ISSN 0016-2523

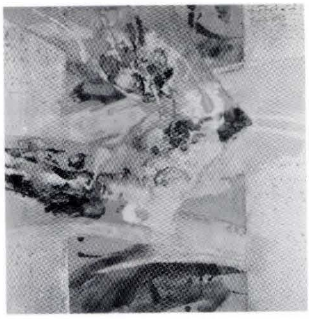
FUJITSU

SCIENTIFIC & TECHNICAL JOURNAL

Autumn 1990 VOL.26, NO.3

Special Issue on Hypermedia Technology

FUJITSU Sci. Tech. J., **26**, 3, pp. 165-257, Kawasaki, October, 1990



This Issue's Cover:
Light by Yukinobu Kudoh

Light is the source of life. All things in nature grow by the grace of light; artificial light, created by mankind, is fully utilized for the cultural and economic prosperity of all. Today, through revolutionary telecommunications by light, we can transmit bulk information in an infinitely easier manner.

The broad-band ISDN will enable 1 000 times the information of today to make the rounds of the world through optical fibers. To realize this, Fujitsu is conducting R&D in new optical telecommunications technologies for superhighspeed optical telecommunications and optical switching systems. To the glory of all.

FUJITSU Scientific & Technical Journal is published quarterly by FUJITSU LIMITED of Japan to report the results of research conducted by FUJITSU LIMITED, FUJITSU LABORATORIES LTD., and their associated companies in communications, electronics, and related fields. It is the publisher's intent that FSTJ will promote the international exchange of such information, and we encourage the distribution of FSTJ on an exchange basis. All correspondence concerning the exchange of periodicals should be addressed to the editor.

FSTJ can be purchased through KINOKUNIYA COMPANY LTD., 3-17-7 Shinjuku, Shinjuku-ku, Tokyo 160-91, (Telex No. 2424344, answerback KINOKUNI J).

The price is US $7.00 per copy, excluding postage.

FUJITSU LIMITED 1015 Kamikodanaka, Nakahara-ku,
Kawasaki 211, Japan
Cable Address: FUJITSULIMITED KAWASAKI
Telephone: +81-44-777-1111
Facsimile: +81-44-754-3562

Printed by MIZUNO PRITECH Co., Ltd. in Japan

FUJITSU

SCIENTIFIC & TECHNICAL JOURNAL

Autumn 1990 VOL.26, NO.3

Special Issue on Hypermedia Technology

CONTENTS

Featuring Papers

165 Preface ● Yasuchika Koike

167 Hardware of Hypermedia Personal Computer: FM TOWNS
● Masaharu Tamai ● Makoto Nakamoto

175 Hypermedia Operating System for FM TOWNS
● Hitoshi Nishiuchi ● Sumio Kimura ● Masahiro Kataoka

187 Hypertext System: TownsGEAR
● Sunao Ueda

197 Hypermedia Personal Computer Communication System: Fujitsu Habitat
● Kazutomo Fukuda ● Tadayuki Tahara ● Toru Miyoshi

207 Musical Instrument Digital Interface Sequencer Software: EUPHONY
● Joe Mizuno ● Satoru Ono

214 CD-ROM Premastering System Using CD-Write Once
● Yasushi Ashikaga ● Kazuhito Fukuoka ● Masafumi Naitoh

224 F6365 Japanese Document Reader
● Jun Sato ● Yoshinori Kuwamura ● Chieko Ohno

234 Fujitsu Advanced Drawing Capture System: FADCS
● Yukikazu Kabuyama ● Teruo Ootake ● Jun-ichi Koizumi

Invited Paper

245 Toward Development of a Civic Infrastructure for the Information Society
● Hajime Enomoto

SYNOPSES

UDC 681.3.02:681.327.68

FUJITSU Sci. Tech. J., **26**, 3, pp. 167-174(1990)

Hardware of Hypermedia Personal Computer: FM TOWNS

● Masaharu Tamai ● Makoto Nakamoto

The FM TOWNS was introduced to the market in February 1989. It is a 32-bit multimedia personal computer featuring a built-in compact disc drive. This computer is Fujitsu's entry model for the multimedia market, a market that is expected to expand rapidly in the future. This paper focuses on multimedia processing, system requirements, and the FM TOWNS design concept.

UDC 681.3.066

FUJITSU Sci. Tech. J., **26**, 3, pp. 175-186(1990)

Hypermedia Operating System for FM TOWNS

● Hitoshi Nishiuchi ● Sumio Kimura ● Masahiro Kataoka

Fujitsu has developed a new personal computer, the FM TOWNS, which can handle both text data and audio/video data. Fujitsu has also developed a hypermedia operating system that enables easy production of programs for audio/video applications. This paper discusses the requirements for a hypermedia operating system and the features of the system used in the FM TOWNS.

UDC 001.81:681.3.02

FUJITSU Sci. Tech. J., **26**, 3, pp. 187-196(1990)

Hypertext System: TownsGEAR

● Sunao Ueda

Personal computers are proliferating all over the world, and many programming languages have been developed to improve the efficiency of application development. Unitil several years ago, programming languages were developed with professional programmers in mind. However, new programming languages and environments are entirely different from conventional ones in many respects. TownsGEAR, one such language, operates on the multimedia personal computer, FM TOWNS.

UDC 681.3.064:79.077

FUJITSU Sci. Tech. J., **26**, 3, pp. 197-206(1990)

Hypermedia Personal Computer Communication System: Fujitsu Habitat

● Kazutomo Fukuda ● Tadayuki Tahara ● Toru Miyoshi

This paper describes the technological development of Fujitsu Habitat, and outlines its structure and features. Fujitsu Habitat is a new type of personal computer communication system that supports multimedia processing. It synchronously generates pictures, sounds, and characters — a feat which cannot be performed by conventional personal computer communication systems.

This system is operated by linking Fujitsu's FM TOWNS hypermedia personal computer to a UNIX bost machine.

Communication between the host and terminals is done using a unique communication protocol that ensures high-efficiency and high-reliability data transmission.

UDC 681.3.064:7.091

FUJITSU Sci. Tech. J., **26**, 3, pp. 207-213(1990)

Musical Instrument Digital Interface Sequencer Software: EUPHONY

● Joe Mizuno ● Satoru Ono

The EUPHONY sequencer software is a music composition/arrangement application which supports the Musical Instrument Digital Interface (MIDI) standard, EUPHONY enables music played on a keyboard to be recorded in real time, edited, and played back. The user can compose and perform orchestral music without a real orchestra. EUPHONY also has graphic editing functions which display performance information, e.g. pitch, volume, and duration, thus increasing editing efficiency. EUPHONY is easier to use than conventional sequencer software. This paper describes EUPHONY and the reasons for its development.

UDC 681.327.68

FUJITSU Sci. Tech. J., **26**, 3, pp. 214-223(1990)

CD-ROM Premastering System Using CD-Write Once

● Yasushi Ashikaga ● Kazuhito Fukuoka ● Masafumi Naitoh

The CD-ROM has become an important computer storage medium. However, recording data onto a CD-ROM requires special equipment, thus making it difficult to develop and deliver applications on CD. The Fujitsu Premastering System has been developed to solve this problem. This paper describes the configuration and functions of the Premastering System.

It also describes the CD-ROM standards, the procedure used to generate a CD-ROM application, and the CD-Write Once recorder developed for the Premastering System.

UDC 003.324.2:681.3.02

FUJITSU Sci. Tech. J., **26**, 3, pp. 224-233(1990)

F6365 Japanese Document Reader

● Jun Sato ● Yoshinori Kuwamura ● Chieko Ohno

Progress in multimedia document processing has increased the need for better automatic document input. Fujitsu's Japanese-language document reader automatically reads printed documents and converts the words into computer character code. Techniques such as automatic block extraction, character segmentation, the extraction of ruled lines, character recognition, and lexical post-processing have been built into this reader. The document reader ensures 99 percent error-free reading at a rate of 20 Japanese characters per second.

UDC 681.327.2(084.11)

FUJITSU Sci. Tech. J., **26**, 3, pp. 234-244(1990)

Fujitsu Advanced Drawing Capture System: FADCS

● Yukikazu Kabuyama ● Teruo Ootake ● Jun-ichi Koizumi

This paper describes FADCS, which enables high-speed input of sketches drawn by hand into personal computers or workstations. FADCS was developed to process general-purpose functions, independent of drawing types. The major functions are image to vector conversion and extraction and recognition of characters. The first function converts image data into vector data. This function can recognize graphic primitives (circles, etc.) to facilitate drawing-specific processing such as actual drawing input. This technique is intended for basic image-to-vector conversion on personal computers. Subsequently, the functions will be improved and the degree of freedom for media operation with multimedia personal computers will be increased.

UDC 008.2:681.3.02
FUJITSU Sci. Tech. J., **26**, 3, pp. 245-257 (1990)

Toward Development of a Civic Infrastructure for the Information Society

● Hajime Enomoto

This paper describes a new civic infrastructure for the information society. It considers several features of culture and civilization, for example, fire as a "magic tool" in ancient times. It also discusses the effect of information networks on communities, and the basics of the acquisition and representation of information. This paper presents the observation that the recursive nature of the use of information in ancient times has parallels in the modern age. For a computer network to have an effect on an information structure, sensory information must be changed to logic information by humans. This means that concepts of confrontation and integration become useful tools for the analysis and synthesis of information, and total cooperation mechanisms will be enhanced by using essential concepts and functions available in a proposed civic infrastructure.

Preface

Special Issue on Hypermedia Technology

●Yasuchika Koike
Director
General Manager of Office & Personal Systems Group

The 1990's will be the decade of the multimedia. Multimedia communication between humans and computers enables natural interfaces which simulate everyday life. Ultimately, this environment will evolve into an intellectual and creative environment with multimedia computers playing an essential role.

We named this interactive multimedia the "hypermedia". Interactive multimedia is the combination of multimedia (audio, images, graphics, text, and numeric information) and the interactive capability of multimedia computers. We believe that hypermedia technology will greatly improve the human computer interface, and that it is the key to user friendly and creative computing. Interaction with computers by audio and video means also stimulates creativity and artistic sensitivity.

Because of recent advances in semiconductors, optical storage, and digital audio/video information processing, hypermedia processing on relatively inexpensive personal computers is now a close reality.

As a first step toward the personal hypermedia processing era, in February 1989, we released the FM TOWNS hypermedia personal computer. The FM TOWNS comes with a CD-ROM and has enhanced audio and video information processing capabilities.

Hypermedia involves the processing of information that is several orders of magnitude larger than information associated with traditional character processing. This necessitates faster processing speeds, a huge storage area, and digital signal processing of audio-video information.

Hypermedia software is very different from conventional character-oriented software. Because it involves sounds and pictures, the development of hypermedia software is more like the production of a movie, and conventional programming is only a small part of the total development effort. Hypermedia software will be developed by a team consisting of a director, artists, musicians, and programmers. Also, the effort required to develop hypermedia software is much greater than for conventional software.

It is, therefore, essential to provide a good hypermedia software development environment for those specialists who are unfamiliar with computers. A good environment significantly improves the productivity of hypermedia software development.

To build a broad hypermedia software base, basic standards must be established. The FM TOWNS is based on an industry standard MS-DOS^{TM} and Fujitsu's hypermedia extension. Fujitsu is now helping in the development of multimedia extension standard for Microsoft® -Windows^{TM} 3.0.

The FM TOWNS uses the ISO9660 standard as its CD-ROM format. We have played a key role in the development of the CD-ROM/XA standard for CD-ROM hypermedia software.

This special issue describes Fujitsu's hypermedia activities, which are centered around the FM TOWNS. The first paper *Hardware of Hypermedia Personal Computer: FM TOWNS* describes the FM TOWNS hardware. It focuses on the functions required to support hypermedia, for example, audio and visual functions. The second paper *Hypermedia Operating System for FM TOWNS* describes the functions of the TOWNS OS and BIOS (BIOS controls the hypermedia I/O devices). This paper also describes a high-efficiency CD-ROM access method. The third paper *Hypertext System: TownsGEAR* describes the TownsGEAR hypermedia software development tool. This tool handles text, images, sounds and voices in an integrated manner.

The fourth paper *Hypermedia Personal Computer Communication System: Fujitsu Habitat* gives an example of a hypermedia system that improves the human computer interface. Habitat is a network based multi-user chatting and roleplaying system that uses pictures and simple animation. It is supported by NIFTY-Serve, a commercial personal computer network.

The fifth paper *Musical Instrument Digital Interface Sequencer Software: EUPHONY* describes a tool for visually composing and editing a music score on a personal computer screen. EUPHONY also controls MIDI musical instruments to perform composed music.

The sixth paper *CD-ROM Premastering System Using CD-Write Once* describes a system which assembles software onto CD-ROM and makes a write-once CD-ROM for verification. This system reduces the time required to make a CD-ROM master for stamping by reducing the iteration time.

The seventh paper *F6365 Japanese Document Reader* describes a system which recognizes Japanese characters on printed documents such as newspapers and magazines. The eighth paper *Fujitsu Advanced Drawing Capture System: FADCS* describes a system which scans and vectorizes lines and recognizes symbols on a drawing. These technologies are important tools in the reuse of previously composed and printed information. This type of information is still a major source of hypermedia information.

The FM TOWNS marks the beginning of the hypermedia era and demonstrates Fujitsu's commitment to making computers more useful and easier to use.

UDC 681.3.02:681.327.68

Hardware of Hypermedia Personal Computer: FM TOWNS

● Masaharu Tamai ● Makoto Nakamoto *(Manuscript received June 8, 1990)*

The FM TOWNS was introduced to the market in February 1989. It is a 32-bit multimedia personal computer featuring a built-in compact disc drive. This computer is Fujitsu's entry model for the multimedia market, a market that is expected to expand rapidly in the future. This paper focuses on multimedia processing, system requirements, and the FM TOWNS design concept.

1. Introduction

The computer was first put to practical use about 30 years ago, and has since been used in many applications. There have been three major development periods in the history of the computer.

The first period, the 1960's, was a period of research and development. The computer was introduced to Japan during this decade and was used mainly in research institutes and university laboratories.

The next period, the 1970's, the computer was used almost exclusively for numeric computation, and its abilities in this area were extensively developed.

The most widely used type of computer was the mainframe. A typical application was as a large system in a bank. The range of computer applications rapidly increased after the appearance of the microcomputer in the mid-1970's.

However, the early word processors and personal computers supported 8-bit processing only, and had little memory or file support. This restriction significantly limited their range of application.

The most important computing development of the third period, the 1980's, was office automation. Computers were now being used for both numeric computation and Japanese-language processing. Personal computers supported a 16-bit CPU, at least 256-Kbyte of memory, and floppy disks having at least 320-Kbyte of space. Both Japanese-language processing and business processing then became possible.

Also, the application of word processors and personal computers to office automation became common.

In the 1990's, the age of experession, multimedia processing is expected to became very important.

During its history, the computer first performed numeric computation, then Japanese-language processing, and has now expanded out of the computer room into personal use. For the first 30 years, multimedia processing of information that included pictures and sound has been weak. However, this situation is expected to significantly improve and become the main focus of the 1990's.

If multimedia can be handled with ease, it will play an important role in the next development period, and the computer will become a powerful tool of human expression.

This paper discusses the FM TOWNS architecture. This architecture meets the system requirements of multimedia processing and yet is inexpensive enough to allow the FM TOWNS be sold as a consumer product.

2. The personal computer as a representation medium

The personal computer serves as a special communication medium between humans. The computer is often referred to as a second langugage and as a means of universal representation. As computer multimedia processing progresses, we may see a time when the personal computer will enable non-linguistic expression.

Currently, information is mostly distributed using paper media: (e.g. books, magazines, and newspapers), and audio/visual media (e.g. CDs, records, television, and movies).

In the future, more and more of this information will be distributed using distributed networks or personal computers.

Characters, photographs, sound (music and voice), and animation will be processed as digitized information in personal computers.

The major differences between multimedia and conventional media are the digitization and interactive processing performed in multimedia operations.

1) Digitization puts data in representation formats (characters, pictures, and sound) for centralized management. These data items can be processed and stored much more freely than conventional analog media.
2) Interactive processing provides an environment in which the user can use information interactively instead of simply obtaining the information.

3. Features of multimedia processing

The basic function of a personal computer is shown in Fig. 1.

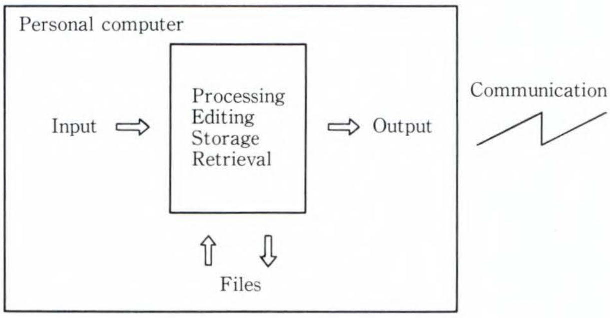


Fig. 1—Basic function of a personal computer.

3.1 Amount of data in multimedia processing is 10 to 1 000 times larger

Multimedia operations process characters, numbers, animation, and natural pictures and sound. Unlike information in the form of numbers or coded characters, picture and sound data is not coded. Therefore, the amount of data required for pictures and sound can be 10 to 1 000 times larger than that required for coded data.

For example, one second of an average person's spoken language can be represented using about ten characters. Since each character requires 16 bits, written representation of one second of the spoken voice will require 16 x 10 = 160 bits.

However, to produce a high-fidelity digitization of the actual sound of the spoken voice requires a sample rate of 20 kHz and an 8-bit quantization level. Therefore, to digitize one second of the spoken voice requires 8 x 10 Kbits = 160 Kbits.

The amount of data required for digitization of the spoken voice is 1 000 times that required to represent it using written characters.

The amount of still and animation data has increased by a factor of ten due to an increase in the number of usable colors. The amount of data has also increased by a factor of ten or higher for animations. Generally, the amount of data required for future animations can be expected to increase to between 100 and 1 000 times the amount required for monochrome animation.

3.2 Realtime processing is required

In multimedia processing, it will become necessary to process extremely large amounts of data and realtime changes in voice and animation data. Future processing systems must meet these requirements.

Desktop publishing (DTP), which was developed in the mid-1980's, requires a huge amount of data for bit-map processing. It can perform three-dimensional processing (if color is handled as a separate dimension), but mainly performs two-dimensional processing. Multimedia processing also requires a realtime

Table 1. Picture and sound sources and associated input devices

	Source	Feature	Means of input
Pictures Natural still pictures Natural animation	Photographs	High resolution Multicolor (24 bits) Plane	Image scanner
	Natural landscapes Natural moving pictures	Multicolor (15 or more bits) Cut out of a cube	Video camera Conversion from NTSC to RGB
	Professionally produced products Television Video software (VTR/LD)	Already processed electronically (NTSC video)	Conversion from NTSC to RGB (a tuner is required for television air checks.)
Music Sound effects Voice	Sounds		Microphone A/D converter
	Professionally produced products Television and radio Records CDs and LDs	Already processed electronically	A/D converter (a tuner is required for broadcast air checks.)

capability, and can therefore be considered as a four-dimensional process.

3.3 Standardization of data is important

The creation of picture or sound data is not easy. Standardization of the data format is important for easy distribution of data and tools. Not only the file format but also the display and sound output formats and the data compression methods must be standardized.

3.4 Requirements of system components

The above discussion indicates that a large amount of data must be processed in realtime during multimedia processing. This section discusses the specific requirements of the system components for realtime processing.

3.4.1 Input

The keyboard is used to input numeric and character data. Specialized devices are necessary to input non-coded information such as visual and sound data. Table 1 lists these other types of data and the devices used to input them.

3.4.2 Processing, editing, storage, and retrieval

These operations are all performed by the personal computer. Hardware shares the total required throughput with software, and how to optimize the allocation of throughput can become a problem. Multimedia processing has only a short history, and research on this problem is just beginning.

Standardization of the data format and processing method for still pictures, sound, and animation data was proposed five years ago. System requirements will depend on these standardizations.

Some of the areas to be standardized are:

1) The storage format and representation format for CD-ROM, CD-I, and CD-ROM XA.
2) The representation format including animation processing.
3) The compression-expansion of still and animated pictures JPEG (still pictures), and MPEG (moving pictures).

3.4.3 Output

Multimedia processing involves the reproduction of natural pictures and sound. Higher quality CRTs, printers, and loud speakers than those found on conventional personal computers will be required.

Datails of the FM TOWNS output are given in sections 4.3 and 4.4.

3.4.4 Files

Multimedia processing requires very large files. Up to now, floppy disks have been the most common way to store files on a removable medium. Optical disks have 10 to 100 times the capacity of floppy disks, and will likely replace the floppy disk in multimedia applications.

The access and read/write speeds of hard discs are sufficient for current realtime processing.

3.4.5 Communication

It will not be easy to increase transmission speeds by the required factor of 1 000. However, it may be possible to increase the transmission speed by a factor of 10 using ISDN, and by a further factor of 10 or even 100 using compression techniques.

4. FM TOWNS hardware

So far, we have discussed the general system requirements for multimedia processing. Next, we will discuss the specifications of the FM TOWNS.

Our objectives were as follows:

1) Give personal computers a new image

The FM TOWNS was designed around three key concepts:

i) Entertainment

FM TOWNS has advanced audio/visual functions that make game playing extremely enjoyable.

ii) Practicality

FM TOWNS has many practical applications.

iii) Pleasant appearance and good ergonomics.

2) Minimize costs so that the FM TOWNS can be sold as a consumer product.

3) Standardize the FM TOWNS graphics and sound.

4.1 CPU controller

Figure 2 shows the FM TOWNS hardware. The section up to the SCSI interface on the left and up to the printer board on the right is called the CPU controller. It controls the basic memory and general input-output and is similar to the controller found in most personal computers. FM TOWNS is compatible with Fujitsu FM R series personal computers for business applications. Unlike the FM R series, the FM TOWNS design was for a low-cost multimedia processing computer.

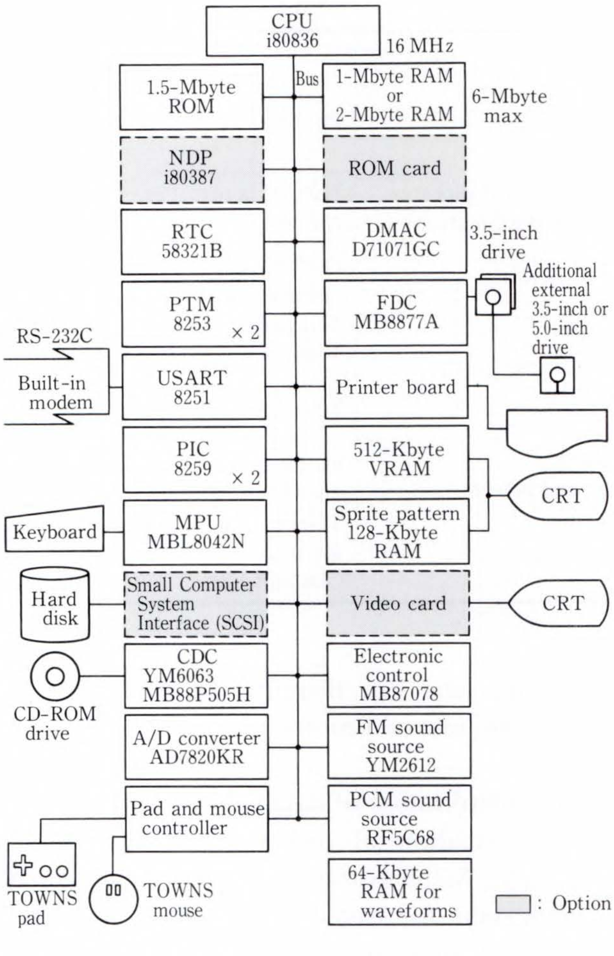


Fig. 2–FM TOWNS block diagram.

A 32-bit CPU is used to handle the large amounts of data involved in multimedia processing, and a common controller IC was used to maintain compatibility. To reduce costs, no high-speed memory access or bus was used.

1) CPU

The FM TOWNS uses Intel's 80386. This device was chosen because of its high cost performance and because it is the world's standard 32-bit CPU.

2) Memory

A 32-bit wide access is used for the main RAM and VRAM. A 16-bit-wide access is used for ROM because ON ROM is not frequently executed.

3) System control and input-output control

System control of the DMAC, timer, interrupt control, and input-output control of the keyboard, printer, floppy disk drive, RS-232C, and SCSI are compatible with the FM R series.

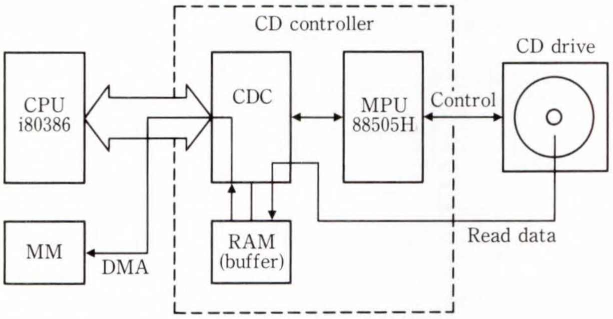


Fig. 3—Block diagram of CD drive.

Compatibility with the FM R series is emphasized because these units are not directly related to the audio-visual or CD-ROM functions of the FM TOWNS. We also took into consideration the need for common software.

4.2 CD-ROM control

Figure 3 shows the CD-ROM controller. CD-ROM drives are extremely versatile storage devices for computer data. However, most drives have an SCSI for high-speed access, and are therefore expensive.

The application objectives of the FM TOWNS CD-ROM differ from the those of conventional CD-ROM drives. The FM TOWNS CD-ROM drive and its interface enable realtime audio-video playback at a low cost.

1) Drive interface

The FM TOWNS CD-ROM drive is connected using a native interface instead of an SCSI. A native interface is used because devices such as CD players, CD graphics devices, CD-ROM XA and the built-in CD-ROM data file are all controlled as an integrated system.

The CDC processes data and the microcomputer controls the drive.

2) Application to CD-ROM XA

A CD-ROM contains data and music in separate files. To play back audio and video data at the same time, music must be palyed back again after it has been copied to the main storage. This complicates control of CD-ROMs.

In 1986, CD-I was proposed as a standard for interleaving, storing, and playing back data and music simultaneously. However, CD-I is a standard for dedicated machines and specifies the CPU and OS. This has limited the application of this standard to personal computers.

The CD-ROM XA standard used for general-purpose personal computers may become a future international standard. In CD-ROM XA, audio and video data can be synchronized, and audio data can be compressed by a factor of between two to 16 using ADPCM. This enables multiple audio data items to be stored in the same file for applications such as multi-language support.

In the FM TOWNS, the controller for CD-ROM XA support has been incorporated in the CD-ROM controller. This was done in anticipation of CD-ROM XA becoming an international standard.

3) Support of CD graphics/karaoke$^{\text{note)}}$

The CD-ROM controller of the FM TOWNS supports CD graphics and karaoke using subdata. Image data is stored as subdata of the music data (CD-DA) frame. The FM TOWNS can read subdata during a musical performance and can display the CD pictures on the screen.

4.3 Screen control section

As mentioned in section 3.1, the screen requirements for multimedia processing depend on the representable objects. The cost of the FM TOWNS display control circuits and the display device (CRT) has been minimized to put the series within the reach of individual users.

The costs were minimized as follows:

1) The video RAM capacity and scan frequency were roughly established.
2) The required resolution was then determined.
3) The minimum circuitry required to achieve this resolution was designed.

4.3.1 Targets related to resolution

The screen control required to do the following was established:

1) Display Japanese-language text (16 x 16-dot font)
2) Display pictures using 32 000 colors

Note: This is a Japanese word meaning recorded instrumental music used to accompany live singing.

(R:G:B = 5:5:5).

3) Display a Japanese-language text screen overlaid with a picture image screen.
4) Display of 256 gradations (16 800 000-color palette mainly for software painting).
5) Fetch a still picture from an NTSC video source and display it.
6) Display animations from a program (mainly for games).

4.3.2 Selection of resolution, colors, and horizontal and vertical frequency

The resolution, number of colors, and horizontal and vertical frequencies are the basic elements of a display.

The cost and processing speed of the FM TOWNS allowed the following performance:

1)	Resolution:	640 x 480-dot
2)	Number of colors:	32 000 colors (256 of 16 800 000)
3)	Horizontal frequency:	15 kHz to 31 kHz
4)	Vertical frequency:	60 Hz

The above figures enable good resolution and color representation of natural still and animated scenes. Dedicated control is used.

4.3.3 Display control

Figure 4 shows the screen control of the FM TOWNS. The section enclosed in the dotted line is the video card for the NTSC video interface. The section enclosed in the solid line is the built-in display control (CRT card).

The CRT card uses four specially developed gate-array ICs. The CRT controller is responsible for general timing control. The video RAM (VRAM) access and sprite input-output controllers control the writing of data to VRAM. The video output controller controls the reading of data from VRAM.

1) VRAM configuration

The VRAM uses a 512-Kbyte dual port RAM consisting of two 256-Kbyte blocks (A/B and C/D). One block can represent, 16 colors at 640 x 480 dots, or 32 000 colors at 320 x 240 dots.

These two blocks can be overlaid to 256 colors at 640 x 480 dots. A dual port RAM is used to minimize the competition time between the following: CPU access to VRAM, and access by a bus master other than a CPU at representation, digitization, and sprite operation.

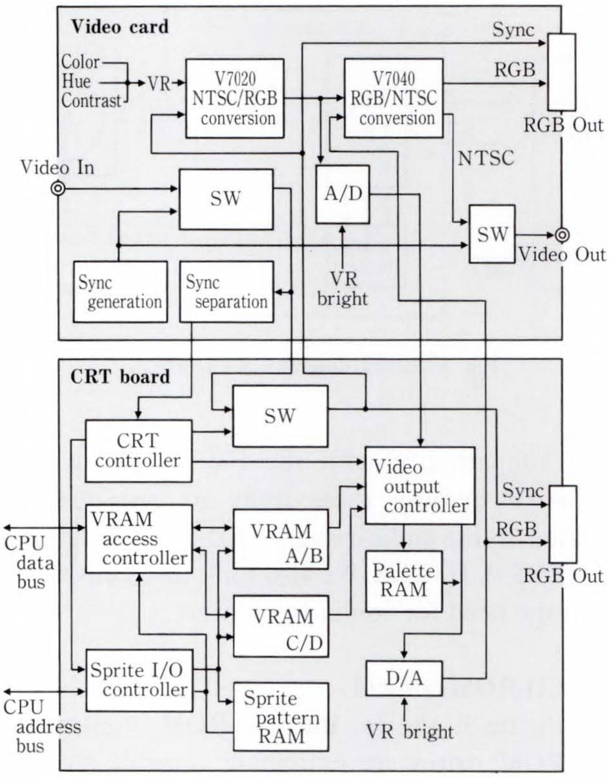


Fig. 4—Block diagram of screen controller.

The two blocks are further divided into sections A, B, C and D. This enables the A side to be displayed while data is being written to the B side and vice versa. The repetition of this operation enables smooth screen animation from programs using sprites and other devices.

2) Sprite input-output controller

Sprite functions are used to rapidly move several patterns on a screen. The FM TOWNS can handle 1 024 sprites of 16 x 16 dots each. Each dot can be one of 32 000 colors.

The sprite function is implemented as follows:

i) The sprite input-output controller alternately writes data to the sprite pattern memory in VRAM C and D. (This is frame operation using VRAM.)

Sprite operation differs from raster operation, and has the following advantages:

a) Each sprite input-output controller uses one IC only.
b) A large number of sprites can be handled because processing in units of frames

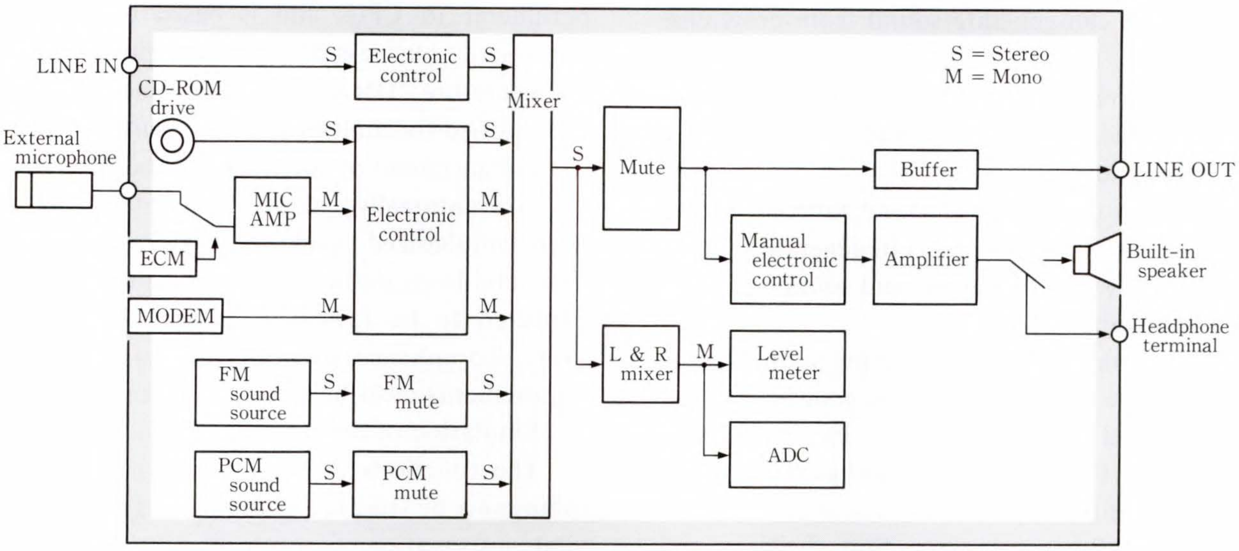


Fig. 5—Block diagram of audio section.

(screens).

3) CRT controller

This controller generates the display timing. When an operation specification is recieved from the program, the CRT controller generates data read timing from VRAM and generates the horizontal/vertical synchronization signal.

4) Video output controller

Data read from VRAM is selectively mixed by the video output controller. This enables two-screen composition by overlapping two video signals from the VRAM block.

5) Video interface

NTSC video is important source of image data. The cost performance of recent VCR, laser disk (LD), and video camera products has greatly improved, and high-quality images can now be processed at low cost.

To take advantage of these developments, the FM TOWNS was designed for easy integration with NTSC video. It can digitize video images and can output FM TOWNS screens in NTSC video format.

An NTSC decoder and A/D converter are provided on a video card to digitize RGB signals (5:5:5) input to the video slot and then output the data to main system VRAM. The timing of this operation must be synchronized to the NTSC video signal. This is done by a synchronized separation circuit on the video card. An image read from VRAM is converted to an analog RGB signal in the main system and is then passed to the video card for encoding into NTSC video. A dot clock adequate for NTSC color is set for the CRT controller encoding to enable high-quality NTSC video encoding.

The FM TOWNS display system can be integrated with NTSC video. NTSC video images can be recorded and output to a video using an inexpensive optional video card.

4.4 Audio section

Figure 5 shows the block diagram of the FM TOWNS audio section. The FM TOWNS audio controller is easy to understand if it is regarded as a mixer.

The audio section accepts sound signals from various sound sources, controls the volume and muting, and mixes and outputs the sound.

4.4.1 Input section

This section describes the following:

1) LINE-IN

Input for external audio equipment.

2) CD-DA

Input for built-in CD drive.

3) Microphone input (built-in and external)

The FM TOWNS has a built-in microphone. An external microphone can also be connected.

4) MODEM

Input for telephone line.

4.4.2 Built-in sound source

The FM TOWNS has two built-in sound

sources and can generate sound from programs. These two sound sources complement each other, and enable the manipulation of sounds such as music, voice, and effects.

1) FM sound source

This is a six-voice FM sound source (stereo).

It uses several data items to generate sounds, and is mainly used for music and sound effects.

2) PCM sound source

This is an eight-voice sampling sound source (stereo). The PCM sound source generates sound from sound data recorded by pulse-code modulation (PCM). The analog-digital converter (ADC) is positioned after the mixers (see Fig. 5). This is an 8-bit converter that digitizes and writes the mixed sound into memroy.

5. Future developments

The FM TOWNS will be developed for the following commercial applications.

1) Professional machine interface for authoring.
2) Personal processor for authorizing and entertainment.
3) Player.

The major technical problems regarding these developments are as follows:

1) Reexamination of CPU peripheral control

The FM TOWNS uses an architecture peripheral to CPUs and is based on Fujitsu's FM R series business personal computers.

Therefore, DMA and timer functions must be modified for multimedia processing.

2) Enhancement of software functions.

Conventionally, hardware development has been emphasized because software platforms have not been defined. Software platforms are expected to be formalized over the next two years. To enhance software functions, software requirements must be given a higher priority.

3) Emulation of humans

The functions of human eyes, ears, and the brain must be studied. These studies will provide vital information for future developments in information processing.

6. Conclusion

This paper presents the development background and the hardware of the FM TOWNS. The FM TOWNS was released only a year ago, and already over two hundred CD-ROM applications have been developed for it.

The FM TOWNS is getting excellent reviews as a multimedia personal computer for games, education, hobbies, and many other applications.

Masaharu Tamai
Engineering Dept.
Personal Systems Div.
FUJITSU LIMITED
Bachelor of Electrical Eng.
Kyoto University 1969
Specializing in Pattern Recognition

Makoto Nakamoto
Engineering Dept.
Personal Systems Div.
FUJITSU LIMITED
Bachelor of Electronic Eng.
Kanagawa University 1974
Specializing in Development of
Operating Systems

UDC 681.3.066

Hypermedia Operating System for FM TOWNS

● Hitoshi Nishiuchi ● Sumio Kimura ● Masahiro Kataoka

(Manuscript received June 7, 1990)

Fujitsu has developed a new personal computer, the FM TOWNS, which can handle both text data and audio/video data. Fujitsu has also developed a hypermedia operating system that enables easy production of programs for audio/video applications. This paper discusses the requirements for a hypermedia operating system and the features of the system used in the FM TOWNS.

1. Introduction

Nowadays, personal computers are being used for entertainment as well as for business. They are being applied in more and more areas, for example, music and animation. In response to this growth, Fujitsu has developed a new personal computer which can simultaneously handle audio data (music and voice) and video (text and image) data.

This personal computer has two main features. The first is the combination of high-resolution multiple-color video functions that display natural images and audio functions that handle sound from CD-DA (compact disk digital audio), FM (frequency modulation), and PCM (pulse coded modulation) sound sources.

The second feature is a built-in CD-ROM drive. This greatly extends the range of applications. The CD-ROM drive can be used as a playback sound source and as multimedia storage device for video data. The CD-ROM drive enables the development, use, and distribution of programs that use large quantities of text, video, and audio data. This environment dispenses with the current need for processing in segments.

A very important part of a personal computer is its operating system, and the one chosen for the FM TOWNS is MS-DOS^{TM}. This system was selected so that the FM TOWNS could run software written for conventional personal computers. Multimedia processing functions have been added as an extension to MD-DOS^{TM}. These include CD-ROM control, multimedia data processing control, and an adapter which can extend MS-DOS^{TM}-specified segmentation from 64-Kbyte units to up to 4 Gbytes units. These extensions are used as the core of the operating system. Other functions to assist operation, for example, application startup and setup have been added to achieve a hypermedia operating system. This paper first explains the requirements for a hypermedia operating system and facilities. It then explains the graphical user interface (GUI), which enables easy use of the operating system functions.

2. Hypermedia operating system

2.1 Requirements for a hypermedia operating system

2.1.1 Data to be handled

Conventional personal computers handle mainly text data, but it will become increasingly important for personal computers to also handle audio and video data. (The image data file format for video data and the sound data file formats for audio data are given in Table 1.)

The FM TOWNS digitizes image and sound

Table 1. File formats of digital image data and digital sound data

a) Example of digital image data
(TIFF: Tagged Image File Format)

TIFF header: II
(Intel byte order)

Tag

Subfile type:	full image
Image width:	640
Image length:	480
Bits per sample:	8
Compress:	1 (no)
Photometric interpretation:	1 (0 for black)
X resolution:	75
Y resolution:	75
Planer configuration:	1 (packed pixel)

Image data
(byte per pixel)

b) Sampling based digital sound data (PCM)

Header

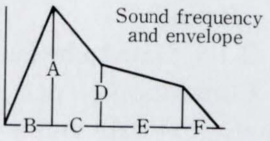
Sound name/ID:	XXXXXXXXYYYY
Data length:	*nn* (byte length)
Loop point:	(loop start offset)
Loop length:	4 096 (byte length)
Sampling rate:	620 (kHz x 62)

Sound data
(PCM sampling data)

c) Tone data file of instrumental music data: PCM
(Eight simultaneous instruments)

Header

Program name:	XXXXXXXXYYYY
Split 1:	Scale for 1st instrum.
⋮	⋮
Split 8:	Scale for 8th instrum.
Sound ID_1:	ID for 1st instrum.
⋮	⋮
Sound ID_8:	ID for 8th instrum.
Envelope 1:	
total level:	A in figure
attack rate:	B in figure
decay rate:	C in figure
sustain level:	D in figure
sustain rate:	E in figure
release rate:	F in figure
Envelope 2:	
⋮	
Envelope 8:	



d) Score data file of c)

Header

Title
File name
......
Creation date
Mute
Port
Channel
Key bias
Transpose
Play filter
Instrument name
......

Score data

pitch, interval*
program change**
pitch bend**
control change**
polyphonic after touch**
channel after touch**
tempo** etc.

*: Repeat
**: Option

data shown in Table 1. One screen of 256-color video requires 300 Kbytes of data. Ten minutes of PCM data (19.3 kHz) requires 1.2 Mbytes.

To handle these large quantities of data on a personal computer, special consideration must be given to filing, audio/video data control, and program processing.

