doi:10.2498/cit.2006.04.01

Application of a Visual Computer Simulator into Collaborative Learning

Yoshiro Imai[1], Keiichi Kaneko[2] and Masaki Nakagawa[2]

[1]Kagawa University, Hayashi-cho, Takamatsu-shi, Kagawa, Japan
[2]Tokyo University of Agriculture and Technology, Koganei-shi, Tokyo, Japan

A visual simulator has been developed for understanding computer structure and behavior. It is written in the Java programming language, downloaded from a Web server, and executed on major browsers. It has been designed to illustrate how a computer works visually and runs not only as a Java stand-alone application, but also as a Java applet. Our visual simulator reads program files written in an assembly language, executes them in step-by-step manner as well as in automatic (i.e., continuous) manner and demonstrates how those programs are processed graphically by computer.

In order to improve functionality of the simulator, some modules have been newly added, which are designed to perform, for example, on-line message service, guideline display and built-in e-mail handler. In this paper, new facilities of our simulator will be mainly explained. They improve its user interface and provide communication support between teachers and students for the purpose of distance education.

Keywords: visual simulator, collaborative learning, graphical user interface, built-in e-mail handler, communication support.

1. Introduction

People have already recognized the importance of information technology-based education in every respect. It is indispensable to understand the intrinsic function of a computer and to learn the expanding capability of information technology. Many universities and technical colleges have been suffering from the lack of efficient teaching systems to educate the intrinsic function of a computer for a long time. Several teaching systems have been proposed and developed all over the world [1][2][3].

It is very important not only to understand a computer efficiently, but also to excel in the availability of computer-related teaching systems. Namely, teaching systems must be evaluated through their practical usage in terms of whether it can be used efficiently or not by universities and/or technical colleges.

In this paper, the outline of a visual simulator is introduced, which is developed for the purpose of illustrating the structure of a computer and its behavior. Its online message function, guideline function and built-in e-mail handler, which are adopted in the revised version of the simulator as its communication support facilities, are explained. Finally, it is reported how effectively its built-in e-mail handler may perform information interchange between users. It is discussed whether such communication support facilities are useful for distance learning between teachers and students.

2. Outline of Visual Computer Simulator

In this section, a visual computer simulator called "VisuSim" [4] is introduced. Its facilities are summarized as follows:

- It is used as a teaching system to teach computer science, namely, from Computer Architecture to Assembly Programming Exercise.
- It graphically illustrates the internal structure and register-transfer level behavior of von Neumann type computer.
- Its simple GUI is equipped for several operations such as initialization, program loading, step-by-step execution and automatic (i.e., continuous) running.

- It simulates a computer, which receives an assembly program into its memory, reads each instruction of the program from the memory, processes it with registers and ALU and stores the result into register and/or memory.
- It has a single source code which can be executed as a Java stand-alone application as well as a Java applet.
- It is delivered by means of download from a Web server, upon the request from a client PC.
- It can run on the major Web browsers with Java Virtual Machine.
- It is absolutely independent from a particular environment, such as specific Operating System and Hardware, because its code is written in pure Java language.

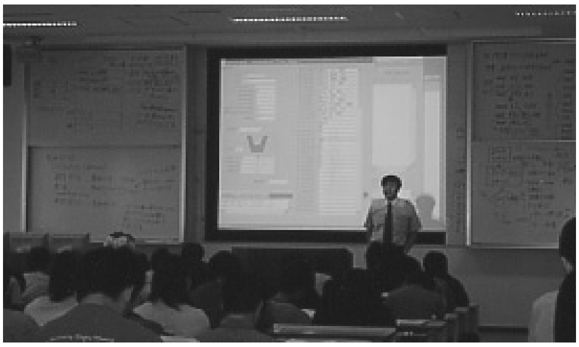


Fig. 1. Practical Education with "VisuSim"

Figure 1 shows a photo of real usage of a visual computer simulator, "VisuSim" in our class. It is displayed on the screen between whiteboards of the lecture room. Students in our class can understand computer behavior and structure while operating "VisuSim" step by step. It is one of the most popular ways a teacher of Computer Science can utilize "VisuSim" as a teaching system to illustrate how the computer works. With a projector and a large screen, the teacher can directly show his students the output of PC where "VisuSim" is operating.

