Robotica (1997) volume 15, pp 99–103. © 1997 Cambridge University Press

A graphical robot language developed in Japan

Tamio Arai,* Toshiyuki Itoko,† and Hidetoshi Yago‡

SUMMARY

A graphical robot programming system has been developed. This system with a graphical interface is user-friendly and easy-to-learn for low-skill users. It has been developed as a prototype system under a project by the Japan Robot Association (JARA) since 1994. The system runs on a personal computer and consists of a graphical user interface and an editing system. It is designed for programming an arc welding robot in small batch production and is expected to provide low-skill users with a means to use industrial robots with ease.

KEYWORDS: Robot language; Graphical interface; Arc welding.

1. INTRODUCTION

From a viewpoint of investment returns the introduction of industrial robots into mass production works resulted in lower product costs. However, recently the market for industrial robots seems to have become saturated. On the other hand, the processing capability of personal computers and consequent software environments, especially in the field of graphical user interface, have remarkably progressed, so that they could play a key role in expanding the industrial robot market.[1,2]

The Japan Robot Association (JARA) established a Professional Committee of Software Development in 1994, responding to environmental changes and software, and circulated a questionnaire about the need for small to medium-size companies. The results showed that there was a strong need for a simplification of programming and elimination of special skills for robot programming.[3]

In order to meet this need, the JARA started a challenging project to develop technologies to realize a task-level[4] programming system with a graphical user interface with which a robot user can construct a program with ease and without any skill; such a system is desired but not yet realized for practical use.[5]

2. ROBOT PROGRAMMING SIMPLIFICATION PROJECT

The challenging project started as a 3-year project in 1995 to develop a robot programming simplification system, which runs with a graphical user interface on a personal computer. This project, called ‘‘Robot Programming Simplification Project’ is progressing under the Information Technology Consortium (ITC) as a project of the Information-technology Promotion Agency (IPA).

* The University of Tokyo (Chairman of the Professional Committee of Software Development of the JARA), Tokyo (Japan).
† Kawasaki Heavy Industries, Ltd. (Japan).
‡ Matsushita Industrial Equipment Co., Ltd. (Japan).

In the project, the target users are workers who are not familiar with industrial robots in small batch production works, especially in small to medium-size companies. The objective is to produce a means for workers to construct a robot program by using a platform of a personal computer with a graphical user interface at a high cost performance.

A schematic diagram of the robot programming simplification system is shown in Figure 1. The special features of this concept are as follows:

- Making it possible for users to teach the robot behavior not at a motion instruction level but at a task instruction level by a graphical interface.
- Automatic motion program generation from a task-level program and a knowledge database for robot programming and task accomplishment.

Recently, a graphical programming system, by which the users can make robot language programs, has been proposed.[6] But such a system seems merely to replace the robot programming language in texts by iconic symbols, while the robot program simplication system endeavours to reduce the teaching process by enhancing the programming language level from a motion instruction level to a task instruction level.

Consequently, it is necessary to automatically deploy a task-level program for a robot-language program, as used in present actual industrial robots; this necessitates advanced information processing.

3. PROTOTYPE SYSTEM

3.1 System development

A prototype of the robot programming simplification system has been developed by Kawasaki Heavy Industries, Ltd. and Matsushita Electric Industrial Co., Ltd. for JARA.

The project focused the development for the first year on ‘‘the efficiency of the task-level teaching for simplification of robot motion programming’’. The main target was a simplified task-level teaching method using a graphical interface. Therefore, simple pattern processing was employed to convert a task-level program to a motion-level program in the prototype.

The feasibility of the generated robot motion program has been verified by the prototype system, using a robot simulator. A method to generate efficient motion incorporating the operator’s skill will be developed in the future in the project.

3.2 System configuration

a. **Hardware configuration** The hardware configuration is shown in Figure 2. The teaching system is constructed

https://doi.org/10.1017/S0263574797000118 Published online by Cambridge University Press

