ISSN 0388-0494

Winter 1981

Jipdec Report

Japan Information Processing

Development Center

Distributed Database System JDDBS-I

Jipdec Report

Winter 1981

CONTENTS

Japan Information Processing Development Center (JIPDEC) was established in 1967 with the support of the Government and related industrial circles. JIPDEC is a non-profit organization aimed at the promotion, research and development of information processing and information processing industries in Japan.

Jipdec Report has been prepared with the assistance of the Japan Keirin Association through its Machine Industry Promotion Funds. These funds are part of the profits that the association had obtained from the sponsoring of bicycle races.

Publisher	Koshichi Ueno
Editor	Yuji Yamadori

Jipdec Report is published quarterly by Japan Information Processing Development Center, Kikai Shinko Kaikan Bldg., 5-8, Shibakoen 3 Chome, Minato-ku, Tokyo, 105, Japan. Annual subscription price is 12,000 yen.

Printed by
Nichigai Associates, Inc.
Printed in Japan, February, 1981

Information Week

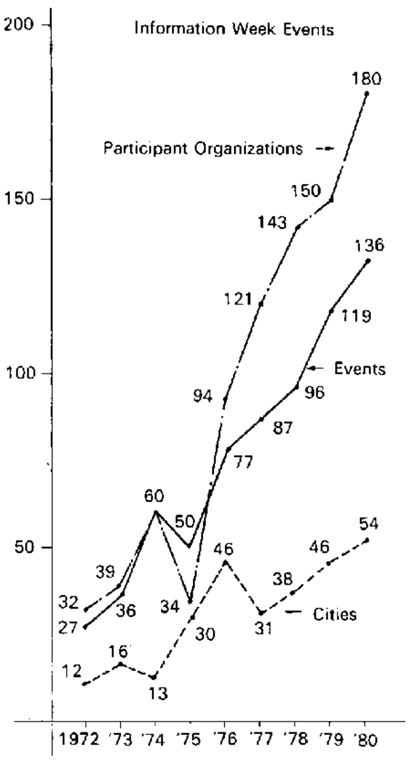
—136 Events in 54 Cities across the Nation
9th Information Week draws to a close—

"Information Week" was first held in 1972 as a result of a cooperative effort between the government and organizations in the private sector. The main themes involved are "Life and Information," and this year saw the ninth staging of Information Week, which opened with a ceremony on October 1 and involved 136 events in 35 prefectures.

The Ministry of International Trade and Industry and six other government ministries and agencies cooperated in settling on the first week in October for Information Week, the thinking behind it being that in view of the fact that the increasing use of information processing, mainly by computer, is now closely bound up with the lives of the people and will continue to be so, the average citizen should have an opportunity to gain a factual understanding of and interest in information processing.

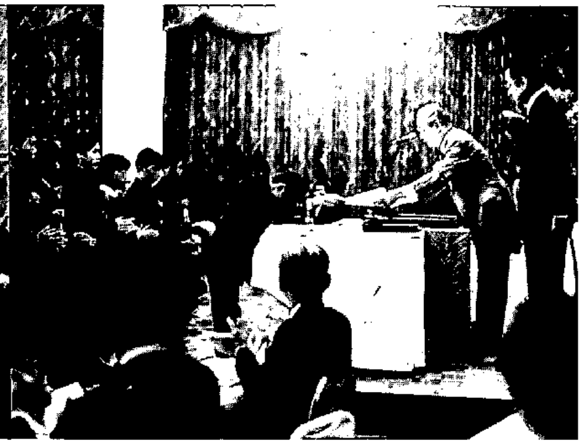
Present at the October 1 ceremony which marked the opening of Information Week were the Minister of International Trade and Industry, Mr. Tanaka, and the Minister of Posts and Telecommunications, Mr. Yamanouchi. Awards were presented to individuals and organizations who have made major contributions to promoting information processing, and for outstanding information processing systems. Each year people are invited to submit catch-phrases or slogans pertinent to the themes involved in Information Week. This year the winning slogan was "Creation of a new age through information," for which the contributor, a Chiba prefectural government employee, received a prize.

This year also saw the first staging of a nation-wide programing contest for senior high-school students, with the grand prize

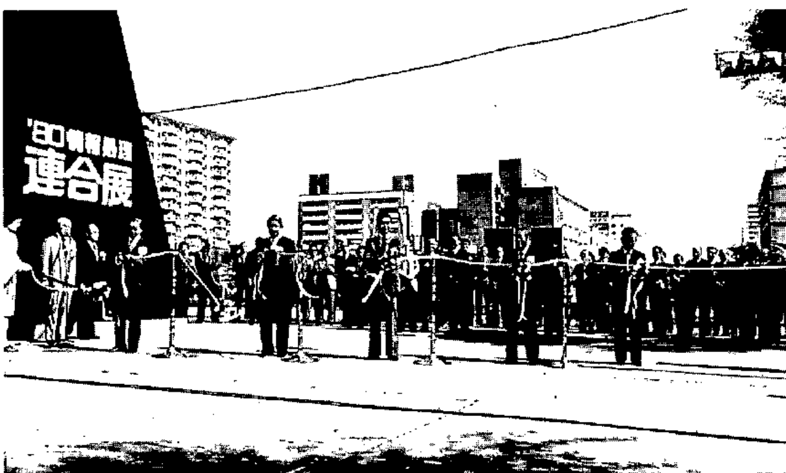


Information Week

▲ Ceremony which marked the opening of Information Week

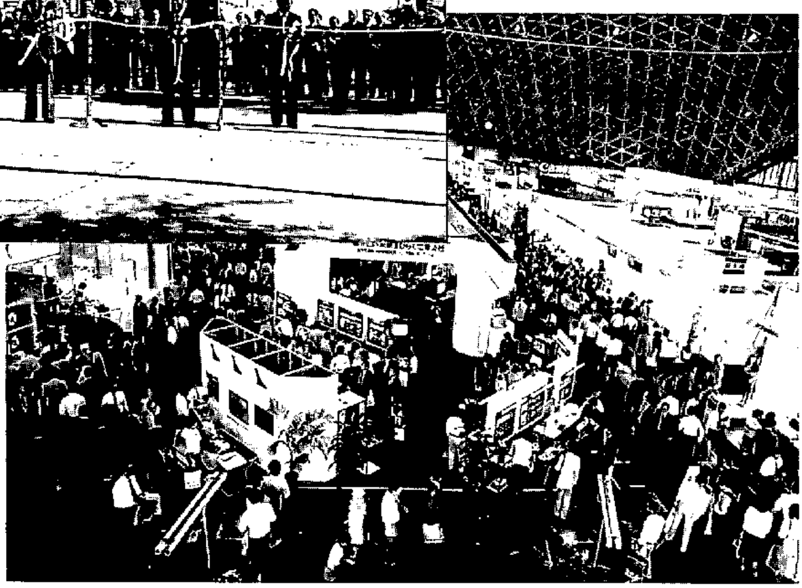


▲ Presentation Ceremony
Rokusuke Tanaka, Minister of International Trade and Industry, Presenting the Prizes



'80 Information Processing
▼ Federation Exhibition Hall

▲ Cutting the Tape for the '80 Information Processing Federation Exhibition



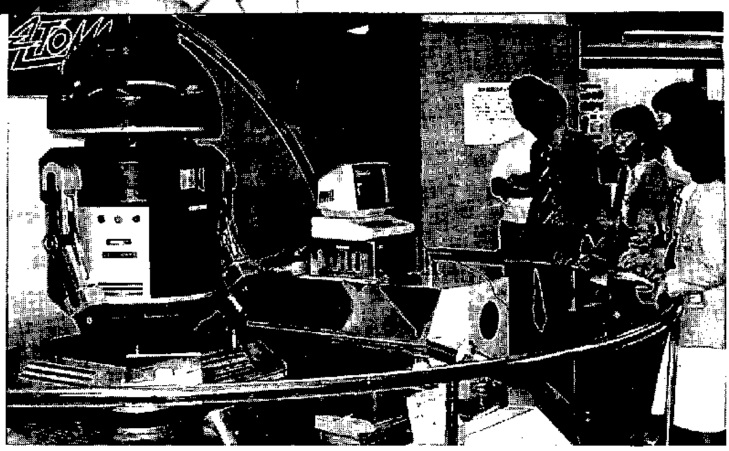
◀ Software Convention '80

'80 Life and Informationalization Exhibition ▶

Mascot Girl

◀ Information Week Symposium

Visitors Playing with "Atom" the Robot ▶



going to "Fortran to Cobol," the winning effort of a student in the 2nd year of the data processing course of Tokyo's No. 2 Commercial Senior High School. This prize, too, was awarded at the same ceremony.

Principal Events of Information Week

1. New post office equipment

Tokyo Central and 58 other post offices started general use of postal zip-code sorting equipment which utilizes OCRs.

2. Shows and displays

(1) '80s Life and Information Festival

This involved the use of microcomputers in small consulting sections, dealing with matters relating to diet, fashion, health and toys and the display and demonstration of video information systems, holograms, video disc devices and the Captain (a home visual information service) system.

This was held in Tokyo, Sapporo and Kitakyushu under the joint sponsorship of JIPDEC, the Visual Information System Development Association and the Medical Information System Development Center. In Kobe these events were sponsored by the Kansai Institute of Information Systems.

(2) Federation of Information Processing Exhibition

Coinciding with the 8th World Computer Conference, IFIP Congress '80, and the 3rd World Conference on Medical Informatics, Medinfo '80 Tokyo, this was jointly sponsored, in Tokyo, by the Information Processing Society of Japan, the Medical Information System Development Center and the Japan Electronic Industry Development Association, and corresponded to the former Data Show.

(3) Data communications exhibition, Electronic communications fair

Demonstrations were given of various applications for data communications terminal devices and communications devices for business purposes. These were sponsored in Nagoya, Osaka, Hiroshima and Matsuyama by NTT (Nippon Telegraph and Telephone Public Corporation).

(4) Medical Information System Exhibition

The Medical Information System Development Center sponsored demonstrations, in Tokyo, of medical information systems and devices.

(5) International Communications Exhibition

This exhibition, held in Tokyo, by KDD (Kokusai Denshin Denwa Co.), used models and wall illustrations to show what was involved in communications satellites and international communications systems.

(6) Software Show '80

Held in Tokyo by the Japan Software Industry Association, this show demonstrated programing, the aim being to promote the increased use of software.

(7) Office Computer Exhibition

This involved the display and demonstration of microcomputers and office computers, and was held in Osaka by the Osaka Science and Technology Center.

3. International Conferences, Seminars and Symposia

(1) 8th World Computer Congress—IFIP Congress '80

Held in Tokyo, this was divided into ten sections. There were 35 special invitational talks, 56 general lectures and 17 panel discussions, with the topics including basic computer logic, architecture, hardware, software, databases, computer networks and communications, computer utilization in science and industry, applications in business and government administration, education and training, and various related social and economic problems.

The first half of the Congress, which was jointly sponsored by IFIP (International Federation of Information Processing), IPSJ (Information Processing Society of Japan), and ACS (Australia Computer Society), lasted eight days, with the second half being scheduled for Melbourne, Australia. Participants numbered 1,500 from Japan and a further 720 from 53 other countries.

(2) The 3rd World Conference on Medical Informatics

This lasted six days and involved 28 reports on clinical medicine, hospital administration, education, health insurance, technological information, and research and working systems.

The main sponsors were the Medical Information System Development Center, the Kansai Institute of Information System, the Information Processing Society of Japan, and the Japan Society of Medical Electronics and Biological Engineering.

(3) Talks and films on life and informationalization

These were held by JIPDEC in Sapporo, Toyama, Tottori, Takamatsu, Kitakyushu and Naha and were about such topics as use of the computer in business and in the home.

(4) Junior high school computer classes

This involved giving talks to students, informing them about computer fundamentals and programing. These talks were sponsored in Nagano and Kyoto by JIPDEC's Institute of Information Technology and the prefectural education commitees, and in Osaka by the Kansai Institute of Information Systems.

(5) Senior high school computer seminars

This involved talks and practical training on computers and their applications, and was also arranged by JIPDEC's Institute of Information Technology and the prefectural education committees, and held in Aomori and Fukuoka.

