UDC 681.327

Efficient Data Entry System Based on Document-recognition Technology: AutoENTRY

●Kazunori Yamamoto ●Masaki Kondo ●Takayuki Matsui ●Shinichi Eguchi

(Manuscript received October 4, 1999)

This paper describes an efficient data entry system called "AutoENTRY" which uses object-oriented features to construct a system for business use. The system can be used to flexibly and easily design a system to suit the user's needs. The new system incorporates several new technologies developed by Fujitsu for job modeling, adjusting the object-oriented features, and recognizing various objects (e.g., form layouts, characters, items). The new system solves problems regarding the specialization of documents for optical character readers, reduces the enormous cost and time required for developing a new system, and facilitates integrated management of multi-platform systems. This paper outlines the object-oriented and customer-oriented AutoENTRY system.

1. Introduction

This paper describes the AutoENTRY system, which enables flexible and easy design of an imaging-OCR (integrated image processing device with the functions of an optical character reader) system. This system overcomes the following problems with current auto entry systems:

1) Current systems require special document formats.

Current systems place the following restrictions on document formats:

– A frame must be provided for each character to be input.
– There are many restrictions on the spacing between frames.
– A high printing precision is necessary.

Moreover, current systems cannot read commercially available forms and customized forms that are made using a word processor.

2) Current systems are very expensive to build.

The system engineer must create the programs in the field, which makes it expensive and time consuming to build a large-scale system.

3) Integrated management of the system environment is not possible.

There are no packages which can integrate different system environments (e.g., multi-workstation systems composed of two or more workstations and personal computers) and control them uniformly.

This paper describes some new recognition technologies and package architectures that solve these problems.[1)-10)]

2. Outline of the AutoENTRY system

The AutoENTRY system is an imaging-OCR package with the three functions described below (see **Figure 1**). This system has one processing unit for each function, and the composition of each processing unit depends on the user's requirements.

1) Development support function

A business application is constructed using conversational definitions. In addition, an optical character recognition interface is provided for advanced customization.

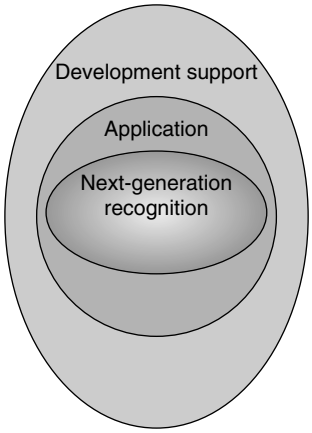

Figure 1
Three functions of AutoENTRY system.

Company A

DATE:19XX/XX/XX
No: 456789

Name of bank	Name of person	Account number	Amount	Commision
Bank A	Amanda	111111	123,000	6,150
Bank B	Ben	222222	45,600	2,280
Bank C	Charles	333333	7,890	395
Bank D	Dan	444444	321,000	16,050
Bank E	Eric	555555	65,400	3,270
Total amount			562,890	28,145
Total commision			28,145	

Figure 2
Example document used in financial institutions.

2) Application function

A function for system construction is provided as standard (control of faxes, image reader, and client server; and transaction management).

3) **Next-generation recognition function** (see **Figure 2**)

Recognition of documents commonly used in financial institutions (recognition of typed and cursive script, figures, alphabets, katakana, and Chinese characters; and knowledge processing).

AutoENTRY consists of two systems: a runtime system and a development system (see **Figure 3**).

These functions are distributed to two or more WSs (workstations or personal computers),

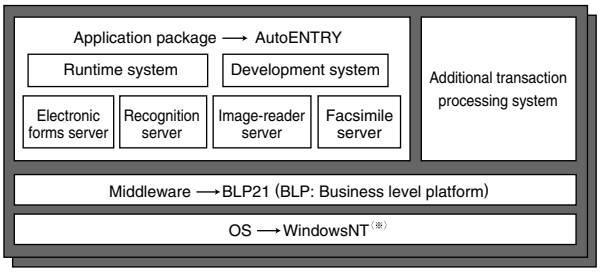

Figure 3
Configuration of software.

and the processing load of individual WSs is distributed to the hardware configuration shown in **Figure 4**. This enables the load to be suitably balanced throughout the entire system according to the system scale. If necessary, however, all of the functions can be implemented on a single WS.