In another case, students can download "VisuSim" from the specific Web server, utilize it on their PCs, compute (or simulate) their programs and, by means of stepwise and/or continuous executing facilities of "VisuSim", investigate whether their programs work correctly or not. In such a case, "VisuSim" is distributed to a client as a form of Java applet. Some sample assembly programs are prepared and downloadable from the same Web server. They are stored into the specific area of "VisuSim" and transferred into its memory by an easy button operation. Figure 2 shows a typical usage of "VisuSim" as a Java applet.

With adjustment of description for the HTML file, some kinds of sample programs are easily selectable based on clients' requests and, moreover, the online help message of "VisuSim" can be also provided according to the understanding level of clients. In order to use "VisuSim" properly, it is possible to provide efficient sample programs and suitable online help messages for the aim of using "VisuSim", for example, in teaching information literacy, supporting assembly programming, and so on.

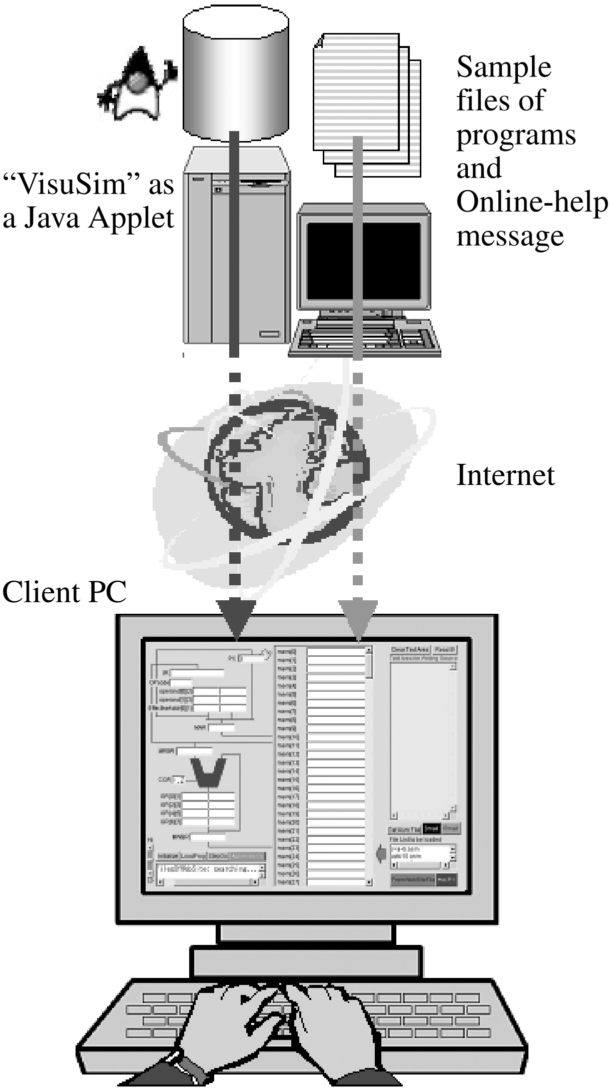

"VisuSim" as a Java Applet
Sample files of programs and Online-help message
Internet
Client PC

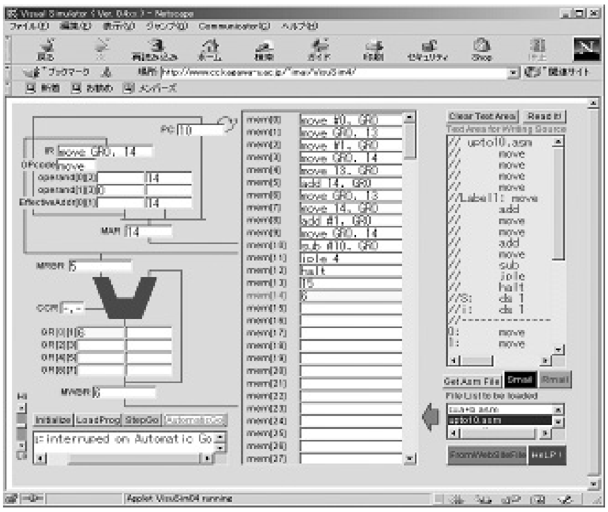
Fig. 2. Usage of "VisuSim" as a Java Applet

3. Communication Support Facilities for Computer Simulator

As a teaching system, it is very important to provide useful user interface and efficient communication facilities between users, such as students and teachers. GUI and E-mail handling facilities of our visual computer simulator are described in this section.

