

A Visual Computer Simulator and its Applications to An ICT-based Higher Education

Yoshiro Imai, Keiichi Kaneko and Masaki Nakagawa

Abstract—A visual computer simulator, called "VisuSim", has been developed and applied to real education for Computer Science at Kagawa University. It is one of educational tools associated with lectures on Computer Architecture and Assembly Programming. The simulator is written in the pure Java programming language, executed in several kinds of environment (free for types of OS and CPU) and invoked in the applet mode as well as in the stand-alone one. Typical usages of our simulator are two-fold. Teachers employ our simulator, "VisuSim" to demonstrate visually how a computer processes its program. Students use the simulators in class or access our website, download "VisuSim", and invoke applet-based "VisuSim" on their browsers. After receiving feedback from users, we have provided communication support facility as a new function of "VisuSim". With such a communication facility, can send their programs, ask questions on them to their classmates and the classmates can test received programs with their "VisuSim" to think together about the questions.

Index Terms—Educational tool, Collaborative learning, Visual simulator, Java programming language, Computer literacy

I. INTRODUCTION

AN Information and Communication Technology(ICT)-based education is very important and efficient for students, especially ones of computer science. Learning methodologies are introduced for ICT-based education all over the world. And several useful keywords and concepts are suggested and proposed in the real engineering education.

Visualization is a keyword to prompt building an educational tool with graphical facility to improve effectiveness of education. Interaction is an essential keyword to realize a smart user interface of a learning system [1]. Distance education and collaborative learning are useful keywords to design suitable education environment for many users[2][3]. With such an environment, not only students but also a teacher can communicate with each other and share the same thought and

Yoshiro Imai is an associate professor at Kagawa University, 2217-20 Hayashi-cho, Takamatsu-shi, Kagawa, 761-0396 Japan (corresponding author, phone: +81-87-864-2244; fax: +81-87-864-2244; e-mail: imai@eng.kagawa-u.ac.jp).

Keiichi Kaneko is an associate professor at Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588 Japan (e-mail: k1kaneko@cc.tuat.ac.jp).

Masaki Nakagawa is a professor at Tokyo University of Agriculture and Technology, 2-24-16 Naka-cho, Koganei-shi, Tokyo, 184-8588 Japan (e-mail: nakagawa@cc.tuat.ac.jp).

concept for several items and objects.

In this paper, we present a visual computer simulator is introduced, which is designed for demonstration of the internal behavior and structure of a computer. The visual simulator gives users a simple GUI and provides smart interactive facilities for manipulation. It works as a web application, equips communication support function and exchanges information between users. Therefore, it plays a central role in the environment for collaborative learning.

II. VISUAL COMPUTER SIMULATOR

First of all, design concepts of our visual computer simulator are summarized as follows :

- a) It is used as an educational tool for several computer literacy from computer architecture to exercise of assembly programming.
- b) It illustrates the internal behavior and structure of von Neumann type computer.
- c) It is equipped with a simple GUI with interactive manipulations and button-based operations.
- d) It performs graphical simulation of computer with two different simulation modes: stepwise and continuous executions of programs
- e) It provides an effective environment of assembly program debugging for beginners.
- f) It is implemented as a single source code written in the pure Java programming language.
- g) It is easily distributed from the Web server into users through network connectivity.
- h) It works on the major Web browsers such as Microsoft IE and Netscape so that it does not need additional software.
- i) It can works not only as Java stand-alone application but also as Java applet.
- j) It is independent from kinds of operating systems and machine-specific environments, namely available in multiple platform environments.

Some of the above concepts are not original ideas of us nor not specific for our simulator. They are efficient ideas for designing educational tools, of course, and effective methods for usage of such tools in the real education. It is our original approach to develop such a visual simulator with Java for educating the internal behavior and structure of computer. Visualization is one of the most important concepts for the real education scheme. They say "Visualization Methods for

Learning" seem to be available and fruitful for various fields of education.

A simple GUI is necessary and effective for users, especially beginners, to manipulate such an educational tool. Button-based operation is simple and easy to use, where the 4 major buttons are for initializing system, loading program, and executing in the two ways described later. Figure 1 shows an overview of GUI of our visual simulator. So a useful educational tool will tend to be one of Interactive Learning Systems.