3. Features and technologies of AutoENTRY

3.1 Expanded range of acceptable documents (see Chapter 4)

- **Development system package**

This package makes it easy to create all of the properties (type of data to be treated as a unit of processing, input method, instructions for the next processing, etc.) needed by the runtime system package. The TEIGI-JOHO can be automatically distributed to the runtime system package through the network. The validity of set information can be confirmed in the runtime system package beforehand.

- **Runtime system package**

This package reads a document and then encodes and outputs its contents. It has the following server functions:

– Function for controlling an image-reader
– Function for recognizing the layout composition and text of documents
– Function for correcting the recognition results
– Function for creating a database and managing information about documents previously input.

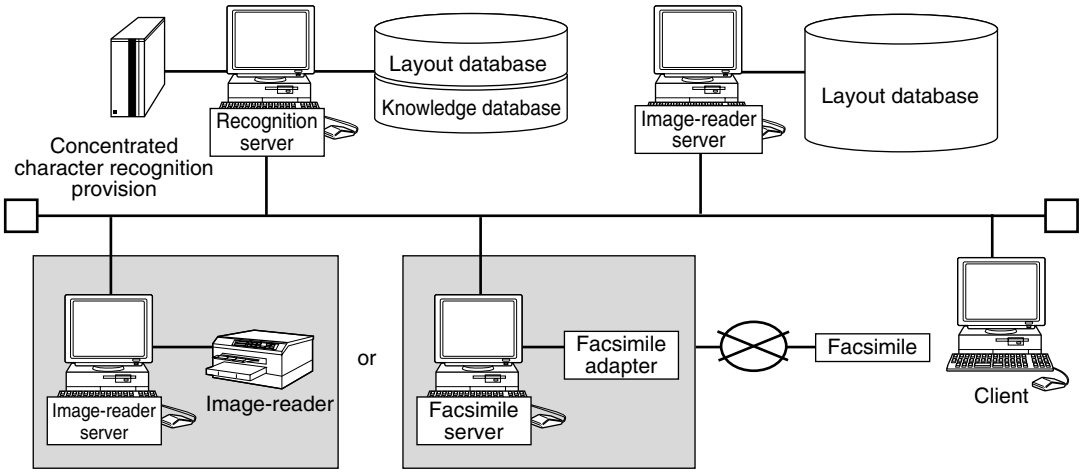

Figure 4
Configuration of hardware.

AutoENTRY features various document layout recognition technologies (i.e., knowledge processing and the technologies for ruled line extraction, item extraction, contact character extraction, and form identification). These technologies enable the system to recognize any document that has been previously input.

1) Easy definition of document layout

First, the layout information that has been automatically extracted is displayed on the screen. Next, the user defines the meanings of the items on the screen through a conversational process using the mouse and keyboard. In addition, the user can specify information unique to the document by using the automatic detection function. These features make it possible to quickly and easily specify a layout.

2) Easy design of transaction processing system (see Chapter 5)

A packaged definition (file definition, workflow definition, screen definition, document definition, etc.) can be set by using a consistent conversational method. A workflow definition can be made by using the Excel[TMnote)] spreadsheet program and the characteristic logic of the business in which the system will be used. These features help users to design a custom-made system for their businesses.

3) Scalable multi-server function

This function enables decentralization from a single WS to two or more WSs and enables the user to flexibly adjust the capacity of the server according to the scale of the user's business. Also, when two or more servers are used, this function enables the processing for large amounts of data to be automatically switched to a free server to achieve high response speeds.

The AutoENTRY system uses four recognition technologies: layout recognition, item recognition, character recognition, and knowledge processing. These technologies enable automatic extraction of information in a document (see **Figure 5**).

4. Recognition technologies

Layout recognition (see **Figure 6**)

This technology extracts the unique layout information of a filled-in form and then compares the extracted information with the layouts stored in the layout database. If the extracted information matches a layout in the database, the information about the matching layout in the database is extracted. If no match is found, the system assumes that the form is a new type of form.

note) Excel is a registered trademark of Microsoft Corporation.