3.1. GUI of Visual Computer Simulator

The visual simulator, "VisuSim", provides the graphical user-interface shown in Figure 3. In this case, the simulator is invoked as a Java applet on the browser, namely, Netscape Navigator.



```
PC      10
IR      move GR0, 14
OPcode  move
operand[0][2]        14
operand[1][3]  0
EffectiveAddr[0][1]  14
MAR     14
MRDR    5
CCR     -, -
GR[0][1] 6
MWDR    6

mem[0]   move #0, GR0
mem[1]   move GR0, 13
mem[2]   move #1, GR0
mem[3]   move GR0, 14
mem[4]   move 13, GR0
mem[5]   add 14, GR0
mem[6]   move GR0, 13
mem[7]   move 14, GR0
mem[8]   add #1, GR0
mem[9]   move GR0, 14
mem[10]  sub #10, GR0
mem[11]  jple 4
mem[12]  halt
mem[13]  15
mem[14]  6
mem[15]
mem[16]
mem[17]
mem[18]
mem[19]
mem[20]
mem[21]
mem[22]
mem[23]
mem[24]
mem[25]
mem[26]
mem[27]

1: interrupted on Automatic Go
```

```
// upto10.asm
//          move
//          move
//          move
//          move
//Label1:   move
//          add
//          move
//          move
//          add
//          move
//          sub
//          jple
//          halt
//S:        ds 1
//i:        ds 1
//------------------
0:          move
1:          move
```

```
sum+2.asm
upto10.asm
```

Fig. 3. GUI of "VisuSim" as a Java Applet

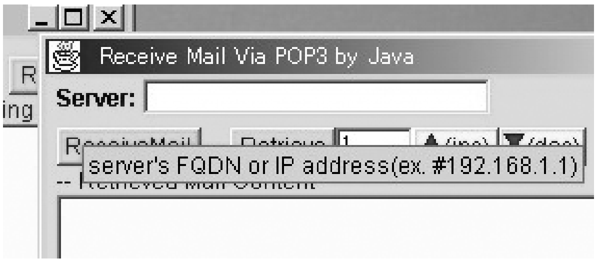
"VisuSim" has 4 major buttons on the left side of the bottom of its GUI for operating itself interactively. From left to right, those buttons are for initializing (or resetting) system, loading assembly program into memory, executing the program step by step (called "stepwise execution mode"), and running the program continuously (called "automatic go mode"). On the left from the buttons, there is a vertical slide bar for speed control in the automatic go mode. Below the 4 buttons, there is a text area for displaying the internal state of "VisuSim".

On the left side of the GUI, there is a CPU area including program counter, instruction register, memory address register, ALU, general-purpose registers, condition-code register, and so on. In the center of the GUI, there is a memory area. Its capacity is 128 words. In order to be more convenient for assembly programming and illustration of instruction set, an assembly mnemonic code can be directly described in the content of memory.

Users sometimes want to check the state of "VisuSim" or modify value of the content of a specific register, although it is running in the automatic go mode. Users can click again the button labeled "Automatic Go" for running program continuously. After the button turns into reverse color, namely red characters "Automatic Go" on the white background, "VisuSim" stops executing and waits for checking and/or modifying. This function is very useful for interactive program debugging. Figure 3 shows that "VisuSim" has stopped by clicking again the button labeled "Automatic Go" while it is running the program continuously.

On the right side of the GUI, there is a special text area for writing an assembly program in the conversational mode of "VisuSim". And below that text area, there are 3 important buttons for getting sample files in the local disk (or Web server where "VisuSim" is stored as a Java applet), invoking an SMTP-client and a POP3-client, from left to right. The latter two buttons will be described precisely in the next subsection. After clicking the button for getting sample files, the text area below the 3 buttons can indicate the suitable programs to be loaded into memory, for example, from the local disk. On the right side of the bottom, there are 2 buttons for selecting direction of input source program into memory and invoking online help message from the local disk (or the Web server). Online help message mechanism will be explained later in this subsection.

