

Distance Education System for Real Experiments over the Web Integrated by Swing and Java3D – β -ray Radiation –

¹Hiroshi Nakano, ²Takayuki Mizutani, ¹Yasuyuki Nakamura and ¹Susumu Matsuo

¹Division of Informatics for Natural Sciences, School of Informatics and Sciences,
Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601 JAPAN

²Division of Informatics for Natural Sciences, Graduate School of Human Informatics,
Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601 JAPAN

Abstract

This paper presents a distance education system for the interactive experiments on physics over the World Wide Web integrated by Swing and Java3D API on Java2. Our system provides the real experiments over the web and are applied to the subject of experimental physics; the β -ray radiation. Experiments with radiation sources are sometimes kept away from educational courses because the radiation source must be controlled carefully in spite of its importance. With our system, students can measure counts of β -ray without treating radiation source directly at any time from any places over the web. The client programs run as Java applets on web browsers or as Java applications. Using the Swing API, they have a common desktop environment that provides exactly the same graphical user interface with the internal window system on any operating system supporting Java2 and Java3D. The 3D view window of the experimental equipments is realized by Java3D API and is very helpful to understand the configuration of equipments and principles of the measurement. The client programs handle the TCP/IP communication to the server program written in Java with native methods on the Linux server built with general PC and experimental equipments.

key words: distance education, real experiments, physics, web, Java3D, internationalization

1 Introduction

Distance educations (DE) for physics are studied by many teachers, schools, and universities and so on through the spread of the World Wide Web. A lot of works of Java applets have been developed [1–3] and almost all applets are simulation programs that are effective and helpful in understanding physical phenomena. There are, however, few works of experimental physics over the web [4–8]. Of course, it is very important to study experimental physics using the equipments in front of students by their hands. We think that DE systems for experimental physics have also some advantages; they are ready at any time from any places over the web and are very effective for some themes that have problems in small schools because of the scale of experiments or difficulties of treatment.

We are providing the interactive experiments

on two subjects of physics; the β -ray radiation and the electric circuits, over the World Wide Web. [9] We are introducing here new system for the β -ray radiation experiment [7] integrated by Swing [10] and Java3D [11] API on Java2 [12]. These kinds of experiments are sometimes kept away from educational courses because the β -ray source should be controlled carefully in spite of its importance, not only for the physics interests but also for the natural laws of probability.

We tried to solve here three problems of such DE systems for remote experiments; imaging the construction of the remote experimental equipments, exactly the same usage on any web browser and operating system, and the internationalization. It is very helpful for imaging the experimental configuration to move an eye-point freely in a 3D view. The internal frames on a common desktop provide the exactly same window system. The internationalization can be done using the locale-

sensitive objects and the Unicode system [13].

We adopted several new technologies to our system; Swing API on Java2 for the internationalized general graphical user interface (GUI) on the common desktop, Java3D API for the real 3D view of the remote experimental equipment, Java [14] for the clients and the server programming with native methods, the Linux GPIB package [15] for the communication to the experimental equipments, the cqcaml driver [16] for getting real-time images, the Apache http server [17] on the Linux [18] system for the stable operating system and many GNU [19] tools for developing software.

References 7 on ISIMADE'99 shows the basic hardware construction and server programs of the system. This paper shows new client program that provides exactly the same GUI using the internal window system on many operating systems, and provides the 3D view window of the experimental equipments helpful to understand the configuration of equipments and principles of the measurement, and adopts the internationalized messages referred to the locale.

2 Experiment on β -ray Radiation

Our DE system for the experiments on β -ray radiation provides the real experiments over the World Wide Web and requires Java2 (including Swing API) with Java3D API executable environment for the clients. The client program can run as an applet on the web browser or a Java application.

When clients access to the server by the web browser, they get a Java applet on the screen as shown in Fig.1. The applet has several internal windows; *Control Panel*, *3D View of Equipment*, *View (small)* (a small live image), *Beta-ray (Al No...)* (raw experimental data for several conditions), *Distribution (Al No...)* (distribution of β -ray counts for several conditions), *Absorption by Al plate* (Al Absorption plate dependence of averaged β -ray counts).

The client can set the conditions of the experiments by the *Control Panel*; the number of measurements N and the thickness of the Al absorption plates placed between the β -ray source and the Geiger-Müller (GM) counter. We use a radioactive isotope $^{90}_{38}\text{Sr}$ and $^{90}_{39}\text{Y}$ as a source and detect β -rays $^{90}_{38}\text{Sr} \rightarrow ^{90}_{39}\text{Y} + e^-$, $^{90}_{39}\text{Y} \rightarrow ^{90}_{40}\text{Zr} + e^-$.