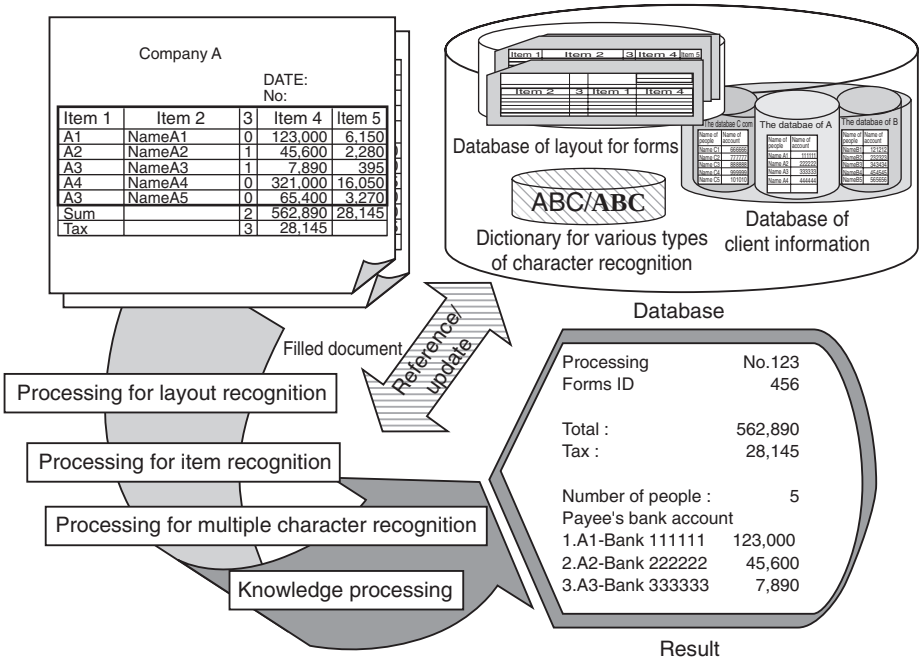

Figure 5
Four recognition technologies used in AutoENTRY system.

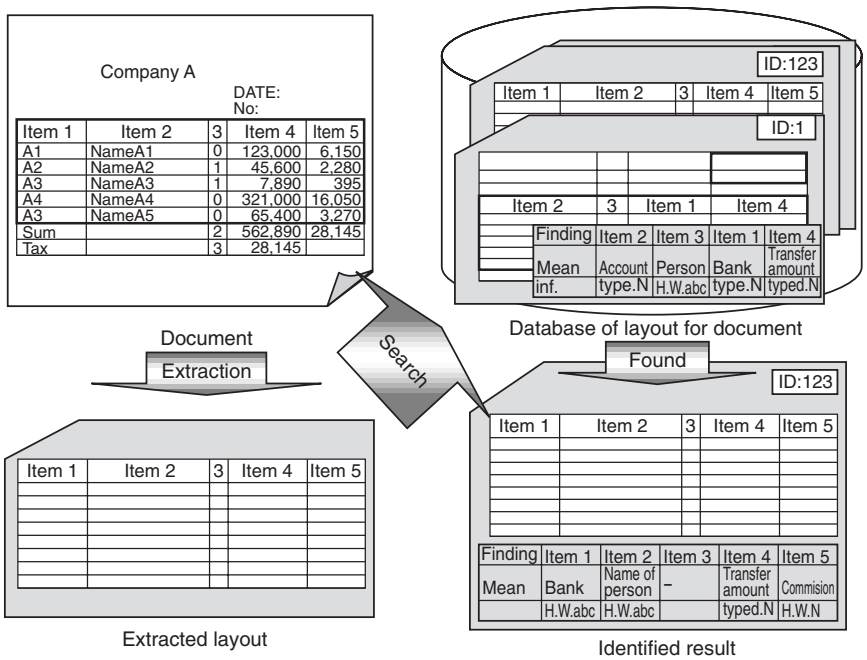

Figure 6
Layout recognition.

Item recognition (see **Figure 7**)

This technology recognizes the columns that the user needs to fill in and removes any differences between the input document and the matching layout data in the database (these differences can be caused by a faulty input device or an input document in poor condition).