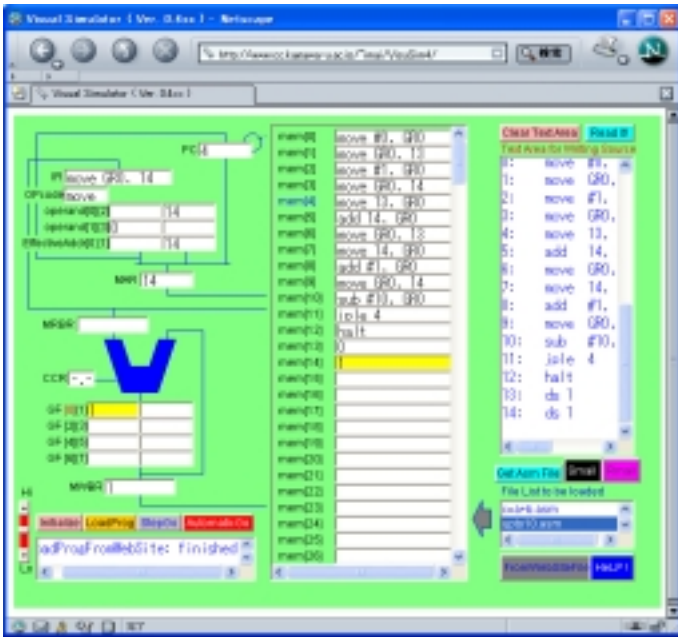


Fig. 1. Overview of the Visual Computer Simulator

The Java programming language is one of the most convenient and powerful languages for system description of GUI as well as network programming. And Java applet is suitable to realize a smart educational tool, which can be downloaded from a specific Web server and work on a major Web browser. Such a tool does not need additional software and specific environment for its execution. That is one of the very efficient procedures to deliver an educational tool to the students by means of universal mechanism of network connectivity.

An educational tool is sometimes used for teachers to demonstrate several concepts and ideas in their classroom lecture. Otherwise, it may be used for students to resolve their problem or homework even at their home or out of school. In the former case, the tool is utilized as teaching support equipment and operated by teachers who invoke it on his PC. It is good that the tool is invoked as native application software. So it is useful to be invoked as a Java stand-alone application.

In the latter case, on the other hand, the tool is manipulated as self-teaching software downloaded from the Web server. It is suitable that the tool is invoked as a special application on the students' PC with no other supporting environment. Therefore,

it is better to be invoked as a Java applet. Figure 1 shows an example of our visual simulator which is invoked as a Java applet. That is a reason for us to design our simulator which can work as both Java applet and Java stand-alone application.

III. FACILITIES OF VISUAL SIMULATOR

Our visual simulator is called "VisuSim". Its several facilities are introduced in this section. Simulator "VisuSim" has two kinds of execution modes of program: one is stepwise and other is automatically continuous. As a function of our simulator, instruction sets for simulating computer are described later. Sample programs are useful for users of "VisuSim". So it is very important how to distribute sample programs for "VisuSim". And then other facilities are explained in order to understand "VisuSim" more precisely.

A. Stepwise and Continuous Execution

"VisuSim" provides two different kinds of simulation modes: one mode is step-by-step execution of program (called "stepwise mode" for short) and another is automatically continuous execution (called "automatic mode" for short). The former is convenient to trace program step-by-step and find illegal points or wrong description of program. The latter is helpful to execute a series of code block continuously in order to confirm whether a whole program works correctly or not. Two kinds of simulation modes can be selected as users specify which buttons for execution, shown in Figure 2.

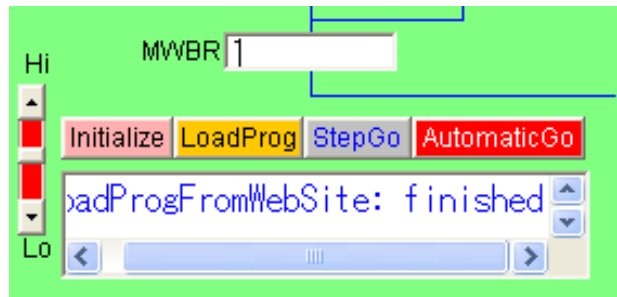


Fig. 2. Buttons for Selecting two kinds of Execution of "VisuSim"

Each execution mode is summarized as follows:

(1) Stepwise mode execution:

When the button for "Step Go" is clicked, the internal simulator module of "VisuSim" activates only one machine cycle execution. A machine cycle of this mode strictly consists of instruction fetching, instruction decoding and execution. Sequential execution of the specific part of program needs consecutive operations of clicking the bottom for "Step Go". This simulation mode is suitable to modify contents of registers, memory and so on interactively while an execution is broken (i.e. stopped). Namely, it is convenient to stop execution of program at any point, change the current state (i.e. contents of several registers, memory and so on), and then restart from such a previous broken point.