2.1.2 Filing (storage medium)

Storage using the floppy disks and hard disks in general use presents a few problems. Floppy disks have a maximum capacity of only 1 Mbytes. This is insufficient for audio/video data. Also, hard disks are expensive and are not portable. The CD-ROM solves these problems because it is inexpensive, portable, and has a capacity of 540 Mbytes. However, although its data transfer rate is faster than that of floppy disks, it unfortunately takes several hundred milli seconds to seek data on a CD-ROM. Besides deciding how to store data, we had to establish a standard procedure to easily execute programs from CD-ROM. Therefore, it became important to achieve high-speed access and high-speed startup of the operating system.

2.1.3 Controlling multimedia

There are various types of audio/video data. Since video data must be input and output in synchronization with its sound data, a synchronous control method is required. Also, the program interface used with the chosen control method should be standardized.

2.1.4 Programming control

In MS-DOS^{TM}, the program code and data areas of a program are divided into 64-Kbyte segments, and programs must be designed around this segmentation. When variable-length data such as audio/video data is handled, processing for segmentation becomes complicated. To enable users to design their program programs without having to consider segmentation, address space expansion was required as an MS-DOS^{TM} extension.

The functions developed to meet this requirement are described below.

2.2 Functions and structure of hypermedia operating system for FM TOWNS

2.2.1 MS-DOS^{TM} and function extension

MS-DOS^{TM} is a standard operating system for personal computers, and is intended for single-user single-task processing on an i8086 series CPU. MS-DOS^{TM} is made up of a kernel, a shell, and utilities having functions that include task and environment management, memory management, character device input and output, and file and directory management. For the development environment, languages such as BASIC and C-language are provided in addition to the text editor, assembler, linker, and debugger. There are many programs for MS-DOS^{TM} available on the market.

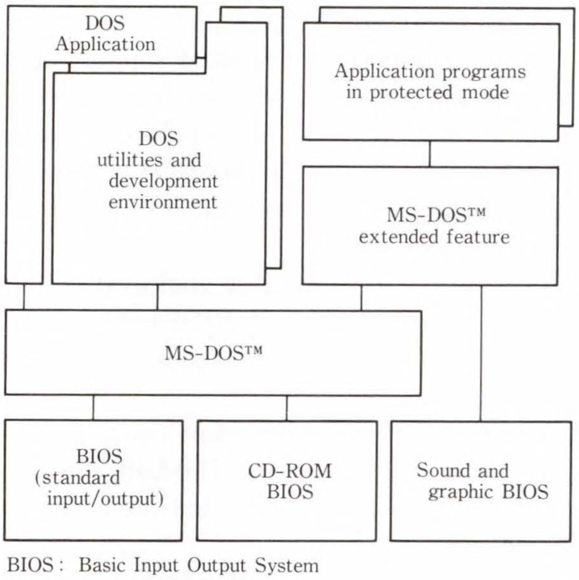

DOS Application
DOS utilities and development environment
Application programs in protected mode
MS-DOS™ extended feature
MS-DOS™
BIOS (standard input/output)
CD-ROM BIOS
Sound and graphic BIOS
BIOS: Basic Input Output System

Fig. 1—Block diagram of hypermedia operating system.

The limitations imposed by 64-Kbyte segments and a 1-Mbyte memory space prevents MS-DOS^{TM} from being used on a hypermedia personal computer equipped with the audio/video functions of a CD-ROM and FM sound source.

We have selected the i80386 as the FM TOWNS CPU, and have adapted MS-DOS^{TM} as its core and added several extensions to achieve a hypermedia operating system (see Fig. 1).

1) MS-DOS^{TM} extensions

An extended feature to support the protected mode of the i80386 was also developed. Using this feature, the space available for application programs has been extended from 1 Mbytes to 4 Gbytes, and the difficulties caused by 64-Kbyte segmentation have been overcome. This extended feature can be used along with standard applications to allow the execution of large application programs.

2) CD-ROM support

We have developed CD-ROM support input functions for program and data files according to the format specified in ISO9660. To make this function available to MS-DOS^{TM} applications, we have extended the BIOS-level functions.

3) Audio/video support

In addition to supporting audio/video functions, we have provided a BIOS-level support for mouse and pad functions, and have improved the C-language libraries.

4) Improvement of development environment

We have developed the F-BASIC386, and have improved the C-compiler, assembler, linker, and debugger.

2.2.2 MS-DOS^{TM} extended feature

This subsection explains the main functions of the MS-DOS^{TM} extended feature described above.

1) Execution of application programs in protected mode

Application programs larger than 1 Mbytes cannot normally run under MS-DOS^{TM}. However, the MS-DOS^{TM} extended feature enables access to an expanded address space of 1 Mbytes or more; therefore, large application programs can be run with an application program argument transfer.

2) Support for MS-DOS^{TM} system calls

When an application program in protected mode requests a system call, the MS-DOS^{TM} extended feature converts the request to the equivalent MS-DOS^{TM} system call. The system reenters protected mode after the completion of system call transfer processing. This transfer processing permits MS-DOS^{TM} system call support.

3) Memory management

Unused MS-DOS^{TM} memory can be controlled to allow application program loading. The remaining memory area can be dynamically acquired by the application program.

4) Interrupt control

An application-specific interrupt handler can be written because the system now supports a function equivalent to the MS-DOS^{TM} BIOS interrupt control function. A hardware interrupt handler can also be written.

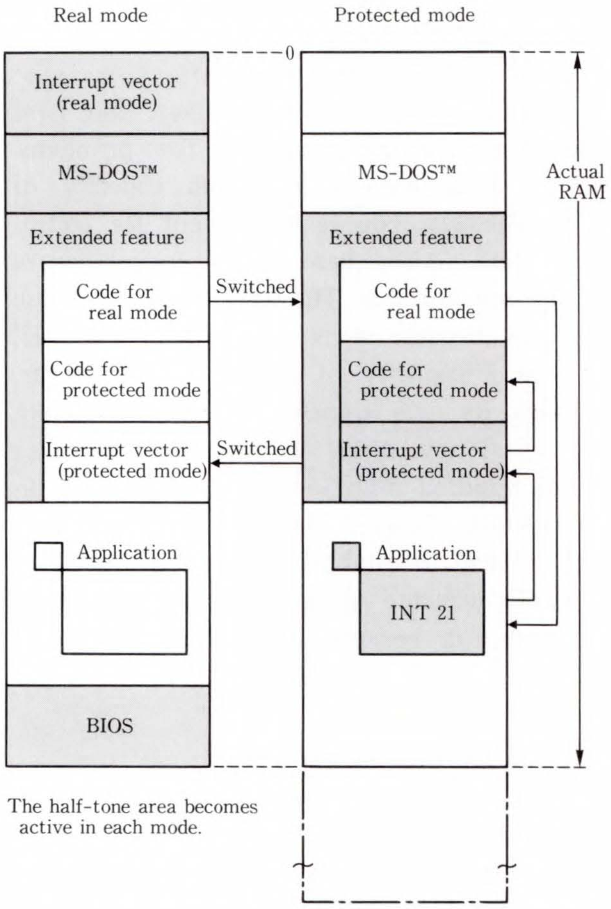


Fig. 2—Operation of hypermedia operating system.

2.2.3 Principle of hypermedia operating system

Using the functions of the MS-DOS^{TM} extended feature, the real and protected modes of the i80386 are dynamically switched to perform the desired operations (see Fig. 2).

2.2.4 Memory management of hypermedia operating system

By supporting segmentation and paging for the i80386, the hypermedia operating system achieves memory management that allows application programs to use a linear space logically exceeding 1 Mbytes. Therefore, large application programs can be run. Each module is designed to accommodate future expansion.

Figure 3 shows the memory allocation of the hypermedia operating system.

2.3 Multimedia support functions

This section describes the multimedia support functions that enable users to write

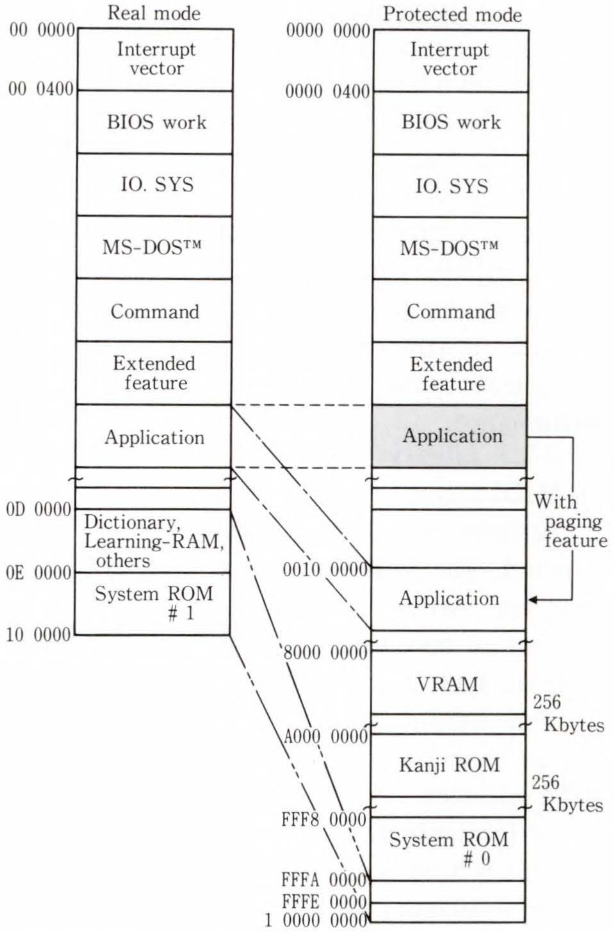


Fig. 3—Memory allocation of hypermedia operating system.

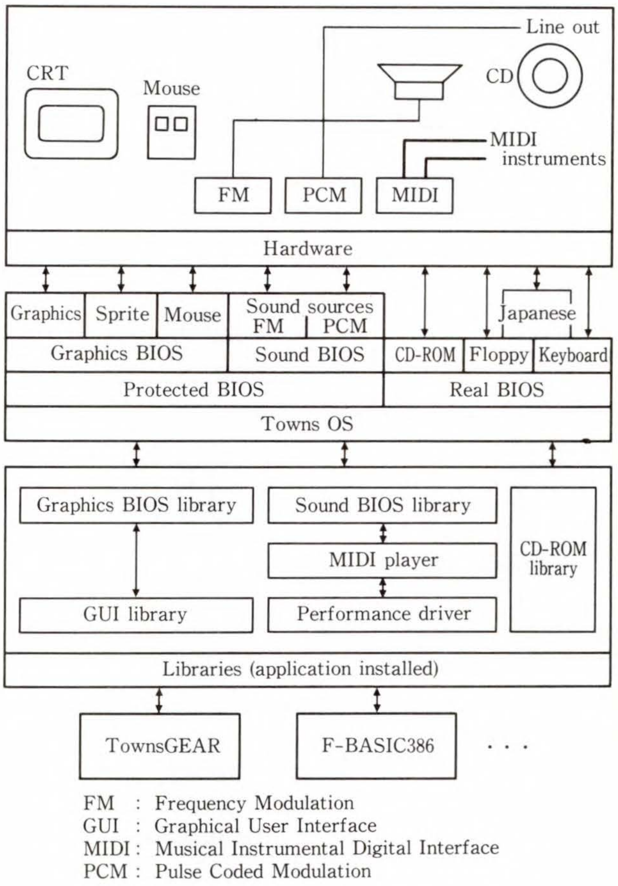


Fig. 4—Software environment.

and use multimedia applications for characters (text), graphics, animations, and music.

2.3.1 Overall configuration

Figure 4 shows the overall configuration of the supported software. The software is divided into two groups: the Basic Input Output System (BIOS) and the libraries incorporated in applications.

There are two types of BIOS: protected and real mode. Protected mode BIOS operates in i80386 protected mode to achieve high-speed processing of graphics and sound data.

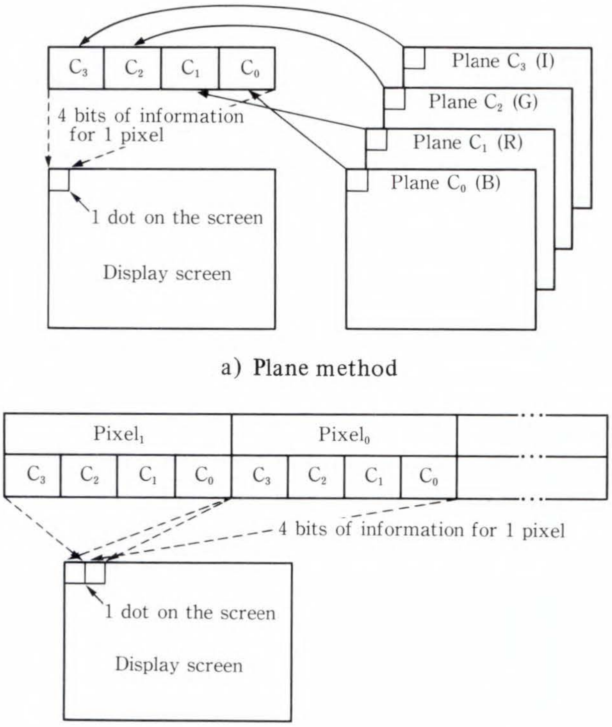
Real mode BIOS consists of floppy disk drivers, keyboard, and Japanese I/O and is shared with MS-DOS™. These functions are implemented as a real mode BIOS to duplicate the conventional MS-DOS™ environment for efficient software development. We will only explain the protected mode BIOS here. The following libraries are provided: a BIOS library that allows BIOS to be called from an application written in C-language, a GUI library that enables easy program development and a common user interface for each application, a performance driver for musical performance, and a Musical Instrumental Digital Interface (MIDI)[1] player.

2.3.2 BIOS function

1) Graphic BIOS

Instead of the conventional plane method in which display control is done for individual colors, the graphics BIOS uses the packed pixel method (see Fig. 5). This method uses video RAM (VRAM) to enable data access in units of one dot. Graphics processing is speeded-up using the i80386's double word accessing feature.

Application programs such as an image editor which edits graphics data can be easily created using this BIOS. Basic operations are



a) Plane method

b) Packet pixel method

Fig. 5–Methods of BIOS function.

supported, such as drawing arcs or rectangles, rotating and moving figures, and reading and writing dot data.

2) Sprite BIOS

BIOS fully supports sprite functions implemented by hardware. It provides functions to register sprite patterns and display grouped patterns. These functions make it easy to create games which require high-speed movement of objects across the screen.

3) Mouse BIOS

This BIOS supports the use of a mouse as an input device. It has functions to read distances of mouse movement, read the status of the right and left buttons, control the mouse cursor shape, and set the speed of mouse pointer movement.

4) Sound BIOS

The sound BIOS has functions to set the following: the performance mode of the FM sound source, audio mode of the PCM sound source, instrument mode of the PCM sound source, PCM recording mode, and input/output mode of the sound source.

In FM performance mode, a six-part performance can be achieved (the voices can be chosen from a selection of 128).

In PCM audio mode, sounds can be played back while data is being transferred from main RAM to waveform RAM in order to obtain the PCM sound source. In PCM instrument mode, BIOS performs envelope processing and sound sampling in the pitch range assigned to each instrument. This produces realistic instrument sounds, and enables an eight-part performance.

2.3.3 Library

1) MIDI player

Performance data is based on the MIDI standard. The MIDI player has a function that emulates MIDI instruments using special customized hardware. MIDI data can be played using the built-in sound sources.

The output sound of the MIDI player can be changed using commands output to the MIDI interface.

2) Performance driver

This is a library for performing music based on performance data from the EUPHONY[1)] driver. (The EUPHONY driver is a tool used to generate MIDI music data.) The performance driver makes it easy to generate and perform expressive music using dedicated software.

3) CD-ROM library

Music at a specified track number can be played back for a digital audio (DA) performance. For greater precision, the hour, minute, and frame (1 frame = 1/75 s) to be played can be specified. Unlike the synthesized sounds produced by an FM or PCM source, this sound source reproduces music produced using nonelectric instruments.

4) GUI library

A GUI library is provided to increase efficiency in the development of applications. Using this function, a desktop menu can be operated at application startup; and menu formats, help windows, and Japanese-language processing can be performed.

2.4 CD-ROM control

This section describes the characteristics of

CD-ROMs used as a storage medium for multimedia data, and describes how the FM TOWNS's CD-ROM is controlled by its hypermedia operating system.

2.4.1 Characteristics of CD-ROMs

1) International standard large-capacity

Low-cost CD-ROM have a capacity of 540 Mbytes, can be used with CD-DA, and are easier and to manufacture than other storage devices.

For data reliability, the CD-ROM control facility uses a mode in which the error correction function is enhanced for CD-DA playback. The CD-ROM is reliable enough for computer data (its uncorrectable read error rate is as low as 10^{-12}).

Standards for CD-ROMs were established in the form of the Yellow Book. The logical format standards are given in the "Logical Format Stipulated at the High Sierra Convention" and "Logical Format Stipulated at the ISO9660 Convention". These standards have established the CD-ROM as an international standard medium. CD-ROM facilities must support this international standard format.

2) Data transfer rate and access time

Unlike hard disk data, CD-ROM data is recorded in a spiral moving from the center to the edge of the disk. Data on CD-ROM is recorded at a fixed linear rate. Thus the speed of rotation must be changed according to the position of the data being read. Each data-block is controlled in units of minutes, seconds, and frames. One frame contains 2 Kbytes, and a data block for one second contains 75 frames. The data transfer rate of CD-ROMs is 150 Kbyte/s for continuous reading. This rate is not very fast, but is sufficient for a still picture image of about 150 Kbytes to 300 Kbytes. However, the seek-time is rather long since the disk rotates slowly and the rotation speed must be changed according to the position of data on the disk. Therefore, access is much slower compared to hard disks.

3) Functional characteristic

There are two types of application processing. One type is random access of data (e.g. dictionary retrieval). The other type is sequential access for reading screen data and PCM audio data.

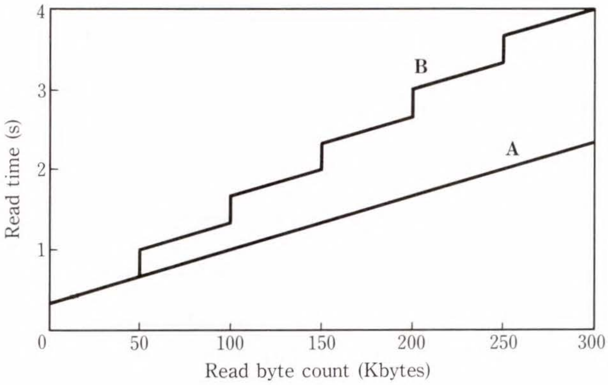


Fig. 6—CD-ROM performance characteristics.

For dictionary-retrieval type software, the seek time mainly depends on the CD-ROM drive hardware; whereas, for sequential-read type software, performance varies widely depending on the access method used by the application software. If the CD-ROM data to be sequentially read is large, a scheme of only one read with no intermediate seek will bring the average data transfer rate closer to the normal data transfer rate. When data is divided for multiple reads, a seek occurs at each interval, which lowers the transfer rate.

Figure 6 shows the processing times needed to read a file once when the files are divided into blocks of 50 Kbytes. Line **A** shows the processing time when reading is not interrupted by seeks, and line **B** (stepped line) shows the time when individual blocks are read.

2.4.2 CD-ROM control configuration

There are three read paths for CD-ROM data access (see Fig. 7). There is also a bootstrap read function to start applications with a single access to CD-ROM (this is not shown in Fig. 7).

1) Basic read path

The basic path used to read data from the application program is the center path in Fig. 7. Since this path uses the extended MS-DOS™ function call to enable file reading using conventional techniques, the application development load is low. When the application program reads a file of more than 64 Kbytes, this DOS extended feature also performs data buffering and supports reading of data amounts above the

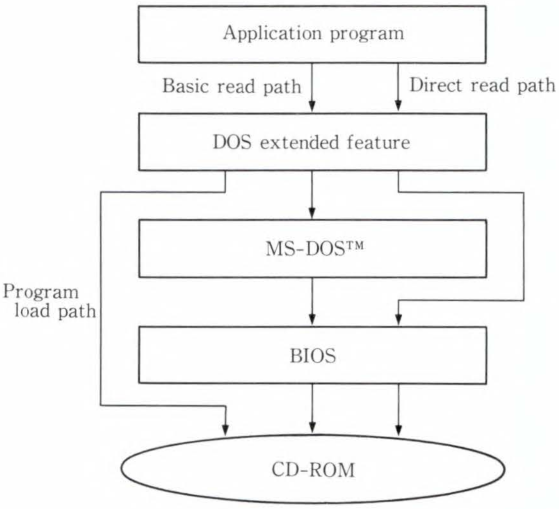


Fig. 7–Control configuration.

segment limit. However, if data exceeding 64 Kbytes is read using this standard path, maximum performance cannot be achieved because of the seeks executed for data buffering (line **B** in Fig. 6).

2) Program load path

The size of an executable file in hypermedia application software tends to be large since it often includes video data. As explained in article 1) above, maximum performance cannot be achieved when loading an executable file using the basic read path. Since an executable file can be read sequentially, the left-hand program load path in Fig. 7 is provided to solve performance problems encountered when reading using the basic read path mentioned above. By using this load path, an executable file of more than 100 Kbytes can be directly read into memory; thus high-speed transfer is achieved without the need to seek. Loading an executable file using this program load path simplifies the development of CD-ROM application software because programmers need not consider the CD-ROM access method.

3) Direct read path

Some application programs (e.g. game software) require high-speed access and are therefore difficult to develop.

The right-hand path shown in Fig. 7 is provided to enable these programs to transfer physical blocks directly from CD-ROM to memory without MS-DOS^{TM} intervention.

4) IPL path

The hypermedia operating system was designed for easy home-use, and its programs must be able to run directly from CD-ROM. The application CD-ROMs contain an IPL (initial program loader) so that the operating system can be changed by simply changing the CD-ROM. The system ROM contains a bootstrap program that reads the IPL program from CD-ROM and then initializes it.

3. System view

To make the FM TOWNS hypermedia personal computer easy to use at home, the GUI uses a graphical display. All that is needed for input is a pointing device such as a mouse. (The operations to start applications and enter data are described below.)

3.1 Starting applications

There are two ways to start application programs.

3.1.1 Automatic starting as an extension of power-on and system reset operations

In this method, the program is automatically started when the computer is turned on and when a CD-ROM is loaded.

The FM TOWNS is the first personal computer to have a built-in CD-ROM. To take advantage of this feature, operations to start application programs are automated using the following functions:

1) Loading and execution of IPL from CD-ROM
2) Loading and execution of operating system from CD-ROM
3) Automatic starting of an application.

These operating system functions and application programs are recorded in the same area on the CD-ROM. This starting method is used for all single application programs (e.g. game software).

3.1.2 Selecting an application from a menu

This method is used to select and start a program after the operating system on a CD-ROM (or a disk copied from the CD-ROM) has started. This method is used to start two or more application programs on the same

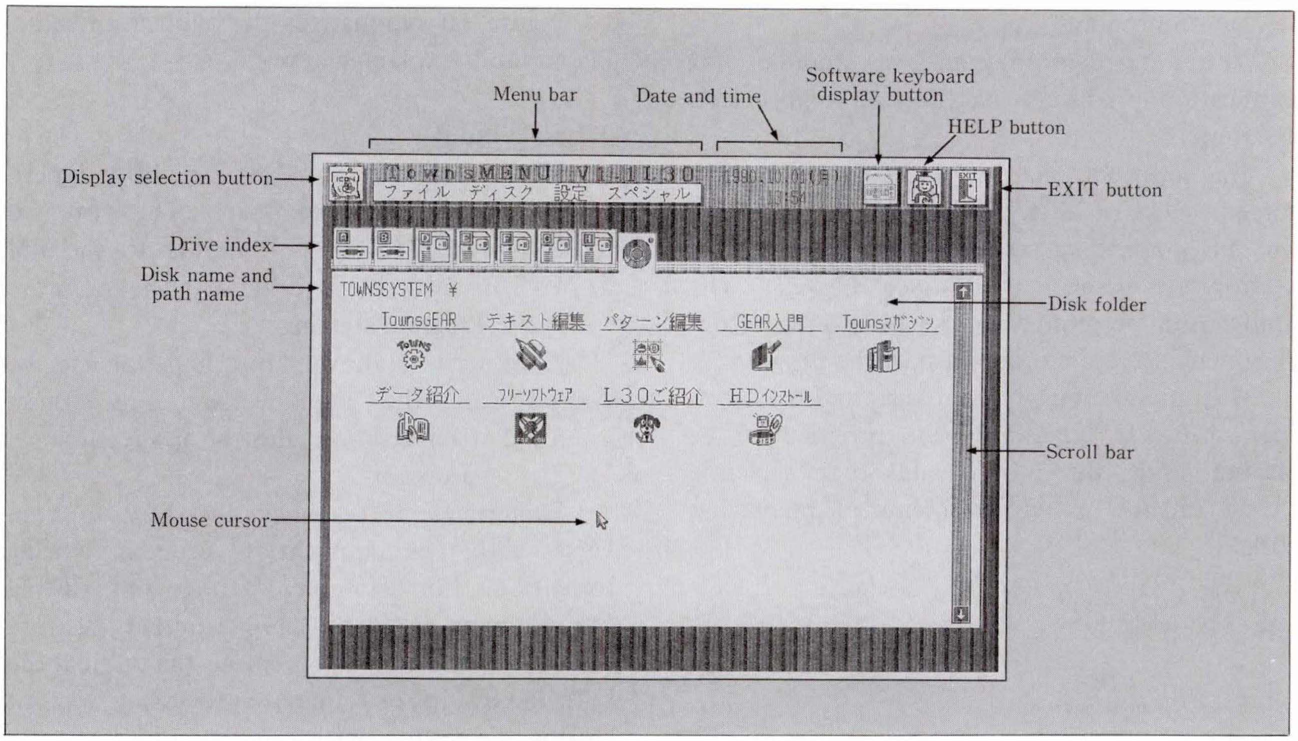


Fig. 8—Graphic menu screen.

CD-ROM. It is also used when an application program is to be copied from CD-ROM to hard disk.

When the operating system starts up, a menu is displayed as the first application program. An application program can then be selected using the mouse (see Fig. 8).

There are three ways to select an application program from the graphic menu:

1) Selecting the icon corresponding to the application program

Each program can be assigned and represented by an icon, and up to 20 assigned icons can be simultaneously displayed. An application program can be started by selecting its icon using the mouse.

Application programs provided by the system are assigned an icon before shipment.

2) Selecting an application program from a menu pulled-down from a menu bar

A menu bar is displayed at the top of the menu screen. Some examples of menu titles are file, disk, set-up, and special.

When a menu title is selected using the mouse, the corresponding pull-down menu is displayed. An application program can be started by selecting one of the menu functions (for example, "copy" of menu bar "file").

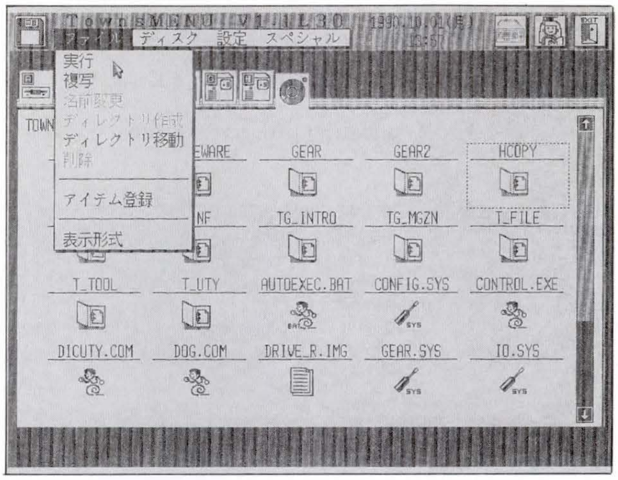


Fig. 9—Application program starting from the file name list.

This method is used for all system utilities (e.g. disk initialization and file copying). In addition, the dictionary management utilities, CD player, and pattern editor are also used.

The file to be processed by the selected function (e.g. a file to be copied) is selected from the file name list.

3) Selecting the file that contains the appli-

cation program

This starting method is suitable for application programs that do not require icon assignment.

In addition to the above icon display screen, a list of all file names can be displayed on the graphic menu screen by selecting the appropriate display mode (see Fig. 9). The application program can also be selected from this menu screen in order to start the program.

The application program is started by selecting the file containing the program to be started using the mouse, and then selecting "run" on the pull-down menu of menu bar "file".

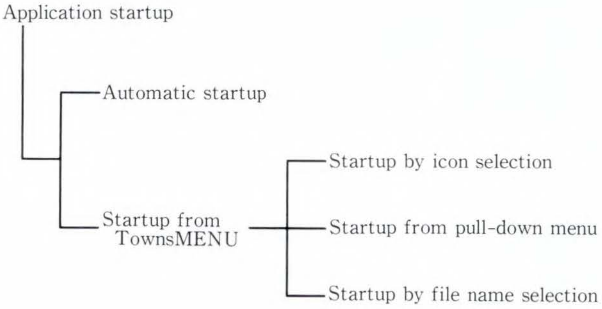

Application startup
Automatic startup
Startup from TownsMENU
Startup by icon selection
Startup from pull-down menu
Startup by file name selection

Fig. 10—Application program startup.

Figure 10 summarizes the application starting methods explained above.

3.2 Data input

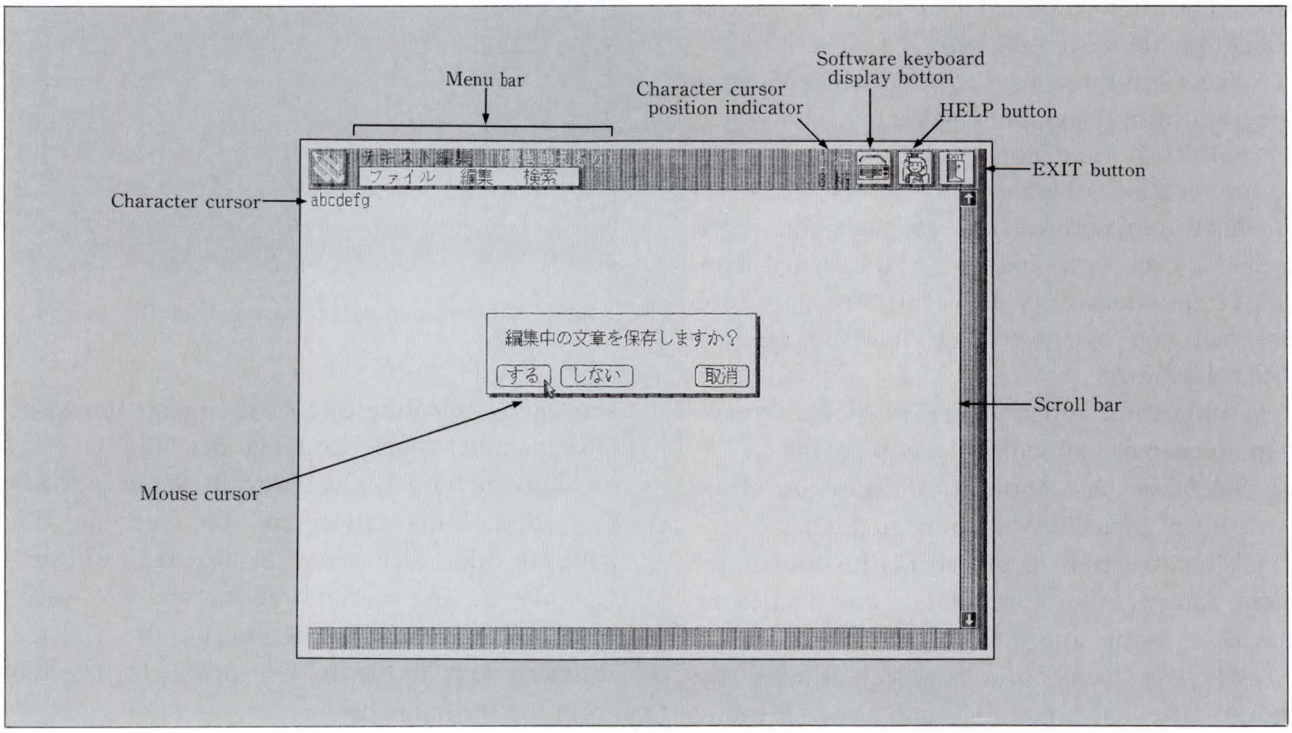
Many users in Japan have had little experience in operating a keyboard. Therefore, to make it easier to use, all inputs to the FM TOWNS are made by pointing with the mouse.

3.2.1 Program starting

As described above, an application can be started using the graphic menu and mouse.

3.2.2 Conversation with the application program

Sometimes, it becomes necessary to converse with the application program during execution. For example, during text editing the program needs to know whether to store the results in a new file, replace the original file with the results, or terminate processing without storing the results. With conventional computer techniques, these instructions are given by parameters specified at program startup or by the character string "save" at termination. To facilitate the above, FM TOWNS allows the application to display options on the screen using the mouse (see Fig. 11).



abcdefg

編集中の文章を保存しますか？

Fig. 11—Example of mouse operation for conversation with an application program.

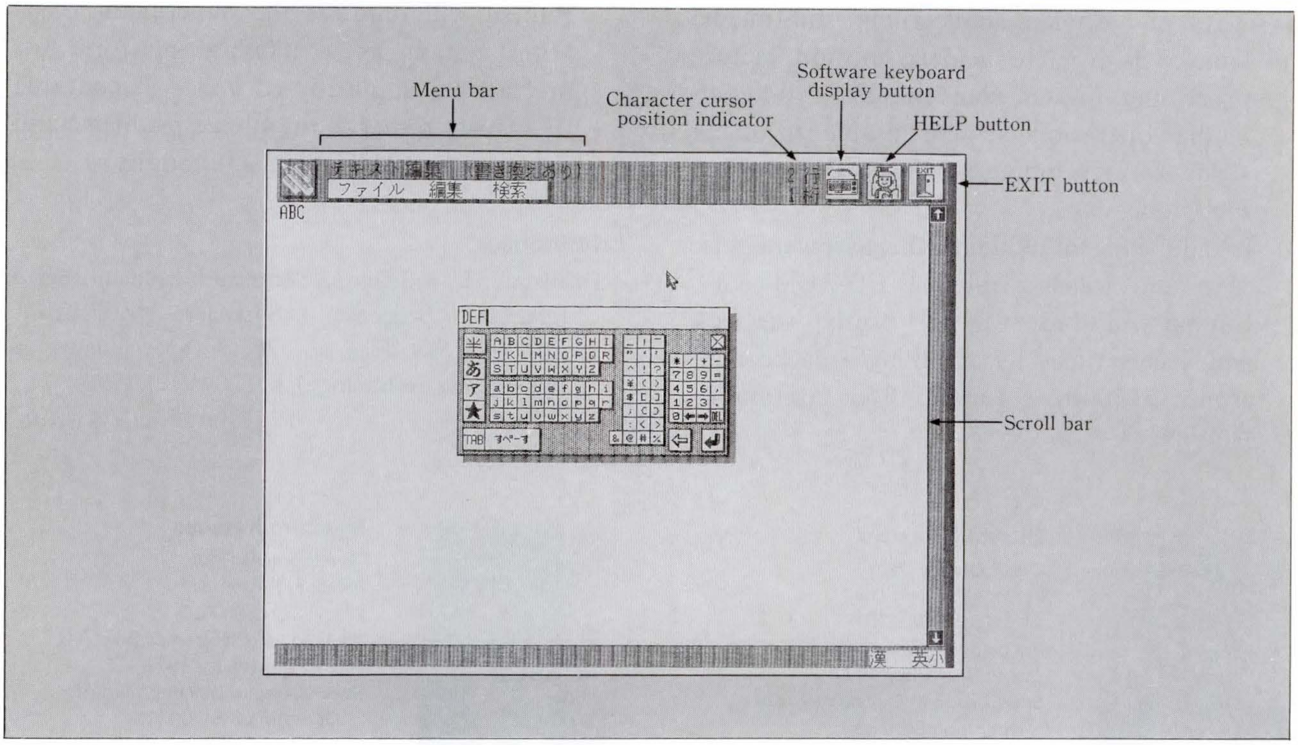


Fig. 12—Software keyboard.

The operating system has a library (GUI library) to standardize and provide these instructions for all application programs.

3.2.3 Input character string

The editing results can be stored in a new file without using the keyboard. This is done using the mouse to select the required characters from a keyboard displayed on the screen (see Fig. 12). (Alphanumerics, katakana, hiragana, and kanji are available.)

A conventional keyboard can be operated quickly once you get used to the key arrangement; but the arrangement is difficult to remember.

The keys on the FM TOWNS software keyboard are arranged in alphabetical or kana order.

The operating system provides a library (software keyboard library) to standardize and facilitate the key operations of application programs.

All operations for the FM TOWNS, from running a program to inputting data, can be performed using only the mouse. (There is no need to use the keyboard.)

4. Conclusion

Fujitsu has implemented a hypermedia operating system on the FM TOWNS and has marketed the resulting product. Several types of application programs, for example, business, creativity tools (music and graphics), and games are currently being distributed for the FM TOWNS on CD-ROM. The FM TOWNS features hardware that has full multimedia functions and can be run using software on CD-ROM. Because of these advantages it has over other personal computers, the FM TOWNS has been enthusiastically received by the market. Fujitsu is now expanding the range of application programs. However, the hypermedia operating system presents many problems regarding operability, application expansion, and the application development environment.

Some of these problems are outlined below.

1) The window functions of the operating system must be improved to provide an environment with several windows.

   Fujitsu plans to solve this problem by implementing Microsoft®-Windows™ as a standard feature.

2) Another problem concerning multimedia control is how to achieve motion picture processing. Fujitsu plans to make this hypermedia operation system usable in other fields by developing an image compression technique.
3) In addition to reducing the hardware seek time and transfer rate of CD-ROM data, Fujitsu will study ways to overlap the seek and access times by installing a cache controller to improve functions for continuous reading.

Fujitsu will improve the hypermedia operating system as a standard platform for multimedia applications, and will continue its efforts to solve the above problems and enhance the FM TOWNS's functions.

References

1) Mizuno, J., and Ono, S.: Musical Instrument Digital Interface Sequencer Software: EUPHONY. *FUJITSU Sci. Tech. J.*, **26**, 3 (Special Issue on Hypermedia Technology), pp. 207-213 (1990).

Hitoshi Nishiuchi
Development Dept.
Personal System Div.
FUJITSU LIMITED
Bachelor of Mathmatics
Osaka University 1971
Specializing in Development of
Operating Systems

Sumio Kimura
Development Dept.
Personal Sustem Div.
FUJITSU LIMITED
Hagi High School of
Engineering 1974
Specializing in Development of
Operating Systems

Masahiro Kataoka
Development Dept.
Personal System Div.
FUJITSU LIMITED
Bachelor of Electronics Eng.
Niigata University 1979
Specializing in Development of
Operating Systems

UDC 001.81:681.3.02

Hypertext System: TownsGEAR

● Sunao Ueda *(Manuscript received July 25, 1990)*

Personal computers are proliferating all over the world, and many programming languages have been developed to improve the efficiency of application development. Until several years ago, programming languages were developed with professional programmers in mind. However, new programming languages and environments are entirely different from conventional ones in many respects. TownsGEAR, one such language, operates on the multimedia personal computer, FM TOWNS.

1. Introduction

TownsGEAR[1),2)] is a multimedia application authoring tool. It is supplied on CD-ROM together with the FM TOWNS.

The FM TOWNS has been developed for a wide range of users. It features enhanced images, sound, and music as well as a high processing speed. These features are controlled by a human computer interface (HCI) so that it can be used even by people unfamiliar with personal computers. The installation on CD-ROM enables users to use the enormous amounts of data required for high quality color images and sounds.

TownsGEAR is used to create applications that take advantage of the above features of the FM TOWNS. It provides a very simple programming environment with which users can create their own applications.

This paper focuses on the following three features of TownsGEAR: the elements it can handle, its program structure, and the formats in which data is represented.

2. Elements TownsGEAR can handle

TownsGEAR can handle multimedia elements in an integrated manner. Media elements that personal computers can handle come in two types: those that can be processed by the computers, and those that require external devices. Among media elements belonging to the first category, the following can be handled by TownsGEAR (all music and sound data is output in stereo):

1) Characters (standard fonts, multiple-fonts, and vector fonts)
2) Values
3) Images (16- or 32 768-color)
4) Music from a Frequency Modulation (FM) tone or Pulse Code Modulation (PCM) source
6) Compact disc digital music and sound from a CD-ROM drive
7) Time information

The media elements that can be handled using external devices are as follows:

1) Music from a Musical Instrument Digital Interface (MIDI)[3)] source
2) Video images from a video disc player.

3. Structure of TownsGEAR

3.1 Object-oriented design

The main features of TownsGEAR is that it makes the hardware features as easy to use as possible and has a software structure that can be adapted to future hardware expansion.

The program structure of TownsGEAR is designed to extend maneuverability, expansibility, and maintainability. An object-oriented program structure has been adopted in which one type of media corresponds to one object (class).

All TownsGEAR objects have their own

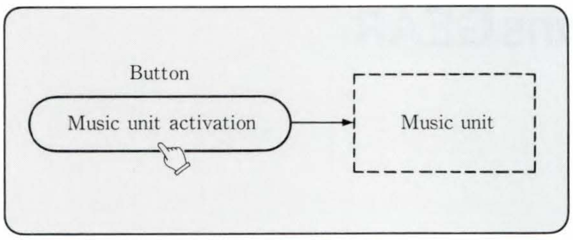


Fig. 1—Link from a button to the music unit.