(6) Computer seminars

These seminars were sponsored by JIPDEC's Institute of Information Technology. Held in Tomakomai, Kesennuma, Hiroshima, Ise and Fukui, the seminars were intended for computer users and dealt with management and computer utilization.

(7) Office automation and networks

Sponsored by the Ministry of Posts and Telecommunications and the Japan ITU Association, held in Tokyo, this involved panel discussions of information processing systems which deal with written Japanese, especially the kanji, and with networks.

(8) Government administration and information systems

This was a symposium sponsored by the Administrative Management Agency and held in Tokyo, and was concerned with surveys and studies dealing with information processing in government administration, and panel discussions on information disclosure and the protection of privacy.

(9) Symposium on information innovation—'80s Topics

This was held in Tokyo under the sponsorship of JIPDEC and the Japan Information Center Association, and involved talks and panel discussions (see below).

—International harmony and the role of the IFIP
—Progress of informationalization and international harmony
—Future changes in society
—Panel discussions
—New computer architecture
—Future management—responding to change
—Panel discussion

(10) Data Show '80—international symposium VLSI technology and information processing

This was held in Tokyo and sponsored by the Japan Electric Industry Association and the Communication Machine Industry Association, and involved talks and panel discussions.

—Impact of VLSI technology on information processing

—Office automation in the U.S.A.—present situation and trends
—New information media of the '80s
—Japanese-language information processing —present situation and trends

(11) Software Convention '80

This was held in Tokyo under the sponsorship of the Japan Software Industry Association and involved talks and panel discussions on the software industry in the '80s and trends in the American software industry.

(12) Software technology, present and future

This was held in Tokyo, sponsored by the Information Technology Promotion Agency of Japan, and consisted of talks dealing with trends in data structures in database systems.

(13) Report on results of research and development work on pattern information processing system

This involved the announcement, in Tokyo, of the results of research and development work on a pattern information processing system, a major project implemented by MITI's Agency of Industrial Science and Technology. The sponsor was the Japan Industry and Technology Association.

(14) Telecommunications in the '80s

This consisted of panel discussions on the outlook for telecommunications in the '80s, and was held in Tokyo under the sponsorship of the Institute of Telecommunications.

(15) The microcomputer revolution in the office

Sponsored by the Japan Data Processing Association and held in Tokyo, this involved debates on how the advance of the microcomputer into the office will change the role of the data processing department.

(16) New communications policy outlook

This involved lectures in Tokyo, sponsored by the Japan Data Communication Association and dealing with the outlook for future communication policies.

(17) Trends in information systems overseas

Held in Tokyo and sponsored by the All Japan Truck Association, this was a symposium on trends in information systems in other countries.

(18) Computer seminar for trainees from overseas

This computer seminar for trainees from other countries was held in Tokyo under the sponsorship of the International Engineers Instruction Association.

(19) Public transportation and information systems

This consisted of lectures about public transport and information systems, and was sponsored by the Transportation and Economy Research Center.

(20) Symposium on microcomputers and office computers

This symposium, held in Osaka, on microcomputers and office computers was sponsored by the Osaka Science and Technology Center.

(21) Urban life cycle plan

This was a symposium, held in Kobe and sponsored by the Institute of Social Systems, dealing with changes in the population structure, and the curben life cycle.

(22) Computer Utilization Technology Research Association

Held in Tokyo and sponsored by MITI's Agency of Industrial Science and Technology, this consisted of panel discussions, examples of research and the announcement of the results of research by the Research Association, the members of which belong to the computer departments of government ministries and agencies.

(23) Administrative Information Processing Research Association

A research conference was held in Tokyo on administrative information processing in national and local governments, under the sponsorship of the Administrative Management Agency and the Ministry of Home Affairs.

(24) Demonstration of scientific and technological document information on-line retrieval system

Held in Tokyo, the demonstrations were of the Japan Information Center of Science and Technology's on-line scientific and technological document information retrieval system, which uses Japanese (kanji).

(25) Demonstration of on-line patent information retrieval system

Sponsored by the Japan Patent Information Center and the Japan Institute of Invention and Innovation, these demonstrations were of PATOLIS (for Patent On-line Information System), and were held in Sendai, Osaka, Okayama and Fukuoka.

(26) Maintenance technicians' awards

The Japan Electronic Computer Co. sponsored this Tokyo event, which involved the presentation of awards to outstanding computer maintenance technicians.

(27) Computer-related staff awards

Held in Tokyo and sponsored by the Local Authority Systems Development Center, this involved the presentation of awards to outstanding computer staff members of local governments.

Certified Data Processing Examination of Japan

1 Outline of the Certified Data Processing Examination System

The Certified Data Processing Examination of Japan, which was first inaugurated in the year 1969, was implemented in 1970 as the Information Processing Specialists Examination in accordance with Article 6 of the Law Concerning the Information-Technology Promotion Agency (IPA Law) and is scheduled to be given for the 12th time this fiscal year. This Certified Data Processing Examination is conducted as one part of the overall policy to ensure the training of information processing specialists who must perform a central role in the development of information processing. In particular, it is aimed at

① the improvement of IP technology by means of providing a stimulus and establishing a goal for the IP specialists who work in private enterprise, corporations, government offices, etc.

② helping to ensure the level of education of IP specialists by means of setting forth standards concering the abilities one should possess as an IP specialist and

③ furnishing a useful and objective yardstick for evaluation at the time of employment, placement and promotion of IP specialists at private enterprises, corporations, government offices etc. where computers are utilized and through this, striving to establish the social position of IP specialists.

In 1969 and 1970, the Data Processing Examination was given in two categories only, those of Junior Programmer and Senior Programmer. But, since 1971, in an effort to perfect this testing system, a special category of the certified data processing examination has been implemented for systems engineers who are engaged in the design and analysis of information processing systems.

This test differs from other national tests in that it doesn't bestow qualifications or licenses on those who pass it. That is, even if a person fails to successfully complete this test, it doesn't mean that he or she will be unable to engage in work in the data processing business. But rather, it is a test which strives to achieve the three aims outlined above by acknowledging the set level of abilities and technological prowess which one should possess as an IP specialist.

2 Implementation of the Certified Data Processing Examination

(1) Test Categories, Examinees and Examinees level of proficiency per category

As shown in the chart below, the Certified Data Processing Examination is divided into and carried out in three different categories, the Special Systems Engineer Examination, the Grade 1 or Senior Programmers Examination, and the Grade 2 or Junior Programmers Examination.

Test Category	Examinees and their level of proficiency per test category
Special Systems Engineer Examination	This category was created to test those IP specialists who are engaged primarily in the design and analysis of IP systems, and assumes that these individuals are college graduates or possess a general knowledge equivalent to college graduates, have had more than 3 years of practical, on-the-job experience in IP, are knowledgeable about computers and various other speciality fields, and are able to carry out systems analysis and design.
Grade 1 (Senior Programmers Examination)	This category tests those IP specialists who are engaged primarily in Programs Design, the Generation of Advanced Program and the training of Grade 2 or Junior Programmers, and assumes that these individuals are Senior Programmers who possess a general knowledge equivalent to that of college graduates, and have had more than 3 years programming experience.
Grade 2 (Junior Programmers Examination)	This category tests those IP specialists who are engaged primarily in the generation of programs in accordance with set program specifications and assumes that these individuals are Junior Programmers who possess a general knowledge equivalent to that of high school graduates, and have more than one year programming experience.

(2) Qualifications of Candidacy for Examination

Qualifications for taking the examination in the categories of Senior Programmer and Junior Programmer are not restricted and anybody, regardless of educational background, sex, age, or experience etc., can take these tests. But, when it comes to the Special Systems Engineer Examination Candidacy is restricted to those individuals who are 25 years of age or older as of April 1 of the testing year.

(3) Testing Periods

The Special Systems Engineer Exam, Senior Programmer Exam and the Junior Programmers Exam are all held on the same date once a year.

3 General Information Concerning Tests to Date

(1) Changes in Number and Types of Examinees etc. since 1969

The exact figures for the number of people tested, and the number and percent of people who successfully completed these exams from 1969, when they were first given, to 1979, the 11th time these test were held, can be found in List 1 and Diagrams 1–4. As for changes in the number of examinees, we see that the Special Exam, the Senior Programmers Exam and the Junior Programmers Exam all display the same trends. The Special Exam wasn't initiated until 1971 and the number of candidates for this test gradually declined between 1971 and 1974, and then, from 1975, has been steadily increasing every year since. If we look at the fluctuations in the number of examinees for both the Senior Programmers and Junior Programmers Examinations, we find exactly the same thing. From 1969 to 1972 the numbers gradually decreased, and then on and after 1973 began to steadily increase each year. In short, the number of people tested in all three categories continued to decline for three years following the initial exam period in 1969, and then began and have continued to increase since that time.

Table 1 Number of Applicants and Persons Passing the Certified Data Processing Examination

	System Engineer			Senior Programmer			Junior Programmer		
	Number of Applicants	Number Who Passed Examination	Pass Rate	Number of Applicants	Number Who Passed Examination	Pass Rate	Number of Applicants	Number Who Passed Examination	Pass Rate
1969	—	—	—	10,527	811	7.7	22,057	1,832	8.3
1970	—	—	—	7,179	977	13.6	16,249	1,649	10.1
1971	2,161	244	11.3	5,634	568	10.1	13,499	1,279	9.5
1972	1,577	236	15.0	4,469	406	9.1	9,747	2,280	23.4
1973	1,479	257	17.4	5,215	631	12.1	10.562	2,304	21.8
1974	1,501	215	14.3	5,600	544	9.7	10,962	2,024	18.5
1975	1,756	189	10.8	6,586	495	7.5	12,469	2,636	21.1
1976	1,883	244	13.0	7,711	866	11.2	15,088	3,085	20.4
1977	2,339	229	9.8	8,178	881	10.8	17,565	3,417	19.5
1978	3,204	295	9.2	9,406	973	10.3	21,112	4,138	19.6
1979	3,887	442	11.4	11,461	1,327	11.6	25,407	5,089	20.0
Total	19,787	2,351	11.9	81,966	8,479	10.3	174,717	29,733	17.0