This mode could make "VisuSim" compute only the limited part in the long size code of program, if program counter is set to the specific location and the button for "Step Go" is being clicked during the suitable times. Such a process can point out the illegal description of programs, offer a better modification for improving, and finally prove code consistency of those examined programs. A detail testing of program can be realized in this mode. For example, it is useful to illustrate how computer works.

(2) Automatic mode execution:

When the button for "Automatic Go" is clicked, the internal simulator of "VisuSim" starts on continuous repetition of machine cycle execution until decoding halt instruction or occurrence of interruption. This mode is much useful for a relatively long-term demonstration of executing program where "VisuSim" must interpret a program including, for example, several iterations, recursive procedure calls or other kinds of complicated processes.

Because this mode of execution looks like playing slide show, it is helpful to trace a whole target program sequentially and verify whether program is computed accurately or not. While "VisuSim" repeats its machine cycles in the automatic execution mode, pushing the button for "Automatic Go" can trigger occurrence of interruption. Figure 3 shows such a situation. At the interruption of "VisuSim", pushing such a button again makes "VisuSim" begin to restart its machine cycles from the very interrupted point.

Speed control is available for slide bar to be varied from higher level to lower one, shown in Figure 3. Each personal computer has various CPU with different type of frequency. It is useful to adjust simulation speed of "VisuSim" among its different execution environment.

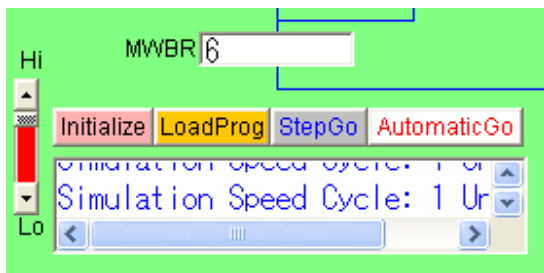


Fig.3. Speed Control Slide Bar and Interruption of Execution

B. Functions of Simulation

Structure of program is sometimes discussed in a lecture of computer literacy including an assembly programming exercise. Of course, 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 students sometimes do not understand computer behavior only through high-level languages 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 and so on.

Table 1 shows a brief explanation of Instruction set for VisuSim. Instruction sets of "VisuSim" are separated into the following 4 groups: control instructions, jump instructions, unary operation instructions, and binary operation instruction. Typical addressing of it includes direct addressing, indirect addressing or immediate value addressing.

TABLE I
INSTRUCTION SET TABLE FOR "VISUSIM"

Beginners sometimes suffer from lack of a suitable educational tool which visualizes how a computer processed a complicated iteration. For example, it is very useful to illustrate a concept of effective addressing with such a tool, because teachers frequently meet difficulty to show a schematic distinction between direct addressing, indirect one and

	Opcode	1st Operand	2nd Operand
Control Instruction	halt		
	noop		
Jump Instruction	jump	Direct Address	
	jpgt		
	jpge	Register Id	
	jplt		
	jple	IndirAddr(RegId)	
	jpeq		
	jpne		
call			
	ret		
Unary Operation Instruction	neg	Direct Address	
	push	Register Id	
	pop	IndirAddr(RegId)	
Binary Operation Instruction	add	Direct Address	Direct Address
	sub	Register Id	Register Id
	move	IndirAddr(RegId)	IndirAddr(RegId)
		Immediate Value	
Instruction	and	Direct Address	Direct Address
	or	Register Id	Register Id
	xor	IndirAddr(RegId)	IndirAddr(RegId)
		Immediate Value	

immediate value one graphically.

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/or heap. Recursive subroutine calling 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 students of information engineering to have a graphical explanation of internal behavior and structure of computer. So it will be very effective for such students if a suitable educational tool is easy to illustrate how a stack area grows and reduces while program is executed in computer. With such an educational tool, the students can understand precise mechanism of subroutine call through an assembly programming exercise.

Our visual simulator, "VisuSim" can provide several kinds of illustration about the internal behavior and structure of computer, which includes the following themes: effective addressing, iteration handling, parameter passing, stack growing/reducing, and so on.