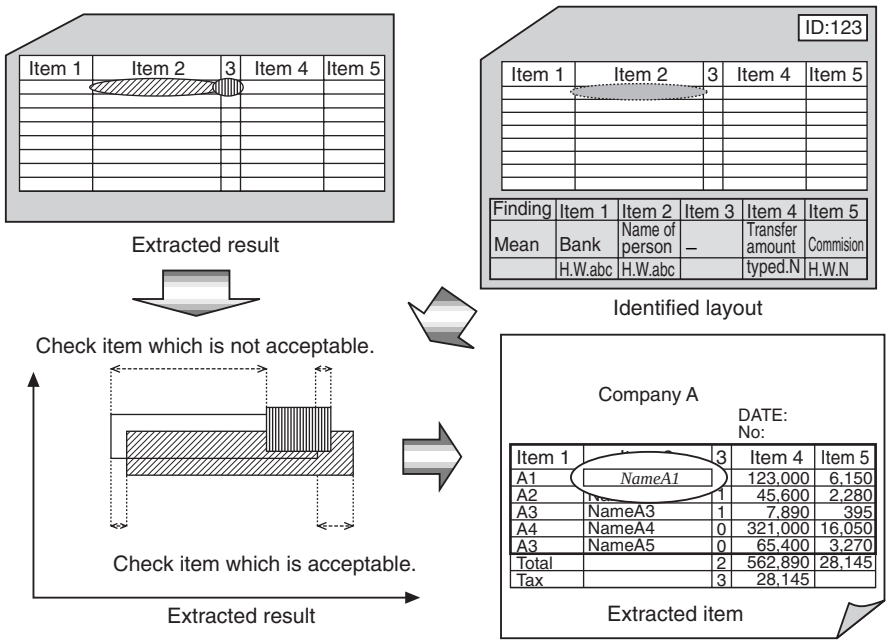

Figure 7
Item recognition.

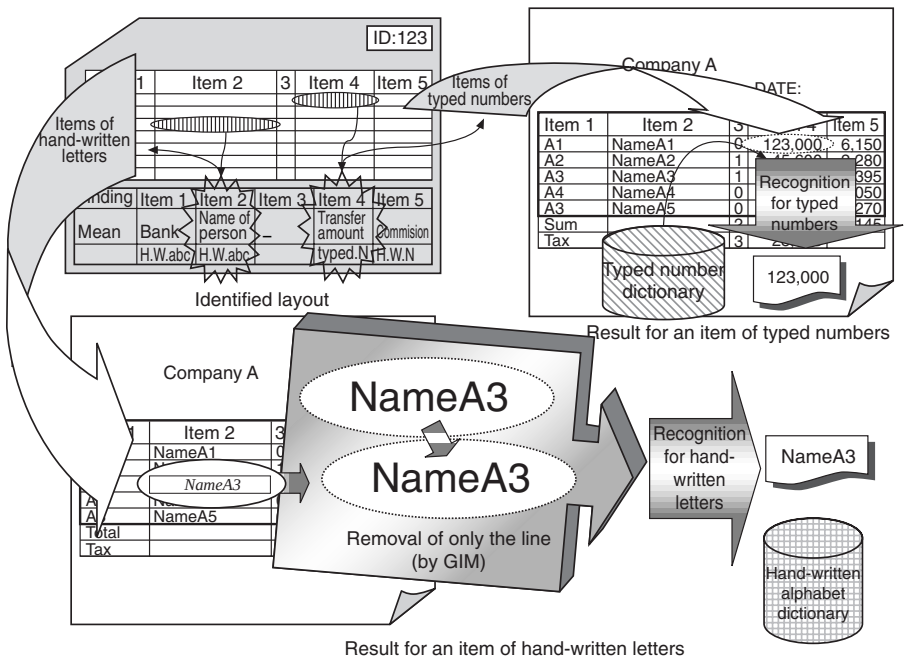

Figure 8
Multi-character recognition.

Multi-character recognition (see **Figure 8**)

This technology recognizes the characters in each item by selecting the best recognition method for the types of characters they contain. By selecting the best method, a higher recognition accuracy can be obtained. The extracted characters are recognized using the global interpolation method, which removes ruled lines if they are in contact with characters.