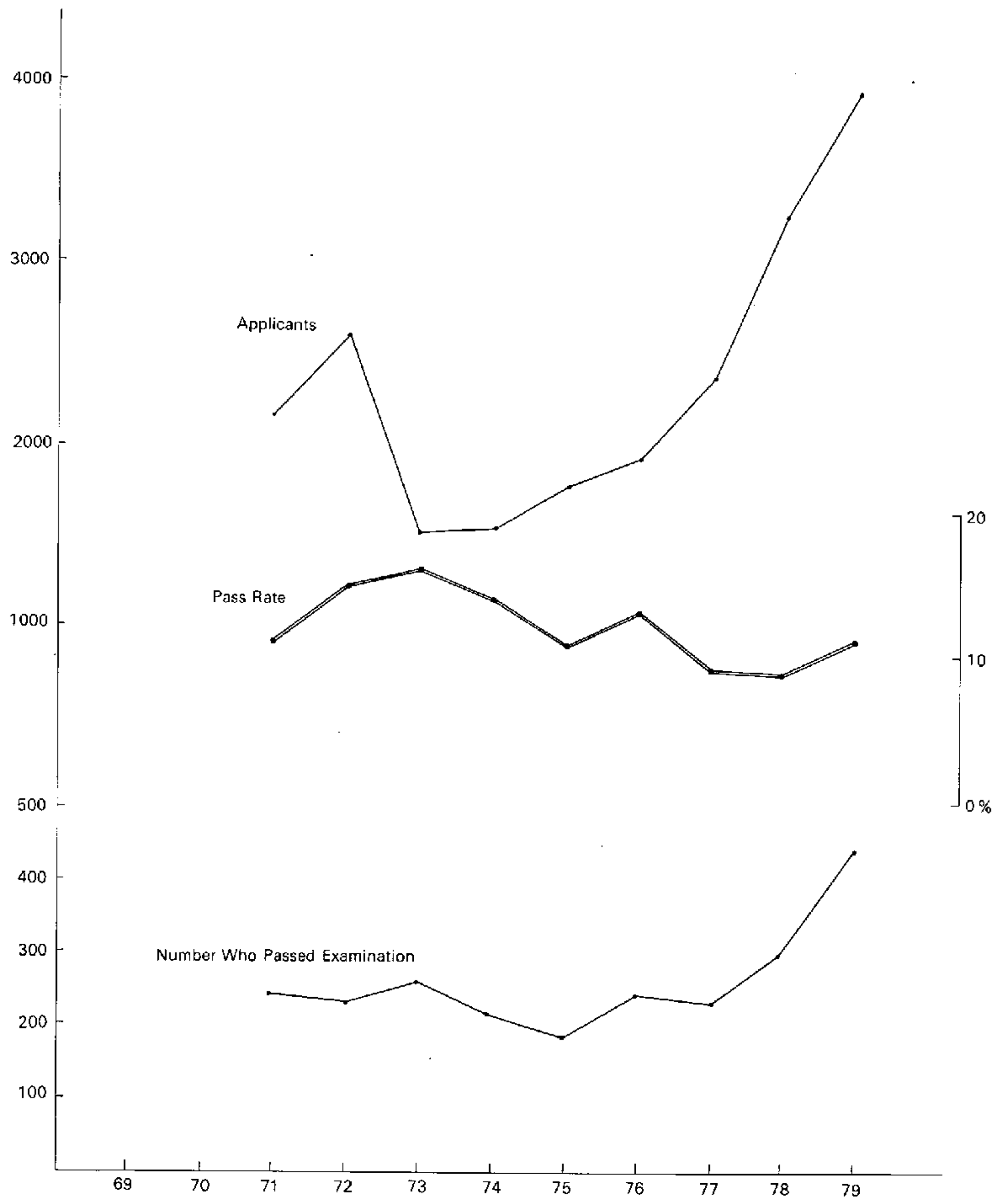


Fig. 1 Systems Engineer

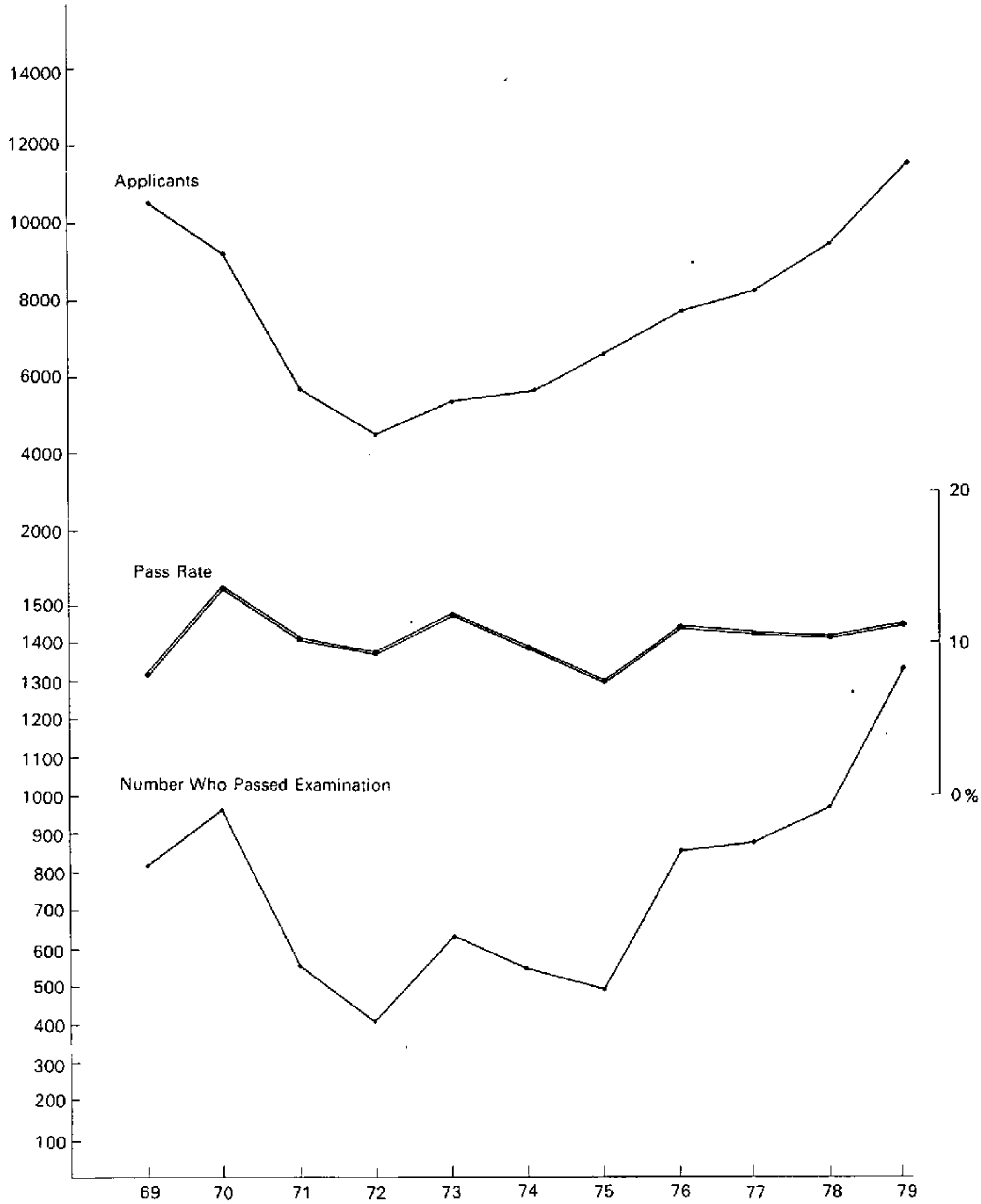


Fig. 2 Senior Programmer

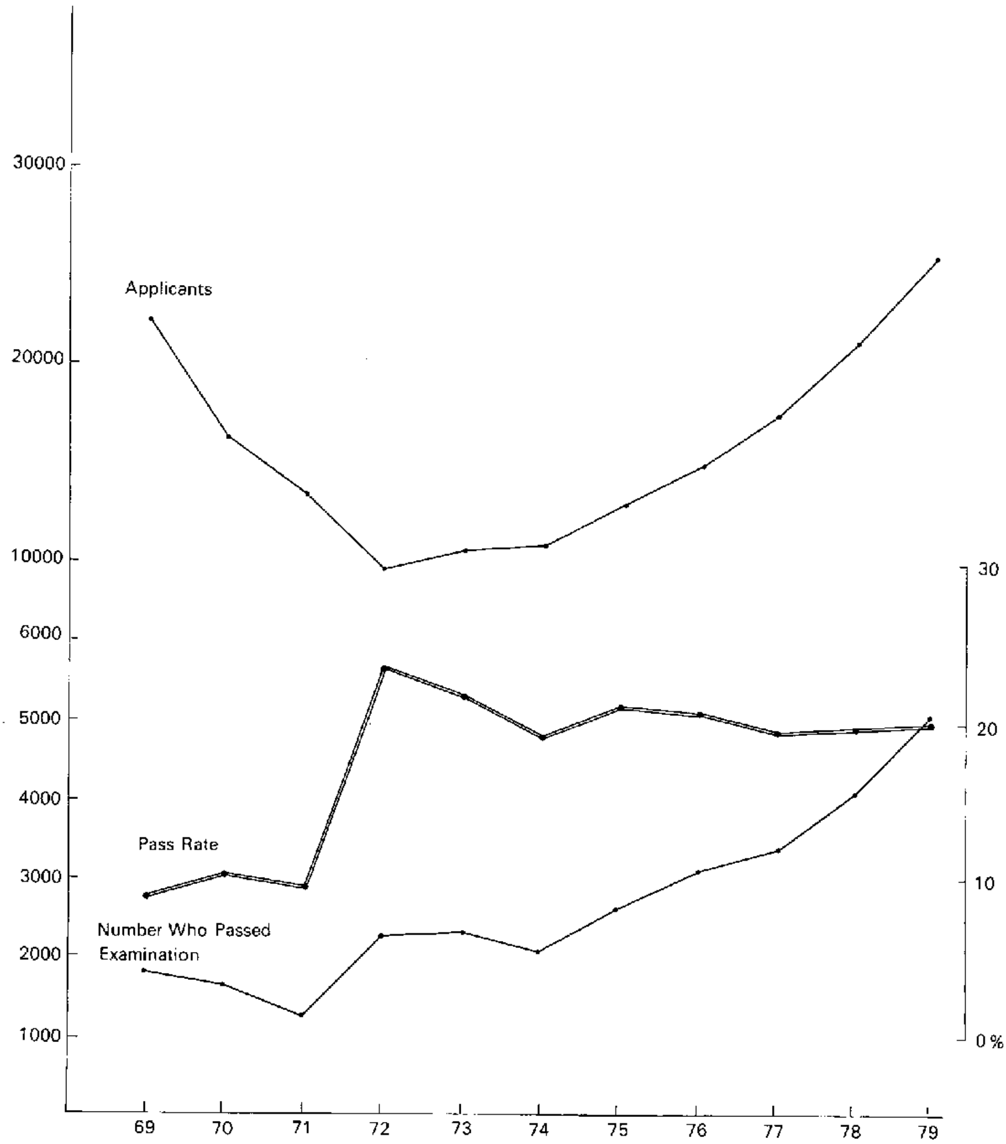


Fig. 3 Junior Programmer

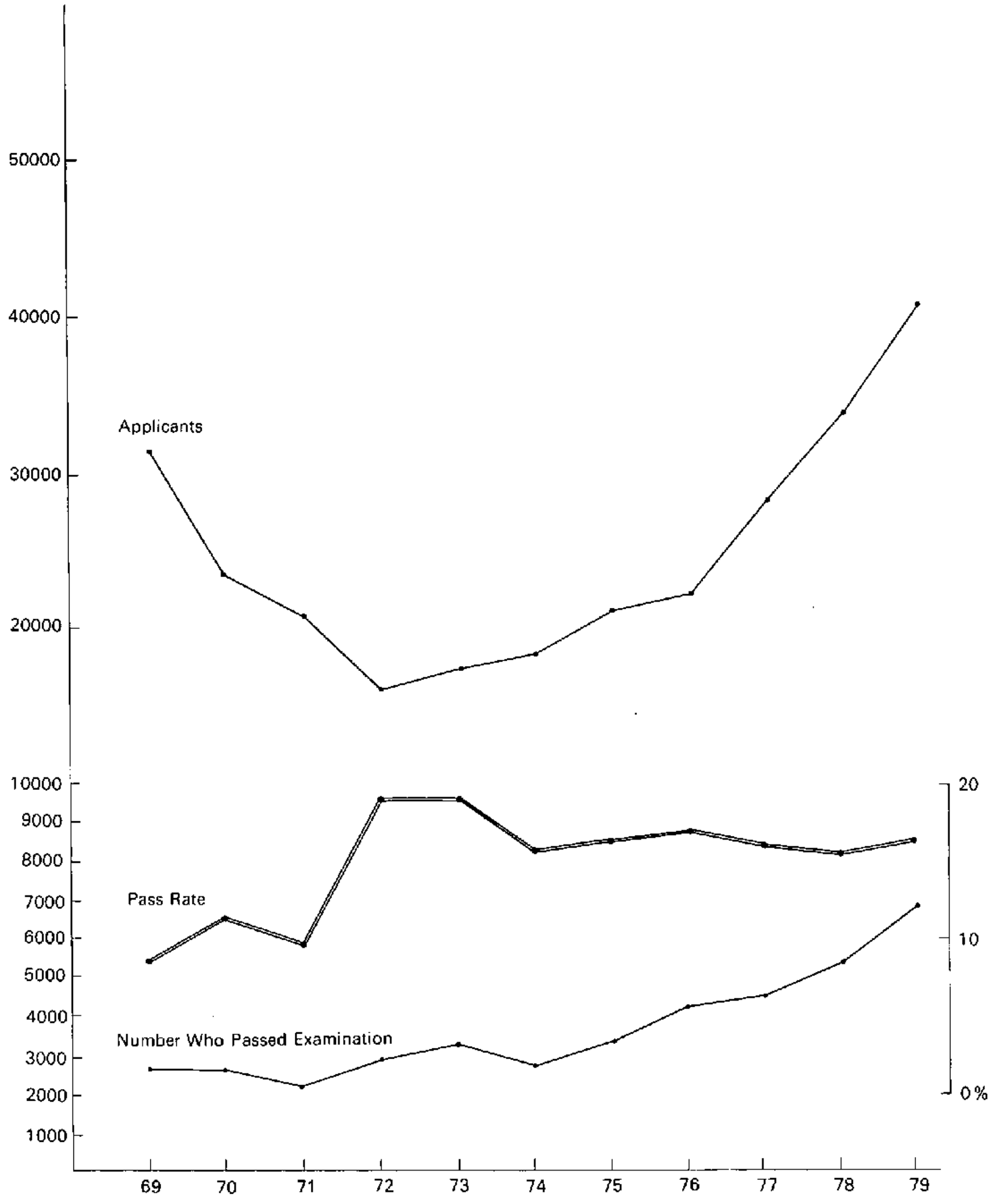


Fig. 4 Total (Systems Engineer, Senior, Junior Programmer)

As indicated above, the number of people tested in all three categories has tended to increase in recent years and especially in the last 3 years has increased markedly. Incidentally, it might be mentioned in passing here that the number of examinees tested from 1974 to 1976 in the Special Exam category increased from 1,501 to 1,883 or 25.4%, and in the category of Senior Programmers from 5,600 to 7,711, for a rate of increase of 37.7%, and in the Junior Programmers category from 10,962 to 15,088 or 37.6%. In relation to this, the number of people tested from 1977 to 1979 in the Special Exam category jumped from 2,339 to 3,887, for a rate of increase of 66.2%, in the Grade 1 or Senior Programmers category from 8,178 to 11,461 for a rate increase of 40.1%, and in the Grade 2 or Junior Programmers category from 17,565 to 25,407 for a rate increase of 44.6%, while the rate of increase grew in all three categories, the most marked increase can be found in the Special Exam category.