Generally speaking, numbers of statement are not so many in the case of assembly program. But the case of recursive programming needs capacity of memory space so that "VisuSim" can provide either 256 or 512 words for its memory space according to its executing environment, before it is invoked.

C. Getting Sample Programs and Online Help Message

"VisuSim" can be used as an educational tool for assembly programming environment as well as demonstration of internal computer behavior. In the case of using "VisuSim" as an assembly programming tool, a user writes an assembly program directly into the field "Text Area for Writing Source".

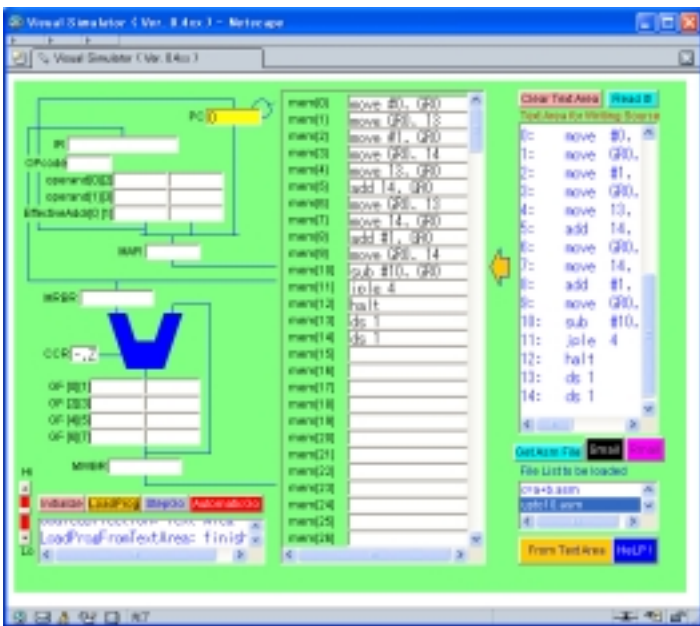


Fig.4. Writing Program into "Text Area for Writing Source"

Of course, at first, it can write a program with some kind of editor and then copy that program onto that "Text Area" of "VisuSim". Initially, "VisuSim" does not read a program from that "Text Area" as a default setting. So the user had better click a toggle button in order to change the direction of loading program into the suitable situation where the content of that "Text Area" is automatically transferred into memory space of

"VisuSim". The toggle button may be labeled as "FromWebSiteFile" or "From Local File". That button is labeled as "FromWebSiteFile" when "VisuSim" is invoked as a Java applet. Otherwise, that button is labeled as "From Local File" when it is invoked as a Java stand-alone application.

In the case of using "VisuSim" as a demonstration tool, especially, a user thinks that it is one of the useful and convenient facilities for "VisuSim" to search sample program files at possible places, list them up on the GUI window, select the suitable one and load it into the memory space of itself. As "VisuSim" works not only as a Java stand-alone application but also as a Java applet, it needs special file-obtaining facilities. They are implemented with a series of the following operations, which are divided into the next two modes:

(1) When "VisuSim" is invoked as a stand-alone application, the above facilities are realized as follows:

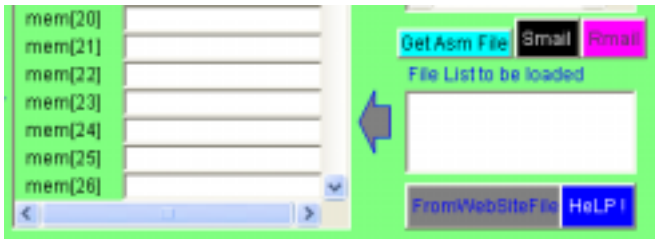
1. Searching the files named "*.asm" at the directory of the local host where "VisuSim" is stored, when the button "Get Asm File" of the GUI is clicked.
2. Getting file information.
3. Updating the content of the list field "File List to be Loaded" according to the information.
4. Selecting a specific assembly program file when the content of the list, when it is pointed by mouse.
5. Loading the program from the file into memory space of "VisuSim", when the button "LoadProg" is clicked.

