A Graphical User Interface of a Visual Simulator for Computer Literacy and Architecture Education

Yoshiro Imai*, Keiichi Kaneko** and Masaki Nakagawa**

* Kagawa University, 2217-20 Hayashi-cho, Takamatsu city, 761-0396, Japan
** Tokyo University of Agriculture and Technology, 2-24-16 Nakacho, Koganei city, 184-8588, Japan
imai@eng.kagawa-u.ac.jp, {k1kaneko, nakagawa}@cc.tuat.ac.jp

Abstract: A visual computer simulator has been designed and developed to help learners to understand computer literacy and architecture. It is used in real lectures and exercises of computer architecture. Our simulator is written in Java and works as an applet as well as a stand-alone application. Its graphical user interface (GUI) provides visual simulating function in the register-transfer level and assists beginners of computer science to comprehend how computer works. This paper describes details of its GUI and reports a statistical evaluation from its users (learners). And, it is confirmed that an educational effect of its GUI is statistically significant.

Introduction

As computers advance by rapid strides, recently, education with computers, so-called an IT-based education, becomes much more popular in many kinds of educational fields, especially engineering education. In order to gain wild range of academic activities, it is necessary for learners to understand computer more and more. The more a computer itself evolves, the more its internal structure and behavior becomes like a black-boxed. So it is very import for learners to understand the internal structure and behavior of a computer precisely, if they want to utilize a computer in the various application fields.

It becomes much more necessary for learners to understand Computer Literacy and Architecture in the efficient way (Yehezkel, 2001), (Anido-Rifón, 2001), (Null, 2003). Of course, it takes relatively long time for beginners to study how a computer works. Some types of simulators have been designed and developed to educate Computer Literacy, Information Science and Computer Architecture as useful educational tools. By the way, it is one of the most efficient and effective approaches for learners to use Web-based educational tool (Chen, 2005). Many researches and projects have provided such useful tools during a few years. Web-based e-Learning system seems to be one of the most fruitful results of IT-based education. However, there is an important problem how to design and develop Web-based educational tool suitably.

There may be no royal way to design or develop a general-purpose Web-based educational tool or e-Learning system. In order to design and develop a specific educational tool, therefore, it is useful to 1) survey the related works and obtain the suitable hits or ideas from them, 2) bring objectives (design concepts) into focus, 3) implement a prototype in a reasonable way, 4) apply it to real education field and 5) evaluate it based on viewpoint of effectiveness.

We have done in the above way as this paper describes in the following sections. Our paper focuses how to design and develop a graphical user interface (GUI) of our educational tool. So the paper presents some kinds of related works, overview of a GUI of our tool, its typical functions, application of it into real education, a brief evaluation of it by means of chi-square test for an examination result of its users (learners) and finally concluding remarks, sequentially.

Related Works

Visualization is an absolutely necessary keyword and idea to improve the learner's understanding level. For example, when instructors educate their learners about computer, they want to use effective educational tools. These tools are expected to have some kind of function to visualize what is difficult to understand. With such tools, many learners will understand computer in a shorter period than other cases without using visualization tool (Yehezkel, 2001), (Anido-Rifón, 2001), (Null, 2003). The first half of this section focuses the following three researches as the related works based on simulators with visualization.
Yehezkel et al. have pointed out that teaching computer architecture is not an easy task. So they provide three types of simulators with visualizations for different computer architectures. They are (a) EasyCPU for the Intel 80x86 families; (b) Little Man Computer for a general von Neumann computer architecture; and (c) RTLsim simulator for a MIPS-like CPU (Yehezkel, 2001). They are excellent works, but their GUIs are not general nor common. It is difficult for beginners to use different GUIs of education tools in the related educational fields.

Anido-Rifón et al. have designed, implemented and tested a Web-based learning system in pure Java. They have attempted to stay within those standards what are suggested for distance learning, particularly Web-based collaborative distance learning. From their paper, an architecture education of their system seems to be well conceived because of its visualization. Their academic results and opinions from their students who have utilized their system are generally positive (Anido-Rifón, 2001). But the function of simulation is limited for a mid-level or lower computer course, so it seems to be not so useful enough to be applied to assembly programming exercise including recursion.