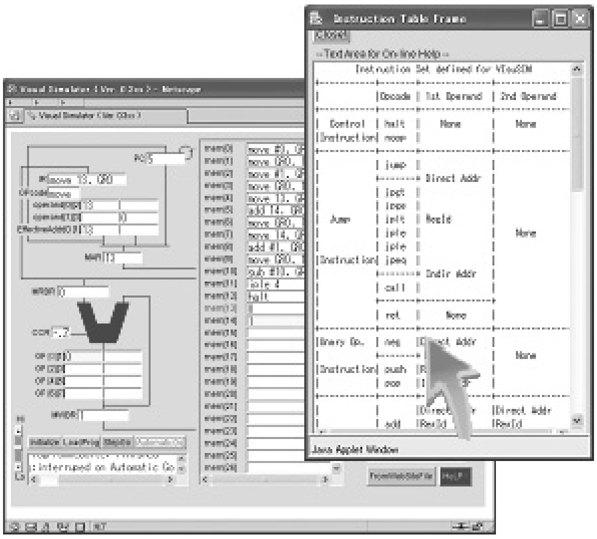
In the interactive system, it is very important how to lead users through suitable button operation and/or text-field description. Guideline display function is useful and efficient in reducing wrong operations and misunderstanding of users. In order to improve the user interface of "VisuSim", a dialogue balloon shown in Figure 4 provides a guideline to explain how to use the button and what to describe in the text field by means of pointing the mouse cursor at the near position of the target button/textfield. The example below shows a guideline how to use the text field of "Server" and instructs users how to describe FQDN (fully qualified domain name) or IP address.



server's FQDN or IP address(ex. #192.168.1.1)

Fig. 4. Guideline Function of "VisuSim" as a Java Stand-alone Application

A help message function is necessary to utilize any system properly. "VisuSim" can display a content of the selected file as an online help message if it is located as a specific name. A teacher or administrator of "VisuSim" can select the suitable file for his students so that such a file will be read and displayed as a help message of the system. In a teaching system, especially, it is very useful to display an effective help message according to the intelligibility of users. Figure 5 shows a help message generated from "VisuSim".


Visual Simulator (Ver. 0.2xx) - Netscape
Instruction Table Frame
Close!
-- Text Area for On-line Help --
Java Applet Window
FromWebSiteFile
HeLP

```
              Instruction Set defined for VisuSIM
+-----------+-------+---------------+-------------
|           |Opcode | 1st Operand   | 2nd Operand
+-----------+-------+---------------+-------------
| Control   | halt  |    None       |    None
|Instruction| noop  |               |
+-----------+-------+---------------+-------------
|           | jump  |               |
|           +-------+ Direct Addr   |
|           | jpgt  |               |
|           | jpge  |               |
|  Jump     | jplt  | RegId         |    None
|           | jple  |               |
|           | jple  |               |
|Instruction| jpeq  |               |
|           +-------+ Indir Addr    |
|           | call  |               |
|           +-------+---------------+
|           | ret   |    None       |
+-----------+-------+---------------+-------------
|Unary Op.  | neg   |Direct Addr    |
|           +-------+               |    None
|Instruction| push  |R              |
|           | pop   |               |
+-----------+-------+---------------+-------------
|           |       |Direct Addr    |Direct Addr
|           | add   |RegId          |RegId
```

Fig. 5. Online-Help Message Function of "VisuSim" as Java Applet

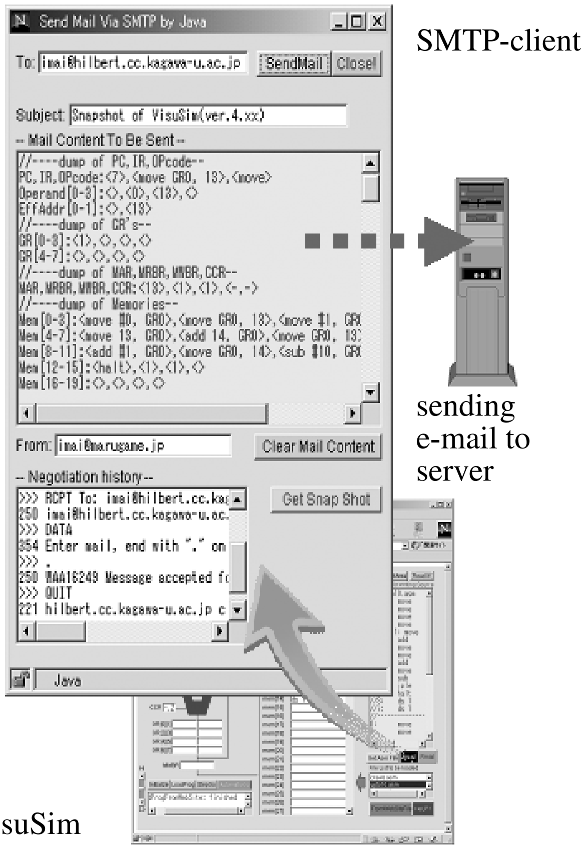
By means of clicking the "HeLP" button, "VisuSim" can read and display the specific help message file from local disk (when it is invoked as a Java stand-alone application) or from the Web server (otherwise, invoked as a Java applet). With such an online-help message facility users can utilize "VisuSim" by themselves and make good use of improved user interface. It reduces unnecessary responses among users so that it performs qualitative repletion of communication between users.