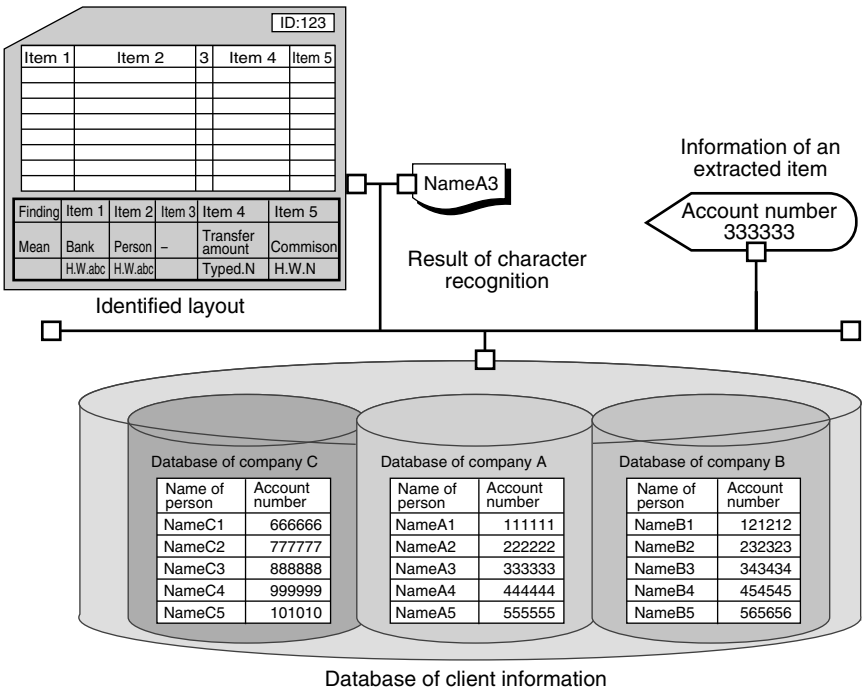

Figure 9
Knowledge processing.

Knowledge processing (see **Figure 9**)

This processing checks the character recognition results by referencing the user data and special word dictionaries. It also obtains any information that is linked with the reference and check functions.

5. Package architecture

The package architecture is an object-oriented architecture that packages individual business models for each common processing. Individual packages are specified by defining their operations using consistent operations. By combining these operations, a business transaction processing system can be constructed. Because AutoENTRY makes it easy to design business transaction processing systems, it solves the following problems:

- **Current problem with system development**

Current systems are constructed by integrating an existing workflow engine and a recognition engine using business-specific applications. However, because the data storage formats differ between engines, each time data is transferred along multiple paths, data conversion is required and therefore the development efficiency is very low. Furthermore, because the business dependency of individual processing is high and usability between different engines is low, it is difficult to use a system in a business other than the one for which it was created. As a result, many person-hours are required for system development.

- **Method of realizing package architectures**

AutoENTRY meets the following two requirements for making a business transaction system packaged.

1) Modeling of applications using object-orientation
2) Business-specific separation/incorporation of processing

Because AutoENTRY meets these two requirements, it can cover a wide range of businesses.

1) Modeling of applications using object-orientation

This promotes reuse of packages. First, the

processing object data called the "object," which is a basic component, must be determined. The processing data is transferred between two or more processing stages on the workflow engine. At each processing stage, how a stage component (called a "method") is operated is defined based on the properties. Using a common interface, a method is provided with the functions necessary for an entry task. The functions cover the processing from reading forms to outputting the processing results.

A design system that can perform a consistent user-specified method must be based on the same business model. Also, such a system must be constructed without corrupting the consistency of the processing method. AutoENTRY meets both of these requirements.

2) Business-specific separation/incorporation of processing

For the processing that is specific to the user's business, the logic is separated from the method and then set outside the method using a worksheet of Microsoft Excel™. Because the worksheet is created using a commonly used spreadsheet program, a correlation check can easily be specified for large amounts of data. Also, if a process cannot be performed using Excel or a higher processing performance is requested, AutoENTRY has a built-in function for logic created using ActiveX.