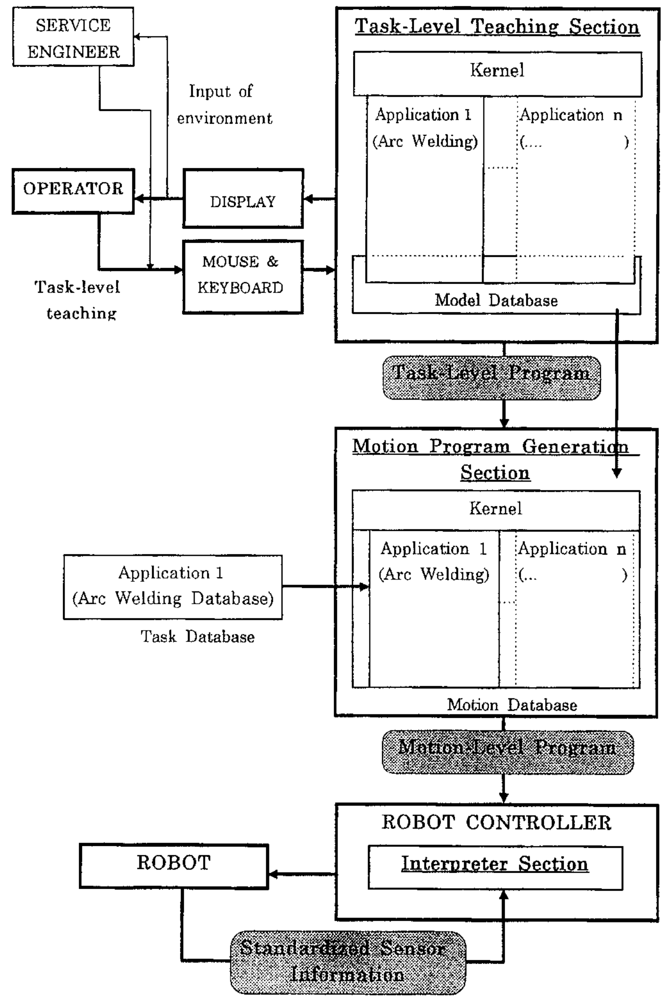

Fig. 1. Schematic diagram of the robot programming simplification system.

on a PC (IBM-compatible). The basic specifications of the PC are as follows:

- CPU (Pentium 133 MHz)
- Memories (64 MB)
- Color display (17″)
- Others: Hard disk, CD-ROM, mouse, keyboard

The system is connected to a 3D-CAD computer or a robot simulator by LAN to create a work model or to set up a motion environment. By providing a serial connection to a robot controller using RS232C, it allows us to transfer the created motion-level programs.

b. **Software configuration.** The software configuration is shown in Figure 3. This system uses Microsoft Windows NT3.51 as the OS and Microsoft Visual C++ as a development language.

The software is composed of the following five modules:

- System kernel: communication and file access management
- 3D-display & part selecting: work model display & welding parts selection
- Graphical sequence editor: task-level program editing
- Welding condition editor: welding condition setting
- Main tool: display menu and select functions

Each module is an independent process and performs data communication by means of socket function. Communication is controlled by the system kernel. The system kernel also controls access to the files including work models and all robot programs.

The remaining four modules control the display windows for the operator to operate on the screen.

3.3 Task-level teaching method

a. **Teaching flow.** The overall teaching flow in the case of task-level teaching is shown in Figure 4. A work model and environment setting data which specify the robot and work positions will be created and input will be by a general-purpose 3D modeler on a separate device from this system.

Once all the data are loaded on this system, a welding

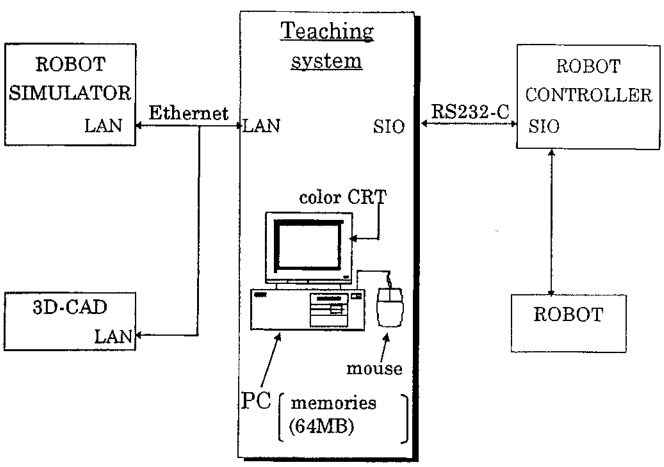

Fig. 2. Hardware configuration.