The overall success rate for all test periods from the 1st to the 11th (total number of successful candidates ÷ total number of candidates × 100) by category is as follows: Special Exam = 11.9%, Grade 1 or Senior Programmers Exam = 10.3% and Grade 2 or Junior Programmers Exam = 17.0%.

(2) Number of Examinees by Industry (for the most recent 3 year period only)

① Special Exam Category

The substantial increase in the number of examinees from computer manufacturing and sales companies can be raised as the biggest feature for this 3 year period in this category. The total number of examinees from such companies showed a three fold increase, from 426 in 1977 to 1,389 in 1979.

Further, while the number of examinees from Software firms has increased from 581 to 969 (1.7 fold increase) and those from computer centers and so on from 559 to 726 (1.3 fold increase) the fact that the increase in the number of examinees from general enterprise groups from 660 to 702 (1.06 fold increase) is very minimal is another conspicuous tendency in this category.

② Grade 1 or Senior Programmer's Exam Category

During the most recent three year period, the number of examinees from Software firms has increased from 2,538 to 3,947 (a 1.5 fold increase), those from Computer Centers, etc., from 1,780 to 2,438 (a 1.4 fold increase), from Computer Manufacturing and Sales Companies from 931 to 1,542 (1.7 fold increase), and the number of students who have taken this exam, from 458 to 953 (a 2.1 fold increase). The most prominent factors here are the increased number of people from Software firms who have taken the exam, and the percent of increase among students taking the exam. Within this trend toward ever increasing numbers in the respective industries, the general enterprise groups have shown a decrease from 2,112 to 2,086.

③ Grade 2 or Junior Programmer's Exam Category

Again for the most recent three year period only, the number of examinees in this category from Software firms has increased from 3,100 to 5,389 (a 1.7 fold increase), those from computer centers from 2,512 to 3,042 (a 1.2 fold increase), those from computer manufacturing and sales companies from 924 to 1,366 (a 1.5 fold increase) and the number of students taking this exam has increased from 7,064 to 11,440 (a 1.6 fold increase). However, the number of examinees in this category from general enterprise groups has not shown much of an increase at all, moving up only from 3,257 to 3,330.

④ **Future Prospects**

This tendency toward increases in the number of examinees which has been evident for the past several years, is expected to continue in future. This is due in large part to the social setting. The good evaluation given this test by society has taken root over the past 12 years and such things as the Test Incentive System (persons who successfully complete the test are paid either a monthly or lump sum allowance by their company) is continuing to gain in popularity in various enterprises and for students (especially students enrolled at computer related technical schools) successful completion of this test has come to serve as a kind of 'license' for future employment.

Distributed Database System - JDDBS-I

Makoto Takizawa
Development Dept., JIPDEC

Abstract

This paper presents the distributed database system, called JDDBS-I, which aims at integrating heterogeneous DBMSs using a packet-switched computer network. In this paper, we propose the four-schema structure as the gross architecture of the distributed database system. Furthermore, we discuss the heterogeneity problem, i.e. the homogenization of heterogeneous database systems and the translation of a common relational query into a DML program executable on the database system. We also discuss the distribution problem, i.e. the integration of multiple different databases into one logical database and the decomposition of a global query into a sequence of local queries.

Contents

I. Introduction

The advent of the techniques of computer networking and database management has increased the possibility of organizing a number of databases into a cooperative system through communication networks [ADIBM78, ROTHJ77, STONM77]. Such systems are referred to as distributed database systems.

We would like to define distributed database systems as those systems which satisfy the following conditions:

a) database systems which are physically distributed and connected by communication networks,
b) databases which are semantically related with each other, and
c) those virtualized into one logical database system.

This definition means that users can access and describe required data against the distributed database system as if it were a single database system.

When we try to develop such a system from existing database systems by connecting them through a communication network, the main problem we face is the differences inherent in the various database systems. Here, these differences are defined in terms of

a) the data models and languages, and
b) the semantic structures of the stored databases.

In other words, those two differences have to be solved in order to realize the distributed database system. We call the former problem, i.e. to solve the differences of data models and languages, the heterogeneity one, and the latter, i.e. to solve the differences of semantics of databases, the distribution one [TAKIM 80].

At JIPDEC, we have been trying to develop the distributed database system called JDDBS (JIPDEC DDBS) since spring 1977, and finished its 1st phase called JDDBS-I last year. The goal of JDDBS-I was to integrate heterogeneous DBMSs, say CODASYL DBMS, which are distributed over our in-house computer network called JIPNET [YAMAK77], into a logical one. In the JDDBS-I project, we relied on the basic concept and methodology of the distributed database system. That is, our results were,

a) the four-schema structure [TAKIM78, 79] as the gross architecture of the distributed database system,
b) the general relational interface system to the CODASYL DBTG database system, i.e. the solution to the heterogeneity problem [TAKIM80a],
c) the integration method of databases distributed over networks and distributed query processing [TAKIM80b].

In this paper, we would like to present these results in more detail. First, in ch. 2, the four-schema structure concept as the gross architecture of the distributed database systems is discussed. Ch. 2 describes the solution of the heterogeneity problem, i.e. how to translate the CODASYL schema into a common relational schema and translate the relational query into the COBOL DML program. Ch. 3 discusses the distribution problem, i.e. how to integrate the relations at different sites into a required view and process a query against that view on different sites throughout the network.

II. The gross Architecture of Distributed Database System

Our approach towards solving the heterogeneity and distribution problems is called the four-schema structure concept [TAKIM78, 79] which is proposed as the gross architecture of the distributed database system. The data model and language of the database system is considered to be a means of describing and accessing the semantic structure of its database. Hence, first, data models and languages have to be homogenized. Then, the semantic aspect described in terms of a common model can be integrated into one logical data description. We call the former homogenization and the latter integration. Our approach called a four-schema structure is based on such designing process.

The four-schema structure as shown in Fig. 1 consists of four schema-layers, mappings among the layers, and data dictionary/directory (DD/D) [fig. 3]. There are four schemas, local internal (LIS), local conceptual (LCS), global conceptual (GCS), and external (EXS) schemas. The LIS corresponds to a schema or subschema of an existing DBMS, which describes data usable under network situation. The LCS is a description of the LIS in terms of a relational model. The GCS is a relational model description of data integrated from the LCSs. At both LCS and GCS levels, QUEL [HELDG75] is provided. The EXS is a description of data of interest to an application in terms of a data model suited for it. We do not discuss it, because at the GCS

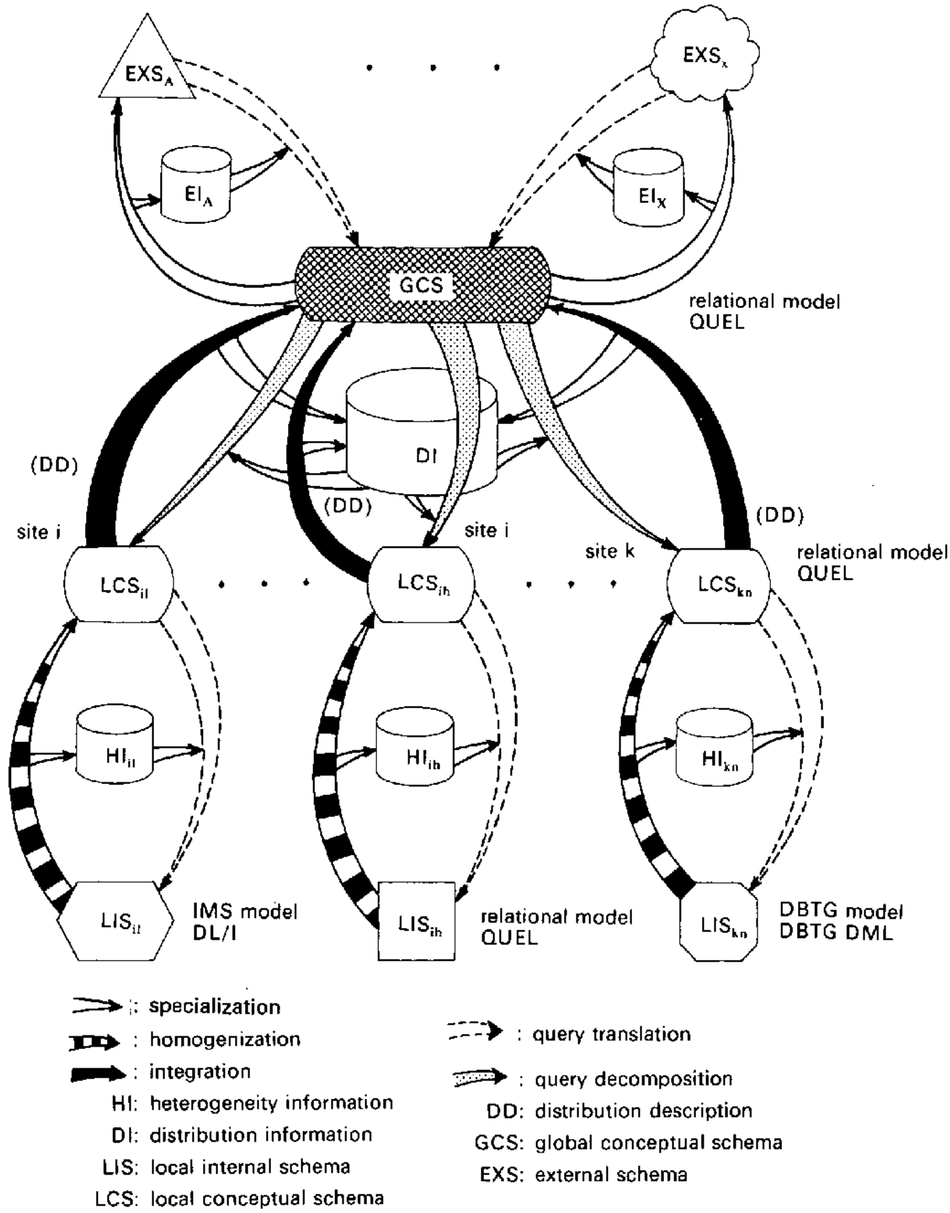


Fig. 1 Four-Schema Structure (FSS)

level the distributed databases are virtualized as one database.