- **Application to client-server system**

A client-server system can be constructed by allocating the stages to individual WSs in a system. One WS can handle multiple stages, or one stage can be executed by multiple WSs. Intensive processing, for example, recognition processing, is enabled by distributing tasks to two or more WSs depending on the required processing ability. **Figure 10** shows an example construction for business transaction processing using AutoENTRY. This example covers the processes from forms checking and correction to forms approval. Figure 10 also gives an example of applying AutoENTRY to a business model. To make the application efficient, the reading, recognition, and output stages are added. If the same screen method is used for the checking/correction and approval stages, a unique operation for each stage can be specified using properties and then different operations can be executed. When the execution scale is small, a single workstation can handle all the stages (see **Figure 11**). When the execution scale is large, two or more WSs are con-

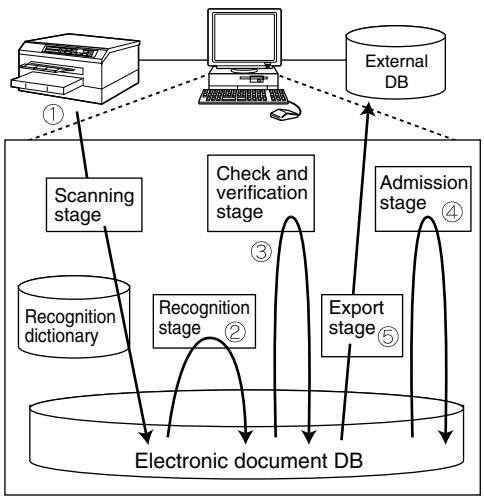

Figure 11
AutoENTRY system using single workstation.

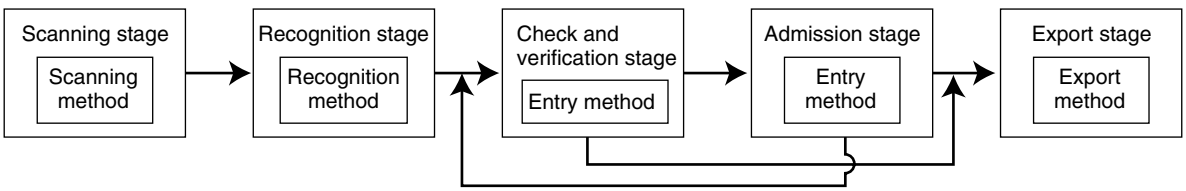

Figure 10
Example construction for business transaction processing using AutoENTRY.

nected in a client-server configuration and the stages are allocated depending on the required ability (see **Figure 12**).

6. Conclusion

The AutoENTRY system enables a user to construct a customized data entry system quickly and easily. Two examples of the effect of introducing the AutoENTRY system are shown below.

System A (Practical system used at a company)

Types of forms	3		
Number of items	170	30	8
Number of screens	4	2	1
Cooperation with external applications	ORACLE cooperation (SQL)		
System scale (number of PCs)	35 (5 servers, 2 readers, 28 clients)		
Processing rate	2000 sheets/day		
Other functions	Function for limiting processing by secret identification code		

Time required to construct system:
With AutoENTRY system ····· 10 person-months
Without AutoENTRY system ····· 200 person-months

System B (Demonstration system)

Types of forms	1
Number of items	65
Number of screens	1
Cooperation with external applications	None (CSV files are output)
System scale (number of PCs)	1
Processing rate	100 sheets/day
Other functions	None

Time required to construct system:
With AutoENTRY system ····· 1 person-day
Without AutoENTRY system ····· 2 person-months

Using an AutoENTRY system already constructed for practical use, 88 out of 100 different types of forms were identified. The recognition result was excellent because there were similar forms that were difficult to identify compared to the case when only simple forms were used and also because conditions in the user's operation environment, for example, the media conditions, were not optimal.

Acknowledgements

The authors wish to thank Mr. Katsumata, the director in charge of system development at the financial delivery system division of Fujitsu Ltd., for his contribution.

References

1) S. Naoi: High-speed Labeling Method Using Adaptive Variable Window Size for Character Shape Feature. IEEE Second Asian Conf. on Computer Vision (ACCV'95), **1**, pp.I-408-I-411 (1995).
2) S. Naoi, M. Yabuki, A. Asakawa, and Y. Hotta: Global Interpolation in the Segmentation of Handwritten Characters Overlapping a Border. *IEICE Trans. INF. & SYST.*, **E78-D**, 7, pp.909-919 (1995).
3) S. Naoi, M. Suwa, and M. Yabuki: Quantitative Evaluation of Improved Global Interpolation in the Segmentation of Hand-