(2) When "VisuSim" is invoked as a applet, the file-obtaining facilities are realized as follows:

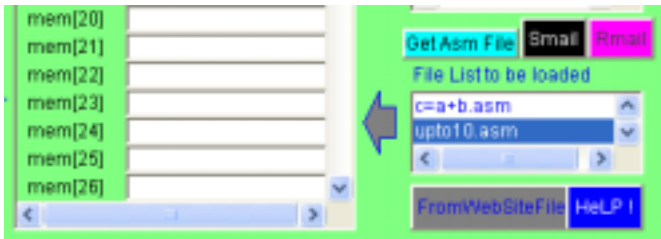
1. Sending an HTTP-based request for searching the files named "*.asm" is transferred into the Web server specified with a param-tag for the URL in the HTML file, when the button "Get Asm File" of the GUI is clicked. In this case, the list field "File List to be Loaded" is empty shown in Figure 5(a).
2. Getting information of files through an HTTP-based reply from the Web server.
3. Updating and selecting are the same processes as described above. In this case, the list field "File List to be Loaded" is updated with assembly program files shown in Figure 5(b).
4. Reading the content of selected program file from the web server to a buffer of "VisuSim" through an HTTP-based connection, when the button "LoadProg" is clicked.
5. Loading the program content from the buffer into memory space of "VisuSim" after the above reading.

The same mechanism has been employ for "VisuSim" to get any kind of file from the directory of the local host or from the Web server just like the above operations. In the situation to get the file as online help message, "VisuSim" can obtain the suitable file either from the directory of the local host or the Web server. If a teacher carefully prepares an HTML file for invoking "VisuSim", he/she can customize a more-fitted environment for students by means of providing the suitable

level of sample programs and online help messages.



(a)



(b)

Fig.5. Searching Assembly Program and Display the Results

The modes (1) or (2) can be automatically selected by "VisuSim" itself according to the working status for invocation as an applet or as a stand-alone application. So users do not need to take any notice of the above modes.

sources, and the later is effective to indicate how to operate "VisuSim" by means of guideline message display which is generated by mouse-cursor positioning at the button or the text field of "VisuSim". Figure 6 shows an online-help-message window which is generated by clicking the button "HeLP" of "VisuSim".

Guideline display is desined to explain a purpose and usage of each button or field in the GUI of "VisuSim". Display of guideline may be effective and important which explains purpose of each feature in the GUI and indicates how to use it. Figure 7 shows an example of guideline prepared in VisuSim whose message can be invoked by means of moving mouse cursor close to the position of target feature of GUI.



Fig.7. Guideline Function of "VisuSim"

Users such as a teacher and students can utilize "VisuSim" efficiently and effectively through equipment with sample programs, smart transferring mechanism of those programs, display of online help message and guideline generation.

D. Other Facilities

"VisuSim" has two e-mail handlers as its built-in modules. One of the e-mail handlers is an e-mail sending module as an SMTP-client and another 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, which has two characteristics as follows:

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

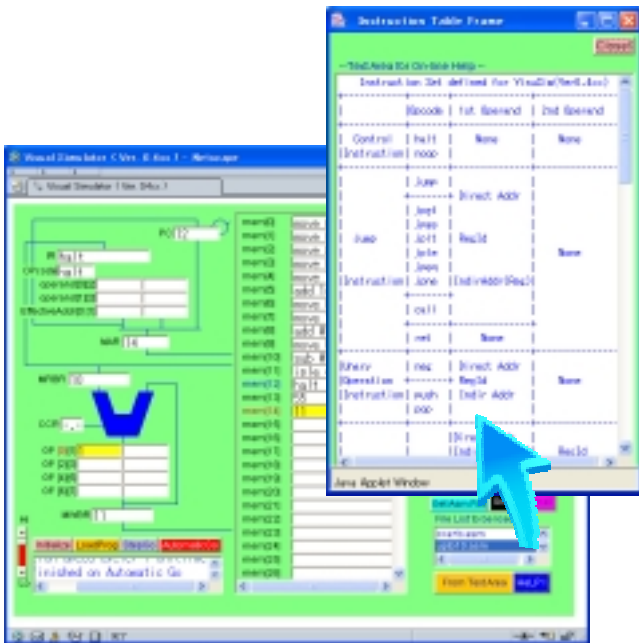


Fig. 6. Online Help Window overlapped on "VisuSim"

Both of online help message and guideline display are also useful for students to manipulate "VisuSim" after school or at home. The former is helpful to give students some kinds of concrete advices how to describe assembly programs by means of showing instruction set table and some sample program