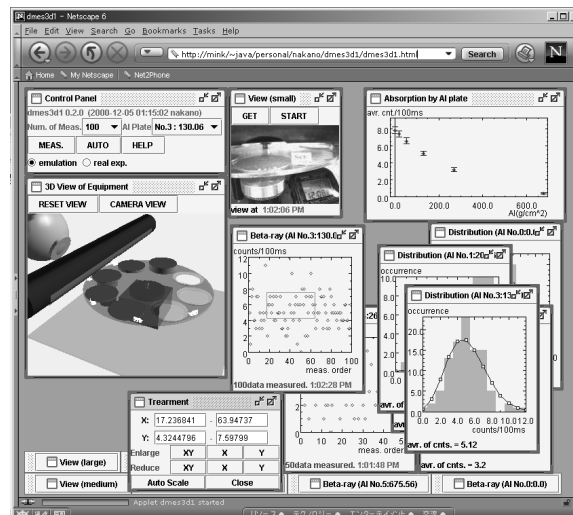


Figure 1: The screen shot of the β -ray experiment on the web browser.

When the *MEAS.* button is pushed, the client sends the start command to the server and the measurement will start. During measurement the server is sending counts of β -ray per 100 ms for N times and the client start to plot data on the graph with auto-scaled axes in the *Beta-ray* window. At the end of the measurement, *Distribution* window appears or comes up at upper layer and shows the graph of the distribution with a fitting curve of the Poisson distribution. One data point will be plotted with an error bar on the window of *Absorption by Al plate*. During the measurements the *MEAS.* button is changed to the *CANCEL* button that will interrupt any measurement and show the results until the time. The *AUTO* button provides the fully automatic measurements changing all Al absorption plates. We also provide the emulation mode, which simulates the program of real experiment, and it is convenient to get a feel when other user occupies the system or the network has troubles.

The thickness of the Al absorption plate can be changed by selecting the chooser on *Control Panel* window. When Al plate is changed, the live images on the *View (small / medium / large)* window will be reloaded if the window is active. The position of Al plate in the 3D view on the *3D View of Equipment* window will also be changed. Figure 2 shows the examples of small and medium live images and 3D View when No.5 the thickest plate Al plate is selected. The *View (small / medium / large)* windows provide the three different sizes of

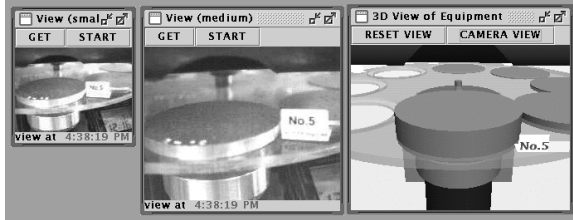


Figure 2: The small (left) and medium (center) live images and the 3D View (right).

the images and slow animations, which reload the image at intervals of 15 s when *START* buttons is pushed.

This experiment helps students to understand that the distribution of β -ray obeys the Poisson or binomial distributions and the absorption of the β -ray, which comes through Al plate, depends on the thickness of the plate. Figure 3(a) and (b) show the example of N dependence of the distribution roughness and students will get a different distribution in the same condition with similar roughness. The difference between Fig.3(b) and (c) shows the Poisson distribution with different mean values caused by the absorption of β -ray. All of these results can be obtained by the real experiments over the web.

3 3D Visualization

The core part of β -ray measurements consists of a β -ray source, a GM tube detector, a turntable with five Al absorption plates, a detection switch and small video camera as shown in Fig.4. [7] It is not easy to imagine the construction of this small laboratory only with this photograph. The correspondence between the laboratory and the images obtained by the small camera is not shown clearly.