The mapping of the LCSs to the GCS is integration and its inverse mapping is a query decomposition. It is a process where a query based on the GCS is decomposed into a sequence of queries based on the LCSs using information (call it distribution information) removed in the integration. Both queries are written in QUEL. The mapping of the LIS into the LCS is homogenization and its inverse is a query translation. It is a process where such decomposed queries are translated into a sequence of DMLs executable on the database using information (call it heterogeneity information) removed in the homogenization [TAKIM79, 80a]. The distribution and heterogeneity information are totally called the data dictionary/directory (DD/D). The latter exists at its corresponding site, and the former at every site.

Fig. 2 shows the relationships among the processing functions based on the four-schema structure in terms of the ANSI/X3/SPARC [TSICD78].

The reason why the relational model is adopted as the common model is that it provides the most simplicity of description of and access to data from the system viewpoint.

III. Heterogeneity Problems - Homogenization and Integration [TAKIM79, 80a]

In this chapter, we would like to present the solutions to the heterogercity problems, i.e. how to realize the mappings among the local internal schema (LIS) and the local conceptual schema (LCS).

1 Homogenization

Homogenization is a process which translates the LIS based on a data model into the LCS based on a common relational model. Especially, we try to translate a CODASYL DBTG schema into a relational one [TAKIM79]. Our translation mechanism is based on the following rules.

a) From a record-type, a relation (ESR) is generated. The items in the record-type become the attributes. If the item(s) is either a CALC item with DNA or is specified as a DNA in a singular set type, it becomes the key attribute. Otherwise, a new virtual attribute is created, which is considered to take the database keys as its value.
b) From a set-type and a link record-type, a relation (RSR) is generated. The RS-relation consists of two key attributes, one from the key attribute of the ES-relation of the owner record-type and the other from the member one. If the link record-type has items, the attributes corresponding to them are generated. Here, the relation generated from rule a) is called an entity-set (ES) relation and the one from rule b) a relationship-set (RS) relation.

Fig. 5 shows the relation schemes in the LCS generated from the CODASYL schema shown in Fig. 4.

2 Query Translation (QT)

Query Translation (QT) is a process which translates a relational QUEL query referencing LCS relations into a DML program referencing LIS elements and executable on the database system say CODASYL DBTG DML program.

(1) Assumptions

A QT is defined as a process which translates a local query in QUEL which references only relations in a site into a sequence of DMLs executable on a local DBMS. Before describing our QT, we will define our basic assumptions as follows: a) Local queries are in a conjunctive normal form and aggregate-free, i.e. they do not contain any aggregate functions. b) Only retrievals are considered. c) Joins always have

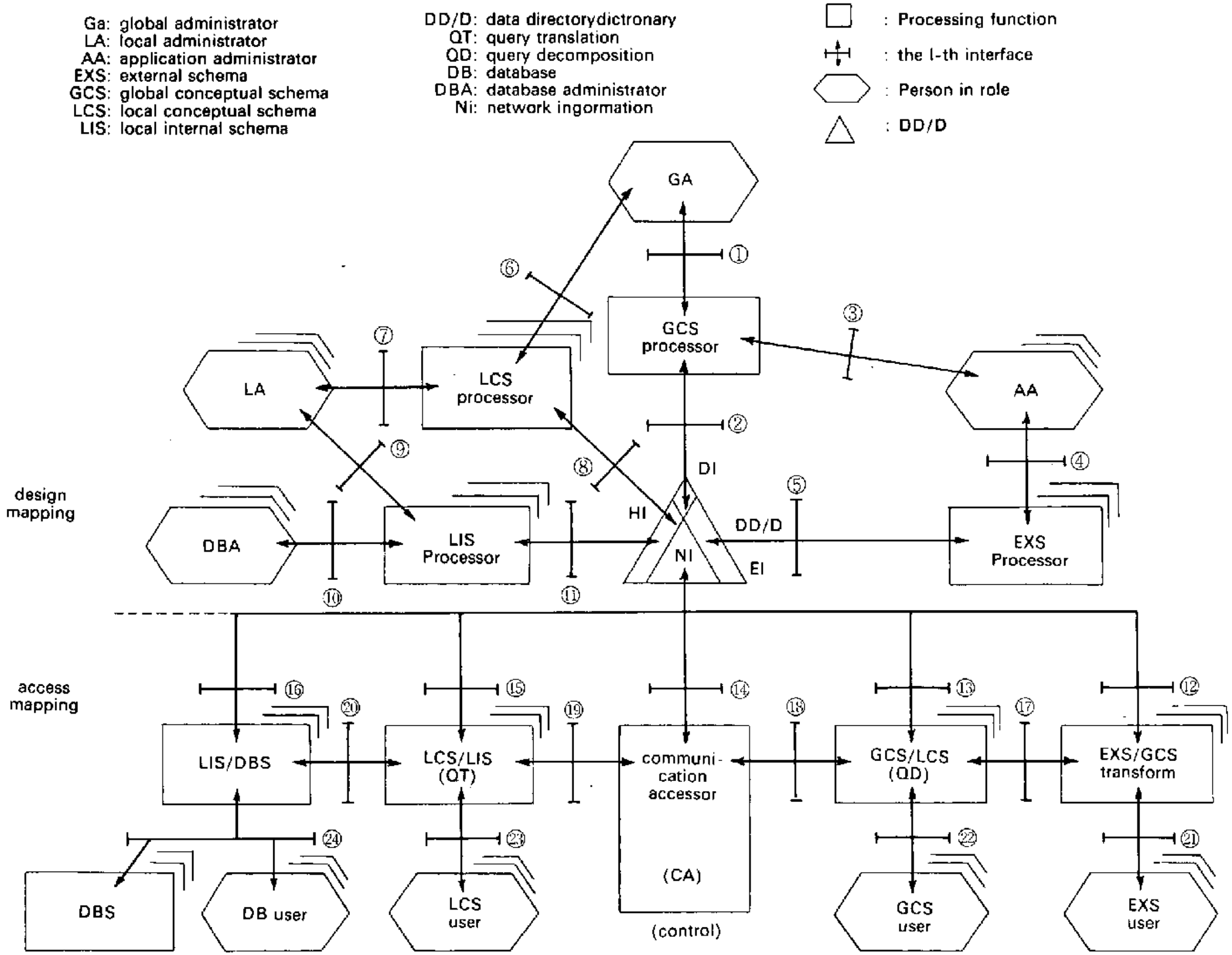


Fig. 2 Schemata of distributed database system

Four-schema structure

1. four schema-layers i) external schema (EXS)
   ii) global conceptual schema (GCS)
   iii) local conceptual schema (LCS)
   iv) local internal schema (LIS)
2. mappings between neighbouring schema-layers
   2.1 design mappings i) homogenization: LIS→LCS
   ii) integration: {LCS}→GCS
   iii) specialization: GCS→EXS
   2.2 access mappings i) query decomposition: GCS→{LCS}
   ii) query translation: LCS→LIS
   EXS→GCS
3. DD/D
   i) heterogeneity information (HI): LIS→LCS
   ii) distribution information (DI): {LCS}↔GCS
   iii) external information (EI): GCS↔EXS

Fig. 3 The constituents of the four-schema structure

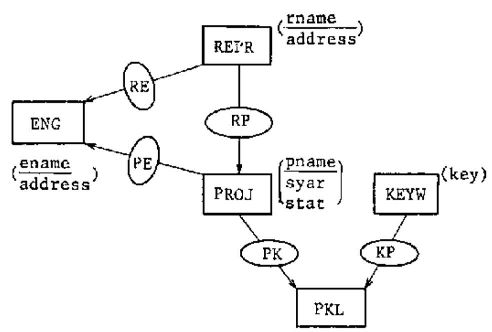


Fig. 4 The LIS

ESR REPR (rname, address)
ENG (ename, address)
PROJ (pname, syar, stat)
KEYW (key)
RSR RP (rname, pname)
RE (rname, ename)
PE (pname, ename)
PKL (pname, key)

Fig. 5 The LCS of fig.4

to reference RSRs and their related ESRs, where RSRs and ESRs stand for the relations representing the relationship-sets and entity-sets respectively. All other joins are considered as meaningless. d) Response time is proportional to the number of occurrences accessed. e) The target DML is a DBTG type.

(2) An Example

This section gives an example of a local internal schema (LIS) and a local query used in this paper. Let us consider the following query against Fig. 5: "find projects and their members such that the representatives of projects that have been doing research on databases since 1975 have their followers in these projects." This is written in QUEL as follows:

range (e, p) (ENG, PROJ);
range (re, pe, rp, pkl) (RE, PE, RP, PXL);
retrieve into R (p.pname, e.ename) where pkl.key="DB" and pkl.pname=p.pname and p.pname=rp.pname and rp.rname=re.rname and re.ename=e.ename and e.ename=pe.ename and pe.pname=p.pname and p. syar≥1975 and p.stat="ON";

(1)

(3) Structure Transform (ST)

The differences between the QUEL and COBOL DML are that

a) QUEL is based on the relational model and the COBOL DML is based on the DBTG model, and

b) QUEL is non-procedural and the COBOL DML is procedural.

Hence, first of all, the meaning of the query in QUEL has to be expressed in terms of the CODASYL model. Its transformation is called the structure transform (ST).

A Relational Query Graph (RQG)

A graph corresponding to eq. (1) is shown in Fig. 6. It consists of nodes and three kinds of links. The nodes stand for variables. The links between the nodes represent joins between them. Two kinds of terminal links (→ and —) show result-attributes and restrictions, respectively.

Here, we emphasize that the graph can express only the conjunction of formulas consisting of joins and restrictions. We call such a graph a relational query graph (RQG).

B Hidden Structures and DBTG query graph (DQG)

Let us consider the transform of the structure of queries and DBTG DMLs. First, we substitute DBTG elements, i.e. record-types and set-types, for relations referenced by the query using the heterogeneity information. This results in the graph shown in Fig. 7. The boxes indicate record-types and the ellipses set-types. The symbols in them represent variables that range over them.

Here, let us consider the node, r, shown by the dotted line. Although it stands for a record-type REPR, it does not appear in the query. The reason is that, since RSRs in the LCS, i.e. re and rp, have the attribute, rname, which is also a key of r, the queries with respect to rname may reference re and rp instead of r. We call the record-types which exist on the LCS but do not appear in the query explicitly hidden structures (EHSs). Next, let us take up pk and kp which represent set-types. Since a link record-type (pkl) is represented as an RSR, set-types (pk and kp) related to it are not expressed on the LCS as shown in fig. 2. We call the set-types which do not appear in both LCS and query implicitly hidden structures (IHSs). EHSs and IHSs are generally called hidden structures.

IHSs and EHSs can be revealed using the heterogeneity information (HI). The HI is composed of three relations named ESR, ATT, and

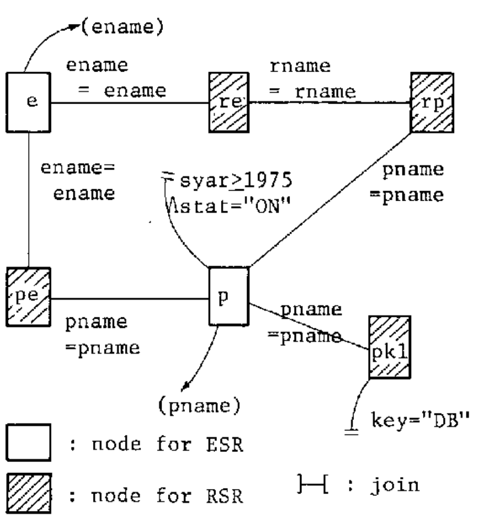


Fig. 6 The RQG of eq. (1)

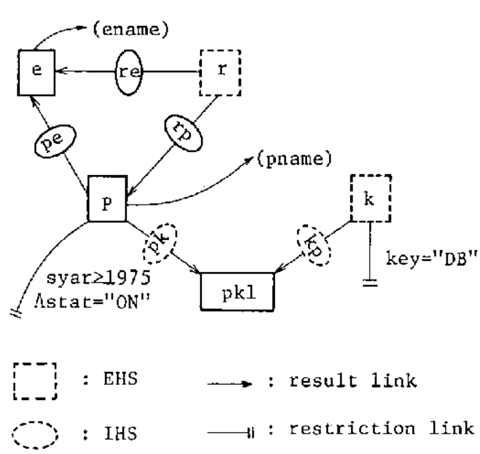


Fig. 7 Hidden Structures

RSR. The ESR relation contains the information concerning the entity-sets, and the RSR relation the relationship-sets. The ATT relation represents the attributes belonging to the relations in the LCS. These schemes are shown in Fig. 8. Let us consider the hidden structure r. It can be found by searching the RSR relation for the record-type, i.e. source-/destination-es-name atrribute, between two set-types rp and re. Fig. 9 shows a graph in which all the hidden structures are made clear. It represents non-procedurally all the record-types and set-types, and the qualification on them, which are required on query processing. We call it a DBTG query graph (DQG).