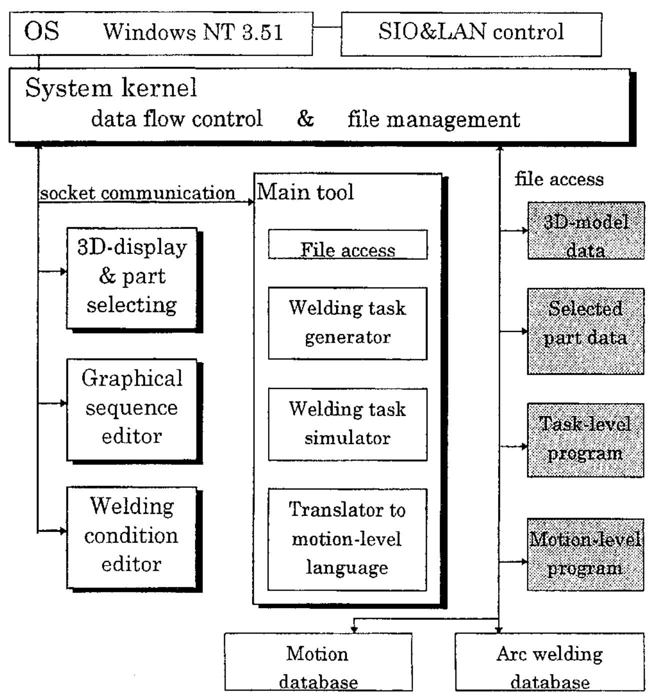

Fig. 3. Software configuration.

https://doi.org/10.1017/S0263574797000118 Published online by Cambridge University Press

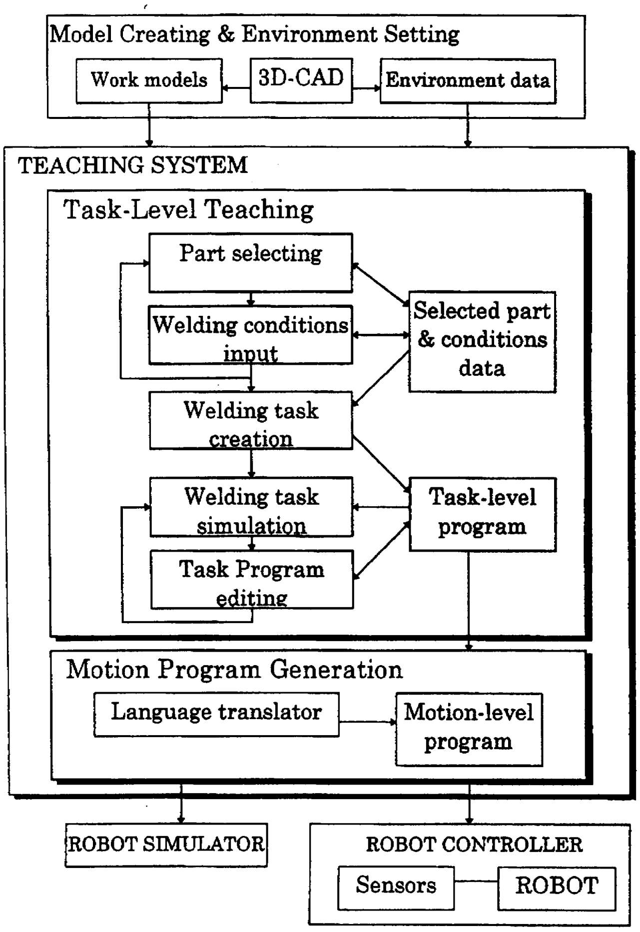

Fig. 4. Task-level instruction flow.

part of the work model displayed on the screen is taught. Then welding conditions of the selected welding part are set. This procedure is repeated for every and all welding parts.

From the specified welding parts and condition data, a task-level welding program is automatically created. The task here includes necessary robot operations, such as detection of work deviation, torch approach at a welding start point, and torch removal at a welding end-point, in additon to welding operations.

Then the created task-level program is checked for its procedure with a task simulator on this system. If a part of the procedure needs to be corrected, the program is corrected accordingly. The simulation and correction of the program are repeated until an appropriate procedure is created.

The completed task-level program is then transferred to the motion program generation section to be converted into a motion-level program. The motion-level program is also generated automatically with the environment data set. The motion-level program is then transferred to a robot simulator or a robot itself to check the actual motion.

b. **Windows structure.** This section describes the structure and operation of display windows for individual tasks.

All four displays used for this system are shown in Figure 5. These four windows are displayed on the same screen at the start-up. Figure 6 shows the structure of the windows shown in Figure 5.

- *Main tool window:* consists of a title part and a menu bar.
- *3D-display window:* displays a 3D picture.
- *3D control panel:* is a child window of the 3D-display window to choose the 3D operation.
- *Graphical sequence window:* displays a task program.
- Welding condition editor: displays conditions.