A 3D view window of the small laboratory is implemented by using Java3D [11] API on Java2 as shown in Fig. 1. The eye-point can be moved freely by dragging a mouse. The 3D view may therefore be rotated, zoomed or shifted as shown in Fig. 5. It is very helpful to imagine the construction of the laboratory and to understand the principle of the measurement. The present position of turntable with Al absorption plates is reflected in the 3D view window, and it is also helpful to imagine what happens in the remote laboratory. The 3D view is drawn by the client, which means that the server has no load and the

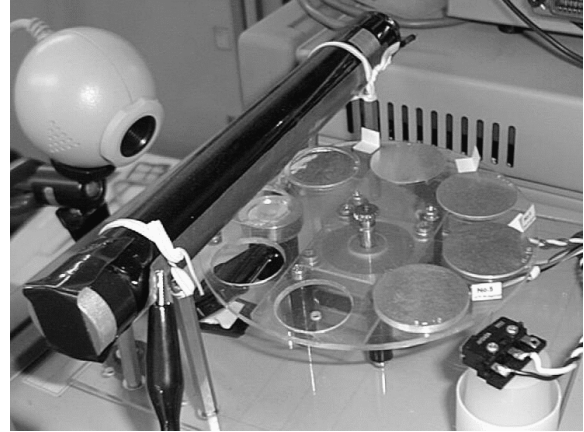


Figure 4: The small laboratory of β -ray measurements.

network traffic is much fewer than the video transmission.

4 The Common Graphical User Interface

The client program has a common desktop and all windows belong to this desktop. Every window is implemented by the `JInternalFrame` class of Swing API [10]. The `JInternalFrame` class provides exactly the same GUI environment including the window's frame on any operating system or browsers which support Java2. The internal windows have the same type of frames as shown in Fig. 1. Figure 6(a) shows the two types of windows where the treatment window *Treatment* works like a dialogue. We use the `JInternalFrame` instead of `JDialog` class for *Treatment* to realize the same look and feel.

The AWT components are slightly different in the operating systems and the usual frames completely depend on the window system. The Swing components within the `JInternalFrame` class are therefore very powerful to construct such a common GUI environment.

The Swing components are normally lightweight and cannot be mixed with heavyweight components. The `Canvas3D` class implements the 3D view, which is a heavyweight class. The internal frame for the 3D view is therefore always kept at the top layer of the desktop and the `JComboBox` components on the Control Panel are modified to the heavyweight component by the method of set-

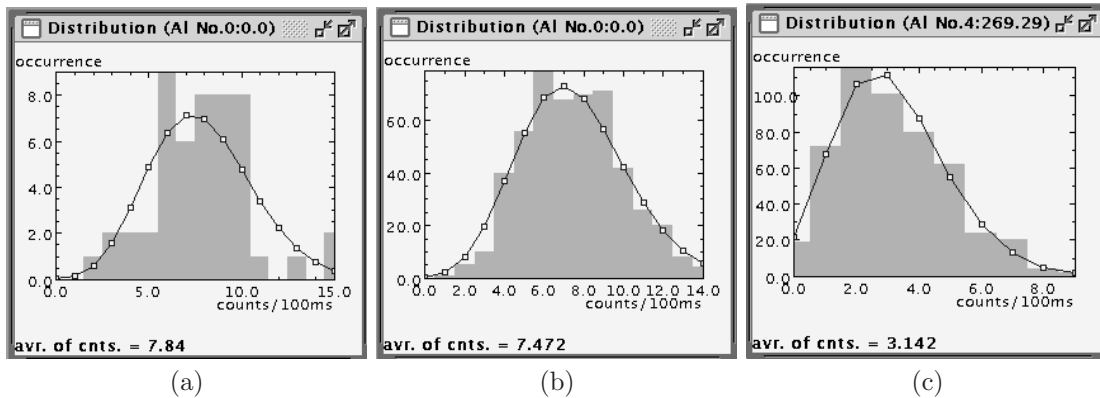


Figure 3: The results of β -ray experiments in various conditions: (a) 50 measurements without Al absorption plate, (b) 500 measurements without plate, (c) 500 measurements with the No.4 rather thick plate.