Null et al. have prepared MarieSim (a computer architecture simulator based on the MARIE architecture) to teach beginners to study computer organization and architecture. It provides interactive tools and simulations to help them deepen their understanding. The graphical environment for MarieSim is written in Java Swing and seems to be useful in introductory computer architecture (Null, 2003). But MARIE employs accumulator-based simple architecture, so that it is not so suitable enough to execute recursive assembly program and moreover the MarieSim is not completely web-based but Java stand-alone application. It is not so convenient.

Web-based educational tools have realized powerful and fruitful results from scientific field to computing one (Humar, 2005), (Grigoriadou, 2006). The second half of this section focuses the following two researches as the related works based on Web-based e-Learning system with effective GUIs. Humar et al. have proposed a strategic approach to integrate already-developed components for development of a web-based learning environment. Although examples from their system only demonstrate how the system can be used with a course on electromagnetism, however, their basic approach must be applicable in other fields of engineering and natural science (Humar, 2005).

Grigoriadou et al. have introduced a Web-based educational environment for teaching the computer cache memory and shown their aims to support and enhance the learning process for such a special and normally-invisible memory. The results obtained from the application/evaluation of such a Web-based environment are to indicate that the simulation and such an approach can effectively support and enhance the learning process (Grigoriadou, 2006).

From the surveys of above described works, we have decided to design a Web-based educational tool with visualization and implement an effective GUI in order to support user-friendly learning process. The following three sections illustrate the detail of our Web-based educational tool for Computer Literacy and Architecture as focusing the characteristics of its GUI.

A Graphical User Interface of a Visual Computer Simulator

Our visual computer simulator is called VisuSim and originally designed for an educational tool to demonstrate how a computer works graphically (Imai, 2006). Figure 1 shows an overview of VisuSim and its GUI as a Java applet downloaded from a dedicated web server. The server includes a Java-applet code for VisuSim, several kinds of sample programs for it and messages for its users. According to its user, the server transfers a code of VisuSim and files of sample program/ message to user's browser.

The GUI of VisuSim is organized with following parts and buttons. The GUI facilitates to illustrate how a computer works and gives users of VisuSim (i.e. learners of VisuSim) graphical explanation of internal behavior and structure of computer.

(1) of Figure 1 shows Control Unit part of VisuSim; This part includes Program counter (PC), Instruction register (IR), Opcode decoder and special fields for effective addressing. It illustrates the instruction fetching, its decoding, generating effective address and updating PC. It also explains how the instruction in the memory is transferred from memory into IR and then it is decoded to generate effective address.
(2) of Figure 1 shows Processing Unit part of VisuSim; This part includes Arithmetic Logical Unit (ALU), Condition code register (CCR) and General registers (GRs). It calculates unary operation and binary operation and generates according result and the program state into GRs and CCR, respectively. The data of memory whose address is specified by Memory address register (MAR) is transferred from memory into Memory data register for reading (MRDR). And it is calculated with the content of specified GR. When the content of GR is updated, its location has been illustrated by means of changing its color (i.e. turned to yellow). And the index of GR has been also changed into red character. So users of VisuSim can easily understand which GR has been updated and whether CCR has been changed or not, during instruction processing.

(3) of Figure 1 shows Memory Unit part of VisuSim; This part includes 256/512 words of memory, MAR, MRDR and Memory data register for writing (MWDR). In order to understand assembly programming more easily, the contents of memory can be described with assembly codes and/or numerical values. Indications are equipped for learner's convenience as follows; The address of memory, which is read, has been changed into blue characters. And the location of memory whose content is updated has been easily detected by means of changing its color of content's frame.