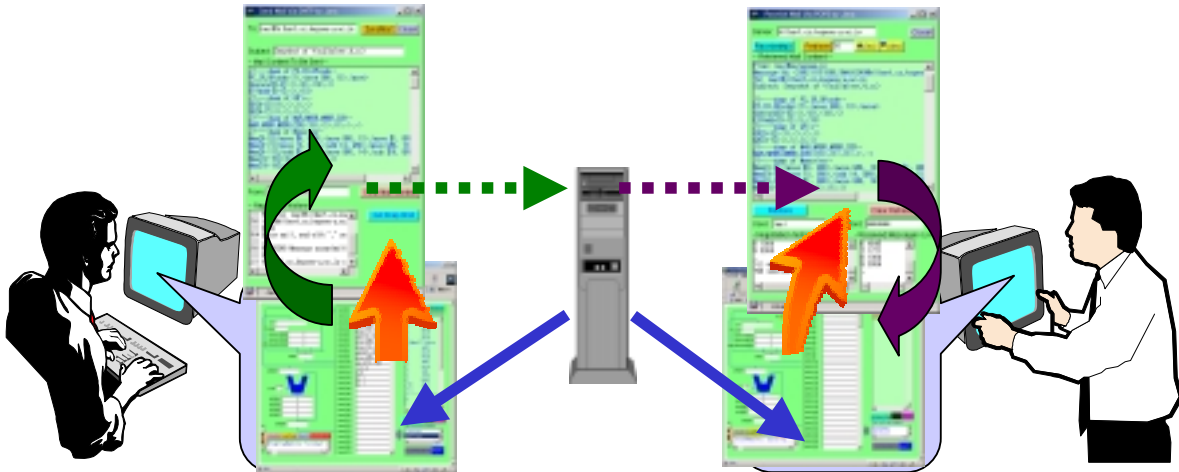


Fig.8. Communication Support of "VisuSim" and Sharing Information between Student and Teacher

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 (i.e. contents of registers and memory in the specific "VisuSim") between some users.

For example, when a student, user of "VisuSim", has met problems and try to resolve them with any help from teacher, he can invoke the SMTP-client of "VisuSim", save such the internal state into a mail text, and send the e-mail with such information to teacher. The left half of Figure 8 shows the above procedures.

A teacher, another user, shown in the right half of Figure 8, can receive the mail text and gain a resurgence of duplicated image of "VisuSim" from the 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 of "VisuSim" between users: namely, student and teacher. This is a powerful facility to check the remote state of "VisuSim" on the local one.

In the SMTP-client, "VisuSim" provides a special function called "Snap Shot", which can duplicate the internal state into the mail text. In other words, users can store the internal *image* of "VisuSim" as mail content by means of clicking a button labeled "Get Snap Shot". Such a stored image is treated as the backup copy of "VisuSim".

On the other hand, just like SMTP-client, the functions of POP3-client of "VisuSim" are reading the e-mail from other "VisuSim". The POP3-client can receive the e-mail text with an internal state from other "VisuSim", restore such an information to the internal state of the receiving "VisuSim" itself, and gain a resurgence of the remote "VisuSim" on the local one. Such operations are easily carried out by means of clicking a button labeled "Restore" only.

Second of the facilities is to permit specification of e-mail address with IP address. In the normal case, an e-mail address is represented with domain name. Using "VisuSim" as an educational tool 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 without 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 IP-based special one as an e-mail address. The former is like as "imai@eng.kagawa-u.ac.jp" and the later is like as "imai@#192.168.1.1", for example. It is very convenient to utilize an experiment server with no domain name and begin very much easily a trial (or short-term) service of "VisuSim" without complex and time-consuming adjustment of the LAN administrator.

Both the SMTP-client and the POP3-client are designed 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.

IV. APPLICATION OF SIMULATOR TO REAL EDUCATION

Real applications of "VisuSim" are introduced in this section. First of all, "VisuSim" is used as an educational tool for not only me but also other teachers in their classroom lectures. Usage in assembly programming exercise is another target for "VisuSim". And "VisuSim" can provide an environment for collaborative learning in the final case of our introduction.

A. Real Education with Visual Simulator

Figure 9 shows an image of real education with our visual computer simulator, "VisuSim" for the lecture of computer literacy. It is displayed on the screen of the lecture room. Students of the class can understand computer behavior and structure while operating "VisuSim" step by step.