The functions of individual display windows are as follows:

(i) *Main tool window:* The main tool window displays menus to select functions, such as motion-level program generation, file open/close, and display change.

(ii) *3D-display window:* The 3D-display window displays work models. The operator teaches welding paths by selecting the start and end-points of the weld line segment with the mouse. Line data, such as linear or circular, and path data, such as path completion and deletion, are controlled by the control panel shown in Figure 5. The color of the weld portion is changed once teaching has been completed, so that the selected line is distinguished from the other parts of the model.

(iii) *Graphical sequence window*: The graphical sequence window is used to carry out graphical editing of a task-level program. A program is described with blocks (leaf), which indicate individual tasks, horizontal lines (branch), which indicate paths, and vertical line (trunk), which indicates the flow of welding operations. Thus, the operator can easily understand the construction of the program.

Each block indicates a welding line segment; therefore, a welding path with multi-line segment data can be described as a horizontal line with multiple blocks.

To change the program sequence, the position of the intended item is changed on the screen with the mouse. The selection of items on the 3D-display window and the specifications on the sequence window complement each other, i.e. an item selected on the 3D-display window will be specified on the sequence window.

(iv) *Welding conditions window:* The welding conditions window requests the operator to set conditions to carry our welding operations if necessary. It also displays and sets conditions pertaining to each task-level command, i.e. if the command is "welding", then welding conditons such as leg length, thickness of the work, and direction are displayed. If the command is "detection", then the detection pattern and position are displayed.

c. **Task simulation.** This function is provided to check the sequence of developed task-level program on the screen with the windows mentioned above. Once the simulation is executed, the welding part on the

https://doi.org/10.1017/S0263574797000118 Published online by Cambridge University Press