(4) Access Path Generator (APG)

The next problem is to find a desirable serial access path from the DBTG query graph

```
ESR (entity-set-name, area-root-name, number-of-
     keys, access-mode, degree, width, size,
     protection-flag, integrity-flag, cardinality)
ATT (es-rs-name, es-rs-type, attribute-no, attri-
     bute-name, value-set, role-of-attribute,
     character/decimal, length, protection-flag,
     integrity-flag, cardinality, selectivity)
RSR (relationship-set-name, source-es-name,
     destination-es-name, access-mode, relation-
     ship-construct, degree, width, size, source-
     set-name, destination-set-name, protection-
     flag,integrity-flag, cardinality)
   c.f. es: entity-set    rs: relationship-set
```

Fig. 8 Heterogeneity Information (HI) Schema

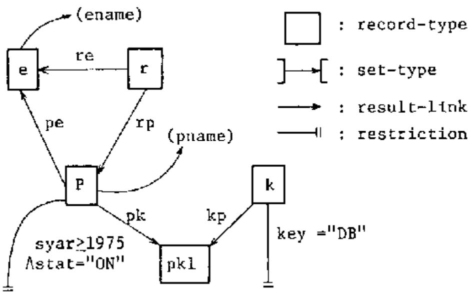


Fig. 9 DBTG Query Graph

(DQG), which has to include all the nodes and links in the DQG. The objectives are a) to minimize the number of intermediate results, and b) to minimize the occurrences accessed. If we have some intermediates, more than one file to store them and set operations, not supported by the target DMLs, to process them are required. Hence, we believe that it is desirable not to create any intermediate from the operational viewpoint. From d) in 2.1, we also have to achieve the second goal in order to realize quick access.

A The Expected Number of Occurrences to be accessed

We try to estimate the accessed occurrences in this section. Let n_i be the i-th node of the DQG, a_{ij} the j-th attribute of n_i, c_i the cardinality of n_i, and s_{ij} the selectivity [HEDNA78] of a_{ij}. Using this notation, the expected number of occurrences to be accessed in n_i (denoted by OCA_i) which satisfy the equi-restriction "$a_{ij}=v$" can be computed to be $c_i\, s_{ij}$.

Furthermore, let s_i be the selectivity of the formula concerning n_i, s_i' the selectivity of the equi-restrictions referencing the CALC-item (call them CALC-equi-restrictions), and s_i'' the selectivity of restrictions other than CALC-ones where $s_i = s_i'\, s_i''$. If CALC-equi-restrictions exist, $s_i' \leq 1$, otherwise $s_i' = 1$ because all the occurrences in n_i have to be accessed (the occurrences are assumed not to be sorted). If n_i is a starting node of the access path, the OCA_i depends on its access mode, i.e. $OCA_i = s_i' * c_i$.

Next, let us consider the case where n_i is accessed from n_j via a set-type S. Here, let sl_{ji} be the expected number of occurrences of n_i linked to one occurrence of n_j through S.

$OCA_i = OCA_j\, sl_{ji}$ where

$$sl_{ji} \begin{cases} =1 \text{ if } n_i \text{ is an owner of S and } n_j \text{ a} \\ \quad \text{a member} \\ \geq 1 \text{ if } n_i \text{ is a member of S and } n_j \text{ an} \\ \quad \text{owner.} \end{cases} \qquad (2)$$

Suppose that k nodes $(n_1, \ldots, n_k)$ are chained linearly. The expected number of occurrences to be accessed in the chain is

$$c_1 s_1'(1 + s_1'' sl_{12}(1 + s_2 sl_{23}(1 + \cdots (1 + s_{k-1} sl_{k-1,k}) \ldots))). \quad (3)$$

where $s_1' = 1$ and $s_1 = s_1''$ unless the n_1 is CALC.

B Access Tree (AT)

The access path corresponds to the order of visiting pairs of a link and its linked node in the DBTG query graph (DQG). This problem is similar to generating a minimal cost spanning tree. The difference is that our tree (call it an access tree or AT) includes all the DQG links uniquely together with all DQG nodes, some of which may be redundant. The first property results in the second, i.e. the redundancy of nodes. We call such redundant AT nodes confluent nodes for the DQG node. The AT is generated based on a depth-first search. Our algorithm (call it DFA) is shown in fig. 10.

Suppose that the node k has the least occurrences to be accessed in the DQG of fig. 9. The AT with k as a root node is generated as shown in fig. 11. The nodes represent record types and the specially hatched ones are confluent. The arcs stand for set-types whose directed relationship from owner to member is shown as an arrow.

```
0. [initialization]
    let STB(x) be an adjacency node list for
    the node x in which branches (x,y1),...,
    (x,yn) are sorted in the ascending order of
    the occurrences to be accessed (OCA) of the
    adjacent nodes, i.e.
   OCAy1 ≤ ... ≤ OCAyn , where OCAyi = OCAx·Slxyi
    for all the nodes (x) in the DQG, mark them
    NEW and create STB(x); AT ← Λ; pushdown (Λ);
1. select the starting node(x), which has the
   least OCA in all nodes;
2. pushdown(x);  if x is marked NEW, then mark
   it OLD; link the AT node x to the DQG node x.
   if x is marked OLD, then mark both it and the
   AT node linked to the DQG node x CONFLUENT.
3. if STB (x) is empty, go to 7.
4. [search STB(x)]
    get the first pair (x, y1) from STB(x);
5. link the node y1 to the AT;
6. [delete (x, y1)]
    delete (x, y1) from STB(x) and delete
    (y1, x) from STB (y1); x ← y1; go to 2;
7. [pop up]
    popup (x); if x = Λ, terminate; go to 3;
```

Fig. 10 The Algorithm for Creating the AT

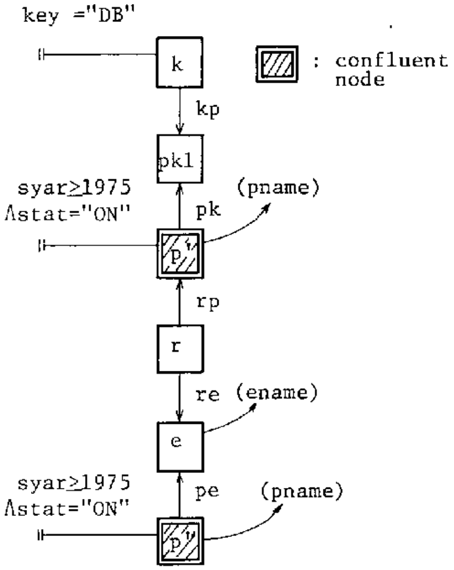


Fig. 11 Access Tree by DFA

Let us look at the AT nodes p' and p" in fig. 11. They are confluent nodes for the DQG node p. This means that, under an access sequence of occurrences, the occurrences of p' have to be the same as p". Hence, when the access to p" is completed, its occurrences have to be compared with p'. If both are the same, the access is safe, otherwise a failure.

The problem is how to simplify this comparing process. The characteristic of our AT is that, given a node in the DQG, all the confluent nodes for it always appear in the subtree in which one of them is the root node. Hence, only the first confluent node needs to output the value of the key attribute to the result-relation, and, in the other ones the current value of the key is only compared with the first value. This means that, only when the key attribute of the confluent node is not included in the result-attribute list, the attribute for storing its value is added to the result relation. This results in the least intermediates. It is an important advantage of our algorithm.

C Generation of an Access Path from the AT

An access path consists of ordered pairs each consisting of an arc and its related node. This order corresponds to a preorder traversal of the access tree (AT). Each pair in the access path corresponds to a unit of an access of the target DBMS, e.g. FING NEXT . . . , FIND OWNER . . . for DBTG.

Let us take fig. 11 as an example. From the AT, we can get the sequence [(−, k) (kp, pkl) (pk, p*) (rp, r) (re, e) (pe, p*)] where nodes marked with asterisks indicate confluent nodes.

(5) DML Generator (DMLG)

The last module, DMLG, takes the access path and generates a sequence of COBOL blocks containing DBTG DMLs, each of which corresponds to an access unit of the path. There are ten kinds of access units. They are first classified into four classes: independent, starting, intermediate, and leaf units. Independent and starting units are divided into two subclasses: CALC and non-CALC units. The other classes are divided into two subclasses: owner and member units with respect to set-types represented by the arcs. Furthermore, non-CALC and member units which are confluent are said to be CN-non-CALC and CN-member units respectively. Since they may be accessed more than one time through different set-types, the currency with respect to each set-type has to be saved. We try to match each access unit against ten classes. If matched, corresponding DMLs are automatically generated.

Let us take the third unit which includes pk and p. This unit is described as follows:

(3(SET PK) (REC P (RSLT (PNAME)) (RSTR (AND (GE SYAR 1975) (EQ STAT "ON")))) (UTYPE OWNER CONFLUENT LEAF) (ERTRN 2) (TRTRN 2))

This is an access unit corresponding to an OWNER record-type. Since p is also confluent, it plays a twofold role as result-attribute and working attribute for the confluent nodes. TRTRN (ERTRN) shows the unit-no to which the access is returned from the current unit, when occurrences to be accessed are exhausted and satisfiable ones are found (not found, respectively). This unit shows that if the current occurrence of p does not satisfy the condition, i.e. syar$\geq$1975 and stat="ON", PKL's next occurrence is accessed.

IV. Distribution Problem - Integration and Query Decomposition [TAKIM80b]

The distribution problem is how to realize the mapping among the global conceptual schema (GCS) and the local conceptual schemas (LCSs).

1 Integration

The integration is a process for defining GCS relations from existing LCS relations distributed over a network. It is also similar to the definition of views [STONM76] in a relational model. In designing a relational database, one universal relation is vertically decomposed into normalized relations. Views can be defined by means of joins as multirelational operations. But in bottom-up designing, relations have already existed at each site. This implies that unions are also required. Unfortunately, relational calculus languages like QUEL do not provide such capabilities. Our language for defining GCS relations which is called GSDL is an extension of QUEL so that the unions can be taken.

The GSDL consists of three kinds of statements, drange, define, and drop. The drange is used to define tuple variables against the LCS relations along with their existing sites:

drange $(x_1, \ldots, x_m)$ $(X_1{:}s_1, \ldots, X_m{:}s_m)$;

where each x_i for $i=1, \ldots, m$ stands for a tuple variable ranging over an LCS relation X_i at site s_i.