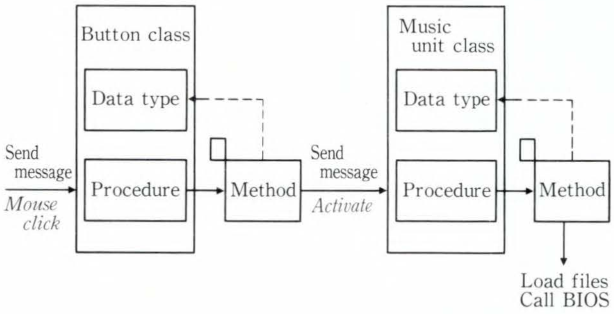


Fig. 2—Program status at the time of button click.

classes. For example, text frames belong to the class "text". Each class is assigned data types (text, attributes, and others) and methods (display, dismiss, editing, deletion, and other processing programs). All processes on objects are transmitted in the form of messages. The method of creating links using buttons is described below (see Figs. 1 and 2).

1) The object to be processed is selected according to the mouse cursor position.
2) A mouse click is found in the event queue.
3) The message "mouse click" is transmitted to the object (button).
4) The button handles the mouse click message.
5) The method checks the button data. If the data indicates a link, the method transmits message "activate" to the link destination.
6) The object at the link destination is activated. In this case, a musical performance starts.

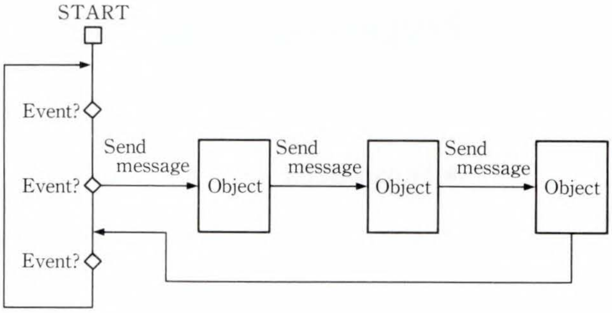


Fig. 3—Structure of main loop.

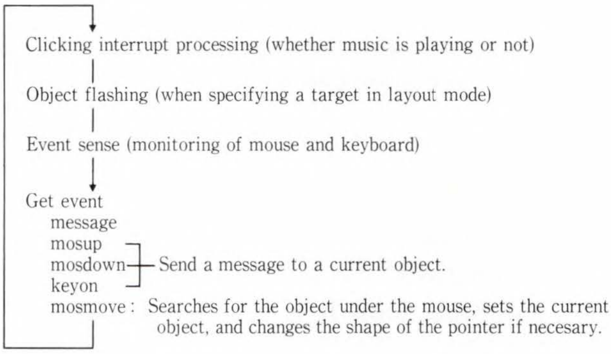

Clicking interrupt processing (whether music is playing or not)
Object flashing (when specifying a target in layout mode)
Event sense (monitoring of mouse and keyboard)
Get event
message
mosup
mosdown
keyon
Send a message to a current object.
mosmove : Searches for the object under the mouse, sets the current object, and changes the shape of the pointer if necesary.

Fig. 4—Operating status at the time of button click.

3.2 Event processing and messages

The HCI library standarizes the HCI and improves efficiency when developing applications for the FM TOWNS. It has about 500 functions. The main routines are event-driven; therefore, TownsGEAR also has event-driven main routines. These events include mouse events, keyboard events, software keyboard events, and post events for TownsGEAR itself.

In general, event-driven programs transmit messages in the following way (see Fig. 3):

1) The location of the event is found, then the event is transmitted as a message to the pertinent object.
2) The method of the class corresponding to the object is activated.
3) If the method ends its processing without transmitting a message to another object (i.e., a primitive method has been activated), control returns to the event wait loop of the main routine.

3.3 Structure of actual event loop

In TownsGEAR, all messages are stacked in event queues. This means that messages are not transmitted from object to object as in the preceding figure. When one method ends, program execution returns to the event loop of the main routine, then the next message is processed (see Fig. 4).

4. User interface

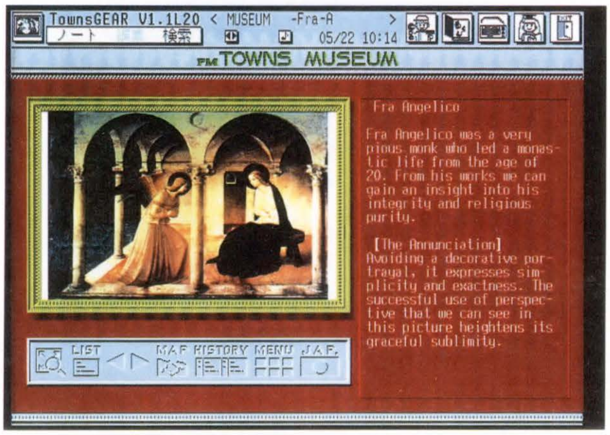
4.1 Notebook

TownsGEAR simulates a notebook to make its application structure easy to understand. This notebook provides a card-type format. The hypertext features systematic linkage implemented using scrolling, cards, and windows. Cards were chosen because they present data in an easy-to-understand way.

An application (notebook) is a set of pages (one screen per page). Components (objects) can be placed on each page. A component is a function that displays characters or images or plays music. Other components establish links between components, pages, and applications and run a program assigned to each component. Using these features, the creation of applications is very similar to the creation of a notebook. Operations correspond to turning pages and finding reference pages. The difference between this system and an actual notebook is that the system is a hypertext system and that audio and time information can be handled in addition to text and pictures.

If more than one page uses the same layout, an overlay function is available. This function allows each page to be viewed through the form, thus reducing the amount of data required. One page of TownsGEAR corresponds to one screen. The resolution is 640 x 480 dots. The screen can display 16 colors. The background has a 32 768-color display mode. This back screen shows through the window of a transparent-color area of the front screen. In this way, these two screens can be simultaneously displayed, one superposed on the other (see Fig. 5).

There are two types of TownsGEAR notebooks (applications): those created by users, and those generated by the system. Those generated by the system are called CONTENTS. As its name implies, CONTENTS is a set of icons arranged on the page to symbolize applications and their directories. The user can open applications or read directories by clicking these icons. When an application is created, one CONTENTS is automatically generated in the directory.



Fra Angelico

Fra Angelico was a very pious monk who led a monastic life from the age of 20. From his works we can gain an insight into his integrity and religious purity.

[The Annunciation]
Avoiding a decorative portrayal, it expresses simplicity and exactness. The successful use of perspective that we can see in this picture heightens its graceful sublimity.

Fig. 5–Example of screen superposing.

4.2 Operation mode

There are two types of operation modes. Layout mode is used at application creation, and execution mode is used when entering data or running applications. In layout mode, the user sets up the layouts, arranges components to display the necessary elements on each page, and specifies the sizes, positions, and colors of the elements. If data to be displayed on the components is already available, the user can designate file names for the frames and units. To enter new data in text or image frames, the user selects the execution mode and activates the editor by clicking the necessary components.

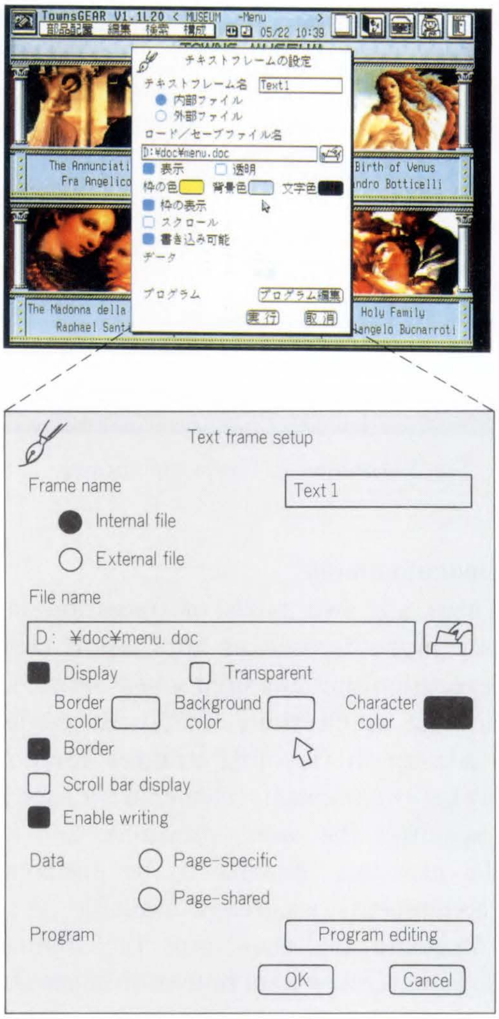
After components have been loaded with data, their attribute is changed to "write-protected" so that they can only be displayed. The editors and attributes are discussed in detail in chapter 5.

5. Types of components

5.1 Text frame

The text frame is an area which displays characters. When clicking in the frame in execution mode, the built-in text editor is started to enable data to be read from an external text file and to enable various kinds of editing.

To exit the editor, click outside of the frame. Data input and display are also controlled by the text editor. Character strings in the frame can be manipulated using programs written in the control language described later.



Text frame setup

Frame name [Text 1]

● Internal file

○ External file

File name

D: ¥doc¥menu. doc

■ Display □ Transparent

Border color □ Background color □ Character color ■

■ Border

□ Scroll bar display

■ Enable writing

Data ○ Page-specific

○ Page-shared

Program [Program editing]

[OK] [Cancel]

Fig. 6—Text frame setup.

The attributes of this component are as follows (see Fig. 6):

1) Frame name

When establishing links between components and between pages and applications, the frame name serves as an ID which designates a link destination if the link destination is a component. If no frame name is specified, the system automatically provides one.

2) Location of text storage

Specifies where to save data.

3) File name

An external text file can be read and displayed. If "external file" is specified for the above attribute, text data will be saved in this file.

4) Display

Specifies whether to display the text frame. This attribute can be controlled by the control-language program so that the text frame can be turned on or off anytime like a pop-up window.

5) Transparent or non-transparent background

When transparent is specified, only characters are displayed. Other components underneath this frame or the background color of the page are also displayed.

6) Colors of border, background, and character

The color of each frame can be specified. Eight colors in high or low intensity can be selected making a total of 16 selections. One of these selections is transparent, which lets the 32 768-color background show through.

7) Borders

Specifies whether to enclose the text frame in borders. If "display" is chosen, the border will be displayed in the specified color.

8) Scroll bar display

If "display" is specified, a scroll bar is displayed on the right side of the text frame. This bar allows the display area to be scrolled.

9) Enable or disable writing

If enable is specified, the text editor starts when a frame is clicked in execution mode. If "disable" is specified, no action is taken.

10) Page-specific or page-shared data

This attribute is only applicable to text frames on pages to which overlay is assigned. Specify "page-specific" to use the same frame position, size, and color allocation for each page, and to assign specific text data to each page.

Specify "page-shared" to share not only the framework but also text between all pages. On pages to which the overlay frame is assigned, the same text frame with the same contents are displayed at the same location.

11) Program editing

If this atrribute is specified, the text editor is started to allow the creation of programs to be attached to the frame. Programs written in control language are discussed in the next chapter.

5.2 16-color display frame

This frame is an area which displays images in 16 colors. Clicking in this frame in execution mode activates the image editor. When the editor

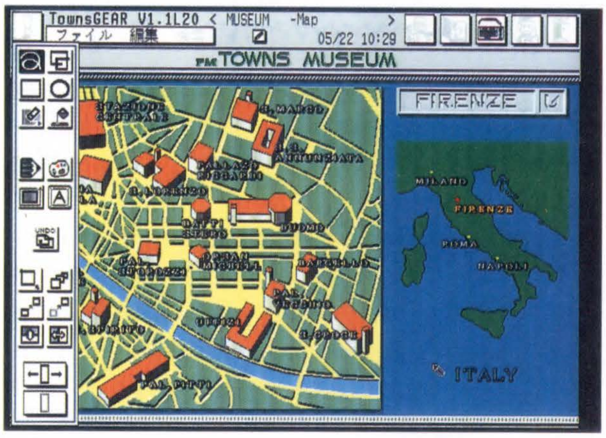


Fig. 7—Editing menu of image editor.

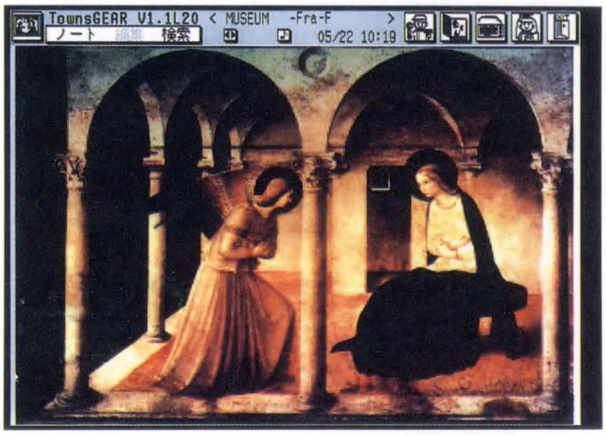


Fig. 8—Image frame displayed in 32 768 colors.

is active, data created by another paint tool can be read, graphics can be drawn, and various kinds of editing can be performed. The editor is deactivated when the mouse is clicked outside of the frame. All standard drawing features are available. The editing features include enlargement, reduction, moving, duplication with enlargement or reduction, inversion, and reversion (see Fig. 7).

Since images require a large amount of data, they are always saved as an external file. This prevents the application file from becoming too big. The attributes of this frame are the same as those of the text frame.

5.3 32 768-color display image frame

This frame is an area which displays image data in 32 768 colors (see Fig. 8). Color data is displayed on the back screen and all other components are displayed on the front screen. This frame is an area used to view the back screen through the front screen. It has two modes: 640 × 480-dot high-resolution mode, and 320 x 240-dot low-resolution mode. In the standard high-resolution mode the frame is one-fourth the size of the screen. In low-resolution mode the frame takes up the entire screen.

Since such color data is rarely changed, this frame cannot be edited. Data to be displayed in this frame can be created in the following three ways:

1) Video digitization

Video and TV signals can be digitized by installing a video interface card into the FM TOWNS expansion slot.

2) Paint tool

Data can be placed in a file using TownsPAINT or other paint tools compatible with TownsGEAR.

3) Image scanners

Photographs or graphics from the color image scanner can be digitized and placed in a file using TownsPAINT or other paint tools compatible with TownsGEAR.

The special attribute that can be assigned to this frame is:

1) Digitize

This attribute specifies use of the digitizing feature. If "on" is specified, clicking in this frame in execution mode causes the FM TOWNS screen to change to a video monitor. If "off" is specified, no action is taken.

5.4 Music unit

The music unit is a component that plays music files. Music data conforms to the MIDI standard, and FM TOWNS music can be played using the built-in sound source as an MIDI source. Therefore, agogics and other effects can be reproduced with fidelity. Clicking in the area of this unit in execution mode causes the music unit to read and play back music files.

The music unit has two playback modes. One uses the internal speaker and allows six parts of an FM tone generator and eight parts of a PCM sound source to be played at once.

Timbre files needed for these sound sources can be created using an editor. Also, files in the same system CD-ROM as TownsGEAR can be used in the data library. The other playback mode uses a MIDI interface card in the expansion slot of the FM TOWNS. This card drives MIDI sound generators.

The sound unit can create music of high enough quality for most applications, so TownsGEAR does not create music files. Data for this unit must be created with other music software.

The attributes of this unit are the same as those of the text frame.

5.5 Sound unit

The sound unit records and plays back sound using a built-in PCM sound source. This unit is set to record or playback mode and then placed on the page. Recording and playback cannot be done at the same time. Sound can be input through the built-in FM TOWNS microphone, an external microphone, or LINE IN.

Operation of the PCM sound source as a music unit differs from its operation as a sound unit. The PCM sound source has a dedicated 64-Kbyte RAM. When playing music, information about tones and timbres of up to eight instruments is stored in this RAM. Loop processing and pitch processing (increasing or reducing frequencies) are performed on this information.

When the sound unit is operating, the RAM is used as a memory for a linear souce. An eight second sample can be taken at a sampling rate of 8 kHz.

The PCM sound source can operate in these two modes at the same time if the appropriate RAM assignments are made; however, the number of instruments available will be reduced to seven.

To prevent an initial silence, the sound unit only starts when the signal reaches a specific level.

The special attribute of this unit is:

1) Recording and playback

This attribute specifies the use of the unit as a recording or playback unit.

5.6 Button

Buttons activate functions or make links to other notebooks, pages, components, and control-language programs. Clicking a button in execution mode executes the corresponding link or program.

Linking passes control to another component. Clicking a button whose link destination is a notebook displays the first page of that notebook. This is also true of pages. If a component is specified, the page on which it is placed is displayed and the assigned fuñction is activated. If the component is a music unit, the music is played. If the component is a 16-color image frame, the image editor is activated. (If the component is write-protected, it is only displayed.)

A program assigned to a button is executed when the button is clicked. But a program assigned to another component is executed when control is passed to that component from the program.

Buttons can be shaped like actual buttons, enclosed in rectangles (with or without rounded corners), or identified by user-specified character strings in the borders. They can also be transparent and displayed as icons. The 128 icons stored in ROM can be used and modified using the icon editor.

Buttons have the following attributes:

1) Button name

Same as that of the text frame.

2) Display

Same as that of the text frame.

3) Colors of border, background, and character

Specifies the color of each frame. Eight colors in high or low intensity can be selected giving a total of 16 selections.

4) Display of characters

If the button is enclosed in a rectangle, this attribute specifies display of the character string in the rectangle. The size of the button automatically changes with the length of the character string. If the button is an icon button, the character string is displayed below the icon and is left-justified.

5) Button shape

Buttons can be rectangles, round-cornered

Table 1. Visual effects

Effect	Screen change
ZOOM OPEN	From center out
ZOOM CLOSE	From periphery to center
WIPE LEFT	From right to left
WIPE RIGHT	From left to right
WIPE UP	From buttom to top
WIPE DOWN	From top to bottom
SCROLL LEFT	From right to left
SCROLL RIGHT	From left to right
SCROLL UP	From buttom to top
SCROLL DOWN	From top to bottom
HORIZON OPEN	From center to top and bottom
HORIZON CLOSE	From top and bottom to center
VERTICAL OPEN	From center to right and left
VERTICAL CLOSE	From right and left to center
DISSOLVE LINE	Dissolve (line)
DISSOLVE POINT	Dissolve (point)
DISSOLVE 16	Dissolve (16 segments)
DISSOLVE 64	Dissolve (64 segments)

rectangles, or icons. An icon can be selected from the icon file.

6) Link destination

Specifies the link to which control will be passed when the button is clicked. When this attribute is being specified, the dialogue box used to set the button and change the button attributies is not displayed. Also, during this specification, control can be passed to any page of any application. Once a link destination is specified, control returns to the dialogue box. Linking and program execution cannot both be selected.

7) Visual effect

When linking is used to display another page or notebook, this attribute selects one of 18 methods in which to move to a new page. The default method is a wipe-down (see Table 1).

8) Program editing

Enables editing of the program to be assigned to the button.

5.7 Timer

Activates a program. The timer functions are similar to the button functions. However, the timer differs from the button in that it activates the program at a specified time and has no link function. The timer attributes are the same as those of the button.

6. Control language

6.1 Outline

Applications can be created by arranging components and displaying appropriate data on pages. Dynamic screen change is also possible using links. TownsGEAR also has a BASIC, called GearBASIC. GearBASIC is useful when creating high-level interactive applications that evaluate input data and perform detailed processing on that data. GearBASIC is a TownsGEAR version of F-BASIC386. Control structures such as IF ... THEN and FOR ... NEXT, arithmetic functions, and CD control functions of F-BASIC386 are embodied in GearBASIC. GearBASIC is a simple BASIC, for example, ON ... is omitted. GearBASIC has no line numbers, and labels are used in GOTO statements. In addition, GearBASIC has statements for object handling and other TownsGEAR specific operations.

6.2 List of statements

The statements of GearBASIC and a function outline of the statements specific to GearBAISC are shown in Table 2.

6.3 SYSTEXT

A 255-byte memory area called SYSTEXT is acquired when TownsGEAR is run. This area can be read and written using GET$ and PUTS of GearBASIC. Programs can transfer data between notebooks via this area. The example below shows a program assigned to a button that creates a calendar for a specific year. This example is on the last page of CALENDAR, a sample page of TownsGEAR, and shows how values are passed via SYSTEXT to the programs of the buttons of each page.

```
A$ = GETS$ (“A.D.”, 0)
PUTS “SYSTEXT”, A$ + “Year”, 1
A = VAL (KACNV$(A$))–1
A$ = AKCNV$ (MID$(STR$(A), 2, 5)) +
     “Year”
```

```
PUTS "SYSTEXT", A$, 2
N = A + 1
C = 0
FOR Y = 1900 TO A
IF (Y MOD 4) = 0 OR (Y MOD 100) <> 0
                AND (Y MOD 400) = 0
THEN C = C + 1
NEXT
D = ((A - 1) - 1900)*365 + C + 304
PUTS "SYSTEXT", STR$(D), 3
EFFECT ZOOM(OPEN)
GO "NOV-DEC/N-D"
END
```

7. Data format

7.1 Standardization of data fomats

Text and image data can be input and edited on TownsGEAR. However, FM TOWNS provides a standard data format in which data files can be transferred between different editors or applications so that sophisticated editing functions of other wordprocessor can be used. Consequently, data files created using other eidtors can be imported or exported from TownsGEAR. TownsGEAR also conforms to

Table 2. GearBASIC statement

a) Statements and functions to act on objects

Statements	Functions
GEAR NEW	Creates a component in the current page
GEAR POS SET	Sets the coordinates of an object
GEAR POS GET	Gets the coordinates of an object
GEAR COLOR SET	Sets the color of an object
GEAR COLOR GET	Gets the color of an object
GEAR SAVE/ LOAD	Saves the file of an object or reads a file into an object
GEAR SHOW/ ERASE	Changes the attribute of an object
GEAR EXEC	Performs the function assigned to the component
DSP ON/OFF	Displays or hides an object temporarily
CLRF	Clears data from a text frame
GETS	Reads a character string from a text frame
PUTS	Writes a character string into a text frame
FIND	Searches a text frame for a specified character string
REPLACE	Replaces a specified character string in a text frame with another character string
GO	Calls another notebook, page or component (program)

b) Statements for drawing

Statements	Functions
VIEW ~ VIEW END	Designates a 16-color image frame for drawing
CIRCLE	Draws a circle or an arc
LINE	Draws a rectangle or a line
PAINT	Fills an area enclosed in boundaries
PSET, PRESET	Displays arbitrary dots in foreground or background color
PALETTE	Changes the assignment of the 16 colors

c) Others

Statements	Functions
GUIDE	Displays a guide (dialogue box)
EFFECT	Adds a visual effect when moving between pages
VERSION	Gets the version number of TownsGEAR

Table 3. TownsGEAR data files

Data type	File format	Extension	Available tool
Text	MS-DOS™ text file	*.*	Text editors
Image in 16 colors	TOWNS standard image file	*.TIF	TownsPAINT, F-BASIC386
Image in 32 768 colors			
MUSIC		*.EUP	EUPHONY
Timbre for FM sound generator	TOWNS standard instrument file	*.FMB	TownsSOUND, F-BASIC386
Timbre for PCM sound source	TOWNS standard instrument file	*.PMB	TownsSOUND, F-BASIC386
Sound for PCM sound source	TOWNS standard sound file	*.SND	TownsSOUND, F-BASIC386

Table 4. Towns data library

File type	Number of files	Remarks	Quantity
Instrument for FM sound generator	7	51 instruments have been added to the 77 instruments of early FUJITSU personal computers	6
		This file contains the most popular MIDI sounds in Japan in the standard order	1
Instrument for PCM sound source	8	Drums	1
		Electric drums	2
		Combination of rhythm instruments	5
Music	5	Piano music by Chopin	2
		Original music	3
Message	262	Words of greeting	86
		Words of warning	27
		Words of instruction	40
		Words expressing emotion	67
		Notice	20
		Others	22
Sound effects	187	Nature	15
		Street	14
		Festival	8
		Animals	31
		Vehicles	20
		Passages of Japanese music	8
		People	14
		Crowd	6
		Cartoon	3
		Weapons	10
		Domestic sounds	27
		Sounds of a period drama	12
		Sports	19
Line drawing illustrations (one-forth the screen size)	214	Logos, characters	42
		Icons	27
		Cards	9
		Patterns, frames	5
		For letters and postcards	67
		Cooking and related matters	14
		Stamps	18
		For calendars	26
		Illustrations in 16 colors	6
Line drawing illustrations (full screen) More than one picture/screen	189	Logos, characters	11
		Four seasons	14
		Maps and related matters	11
		Stamps	6
		Living things	14
		Personal belongings	5
		Clothes	2
		Vehicles	4
		Postal matters	11
		Events	12
		Photographs (in 32 768 colors)	5
		Color illustrations (in 32 768 colors)	6
		Icons	19
		Patterns, frames	7
		Plants	9
		Landscapes from around the world	12
		People	15
		Room interior	4
		Food	12
		Make-up	2
		For animations	1
		For cartoons	4
		Learning, study	3

this standard file format. Various data files of 16- or 32 768-color images, or sounds can be generated in TownsGEAR. The data listed in Table 3 is available.

7.2 Text files

Text files are in standard MS-DOS™ format. The following can be read and written: data files created using the text editor contained in the same CD-ROM as TownsGEAR, various other text editors, and various wordprocessors which run under MS-DOS™.

Programs to be assigned to components can also be created using other text editors and can be read into the program editor screen of TownsGEAR. Text is saved in the application file.

7.3 Image files

Image files conform to the Tagged Image File Format (TIFF) adopted by various editors and application software, including Aldus Page Marker®.

7.4 Music files

Music files can be read and edited by EUPHONY[3]. The music data consists of performance information compatible with the MIDI standard and instrument files. Music files are not compatible with editors of other computers. However, they conform to MIDI which is a universal data format; so, once they are output as MIDI data any MIDI sequencer can use its own software to read and edit them.

7.5 Other files

Timbre files and sound files are data files for the internal sound sources of FM TOWNS. They can only be read or written by the editors and application software of FM TOWNS.

8. Data library

The storage capacity of the system CD-ROM that holds TownsGEAR is approximately 540 Mbytes. The FM TOWNS data library is provided to take advantage of this enormous capacity and to enable easy access to data.

The total storage of the data library is approximately 50 Mbytes. The contents of the data library are listed in Table 4. The library contains a wide variety of high-quality data, including data that is difficult to create. For example, data files for the sound unit contain 262 phrases spoken by famous actors and 187 sound effects carefully selected from the huge sound effects library at NHK (Japan Broadcasting Corporation).

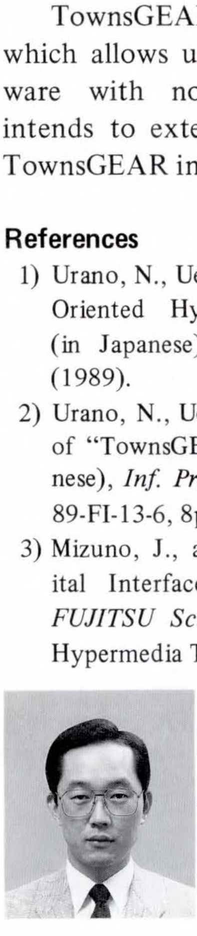
9. Conclusion

TownsGEAR is a new platform for applications of the FM TOWNS multimedia personal computer. TownsGEAR provides an easy to use environment for beginners because it uses pull down menus and icons. Fujitsu will further improve TownsGEAR, and at present is expanding the functions and improving its operation efficiency. Enhancing TownsGEAR application software is also an important theme. One of the objectives for developing TownsGEAR was program development by users rather than programmers.

TownsGEAR provides an environment which allows users to develop application software with no previous experience. Fujitsu intends to extend the range of applications of TownsGEAR in the future.

References

1) Urano, N., Ueda, S., and Sakazume, H.: Multimedia Oriented Hyper Text System "TownsGEAR". (in Japanese), *NIKKEI BYTE*, **61**, pp. 287-293 (1989).
2) Urano, N., Ueda, S. and Sakazume, H.: Developing of "TownsGEAR" as a multi-media tool. (in Japanese), *Inf. Proc. Soc., Jpn., S. I. G. Rep.,* **89**, 47, 89-FI-13-6, 8p., (1989).
3) Mizuno, J., and Ono, S.: Musical Instrument Digital Interface Sequencer Software: EUPHONY. *FUJITSU Sci. Tech. J.*, **26**, 3 (Special Issue on Hypermedia Technology), pp. 207-213 (1990).



Sunao Ueda
Personal Computer Dept.
FUJITSU PERSONAL COMPUTER SYSTEMS LIMITED
Bachelor of Music
Musashino Conservatoire of Music 1975
Master of Music
Musashino Conservatoire of Music 1977
Specializing in Piano Music and Acoustics

UDC681.3.064:79.077

Hypermedia Personal Computer Communication System: Fujitsu Habitat

● Kazutomo Fukuda ● Tadayuki Tahara ● Toru Miyoshi

(Manuscript received March 16, 1990)

This paper describes the technological development of Fujitsu Habitat, and outlines its structure and features. Fujitsu Habitat is a new type of personal computer communication system that supports multimedia processing. It synchronously generates pictures, sounds, and characters — a feat which cannot be performed by conventional personal computer communication systems.

This system is operated by linking Fujitsu's FM TOWNS hypermedia personal computer to a UNIX host machine.

Communication between the host and terminals is done using a unique communication protocol that ensures high-efficiency and high-reliability data transmission.

1. Introduction

The FM TOWNS hypermedia personal computer has been on the market since February 28, 1989. The standard model comes with a 540-Mbyte CD-ROM drive and powerful audio-visual functions to facilitate hypermedia processing.

This paper describes Fujitsu Habitat, which is a network based multi-player game utilizing the hypermedia capabilities of the FM TOWNS. Fujitsu Habitat creates an imaginary world in a network host computer. People having access to an FM TOWNS can live in that world and interact with other users.

Each user is given a body and an interchangeable head. Users can talk with each other and move around in a virtual world-animation. Fujitsu Habitat is an interesting and very enjoyable audio-visual multiplayer game. The present population of Fujitsu Habitat is about 2 000 and is expanding.

Currently, this service is offered as a personal computer communication service of "NIFTY-Serve" by N.I.F. Corp. (a joint corporation of Fujitsu and NISSHO IWAI Corp.).

Although Fujitsu Habitat is just a game at this stage, some people have formed companies in that world, and are even doing business. This shows how a hypermedia network society can evolve, and how it may become useful. It is a challenging experiment and we are very interested in seeing how it develops.

Habitat was originally developed by LUCASFILM Ltd.[1] in the U.S.

Fujitsu licensed it from LUCASFILM Ltd. and then modified it into Fujitsu Habitat.

We have modified all the pictures (e.g. heads, bodies, scenery, and buildings) and audio data to suit Japanese users.

The functional distribution between the network host computer and the FM TOWNS[2] has been redefined to fully utilize the hypermedia capabilities of the FM TOWNS. All the data for pictures and sounds is contained in a CD-ROM loaded in all participating FM TOWNS computers. Communication is very efficient because the only information that needs to be sent is information required to select pictures

Fujitsu Habitat is based on LUCASFILM technology. The UNIX operating system was developed by and is licensed by AT&T.

Table 1. Basic commands

Command	Functions
GET	Get object
PUT	Place object
DO	Take action (e.g. sit on chair, open/close door)
GO	Walk
HLP	Get help message
WSP	Wisper to specified person

Table 2. Function commands

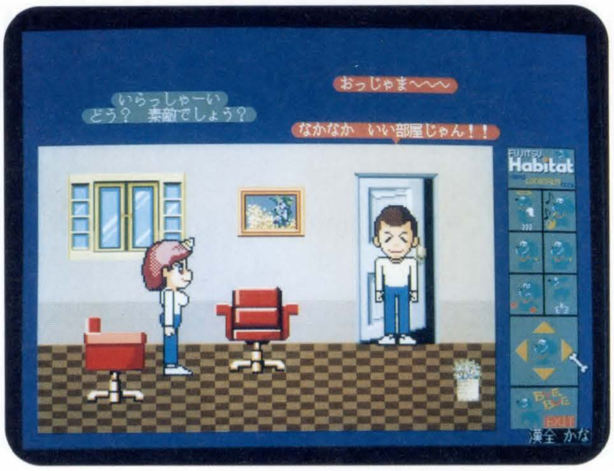
No.	Functions
1	Change Avatar
2	Change sound or music
3	Raise right hand
4	Raise both hands
5	Punch
6	Jump
7	Change direction
8	Go back to menu screen and terminate Fujitsu Habitat

and sounds. This avoids the need to transmit bulky multimedia data. The communication protocol has also been improved to achieve higher transmission efficiency and reliability.

2. Outline of Fujitsu Habitat and its system

Fujitsu Habitat is an imaginary visual world created in a network. It consists of zones, some examples of which are the residential zone, downtown zone, park zone, forest zone, and dungeon zone. Each zone consists of more than ten interconnected regions. A region equals one screen display. For example, the downtown zone consists of a cafe, a bar, and toyshop. There are about four hundred scenes in total.

When a user subscribes to this system, he or she must first select a body and a head (available at the head shop) to form a character referred to as the user's Avatar. (An Avatar is the combination of a body and one of more than 1 000 ready made male, female, old, young, animal and monster heads.) The user can then start his and her life in Fujitsu Habitat through an Avatar from a condominium in the residential zone. Money to live on is given when the user subscribes. Users manipulate their Avatars by giving commands using a mouse. For instance, if a user gives a GO command while pointing to a place on the ground, his or her Avatar will walk to that place. If it is given a command to go to the edge of the screen, it moves to the adjacent area. If it is given a GET command for an object on the ground, it will pick it up. These basic commands are shown in Table 1. Table 2 shows function commands for other movements such as raising hands, changing direction, and punching.


おっじゃま〜〜〜

いらっしゃーい
どう？ 素敵でしょう？

なかなか いい部屋じゃん！！

Fig. 1–Terminal screen (scene in a condominium).

When users meet other Avatar in the screen world, they can talk to the users of those Avatars through the keyboard (see Fig. 1).

In this imaginary world, an Avatar can go shopping with money and chat with friends. Many users can have simultaneous access to the same screen and thereby have an interactive experience in another world.

Figure 2 shows the network configuration of Fujitsu Habitat. It consists of the FENICS network (Fujitsu digital network) and public lines.

Fujitsu Habitat can be accessed at the standard national charges by connecting an FMTOWNS (hereinafter called a terminal) to an access point. The host center consists of the Fujitsu Habitat Center and the NIFTY-Serve Center.

Access to Fujitsu Habitat requires an ID card subscription to NIFTY-Serve. The NIFTY-Serve Center is responsible for password control and

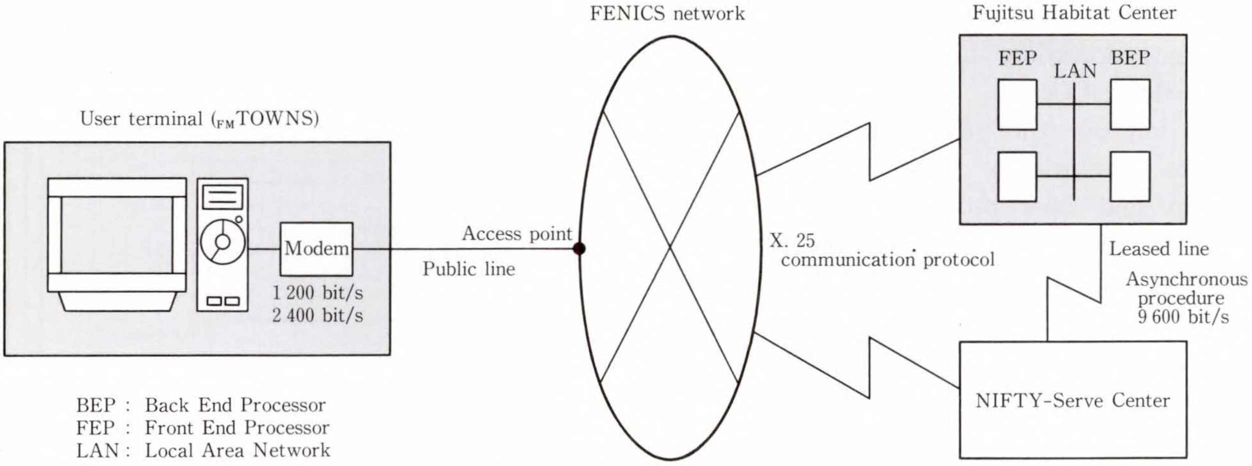


Fig. 2—Configuration of Fujitsu Habitat network.

Table 3. Service requirements of Fujitsu Habitat

Item	Improvement
Realtime processing	Use a process to share data between host and terminals that minimizes communication traffic.
Terminal operability	Make Fujitsu Habitat easy to use.
Expansibility	Use a program structure that allows the Fujitsu Habitat Center to be expanded into a multi-host system in anticipation of large increase in the number of users. Distribute processing between host and terminals to minimize software modifications even when the functions are extended.
High reliability	Localize troubleshooting and automated task recovery processing. Use a communication protocol suited to public communication networks.

accounting. To meet these service requirements, improvements have been made to realtime performance, terminal operability, expansibility, and reliability. These improvement are outlined in Table 3 and are discussed in the next chapter.

3. System configuration

3.1 Concept of function distribution between host system and terminals

The functions of Fujitsu Habitat are distributed between the host system and terminals. A very important aspect of designing a distributed processing system is how to assign the functions between the host and terminals. The response time is greatly affected by this assignment.

When transmitting visual data, it is usually important to minimize the amount of data to be sent. In the case of Fujitsu Habitat, direct transmission of visual data would involve more than 10-Kbyte per transaction. This would not be practical from the viewpoint of response time. To solve this problem, the image data of Fujitsu Habitat is stored in the terminal's CD-ROM. The parts of control information are controlled by the Fujitsu Habitat Center. Images are retrieved from the CD-ROM according to commands from the Fujitsu Habitat Center.

The same processing method is also used for musical tones and audio effects.

By adopting this method, this system has a response time short enough for a realtime system. This method gives the flexibility required to immediately act on the control information sent from the Fujitsu Habitat Center in response to events occuring in Fujitsu Habitat.

The host program controls object information indicating the following: whether an object (e.g. a tree, building, or head) can be picked up or carried, how various objects can interact with each other, Fujitsu Habitat's world map information, and Avatar information (indicating user IDs, and the assets and positions of Avatars). It also issues commands to the terminals.

The terminal program processes animation (e.g. walking, jumping, and hand raising) and generates audio effects (e.g. a bang when a door is closed) according to commands from the Fujitsu Habitat Center.

To improve the efficiency and reliability of data transmission between the terminal program and host program, a unique communication protocol called the Fujitsu Habitat-link protocol (hereinafter called H-Link) was developed (see section 3.4).

We will now outline the total system processing flow for the function distribution mentioned above.

The Fujitsu Habitat system consists of three major phases: the log-in phase, service phase, and log-out phase.

In the log-in phase the host transfers the user ID and password transmitted from the terminals to the NIFTY-Serve Center. It then receives the sendback information from the NIFTY-Serve Center after checking it against the user control information kept there. The host then issues a request for the accounting information. While this is happening, the terminal displays a Fujitsu Habitat picture (see Fig. 1) and waits for the user's input. When it receives the input the terminal changes to the service phase.

In the service phase, the terminal processes instructions from the mouse and conversation text input from the keyboard. It then post the actions and associated coordinates to the Fujitsu Habitat Center as an internal command. The host then does the following: analyses the transferred internal command, determines whether the command can be carried out, determines the consequences of the action (if executable), and finally transfers the results (i.e. the changes to be made to the animation) as an internal command. The terminal then processes the animation and waits for the next input from the user.

In the log-out phase, the user issues the log-out instruction; then the terminal posts it to the Fujitsu Habitat Center and disconnects the line. The host updates the user control information to the latest user status and informs the NIFTY-Serve Center that the accounting information has been obtained.

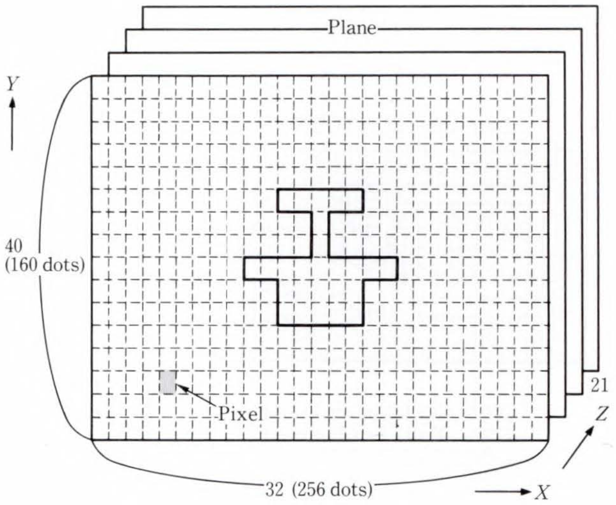


Fig. 3—Map configuration.

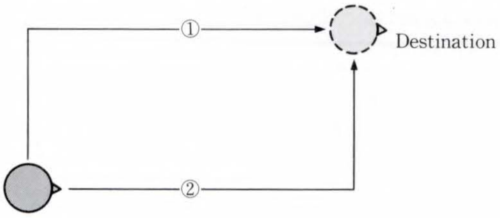


Fig. 4—Avatar movement route (top view).

3.2 Screen control

The screen control function enables display of a smoothly animated three-dimensional world. This function achieves this by mapping and by separating objects into parts. The attractiveness and realism of the Fujitsu Habitat display owes much to this function.

The map is an area in the host which indicates how an object is to be displayed and positioned in a region. As shown in Fig. 3, it is a three dimensional expression with X being the length, Y the height, and Z the depth. Objects in the screen background are positioned in the deepest plane. Movable objects and Abatars are positioned in other planes. The terminal draws the screen in stages from the back plane to the front plane. This process superposes near objects over far objects. The map is also used to identify objects pointed out by the mouse.