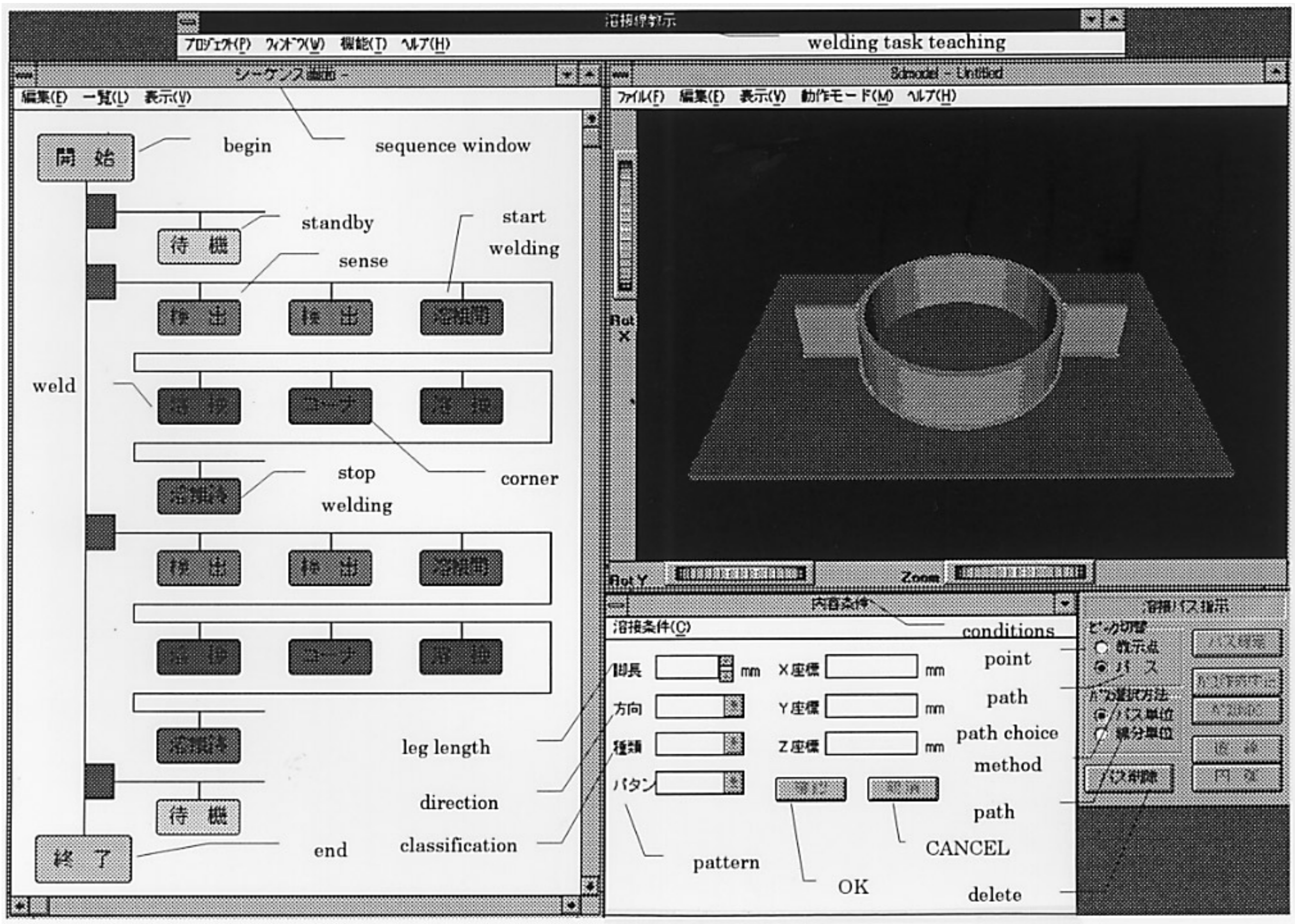

Fig. 5. An example of displayed graphical interface. (Captions are given in Japanese **and** English).

3D-display window and the command block on the sequence window are highlighted in the sequence. The set conditons are shown on the welding conditions window.

4. EVALUATION AND DISCUSSION

JARA carried out evaluation tests for the prototype system developed to verify the usefulness of the task teaching concept by using a robot simulator in February 1996. Many persons who were not familiar with robot programming tried to construct a robot program for arc welding tasks for the rather simple work shown in Figures 5 and 7, after a brief instruction period.

The usefulness of the proposed task-level teaching has been confirmed. In the case of corrugation welding for the work shown in Figure 7, the results obtained showed that the time for teaching by the task-level teaching system could be reduced to about one-sixth by using a teaching aid. Also, various subjects to be improved have been pointed out, such as enhancement of the expression capability of the task-level language, expression of routine tasks, conditional expressions, and so on.

Through evaluation tests, the validity of the concept proposed and the fact that the task level teaching system developed is easy to use for beginners and is a user-friendly, for easy-to-learn robot programming system have been clarified.

The authors think that the reasons are as follows:

- The interface is familiar to a user who has experience in operating a personal computer.
- A graphical display makes it easy for the user to understand intuitively the instructing task.
- Only a minimum number of instructions guided by the system are required to produce the robot-language program.

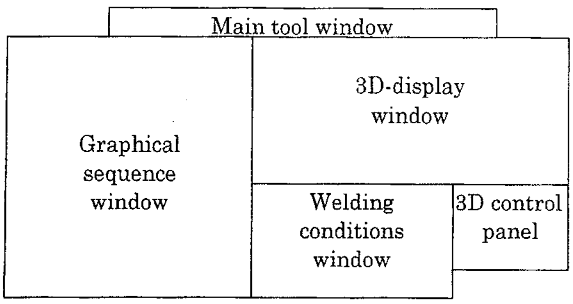

Fig. 6. Windows structure.

5. CONCLUSION

A prototype teaching system for an arc welding robot, which enables an operator to construct a robot program by teaching task-level instructions through the graphical user interface, has been developed, and the usefulness of the system has been verified. In 1996, according to the results, practical technologies to automatically deploy a task-level program into a motion-level robot program, which needs advanced information processing technol-

https://doi.org/10.1017/S0263574797000118 Published online by Cambridge University Press