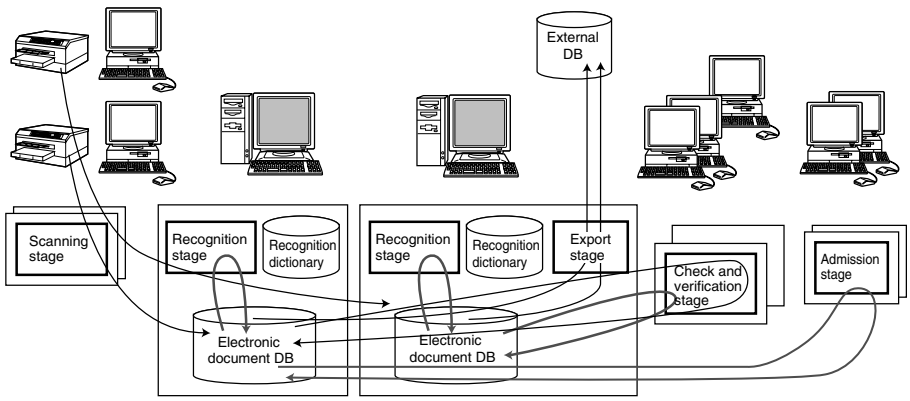

Figure 12
AutoENTRY system using multiple workstations.

written Numbers Overlapping a Border. *IEICE Trans. INF. & SYST.*, **E79-D**, 5, pp.456-463 (1996).
4) S. Naoi, Y. Hotta, M. Yabuki, and A. Asakawa: Global Interpolation in the Segmentation of Handwritten Characters Overlapping a Border. IEEE International Conf. on Image Processing (ICIP-94), **1**, pp.149-153 (1994).
5) S. Naoi, Y. Hotta, M. Yabuki, and A. Asakawa: Segmentation for Handwritten Characters Overlapped a Tabular Formed Slip by Global Interpolation. SPIE Visual Communications and Image Processing'94 (VCIP'94), **2308**, pp.2011-2019 (1994).
6) S. Naoi and M. Yabuki: Global Interpolation Method II for Handwritten Numbers Overlapping a Border by Automatic Knowledge Aquisition of Overlapped Condition. IAPR Fourth ICDAR, **2**, pp.540-543 (1997).
7) H. Takebe, Y. Katsuyama, and S. Naoi: Precise and Fast Form Identification Method by Using Adaptive Base Lines for Matching. Third Asian Conference on Computer Vision-ACCV'98, pp.137-144 (1998).
8) Y. Hotta, S. Naoi, and M. Suwa: Handwritten Numeral Recognition Using Personal Handwriting Characteristics Based on Clustering Method. Third IEEE Workshop on Applications of Computer Vision WACV'96, pp.284-289 (1996).
9) M. Suwa, S. Naoi, and Y. Hotta: Detection of Deleted Patterns in Handwritten Digits Using Topological and Geometrical Image Features. Electronic Imaging: Science and Technology (EI'98) by IS&T and SPIE, **3305**, (1998).
10) K. Fujimoto, H. Kamada, and K. Kurokawa: A Fast, Precise Pre-Classification Method using a Candidate Table Designed by Projections of Feature Regions. 6th International Workshop on Digital Image Processing and Computer Graphics, 1997.

Kazunori Yamamoto received the B.S. degree in Industrial Engineering from Nihon University, Chiba, Japan in 1981. He joined Fujitsu Terminal Systems Ltd., Maebashi, Japan in 1983. He then moved to Fujitsu Ltd., Atsugi, Japan in 1999, where he has been engaged in development of financial systems.

Masaki Kondo received the B.S. degree in Engineering from Akita University, Akita, Japan in 1984. He joined Fujitsu Akita System Engineering Co., Ltd., Akita, Japan in 1984, where until 1995 he was engaged in development of bank terminal systems. He participated in the development of the New Card Center System for Nippon Shinpan Co., Ltd. from 1995 to 1997.

Takayuki Matsui received the B.S. degree in Computer Science from Yamanashi University, Yamanashi, Japan in 1983. He joined Fujitsu Ltd., Inagi, Japan in 1983. In 1997 he moved to Atsugi, Japan, where he has been engaged in development of software for financial systems.

Shinichi Eguchi received the M.S. degree in Physics from Sophia University, Tokyo, Japan in 1989. He joined Fujitsu Ltd., Inagi, Japan in 1989. Since 1994 he has been engaged in development of image-processing for optical character readers.