3.2. E-Mail Handling Facilities of Visual Computer Simulator

"VisuSim" has two e-mail handlers as its built-in modules, which are both written in pure Java language. One of the e-mail handlers is an e-mail sending module as an SMTP-client and the other is an e-mail receiving module as a POP3-client. When users utilize "VisuSim" and want to correspond with others, they can invoke e-mail sending or receiving module of "VisuSim". These handlers are available whenever "VisuSim" is invoked as a Java stand-alone application as well as a Java applet.

An SMTP-client is a simple mail transfer protocol-based sending module shown in Figure 6, which has two characteristics as follows:

- By an easy operation, it can transfer all the contents of registers and memory of "VisuSim" into another.
- It can allow users to specify the target e-mail address that not only includes a domain name, but is also based on the IP address.



To: imai@hilbert.cc.kagawa-u.ac.jp

Subject: Snapshot of VisuSim(ver.4.xx)

```
//----dump of PC,IR,OPcode--
PC,IR,OPcode:<7>,<move GR0, 13>,<move>
Operand[0-3]:<>,<0>,<13>,<>
EffAddr[0-1]:<>,<13>
//----dump of GR's--
GR[0-3]:<1>,<>,<>,<>
GR[4-7]:<>,<>,<>,<>
//----dump of MAR,MRBR,MWBR,CCR--
MAR,MRBR,MWBR,CCR:<13>,<1>,<1>,<-,->
//----dump of Memories--
Mem[0-3]:<move #0, GR0>,<move GR0, 13>,<move #1, GR
Mem[4-7]:<move 13, GR0>,<add 14, GR0>,<move GR0, 13
Mem[8-11]:<add #1, GR0>,<move GR0, 14>,<sub #10, GR
Mem[12-15]:<halt>,<1>,<1>,<>
Mem[16-19]:<>,<>,<>,<>
```

From: imai@marugame.jp

```
>>> RCPT To: imai@hilbert.cc.kag
250 imai@hilbert.cc.kagawa-u.ac.
>>> DATA
354 Enter mail, end with "." on
>>> .
250 WAA16249 Message accepted fo
>>> QUIT
221 hilbert.cc.kagawa-u.ac.jp c
```

Fig. 6. Invocation of built-in SMTP-client

On the other hand, a POP3-client is a post office protocol-based receiving module. It has also the following functions according to the above characteristics of the SMTP-client.

The SMTP-client is a special e-mail sender whose facilities are as follows. First of them is useful to share with the internal state and contents of registers in the specific "VisuSim" between some users. For example, when a user of "VisuSim" meets problems and resolves them with the help of a teacher or friends, he/she can invoke the SMTP-client, store the internal state and contents of "VisuSim" registers as a mail text, and transfer such information to his/her teacher or friends through an e-mail message.