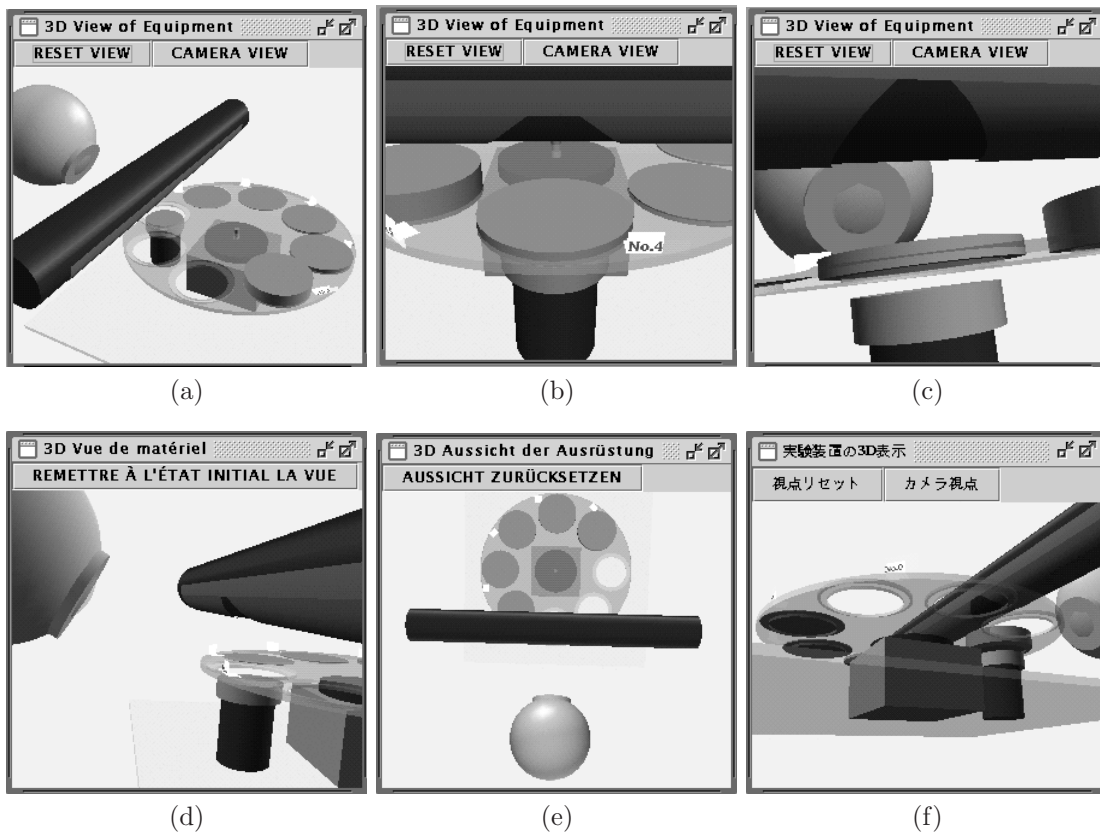
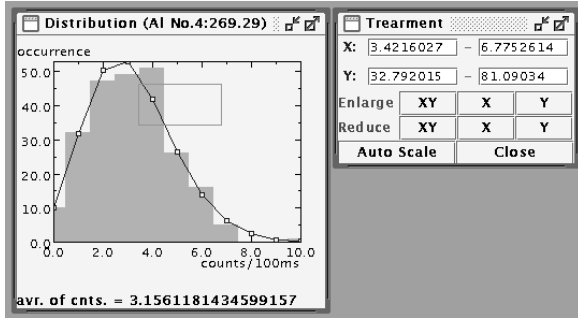
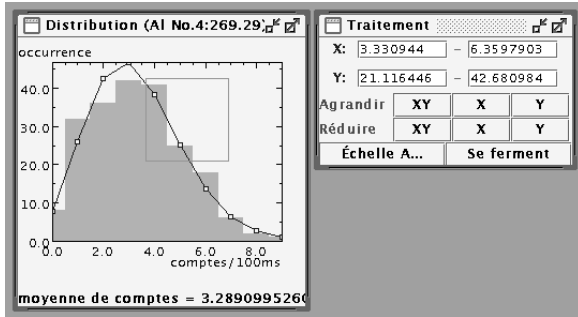


Figure 5: 3D views of the small laboratory from various eye-points.



(a)



(b)

Figure 6: Internal windows for the distribution graph and the treatment with (a) the English locale and (b) the French locale.

LightWeightEnabled(false) in order to avoid the conflict with the Canvas3D. This problem may be solved by the improvement of the both Swing and Java3D APIs in near future.

5 Internationalization

All of messages on labels, buttons, menus, axis names and so on are internationalized using the local-sensitive objects on Java2. Figure 7 shows the screen shot of the web browser with the Japanese locale. The strings are completely translated to Japanese compared with ones in Fig. 1. Figure 5(f) is also an example of the Japanese locale. Figures 5(d) and 6(b) are browsed with the French locale and Fig. 5(e) for the German local.

