system. Therefore, we proposed a typing tool based programming training system in order to improve the ability of source code writing.

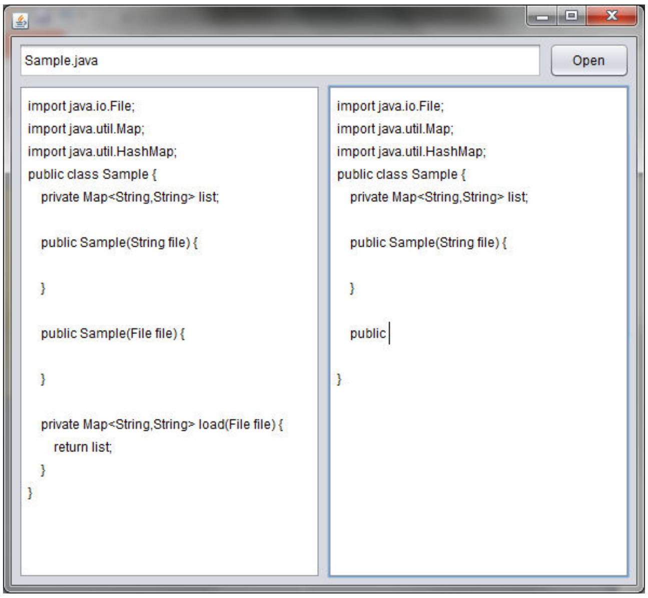
2.2 Typing tool based programming learning system

In this paper, we proposed a typing tool based programming training system that presents a source code as the subject word for each syntax structure. Proposed application has following four aims: the learning of touch typing, the learning of the syntax structure of programming language, the learning of source code writing process, and the improvement of continuity in programming language practice. In the scene of programming lecture, the teachers have a problem that the slowness of programming task disturbs the learner's understanding and the progress of lecture. The proposed training system is expected to resolve this problem by the training of touch typing and source code writing with continuous self-learning.

Figure 2 is a prototype of source code typing tool. The subject words are shown in the text pain of left side and the learner writes the words to match the subject words. The subject words are incompletely shown in initial state. The subject words are updated when the learner correctly write the subject words. Moreover, the input error check method of our typing tool is different from the typical typing tools. The typical typing tool requires the keyboard input and check the input errors for each word. Our typing tool check the input errors for each time that line feed is input in order to check syntax error in the future. Additionally, the number of times that learner hit the back space and delete key is counted for the typing error summary.

Our typing tool presents a part of source code for the subject words. The parts of source code is derived by blocks of syntax structure on programming language such as class definition, method definition, if statement, for statement, and etc. Typically, this syntax structure can be seen as the indention of source code in general programming languages. Our typing tool presents the parts of source code for each the indention level.

In this research, we defined the source code partition rules for JAVA language. The details of source code presentation method are described in next section.



```
import java.io.File;
import java.util.Map;
import java.util.HashMap;
public class Sample {
   private Map<String,String> list;

   public Sample(String file) {

   }

   public Sample(File file) {

   }

   private Map<String,String> load(File file) {
      return list;
   }
}
```

```
import java.io.File;
import java.util.Map;
import java.util.HashMap;
public class Sample {
   private Map<String,String> list;

   public Sample(String file) {

   }

   public

}
```

Fig.2. Prototype of source code typing tool

3 IMPLEMENTATION ON JAVA

In JAVA language, source code can be grouped by nested structure of class and interface definition, field definition, method definition, if statement, for statement, while and do statement, switch statement. The JAVA source code is partitioned into these nested syntax structures and these parts are presented by the sequences from super to sub nested blocks similar to the common writing sequence of expert JAVA programmer. Therefore, proposed source code presentation method is expected to effective for elementally programming training that requires the learning of the source code writing procedure and the fundamental syntax structure.

```
/* SAMPLE PROGRAM */
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;
import java.util.Map;

public class Sample {

   public static String DELIMITER = ",";
   private Map<String, String> values;

   public Sample(String file) {
      this(new File(file));
   }

   public Sample(File file) {
      values = load(file);
   }

   private Map<String, String> load(File file) {
      <… code …>
      return list;
   }
}
```

Fig.3. Sample source code

For example, the source code presentation process is explained using an original source code of sample program shown in Figure 3. Only class definitions and package definitions are presented for initial subject words like as a Figure 4. The learner should write to match the bolded words.