When a user issues the GET command for an object on the screen, the terminal sends the object's X, Y coordinates and the command type. (The terminal has no concept of depth and does not handle Z-coordinate data.) The host program then checks whether objects found in the *Z* planes at the *X*, *Y* positions are portable. (The planes are checked starting with the front plane).

For a GO command, the host program checks whether the specified position is already occupied. Route checks are made in two ways to determine whether a specified route is clear (see Fig. 4). When a movement is completed, other Avatars in the scene are reoriented so that they all face the screen. This reorientation is made in preparation for the next move instruction from the mouse.

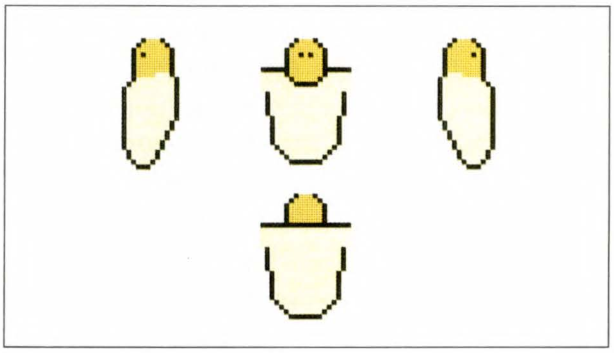
The FM TOWNS has a powerful sprite function, and the terminal program makes full use of it to display animations. To give depth to the scene, display priority is given to objects at the front. The nearer an object is to the back plane, the lower its display priority. Object image data is divided into small segments to enable smooth animation of a variety of movements. For example, an Avatar consists of fourteen segments, each of which is further divided into four patterns corresponding to views from the front, rear, right, and left. Further more, each pattern can be viewed from sixteen different angles (see Fig. 5).

When making an animation, the terminal draws the parts according to commands from the host. When an Avatar walks or raises its hand, all necessary control information specifying the parts to be moved for each phase and the delays between displays is simultaneously managed. The commands that must be simultaneously executed are collectively referred to as cell data. The terminal program begins an animation by referencing this data. The selection and display order of cell data can be changed by a command from the host. An action such as "face forwards and walk backwards" is processed by modifying the host program.

3.3 System control

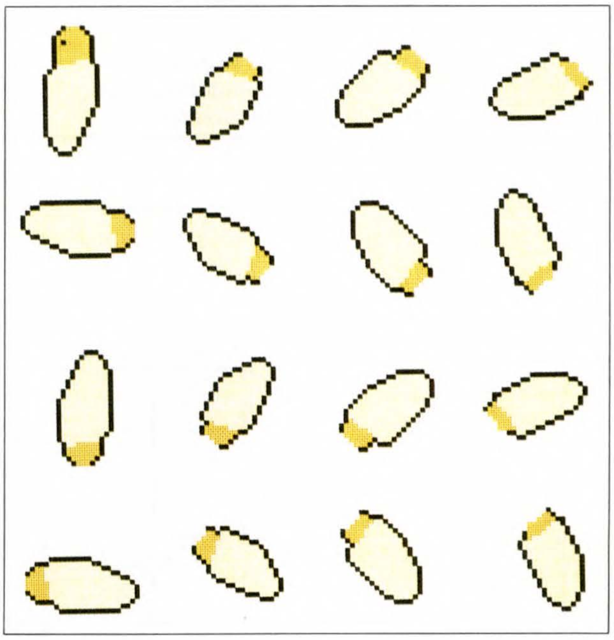
This section introduces the concept of multi-host distributed processing, file control, and impediment control of the host computer system.

Left	Front	Right
	Rear	



a) Four views of a body

0°	30°	45°	60°
90°	120°	135°	150°
180°	210°	225°	240°
270°	300°	315°	330°



b) Further divisions of body views

Fig. 5—Object parts.

In anticipation of future increases in the number of users and communication paths, the Fujitsu Habitat Center was built as a multi-host structure using the LAN system to balance the load between host computers. The processing is distributed as shown in Table 4.

For inter-task communication using a multihost configuration, it is usually necessary for the inter-task communicators to decide which host should have the other task. Furthermore, because LAN communication of this system is made at the inter-host communication section, an application program can be produced without concern for the physical configuration. Figure 6 illustrates this concept.

The DB control section controls the Fujitsu Habitat data base. This data base is a collection of smaller data bases, for example, the user ID; user information data base for names; region data base for the region connection status; object data base for region manipulation for objects; and the mail, bulletin board, and document data bases for the text of memos. This information is controlled using C-ISAM[note)], and deadlocks are avoided by using a general exclusive control system.

Table 4. Distribution of processing

Host type	Major distributed functions
FEP (Front End Processor)	Communication control section User control section NIFTY-Serve control section Inter-task communication section LAN communication section
BEP (Back End Processor)	Region control section Data base access section Inter-task communication section LAN communication section

We will now describe the fault control method used in this system.

During online operation, if an application task (consisting of a user control section and a region control section) fails to continue processing due to an abnormality (e.g. a process contradiction), the work can still be continued without online disconnection. This is achieved by activating the task recovery function, which recovers the failed task only (see Fig. 7).

Task recovery processing is done in the following stages: return of stored data in the task, garbage collection such as enforced unlock-

Note: C-ISAM is a registered U.S. trade mark of INFORMIX SOFTWARE Inc.

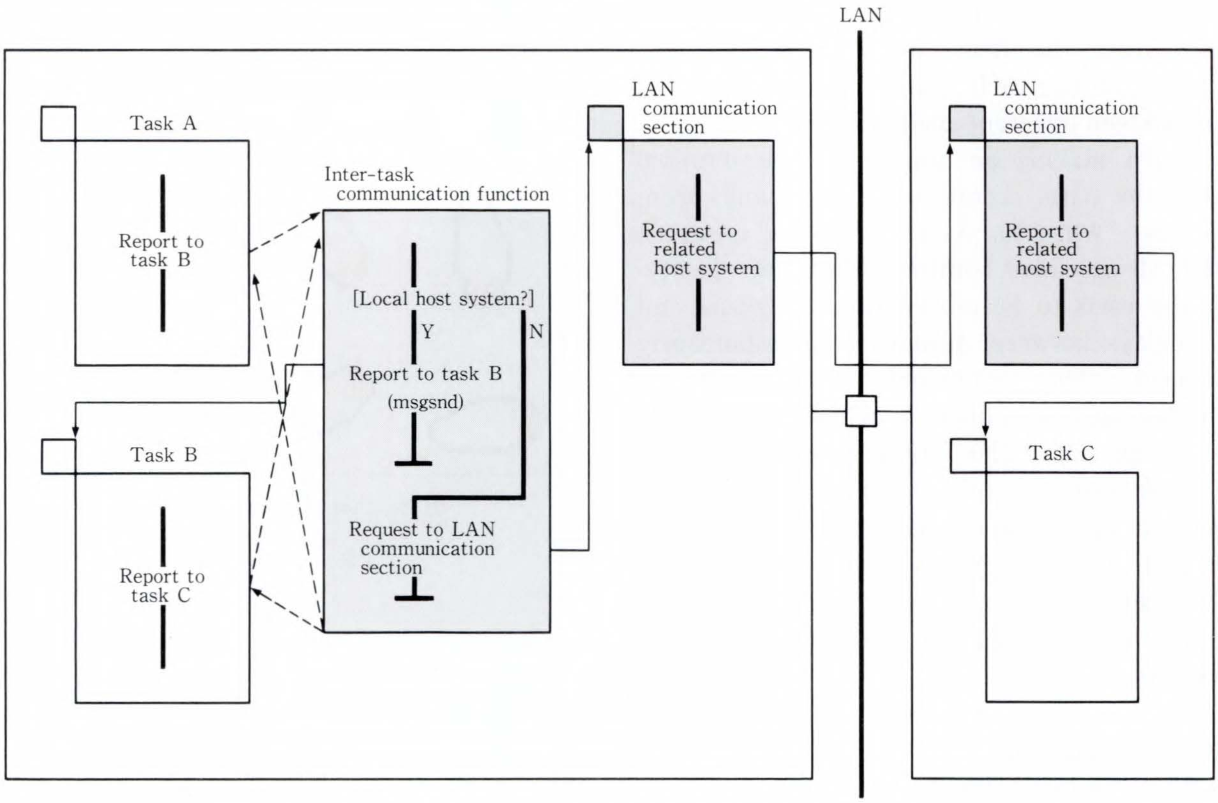

LAN
Task A
Report to task B
Inter-task communication function
[Local host system?]
Y
N
Report to task B
(msgsnd)
Request to LAN communication section
Task B
Report to task C
LAN communication section
Request to related host system
LAN communication section
Report to related host system
Task C

Fig. 6–Concept of inter-task communication.

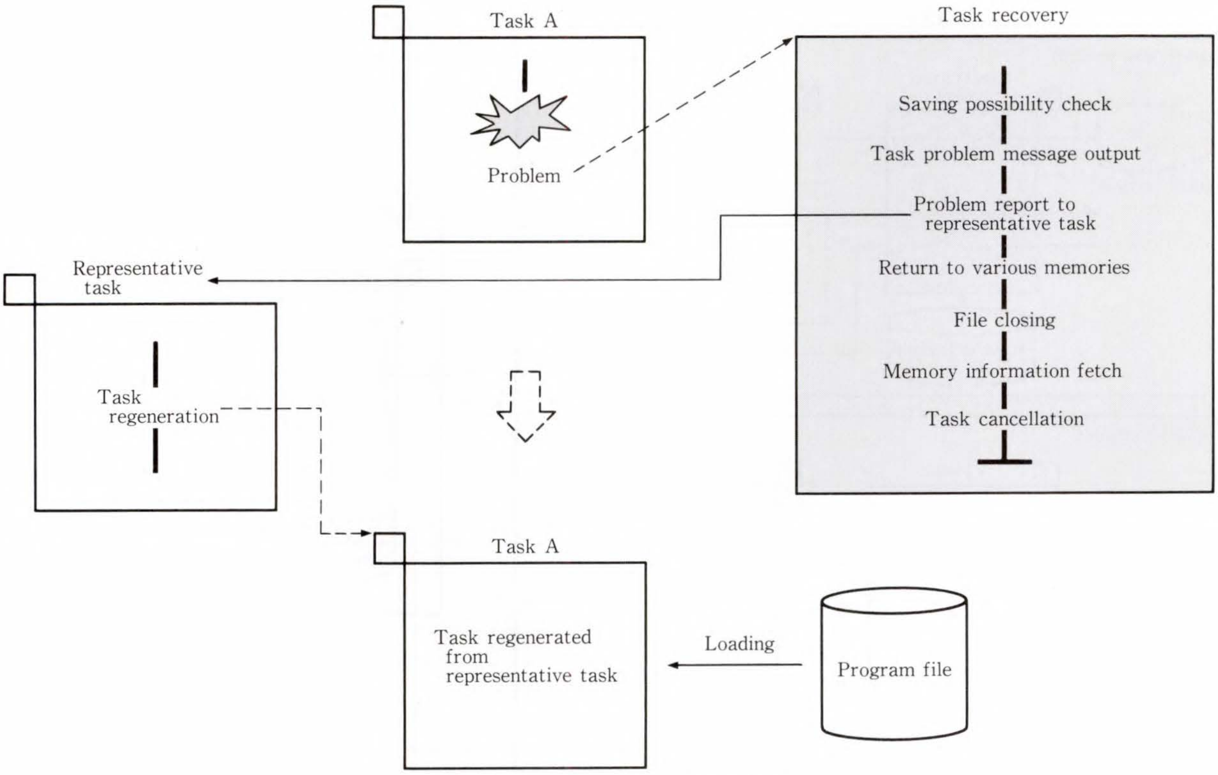


Fig. 7—Concept of task recovery.

ing of locked resources, collection of memory information required for failure analysis, output of a message giving the error cause, and program reloading.

Task recovery processing does not perform control line tasks other than application tasks.

For a multi-host configuration, system monitoring must be integrated to reduce the burden on the center operator and to enable the system controller to monitor the total system. In this system, a master/slave relationship is established between hosts, and work can be done through the master host. The master/slave relationship is stored in the host definition file, so the starting program must first read the memory. The console terminal of the master host not only stores the system control inputs for start and termination of Fujitsu Habitat, but also displays messages received from each program. The slave host accepts non-interference commands, for example, running status indication commands (see Fig. 8).

3.4 H-Link protocol

This section describes the concept and characteristics of the H-Link communication protocol, used between Fujitsu Habitat and terminals.

3.4.1 Background

Fujitsu Habitat was designed to transfer pictures, music, and sound effects between the host and terminals with the minimum of data transmission.

Another design target was low-cost yet high-reliability data transmission. Because data exchanged between the host and terminals is not image information but control information, data transmission errors might severely disrupt the service.

Fujitsu Habitat can be accessed via public communication networks using general purpose hardware for personal computer communications.

Personal computer communication is usually performed using communication protocols, for example, asynchronous procedures and XMODEM (see Table 5), or using special hardware such as an MNP$^{note)}$ modem. Both these methods have advantages and disadvantages

Note: MNP is a registered trade mark of Microcom Inc.

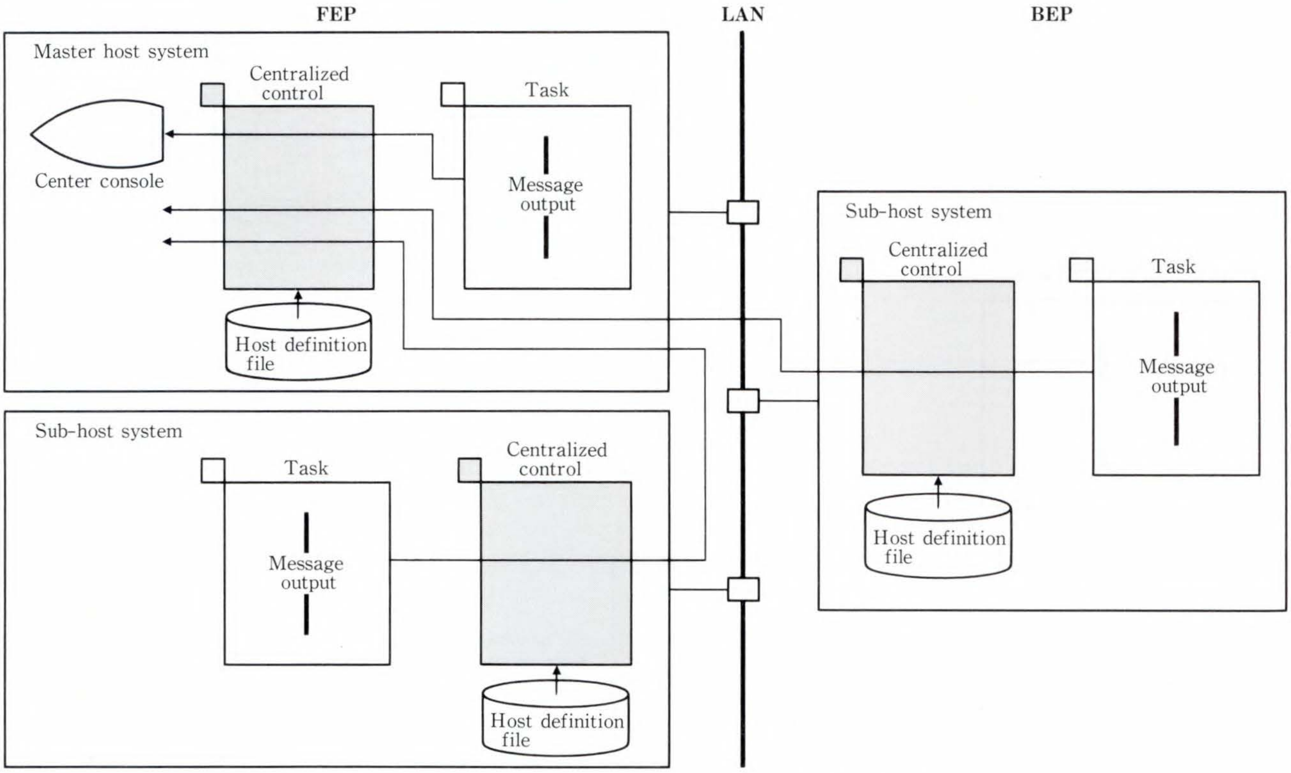
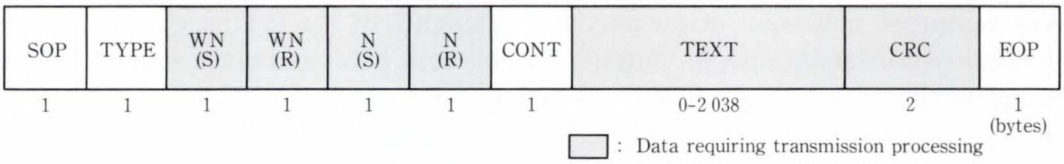


Fig. 8—Centralized supervision at Fujitsu Habitat Center.



SOP	TYPE	WN (S)	WN (R)	N (S)	N (R)	CONT	TEXT	CRC	EOP
1	1	1	1	1	1	1	0-2 038	2	1 (bytes)

▨ : Data requiring transmission processing

Fig. 9—H-Link protocol data format.

Table 5. Comparison of communication protocols

Communication protocol	Efficiency	Reliability	Cost	General evaluation
Asynchronous procedures	Low	Low	Low	Low efficiency and reliability
XMODEM	Low	Medium	Low	Low efficiency due to confirmation of ACK/ NAC transmission per packet
MNP mounted modem	High	High	High	Too expensive due to adjustments required for user's hardware

regarding efficiency, reliability, and running costs.

To satisfy the design goals of efficiency, reliability, and economy for Fujitsu Habitat, we developed the H-Link communication protocol.

3.4.3 Data format of H-Link protocol and its characteristics

Figure 9 illustrates the data format of H-Link protocol.

Conversational messages, image control, and other information is transferred between host and terminal in this format. One information packet includes control information and transmission data having a variable word length. The maximum amount of data in one packet is 2 048 bytes. Each packet starts with SOP (Start Of Packet) and ends with EOP (End Of Packet). The type section of each packet indicates the type of information it contains (see Table 6).

If a message is too large for one packet, the

Table 6. Functions of TYPE section

TYPE	Information type	Function
"I"	Data (information)	Used for normal data
"P"	Data (polling)	When the number of windows reaches a specified value
"R"	Response request (request)	Confirmation request
"A"	Status report (ACK)	Normal status response
"N"	Resend request (NAK)	Packet resend request

Table 7. Types and functions of CONT section

CONT	Information type	Function
"O"	ONLY	Indicates packet contains entire message
"F"	FIRST	Indicates packet contains first part of a multi-packet message
"M"	MIDDLE	Indicates packet contains middle part of a multi-packet message
"L"	LAST	Indicates packet contains last part of a multi-packet message

Table 8. Features of H-Link protocol

Targets	Characteristics
High transmission efficiency	Data can be transmitted and received as binary data. Window control enables continuous sending of packets. Capable of full duplex communication.
High reliability	CRC error detection. Confirmation of lost packets and overlaps using a sequence number. Non-response detection using time monitoring. Monitoring of non-communication and assessment of line quality at a rated distance.

remainder is transmitted using additional packets. For multi-packet messages, reception confirmation for each packet would lower the transmission efficiency. To overcome this problem, the required number of packets (window numbers) is determined by the host and terminal, and confirmation is made when all packets have been transmitted. The window serial number (WN) checks for logical errors in the received transmission. The packet serial number (N) prevents a packet from being lost or overlapped with another packet.

CONT (Continue) indicates whether a message continues in another packet, and is used by the receiver to assemble the packets. The types and functions of the CONT section are shown in Table 7.

The text section contains a transmission message or object control data. Transparent processing of data is performed to enable the sending and receiving of binary data.

The CRC section is the CRC value from SOP to TEXT. If a CRC error is detected, a retransmission request is made.

The use of the H-Link message format ensures a high transmission reliability and efficiency. The characteristics of this format are summarized in Table 8.

4. Conclusion

This paper has focused on the key technologies used in the Fujitsu Habitat system; it has also outlined the system configuration and programs.

This system was developed as a personal visual communication system. By minimizing the volume of data to be transmitted, it achieves a higher response speed than conventional graphic communication systems. This minimization and subsequent high speed was made possible by Fujitsu's large-capacity CD-ROM mounted personal computer FM TOWNS, and by the development of the H-Link protocol.

This service was initiated on January 26, 1990, and was initially provided free of charge. It was continued on a commercial basis starting from February 10, 1990. Within its first six months, more than 2 000 people ranging from eleven to sixty years of age had become users. All users, young and old, male and female, are enjoying themselves in Fujitsu Habitat. (Some enthusiastic members are reportedly spending more than 150 h/month on the program.) This new type of system for communication by personal computer is now attracting considerable attention.

Fujitsu is now developing a "Gameland" and a "Business Land" to further entertain the inhabitants of Fujitsu Habitat and other customers. These developments are expected to be completed within a year.

References

1) Morningstar, C., and Farmer, F.R.: The Lessons of Lucasfilm's Habitat. 1st Conf. Cyberspace, Austin, 1990, Univ. Texas, pp. 1-16.
2) Kanda, Y. ed.: NETTOWN, –a fresh new way to look at emerging business ... –. 1st ed., Tokyo, Diamond 1989, 189pp.

Kazutomo Fukuda
FM TOWNS Systems Dept.
Personal Computer Systems Eng. Div.
FUJITSU LIMITED
Bachelor of Electronics Eng.
Nagoya Institute Technology 1969
Specializing in Development of Application Software

Toru Miyoshi
FM TOWNS Systems Dept.
Personal Computer Systems Eng. Div.
FUJITSU LIMITED
Bachelor of Economics
Specializing in Development of Software for Personal Computer Systems

Tadayuki Tahara
FM TOWNS Systems Dept.
Personal Computer Systems Eng. Div.
FUJITSU LIMITED
Hiroshima Technical High School 1975
Specializing in Development of Application Software

UDC 681.3.064:7.091

Musical Instrument Digital Interface Sequencer Software: EUPHONY

● Joe Mizuno ● Satoru Ono *(Manuscript received March 6, 1990)*

The EUPHONY sequencer software is a music composition/arrangement application which supports the Musical Instrument Digital Interface (MIDI) standard. EUPHONY enables music played on a keyboard to be recorded in real time, edited, and played back. The user can compose and perform orchestral music without a real orchestra. EUPHONY also has graphic editing functions which display performance information, e.g. pitch, volume, and duration, thus increasing editing efficiency. EUPHONY is easier to use than conventional sequencer software. This paper describes EUPHONY and the reasons for its development.

1. Introduction

Electronic music began in 1964 with Robert A. Moog's Moog Synthesizer. Electronics enabled new sounds to be synthesized, and eventually enabled music to be composed and performed using solely electronic instruments.

In early analog synthesizers, sound generation was mostly controlled using analog voltage levels. For example, pitch would rise one octave as voltage was increased by one volt. Performance data could thus be created from control voltages instead of from a keyboard.

The first analog electronic sequencer repeated information contained in registers according to the settings of panel switches. The number of switches ranged from eight to several dozen. Later, large-capacity small-sized digital sequencers were developed. These sequencers were able to digitize, store, and playback music. The music created using these simple systems became commonplace by the second half of the 1970s.

In that period, sequencers were regarded as substitutes for musicians. They were used to play phrases that were too fast, monotonous, or difficult for musicians to play. But since only the on/off and pitch data could be defined, it was difficult to produce expressive music using these sequencers.

Later developments enabled synthesizers to be controlled from a keyboard. This allowed large amounts of data to be transmitted, for example, pitch, duration, and modulation; but the makers all used different standards, so only synthesizers of the same maker could be connected.

The MIDI (Musical Instrument Digital Interface) standard was set up in 1981 by the major electronic instrument companies. With the advent of MIDI[1)-3)] (see Appendix), possibilities for electronic instruments and their control devices expanded rapidly. MIDI systems enabled the production of complex music. Because of reductions in the cost of digital electronics, even non-professional musicians began to use this technology. In addition to hardware sequencers, many software sequencers were being developed for personal computers.

2. Euphony development

2.1 MIDI sequencers

Many sequencers have been developed. Table 1 lists the major sequencers on the market in 1987 when development of EUPHONY was started.

These sequencers were easy to carry and

operate, and became widely used in music production and live performances. However, they still had several drawbacks.

One of these draw backs was that their display screens were too small. The largest screen displayed only two or three 40-character lines. It was therefore difficult to view the music as a whole. No problem arose when entering monotonous sequences of numbers, but the functions and speed of these sequencers were not suited to complicated operations or detailed editing.

Early sequencers also had very little memory. Even if external storage such as cassette tape or disk was used, the internal memories still restricted processing to small amounts of data.

2.2 Advent of software sequencers

A sequencer is simply a dedicated computer; realizing this, people soon began to write programs that gave personal computers sequencing functions equivalent or superior to hardware sequencers. This, of course, required the addition of a MIDI interface to these computer. Table 2 lists the major software sequencers available in 1987.

Table 1. Major hardware sequencers in 1987

Manufacturer	Model	Recording tracks	Capacity (notes, approx.)
YAMAHA	QX3	16	24 000
YAMAHA	QX5	8	20 000
ROLAND	MC-500	6	30 000
KORG	SQD-8	8	8 200
CASIO	SZ-1	4	3 600
KAWAI	Q-30	2	5 000

These products overcame many of the disadvantages of the sequencers mentioned earlier. Their large screens enabled the simultaneous display of large amounts of data, and they had a much greater storage capacity. However, these sequencers were still unsatisfactory in many ways.

In these early sequencers, sounds were represented by notes on a staff, or by numbers indicating pitch, volume, and velocity. However, staff notation is imprecise and allows for considerable latitude of interpretation. The numeric, system, on the other hand, is a string of numbers that is difficult to interpret as music. Users, therefore, soon began to ask for a system that displayed music more accurately but in a way that was easy to read and understand.

Users also wanted to be able to edit data in blocks and restore a program to its original condition. This required a whole new software concept.

EUPHONY was developed with the above points in mind.

3. Features of EUPHONY

3.1 MIDI interface card

The falling prices of MIDI devices enabled large systems to be constructed at relatively low cost. However, large systems have their own particular problems.

One of these problems is the time lag due to the transmission of large amounts of data on a single MIDI cable. This lag is called the MIDI delay. MIDI can control up to 16-MIDI devices independently using a single cable. But, to make MIDI devices generate notes simultaneously,

Table 2. Major software sequencers in 1987

Manufacturer	Product	Computer	Recording tracks	Display format
COME ON MUSIC	PCX-PC98	NEC PC-98 01 series	10	Numbers and a simple staff
ROLAND	RECLIET	NEC PC-98 01 series	8	Numbers
DYNAWARE	PRELUDE	NEC PC-98 01 series	16	Staff
MARK OF THE UNICORN	PERFORMER	APPLE MACINTOSH series	200	Numbers

the data must arrive at all devices at the same time. If the devices are connected as a serial chain, the note generated by the last device on the chain can lag behind the one generated by the first device by several milliseconds. This MIDI delay is quite noticeable, and will spoil a performance.

Also, musicians naturally wanted to use the growing number of sound sources, and the 16 instrument limit soon became too restrictive.

EUPHONY solves the above problems. We designed the MIDI interface card for the FM R series so that parallel data transmission can control multiple MIDI OUT ports, and a total of four cards can be used together. This eliminates the MIDI delay and enables the connection of up to 128 MIDI devices, thus significantly increasing the number of parts that can be played.

3.2 Realtime recording

Realtime MIDI recording is a normal function of software sequencers. The MIDI transfers performance data by realtime transmission of data such as key down, key up, modulation wheel movement, and patch change. This data also includes the velocity and pressure with which the keys are pressed. The time data and transmitted data are combined and then stored in the memory. This enables realtime recording. Like a tape recorder, the performance can be reproduced by outputting the stored data to a sequencer in real time. Both a sequencer and a tape recorder can record and play back performances, but only a MIDI sequencer allows extensive editing. A MIDI sequencer can easily remove or change the data in its storage but a tape recorder cannot. Also, editing or dubbing by sequencer does not affect the sound quality, and a sequencer enables even poor players to produce advanced music.

EUPHONY enables easy editing of music.

3.3 Graphic editing

3.3.1 Musical notation that is easy to read

Conventional sequencers represent music as numbers. Such a system makes melody and chord progressions difficult to grasp and hampers the flow of creative thought.

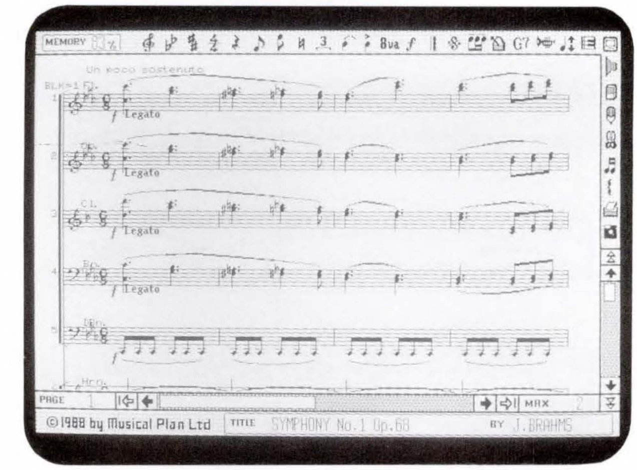


Fig. 1—Example display of SCORE.

Also, musicians without a background in computers usually find such sequencers very difficult to operate.

EUPHONY, however, displays data graphically so that phrases and melodies can be quickly recognized.

3.3.2 Representation using scores

Music is most often represented using traditional staff notation. It is the most orthodox and effective way of representing music. However, music editing software has problems in setting the spaces between notes and between measures. To overcome this problem, EUPHONY uses a screen called SCORE (see Fig. 1).

The user can place notes in the staff and instruct EUPHONY to perform them. We have developed an interpreter that converts music in staff notation into performance data. This produces scores that are easy to read and perform. EUPHONY can handle scores of up to 24 staffs, which is sufficient for even orchestral music. These scores can be printed (see Fig. 2), so desk-top music publishing of the composed music is also possible.

3.3.3 Expression on roll paper

Although a score is the usual way of representing music on paper, a score does not always correspond to performance data. On a score, the strength and length of a note are indicated by symbols such as *f* (forte), *p* (piano), stac (staccato), and *leg* (legato). But because these symbols do not represent absolute quan-

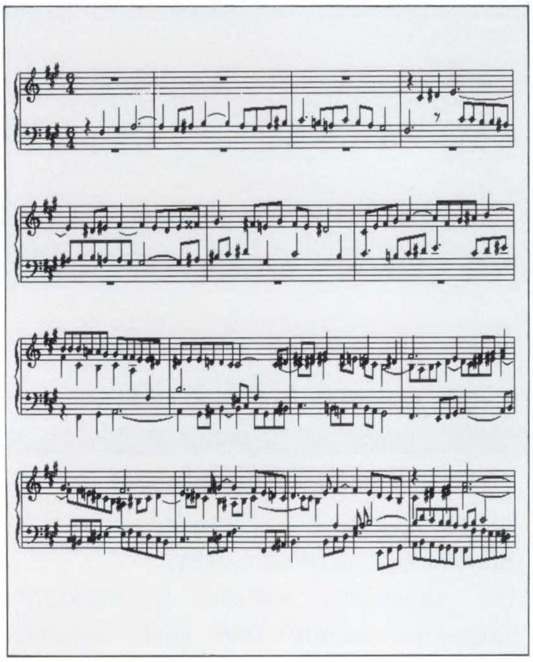


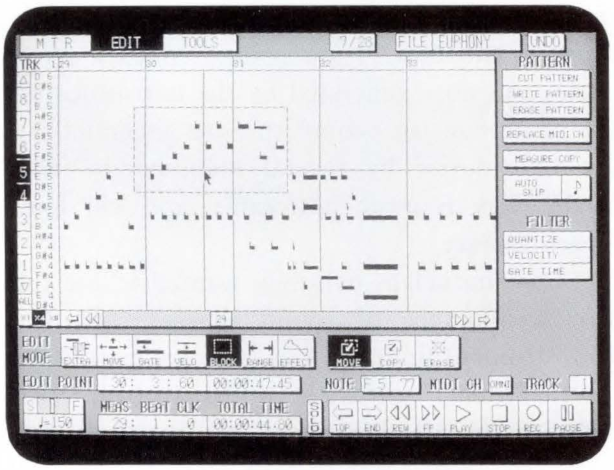
Fig. 2—Example print of the score.

tities, a performance can depend on the player and or the phrases played before and after the note. A score is therefore unsuitable if exactly reproducible performances are required. Such reproducibility calls for a more accurate means of expressing performance data.

MIDI classifies music elements into pitch, volume velocity, modulation wheel, sustain pedal, voice, and the most important element, timing. On a score, the horizontal axis represents time, but the horizontal distance between notes is not proportional to time. Volume is defined but only approximately. Although the pitch of a note is exactly defined the pitch is not always proportional to the note's vertical position because it is usually modified by sharps and flats.

We have devised a way to accurately represent the nuances of music. The new method, shown in Fig. 3, resembles the rolls used in player-piano mechanisms.

On a EUPHONY score, the horizontal and vertical axes represent time and pitch respectively. The duration of a note is represented by its length. The musical scale is



EDIT POINT 30: 3 : 60 00:00:47.45 NOTE F 5 77 MIDI CH OMNI TRACK 1

♩=150 MEAS BEAT CLK 29: 1 : 0 TOTAL TIME 00:00:44.80

Fig. 3—Example display of EDIT.

arranged linearly on the vertical axis so the relative pitch of a sound can be easily determined. The volume of each note is represented by the thickness of its line. Although there are 127 possible levels, it would by impossible to distinguish all of them; therefore, we restricted the number of volume levels on the score to eight. These volume levels are displayed during editing only.

Reading pitch and duration by counting pixels on the CRT would be very tedious. Therefore, the time position can be determined using a cursor, and the position and pitch are indicated numerically at the bottom of the screen. Music is therefore expressed precisely using both numbers and graphics. Data can be edited by scrolling the score vertically or horizontally using the scroll bar and by changing the display magnification. Data can be displayed for each track and for each MIDI channel on each track.

3.3.4 Note editing

To change the volume of a note, a mouse is used in velocity mode to select the note and then increase or reduce its thickness and hence its volume. The velocity is displayed numerically to the side of the note. To change the timing or pitch of a note, its position is changed using the mouse in move mode. Similar operations are made to change the duration of a note and to delete a note or add a new one. Because the length and strength of a sound can be specified either exactly or roughly, human inconsistency can be easily simulated.

3.3.5 Block editing

Visualization of performance data enables phrase editing as well as note editing. The elements of a note are its pitch, length, and volume. Notes form phrases, phrases form passages, and passages form music. With earlier software sequencers, note editing but not phrase editing was possible. Therefore, a phrase had to be edited by editing each component note. EUPHONY can edit an entire rectangular area corresponding to the time and pitch of a note. It is often better to specify a phrase using a free curve, but in most cases a rectangular area is sufficient. Several block editing functions are provided to edit each note in a specified phrase.

Generally, the timing of a poor player is inconsistent. In EUPHONY, timing can be corrected by adjusting the timing of sound generation using the quantitize function. This function adjusts the timing of notes using a grid. This was sometimes done during recording, so satisfactory and unsatisfactory parts could be edited separately.

EUPHONY can also separately edit parts.

The move function advances or delays the performance of a phrase contained within a rectangular area specified by the user.

EUPHONY can also copy phrases, perform forte or staccato phrases, and change notes.

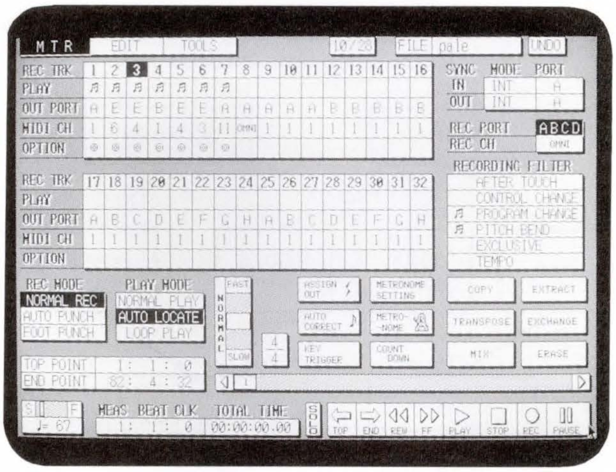
3.3.6 Effect editing

Effect data such as patch, pitch bend, control changes, and tempo is displayed graphically on a single time axis so that their effect on the music can be quickly appreciated. Batch editing of effect data is also possible.

3.4 User interface

3.4.1 User-oriented design

An easy to use power supply is an essential requirement for an electronic music system. The most important requirements when creating an application are a unified procedure for calling functions, categorized arrangement, easy-to-read screens, fast processing, and an overall flow that enhances a musicians creativity (see Fig. 4).



FILE pale

REC TRK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
PLAY	♬	♬	♬	♬	♬	♬	♬									
OUT PORT	A	E	E	B	E	E	A	A	A	A	A	B	B	B	B	B
MIDI CH	1	6	4	1	4	3	11	OMNI	1	1	1	1	1	1	1	1
OPTION	◎	◎	◎	◎	◎	◎	◎									

REC TRK	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
PLAY																
OUT PORT	A	B	C	D	E	F	G	H	A	B	C	D	E	F	G	H
MIDI CH	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
OPTION																

SYNC	MODE	PORT
IN	INT	A
OUT	INT	A

REC PORT ABCD
REC CH OMNI

RECORDING FILTER
AFTER TOUCH
CONTROL CHANGE
♬ PROGRAM CHANGE
♬ PITCH BEND
EXCLUSIVE
TEMPO

REC MODE: NORMAL REC / AUTO PUNCH / FOOT PUNCH
PLAY MODE: NORMAL PLAY / AUTO LOCATE / LOOP PLAY

4/4

TOP POINT 1 : 1 : 0
END POINT 32 : 4 : 32

♩= 67
MEAS BEAT CLK 1 : 1 : 0
TOTAL TIME 00:00:00.00

Fig. 4 – Example display of Multi Track Recorder.

3.4.2 Undo function

This function is indispensable when using this type of software. If mistakes are not correctable in software, they will have to be made by hand on the printed score. Often, the user cannot completely appreciate the effect of a change until it is performed. By using the undo function, the music can be restored to its original state if the results are unsatisfactory. This editing can be done during a performance because EUPHONY retains both the before- and after-editing data. This function is of great benefit to composers of music, and in part, has been made possible because of dramatic reductions in the cost of memory devices.

3.4.3 Screen drawing speed

Although the time required for each operation is not specified, it is the key factor that governs the performance of software. A fast response makes software easy to use but is difficult to achieve. The MTR program of EUPHONY reduces the time taken to draw a new screen. MTR achieves this by writing screen data in another color in advance and then switching the screen very quickly by switching the pallete. Each subwindow consists of lines and characters only, so figures can also be drawn. The score program stores screen data in main storage, thus enabling high-speed scrolling.

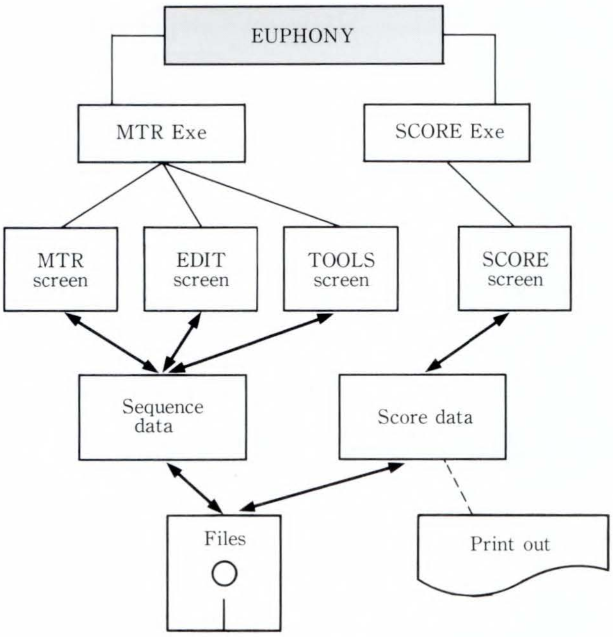


Fig. 5–Data flow.

3.5 Data sharing

EUPHONY consists of two programs, MTR and SCORE (see Fig. 5). This division is necessary because the 1 Mbyte-storage capacity of MS-DOS™ is insufficient for all EUPHONY functions.

MTR processes performance data, and uses three screens, MTR, EDIT, and TOOLS. TOOLS is used to control MIDI devices. The SCORE program is used to compose, copy, and print the score.

It has a data conversion function which shares data with MTR. MTR can add nuances to the basic music produced using SCORE. MTR can also produce a score from a live performance. As mentioned earlier, performance data does not always correspond to the score; it is therefore difficult to convert performance data into a SCORE. We are currently studying ways to improve MTRs performance in this area.

4. Conclusion

EUPHONY is easy to use. Previously, the entire computing power of a personal computer was required to achieve the basic functions of EUPHONY.

In the 1990s, 32-bit personal computers will be common. To improve EUPHONY, the base machine will be changed from an MS-DOS™ (16-bit) FM R series to the 386 OS (32-bit) FM TOWNS. Also, EUPHONY will include a composition support system based on music theory, and will support SMPTE recording, editing, and mastering.

5. Appendix

The MDI signal is sent through a DIN cable as serial digital data at 31.25 kbauds. It takes about one millisecond and two or three bytes to send one note-on signal.