(4) of Figure 1 shows Speed Control Slide bar, four Control Buttons, and a message text field. Their image are enlarged and displayed on the top of Figure1. Four buttons are for initializing VisuSim, loading program from the web server / a text area for writing program source described later, executing program in the step-by-step mode, and running program in the continuous mode, respectively. In order to understand a detail of processing program, users can adjust the execution speed of VisuSim by means of the Speed Control Slide bar at the left-side of four buttons. The button (“StepGo”) is available to check the behavior of computer step by step, while the button (“AutomaticGo”) is suitable to trace user's algorithm.

(5) of Figure 1 shows the text area for writing program source. Users write directly their programs in this area. And they switch button (“LoadProg”) to copy the content of this area into memory. Finally users can choose to execute their programs in the step-by-step mode or run them in the continuous one.
(6) of Figure 1 shows other kinds of control buttons for switching input direction to transfer program (i.e. getting program from the web server or copying program from a text area for writing program source) and invoking on-line help message. In the Java-applet mode of VisuSim, it can obtain some kinds of files for sample programs and on-line help message from the web server.

When VisuSim is invoked, it automatically recognizes its executing environment and decides whether it should be performed as an applet or as a stand-alone application. VisuSim has the GUI described above, illustrates an internal behavior and structure of computer and facilitates that its learners can understand how a computer work in the register-transfer level. It reads sample assembly programs from the web server that is the same server for VisuSim itself. Downloaded program is located in the scratch pad text area for writing program source. Learners can write their program directly in the same area. Any program can be transferred from the area into memory by means of simple operation. And it is processed by a graphical CPU simulator (i.e. register-transfer-level simulation) of VisuSim and investigated whether it is correct or not.

Functions of Visual Simulation

Table 1 shows a brief explanation of Instruction set for VisuSim. Such an instruction set is separated into the following 4 groups; control instructions, jump instructions, unary operation instructions, and binary operation instructions. Typical addressing of VisuSim includes direct addressing, indirect addressing or immediate value addressing.

<table>
<thead>
<tr>
<th>Instruction Set of Computer Simulator</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table 1</strong></td>
</tr>
<tr>
<td><strong>Opcode</strong></td>
</tr>
<tr>
<td><strong>Control Instruction</strong></td>
</tr>
<tr>
<td><strong>Jump Instruction</strong></td>
</tr>
<tr>
<td><strong>Unary Operation Instruction</strong></td>
</tr>
<tr>
<td><strong>Binary Operation Instruction</strong></td>
</tr>
</tbody>
</table>

Structure of program is sometimes discussed in a lecture of Computer Architecture including an assembly programming exercise. Assembly programming is not so easy for beginners to understand, but it is efficient to comprehend not only the internal structure of computer but also the detail of programming techniques, because learners cannot understand detail behavior of computer only from viewpoint of high-level language programming. Programs including iteration with index registers belong to difficult subjects so that beginners hardly understand how to use a computer and write a suitable program. In general, iteration is implemented with combination of conditional jump, indirect addressing, unconditional jump and so on.
Frequently beginners suffer from lack of a suitable educational tool that can demonstrate visually, for example, how a computer processes a complicated iteration. It is very useful to illustrate a concept of effective addressing with such an educational tool. Not only learners but also instructors sometimes want to utilize such a tool to show a schematic distinction between direct addressing, indirect one and immediate value one by means of suitable example. Explanation of mechanism for subroutine call and return is one of the most essential themes in Computer Literacy. It must involve calling sequence with parameter passing and allocation of stack and heap.

Recursive subroutine call may be frequently included in the above themes. As a stack frame is easily handled by a few suitable instructions, recursive programs can be introduced at the assembly programming level. It is very important for learners of information engineering field to have a graphical explanation of internal behavior and structure of computer. So it will be very effective for such learners if a suitable education tool is easy to illustrate how a stack area grows and reduces while program is executed in computer. With such an educational tool, its users (learners) can understand precise mechanism of subroutine call through an assembly programming exercise.