When the learner input the line feed or move cursor to another line, the typing error was checked and the words include the error are highlighted in input text pain. In this time, the number of typographic error is counted for the result summary. The number of times that learner hit the back space and delete key is separately counted as keyboard input error in order to separate the result to typographic error and keyboard input error. The typographic error is considered to be occurred by the lack of understanding of programming language syntax. From this function, the learner can easily discover the error words in real time, and understands that the week point is syntax of programming language or keyboard input.

When the learner completes the initial words writing, the field, constructer and method definitions are newly presented as 2nd subject words shown as bold words in Figure 5. After the completion of 2nd subject words, the complete source code is presented as 3rd subject words.

```
/* SAMPLE PROGRAM */
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;
import java.util.Map;

public class Sample {


}
```

Fig.4. Initial subject words

```
/* SAMPLE PROGRAM */
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.HashMap;
import java.util.Map;

public class Sample {

   public static String DELIMITER = ",";
   private Map<String, String> values;

   public Sample(String file) {

   }

   public Sample(File file) {

   }

   private Map<String, String> load(File file) {

   }
}
```

Fig.5. 2nd subject words

After the training, the error summary is presented to the learner. Currently, the summary consists of the number of typographic error, the number of keyboard input error, and average times of key board input. We plan to enhance the summary based on the syntax check to be able to suggest the lack of understanding on syntax of programming language in the future.

4 CONCLUSION

In this paper, we proposed a typing tool based programming training system that presents a source code as the subject word for each syntax structure in order to train the touch typing and the source code writing in continuous self-learning. Moreover, we develop a prototype of proposed typing tool based programming training system for JAVA programming language. This system expected to effective for training touch typing, understanding syntax structure, and continuous self-learning.

For future works, we plan to evaluate the effectiveness of our system on the practical use in the self-learning the programming lectures because it is necessary to separately evaluate on the effectiveness of touch typing training, effectiveness of programming training, and the continuity of self-learning.

ACKNOWLEDGMENTS

This research was supported by The Ministry of Education, Culture, Sports, Science and Technology (MEXT), Grant-in-Aid for Scientific Research (C) (24501207).

REFERENCES

[1] M Prensky (2005), Computer games and learning: Digital game-based learning, New York: McGraw Hill

[2] JM Keller (1987), Development and use of the ARCS model of instructional design, Journal of instructional development,Springer

[3] Nunohiro E, Mackin K, Ohshiro M, Matsushita K, Yamasaki K (2007), Implementation of a GA driven programming training support system, The 12th International symposium artificial life and robotics, pp.517-522.

[4] Nunohiro E, Matsushita K, Mackin K, Ohshiro M, Yamasaki K (2008), Design and development of a puzzle based programming learning support system with genetic algorithm (in Japanese), Transactions of Japanese society for information and system in education, vol.25, No.2, pp.207-213.

[5] Yoneyama Y, Matsushita K, Mackin K, Ohshiro M, Yamasaki K, Nunohiro E (2008), Puzzle Based Programming Learning Support System with Learning History Management, The 16th International Conference on Computers in Education (ICCE2008), pp. 623-627

[6] Yamakawa Y, Ohshiro M, Matsushita K, Mackin K, Nunohiro E (2010), Programming Learning Support System 'CAPTAIN' with Motivational Study Model, The 18th International Conference on Computers in Education (ICCE2010), pp. 171-175

[7] Ohshiro M, Mackin K, Matsushita K, Nunohiro E,Yamakawa Y (2010), Programming Learning Support System with Learning Progress Monitoring Feature, Joint 5rd International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent Systems (SCIS-ISIS 2010), pp.1465-1468

[8] Ohshiro M, Matsushita K, Mackin K, Yamaguchi T,N unohiro E (2012), Programming Learning Support System with Competitive Gaming Using Monitoring and Nickname s, The 17th International symposium artificial life and robot ics, pp.238-241.