The MIDI signal includes the following messages:

1) Channel message

This is a set of instrument performance information. Channels 1 to 16 are handled independently.

i) Voice message

a) Note on/off
On/off state of a note

b) Pitch bend
Discrete changes in pitch

c) Program change
Voice specification

d) Control code control
Effects such as volume and position

ii) Mode message

a) Omni
All 16 channels are received

b) Mono
One note at a time only

c) Poly
Chords can be played

2) System messages

This is a set of supplemental performance information.

i) Common message

a) Song position pointer
Position during a sequence performance

b) Song select
Song specification

ii) Realtime message

a) Clock
Clock for synchronous performance by multiple devices

iii) System exclusive message

a) Specified by the manufacturer (voice

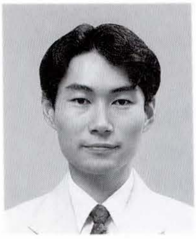
tone data and sequencer data)

References

1) Japan MIDI Standard Committee: MIDI 1.0 Standard. Document Ver. 4.1, (Japanese Ver.), 1989, 88p.

2) Furuyama, S.: MIDI and Synthesizers. (in Japanese), *bit*, **19**, 10, in Suppl. Computers and Music, 1987, pp. 203-216.

3) — : *Latest sequencers.* (m. mag., in Japanese), 1987-12, Ritto Music.



Joe Mizuno

Personal Computer Develop. Dept.
FUJITSU PERSONAL COMPUTER
SYSTEMS LIMITED
Mitaka High School 1984
Specializing in Software Design of FM Series Personal Computers

Satoru Ono

Personal Computer Develop. Dept.
FUJITSU PERSONAL COMPUTER
SYSTEMS LIMITED
Bachelor of Education
Ni-igata University 1987
Master of Education
Tokyo Gakugei University 1989
Specializing in Musical Composition

UDC 681.327.68

CD-ROM Premastering System Using CD-Write Once

● Yasushi Ashikaga ● Kazuhito Fukuoka ● Masafumi Naitoh

(Manuscript received May 9, 1990)

The CD-ROM has become an important computer storage medium. However, recording data onto a CD-ROM requires special equipment, thus making it difficult to develop and deliver applications on CD. The Fujitsu Premastering System has been developed to solve this problem. This paper describes the configuration and functions of the Premastering System. It also describes the CD-ROM standards, the procedure used to generate a CD-ROM application, and the CD-Write Once recorder developed for the Premastering System.

1. Introduction

In 1980, Sony and Philips proposed a standard for Compact Disc-Digital Audio (CD-DA). This was followed in 1983 by a standard for the computer CD-ROM. Recently, standards have been proposed for Compact Disc-Interactive (CD-I), Digital Video Interactive (DVI), and CD-ROM Extended Architecture (CD-ROM XA), all of which are based on the CD-ROM.

By using these standards, a single compact disc can contain enough audio, image, and text data to fill over 500 floppy disks. It is because of this large storage capacity that the CD-ROM plays such an important role in the development and use of multimedia applications.

Producing a CD-ROM multimedia application involves the use of an authoring system and a premastering system. The authoring system assists in the application development processes, i.e. data capturing, data editing, and program debugging. The premastering system converts the data generated by the authoring system into CD-ROM format data and enables debugging of the CD-ROM.

However, because the CD-ROM is a read-only medium, it is difficult for developers to debug applications. Further more, large and complex equipment is required to record on CD-ROMs.

The Fujitsu Premastering System solves these problems by allowing developers to use a small and convenient personal computer based system to develop and record CD-ROM applications.

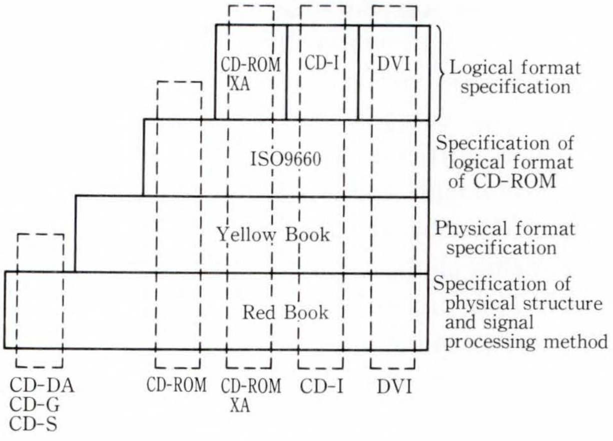
2. Standards for CD-ROM[1)]

There are three current types of digital audio CDs: the 12-cm Digital Audio (CD-DA), 12-cm Graphic (CD-G), and 8-cm Single (CD-S). Most of the computer CD-ROMs in use today (i.e. CD-ROM, CD-I, DVI, and CD-ROM XA) use a superset of the CD-DA format. For CD-I, a standard for image and audio data has been defined in addition to a standard for the controlling OS. For DVI, standards for animation and audio data have been defined. CD-ROM XA is based on a subset of the CD-I standards, and has some unique features. Figure 1 illustrates these main CD standards.

From Fig. 1, it can be seen that the Red Book defines the lowest order standard and is used for CD-DA. This standard specifies the physical structures of the disc such as its size.

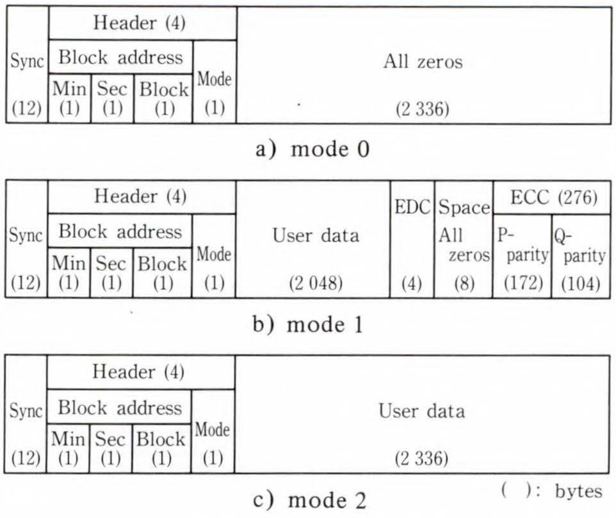
It also specifies the methods of processing digital signals, for example, the modulation method and the type of subcode signal. The CD-ROM is based on this standard. Refer to Table 1 for the main specifications in the Red Book.

The Yellow Book is the next standard above


CD-ROM XA
CD-I
DVI
Logical format specification
ISO9660
Specification of logical format of CD-ROM
Yellow Book
Physical format specification
Red Book
Specification of physical structure and signal processing method
CD-DA
CD-G
CD-S
CD-ROM
CD-ROM XA
CD-I
DVI

CD-DA : Compact Disc-Digital Audio
CD-I : Compact Disc-Interactive
DVI : Digital Video Interactive
CD-ROM XA : CD-Read Only Memory Extended Architecture

Fig. 1—Main CD standards.



EDC : Error Detection Code   ECC : Error Corection Code

Fig. 2—Data structure specified by Yellow Book.

Table 1. Main parameters specified in Red Book

Items	Specifications
Playing time	About 60 min (single side, two channels)
Scanning velocity	1.2-1.4 m/s
Sense of rotation	Counter clockwise seen from reading side
Track pitch	1.6 μm
Diameter of the disc	120 mm
Thickness of the disc	1.2 mm
Diameter of center hole	15 mm
Starting diameter of program area	50 mm
Number of channels	2 and/or 4
Sampling frequency	44.1 kHz
Quantization	16 bits linear/channel (2's complement)
Channel bit rate	4.321 8 Mbit/s
Pre-emphasis	No or 50/15 μs

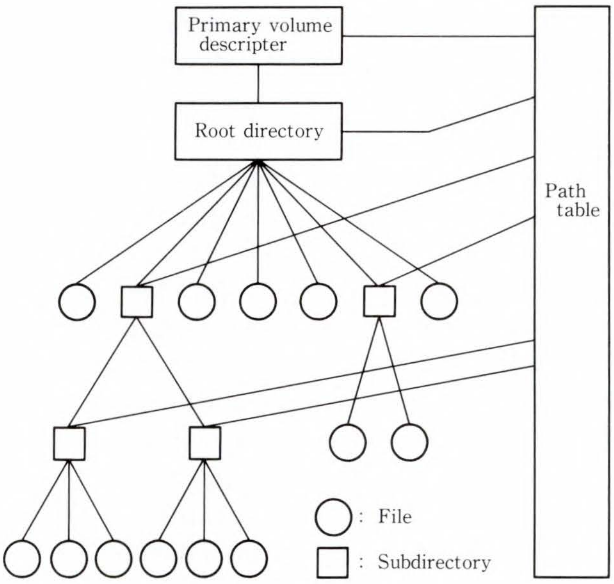
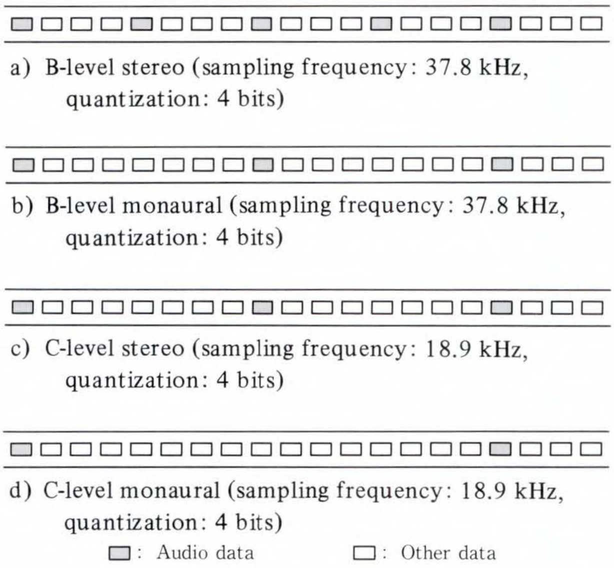


Fig. 3—CD-ROM logical format specified by ISO9660.

the Red Book and specifies the sector structure of CD-ROMs. In the Yellow Book, the 96 frames containing music data as specified in the Red Book are defined as one block, which is the unit of CD-ROM data. Each block consists of the following: a SYNC section, a header section the containing block address and mode difinition, a user section containing the user data, and a section containing additional data. There are three different modes that can be defined in the header block (see Fig. 2). The CD-ROM addresses are in minute, second, and block units. There are seventy-five blocks in each second and sixty seconds in each minute.

Figure 2 shows that in mode 0 the user data and additional data sections are set to zero. This mode is used to place a delimiter before and after a storage area. In mode 1, 2 048 bytes of user data can be recorded in one block. The user data is followed by the additional data section which contains the error detection codes (EDC) and error correction codes (ECC). Mode 1 is generally used for standard computer data and



a) B-level stereo (sampling frequency: 37.8 kHz, quantization: 4 bits)

b) B-level monaural (sampling frequency: 37.8 kHz, quantization: 4 bits)

c) C-level stereo (sampling frequency: 18.9 kHz, quantization: 4 bits)

d) C-level monaural (sampling frequency: 18.9 kHz, quantization: 4 bits)

▨: Audio data □: Other data

Fig. 4—Block arrangement in CD-ROM XA.

program files. Mode 2, which contains two different types (forms 1 and 2), allows 2 336 bytes of user data to be written and is used by CD-ROM XA and CD-I.

The Yellow Book standard is followed by the ISO9660 standard which specifies the logical format of the CD-ROM. In the ISO9660 standard (shown in Fig. 3), the CD-ROM files are managed by hierarchical directories. The highest order directory is defined as the root directory and can be retrieved by reading the Primary volume descriptor. ISO9660 also defines a path table, allowing files with a complicated directory structure to be retrieved quickly.

CD-ROM XA also uses the ISO9660 format and extends it to support the audio/visual environment. This extension involves the use of Adaptive Differential PCM (ADPCM) instead of the linear PCM used by CD-DA to allow other data to be read during audio playback. By using ADPCM, interleaving the data in each block, and utilizing the empty blocks to store image data (see Fig. 4), it is possible to read and display images from the CD-ROM while reading a narration. This process can also be used for animation, but a method of doing this has not yet been specified. Note that in CD-ROM XA, audio data is written in mode 2 form 2 and non-audio data is written in mode 2 form 1 (see section 4.3).

3. CD-ROM application generation procedure

Before the development of the CD-ROM, the generation of multimedia applications combining graphic data, images, and audio data was extremely difficult. The CD-ROM has enabled such multimedia applications to be created. This chapter describes the procedure used to generate multimedia applications to be recorded on CD-ROM. This procedure is divided into the four steps shown in Fig. 5.

In step one, Planning and scenario creation, a title is selected and a story is created. In multimedia applications, the use of drawings, sounds, and characters is determined at the step two.

Step two is divided into four stages: Program development, Scenario editing, Data editing, and Data capturing. Program development involves the use of compilers, linkers, libraries, and debuggers to create the multimedia program. The Scenario editing, data editing, and data capturing stages are collectively referred to as authoring tools. These tools are used to manipulate the multimedia data, and play an important role in the development of multimedia applications. Step two produces the data necessary for application generation. This data is then edited, captured, and linked to a program. This process corresponds to conventional application development, and to the generation of the program and data to be passed to the next step.

Step three, Premastering, involves the following tasks:

1) Data conversion

Conversion of file formats into the formats specified by ISO9660 and the Yellow Book.

2) CD-WO (CD-Write Once) generation

3) CD-ROM simulation

Debugging by CD-ROM simulation.

The fourth and final step, Mastering, mass produces completed CDs.

The following chapter discusses premastering.

4. Fujitsu Premastering System

Premastering prepares the data to be used in Mastering. This chapter describes the CD-ROM and CD-ROM XA premastering procedure used in the Fujitsu Premastering System.

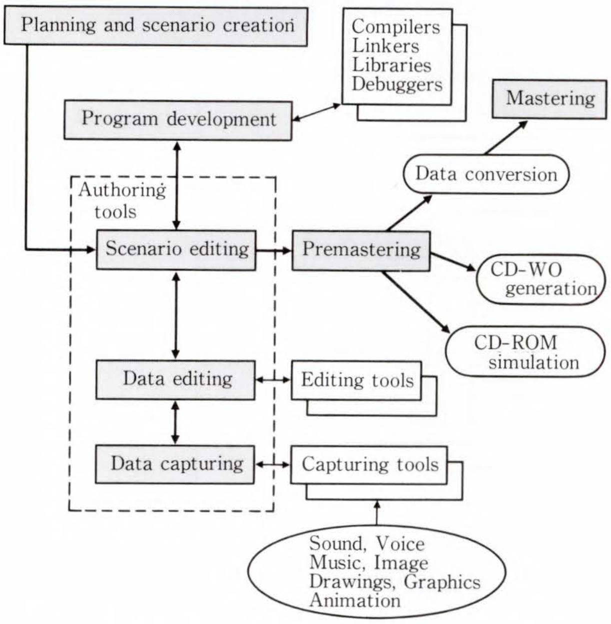


Fig. 5—CD-ROM application generation procedure.

The Fujitsu Premastering System enables easy, low-cost CD-ROM generation using a personal computer. It also uses several control files to automate most of the premastering processes.

Figure 6 shows an example system configuration. The Fujitsu Premastering System consists of a Fujitsu FM TOWNS or FM R series personal computer, a CD-WO recorder and CD-WO recorder interface card, a hard disk for data conversion, and premastering software for MS-DOS™. In addition to the basic system, a hard disk for source data, a magnetic optical disk unit (MO), and a magnetic tape unit (MT) can be added.

4.1 Premastering processes

The Fujitsu Premastering System performs the premastering process in the following four steps (see Fig. 7):

1) ISO9660 conversion

Here the data is converted to the logical format specified by the ISO9660 standard, allowing the data to be quickly accessed by a computer.

2) Interleaved file generation

This step generates an interleaved file conforming to the CD-ROM XA standard. This involves writing image, graphic, and character data between blocks of audio data compressed using ADPCM (see Fig. 4). This enables the non-audio data to be read while the audio is being played.

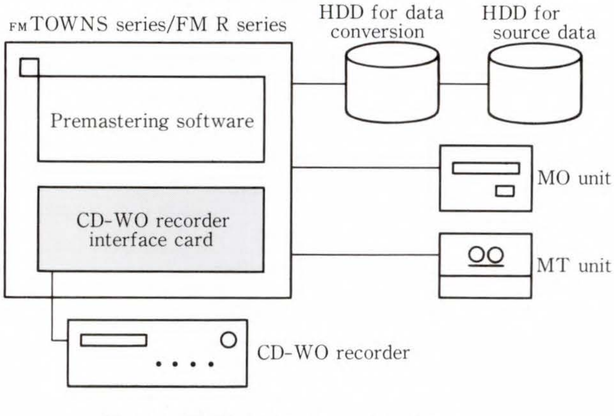
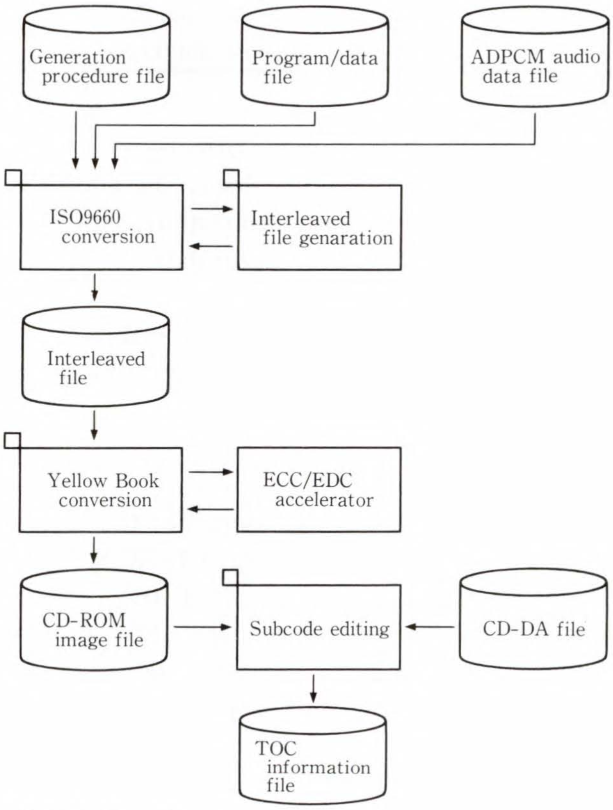


Fig. 6—Fujitsu Premastering System.



TOC: Table of Contents

Fig. 7—Outline of CD-ROM generation process.

3) Yellow Book conversion

Because the CD-ROM contains computer data as well as audio data, it is necessary to perform for error correction. This step adds

Table 2. Volume information supported by Fujitsu Premastering System

No.	Volume information
1	System identifier
2	Volume identifier
3	Volume set identifier
4	Publisher identifier
5	Data prepare identifier
6	Application identifier
7	Copyright file identifier
8	Abstract file identifier
9	Bibliographic file identifier
10	Volume creation date and time
11	Volume modification date and time
12	Volume expiration date and time
13	Volume effective date and time

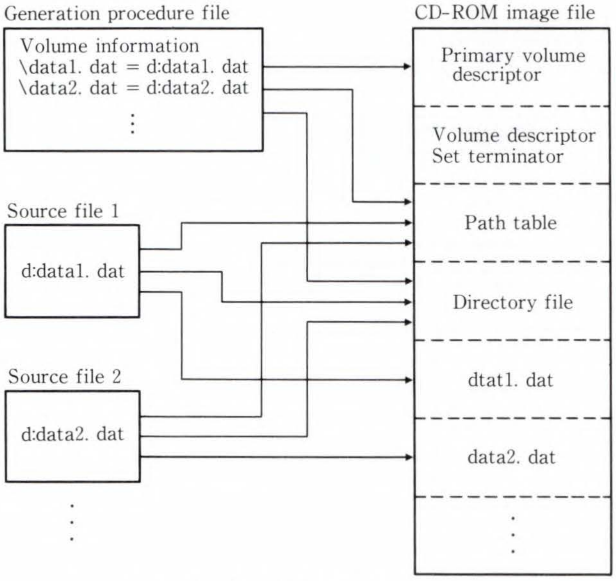

Generation procedure file
CD-ROM image file
Volume information
\datal. dat = d:datal. dat
\data2. dat = d:data2. dat
Primary volume descriptor
Volume descriptor Set terminator
Source file 1
d:data1. dat
Path table
Directory file
Source file 2
d:data2. dat
dtat1. dat
data2. dat

Fig. 8—ISO9660 conversion.

the error detection and correction data as specified by the Yellow Book. In addition, this step adds the synchronization data that indicates the beginning of sector data.

4) Subcode editing

This step adds the address information, track numbers, and index numbers and is the final step.

4.2 ISO9660 conversion

The ISO9660 standard specifies the logical data structure for handling CD-ROM data. This standard is independent of the type of computer or operating system. The Fujitsu Premastering System supports the volume information specified in the ISO9660 standard (see Table 2).

Since the CD-ROM is a read only medium with a hierarchical file structure, the ISO9660 standard defines a double file management system. Therefore, files are managed by both a directory file and a path table. The Fujitsu Premastering System prepares a generation procedue file which correlates the source files with the path and file names in the CD-ROM. This allows the directory file and path table to be generated automatically. The Fujitsu Premastering System supports file names and attributes that are compatible with the ISO9660 and MS-DOS^{TM} file management specifications.

Figure 8 outlines the ISO9660 conversion process.

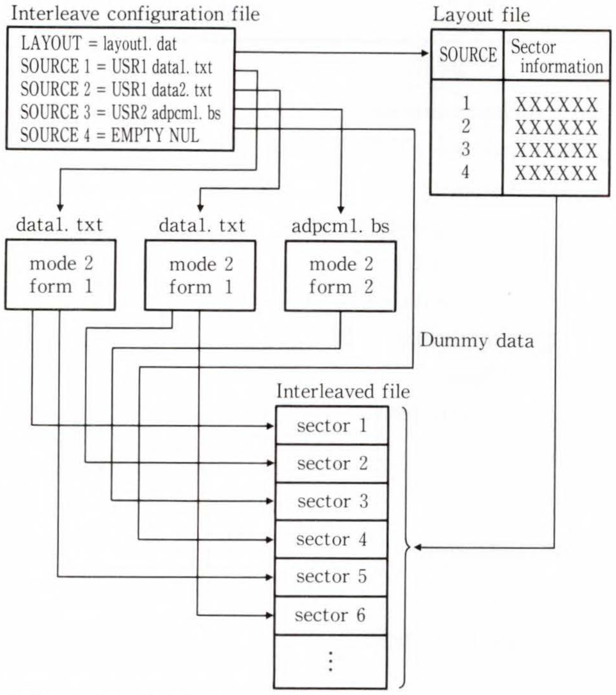
4.3 Interleaved file generation

In CD-ROM XA, it is possible to mix audio and non-audio data. Because of this feature, it is necessary to interleave the audio and non-audio data. To perform the necessary interleaving manually is both difficult and time consuming.

When using the Fujitsu Premastering System, an interleaved file is automatically generated by using an interleave configuration file and a layout file. The interleave configuration file contains information about the source files such as the path names and file types. The layout file describes how the different source files are to be arranged and written.

An example of the interleaving process is shown in Fig. 9. The figure shows a layout configuration file containing five different parameters: LAYOUT, SOURCE 1, SOURCE 2, SOURCE 3, and SOURCE 4. In this example, LAYOUT is defined as layoutl.dat, SOURCE 1 as USR1 datal.txt, and SOURCE 4 as EMPTY NUL.

The layoutl.dat layout file defines the source number, channel number, and sector information for each file sector. In this example, the layoutl.dat file indicates that sector 1 of datal.txt should be copied into the first sector



USR1 : A 2 048 bytes mode 2 form 1 sector
USR2 : A 2 328 bytes mode 2 form 2 sector
EMPTY : Sector for storing dummy data

Fig. 9—Interleaved file generation.

of the interleaved file, sector 1 of data2.txt into sector 2, sector 1 of adpcml.bs into sector 3, and dummy data into sector 4.

File datal.txt and data2.txt contain the user text data that is read during audio playback. File adpcml.bs contains ADPCM-B-level stereo audio data. Note that to satisfy the CD-ROM XA standards for B-level stereo, it is necessary to write one sector of audio data for every four sectors in the interleaved file.

4.4 Yellow Book conversion

The Yellow Book Standard specifies the physical sector format of CD-ROMs and supports three modes (0, 1, and 2) for different types of data to be stored.

Writing data in the standard CD-ROM format is quite easy because only mode 1 is used. However, writing in the CD-ROM XA standard is more difficult because modes 1, mode 2 form 1, and mode 2 form 2 are used. This conversion process has been automated in the Fujitsu Premastering System by using the subheader information written during ISO9660 conversion. The system checks the subheader data for each sector and converts the data to the proper mode and type. The error detection and correction codes are generated by special hardware and are written during this process.

4.5 Subcode editing

The subcodes consists of eight channels (P through W) and contain a total of 7 350 bits for each second of digital audio. Channel P is used to indicate the presence of audio tracks. Channel Q contains the audio data information such as address and track numbers. Channel R through W contain the data to be used for CD-G.

When generating subcodes, the software on the personal computer only needs to prepare the general information necessary for subcode generation. The premastering system will automatically generate all the necessary subcode parameters, for example track numbers, absolute audio start time, audio data size, and track type (CD-ROM or CD-DA) according to the Yellow Book standards.

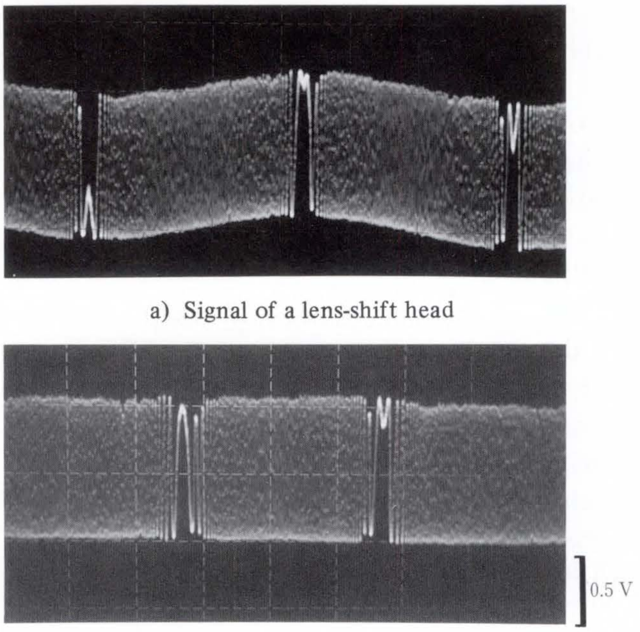
5. CD-WO recorder

This device writes to Compact Disc-Write Once (CD-WO) discs and is compatible with standard CDs. It was developed for use with the Fujitsu Premastering System.

5.1 Optical system

The optical head for the CD-WO recorder uses the head tracking method. In this method, the entire optical system is moved to the track data. Tracking errors are detected using a push-pull method, resulting in the elimination of track sensor offset caused by movement of the objective lens.

Generally, the focusing spot for the recording head of an optical disc drive must have a higher energy level than the spot for the head of an ordinary CD player. Therefore, the one-beam push-pull method based on the lens shift method is used. In this method, only the objective lens is moved resulting in a reduced access time. The disadvantage of this approach



a) Signal of a lens-shift head

b) Signal of the new head developed for this system

Fig. 10—Tracking error signals.

is that the sensor itself is offset by the movement of the objective lens[2]. There are various techniques to compensate for this, such as the use of a lens position sensor. But these techniques complicate the head and servo circuits and their adjustment[3),4)].

The optical system used in the Fujitsu Premastering System employs a tracking method which drives the entire system instead of just the objective lens.

This method was adopted for the following reasons:

1) By using a basic optical system without auxiliary devices, tracking error signals with no error offset caused by lens movement can be obtained.
2) The rotation speed is low (200-500 rpm), so the servo band for the system is lower than the servo band for a high rotation speed device.
3) For computer optical magnetic disks, high speed recording and play back are repeated. This is not the case for a CD-ROM.
4) The optical components and head are lighter.

Figure 10 shows the tracking error signals for a head using the lens shift method and for a head using the new method. Both heads are excited by approximately 150 μm in the radial direction, but the head of the new method has no track sensor offset. Table 3 lists the basic optical head specifications. Figures 11 and 12 show the optical system configuration and the optical head. Figure 13 shows the recording signal for That's CD-R, a CD made by START Lab. Inc., Tokyo, using this head.

Table 3. Optical head specifications

Items	Specifications
Optical characteristics	
Objective lens (λ = 780 mm)	
Focal point (*f*)	3.8 mm
Numerical aperature (NA)	0.5
Light source (LD)	
Oscillàtion wavelength (λ)	780 mm
Maximum output (CW)	35 mW
Emitted light	
Emission efficiency	At least 35 %
Beam spot	Not more than 1 μm
Focus	
Error detection	Non-point aberration method
Actuator	Leaf spring
Resonant frequency	70 Hz
Tracking	
Error detection	Push-pull method
Actuator	Axis rotation method
Resonant frequency	15 Hz

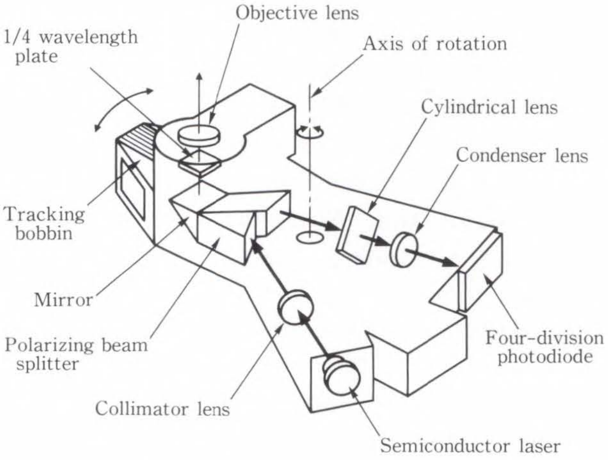


Fig. 11—Head optical system.

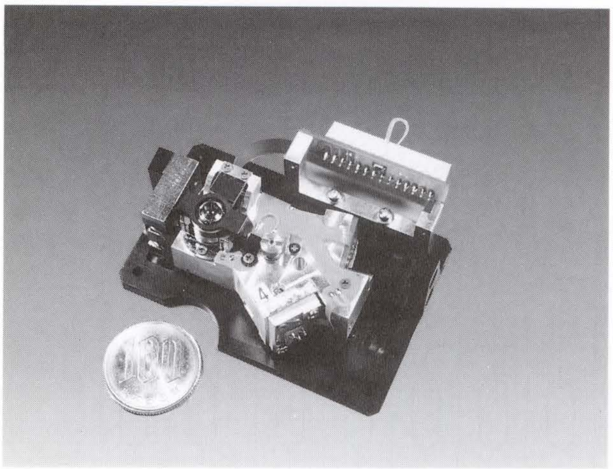


Fig. 12—Optical head.

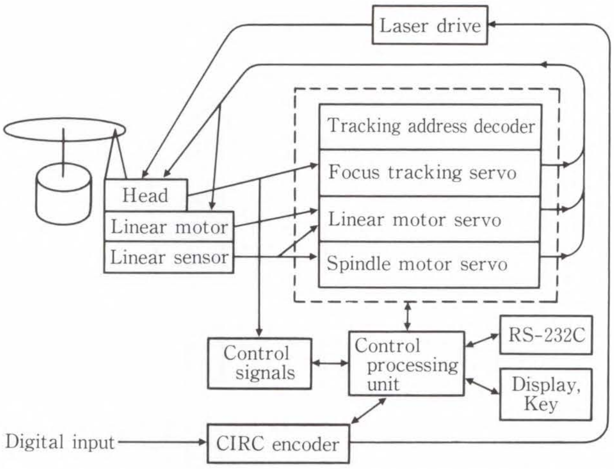


Fig. 14—Signal processing system configuration.

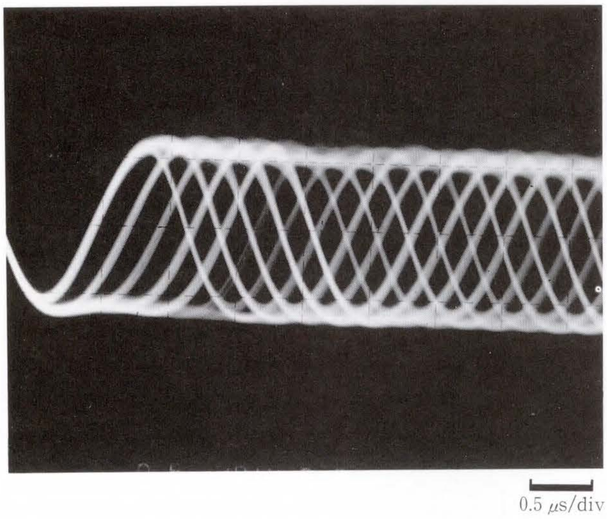


Fig. 13—Playback signal.

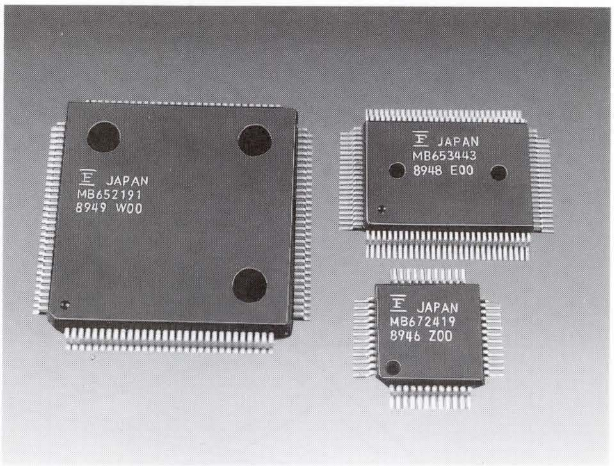


Fig. 15—CIRC encoder chips[6].

5.2 Signal processing

The configuration of the CD-WO writer is shown in Fig. 14.

Following the Red Book specifications, the data is first interleaved so that eight bits form one symbol. Then, Cross-Interleaved Reed-Solomon (CIRC) error correction codes C_1 and C_2 are added to the data[1),5)]. Figure 15 shows the CIRC encoder chips[6], while Fig. 16 shows the CIRC encoder configuration. These 8 bits (1 symbol) are then converted to the 14-bit CD format using Eight-Fourteen Modulation (EFM). These values are then further converted to obtain the Non-Return Zero Inverse (NRZI) signal for the laser drive circuit.

On an ordinary CD, subcode data indicating time and control information are recorded for head position control. However, for unrecorded discs, head position control is difficult because no signal has been recorded. Therefore, the discs used by the Fujitsu Premastering System have a physically wobbled pregroove. (The amplitude of this wobble is about 0.03 μm.) The pregroove is cut according to a signal obtained by modulating the time codes. So, even if a CD is unrecorded, the device obtains the track data of the unrecorded portions by demodulating the pregroove signal that has been filtered from the track error signal. This data can then be used for system and spindle rotation control during recording. To enable head control when there is no signal, a linear sensor indicating the distance of the

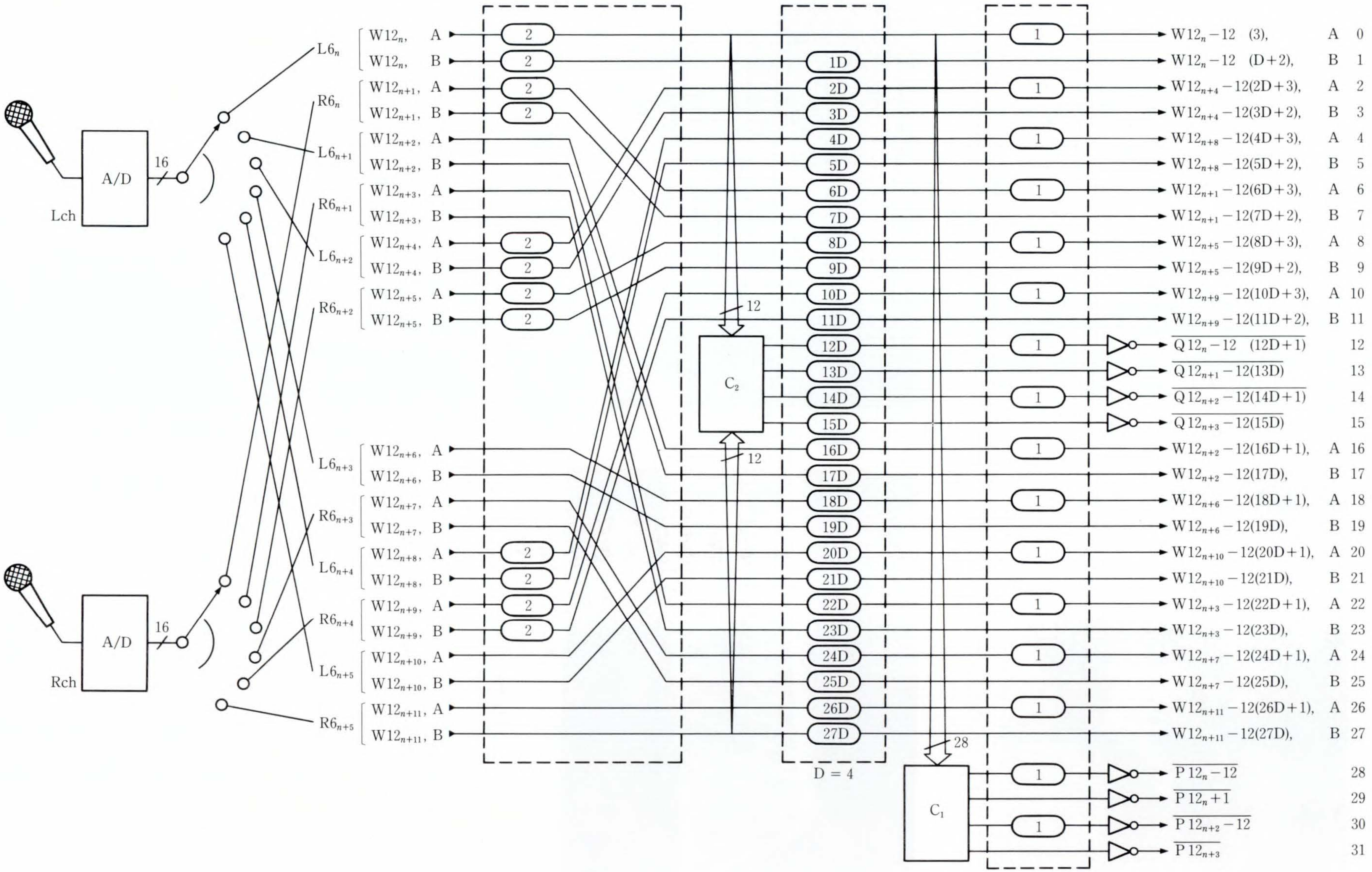

Lch
A/D
16
Rch
A/D
16
$L6_n$
$R6_n$
$L6_{n+1}$
$R6_{n+1}$
$L6_{n+2}$
$R6_{n+2}$
$L6_{n+3}$
$R6_{n+3}$
$L6_{n+4}$
$R6_{n+4}$
$L6_{n+5}$
$R6_{n+5}$
$W12_n$, A
$W12_n$, B
$W12_{n+1}$, A
$W12_{n+1}$, B
$W12_{n+2}$, A
$W12_{n+2}$, B
$W12_{n+3}$, A
$W12_{n+3}$, B
$W12_{n+4}$, A
$W12_{n+4}$, B
$W12_{n+5}$, A
$W12_{n+5}$, B
$W12_{n+6}$, A
$W12_{n+6}$, B
$W12_{n+7}$, A
$W12_{n+7}$, B
$W12_{n+8}$, A
$W12_{n+8}$, B
$W12_{n+9}$, A
$W12_{n+9}$, B
$W12_{n+10}$, A
$W12_{n+10}$, B
$W12_{n+11}$, A
$W12_{n+11}$, B
2
C₂
12
12
1D
2D
3D
4D
5D
6D
7D
8D
9D
10D
11D
12D
13D
14D
15D
16D
17D
18D
19D
20D
21D
22D
23D
24D
25D
26D
27D
D = 4
28
C₁
1
$W12_n-12$ (3), A 0
$W12_n-12$ (D+2), B 1
$W12_{n+4}-12(2D+3)$, A 2
$W12_{n+4}-12(3D+2)$, B 3
$W12_{n+8}-12(4D+3)$, A 4
$W12_{n+8}-12(5D+2)$, B 5
$W12_{n+1}-12(6D+3)$, A 6
$W12_{n+1}-12(7D+2)$, B 7
$W12_{n+5}-12(8D+3)$, A 8
$W12_{n+5}-12(9D+2)$, B 9
$W12_{n+9}-12(10D+3)$, A 10
$W12_{n+9}-12(11D+2)$, B 11
$\overline{Q12_n-12\ (12D+1)}$ 12
$\overline{Q12_{n+1}-12(13D)}$ 13
$\overline{Q12_{n+2}-12(14D+1)}$ 14
$\overline{Q12_{n+3}-12(15D)}$ 15
$W12_{n+2}-12(16D+1)$, A 16
$W12_{n+2}-12(17D)$, B 17
$W12_{n+6}-12(18D+1)$, A 18
$W12_{n+6}-12(19D)$, B 19
$W12_{n+10}-12(20D+1)$, A 20
$W12_{n+10}-12(21D)$, B 21
$W12_{n+3}-12(22D+1)$, A 22
$W12_{n+3}-12(23D)$, B 23
$W12_{n+7}-12(24D+1)$, A 24
$W12_{n+7}-12(25D)$, B 25
$W12_{n+11}-12(26D+1)$, A 26
$W12_{n+11}-12(27D)$, B 27
$\overline{P12_n-12}$ 28
$\overline{P12_n+1}$ 29
$\overline{P12_{n+2}-12}$ 30
$\overline{P12_{n+3}}$ 31

Fig. 16—CIRC encoder configurations.

head from the disc center is provided. In addition to standard control functions, control functions based on this sensor are also used.

The result is improved system reliability.