It is very easy to internationalize Java2 program using the code as shown in Fig. 8 [12]. The first line sets the default locale to variable `alocale`, 2nd line constructs a filename (`messages`) from the strings of "MessagesBundle" and `alocale`. When the client computer has a locale for the English language in the United States, the filename will become "MessagesBundle_en_US.properties"

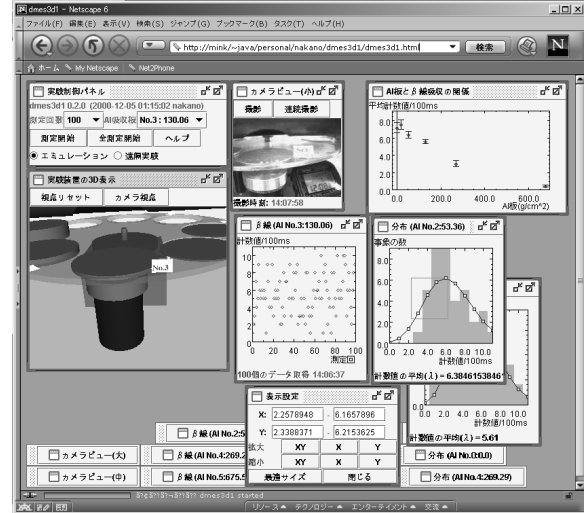


Figure 7: The screen shot of the β -ray experiment on the web browser with the Japanese locale.

(Fig. 9 left). In the case of the French, the German and the Japanese locales, the filenames are respectively corresponding to "MessagesBundle_fr_FR.properties", "MessagesBundle_de_DE.properties" and "MessagesBundle_ja_JP.properties" (Fig. 9 right) which are converted to the Unicode [13] by `native2ascii` command. The parts of strings "en", "fr", "de" and "ja" are language code, a pair of lowercase letters that conform to ISO-639, and "US", "DE", "FR" and "JP" are country code which consists of two uppercase letters and conforms to ISO-3166. Line 3 and 4 create a label and a button object with internationalized strings. Only first 2 lines are necessary to internationalize Java2 programs. These internationalized strings can be modified without recompiling the sources.

6 Conclusions

We developed the interactive experiment system on β -ray radiation over the World Wide Web in any time from any place. Students can do real experiments in setting the experimental conditions and receiving the graphically plotted experimental data on exactly the same GUI environment using the internal frames of Swing API. Natural random phenomena can be studied quantitatively on the obtained distributions. Students can change the absorption plate between the β -ray source and the


```

Locale alocale = (new java.util.Locale("en", "US")).getDefault();
String messages = ResourceBundle.getBundle("MessagesBundle", alocale);
JLabel label = new JLabel(messages.getString("ctl_n"));
JButton button = new JButton(messages.getString("ctl_meas"));

```

Figure 8: Example of the source code of internationalized Java2 program.

<pre> ctl_win = Control Panel ctl_n = Num. of Meas. ctl_al = Al Plate ctl_auto = AUTO ctl_meas = MEAS. ctl_cancel = CANCEL ctl_help = HELP ctl_exp = real exp. </pre>	<pre> ctl_win = 実験制御パネル ctl_n = 測定回数 ctl_al = Al吸収板 ctl_auto = 全測定開始 ctl_meas = 測定開始 ctl_cancel = 測定中止 ctl_help = ヘルプ ctl_exp = 遠隔実験 </pre>	<pre> ctl_win = \u5b9f\u9a13\u5236\u5fa1\$ ctl_n = \u6e2c\u5b9a\u56de\u6570 ctl_al = Al\u5438\u53c5\u677f ctl_auto = \u5168\u6e2c\u5b9a\u958\$ ctl_meas = \u6e2c\u5b9a\u958b\u59c\$ ctl_cancel = \u6e2c\u5b9a\u4e2d\u6\$ ctl_help = \u30d8\u30eb\u30d7 ctl_exp = \u9060\u9694\u5b9f\u9a13 </pre>
---	---	--

Figure 9: Examples of properties files for locales; "MessagesBundle_en_US.properties" for English (left), Japanese native code (center) and the Unicode file named "MessagesBundle_ja_JP.properties" for Japanese (right).

GM tube detector while by verifying the state not only by the live camera but also by 3D view using Java3D API. The 3D view is also very helpful to imagine the construction of the remote experimental equipments and to understand the principle of measurements. The program is internationalized and shows messages in the browser's language where English, French, German and Japanese are supported now and it is easy to support other locales. The internationalization is very important for such a DE system.

Acknowledgments

The authors wish to thank Dr. S. Kashimoto, Professor K. Iwahashi and Professor M. Mori for their helps. The research project of School of Informatics and Sciences and Graduate School of Human Informatics supports this work. This study was supported by a Grant-in-Aid for Scientific Research from Ministry of Education, Science and Culture.