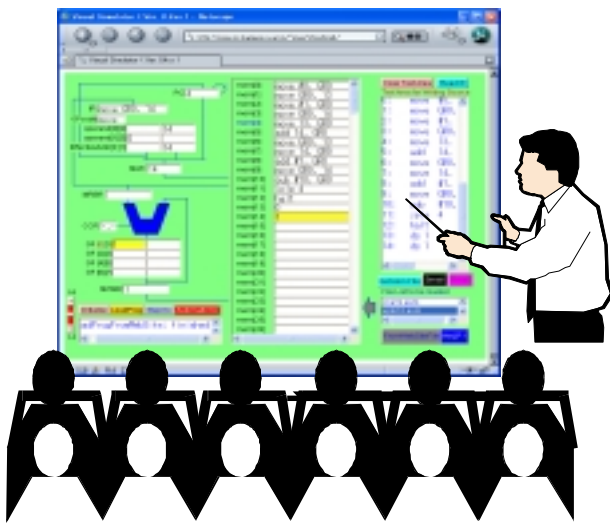


Fig.9. Usage of "VisuSim" as a Demonstration Tool

It is one of the most popular cases that a teacher utilizes "VisuSim" as a teaching system in the class of computer literacy in order to illustrate how a computer works. With a projector and a large screen, the teacher directly shows students the output of PC where "VisuSim" is operating.

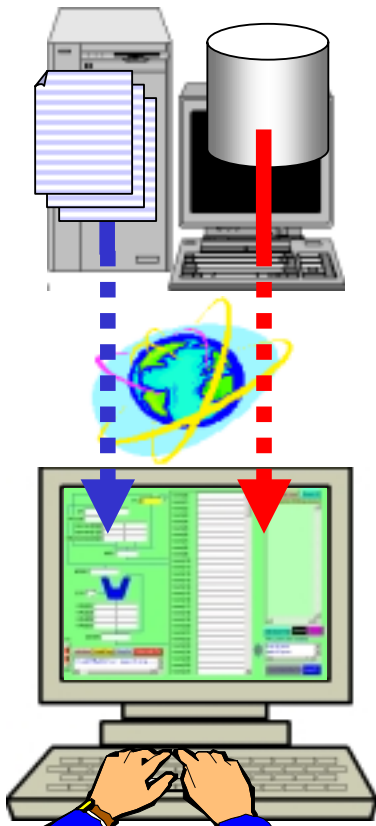


Fig.10. Another Typical Usage of "VisuSim".

In another case, students download "VisuSim" from the specific Web server, invoke "VisuSim" on their PCs, let it simulate their programs and investigate whether their programs work correctly or not. In such a situation, it is very useful to compute a program by means of stepwise and/or continuous executing facilities of "VisuSim".

Some sample assembly programs are prepared and downloadable from the same Web server. They are stored into buffer area of "VisuSim" and loaded into memory space. Figure 10 shows a typical usage of "VisuSim" as a Java applet.

B. Recursive Programming Facilities

Assembly programming capability is essential for students in the course of information systems engineering. This capability is obtained from not only class lecture but also practical exercise. From our Web server, students obtain other examples of assembly programming such as selection with several conditional jump instructions, repetition with indexed addressing, and so on.

Recursive call is one of medium-level themes about assembly programming. For example, calculation of factorial and resolution of Hanoi's tower are suitable themes for students to learn the basic recursion of assembly programming. Because stack-related instructions are equipped, "VisuSim" provides a suitable environment to write an assembly program related with recursive subroutine call.

Consequently, a powerful debugging facility is available through combination of two execution modes. In the class of assembly programming exercise, such a debugging facility is employed to verify whether programs work correctly or not. It is also effective to explain the correspondence between description of high-level language and behavior of computer. "VisuSim" can illustrate graphical behavior of computer, for example, the relation between numbers of repetition and the content of loop control variable in the specific general-purpose register.

C. Application to Collaborative Learning

Other applications of "VisuSim" are to be used as an environment for collaborative learning. An actual situation is introduced as follows:

1. Students want to try their homeworks after school, use "VisuSim" downloaded from the Web server, and then meet some problems.
2. If they cannot resolve such problem by himself, they decide to correspond with their teacher about their problems.
3. They invoke the SMTP-client of "VisuSim" to utilize the "Snap Shot" facility in order to save the current states of "VisuSim" as the e-mail content. It is transferred to the mail server (the same machine of Web server) through an SMTP-client of "VisuSim".