The define is used to define a GCS relation:

```
*---------- DBTG DML ----------
         MOVE FALSE TO LFOUND.
 L0101.  CALL  GET-NEXT-VALUE ((KEYWORD = "DATABASE" ),VAL,MODE).
         IF  MODE = 'END'  GO TO TERM.
         MOVE VAL TO KEYWORD IN KEYWORDS.
         FIND ANY KEYWORDS.
         IF  NOTFOUND = 'YES'  GO TO L0101.
         GET KEYWORDS.
*
 L0201.  IF  L0210 = TRUE  GO TO L0202.
         MOVE FALSE TO LFOUND.
         IF  KEYW-PROJ IS EMPTY  GO TO L0101.
         MOVE TRUE  TO L0210.
         MOVE FALSE TO L0211.       GO TO L0203.
 L0202.
         IF  L0211 = TRUE  OR  LFOUND = TRUE
                   MOVE TRUE  TO L0211
              ELSE MOVE FALSE TO L0211.
 L0203.  FIND NEXT PROJ-KEYW-LINK WITHIN KEYW-PROJ.
         IF  ENDSET NOT = 'YES'  GO TO L0204.
         MOVE L0211 TO LFOUND.
         MOVE FALSE TO L0210.
         IF  L0211 = FALSE  GO TO L0101.
                            GO TO L0101.
 L0204.  GET PROJ-KEYW-LINK.
*
 L0301.  FIND OWNER WITHIN PROJ-KEYW.
         IF  NOTFOUND NOT = 'YES'  GO TO L0303.
 L0302.  MOVE FALSE TO LFOUND.
         GO TO L0201.
 L0303.
         GET PROJECT.
         IF  NOT ( PROJSTAT = "ON" AND PROJSYAR GE  1975 )  GO TO
          L0302.
         MOVE TRUE TO LFOUND.
         CALL  RESULT (PROJNAME ).
```

```
*
 L0401.  FIND OWNER WITHIN REPR-PROJ.
         IF  NOTFOUND NOT = 'YES'  GO TO L0403.
 L0402.  MOVE FALSE TO LFOUND.
         GO TO L0201.
 L0403.
         GET REPRESENTATIVE.
         MOVE TRUE TO LFOUND.
*
 L0601.  IF  L0610 = TRUE  GO TO L0602.
         MOVE FALSE TO LFOUND.
         IF  REPR-ENGI IS EMPTY  GO TO L0201.
         MOVE TRUE  TO L0610.
         MOVE FALSE TO L0611.       GO TO L0603.
 L0602.
         IF  L0611 = TRUE  OR  LFOUND = TRUE
                   MOVE TRUE  TO L0611
              ELSE MOVE FALSE TO L0611.
 L0603.  FIND NEXT ENGINEER WITHIN REPR-ENGI.
         IF  ENDSET NOT = 'YES'  GO TO L0604.
         MOVE L0611 TO LFOUND.
         MOVE FALSE TO L0610.
         IF  L0611 = FALSE  GO TO L0201.
                            GO TO L0201.
 L0604.  GET ENGINEER.
         CALL  RESULT (ENGINAME ).
*
 L0701.  FIND OWNER WITHIN PROJ-ENGI.
         IF  NOTFOUND = 'NO'  GO TO L0703.
 L0702.  MOVE FALSE TO LFOUND.
         GO TO L0601.
 L0703.
         GET PROJECT.
         IF  NOT ( PROJSTAT = "ON" AND PROJSYAR GE  1975 )  GO TO
          L0702.
         MOVE TRUE  TO LFOUND.
         CALL  RESULT (PROJNAME ).
         GO TO L0601.
```

where variables L0210, L0610 are initially set to FALSE.

Fig. 12 DBTG DML Program (Procedure Division) define ⟨grelname⟩ (⟨gatt-list⟩) ⟨sub-def⟩ {:⟨sub-def⟩};

⟨grelname⟩ and ⟨gatt-list⟩ define a scheme of a GCS relation to be defined. ⟨sub-def⟩ is called a subdefinition of the GCS relation:

⟨sub-def⟩::=(⟨target-list⟩) where ⟨qual⟩

The target list and qualification are the same as QUEL. The ⟨sub-def⟩ defines a join of relations like QUEL. The list of subdefinitions divided by colon means that the GCS relation is the union of results each of which is derived with respect to each subdefinition.

The drop is used to remove the defined GCS relation from the GCS.

In order to integrate LCS relations into a GCS relation, the LCS relations have to share the common union-compatible parts. The relationships are expressed by set-theoretical expressions on semantic links as shown in fig. 13. For example, the expression PRJ2 [pno, pname] PROJ [no, name] shows that both projections are union-compatible and PROJ [no, name] contains the same value set as PRJ2 [pno, pname]. PRJ1 PRJ2 represents that both are union-compatible. Fig. 14 shows the definition of a GCS relation PROJECT by means of the GSDL based on the semantic links as shown in fig. 13. Such definitions of GCS relations are stored as the distribution information. This information is stored redundantly at every site and used to process the query decomposition.

2 Query Decomposition

The query decomposition process is composed of two subparts: the decomposition of GCS relations referenced by a GCS query into corresponding LCS relations, and the processing of inter-site joins. Query modification technique [STONM76] can be adopted for doing the first. Since the network causes a bottleneck of the distributed database system, how to process the second efficiently is the most significant in the query decomposition.

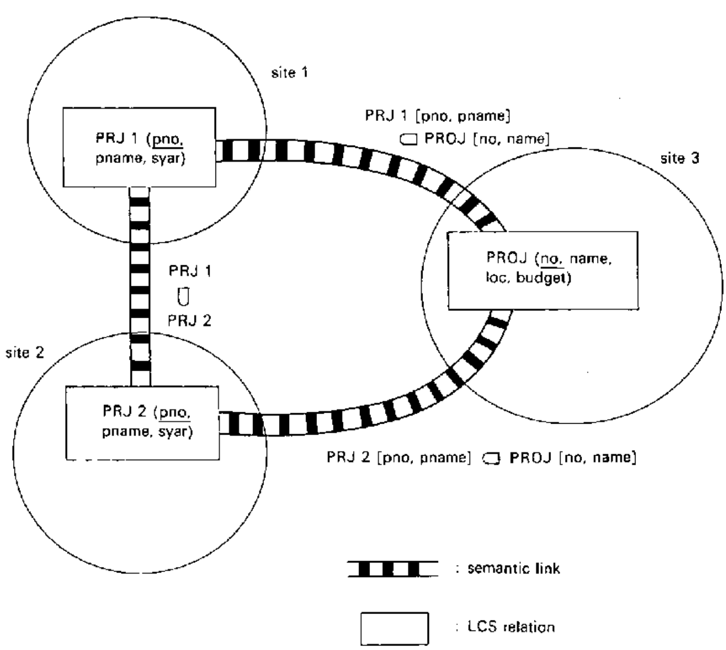


Fig. 13 The semantic links between three LCS relations

```
drange (p1, p2, p) (PRJ1:1, PRJ2:2, PROJ:3);
define ESR PROJECT (no, name, budget, location)  ⎫
  (p. no, p. name, p. budget, location =p. loc)   ⎬   (1)
  where p. no =p1. pno and p. name =p1. pname:    ⎭
  (p. no, p. name, p. budget, location =p. loc)   ⎫
  where p. no =p2. pno and p. name =p2. pname;    ⎭   (2)
```

Fig. 14 The definition of the GCS relation PROJECT

(1) Basic Assumptions

We make the following assumptions [HEVNA78] on the network:

a) It is a site-to-site type.
b) It is always lightly loaded. Therefore, there is no need for considering queueing delay.
c) Local processing costs are neglectable compared with communication costs.
   An additional assumption is made.
d) Its communication cost depends on a distance, and its measure is time. A logical cost, LC_{ij}, is defined as a delay time for transmitting a packet from site i to j. It is also proportional to the number of hops between them.

We also make the following assumptions on processings at sites:

a) Each site has a working space managed in a relational form. Relations transmitted from the other sites and results of joins are stored in it.
b) Each site has two kinds of logical processors: a global database processor (GDP) and local database processors (LDPs) [TAKIM79]. The GDP plays a role of the query decomposition, integration, and management of the distribution information and working space at each site. Especially, the GDP to which a GCS query is stated is called a coordinate GDP (CGDP) that is a centralized controller for processing the query. The LDP is responsible for the query translation, homogenization, and management of the heterogeneity information, and exists against one database. LDPs which support data required by the GCS query are cooperated under the CGDP's control.

(2) Objectives

The query decomposition purposes largely to generate an optimal sequence of stages. The objectives which have been taken up [HEVNA78] are to minimize the communication cost and the response time. The network causes a bottleneck of the distributed database system due to its restricted capacity. On the other hand, each site has some processing capacities. Therefore, in order to achieve these objectives, it is necessary to reduce the network traffic for query processing and process queries in parallel at multiple sites. Works which have been done so far [CHUW79, HEVNA78, EPSTR78, WONGE77] aim at achieving the objectives. Their characteristic is that strategies for transmitting relations are determined by estimating sizes of intermediates in an off-line manner. This estimation is based on statistics on relations such as selectivities [HEVNA78, SELIP79]. The selectivity is based on an assumption that values of the attribute are uniform-distributed. But we think that there are still problems whether actual distribution of values follows well this assumption. In order to make the selectivity more precise, [CHUW79] proposes a method such that selectivities of well-used values are accumulated in the directory each time the values are accessed. However, it is noted that the more precise statistics on selectivities we have, the more storages are required.

We argue the following points. First, the performance information like selectivities and cardinalities have dynamic properies compared with the schema information. Secondly, the query decomposition and its required information, i.e. the distribution information, have to exist at the same site for the efficiency purpose. It implies that every site has to equip a full copy of the distribution information. If each site provides such dynamic information strongly redundantly, the overhead for not only storing them but also controlling consistency of concurrent accesses to them becomes serious and enormous [BERNP80]. Besides the objectives as stated above, therefore, we would like to add one objective, i.e. to keep the information required by the query decomposition as small and static as possible.

(3) Strategies

Our strategies for processing inter-site queries are as follows. First, the CGDP decides cosequent stages dynamically based on the statistics of result relstions of the preceding stages. This results in small and static distribution information.

Secondly, the query parts referencing only one site are processed locally at the site before the inter-site parts are processed, because the local processings are neglectable in cost and result in the reduction of sizes of relations to be transmitted.

Thirdly, if two relations at different sites are to be joined, the smaller one is transmitted to the larger through a path with minimum transmission cost. We do not consider strategies such that two relations at different sites are transmitted to the other site and joined there.

Lastly, even if all the stages issued by the CGDP do not complete, if there exists a relation not being processed and a path with transmission cost less than some threshold value, then it can be transmitted through the path. By that, more than one stage can be processed in parallel.

3 Initial Local Query Processing

Let us assume the GCS query is translated into a query which references LCS relations. It is called a global LCS query. The query can be divided into two parts. One references only one site, and the other multiple sites. The former parts have to be processed, first, closely at one site, because it results in reduction of relations to be transmitted and its cost is assumed to be neglectable. We call such a local processing an initial local query processing.

It is composed of the following functions:

1) to make a query graph [TAKIM80a] of the global LCS query which is called a GLQ graph [see fig. 15],
2) to classify nodes in the GLQ graph into groups each of which consists of the nodes at the same site and connected by join-links in the site, and
3) to generate LCS queries each of which corresponds to each group, and send them to the corresponding sites.

For example, let us consider the GLQ graph in fig. 15.

Fig. 16a shows the resultant query graph. As seen in this figure, it contains only inter-site joins. Hence, it is called a join query graph.

4 Transmission Scheduling

We consider the generation of a transmission scheduling, i.e. an efficient sequence of stages, from the join query graph. Let r′ and r be relations at sites i and j, respectively. The stage consists of two subparts: a transmission of r′ from i to j, and a join of r′ and r and storing of the result as r at j. Hence, let $r'{:}i\rightarrow j$, $c(r'{:}i\rightarrow j)$, and $r'{:}i\rightarrow j{:}r$ be such a transmission, its cost, and such a stage, respectively. Let LDP_k be the local database processor at site k.

Our algorithm for generating the transmission scheduling is operational but not static. This means that the CGDP decides consequent stages based on monitored information on results of the preceding stages. To monitor such

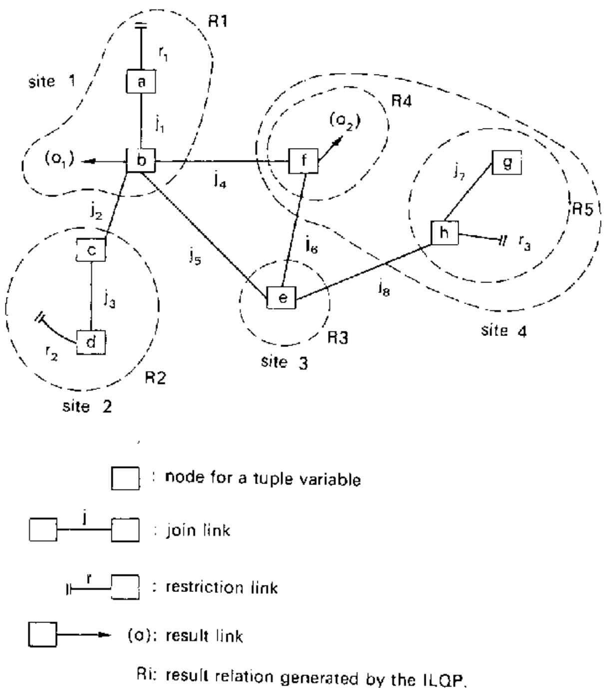


Fig. 15 An Example of the GLQG

results, each GDP manages two kinds of directories along with the distribution information, i.e. logical transmission cost table (LCT) and query processing information (QPI). In the QPI, the performance information of intermediate results are stored in two relations: QPI/REL (site-no, rel-no, cardinality, width) and QPI/ATT (site-no, rel-no, att-no, width). Both maintain the performance information on relations and their attributes produced by stages, respectively. Such information are carried back to the CGDP by ACKs of stages from the destination sites.