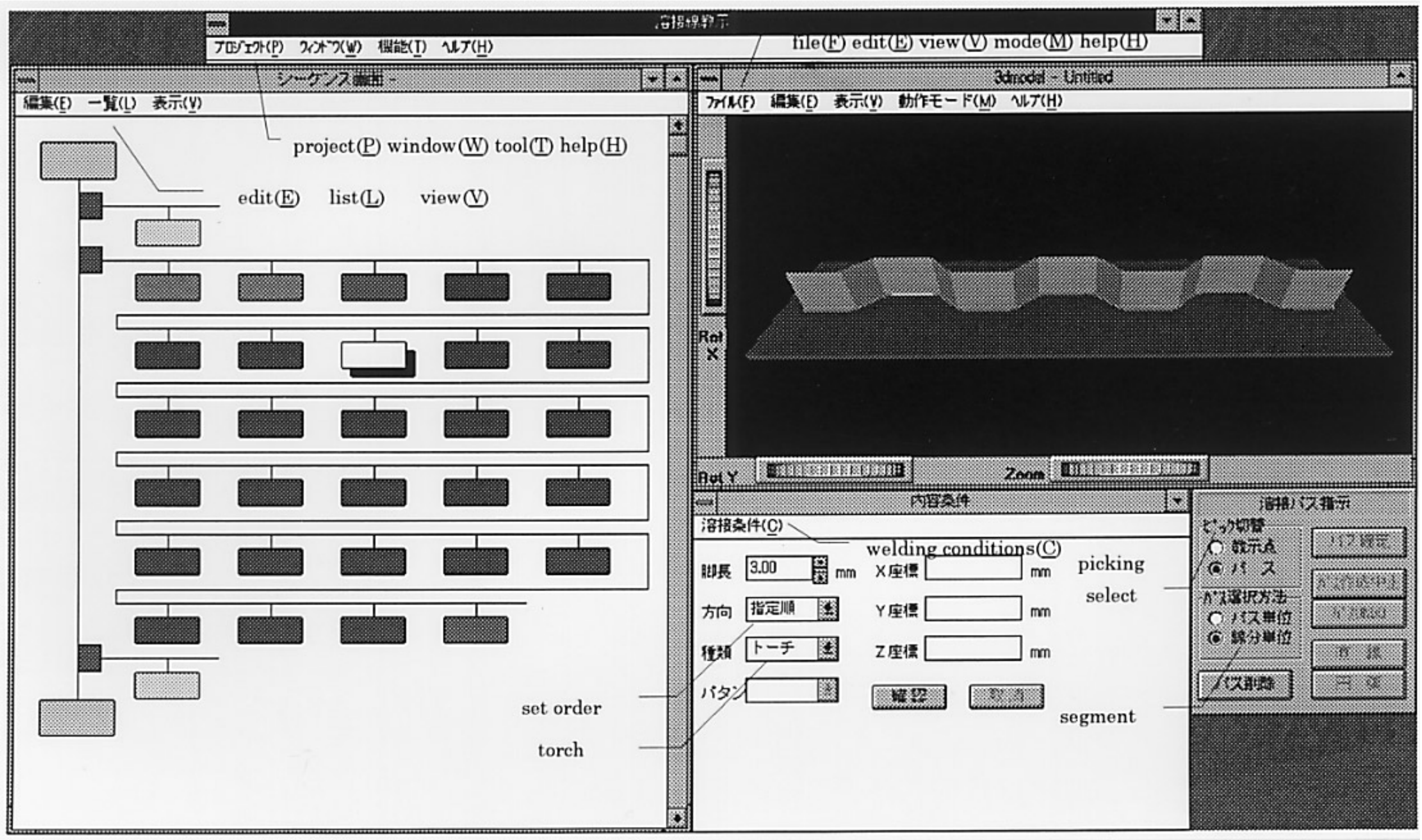

Fig. 7. An example of displayed graphical interface in evaluation tests. (Captions are given in Japanese **and** English).

ogies, is to be developed. Improving the prototype system for arc welding and evaluation tests with an actual industrial robot are to be carried out in 1997.

The authors hope that the robot programming simplification system developed under the project will provide a solution to realize a practical and easy-to-use programming system and consequently expand the robot market, especially in small batch production works.

Acknowledgement

This project has been carried out under contract with the Information Technology Consortium (ITC) as a project of "The Research and Development on Technological Succession and Promotion Systems for Advanced Production Corresponding to Saving Energy" sponsored by the Information-technology Promotion Agency (IPA).

References

1. A. Matsumoto and T. Arai, "The present and Future of Robot Languages" *J. Robotics Society of Japan* **11,** No. 1, 36–39 (1993).
2. T. Ogasawara, "Software Platform for Robotics Research" *J. Robotics Society of Japan* **14,** No. 1, 2–5 (1996).
3. Japan Robot Association *The Report of the Development of Indistrial Robot Programming Technologies in FY1994* (April, 1995).
4. H. Inoue "Research Subjects on Robot Languages" *J. Robotics Society of Japan* **2,** No. 2, 87–90 (1984).
5. Japan Robot Association *The Report of the Development of Industrial Robot Programming Technologies in FY1995* (April, 1996).
6. M. Weck and R. Dammertz, "OPERA-A New Approach to Robot Programming" *Annals of the CIRP* **44,** 389–392 (Jan., 1995).

https://doi.org/10.1017/S0263574797000118 Published online by Cambridge University Press