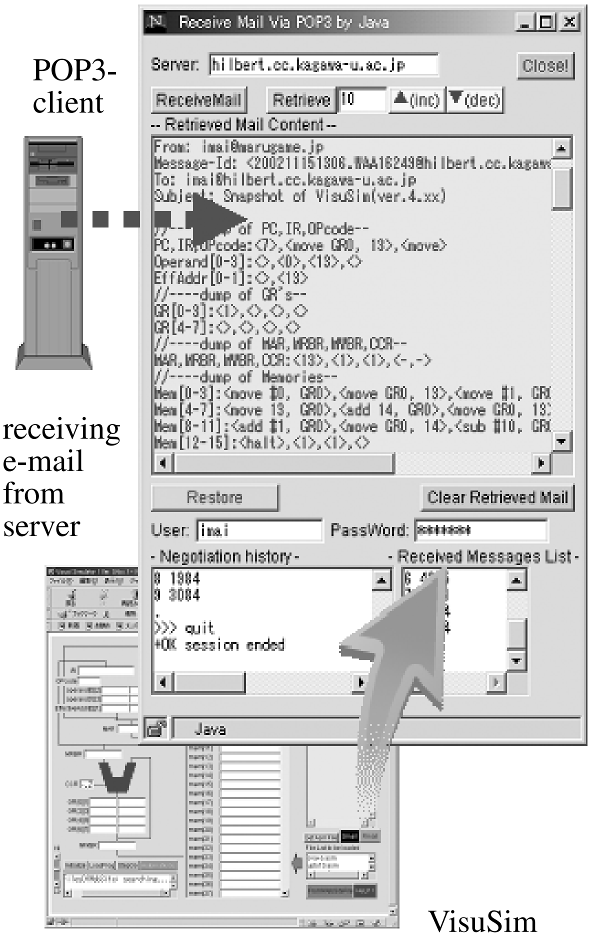
Another user can receive the mail text and gain a resurgence of duplicated image of "VisuSim" with such a mail text. With the function of POP3-client, this is the most suitable mechanism which can provide a smart facility to share the internal state and contents of "VisuSim" among a group of users. This is a powerful facility to check the remote state of "VisuSim" with a local "VisuSim" by a teacher and/or Teaching Assistants.

In the SMTP-client, "VisuSim" provides a special function called "Snap Shot", which can duplicate the internal state and contents of registers into the mail text. In other words, users can store the internal image of "VisuSim" and insert it in the mail content as a backup copy by means of clicking a button labeled "Get Snap Shot". Figure 6 illustrates invoked SMTP-client which includes the internal image of "VisuSim" to be sent.

Second of the facilities is to permit specification of e-mail address with IP address. In a normal case, an e-mail address is represented with domain name. Using "VisuSim" as a teaching system seems to be necessary to acquire the Web server and e-mail server with domain name. In real use of "VisuSim", however, a teacher or administrator cannot acquire such a server, has no time to ask a server to be entered in the domain name system, and then might have to use an experimental server without the formal domain name (FQDN: fully qualified domain name). In such a case, general e-mail handlers do not transfer any mails into the mail server with no FQDN described above.

This is why we must develop a special SMTP-based e-mail handler to transfer mails into an experimental mail server with no domain name. With our SMTP-client, users can represent not only normal notation, but also an IP-based special one as an e-mail address. The former is like `imai@eng.kagawa-u.ac.jp` and the latter is like `imai@\#192.168.1.1`, for example. It is very convenient to utilize an experiment server with no domain name and begin a very easy trial (or short-term) service of "VisuSim" without complex and time-consuming adjustment of the LAN manager.

On the other hand, just like SMTP-client, the functions of POP3-client of "VisuSim" are reading the e-mail from other "VisuSim" and specifying the mail server with its IP address only. The POP3-client can receive the e-mail text with an internal state and contents of registers from another "VisuSim", restore such information to the internal state and contents of registers of the receiving "VisuSim" itself, and gain a resurgence of the remote "VisuSim" in the local one. Such operations are easily carried out by clicking a button labeled "Restore" only. Both the STMP-client and the POP3-client are designed



Server: hilbert.cc.kagawa-u.ac.jp

Retrieve 10

```
From: imai@marugame.jp
Message-Id: <200211151306.WAA16249@hilbert.cc.kagaw
To: imai@hilbert.cc.kagawa-u.ac.jp
Subject: Snapshot of VisuSim(ver.4.xx)

//----dump of PC,IR,OPcode--
PC,IR,OPcode:<7>,<move GR0, 13>,<move>
Operand[0-3]:<>,<0>,<13>,<>
EffAddr[0-1]:<>,<13>
//----dump of GR's--
GR[0-3]:<1>,<>,<>,<>
GR[4-7]:<>,<>,<>,<>
//----dump of MAR,MRBR,MWBR,CCR--
MAR,MRBR,MWBR,CCR:<13>,<1>,<1>,<-,->
//----dump of Memories--
Mem[0-3]:<move $0, GR0>,<move GR0, 13>,<move $1, GR
Mem[4-7]:<move 13, GR0>,<add 14, GR0>,<move GR0, 13
Mem[8-11]:<add $1, GR0>,<move GR0, 14>,<sub $10, GR
Mem[12-15]:<halt>,<1>,<1>,<>
```

User: imai PassWord: *******

```
8 1984
9 3084
.
>>> quit
+OK session ended
```

Fig. 7. Invocation of built-in POP3-client

to work in cooperation with each other, send the internal image of “VisuSim” through the former, and receive it as another duplicate image of “VisuSim” through the latter. Figure 7 shows invocation of the POP3-client and reading of an e-mail with an image of another “VisuSim”.