A Case Study of Real Computer Education with Visual Simulator and its Brief Evaluation

Students of our class can understand the behavior and structure of computer not only through watching demonstration with VisuSim from a teacher but also through manipulation of VisuSim by themselves. It is one of the most popular cases that a teacher of the class of Computer Architecture uses VisuSim as an educational tool in order to explain graphically how a computer works. With a projector and a large screen, the teacher directly gives his students the output of his PC where VisuSim is executing. At the same time, some students may download VisuSim from a dedicated web server, execute VisuSim on their PCs and compute (i.e. simulate) some sample programs. In such a case, VisuSim is distributed to students as a form of Java applet. Some sample assembly programs and online-help messages are prepared and downloadable from the same web server.

Assembly programming exercise is suitable for learners to understand the behavior and structure of computer through their programming processes. This is another case of using VisuSim. For example, a pair of learners is obliged to work together to accomplish a common goal under below conditions. Learners have to rely on each other to achieve the goal. If any one learner fails to do his or her job part, both of them take the consequences. It will be a good example of useful interdependence. In such a case, each learner is held accountable for doing his or her share of given problems and for mastery of all of the jobs to be learned. In order to exchange frequently several information and idea between each other, function of communication by VisuSim may be convenient for a pair of learners to study and exercise assembly programming by themselves in such a collaborative learning environment.

Table 2 shows the relation of learners' scores between the reports using VisuSim for understanding computer and examinations for Computer Literacy. The number of learners is 81. The ranks of the scores for reports are classified from A+ to D (ascending order), while the ranks of the scores for examination are "Superior", "Excellent", "Good", "Fair" and "No Good", respectively. Based on this table, we try to evaluate our visual simulator VisuSim by means of test of significance as follows. \( H_0 \) is a null hypothesis: "using GUI of VisuSim is independent from learners' scores of Computer Literacy." A chi-square test is used to decide whether the null hypothesis \( H_0 \) holds or not. With data of Table 2, the chi-square statistic is

\[
\chi^2 = \frac{(8-27\cdot10/81)^2}{27\cdot10/81} + \cdots + \frac{(8-10\cdot14/81)^2}{10\cdot14/81} = 87.5
\]

By the way, the significance of the chi-square statistic is evaluated for \((5-1)(5-1)\) degrees of freedom. The chi-square percentile with \((5-1)(5-1)\) degrees of freedom at the 5% significance level is \( \chi^2_{0.05}(16) = 26.3 \). From this result, the null hypothesis \( H_0 \) is rejected and the alternative is accepted. An evidence of the above analysis indicates that a relation between using GUI of VisuSim to understand Computer Literacy and the learners' scores of Computer Literacy shows a statistically significant dependency. So it is said to be statistically significant that GUI of our visual simulator is effective to learn Computer Literacy.
### Table 2: Distribution of Learners’ Scores for Report and Examination

<table>
<thead>
<tr>
<th></th>
<th>Superior</th>
<th>Excellent</th>
<th>Good</th>
<th>Fair</th>
<th>NoGood</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>8</td>
<td>13</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>15</td>
<td>10</td>
<td>4</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>10</td>
<td>26</td>
<td>17</td>
<td>12</td>
<td>14</td>
<td>81</td>
</tr>
</tbody>
</table>

### Concluding Remarks

GUI, functions and applications of our visual simulator VisuSim are introduced in this paper. The system configuration and some applications of VisuSim are also described and explained concretely in the paper. VisuSim is designed as a Web-based educational tool in order to illustrate visually how a computer works. With its facility of GUI, simulation and communication, it is very useful for learners to utilize such a tool in real education field. We can conclude our study as follows:

1) With VisuSim, a graphical demonstration is available in classroom lecture on Computer Architecture as well as Assembly Programming. It is easy and transparent for even beginners to understand the internal behavior and structure of computer more precisely;

2) VisuSim can provide an effective GUI of simulation to learners. It is confirmed that an educational effect of its GUI is statistically significant.

### Acknowledgements

This research is being partially supported by the MEXT Research and Education Fund for Promoting Research on Symbiotic Information Technology. This study was partly supported by Grand-in-Aid for scientific research from the Ministry of Education, Culture, Sports, Science and Technology under the contract No. 19500044.

### References