6. Conclusion

The Fujitsu Premastering System enables easy mastering and production of CD-ROMs. It consists of a personal computer, a CD-WO recorder and interface card, and software. Since most of the steps necessary to master a CD-ROM have been automated, this system is ideal for the development of CD multimedia applications. One of the future goals for the Premastering System will be a CD-ROM simulation that enables true simulation of the FM TOWNS CD-ROM. Such a simulation will make it easier for developers to debug CD-ROM applications.

References

1) Tamura, Y., Fuma, M., and Kanei, S.: Technology of CD Families. (in Japanese), *J. Inst. Televis. Engrs, Jpn,* **43**, 6, pp. 587-593 (1989).

2) Fukumoto, A. et al.: CD Fluctuation by Returning Light of Optical Disk Pickup Tracking Signal. (in Japanese), Proc. 32nd Jpn. Soc. Appl. Phys. Related Soc., Spring Meet., Tokyo, 1985, p. 107.

3) Murakami, T., Hoshino, I., and Mori, M.: Reduction Method of Residual Push-pull Tracking Error Signal. (in Japanese), Nat. Conf. Rec., Inf. Syst. Commun. Electron. Engrs., Matsuyama, 1984, p. 266.

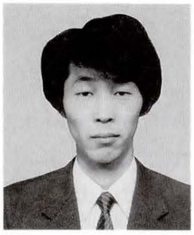
4) Tanigi, H., Ookoshi, S., and Yamada, N.: A Compensation Scheme for Lens-shift-induced Tracking Error Offset in Optical Pick-ups. (in Japanese), Nat. Conf. Rec., Inf. Syst. Commun. Electron. Engrs., Matsuyama, 1984, p. 267.

5) Takada, J.: Algebraic System and Coding Theory. (in Japanese), Tokyo, Maki-shoten, p. 167.

6) Nakajima, H., and Ogawa, H.: Compact Disk Reader. (in Japanese), 2 ed., Tokyo, OHMSHA LTD., 1988, 268p.

Yasushi Ashikaga
Hardware Dept.
Personal System Div.
FUJITSU LIMITED
Bachelor of Communication Eng.
Osaka University 1981
Specializing in Research and
Development of Personal Computers

Masafumi Naitoh
Multi-Media Engineering Dept.
Information Equipment Div.
FUJITSU GENERAL LIMITED
Bachelor of Electric Eng.
Chuo University 1980
Specializing in Research and
Development of Video Disk and
Data Disk Systems



Kazuhito Fukuoka
Engineering Dept.
Personal System Div.
FUJITSU LIMITED
Kisarazu Technical College 1983
Specializing in Research and
Development of Personal Computers

UDC 003.324.2:681.3.02

F6365 Japanese Document Reader

● Jun Sato ● Yoshinori Kuwamura ● Chieko Ohno

(Manuscript received March 5, 1990)

Progress in multimedia document processing has increased the need for better automatic document input. Fujitsu's Japanese-language document reader automatically reads printed documents and converts the words into computer character code. Techniques such as automatic block extraction, character segmentation, the extraction of ruled lines, character recognition, and lexical postprocessing have been built into this reader. The document reader ensures 99 percent error-free reading at a rate of 20 Japanese characters per second.

1. Introduction

The input process has long been a bottleneck in introducing a comprehensive system of document management and efficient electronic publishing. Conventional manual input through keyboards has cost enormous amounts of processing time and labor that have prevented full implementation of high-performance, high-speed document management, including storage, translation, and publishing. The input problem has affected all fields, particularly education, business, industry, and research[1),2)].

Fujitsu has developed a document reader with three goals in mind: Handling nonspecific formats, working at high speed without sacrificing accuracy, and processing multifont, multipoint documents without specific instructions.

2. Basic techniques

Fujitsu developed the following basic techniques as a basis for a document reader that automatically recognizes and reads complex Japanese-language texts while meeting the three goals above.

2.1 Automatic block extraction

Some documents, such as those prepared by a typewriter or word processor, usually have one column per page. Others, like newspapers and magazines, have multiple columns per page. Conventional document readers require that the user designate the location of each column on a page – a tedious, impractical job. Researches at Fujitsu did away with this by developing a technique called automatic block extraction. This technique is based on the analysis of histograms of black pixels. However, with various patterns of documents to be read, Fujitsu added new techniques. Figure 1 illustrates the process[2)]. The term block here means a body of sentences, such as a column or a paragraph on a page.

Automatic block extraction consists of two major steps:

2.1.1 Skew adjustment

The first, for precision, involves measuring the skew angle of the document being read. The document reader counts the number of black pixels along predetermined axes. The angle is ±5 ° in the direction of writing, horizontal or vertical. Histograms of the counted pixels are then prepared and differentiated. The angle of the greatest variation is determined as the skew angle of the document.

2.1.2 Block division

The second step of automatic block extraction is system modification of the writing direction with the skew angle. Histograms of the counted pixels are prepared for the modified

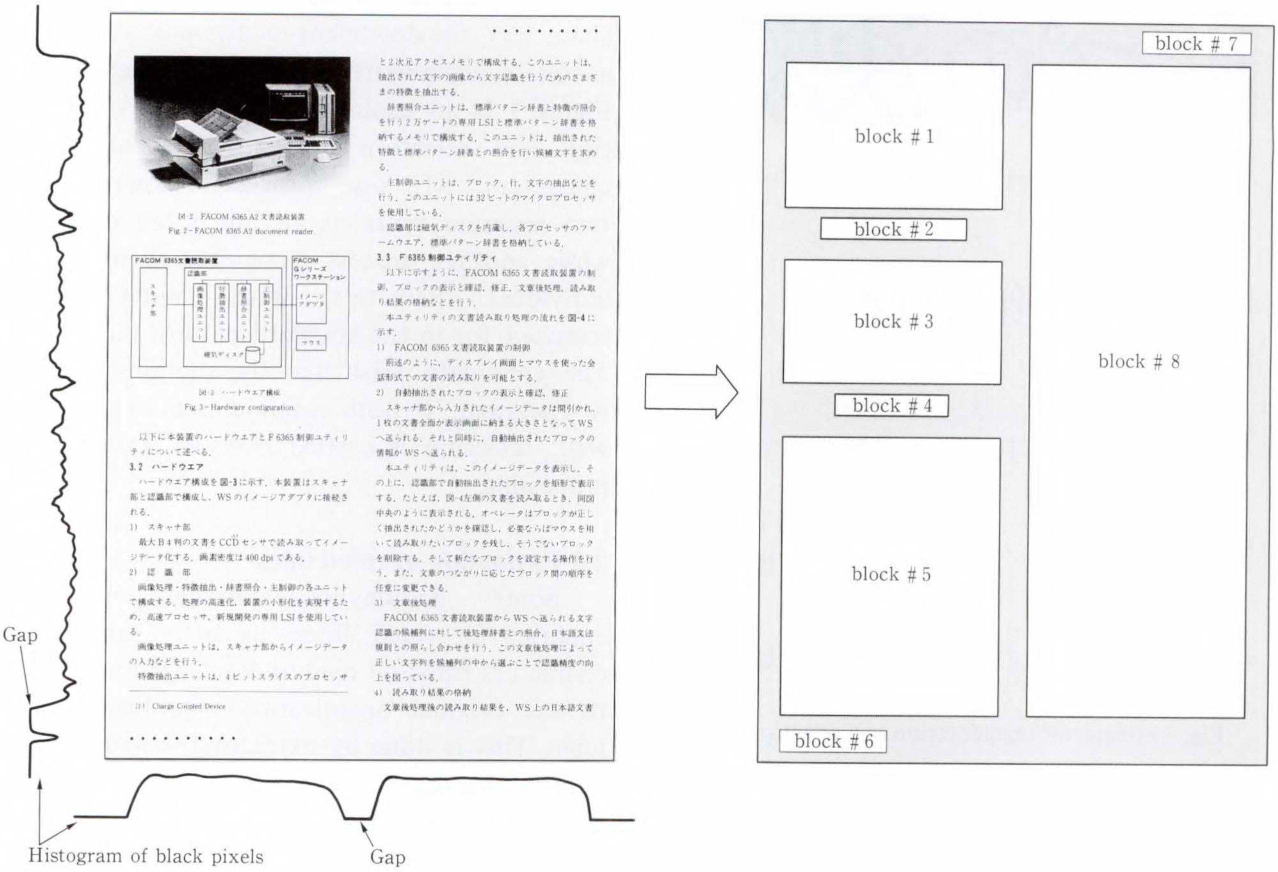


Fig. 1—Automatic block extraction[2].

horizontal and vertical directions. The system then searches for a gaps in the histogram. The gaps indicate blank areas in the document, and are used to divide the page into blocks. Then a histogram of the pixels in the direction perpendicular to the gap is generated. Gaps in this histogram define the blocks.

An actual document may be dirty or the gap between blocks may be narrow making the determination of gaps difficult. This problem is solved by making the gap threshold adjustable. The correct blocks are found by first searching under stringent conditions, then gradually increasing the tolerance.

2.2 Character segmentation

Character segmentation is the determination of a single character in a line. At this stage, if a wrong image is segmented, it is very likely that the character was misread. Character size is determined according to the continuation of black pixels in the image. Black pixels in a character line are projected perpendicular to the direction of the line and checked for continuation. However, with a Japanese language document, this technique alone is not always effective. Other techniques are also used.

Sometimes there is no uniform space between characters. As in typical newspaper headlines, characters may be spaced far apart or so close together that they overlap. Figure 2 is an example of touching and overlapping characters. The system must correctly segment each character even if it is in such a position.

To segment overlapping characters or characters that touch, the document reader estimates the size of the character. At the first segmentation, such a character can be segmented together with other characters as one unit.

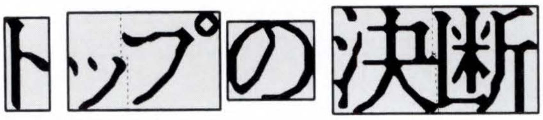


Fig. 2–Touching and overlapping characters.

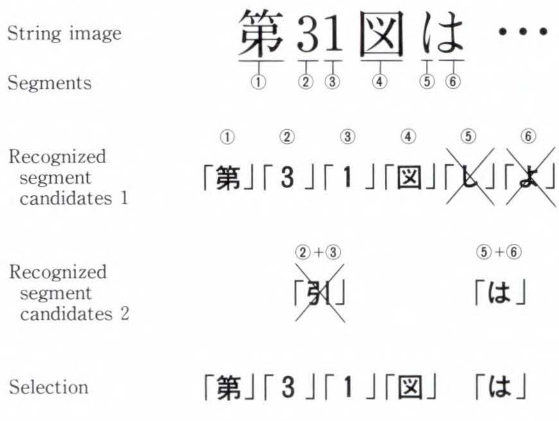


Fig. 3–Character segmentation and selection.

The document reader estimates the character width from the line height and character width measured so far. If the width of the segmented unit is determined to be significantly greater than the estimate, the characters are resegmented according to the estimated width.

To do this, a separation line is placed

1) where there are no black pixels,
2) where the connection of black pixels is thin, or
3) along the line of the estimated character width.

It is not easy to determine the width of a Japanese character. A kanji or hiragana character is almost twice as wide as an alphabetic or numeric character. Some kanji and hiragana characters are composite, that is, composed of more than one discrete unit. Examples of composite characters are 引 and は in the case of horizontal writing, or 点 and こ in the case of columnar writing. The discrete character elements must be integrated into a single character, whereas half-width characters must be segmented as individual characters (see Appendix 1).

To process composite and half-width characters, the document reader selects the most adequate segmentation from several candidates. Figure 3 is an example of character segmentation and selection. For the two half-width characters, 31, one process recognizes the two separate units as an integrated character while another process recognizes them as two individual characters. Both candidates are searched for in the standard pattern dictionary. The candidate least like the dictionary entry is discarded. If both candidates match equally well, selection is carried over to lexical post-processing.

2.3 Extraction of ruled lines

Some documents contain characters enclosed in ruled lines, as in a table. The characters must be read and arranged according to the original organization regardless of the lines. This is done by extracting the ruled lines before characters are read. Fujitsu developed a function to do this.

Ruled lines are extracted regardless of the type of line: continuous, dotted, dashed, or double. Ruled line extraction consists of four steps, as described below.

2.3.1 Making a histogram

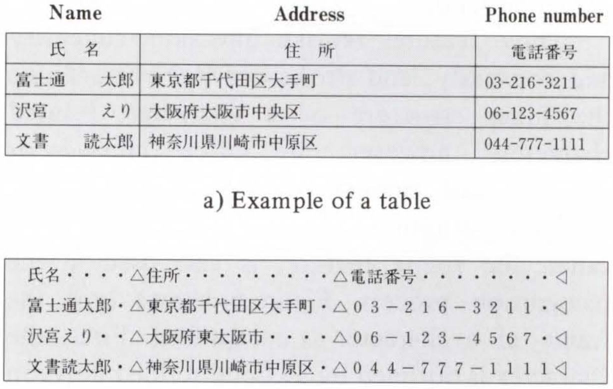
The width of most ruled lines is significantly less than the height of the characters. This characteristic is used to identify ruled lines. First, the document image is scanned along the line of characters and a histogram of black pixels is prepared. A ruled line is identified by a narrow peak in the histogram.

2.3.2 Compressing the image

To reduce processing time and to prevent misreading and identifying character parts as ruled lines, a compressed image is used for extracting ruled lines. The system compresses the original image of $n \times n$ pixels to 1×1 pixel. If there are any black pixels in the original, the pixel representing the compressed image is made black.

2.3.3 Dividing the image

To cope with the skew of a ruled line, the system divides the document image into areas. When a skew of ±3 ° is permitted, a ruled line


Name
Address
Phone number

氏　名	住　所	電話番号
富士通　　太郎	東京都千代田区大手町	03-216-3211
沢宮　　えり	大阪府大阪市中央区	06-123-4567
文書　　読太郎	神奈川県川崎市中原区	044-777-1111

a) Example of a table

氏名・・・・△住所・・・・・・・・・・△電話番号・・・・・・・・・◁
富士通太郎・△東京都千代田区大手町・△０３－２１６－３２１１・◁
沢宮えり・・△大阪府東大阪市・・・・△０６－１２３－４５６７・◁
文書読太郎・△神奈川県川崎市中原区・△０４４－７７７－１１１１◁

Separation symbols　△: Tabulation　◁: End of line

b) Recognized document file

Fig. 4–Example of processing ruled line extraction.

can be broken every 19 pixels or more (see Appendix 2). The document reader then sets the maximum size of the divided area at 19 pixels. Each divided area is searched for ruled lines.

2.3.4 Connecting the lines

All candidates of the same type of ruled line in adjacent areas are connected to determine the length of the line.

A ruled line can also serve as a delimiter between items. After ruled lines are extracted, separation symbols are inserted. This arranges the items for easy conversion into a document file during postprocessing. Figure 4 is an example of processing a table containing ruled lines. Reading accuracy can be improved for table items by adding restrictions for each item. For instance, when a name column is read, postprocessing efficiency can be increased by using a list of proper names. When a phone number column is read, the possible characters can be limited to numerals.

2.4 Character recognition

Fujitsu developed the method of multiple standpoints most applicable to the optimum method of kanji recognition[3),4)]. The method used in the document reader has been modified for recognition of printed kanji. The distance is calculated by mapping on a grid (see Appendix 3). Characters are identified by pattern matching.

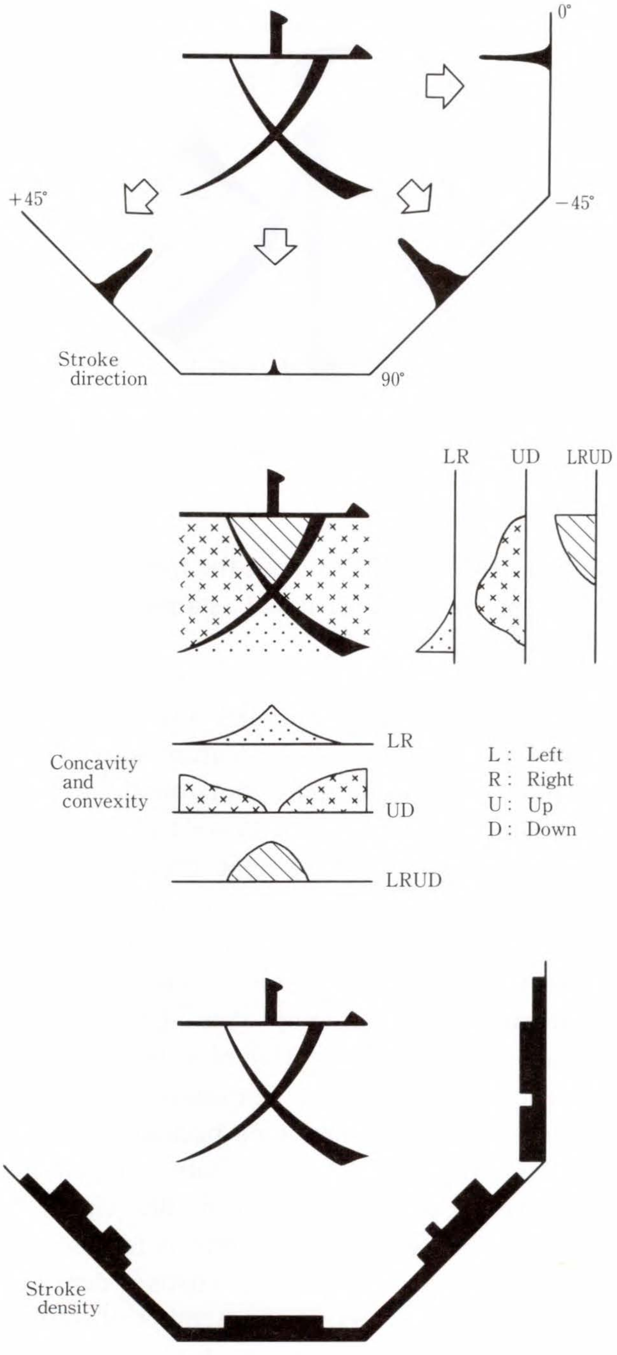


Fig. 5–Feature extraction.

The Japanese language consists of kanji, hiragana, katakana, and alphanumeric characters (see Appendix 1). The structure of these characters is complex and cannot be represented by simple straight lines, curves, and loops. Instead, a character's structure is classified into four main characteristics: stroke, concavity and convexity, complexity, and character

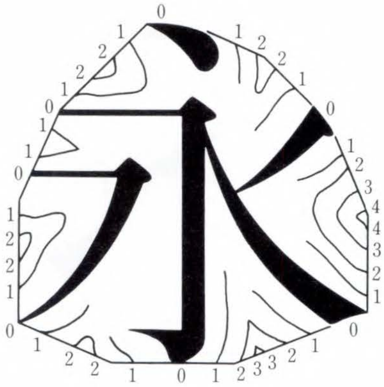


Fig. 6—Frame feature.

outline. The system evaluates the four characteristics in multi-dimensional standpoints and assigns a value to each.

Figure 5 illustrates the characteristics used for character recognition and the method of their extraction. The stroke feature is extracted by four stroke directions: horizontal, vertical, 45 ° from upper right to lower left, and 45 ° from upper left to lower right. Conventionally, all strokes are projected at one time. Fujitsu's document reader labels each direction of a character stroke and projects these different directions onto different axes. This achieves a more precise representation of a character.

The concavity and convexity feature is concerned with a character's background (that is, the white pixels). The feature represents the shapes of areas between the character strokes. The complexity feature is represented by the number of strokes (stroke density). Stroke density refers to the number of strokes crossing scanning lines in four directions: horizontal, vertical, 45 ° from upper right to lower left, and 45 ° from upper left to lower right.

The fields where the above features are evaluated can be divided to process different fonts. However, excessive division is sensitive to character displacement. The document reader divides the character image into two parts and processes each as the field. A character can be divided horizontally or vertically. Division along a 45 ° angle is also possible for evaluating stroke direction.

Three features (stroke direction, concavity and convexity, and stroke density) characterize the inner structure of a character. Printed characters, however, are often blurred or defaced, and cannot be recognized by only the inner structure. The outline of a character, called the frame feature, is also used in the recognition process. Figure 6 shows how the frame of a character is recognized. First, the character is outlined by a circumscribed polygon (up to 16 sides). Then the minimum of distances between any point on the polygon and a black pixel is measured and used as the feature quantity. The frame indicates complexity at the border of a character and is less affected by blur or defacement inside the character.

The character recognition techniques explained above facilitate rapid reading of multifont documents.

2.5 Lexical postprocessing

There are many pairs of Japanese characters that are similar. Such pairs are liable to be interchanged unless more than one character is read in context. For example, 工事, a noun meaning construction can be misread as エ事, the phonetic representation of e and a noun meaning event. Reading characters as characters and as words improves reading accuracy. The postprocessing method described below was developed to read characters as words to improve accuracy, using morpheme analysis[5)-7)]. Figure 7 shows the postprocessing flow.

2.5.1 Word retrieval

Strings of recognized characters are looked up in the system dictionary. Up to four candidates are found for each character in the document. If the first candidate doesn't make up a meaningful word, second and subsequent candidates are examined until a combination of candidates makes sense.

One problem with character strings is that Japanese sentences do not contain a space to separate words, making it difficult to find the end of a word. This means that words of any length must be extracted from candidate character strings and compared with the diction-

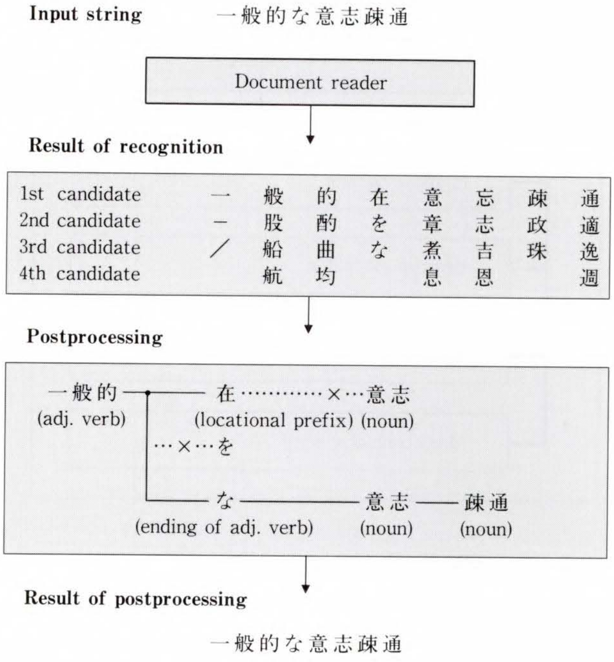


Fig. 7—Postprocessing flow.

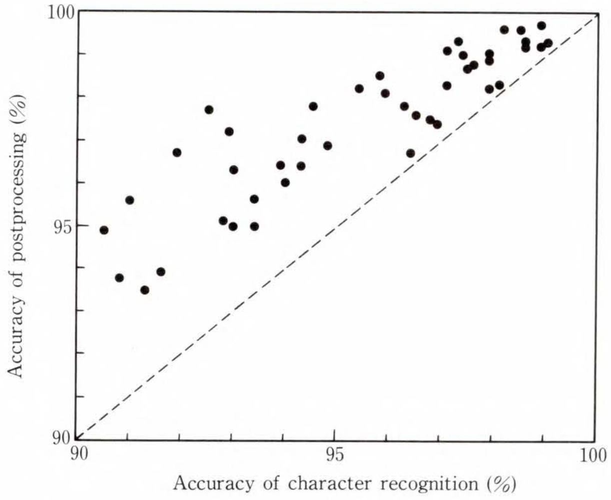


Fig. 8—Effect of the lexical postprocessing.

ary. However, it is impractical to search the dictionary for all possible combinations.

To facilitate dictionary look-up, Fujitsu has adopted the following method: The first character is read and words beginning with that character are retrieved from the dictionary. The retrieved words are then matched with combinations of candidate characters, and unmatched words are discarded. In the example of Fig. 7, words beginning with 一 are searched for. Many words are retrieved, for example, 一 (one), 一門 (family), 一般的 (general), 一揆 (riot), and so on. From this assortment, words that can be made up with the recognized character candidates are selected. In the example of Fig. 7, the words 一般的 and 一 survive. This *en masse* word retrieval speeds up the process of checking words in the dictionary.

2.5.2 Word selection

Each retrieved word is scored for its likelihood, according to the distances based on character recognition and word length. The word found to be most likely is selected.

The score of each word is estimated as follows:

$$\text{Score} = \sum_{i=1}^{l} d_i + l^2 ,$$

l : the length of the word

d_i : the distance of ith character measured by character recognition.

Once a word is determined, the next character is read and the above two processes are repeated.

2.5.3 Syntax checking

Retrieved words are also selected according to the rules of Japanese syntax (see the example in Fig. 7). Once 一般的 is determined, connectability between this word and the next candidate is examined. Neither 在 (a prefix meaning nearby) nor を (a particle) can be placed after the word 一般的. Consequently, な (the suffix of an adjectival verb) is selected as the only possible character. This process reduces misreading and helps to select the correct candidate.

In addition to the above processes, a reasoning function deals with words that are not in the dictionary. This function assumes the correct character candidate from the type of the previous character: kanji, hiragana, katakana, or alphanumeric. Words are searched for from among more than one character segmentation candidate, so as to eliminate errors attributed to incorrect character segmentation.

The lexical postprocessing described above corrects 60 percent to 90 percent of the misreadings done by simple character recognition. Figure 8 shows the effect of the lexical postprocessing. As Fig. 8 shows, the lexical postprocessing achieved a 1-5 percent improvement in recognition accuracy.

3. Hardware technologies

Fujitsu has developed dedicated hardware including dedicated processors for the document reader so as to quickly read Japanese data. This includes a large character set.

Figure 9 shows the handware configuration.

The document reader consists of an image scanner and a character recognition unit. The image scanner reads images at a resolution of 400 dpi.

The character recognition unit consists of the following four subunits:

1) Image processor

This subunit inputs image data from the scanner, and compresses and expands the data.

2) Feature extractor

This subunit collects feature data about a character image. The subunit consists of a memory accessible in two dimensions and a bit slice processor by which features are quickly derived from the character image. The maximum parallel processing performance of the subunit is 14 MIPS. At this speed, mask operations of two-dimensional images can be done 7 million times/s.

3) Pattern matcher

The dictionary reference subunit matches the features of input characters with the dictionary that contains the features of nearly 4 000 characters. The subunit then outputs a list of candidates in the order of the degree of match. The subunit supports an IC dedicated to high-speed searches of the dictionary. A 32-bit parallel reference circuit gives reference speeds of at least 20 Mbyte/s.

4) Main control subunit

The main control subunit controls the above three subunits, and executes automatic block extraction and character segmentation. The control subunit includes a 32-bit microprocessor plus another processor for image processing. The image processor is a recently developed IC dedicated to high-speed image rotation and candidate selection.

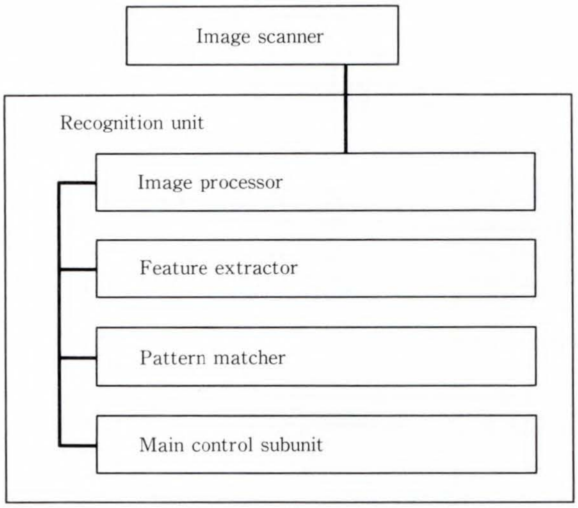


Fig. 9—Hardware configuration.

4. System overview

This chapter explains the system configuration of the document reader, flow of system processing, and specifications and performance of the document reader.

The document reader forms a document reading system when connected to a personal computer or a workstation. The specifications of the document reading system with the F6365 document reader is shown in Table 1. Figure 10 is the system with a workstation. The equipment at the left is the document reader. The image scanner is placed on top of the recognition unit. The equipment at the right is a workstation for office use.

Figure 11 shows the processing flow of the document reading system. The numbers in this figure correspond to articles 1) to 5) below.

1) The document image is fed to the system via the image scanner.
2) The recognition unit extracts blocks from the document image. The result is superimposed on the document image, which

Table 1. Specifications of F6365

Item	Specifications
Objects read	Documents printed in Japanese Horizontal or vertical writing (option)
Maximum document size	JIS B4 size (257 x 364 mm)
Readable character types	Total number of printed characters 3 911 Kanji characters (3 612) Hiragana and Katakana characters (169) Alphanumeric characters and symbols (130)
Readable fonts	Multipoint (Ming-style, Gothic, and textbook-style)
Readable character sizes	Multipoint 7 points to 28 points Characters in any one line must be the same·size
Reading speed	20 characters/s (at recognition unit only)
Processing speed	3 min/page (about 1 400 characters) Including lexical postprocessing Excluding correcting of results
Accuracy	Based on Fujitsu's database With character recognition only 96% After lexical postprocessing 99%

is displayed on the personal computer or the workstation.

3) The operator checks the display to confirm that automatic block extraction has been done correctly. If necessary, the operator can use the mouse to delete unnecessary blocks or add new blocks.
4) After blocking is confirmed, the recognition unit starts to read the characters in the blocks.
5) Recognized candidates are sent to the personal computer or the workstation and undergo lexical postprocessing. The results

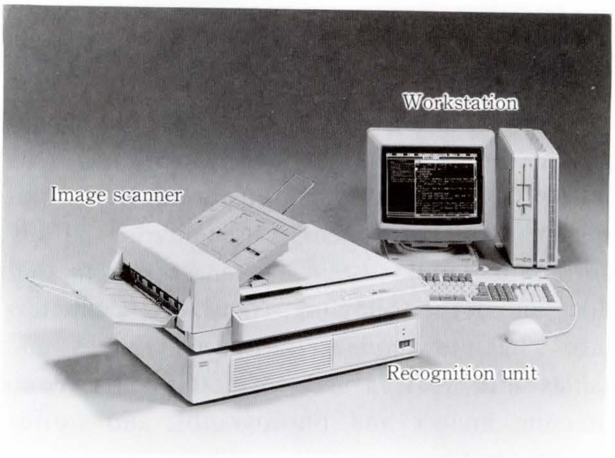


Fig. 10–Document reading system.

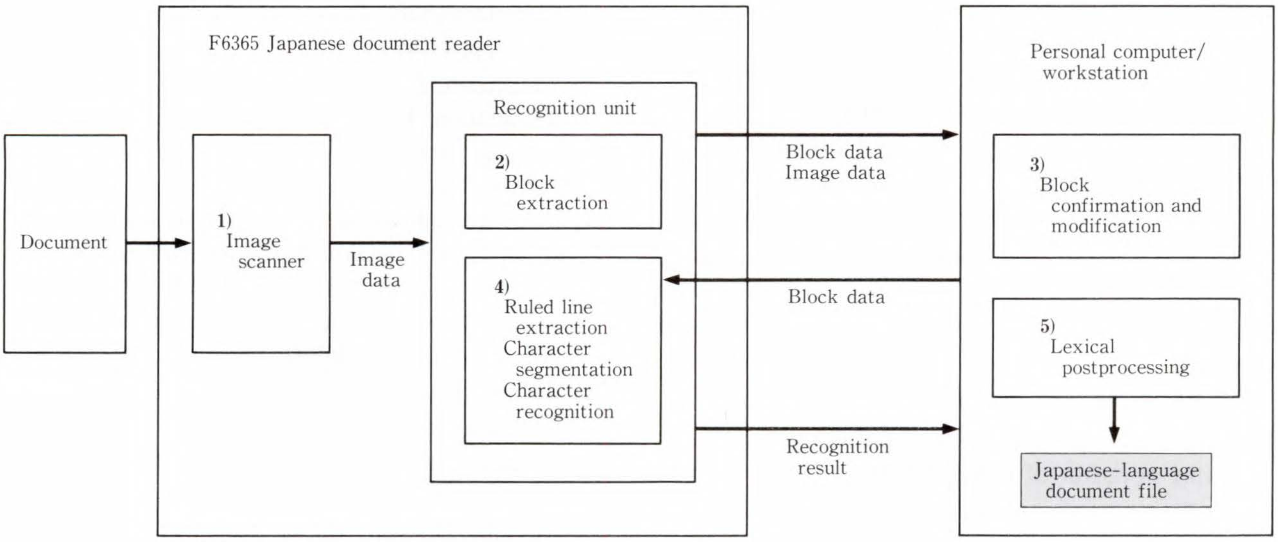

F6365 Japanese document reader
Document
1) Image scanner
Image data
Recognition unit
2) Block extraction
4) Ruled line extraction Character segmentation Character recognition
Block data Image data
Block data
Recognition result
Personal computer/ workstation
3) Block confirmation and modification
5) Lexical postprocessing
Japanese-language document file

Fig. 11–Processing flow of Japanese document reading system.

are stored in the Japanese-language document file. The operator can edit the results with a word processor.

5. Conclusion

As explained above, new techniques for automatic reading of Japanese documents have been developed and realized in the document reader. Fujitsu's objective, an automatic reading system for documents of any format, including magazines and books, has been achieved. It achieved 99 percent accuracy in one fifth of the time of manual input through keyboards. Though, the document reader mentioned above was designed for reading documents in Japanese, Fujitsu is preparing to enhance it for European languages, Asian languages, and other languages.

Future improvement to the document reader may develop in two directions. One is the development of a fully automatic input system for multimedia documents. "Fully automatic input system" means a system that does all of the following: translating characters into character codes, translating figures and tables into vectors and character codes, compressing images and photographs, and storing the data in files.

Another possible improvement of the document reader is a media converter, which translates document image data into character codes. The media converter, an inexpensive version of the document reader with basic functions in one board or one chip, can be incorporated in a variety of systems.

Fujitsu is continuing research and development in both areas: techniques for automatic input of multimedia documents and for compact implementation of media conversion.

6. Appendices

Appendix 1

Written Japanese consists of three types of characters, kanji, hiragana, and katakana.

Kanji consists of several thousands of ideograms. Some kanji are very simple but some are very complex. For example, 一, 二, 三, 鸞, 鬱 and 鸚 are all kanji characters. Some kanji characters consist two or more units. For example 村 consists of 木 and 寸. Characters consisting of two or more units are called composite characters in this manuscript. Both hiragana and katakana are syllables. あ is a hiragana character corresponding to the sound ə, while ア is the katakana character for this sound.

Appendix 2

If the skew of a document is represented by θ, the divergence of the document image from the horizontal is indicated by $\tan\theta$. If the tolerance of the skew of a document is ±3 °, the maximum divergence is $\tan(3°)$, which is about 1/19. Therefore, the divergence in the Y direction is at most 1 pixel for every 19 pixels in the X direction.

Appendix 3

The distance, d, between co-ordinates $\boldsymbol{A}$ and $\boldsymbol{B}$ along grid lines is calculated as follows:

$$d = \sum_{i=1}^{n} |a_i - b_i|,$$

$$\boldsymbol{A}\,(a_1, a_2, a_3, ..., a_n),\ \boldsymbol{B}\,(b_1, b_2, b_3, ..., b_n).$$

References

1) Rosenfeld, A., and Kak, A.C.: Digital Picture Processing. 2nd ed., 2 Vols., New York, Academic Press, 1982.

2) Izaki, Y., Shimoyama, A., and Nishino, F.: FACOM 6365 Document Reader. (in Japanese), *FUJITSU*, **40**, 1, pp. 41-47 (1989).

3) Yamamoto, E. et al.: Handwritten Kanji Character Recognition Using the Features Extracted from Multiple Standpoints. Proc. IEEE 1981 Conf. Pattern Recognition Image Processing, pp. 131-136.

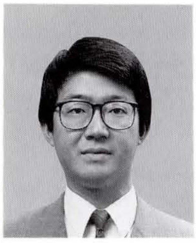
4) Hai, T., Kabuyama, Y., and Yamamoto, E.: A Method for Handwritten Kanji Character Recognition-Recognition Method by Multiple Stand-point and Particular Shape Extraction-. (in Japanese), *Trans. IEICE, Jpn.*, **J68-D**, 4, pp. 773-780 (1985).

5) Nishino, F., and Takao, T.: Implementation of Post-Processing for Japanese Document Readers. (in Japanese), *Inf. Proc. Soc., Jpn., WGNL*, **64**, 6, pp. 45-52 (1987).

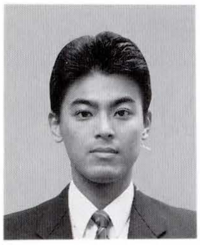
6) Nishino, F., and Takao, T.: Post-Processing for Japanese Document Readers. Proc. Int. Conf. Com-

puter Processing of Chinese and Oriental Languages, Toronto, 1988, pp. 166-169.

7) Takao, T., and Nishino, F.: Implementation and Evaluation of Post-processing for Japanese Document Readers. (in Japanese), *Trans. Inf. Proc. Soc., Jpn.*, **30**, 11, pp. 1394-1401 (1989).



Jun Sato
Engineering Dept.
Advanced System Products Div.
FUJITSU LIMITED
Bachelor of Precision Mechanical Eng.
The University of Tokyo 1982
Master of Precision Mechanical Eng.
The University of Tokyo 1984
Specializing in Character Recognition
and Document Management System



Yoshinori Kuwamura
Engineering Dept.
Advanced System Products Div.
FUJITSU LIMITED
Bachelor of Electronics Eng.
University of Osaka Prefecture 1986
Master of Electronics Eng.
University of Osaka Prefecture 1988
Specializing in Character Recognition

Chieko Ohno
Engineering Dept.
Advanced System Products Div.
FUJITSU LIMITED
Bachelor of Science
Ochanomizu Women's University 1989
Specializing in Character Recognition

UDC 681.327.2(084.11)

Fujitsu Advanced Drawing Capture System: FADCS

● Yukikazu Kabuyama ● Teruo Ootake ● Jun-ichi Koizumi

(Manuscript received March 5, 1990)

This paper describes FADCS, which enables high-speed input of sketches drawn by hand into personal computers or workstations. FADCS was developed to process general-purpose functions, independent of drawing types. The major functions are image to vector conversion and extraction and recognition of characters. The first function converts image data into vector data. This function can recognize graphic primitives (circles, etc.) to facilitate drawing-specific processing such as actual drawing input. This technique is intended for basic image-to-vector conversion on personal computers. Subsequently, the functions will be improved and the degree of freedom for media operation with multimedia personal computers will be increased.

1. Introduction

As the capabilities of personal computers improve, jobs and various media which were processed by general-purpose computers are now processed by personal computers or workstations. Drawings are one of these media. The number of drawings processed has increased due to the progress in Computer Aided Design (CAD) systems.

Display units are widely used in design. However, paper is still used as the means to input drawings to CAD systems.

This is because paper has the following advantages.

1) Large drawings can be viewed quickly (A1- or A0-size drawings are widely used).
2) More than one drawing can be viewed at the same time. (Most designs require more than one drawing.)
3) Many designers prefer paper because it is a familiar medium.
4) Offices whose work involves many drawings tend to use existing drawings to reduce design work.

Digitizers are also used to input drawings prepared by hand. However, digitizers have the following two problems, input is time-consuming, and vector precision depends on the operator.

To solve these problems, automatic input and high-speed, consistent processing is required.

For example, automatic input is required in the following cases:

1) Piping and instrument (P&I) diagrams
2) Machine drawings
3) Drawings of facilities
4) Maps
5) Circuit diagrams
6) Printed circuit artwork.

This paper outlines Fujitsu Advanced Drawing Capture System (FADCS) and describes the techniques of the recognition unit, the main unit of FADCS[1),2)]. This paper also gives examples of processed drawings and describes future expansion.

2. System configuration

FADCS was developed to enable automatic input of sketches drawn by hand into computer systems, especially CAD systems. Figure 1 shows the system configuration of FADCS. The functions of the three units composing the system are outlined below.

1) Image scanner
   i) Conversion of drawings into electrical signals
2) Recognition unit
   i) Binarization of converted signals
   ii) Conversion from image data to vector data
   iii) Separation of text graphics
   iv) Recognition of graphics primitives
   v) Character recognition
3) Personal computer/workstation
   i) The host computer which controls the image scanner and recognition unit
   ii) Drawing-specific recognition processing based on the information obtained by the recognition unit (realized as application software)

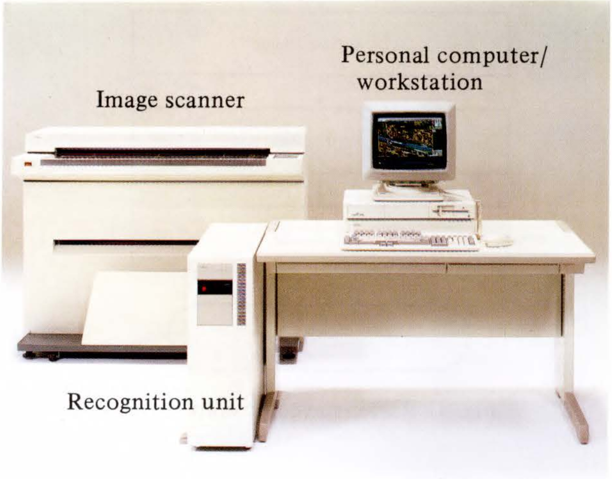


Fig. 1–FADCS (Fujitsu Advanced Drawing Capture System).

Processing after input by the image scanner can also be performed by the personal computer (host). For actual work, however, the input of A3- to A0-size forms is required. When an A3 drawing is read at 400 dpi (dots per inch), the resolution required for reading minute details, the amount of data after binarization is about 4 Mbytes. Image processing is required to convert this data into vectors. Therefore, much time is consumed in processing, and it is difficult to perform actual work only using a personal computer.