References

- [1] Gamelan Java Directory, Java: Educational: <http://gamelan.earthweb.com/javaprogramming/applets/dir.sciencephysics1.html>
- [2] F-K. Hwang, Virtual Physics Laboratory: <http://www.phy.ntnu.edu.tw/java/>
- [3] C. R. Wie, "Educational Java applets in solid state materials", IEEE Trans. Educ., Vol.41, No.4, 1998, pp.354.
- [4] A. Jochheim and C. Rohrig, "The Virtual Lab for Teleoperated Control of Real Experiments", Proc. 38th IEEE Conf. Decision and Control, Vol.1, 1999, pp.819-824.
- [5] M. Chirico et al. "The real experiment execution approach to networking courseware", IEEE Trans. Educ., Vol.40, No.4, 1997, pp.297.
- [6] F. Ruiz et. al, "Cooperative distance learning with an integrated system for computer assisted laboratory work", Third International Conference, CALISCE '96. Proceedings, xiv+480, 1996, pp.220-7.
- [7] H. Nakano et al., "Distance Education for Interactive Experiments on Beta-ray Radiation over the Web", ISIMADE'99, Baden-Baden, Germany, 1999, pp.109-112.
- [8] H. Nakano et al., "Distance Education System for Interactive Experiments on Electric Circuits over the Web", ISIMADE'99, Baden-Baden, Germany, 1999, pp.113-116.

- [9] WWW address of these experiments:
<http://exp1.phys.human.nagoya-u.ac.jp/exp1/index-e.html>
- [10] K. Walrath and M. Campione, *"The JFC Swing Tutorial: A Guide to Constructing GUIs"*, Addison-Wesley, 1999.
<http://java.sun.com/products/jfc/>
- [11] H. Sowizral, K. Rushforth and M. Deering, *"The JavaTM API Specification"*, Addison-Wesley, 1998. <http://java.sun.com/products/java-media/3D/index.html>
- [12] M. Campione, K. Walrath, A. Huml and Tutorial Team, *"The Java Tutorial Continued: The Rest of the JDK"*, Addison-Wesley, 1998.
- [13] J. Aliprand et al., *"The Unicode Standard, Version 3.0"*, The Unicode Consortium (Editor), Addison-Wesley, 2000.
<http://www.unicode.org/>
- [14] M. Grand, *"Java Language Reference"*, O'Reilly & Associate, 1997.
- [15] T. Ecimovic, *"GPiB: Cool, It Works with Linux!"*, Linux Journal, No. 47, 1998, pp.61-63. The Linux Lab Project's official homepage: <http://www.llp.fu-berlin.de/>
- [16] R. Chapman and W. H. Carlisle, *"A Linux-based Lab for Operating Systems and Network Courses"*, Linux Journal, No. 41, 1997, pp.60-65. P. Reynolds, The cqcaml home page: <http://www.cs.virginia.edu/~patrick/cqcaml/>
- [17] B. Laurie and P. Laurie, *"Apache: The Definitive Guide"*, O'Reilly & Associate, 1999. Apache HTTP Server Project:
<http://www.apache.org/>
- [18] E. Siever et al., *"Linux in a Nutshell"*, O'Reilly & Associate, 1999.
The Linux Home Page at Linux Online:
<http://www.linux.org/>
- [19] C. DiBona, S. Ockman and M. Stone, *"Open Sources"*, O'Reilly & Associate, 1999. GNU Project web server: <http://www.gnu.org/>

Authors Short Bibliography

Hiroshi Nakano is an Associate Professor of the Division of Informatics and Natural Sciences, School of Informatics and Sciences, Nagoya University. He received a doctoral degree of science from Kyushu University, Japan, in 1987. His current research interests are distance education system for interactive experiments and simulations, scientific visualizations, and experimental physics (quasicrystals, fine particles and superconductors). At Nagoya University, he teaches computer-aided measurement, introduction to programming languages and general physics.

Takayuki Mizutani is a postgraduate student and has just received a master degree from Nagoya University, Japan, in 2001. He is interested in measurement system integrated by Java, JNI and Java3D.

Yasuyuki Nakamura has received a doctoral degree of engineering from Kyoto University, Japan, in 1995 and is a Research Associate of Nagoya University since 1995. He is interested in statistical mechanics and JAVA programming for physics education.

Susumu Matsuo has received a doctoral degree of engineering from Nagoya University, Japan, in 1975 and is a Professor of Nagoya University since 1989. He is interested in electronic and magnetic properties of metals and quasicrystals. He makes experimental and simulation studies in the properties.