4. Application to Collaborative Learning

An application of “VisuSim” to collaborative learning is introduced as follows;

A student wants to do his homework after school, uses “VisuSim” downloaded from the Web server, and meets some problems which he cannot resolve by himself. He decides to correspond with his teacher about his problems, invokes the SMTP-client to utilize the “Snap Shot” facility for storing the current state and contents of registers of “VisuSim” onto the e-mail text. It is transferred to the mail server (the same of Web server) through an STMP-client for asking his teacher, as shown on the left side of Figure 8.

A teacher checks his e-mails through a POP3-client, finds such a mail text and tries to gain a resurgence of internal image of his student’s “VisuSim” on his own “VisuSim” by means of “Restore” facility. The teacher can resolve the problem, transfer the result through SMTP-client and teach his student how to resolve and/or how to think about such a problem. Without such a teaching system, there might be some difficult problems between a teacher and his students, namely, how to transfer the identical situation from one to another, how to recreate the same image on the remote site, how to communicate the real result with comments from a teacher to students and so on. In such a case, a teacher and his student can be blessed with collaborative learning with the communication support facilities and smart GUI of “VisuSim”.

The above case seems to be a fruitful sample with “VisuSim” for collaborative learning and/or distance learning which has been established between teacher and student. There is another case to learn assembly programming as group meeting with “VisuSim” through network. It will be almost limit free by location and occasion.

5. Conclusion

Our visual simulator called “VisuSim” has been used to educate computer science lectures of Computer Architecture to Assembly Programming. Communication support and user interface of “VisuSim” are reported in this paper and its application to collaborative learning is proposed and demonstrated by means of a practical usage with “VisuSim” through network.

It is summarized as follows:

- “VisuSim” is available in the classroom lecture to illustrate the internal structure and

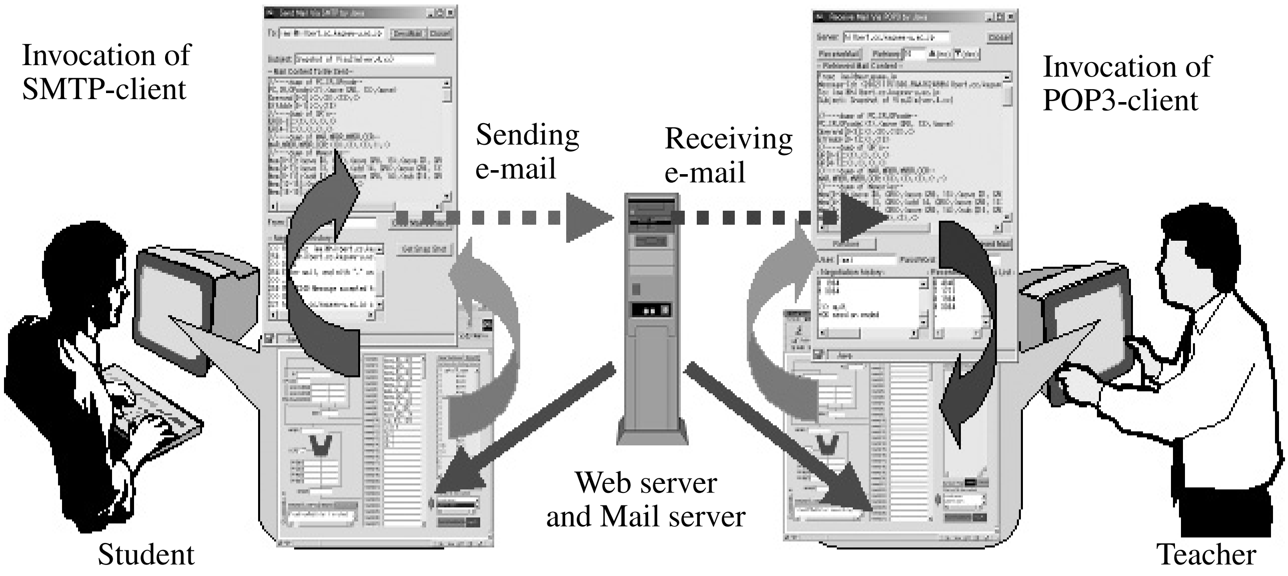

Invocation of SMTP-client
Sending e-mail
Receiving e-mail
Invocation of POP3-client
Web server and Mail server
Student
Teacher