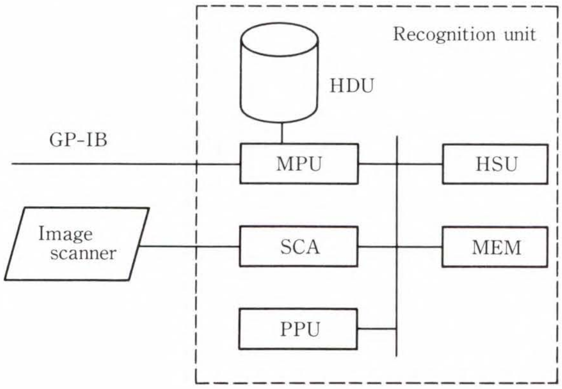
This problem is solved by the recognition unit. This unit enables high-speed processing for general functions, such as vector conversion. With this unit, the personal computer is needed only for drawing-specific recognition processing[3)-5)].

3. Hardware

This chapter describes the hardware of the recognition unit. High-speed processing is provided by hardware devices such as the image processing unit and image data compression/expansion unit. The recognition unit

Table 1. Specifications of the recognition unit

Item	Specifications
Unit	F2512A
Interface	GP-IB
Processors	MC68020, MC68030
Memory	32-, 16-, 4-Mbytes
Size ($l \times b \times h$)	230 × 700 × 700 (mm)
Weight	60 kg
Power supply	0.8 kVA



MPU	Main control, recognition and interface
HSU	High-speed recognition (option)
SCA	Scanner interface
PPU	Picture processing
MEM	Work storage for image and work tables
HDU	Storage for drawing data and programs

Fig. 2–Configuration of the recognition unit and functions of each component.

Table 2. LSIs

LSI	Unit	Gates	Package	Functions
IOC	MPU	10 000	RIT-256	I/O control, DRAM control
VIC	MPU/HSU	9 500	RIT-256	Video-bus interface controller
BNZ	SCA	2 500	RIT-135	Binarization of gray-level image
PLR	PPU	1 700	RIT-135	Pipeline register
LSA	PPU	4 700	RIT-256	Local storage control
SEQ	PPU	5 000	RIT-135	Program sequencer
LRA	PPU	11 000	RIT-64	Image density conversion

can also be miniaturized by the development of special LSIs. Line approximation by image splitting (explained in the next chapter) enables large drawings to be processed with a small amount of memory. This also contributes to unit miniaturization.

Table 1 lists the specifications of the recognition unit. Figure 2 shows the configuration of the recognition unit and functions of each component. Table 2 lists the LSIs that the authors have developed.

4. Recognition techniques

This chapter explains the techniques used by the recognition unit[6]. Figure 3 shows the general flow of drawing recognition by this unit.

4.1 Raster-to-vector conversion

Figure 4 shows the flow of raster-to-vector conversion.

Details of the process are explained below.

4.1.1 Line approximation by image splitting

The recognition unit splits the input image data into small images and approximates the lines in each image. After all the small images have been processed, the unit connects the obtained results and generates the vector data of the entire image. Using this technique, the unit can process large drawings with a small amount of image memory.

Figure 5 shows how the image is split. Around a split image, an area overlapping the adjacent image is taken. This area is used to connect the vectors generated for each split image. When a split image is thinned, the periphery of the thinned image becomes inconsistent with the thinned image of the adjacent one. By taking the overlapping area,

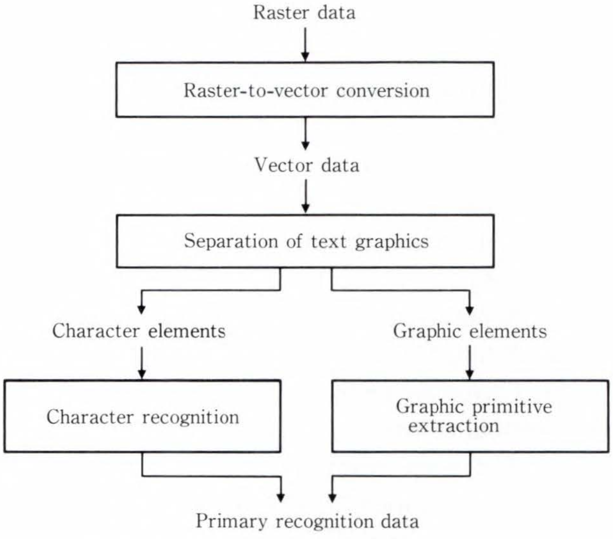


Fig. 3—Flow of drawing recognition.

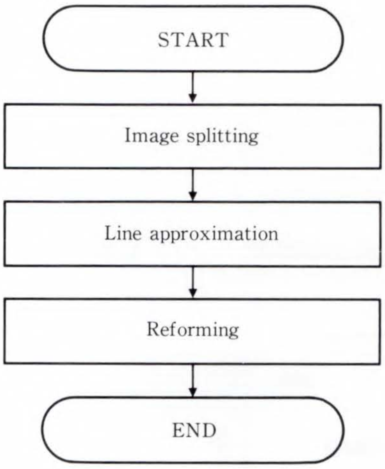


Fig. 4—Raster-to-vector conversion.

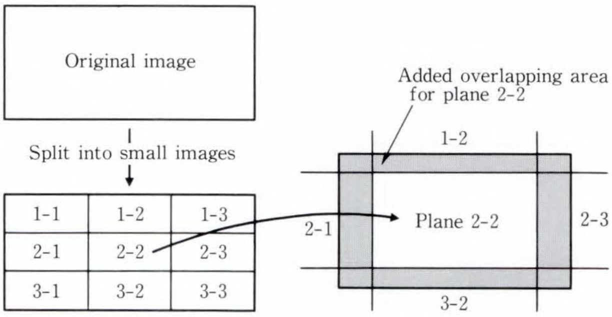


Fig. 5—Image splitting.

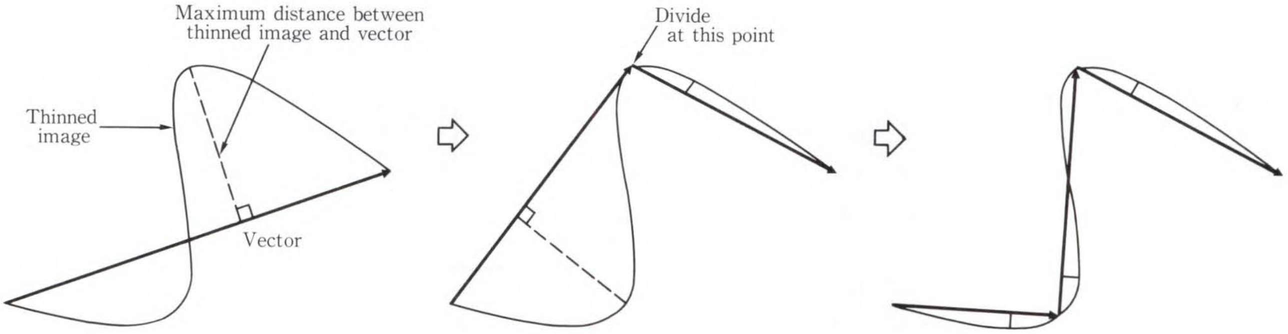


Fig. 6—Line approximation.

the inconsistency can be limited within this area, and the generated vector data can be connected.

In the thinning process, the image data of the center line is generated from the original image data. This thinning process is performed using the Deutsch's algorithm[7].

Then, the thinned image data is approximated to lines by the following steps.

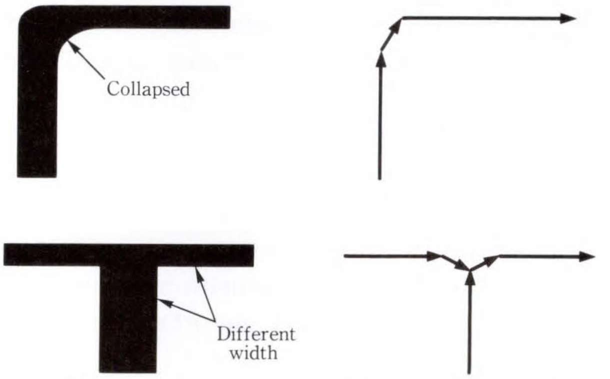
1) The chain of pixels between an endpoint or branching node and another endpoint or branching node is fetched from the thinned image. Then, the vector connecting the beginning and end of the chain is created.
2) After the distance between each point on the chain and the vector is calculated, the point most distant from the vector is found.
3) If the distance is greater than a predetermined threshold, the vector is divided into two vectors at the point found at step 2).
4) If the distance is less than or equal to the threshold, the vector is assumed to be the result of approximating the thinned image.
5) Above steps 1) to 4) are repeated for every chain on the image.

Figure 6 shows an example of processing a curved line in this way.

4.1.2 Reforming

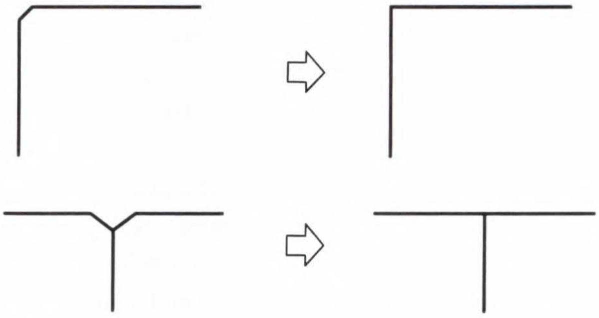
The reforming process automatically reforms distortions in the vectors obtained by line approximation.

Figure 7 shows two examples of distortion. As shown in this figure, a thinned image which differs from the original image may be often generated by thinning a collapsed portion in the image or a portion with different line widths. If such portions are subject to line approximation, distorted or redundant vectors are generated. This reduces the reproducibility of drawing data. The reforming process corrects such distortions and improves the reproducibility of drawing data.



a) Original image b) Result of vectorization

Fig. 7—Examples of distortion.



a) Registered distortion pattern b) Reformed pattern

Fig. 8—Examples of registered patterns used in reforming.

The reforming process collates the generated vector data with registered distortion patterns. Figure 8 shows examples of registered distortion patterns and the results of reforming. The reforming process searches the vector for distortion patterns that match the registered distortion patterns. When the patterns match, they are replaced by the registered reformed pattern. In this way, the vector data is corrected.

Collation with vector data enables various types of distortions to be handled flexibly. For example, as the result of line approximation, distortion patterns specific to the drawing type may appear (e.g. P&I diagrams, machine drawings and drawings of facilities). In such cases, new distortion patterns can be handled by registering the corresponding distortion patterns. To process distortions specific to a drawing, validity/invalidity can be specified for each registered pattern.

4.2 Separation of text graphics

In general, a drawing contains many characters of various sizes and inclinations. How these characters are handled is important for automatic drawing input.

This system separates the vectors obtained by raster-to-vector conversion into graphic elements and character elements. The processing for graphic elements is independent of that for character elements. The separation of these elements has the following advantages: Objective data can be extracted, and elements of both types can be processed in parallel.

To recognize a character, its position, size, and inclination must be detected exactly. This system separates characters and extracts information necessary for recognition by using the technique for extracting character strings explained below.

The extraction of character strings consists of the following three processes.

4.2.1 Extraction of character candidates

First, vector data is searched to extract character candidates. Connected vectors are grouped. The sizes of the areas for each group are obtained, and the groups having the sizes appropriate to characters are regarded as candidates depending on the following characteristics.

1) A character has a relatively small area.

2) It is isolated from other characters and graphic data.

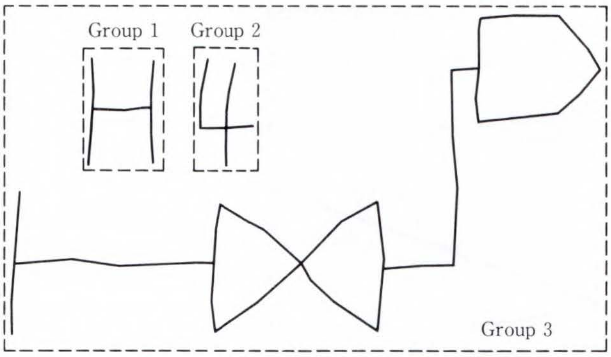


Fig. 9—Extraction of character candidates.

In Fig. 9, the segments indicated by solid lines are divided into the three groups enclosed in dotted lines. Each group whose vertical or horizontal size conforms to the character size is regarded as a character candidate. In this example, group 1 and group 2 are the character candidates.

4.2.2 Integration of character candidates

Character strings are created by integrating adjacent character candidates. The character strings are extracted by repeating the integration of the most appropriate characters and character strings on the basis of the pitch between the character candidates and the angle between the characters or between the character and the string.

1) Pitch

The distances between all character candidates in the predetermined search range are obtained, and proximity values are assigned according to the distances. At this time, a candidate with a shorter distance is assigned a larger proximity value.

When there are character candidates B, C, and D in the search range from character candidate A, as shown in Fig. 10, the distances from candidate A to B, C, and D are measured. The largest proximity value is given to candidate C having the shortest distance, and the smallest proximity value is assigned to D.

2) Angle between a character and a string

For all character strings in the above search range, the proximity values are determined based on the difference between the following angles.

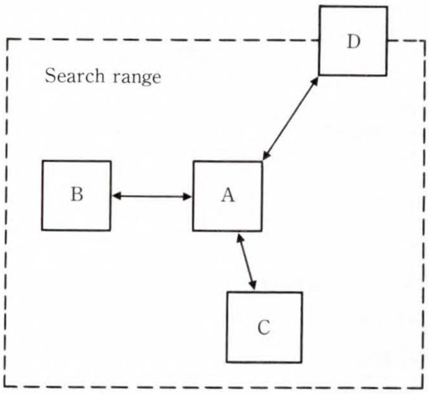


Fig. 10–Pitch.

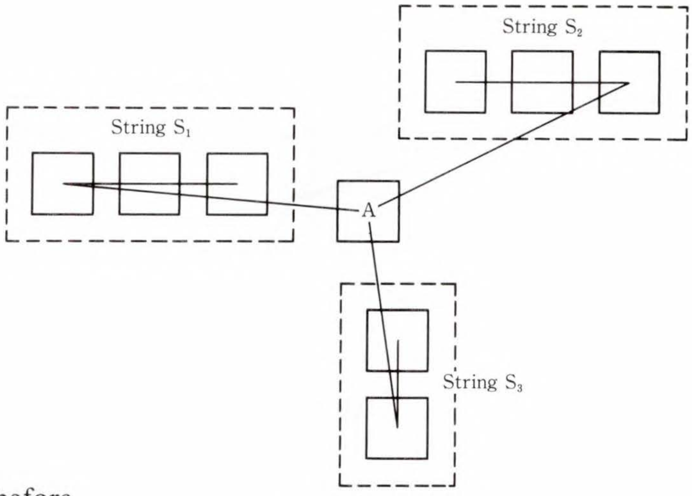


Fig. 11–Angle.

i) Angle of the character string before integration (angle between the horizontal axis and the line connecting the centers of the first and last characters in the character string).

ii) Angle of the character string after tentative integration.

A larger proximity value is assigned if the difference between the angles is smaller.

In Fig. 11, the differences between the angles before and after integration are obtained for character strings S_1, S_2, and S_3 that can integrate character candidate A. The largest proximity value is given to S_3 having the smallest difference. The smallest proximity value is assigned to S_2.

Sometimes, there may be a character for which the integration character string cannot be uniquely determined because the superiority or inferiority of several possibilities cannot be determined only by the distances and angles. This character is regarded as an ambiguous string (consisting of one character) and determination is reserved.

4.2.3 Reevaluation

To improve the precision of extraction, the extracted character strings are reevaluated and verified as follows.

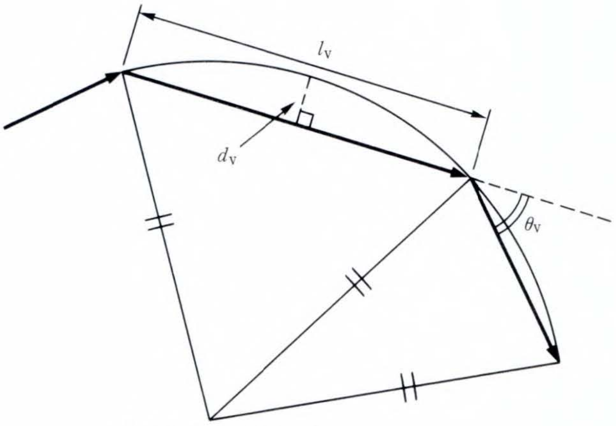
1) An ambiguous string is reevaluated. If the nearest character string is an ambiguous string, the two character strings are integrated.
2) The dispersion of characters in the integrated character string is evaluated. If there is a lot of dispersion, the character string is split at a point where the dispersion decreases.
3) Whether two or more character strings can be integrated is checked. If possible, they are integrated.
4) Periods or colons may not conform to the character size because they are small. When there is a small area in a character string area, it is regarded as a period or colon, and integrated as a character in the character string.
5) If there is no integrable character candidate near a character, the single character is regarded as a character string. At this time a unique predetermined inclination value is given.

4.3 Extraction of graphic primitives

For vectors separated as graphic elements by the separation of text graphics, the graphic primitives (straight lines, circles, arcs, and polygons) are extracted according to the shape and connection of the chain of vectors. This process aims at supplying useful data for drawing-specific processing (symbol recognition) by the host computer.

4.3.1 Extraction of circles and arcs

The recognition unit can extract graphic data of circles and arcs in the drawing, and can output the information (coordinates of the center, and radius) for that portion. Representing a circle or arc can improve the



l_V : Length of a vector
θ_V : Angle with the adjacent vector
d_V : Distance between the vector and the arc created with the adjacent vector

Fig. 12—Parameters to select arc candidate vectors.

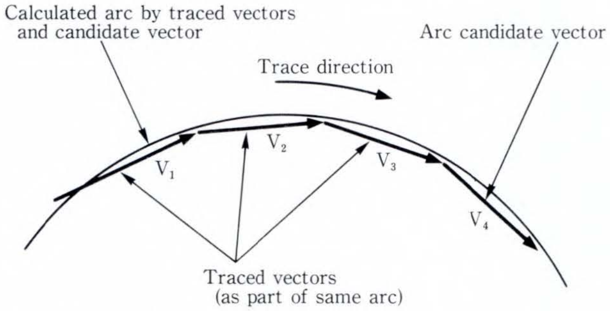

Calculated arc by traced vectors and candidate vector
Arc candidate vector
Trace direction
V_1
V_2
V_3
V_4
Traced vectors (as part of same arc)

Fig. 13—Arc extraction by tracing arc candidate vectors.

reproducibility of drawing data.

In the recognition unit, circles and arcs are extracted using the following steps.

1) The proximity values are obtained as shown in Fig. 12. The vectors for which the above values are less than or equal to the thresholds are regarded as arc candidates. Depending on the proximity values, one of the attributes "high probability" and "low probability" is assigned to the arc candidates.
2) An arc candidate with the attribute of "high probability" is selected. It is used as the starting vector for tracing. The arc candidate connected to that vector is searched.
3) The least-square error between the arc candidate that was found and the traced vectors is evaluated.
4) If the error is greater than the threshold, the trace is discontinued. If the error is less than or equal to the threshold, the arc candidate is traced assuming that it is part of the same arc.
5) Circles and arcs are extracted by repeating steps 3) and 4) if the trace is possible.

Figure 13 outlines the tracing of arc candidates.

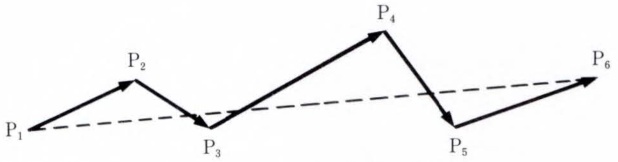


Fig. 14—Straight line extraction.

4.3.2 Straight line extraction

Straight line extraction traces the chain of continuous vectors and approximates the vectors that satisfy the conditions of a straight line. For the vectors approximated by raster-to-vector conversion, a portion which can be regarded as a straight line is generally identified, then determined to be a straight line.

The straight line condition is summarized as follows:

The difference (error) between the points (P_1, P_2, ..., P_6 in Fig. 14) and the tentative straight line ($\overrightarrow{P_1\ P_6}$ in the same figure) is within the allowable range.

For this system, the above condition is qualitatively defined by the following two conditions.

1) The distance between the endpoint of each vector in the chain and the straight line to be approximated is less than the predetermined threshold t_1 for all endpoints.
2) The ratio of the sum of the distances of all endpoints obtained above to the length of the straight line is less than the predetermined threshold t_2.

When both of the above conditions are satisfied, the vector chain is approximated as a straight line connecting the starting point and endpoint of the chain.

In Fig. 14, if the imaginary straight line connecting P_1 and P_6 is assumed for the vector chain $P_1 \rightarrow P_2 \rightarrow$, ..., $\rightarrow P_6$, the distances D between the imaginary line and each of the endpoints P_2, P_3, P_4, and P_5 are obtained from the following: The outer products of imaginary line and segments connecting the starting point P_1 of the imaginary line and each of the endpoints P_2, P_3, P_4, and P_5 are calculated.

The outer product is as follows:

$$|\overrightarrow{P_1 P_6} \times \overrightarrow{P_1 P_n}| = |\overrightarrow{P_1 P_6}| \cdot |\overrightarrow{P_1 P_n}| \sin\theta,$$

where $n = 2, 3, 4, 5$ and θ is the angle of $\overrightarrow{P_1 P_n}$ to $\overrightarrow{P_1 P_6}$.

If the distance between the imaginary line and each endpoint P_n is D_n, $\sin\theta = D_n / |\overrightarrow{P_1 P_n}|$.

In this case, there are four objective endpoints (P_2, P_3, P_4, and P_5).

$$D_n = |\overrightarrow{P_1 P_6} \times \overrightarrow{P_1 P_n}| / |\overrightarrow{P_1 P_6}|.$$

Therefore, when the following conditions are satisfied, the straight line $\overrightarrow{P_1 P_6}$ is extracted from $P_1 \rightarrow P_2 \rightarrow, \ldots, \rightarrow P_6$: $|D_n| \leqq t_1$ and $\Sigma D_n / |\overrightarrow{P_1 P_6}| \leqq t_2$, $(n = 2, 3, 4, 5)$.

4.3.3 Polygon extraction

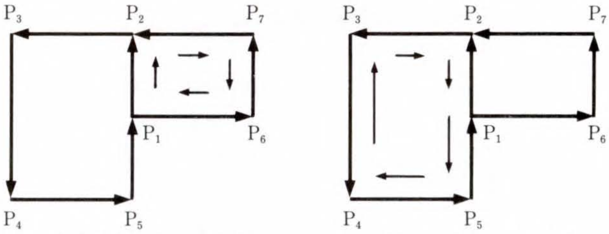
This system extracts the chain of continuous vectors constituting a closed area (surface area) as a polygon.

For example, the following items constitute closed areas in various drawings: houses and blocks in a street map, and symbols in a circuit diagram or plant drawing.

Polygons are used as the basic data for recognizing such items.

A polygon is extracted as follows.

1) In preprocessing, vectors that are not closed loops (vectors with open nodes) are excluded. Though a vector has connected vectors, if all connected vectors are excluded, that vector is also excluded. This process is repeated until no more vectors are excluded. The remaining vectors always constitute closed loops.
2) An untraced vector is selected. The trace is started from this vector along its direction.
3) A vector can be traced twice in different directions. It must not be traced twice in the same direction.
4) If a branching node is reached during the trace, the rightmost vector of the untraced vectors having the branching node is selected, and the trace is continued for this vector.
5) If the trace returns to the starting vector, the locus of this trace is regarded as a polygon.
6) The starting vector is traced in the reverse direction, and steps 2) to 5) are performed.



a) First extraction b) Second extraction

Fig. 15–Polygon extraction.

Figure 15 shows an example of a trace. In this example, vector $P_1 \rightarrow P_2$ is selected as a starting vector. The trace starts in the direction from P_1 to P_2 {see Fig. 15 a)}. Since P_2 is a branching node, there are two routes (to P_3 and P_7). P_7 is selected according to step 4) above. Then, the trace reaches P_1 via untraced vectors ($P_7 \rightarrow P_6$, $P_6 \rightarrow P_1$), and polygon $P_1 \rightarrow P_2 \rightarrow P_7 \rightarrow P_6 \rightarrow P_1$ is extracted. Then, vector $P_1 \rightarrow P_2$ is traced in the reverse direction (P_2 to P_1) {see Fig. 15 b)}, and polygon $P_2 \rightarrow P_1 \rightarrow P_5 \rightarrow P_4 \rightarrow P_3 \rightarrow P_2$ is extracted.

For actual use, the significance of the extracted information is improved depending on the polygon size and connection conditions (whether isolated or in contact with another figure).

4.4 Character recognition

Areas separated as a character by the separation of text graphics are recognized by edge extraction developed with the OCR[8]. Alphanumeric characters and symbols (50 characters in total) can be recognized.

Character recognition consists of the following three steps.

1) Picking up a character area

According to the character position and inclination obtained by the separation of text graphics, an area for one character is picked up from the original image and rotated so that it is horizontal (see Fig. 16).

2) Feature extraction

The edge extraction process extracts the character features (number of loops, right convex curve, left convex curve, right concave curve, left concave curve, and their positional relationships), as seen from four directions (upper, lower, left, and right) (see Fig. 17).

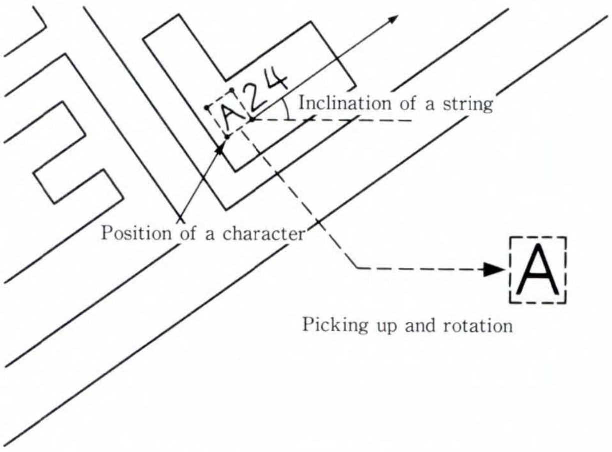


Fig. 16—Picking up a character area and rotating it.

3) Collation

The dictionary for character recognition has a tree structure. The search is made from the top of the tree to an end in the direction satisfying the conditions of the features. When an end node is reached, the code assigned to that node is regarded as the result of recognition.

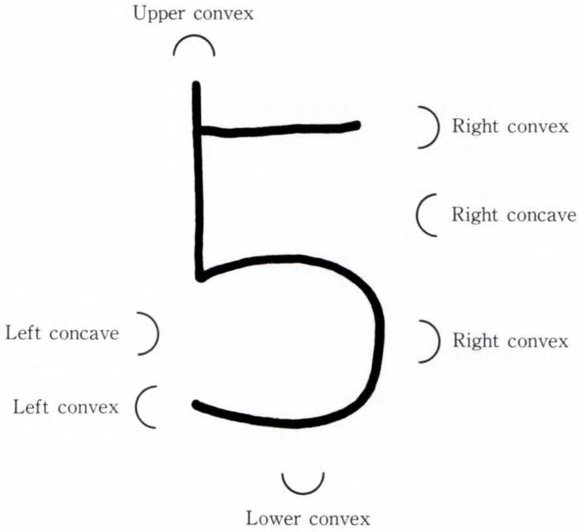
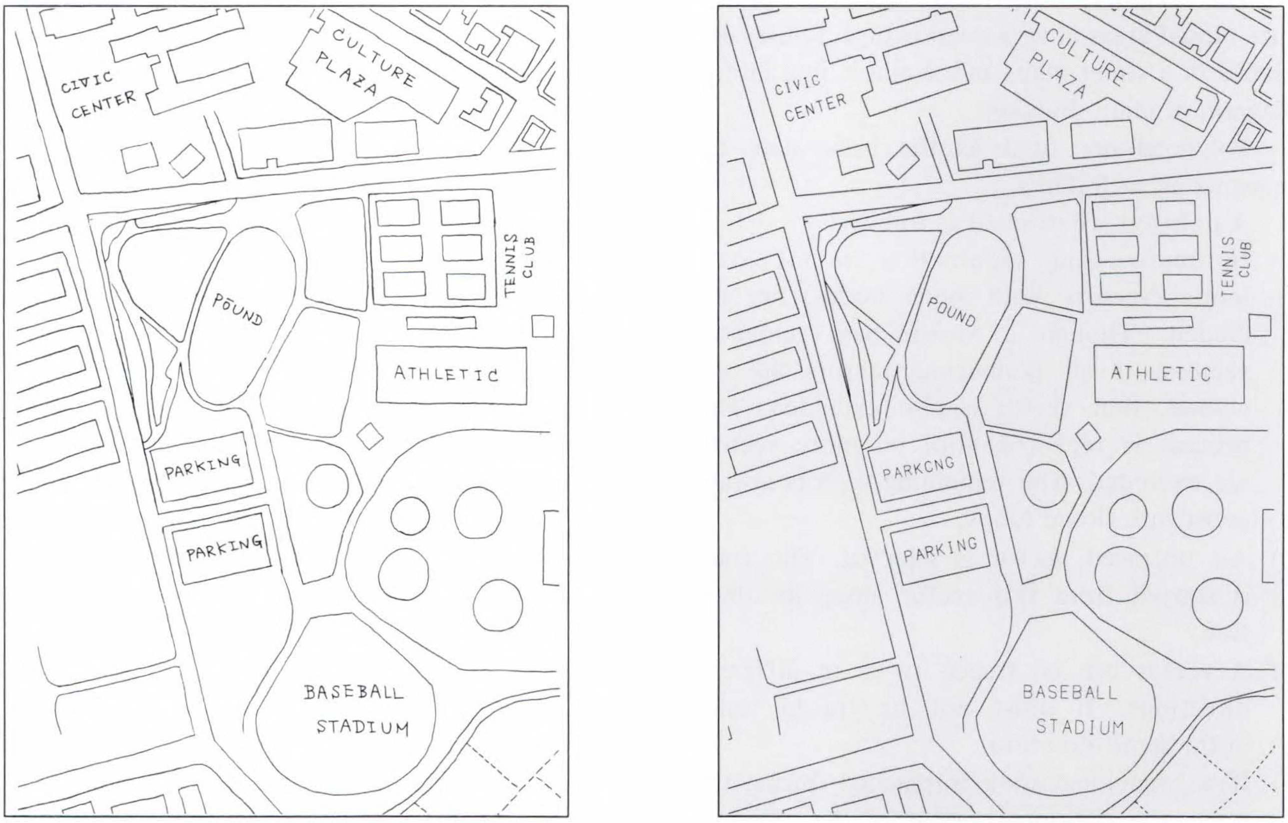


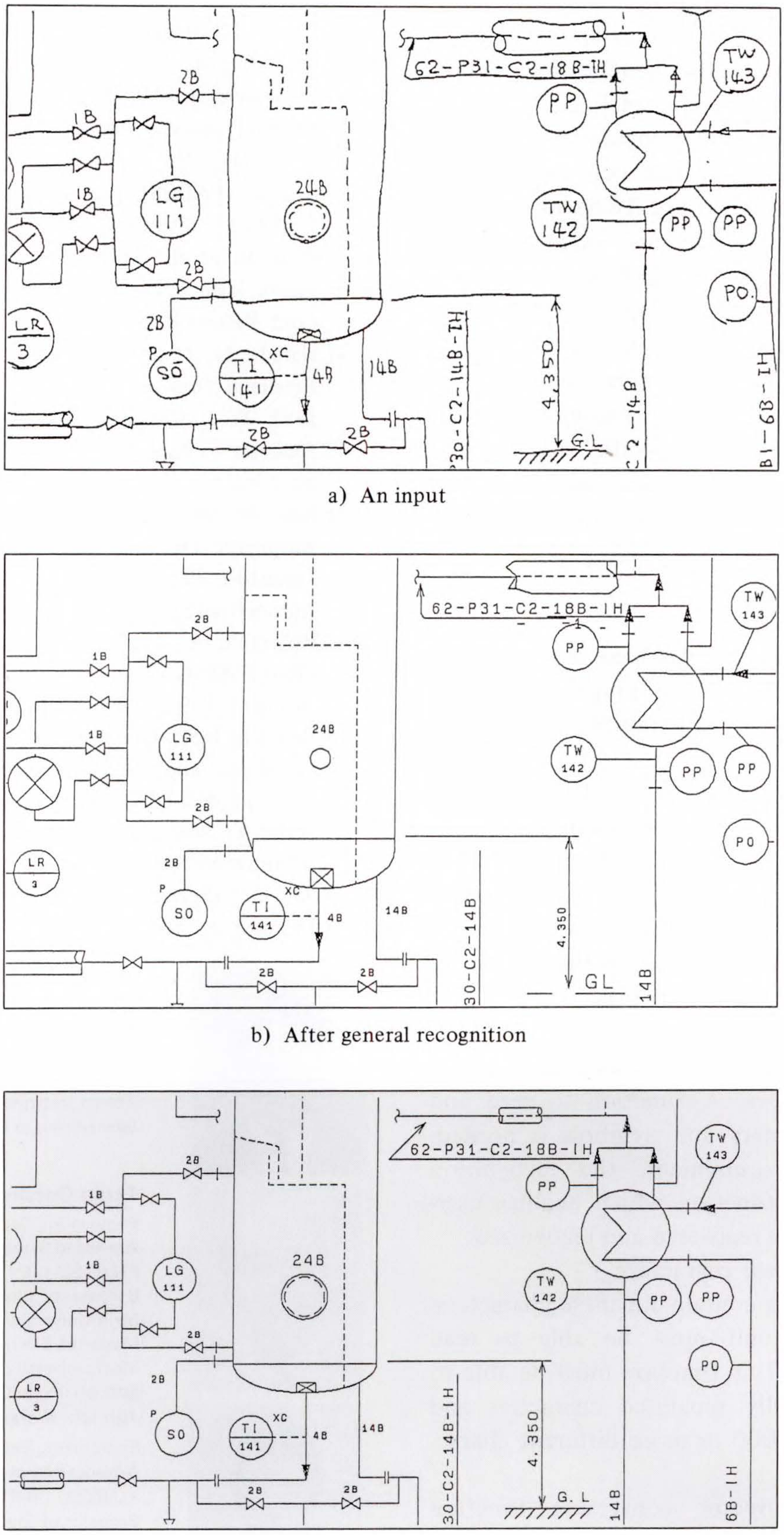
Fig. 17—Features extraction.



a) An input drawing

b) A processed output

Fig. 18—A test drawing.



a) An input

b) After general recognition

c) After drawing-specific processing

Fig. 19—A piping and instrument (P&I) diagram.

5. Examples

This chapter shows two examples. One is a test drawing to show general recognition ability (see Fig. 18). Here, inclined character strings are extracted and recognized. All strings are correctly extracted, but one character is incorrectly recognized in Fig. 18 b). In addition, circles and circular arcs are extracted, and closed loops are extracted as polygons.

The other example is an actual drawing of the results of a P&I diagram (see Fig. 19). This example shows the input, the result of general recognition by the recognition unit, and result of drawing-specific recognition. Specific symbol recognition and other processes were executed by drawing-specific processing with the host computer.

6. Conclusion

The authors have developed a recognition unit which enables drawings prepared by hand to be automatically and rapidly input to a computer system.

For a piping and instrument diagram on A1-size paper (594 mm x 841 mm), conversion to vectors took only 54.2 s. The number of resulting vectors was 16 860.

The performance and functions of the system will be improved to enable the automatic input of various types of drawings.

1) Reading user-defined fonts

Some drawings contain characters written in special typefaces. A function to read and code such characters and symbols is needed. To meet this requirement, the recognition unit will have a function which enables user-defined fonts to be registered and recognized.

2) Reading Japanese characters

Some drawings contain Japanese characters. The recognition unit must be able to read these characters. This function must be able to recognize physically separated characters and be able to read 3 000 or more different characters at high-speed.

The raster-to-vector conversion function developed with this unit will contribute to the development of personal computers that can handle images and vectors freely.

References

1) Endoh, H., Morimoto, M., and Harada, H.: FACOM 2510 Fujitsu Advanced Drawing Capture System. (in Japanese), *FUJITSU*, **38**, 5 (Special Issue: Information Processing Devices), pp. 395-403 (1987).

2) Kabuyama, Y., and Koizumi, J.: Drawing Input System (FADCS). (in Japanese), Inf. Proc. Soc., Jpn., Proc., 35th Conf., 1987, pp. 2179-2180.

3) Ishii, M. et al.: An experimental system of hand-drawn logic circuit diagrams. IEEE Proc. 6th Int. Conf. Pattern Recognition. 1982, pp. 717-719.

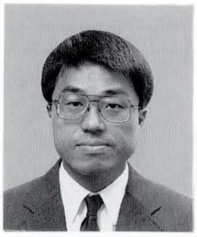
4) Harada, H., Itoh, Y., and Ishii, M.: Recognition of Freehand Drawing in Chemical Plant Engineering. IEEE Proc. Workshop Comput. Architec. Pattern Analysis Image Database Manage., 1985, pp. 146-153.

5) Kase, N., Shiratori, Y., and Takanashi, E.: Practical Automatic Drawing Input System for P&ID. (in Japanese), *FUJITSU*, **40**, 5 (Special Issue: Branch Automation System), pp. 288-293 (1989).

6) Rosenfeld, A., and Kak, A.C.: DIGITAL PICTURE PROCESSING. 2nd ed., Vols. 2, New York, Academic Press, 1982.

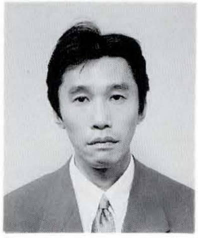
7) Deutsch, E.S.: Thinning algorithms on rectangular, hexagonal, and triangular arrays. *Commun. ACM.*, **15**, 9, pp. 827-837 (1972).

8) Izaki, Y., Isobe, Y., and Takakura, M.: Versatile Information Reader: VIR. (in Japanese), *FUJITSU*, **38**, 5, (Special Issue: Information Processing Devices), pp. 404-407 (1987).



Yukikazu Kabuyama
Engineering Dept.
Advanced System Products Div.
FUJITSU LIMITED
Bachelor of Electrical Eng.
Tokyo Institute of Technology 1975
Specializing in Pattern Recognition

Teruo Ootake
Engineering Dept.
Advanced System Products Div.
FUJITSU LIMITED
Bachelor of Electrical Eng.
Meiji University 1975
Master of Electrical Eng.
Meiji University 1977
Specializing in Circuit Design



Jun-ichi Koizumi
Engineering Dept.
Advanced System Products Div.
FUJITSU LIMITED
Bachelor of Electricity Eng.
Hokkaido University 1982
Specializing in Pattern Recognition

UDC 008.2:681.3.02

Toward Development of a Civic Infrastructure for the Information Society

• Hajime Enomoto

(Manuscript received June 4, 1990)

This paper describes a new civic infrastructure for the information society. It considers several features of culture and civilization, for example, fire as a "magic tool" in ancient times. It also discusses the effect of information networks on communities, and the basics of the acquisition and representation of information. This paper presents the observation that the recursive nature of the use of information in ancient times has parallels in the modern age. For a computer network to have an effect on an information structure, sensory information must be changed to logic information by humans. This means that concepts of confrontation and integration become useful tools for the analysis and synthesis of information, and total cooperation mechanisms will be enhanced by using essential concepts and functions available in a proposed civic infrastructure.

1. Introduction

Today's information society must usher in a new cultural phase and expand the horizons of human mental activity. In this regard, the information society has yet to develop completely with a new civic infrastructure that supports an information culture, and we still face many problems.

Therefore, we should examine the specific research and development problems posted in the basic research on information science from the standpoint of developing a new civic infrastructure.

This paper discusses such problems by considering the essence of a fundamental society that supports a new civic infrastructure as a network with emphasis on an information culture.

2. Culture and civilization

2.1 Mutual dependency of culture and civilization

Culture is a product of human mental activity, and is learned and handed down from generation to generation mainly through the mental association that people have with culture. Although **civilization** is also a human product, it is usually contrasted with culture as a physical product. Civilization may be considered a state in which the technology for human activity is developed with an infrastructure for associating the significance of human activity to existing spiritual culture and moving in new directions.

As far as an organization is concerned, a sense of value common to its members, as well as decision-making and action patterns in the organization, reflect the culture of the organization. These considerations become major factors in stimulating the organization to achieve social recognition through a culture that best suits the organization. This is because such organizational culture implicitly determines the direction and cooperative efforts of the organization.

By considering the **organizational civilization** to be an infrastructure system that supports culture as a mental activity that stimulates organizational activity, we can understand how it determines the direction of the organization's interests and increases organizational efficiency. Consequently, the development of a new civic infrastructure for today's information society will certainly become a vital concern in the future.

The basic mental elements of organizational culture are:

1) Establishing and publicizing operating concepts and visionary insight
2) Evaluating something new and having enterprising spirit for it
3) Expanding potential
4) Compelling cooperative support
5) Smoothing and mutually respecting external relationships.

The methods of achieving the above are considered major functions.

With regard to the above, the basic thought paradigms for developing culture as a mental product vary with the changing spirit of the times. At the same time, however, there are no major changes in the range of intellect intrinsic to humans. Evolutionary changes have occurred in individual physical elements of the civic infrastructure. Such physical elements give rise to industrial products, thus forming the civic infrastructure of such modern industries as transportation, communication, distribution, construction, and public works. Television culture, for example, concomitant with electronic civilization is relatively new to civilization, and has contributed significantly to society. Vast stores of knowledge and information related directly to civilization have been provided through television culture.

The rapid development of computers has also had a major impact on our modern civilization. Microcomputers are now essential components of all types of equipment, and such huge networks as the third-generation online banking system include individual subsystems offering services geared toward the total integration of functions. Although a **software crisis** has been reported, this problem can be solved through human interfacing with workstations.