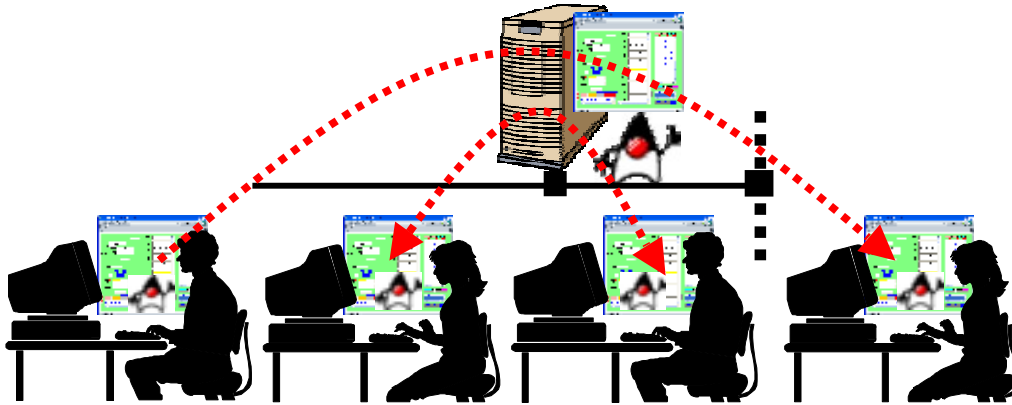


Fig.11. Collaborative Learning with Visual Computer Simulator called "VisuSim"

Every teacher wishes to teach effectively his/her students how to resolve problems. And his/her students want to share the results and thought about the problems with the teacher. Without such a teaching mechanism, there might be the following difficult problems between a teacher and students:

- How to transfer the identical situation from one to another,
- How to recreate the same image on the remote machine,
- How to answer the real result with comments from a teacher to students and so on.

In our case, however, a teacher can share the same image of "VisuSim" with his/her students by means of its communication support facilities and smart GUI. So teacher and students are blessed with a useful environment for collaborative learning.

There is another case to learn assembly programming in a collaborative learning environment, which is realized with "VisuSim" through network connectivity shown in Figure 11. It will be almost free for location and occasion. Assembly programming exercise is suitable for students to understand the behavior and structure of computer through their programming process. For example, a pair of students is obliged to work together to accomplish a common goal under the following conditions:

- Students have to rely on each other to achieve the goal.
- If any one student fails to do his/her part, both of them suffer consequences.

It will be a good example of useful interdependence. In such a case, each student is held accountable for doing his/her share of given problems and for mastery of all of the jobs to be learned. In order to exchange several information between each other frequently, facility of "VisuSim", such as function of communication support, is very efficient and effective for a pair of students to study and exercise assembly programming by themselves in a collaborative learning environment.

Through the process of teaching each other, both of skill and knowledge of students are getting wider and more applicable. And moreover, such a collaborative learning will lead to useful teamwork spirits and cooperative thinking. Those are very important for ICT-based Higher Education, especially, in Japan.

V. CONCLUSION

Our visual simulator called "VisuSim" has been used for ICT-based education from computer architecture to assembly programming. It has several facilities, such as a single pure Java source code for Java applet as well as stand-alone application, simulation functions with two ways, a smart GUI to use, and communication support between users. And its applications are introduced, for example, a demonstration tool to educate von-Neumann-style computer architecture and collaborative learning environment for assembly programming exercise.

Characteristics of "VisuSim" are summarized as follows:

- "VisuSim" is used as a demonstration tool in the classroom lecture to illustrate the internal structure and behavior of a computer and explain graphically how a computer works.
- Online help message and guideline display are useful for beginners to utilize "VisuSim" all by themselves.
- Built-in mail handlers are effective to communicate with each other by means of sending/receiving a program and internal state of "VisuSim".
- Scheme of a collaborative learning is proposed which is performed with communication support facilities and smart GUI of "VisuSim" on the network environment.

ACKNOWLEDGMENT

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 grand-in-aid for scientific research from the Ministry of Education, Culture, Sports, Science and Technology under the 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] C. O'Malley (ed.): Computer Supported Collaborative Learning, Springer-Verlag, 1994, ISBN 0 387 57740 8..
- [3] T.S.Roberts (ed.): Computer-Supported Collaborative Learning in Higher Education, Information Science Publishing, Hershey, Pennsylvania; 2004, ISBN 1-59140-408-8