Fig. 8. Scheme of Collaborative Learning with “VisuSim” on the Network Environment

behavior of a computer and to explain how a computer works.

- Online help message and guideline display are useful for beginners who utilize "VisuSim" by themselves.
- Built-in mail handlers are effective for users who want to communicate closely with each other during operation of "VisuSim".
- The scheme is proposed of a collaborative learning which is performed with communication support facilities and smart GUI of "VisuSim" on the network environment.

Acknowledgements

The authors would like to express sincere thanks to Prof. Shinji Tomita of Kyoto University for his supervising support. This study was partly supported by grant-in-aid for scientific research from the Ministry of Education, Culture, Sports, Science and Technology under contract No. 12040107.

References

[1] KIT K. R. YIP, K.-CHUN LI, A Web-Based CAL System on Computer Architecture and Assembly Language Programming, *Proc. Int'l Conf. Computers in Education* (ICCE2002); 2002. pp. 202–205.

[2] Y. MIURA, K. KANEKO, M. NAKAGAWA, Development of an Educational Computer System Simulator Equipped with a Compilation Browser, *Proc. Int'l Conf. Computers in Education* (ICCE2003); 2003. pp. 140–143.

[3] A. JUTMAN, A. SUDNITSON, R. UBAR, H-DWUTTKE, E-learning Environment in the Area of Digital Microelectronics, *Proc. Int'l Conf. IT-based Higher Education & Training* (ITHET2004); 2004. pp. 28–33.

[4] Y. IMAI, S. TOMITA, A Visual Simulator for Understanding Internal Structure and Behavior of Computer System (in Japanese), *Technical Report of IEICE*, ET99-110(2000-3); 2000. pp. 113–120.

Received: June, 2006
Accepted: September, 2006

Contact addresses:

Yoshiro Imai
Kagawa University
2217-20 Hayashi-cho
Takamatsu-shi, Kagawa, 761-0396
Japan
e-mail: imai@eng.kagawa-u.ac.jp

Keiichi Kaneko
Tokyo University of Agriculture and Technology
2-24-16 Nakacho, Koganei-shi
Tokyo 184-8588
Japan
e-mail: klkaneko@cc.tuat.ac.jp

Masaki Nakagawa
Tokyo University of Agriculture and Technology
2-24-16 Nakacho, Koganei-shi
Tokyo 184-8588
Japan
e-mail: nakagawa@cc.tuat.ac.jp

YOSHIRO IMAI was born on 23 July 1955 in Japan. He received his B.Eng. degree from Kyoto University in 1980. He worked at Takuma National College of Technology from 1980 to 1993. Since April 1993, he has been working at Kagawa University as an Associate Professor. He is interested in computer architecture, systems programming and e-learning design for these subjects.

KEIICHI KANEKO was born on 13 November 1962 in Japan. He received B.E., M.E., and Ph.D. degrees from the University of Tokyo in 1985, 1987 and 1994, respectively. He is an Associate Professor at Tokyo University of Agriculture and Technology. His main research areas are functional programming, parallel and distributed computation, partial evaluation, fault-tolerant systems, and pedagogical systems. He is a member of ACM, IEICE, IPSJ and JSSST.

MASAKI NAKAGAWA was born on 31 October 1954 in Japan. He received B. Sc. and M. Sc. degrees from the University of Tokyo in 1977 and 1979, respectively. During the academic year 1977/78, he followed Computer Science course at Essex University in England, and received his M.Sc. with distinction in Computer Studies in July 1979. He received his Ph.D. in Information Science from the University of Tokyo in December 1988. From April 1979, he worked at Tokyo University of Agriculture and Technology. Currently, he is a Professor of Media Interaction in the Department of Computer and Information Sciences. He has been working on pattern recognition, human interface, especially pen-based interface and educational engineering.