In other words, we must anticipate a newly integrated electronic civic infrastructure mainly consisting of new service functions to accommodate the existing mental culture. We must also continue research on a civic infrastructure to create new culture as part of today's information society and develop a new civic infrastructure system for the next generation.

These needs dictate the essence of **basic research on information science** to be done from the **standpoint of human science.** The time has come to recognize research results as a scientific discipline to be effectively applied to create a better society.

2.2 Magic tool – fire – and pathway to primitive civilization

The **discovery of fire** had a profound impact on mankind's development. This discovery is even considered to have affected the jaw bone structure of modern humankind, as well as the development of a primitive "word" to describe the use of fire. Fire accidentally discovered as the high exothermic phenomenon of fire. By experimenting with the uses of fire, human was able to develop ways and tools to start fire for practical and universal application.

Through such development and understanding the words used to express fire, primitive societies were formed based on the usefulness of fire, while knowledge about fire was widely spread on a network basis. Through this process, such fire control methods as starting and extinguishing fire, as well as tools for fire control, were eventually developed. Methods of cooking using fire, tableware, and cooking utensils made of stone implements and primitive pottery, and vessels for storing food and water were also developed. Such significant developments helped create a "fire-based" civilization, improved such housing conditions as lighting and heating, and significantly expanded the areas of human habitation were primitive civilization first developed.

In this stage, a functional language using "words" to describe the physical sensation of the phenomenon of fire itself, tools, and a means to refer to fire use by pointing to actual objects had also developed systematically in primitive societies. **A community network** was rapidly established as a culture or civilization. To refer to an actual object, the object was transported and shown directly to others, or was indicated on a chart by examining its shape in place of the actual object. Wheels for transporting such objects were invented to to more effectively use resources in the communities and, consequently, the network range expanded. These early communities rapidly expanded to enhance communication between other communities.

Ancient civilization originated in Mesopotamia about 7 000 B.C., with agriculture, advanced construction, sailing ships, and other means of transport being developed. An early calendar and religion were also developed, while cities were built as entities to promote intercultural exchange and cooperation. The convenient systems of the cities were extended to newly constructed networks. In ancient times, the physical products of civilization also interested others in terms of information. The famed "Silk Road" also served to transport and extend new culture and technology between distant lands through greater knowledge. A civic infrastructure gradually became homogeneous between many communities in an interest or profit-motivated form. However, diversity based on such abstract representations as art eventually resulted. Such diversity greatly depended on the environment and tended to create regional art forms. This may be because knowledge related to actual objects was only spread as information without a means of representation on an abstract level.

2.3 Acquisition and representation of information

Hitherto, how ancient civilization developed based on the discovery of fire has been discussed. The representation and transfer of information also play an important role in ancient civilization. Methods of starting fire could have been expressed in a form in which the process or procedure and restrictions on starting fire were integrated. The sensory learning about tools to start fire could also have had a greater effect than logical representation through language.

One **prerequisite of the firing phenomenon** is the use of heat caused by friction – rubbing stones to generate heat –. To generate such friction heat, one may rub a hard stick against a concave board to generate a spark, then set fire to combustible dry leaves, which have a low firing point. Then fan the flames for burning the leaves. The use of frictional heat may have been prescribed with regard to the procedure on using such simple tools and purposes.

The **combustion conditions** are affected by how the fire is to be used. For example, to use fire for cooking, use a combustible object that burns to set fire to the combustibles in an "oven" and restrict the combustion conditions of the combustibles and the amount of air. To use fire for heating, increase or decrease the amount of combustion according to the temperature and purpose, such as lighting.

Conditions for terminating combustion were also required. One condition is that when the combustibles are used up, the combustions conditions are lost. The other condition is that the temperature required to maintain combustion is lowered by pouring water on fire, which will eventually extinguish the fire. These conditions were learned.

Although the above conditions were essential for the "fire-based" civilization, they were not established as systematic knowledge until practical information is used to express the use of fire. If we consider that how such information was acquired and learned, and how the knowledge as their results was handed down in ancient times, the practical information and experiences will be most important for synthesis of knowledge. Many people are aware that such information and knowledge pose problems related directly to human life. Information is spread under interest-motivated conditions according to a desire to experience such multimedia expressions as language, speech, and actual conditions. Through this transfer process, information becomes beneficial to many people, and a desire to provide independent useful information, and a desire to provide independent information transfer services may have existed. Such services should be provided with cooperative spirit of promoting the rapid and wide use of fire.

To benefit from practical experiences in the above process, concepts must be learned in various situations where all items are different. To be conscious of an object that is experienced, an effort must be made to represent knowledge through language and must be obtained experientially as sensory information to complement it. For example, the word concepts of the combustibles and noncombustibles are divided as extensive information for the use of fire. Dry and dead trees burn easily, while green leaves and live trees do not. Such concepts are specifically applied to such environmental objects as data, and lead to the construction of such tools as ovens using fire with noncombustibles.

In relation to the methods of starting fire, the methods of making and using many tools are taught. The method of generating heat using frictional heat is sensibly taught and the concept of firing temperature is also taught as sensory information. Both are learned concretely. Children playing with fire are related to an inherent human interest in the firing process.

A "fire-based" civilization is established and eventually grows when social communities participate in the starting and uses of fire. For that purpose, sensory information obtained from learning through experience is encoded in "language" or "images" (multimedia visual descriptions used to express the required information).

For all members of a society to fully understand and enjoy a civilization that developed from the discovery of fire through the above descriptions of fire-related information, people must first acquire prerequisite knowledge. If the use of high temperature following the combustion phenomenon of fire, and the starting, setting, combustion, and extinguishing processes and their conditions are recognized beforehand, fire can be used effectively. For a flame, however, an oxidizing flame was distinguished from an reducing flame by the amount of supplied air and calcination object in later ages. An oxidizing flame was first used as a flame, and high temperature following the phenomenon of fire and its flame condition was only sensibly and partially understood. However, once fire is comprehended as knowledge in relation to the tools used for fire, the relationship between fire and the environment is abstractly expressed and the relational knowledge about fire gives people the ability to use fire for expanded applications.

The abstract phenomenon of fire can be expressed with only two properties: fire burns in red and is very hot. The combustion conditions require the following environmental representations about fire: the presence of combustibles, high temperature, and the required amount of air.

There are various abstract representations about the use of fire:

1) if food is cooked by fire
   i) disinfection kills bacteria to prevent sickness,
   ii) food softens, and
   iii) pottery made by fire retains its shape, hardens, has durable quality, and is easy to use as containers.

To use fire as an effective resource requires

2) maintenance of a small fire,
3) setting other combustibles on fire from the small fire, and
4) the location of small fire.

Fire-based civilization had been implemented with the above knowledge as a fire-based culture, and various tools are said to have been developed as an infrastructure to support that culture. The profound impact of fire-based culture and civilization is considered to have led to the sanctification and appreciation of fire through festive celebration. This sanctification and appreciation of fire may have originated in a restless world without fire that wished to use fire forever. From this process, some people began providing fire-related services as their work, and the concept of a kind of special service about fire began to take its rise unconsciously in the society. Community culture came into existence in mutual usage of services.

2.4 Computer and civilization

Computer-related culture and civilization are used in a sense that is different from the above, and have been used as special terms strongly related to programs. For example, when practical use of computer started, FORTRAN and COBOL were mainly used as computer languages. In the time when the simple data and control structures such as arrays, IF-THEN, or DO were mainly used, FORTRAN or COBOL culture was used as a technique of using a specific computer language. The culture word was expanded in TSS (Time Sharing System) civilization, and programming was considered as an art. The culture is considered to have been no better than derivative culture that is established only by a specific or local understanding rather than essential culture.

In this sense, these culture functions are determined by the range of fixed processing capabilities. It is difficult to say that the conventional computer civilization had a significant influence on mental activity in the above cultural sense from the viewpoint of basic mental elements 1) to 5) of organization culture (see section 2.1).

Although Neumann-type computers have been mainly used, an advance in technology has achieved high speed and large capacity, rapidly lowered the costs of computer, and enabled large software to be operated on computers for the network. Many people desire and expect advances in software technology to soon contribute significantly to above mental culture elements[1)-5)]. Therefore, basic research on information science should be focused in this direction.

The **concept of the information network** also existed in ancient civilization as described above, and is considered the essence of human civilization. However, this concept has not been clearly understood. To transfer information using words, figures, and images, as well as transport physical objects, the format of the interest-motivated network must be experientially and sensibly learned as basic functions. The concept of multiple services may have also developed spontaneously. This concept is considered to have a format in which information is associated with an object.

A global network is now being formed, and gradually forming the nucleus of the information network and significantly affect society. October 20, 1987 became known as "Black Monday" for the stock market crash and adverse affect on financial software in the same way that the oil crisis adversely affected industrialized nations. The information network has also played a major role in transportation, traffic, and distribution networks, gradually had a major social impact, and became the nucleus of system operations in society.

A **global network** is now being formed, and electronic mail is being used worldwide. Considering the flows of acquiring effective organizational strategy, planning, design, developing better systems, and providing services, a universal model for all people concerned to use and modify on an interactive basis must be created through the steps of these flows. An exchange of model-based views (instance function, customized template-type data structure) about several areas of interest, as well as the operating descriptive language, are also indispensable.

To detail the execution process for analysis, synthesis, and improvement of essential concepts of the information network based on this **operating descriptive language**, a concept must first be structured by clarifying the attribute structures of the concept and listing restrictive properties between these attribute values. Attribute values are then assigned by analyzing the dependence of the attributes. Otherwise, values must be determined according to environmental conditions. After going through this process, a software system is developed for specific use in various ways.

Conversely, when considering the **creation of an essential concept**, such partial structures as individual fragmental requirements and inherent restrictive properties are assumed. New concept structures are created so that there is no conflict with the various, actual examples, and applicability is enhanced. This process may be basic to creation of essential concepts.

The attribute structure of concept may be synthesized from the items related to actual matters. These matters create a new concept in the flow of using the concept actually or concept generation process. It may be significant to accumulate actual facts, find essential concept via the human interface using high-speed, large computer in analysis and synthesis processes, and enhance applicability by extending essential concepts.

Ancient information networks are somewhat different from modern computer networks in that in ancient times, the human was the nucleus of actual processing. Ego and the intention are closely connected with contents of processing for objects. The human recalls the contents of necessary operation from individual events that actually occurred successively, and learned the systematic operation process. The human gives sensitive meanings to objects and operations on a multimedia basis as his or her own internal representation, and uses abstract words to give name objects and operation partially. The human represents concepts by explaining and describing the objects and operations, and harmonized descriptions of contents of human sensitive understanding and contents of logical linguistic knowledge. It is conceivable that the human could satisfy the intentions desired by individuals or groups by doing appropriate operation on the natural flow.

Modern information networks use the high-speed processing capabilities of computers effectively, build all processing elements in the system beforehand as software, and use an operating command system to operate the software. The processing elements are built into the computers in a logical language format rather than through human processing based on the above understanding of objects as knowledge. When the processing elements are related to the sensory information that is difficult to describe in software, designing human interfaces is expected to become very important in the future.

In **ancient city civilizations** that developed as fire-based civilizations, it was necessary to develop tools directly related to human life, distribute and exchange food and such tools, and use "words" to describe functions to spread knowledge about food and tools. Communication through "language and words" caused clusters or groups to form and eventually evolved into the efficient design, construction, use of overall city functions by assignment and grouping. Ego-desired prayer may have also been incorporated as a special function. Although all execution is done as a direct human operation, judgment is needed to initiate and end any operation in individual processes. People have acquired the knowledge required to make such judgment through sensory perception. These individual operations and the operation system in which these operations are integrated have become special services, and gradually developed into a large service system. From this standpoint, it may also be necessary to understand the essence of modern social services based on a computer nucleus, and consider integrating computers with people through harmonious interface functions, and eventually replace computers for the part of human functions.

To establish a parallel or concurrent execution system with security provided, **software process engineering** based on a system interface concept to enable modularization and customization of large systems becomes very important.

The five basic elements of the spiritual culture described in section 2.1 are more or less related to communication as part of human nature, and a system of communication best suited to the social environment of the times must be adopted. Learning about, developing and describing production techniques and social structure are usually handed down as assets of human society and have been useful in developing the infrastructure of material-based civilization. To support mental activity using computer technology, "language" must become the nucleus of the knowledge system to be symbolically developed and exchanged as multimedia. A new information system that mainly uses human interfaces and computer networks from creating to using information must also be developed as a civic infrastructure of the future.

The information system must also be socially distributed and cooperative on a service-oriented basis.

3. Civic infrastructure of information

3.1 Economic description of information structures

Special functions are needed to smoothly operate individual modularized and shared functions, while combining and extending the information structures of all objects including computer hardware and software, into an integrate object, to provide the communication function of effective inter-system interfaces.

In ancient civilization, fragments of knowledge were connected unconsciously or consciously by people, and intent was partially satisfied by forming a series of processes. Knowledge could be exchanged between other groups of people through actual operation and providing supplementary descriptions through "language and words".

In modern industrial society, the essence of its nucleus also remains unchanged, and the uses of energy have been determined with equipment manufacturing information being expressed mainly through drawings and representation rules. It is important for people to restrictively interpret such drawings for a natural understanding of the limited capabilities of the mechanical operation of machines. **Software** is used to incorporate human knowledge into computers, and is considered a new product because the range of computer operation done by software is not fully understand by other people. Reading and understanding everything about software have proved beyond the ability of one person[6].

However, such a situation must be improved. Methodology that enables us to effectively describe information structures is needed to achieve more efficient processing, descriptiveness, understanding, and customization. The economic description information structures is a discipline that describes the information structures as economically as possible. This discipline associates sensory information with logical information by using multimedia for understanding the static and dynamic relationships between objects themselves. This discipline also clarifies the **paradigm of the interface function** between abstraction and actual application by metaconcepts thorough information hiding principles. Therefore, future research on this discipline is necessary.

To satisfy this need, research on the essence of the world of beauty must be conducted. Such research must include a study on methodology, such as learning and interface systems that can express color geometry and harmonious structures by using the computer.

The following concepts of confrontation are effective for the economic description of information structures.

3.2 Confrontation and integration in creating and using information[1)]

Recently, vast amounts of information has been created and distributed in a flood of information. Today's information society must absorb such vast amounts of information and actively acquire and effectively use essential information by using new evaluation function with new originality.

However, much of the raw information can only be understood by a few, specially trained people. Information used and recognized as useful by many people must be publicized as news or its internal representation must be structured as knowledge. Many people must also be informed of the universal application of such information. As the quantity and quality of information increase, it becomes necessary to **create a new service system environment.**

The categories of created and distributed information include **sensory information** and **logical information** as described above. Sensory information is related to a sixth sense in addition to the five inherent senses of people. It is further related to the actual life experiences of individuals and social culture. For example, the words "dad" and "mom" are used in a different linguistic sense than the words "father" and "mother" used in conversation.

Sensory information includes something that cannot be completely understood without the actual experience. When **logical information** is logically clear, the relationship between the input and output information structures is clearly defined. When logical information assumes the form of specifications, it must be converted procedurally and processed by the computer through a human interface.

By accepting the above classifications of sensory and logical information, most information in the world would be considered sensory information. People must also make an effort to convert and integrate sensory information into logical information. Examples of such effort include pattern understanding/recognition, natural language processing, and problem solving techniques. Such high-level human functions as meta logic and learning are also considered sensory functions in the same sense.

The information science most people are concerned with is the branch of human engineering. Further analysis reveals that **sensory perception information, knowledge information, and symbol information** are converted partially to logical information as data in this order. Moreover, information can be created, structured, and logically fixed according to specific rules. Through such a process, created information is used in both the above'and reverse orders. In the flow from sensory information to knowledge information, information is structure through analysis related to a knowledge base of information. In the reverse flow, information is created, and the primary processed in both directions are integrated for **analysis and synthesis methods.** The primary processes may be divided into the problems posed by exchanging and processing information, and the problems posed by communication using information as messages. By using the computer to exchange all information, people must describe and convert such information into usable logical information. It will be very difficult to satisfy such requirements in the future in terms of **"the unknown is the human itself".** Even if a problem, for example, is expressed in a sentence and recognized as an abnormal system state, no suitable action can be taken for it. It is difficult to convert everything into logical information as knowledge due to the various relationships that exist between abnormal system states and the external environment. To properly handle such abnormalities, people must interactively use their superior experience and sensory functions. An even greater effort is required to prevent problems from occurring. This indicates that a description of human knowledge or intellect by using the analogy of a problem includes elements of sensory perception and education.

Various items of information may be used in messages for **communication.** If such information ranges from sensory information to symbolic information, the human ability to understand problems from sensory information to knowledge or symbolic information is basic. A set of basic functions of the network system used for message communication provides a service to enable direct or indirect transmission and a clarification of human intent. Such integrated functions enable computer processing using the network communication function. Consequently, many conversion and processing functions between the different categories of information greater important roles. This is because computer services actualize human intent are more or less added to such intellectual communication.

The operation and procedures applied to such integration can sensitively be conceived, represented, and processed from input to output through sensory perception, thus determining whether an **integrated information development environment** of the software system can actually be established. It is necessary to provide powerful tools, efficiently construct and retrieve the completely arranged knowledge base, and provide services in the environment. Thus, mental activity can only be made more productive by satisfying this requirement.

From the above standpoint, information includes data and operation with sensory and logical parts. The part that adjusts data and operation also has sensory and logical parts. The computer can only process logical information, and tools, knowledge bases, and resources in such an environment are also represented as logical knowledge information. A guide to integrated use is provided to extract required items from excess information. It is necessary to originally create and store logical information from items of sensory and symbolic information in the intuitive image state before constructing a system that can be used by people. Thus, it is only possible to accumulate knowledge step-to-step through the interactive process.

When knowledge is distributed, created, and stored in large quantities, a **network** is necessary as the infrastructure for arranging, integrating, and using such knowledge. Of course, networks are constructed in various ways. When networks are interconnected, there is a great need for agents offering integrated network services to respond to ambiguous requests from customers of various professions. And because both software and system specifications change according to specific customer needs, the **system information structure becomes evolutionary**. In the life cycle, some parts are eventually replaced by other new systems, which are ultimately replaced by new, more advanced systems.

To understand the concept of the information life cycle, one must consider the adjustment function of a social system as customer base. The confronting function analyzes the interrelationship between the various needs of individuals and the standard needs of the community and the integrating function among them must be unified as a total system.

When viewing **an idea as the most valuable type of information**, the individuals living in the community are the source of such ideas. A community full of energy and activity is necessary for accumulating and integrating ideas into large systematic information. Systematic information is initially used separately, then its usefulness is gradually recognized in the community. It is very beneficial when users voluntarily provide services for the **agent function** because such usefulness is recognized and the systematization process is established. Therefore, the concepts of confrontation and integration for various services between a customer and agent must be recognized.

In the same sense, individual sensory perception and logic are universal. High-level communication may be important uncomprehending this confrontation and interactively integrating the gap in terms of the community. People and systems integrate a civilized society from different world views. In any social organization, **the cultural products of human internal activity**, such as mental information based on logical descriptions mainly using language and sensory information based on perception, assume priority over **the technical and physical products** of the civilization. Such cultural products are considered resources that contribute to civilization. Any discussion on civilization must realize that the concepts of confrontation and integration contribute to achieving the highest level of organizational culture possible.

From the same standpoint, **achieving civilized resources** directly and indirectly for society **in cooperation with many people and systems** is very important. If **requested functions are arranged and converted** among people and systems through an intercultural interface, then established and upgraded as a system of civilization, they can be come civic products.

A communication network such as telephones and data transmission, a transportation network operating train and airline services, and a financial settlement network of banks have all distinct cultures. To book a train seat or lodging through a transportation system through travel agency network and have the charge is settled through the bank network via the communication network, and an internetwork interface function must be provided.

To travel aboard a train or airplane, a ticket is now required as proof of purchase for the train or airplane rise. The ticketing system must be based on an intercultural interface function for universal application by worldwide passengers who must possess the tickets.

In view of such current practices of possession, and adjustability at the ticket gate, ticketing machines with a simple automatic check function are commonly used.

Possible solutions to this problem include an IC card that is purchased beforehand and a magnetic print function at user terminals. This card can be also used for hotels and theaters.

Documents written in different languages can be accurately translated, if the characteristics and purpose of the documents and the environment related to creation of the documents are considered. This also applies to the translation of emotional sensory information. When software specifications are described, for example, in a brief sentence structure with templates linked directly to the meaning of the words used, the document need not be translated or can be easily translated. Business documents or technical journals may be understood by the computer, as well as above software documents$^{6)}$.

The concepts of confrontation and integration also exist in this regard. Using translation as an example, the **model** related to such descriptions as the sentence description paradigm **is built into an intuitive image state in response to customer needs on a top-down basis. Certain rules, forms, and restrictions** about languages are assumed and **analyzed on a bottom-up basis.** By using the confronting interface between top-down and bottom-up, they are integrated to determine the specifications of translations system. This reveals the importance of human engineering research, and may indicate how the methodology used in knowledge sciences research shoud be applied in the future.

3.3 Confrontation and integration from the environmental standpoint

Individuals rely on their environment to create information. Because individuals interpret information, they often overlook their dependency on the environment or consider much of the information about any number of environments as knowledge when interpreting and understanding information. Thus, there is the risk of partially compromising knowledge. One important aspect of human society is to integrate the sound and presumed portions of knowledge and **complete the overall knowledge for use**. There is a confrontation of sound and presumed portions of knowledge, and the purpose of the integration function is to interact both knowledge portions.

Common knowledge dictates that "sending" and "receiving" are mutually exclusive with regard to time and have sometimes an interaction. The system of transmission assumes the form of a service system in various environments. Transmission service functions when the system determines the relationship between attributes *destination*, *delivery intention*, *transmission function*, *transmission condition*, *transmission intention matching*, and *reception intention*, starting with the **send and receive relationships.**

If transmission does not function, there is an error somewhere in the system. Thus, it is necessary to inform the sender, and receiver if required, of the error and request instructions on how to handle the transmission service.

The **transmission service exists as an environment** for "sending" and "receiving". If the environmental service cannot be utilized, the total transmission process must be also described by the sender and receiver, and implemented as an actual procedure. Several rules about transmission exist, and are implemented as an actual procedure. The rule about "containers" used to hold the contents being transferred area packed as dictated by the **protocol** used between the sender and transmission system. Transmission service must not be concerned with the container contents except for special contents. They are considered environmental conditions that encompass such social culture problems as personal privacy. When viewed by the subscriber of the service, the restriction on the service system environment does not cover the container contents as reflected in the **transparency** of the service, and results in various serviceability for more sending convenience. When considering the above items as a general-purpose service, an **interface** is considered the intermediary feature rule and on method of interacting between the subscriber and service system. This completes transmission for use.

The categories of created and distributed information include **sensory information** and **logical information. Sensory information** is directly linked to the five human senses, and ranges from emotional information to spoken and visual information. Only some of this information can be acquired as knowledge. In the same way, many environments often rely on sensory information. When information is collected using a sensor, the sensor is often an extension of the five senses. Information obtained by sensors is, however, difficult to get totally as many-sided points as the five human senses.

The computer can only process **logical information** that is described in a procedure. An environment is also related to logical problems. When a problem, for example, is logically defined and parameters about the problem are given, the procedure for solving the problem can vary depending on the computer. Thus, the computer also represents an environment.

When considering an environment from the above standpoint, specific environmental contents may have a hierarchical structure. Such contents should **high information** in terms of their specifications. If they have a parallel structure and one structure interacts with another, the contents can be described under a **consistent integrity condition** as the method of defining such interaction properties as synchronization and cooperation[6]. This means that there is independence between defining a problem or specifications and the program algorithm and resources for processing or control. Therefore, a large amount of information remains hidden. A descriptive gap exists in such a relationship. In other words, there is originality.

With the above in mind, there are many environments in which information about objects is collected and understood, and where services are performed through operation. Therefore, it is essential to distinguish between knowledge that depends on the environment structure and knowledge independent of the environment on a confrontation basis.

Such a distinction, however, is generally difficult to make since knowledge is incomplete. Therefore, an effort must be made to complete knowledge in any given environment. This is related to the life cycle.

In view of the above examples of confrontation, the technique for declaring and defining their properties, classification[1], and separate attributes must be considered.

3.4 Civic infrastructure from the viewpoint of universal and valid properties

To root organized civilization as a functioning organization, a universal and valid sense of value must be substantiated. From the standpoint of universal and valid properties, it has been said since old times that "Truth", "Right", and "Beauty" represent recognition, morality, and aesthetics. These elements must be discussed from the viewpoint of an organized civilization.

"Truth" is established as a civic infrastructure when individual members recognize that the object can efficiently stimulate social activity. "Right" may be related to the mutual interface used to enhance the relationship between the members. "Beauty" is related to various elegant representations of individuals and the total organization.

If a civilization with an information network infrastructure mainly drives with information, its major component may be assumed to be the workstation supported by a large mainframe. Considering that the work environment serves to stimulate human mental activity, "Truth" property may be established by powerful workstation consisting of the **good window system and editor**, and proper templates corresponding to the concept of operations or data and tools exist. The meaningful information will be generated by cross-referencing many templates. "Right" means that the workstation must use the multiwindow communication function, include an effective interface using the **network function** without need to know where resources are located, and must ensure adjustable cooperation between organization members by supporting the environment. "Beauty" is important in terms of workstation capabilities to **represent information elegantly** and **friendly** for performing various activties pleasantly and happily. This means that a multimedia workstation that can efficiently handle such multimedia information as graphs, images and voice must be developed based on human behavioral science$^{2)}$.

The development of multimedia products now takes much time and money. A **graphics workstation** can be developed very easily, and is expected to become common in the future. The **needs related to commodities featuring beauty** may also become a general trend.

The time when civilized society determine cultural superiority or inferiority may soon be at hand. Information will become effective in proper and valid forms: a civic infrastructure will be required for properly training and educating individuals to reveal their intentions and bring their abilities into full play. A business providing networks that function as civic infrastructures is also expected to develop. Thus, the network may become the **only function used to maintain social unity**.

To prepare for the above anticipated situation, a consensus is required. Great effort may be necessary to steadily accumulate the basic elements that comprise civilization according to such consensus. Various services must function socially, and individual unit services must be directly linked to customer interests to better understand customer intentions and to improve service specifications. It is necessary that the system paradigm provides such services that encompass the concepts of "Truth", "Right", and "Beauty".

Execution service is provided in the "request/response" format, and integrates customer requests and requirements with data requested by the service. It is first necessary to convert individual requirements into individual integrity conditions requested by customers to match the data being requested. Individual requirements are provided for each item of data requested by using templates in the multiwindow format.

The templates are filled in to describe mutual restrictions, and are incorporated into the system as specifications. This paradigm is as basic as the man-machine interface, and execution service must be developed based on these specifications$^{2)-6)}$.

"Respond" provides the requester with the results of the execution service. If the consulting results, such as a plan, respond must represent the intent of the plan and its contents as the flow and conditions of the plan, and must present the results for a simple typical example in the multiwindow format. For cleaning a concrete item, for example, respond displays the item in charge and service result as data. For transportation, respond can report the location of an item immediately and confirm that the requester has received the item. Because this service has the properties of "Right" and "Beauty" provided by the service in the "request/respond" process appear in the application of a superior social civic infrastructure.

It is important to handle the above service elements effectively and balanced with cordiality. Proper omission may also be necessary. This also means that dynamic customization should be considered by specifying and displaying necessary data with the multiwindow header on the workstation.

By considering an environment as a service resource, the workstation network may become vital for providing information services, and integrated services between networks may become the nucleus of such a service system.

4. Conclusion

In view of the growing popularity of workstations, a civic infrastructure may be developed by using electronic mail, multiwindow display, editors, and network files in the near future. Consequently, we must conduct research on possible civic infrastructures, extract the essential parts and positive effects, and begin developing a prototype civic infrastructure general toward providing services. There is a great possibility that such an infrastructure will have a profound impact on future human life.

For this reason, we should embark upon a program of intensive basic research on and emphasizing education in information science as soon as possible.

References

1) Enomoto, H.: Analysis of Information by Concepts of Confrontation and Integration. (in Japanese), Res. Rep. Basic Res. Inform. Sci., Inform. Process. Soc., Jpn., (1988).

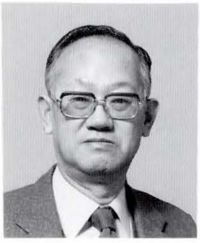
2) Enomoto, H.: Vector Representation Scheme of High Quality Picture. Proc. 3rd. Int., Workshop HDTV, (1989).

3) Enomoto, H.: Proposal of Service Engineering. (in Japanese), *J. Inst. Electron. Commun. Engrs., Jpn.*, **67**, 5, pp. 487-492 (1984).

4) Enomoto, H.: A LOGOCAL MODEL OF SERVICE – Pathway to service engineering –. (in Japanese), *KDD Tech. J.*, **121**, pp. 303-321 (1984); Enomoto, H.: A Model of Service – Proposal of Service Engineering –. *FUJITSU Sci. Tech. J.*, **22**, 1, pp. 40-76 (1986).

5) Enomoto, H.: Software Engineering Handbook. (in Japanese), Tokyo, OHMSHA, LTD., (1986).

6) Saeki, M., Horai, H., and Enomoto, H.: Software Development Process from Natural Language. Proc. 11th Conf. Software Eng., (1989).



Hajime Enomoto
Technical Advisor of Fujitsu
Professor
Shibaura Institute of Technology
Professor Emeritus
Tokyo Institute of Technology
Specializing in Computer Science

Overseas Offices

Abu Dhabi Office
Suite 802, A1 Masaood Tower,
Seikh Hamdan Street, Abu Dhabi,
U.A.E.
Telephone: (971-2)-333440
FAX: (971-2)-333436

Algiers Office
9, Rue Louis Rougie Chateau Neuf,
E1 Biar, Alger 16030, Algeria
Telephone: (213)-2-78-5542
Telex: 408-67522

Amman Office
P.O. Box 5420, Ammán, Jordan
Telephone: (962)-6-62417
FAX: (962)-6-62417

Bangkok Office
Room 405, 4th Floor, Dusit
Thani Bldg., 1-3, Rama IV.
Bangkok, Thailand
Telephone: (66-2)-236-7930
FAX: (66-2)-238-3666

Beijing Office
Room 2101, Fortune Building
5 Dong San Huan Bei-lu,
Chao Yan District, Beijing,
People's Republic of China
Telephone: (86-1)-501-3261
FAX: (86-1)-501-3260

Bogotá Office
Cra. 13 No. 27-50,
Edificio Centro Internacional
Tequendama, Oficina 326,
Bogotá, D.E. Colombia
Telephone: (57-1)-286-7061
FAX: (57-1)-286-7148

Brussels Office
Avenue Louise 176, Bte 2
1050 Brussels, Belgium
Telephone: (32-2)-648-7622
FAX: (32-2)-648-6876

Hawaii Branch
6660 Hawaii Kai Drive, Honolulu,
H1 96825, U.S.A.
Telephone: (1-808)-395-2314
FAX: (1-808)-396-0059

Indonesia Project Office
16th Floor, Skyline Bldg.,
Jalan M.H. Thamrin No. 9,
Jakarta, Indonesia
Telephone: (62-21)-3105710
FAX: (62-21)-3105983

Jakarta Office
16th Floor, Skyline Bldg.,
Jalan M.H. Thamrin No. 9,
Jakarta, Indonesia
Telephone: (62-21)-333245
FAX: (62-21)-327904

Kuala Lumpur Office
Letter Box No. 47, 22nd Floor,
UBN Tower No. 10, Jalan P.
Ramlee, 50250, Kuala Lumpur,
Malaysia
Telephone: (60-3)-238-4870
FAX: (60-3)-238-4869

London Office
2, Longwalk Road, Stockley Park,
Uxbridge, Middlesex, UB11 1AB,
England
Telephone: (44-81)-573-4444
FAX: (44-81)-573-2643

Munich Office
c/o DV18 Siemens AG, Otto-Hahn-Ring 6, D-8000, München
83, F.R. Germany
Telephone: (49-89)-636-3244
FAX: (49-89)-636-45345

New Delhi Office
Mercantile House, 1st Floor,
15, Kasturba Gandhi Marg
New Delhi-110 001, India
Telephone: (91-11)-331-1311
FAX: (91-11)-332-1321

New York Office
680 Fifth Avenue, New York,
N.Y. 10019, U.S.A.
Telephone: (1-212)-265-5360
FAX: (1-212)-541-9071

Paris Office
Bâtiment Aristote, 17
Rue Olof Palme 94006,
Creteil Cedex, France
Telephone: (33-1)-4-399-0897
FAX: (33-1)-4-399-0700

Shanghai Office
Room 1504 Ruijin Bldg.,
205 Maoming Road South,
Shanghai, People's Republic
of China
Telephone: (86-21)-336-462
FAX: (86-21)-336-480

Taipei Office
Sunglow Bldg., 66, Sung Chiang
Road, Taipei, Taiwan
Telephone: (886-2)-561-7715
FAX: (886-2)-536-7454

Washington, D.C. Office
Republic Place, 1776 Eye Street,
N.W. Washington D.C., U.S.A.
Telephone: (1-202)-331-8750
FAX: (1-202)-331-8797

Overseas Subsidiaries

		Telephone	Facsimile
FKL Dong-Hwa Ltd.	338-13, Daehong-Ri, Sunghwan-Eub, Chunwon-Gun, Chungnam, Republic of Korea	(82-417)-4-3660	(82-417)-63-3671
Fujian Fujitsu Communications Software Ltd.	Wu, Li Ting, Fu Ma Lu, Fuzhou, Fujian Province, People's Republic of China	(86-591)-560070	(86-591)-560022
Fujitsu America, Inc.	3055 Orchard Drive, San Jose, CA 95134-2017, USA	(1-408)-432-1300	(1-408)-432-1318
Fujitsu Australia Ltd.	475 Victoria Ave., Chatswood, NSW 2067, Australia	(61-2)-410-4555	(61-2)-411-8603
Fujitsu Australia Software Technology Pty. Ltd.	1st Floor, Techway House, 18 Rodborough Road, French Forest, N.S.W. 2086, Australia	(61-2)-936-1111	(61-2)-975-2899
Fujitsu Business Communication Systems, Inc.	3190 Mira Loma Ave., Anaheim, CA 92806, USA	(1-714)-630-7721	(1-714)-630-7660
(Sales Headquarter)	2411 West 14th Street, Tempe, AZ 85281, USA	(1-602)-921-5900	(1-602)-921-5999
Fujitsu Canada, Inc.	6280 Northwest Drive, Mississauga, Ontario, Canada L4V 1J7	(1-416)-673-8666	(1-416)-673-8677
Fujitsu Component (Malaysia) Sdn. Bhd.	No. 1, Lorong Satu, Kawasan Perindustrian Parit Raja, 86400 Batu Pahat, Johor, Malaysia	(60-7)-481-392	(60-7)-481-771
Fujitsu Component of America, Inc.	3330 Scott Boulevard, Santa Clara, CA 95054-3197, USA	(1-408)-562-1000	(1-408)-748-7655 or (1-408)-727-0355
Fujitsu Customer Service of America, Inc.	11085 N. Torrey Pines Rd. La Jolla, CA 92037, USA	(1-619)-457-9900	(1-619)-457-9968
Fujitsu Deutschland GmbH	Frankfurter Ring 211, 8000 München 40, F.R. Germany	(49-89)-323-780	(49-89)-323-78100
Fujitsu do Brasil Ltda.	Rua Manoel da Nóbrega, 1280-2 Andar, C.E.P. 04001, São Paulo, SP, Brazil	(55-11)-885-7099	(55-11)-885-9132
Fujitsu España, S.A.	Edificio Torre Europa 5ª, Paseo de la Castellana, 95, 28046 Madrid, Spain	(34-1)-581-8000	(34-1)-581-8300
Fujitsu Europe Ltd.	2, Longwalk Road, Stockley Park, Uxbridge, Middlesex, UB11 1AB, England	(44-81)-573-4444	(44-81)-573-2643
Fujitsu Europe Telecom R&D Centre Limited	2, Longwalk Rd., Stockley Park, Uxbridge, Middlesex, UB11 1AB, England	(44-81)-756-0286	(44-81)-573-2643
Fujitsu Finance (U.K.) PLC	2, Longwalk Road, Stockley Park, Uxbridge, Middlesex, UB11 1AB, England	(44-81)-569-1628	(44-81)-573-2643
Fujitsu France S.A.	Bâtiment Aristote 17, Rue Olof Palme 94006, Creteil Cedex, Paris, France	(33-1)-4-399-4000	(33-1)-4-399-0700
Fujitsu Hong Kong Ltd.	Room 2521, Sun Hung Kai Centre, 30 Harbour Road, Hong Kong	(852)-8915780	(852)-721724 or (852)-8383630
Fujitsu Imaging Systems of America, Inc.	3 Corporate Drive, Commerce Park, Danbury, CT 06810, USA	(1-203)-796-5400	(1-203)-796-5665 or (1-203)-796-5723
Fujitsu International Finance (Netherlands) B.V.	Officia 1, De Boelelaan 7, 1083 HJ Amsterdam, The Netherlands	(31-20)-465996	(31-20)-15123
Fujitsu Italia S.p.A.	Via Melchiorre, Gioia No. 8, 20124 Milano, Italy	(39-2)-657-2741	(39-2)-657-2257
Fujitsu Korea Ltd.	9th Floor, Korean Reinsurance Bldg., 80, Susong-Dong, Chongro-Gu, Seoul Special City, Republic of Korea	(82-2)-739-3281	(82-2)-739-3294
Fujitsu Microelectronics Asia Pte. Ltd.	#06-04 to #06-07, Plaza by the Park No. 51, Bras Basah Road, Singapore 0718, Singapore	(65)-336-1600	(65)-336-1609
Fujitsu Microelectronics, Inc.	3545 North First Street, San Jose, CA 95134-1804, USA	(1-408)-922-9000	(1-408)-432-9044
Fujitsu Microelectronics Ireland Limited	Unit D Greenhills Centre, Greenhills Road, Tallaght, Dublin 24, Ireland	(353-1)-520744	(353-1)-520539
Fujitsu Microelectronics Italia S.r.l.	Centro Direzionale, Milanofiori, Strada No. 4-Palazzo A2, I-20090 Assago-Milano, Italy	(39-2)-824-6170	(39-2)-824-6189
Fujitsu Microelectronics Limited	Hargrave House, Belmont Road, Maidenhead, Berkshire SL6 6NE, U.K.	(44-628)-76100	(44)-628-781484
Fujitsu Microelectronics (Malaysia) Sdn. Bhd.	Pesiaran Kuala Selangor, Seksyen 26, 40000 Shah Alam, Selangor Darul Ehsan, Malaysia	(60-3)-511-1155	(60-3)-511-1227
Fujitsu Microelectronics Pacific Asia Limited	Rooms 616-617, Tower B, New Mandarin Plaza, 14 Science Museum Road, Tsimshatsui East, Kowloon, Hong Kong	(852)-723-0393	(852)-721-6555
Fujitsu Mikroelektronik GmbH	Lyoner Strasse 44-48, Arabella Centre 9. OG/A, D-6000 Frankfurt/Niederrad 71, F.R. Germany	(49-69)-66320	(49-69)-6632122
Fujitsu Network Switching of America, Inc.	4403 Bland Road, Somerset Park, Raleigh, NC 27609, USA	(1-919)-790-2211	(1-919)-790-8376
Fujitsu New Zealand Ltd.	6th Floor, National Insulance House, 119-123 Featherston Street, Wellington, New Zealand	(64-4)-733-420	(64-4)-733-429
Fujitsu Nordic AB	Torggatan 8, S-171 54 Solna, Sweden	(46-8)-764-7690	(46-8)-28-0345
Fujitsu Philippines, Inc.	2nd Floor, United Life Bldg., Pasay Road, Legaspi Village, Makati, Metro Manila, Philippines	(63-2)-85-49-51	(63-2)-817-7576
Fujitsu (Singapore) Pte. Ltd.	200, Cantonment Road, #11-01 South Point, Singapore 0208, Singapore	(65)-224-0159	(65)-225-5075
Fujitsu Systems Business of America, Inc.	2986 Oakmead Village Court, Santa Clara, CA 95051, USA	(1-408)-988-8012	(1-408)-492-1982
Fujitsu Systems of America, Inc.	12670 High Bluff Drive, San Diego, CA 92130-2013, USA	(1-619)-481-4004	(1-619)-481-4093
Fujitsu (Thailand) Co., Ltd.	60/90 (Nava Nakorn Industrial Estate Zone 3) Moo 19, Phaholyothin Road, Thambon Klongnung, Amphur Klongluang, Pathumthani 12120, Thailand	(66-2)-520-2630	(66-2)-529-2581
Fujitsu Vitória Computadores e Serviços Ltda.	Avenida Nossa, Senhora da Penha, 570, 8º-Andar, Paraia do Canto-Vitória-Espirito Santo, Brazil	(55-27)-225-0355	(55-27)-225-0954
Information Switching Technology Pty. Ltd.	Level 32, 200 Queen Street, Melbourne 3000, Australia	(61-3)-670-4755	–
Intellistor, Inc.	2402 Clover Basin Drive, Longmont, Colorado 80503, USA	(1-303)-682-6400	(1-303)-682-6401
Tatung-Fujitsu Co., Ltd.	5 Floor Tatung Bldg., 225, Nanking East Road 3rd Section, Taipei, Taiwan	(886-2)-713-5396	(886-2)-717-4644

(The information here as of October 15, 1990)

FUJITSU LIMITED

6-1, Marunouchi 1-chome, Chiyoda-ku, Tokyo 100, Japan
Phone: National (03) 216-3211 International (Int'l Prefix) 81-3-216-3211 Telex: J22833 Cable: "FUJITSULIMITED TOKYO"