Each LCT entry, LC_{ij}, shows the communication cost between sites i and j. At present, LC_{ij} is the number of hops in the shortest path from i to j. Here, $c(r{:}i \rightarrow j)$ is $|r|*LC_{ij}$, where $|r|$ stands for the size of a relation r.

A primitive unit of our algorithm is composed of the following parts: decision of next stage, reduction of the join query graph, and update of the QPI. Initially, all the nodes in the graph are marked FREE.

Suppose that a stage $r'{:}i \rightarrow j{:}r$ is selected as next one. The CGDP modifies the join query graph. First, it marks r′ SOUCE and r DEST in the graph. Then, join-links except one between r′ and r, each of which corresponds to a join-link incident on r′, are attached to r. If r′ has a result-link, it is also attached to r. In relation to such modification of the graph, the QPI/ATT is updated so as to meet the new scheme of r. Let us consider the join query graph in fig. 16a. Suppose that a stage, $R4{:}4 \rightarrow 3{:}R3$, is selected. Then, the graph is reduced to one in fig. 16b. A join-link, j_4', corresponding to j_4 is attached to R3. Thus, j_9 is a conjunction of j_4' and j_5. A result-link, o_2', corresponding to o_2 is also attached to R3.

On receipt of an ACK for the stage, $r'{:}i \rightarrow j{:}r$, from j, the CGDP trys to reduce the join query graph. First, it removes r′ and its related join-

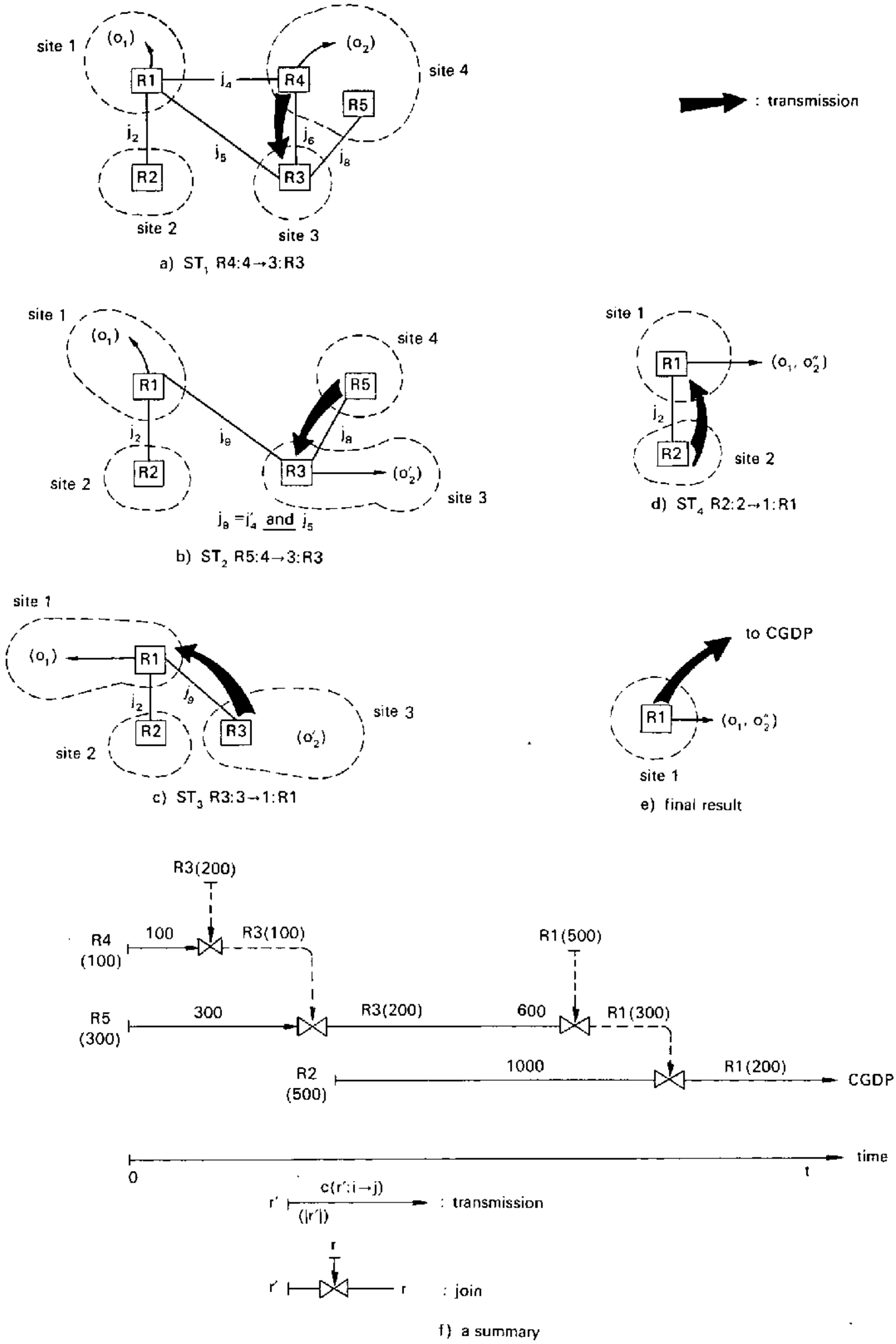


Fig. 16 An Example of the TS

links from the graph. The ACK carries the cardinality of the result relation of the stage. Then, the QPI relations are updated using information in the ACK. That is, all the tuples concerning r′ are deleted from these relations and the cardinality of r in the QPI/REL is updated by the new value carried by the ACK.

Next, we shall decide next stage. A stage, $r':i \rightarrow j:r$, that satisfies the following conditions is selected as the next stage:

1) r′ is marked FREE,
2) r is adjacent to r′ in the join query graph,
3) r is marked either FREE or DEST, and
4) $c(r:i \rightarrow j)$ is not only the minimum in the graph but also less than some threshold value (THV).

The 1st condition ensures that r′ is not being executed. The 2nd condition guarantees that there exists a join referencing r′ and r. The 3rd one ensures that, even if r is a destination of the other stages that have not completed yet, r′ can be sent to r. It means that transmissions of more than one stage can be overlapped in a parallel manner. Since transmission costs are overwhelming, we think these overlappings are effective. The last condition plays a role of protecting relations of larger size from being transmitted when relations of smaller size are currently being executed. If nodes with transmission cost ≤THV are not found, the CGDP waits for completion of stages being executed. If all nodes are marked FREE and no nodes satisfying this condition can be found, the THV value is reset using the QPI. At present, the THV value is determined to be the average size of nodes.

5 An Example of the transmission scheduling

Let us consider fig. 16. Suppose that all the initial local query processings have finished, i.e. all nodes are marked FREE, and the LCT and sizes of relations are given in figs. 17 and 18, respectively. Off course, the initial local query processing and transmission scheduling can be overlapped. Each LCT entry, LC_{ij}, represents a minimum hop number between i and j. Let the THV value be 500. Let ST_k and ACK_k be the k-th stage and its ACK, respectively. The communication costs with respect to join-links are calcuated as follows:

i, j = site numbers

	i	j	LC_{ij}	
	1	2	2	
	1	3	2	
	1	4	5	
	2	3	2	
	2	4	2	
	3	4	1	

Fig. 17 Logical Transmission Cost Table (LCT)

	LCS relations	sizes (in byte)	
	R1	500	
	R2	500	
	R3	200	
	R4	100	
	R5	300	

Fig. 18 The Sizes of Relations

j2: $c(R1:1 \rightarrow 2) = c(R2:2 \rightarrow 1) = 500*2 = 1000$

j4: $c(R4:4 \rightarrow 1) = 100*5 = 500$ $\quad \because |R4| < |R1|$

j5: $c(R3:3 \rightarrow 1) = 200*2 = 400$ $\quad \because |R3| < |R1|$

*<u>j6</u>: $\underline{c(R4:4 \rightarrow 3)} = 100*1 = \underline{100} < 500$ $\quad \because |R4| < |R3|$

j8: $c(R3:3 \rightarrow 4) = 200*1 = 200$ $\quad \because |R3| < |R5|$

Since R4:4→3:R3 has the minimal cost, it is selected as an ST_1 and the transmission and join commands are sent to 4 and 3, respectively. R4 is marked SOURCE and R3 DEST. The graph is modified as shown in fig. 16a. As an ST_2, R5:4→3:R3 is selected [see fig. 16b], because R3 and R5 are marked DEST and FREE, respectively, and $c(R5:4\rightarrow3)=300<500$ that is also the minimum. Here, R4 and R5 are transmitted in parallel.

Then, let us try to decide an ST_3. Here, only R1 and R2 are marked FREE. Costs for possible transmissions are as follows:

j2: $c(R2:2\rightarrow1)=c(R1:1\rightarrow2)=500*2$
$=1000>500$
j9: $c(R:1\rightarrow3)=500*3=1500>500.$

Hence, no satisfactory stage can be found. Since R3 is not marked FREE, we wait for ACK_1 and ACK_2. On receipt of ACK_1, R4 is deleted from the graph and QPI, and ACK_2 is waited for. On receipt of ACK_2, R5 is deleted and R3 becomes FREE. Suppose the size of R3 is 200. Since $c(R3:3\rightarrow1)=600>500$, the THV value is reset, i.e. $THV\leftarrow(1500+1000+500)/3=1000$. So, ST_3 is R3:3→1:R1 and executed [see fig. 16c]. R3 is marked SOURCE and R1 DEST.

Since R2 is FREE and $c(R2:2\rightarrow1)=1000$, R2:2→1:R1 is selected as ST_4 [see fig. 16d], and R2 is transmitted to R1. When both stages complete, the join query graph is reduced to one node graph [see fig. 16e]. Since it is a final result, it is transmitted to the CGDP.

Fig. 16e summarizes this example. The horizontal axis shows time.

V. The System Architecture

Fig. 19 shows the architecture of the global database processor (GDP) and local database processors (LDPs) for query processing. User's query is stated to the GDP at his site, i.e. CGDP. The CGDP takes it and translates it into global LCS queries using the distribution information. The ILQP creates LCS queries from the global LCS query, issues them to corresponding LDPs, and creates the join query graph. The TS issues T and J commands for executions of stages generated from the graph and controls their executions, monitoring their intermediate results. The T command, T (k, i, r′, j), means a transmission of the k-th stage, r′:i→j. The J command, J(k, i, r′, j, r, target-list, qualification, n), means that the source relation r′ to be received from i is joined to the destination r at j with respect to the target-list and qualification. The n is a size of r′, which is maintained in the QPI. By it, the LDP_j can allocate the working space for receiving r′. The target-list includes join-attributes of r′ with respect to its adjacent nodes except r and join-attributes of r with respect to its adjacent nodes except r′ along with the union of result-attributes of r and r′.

An LDP exists for one database. The LDP is composed of two main modules. The one is called the query translation [TAKIM80a]. It translates the LCS query written in QUEL into an executable sequence, e.g. DBTG DMLs, executes it, and stores the result as a relation in the working space (WS). The other is a WS manager (WSM). It is composed of four sub-modules, WS, JOIN, TRANS, and REC. The WS is a storage for storing intermediates. It will be implemented as a SAM file. The TRANS takes a T command from the CGDP and transmits the source relation to the destination. The REC of the destination also sorts it on a join-attribute while receiving it. The JOIN takes a J command and sorts the destination relation on a join-attribute. If the source relation is all received, the JOIN joins them, stores the result as the destination relation, and sends ACK with the information on the result to the CGDP. Since both relations are sorted already, they can be easily joined by means of merge-join technique [SELIP79]. Thus, we think it is easy to implement the WSM.

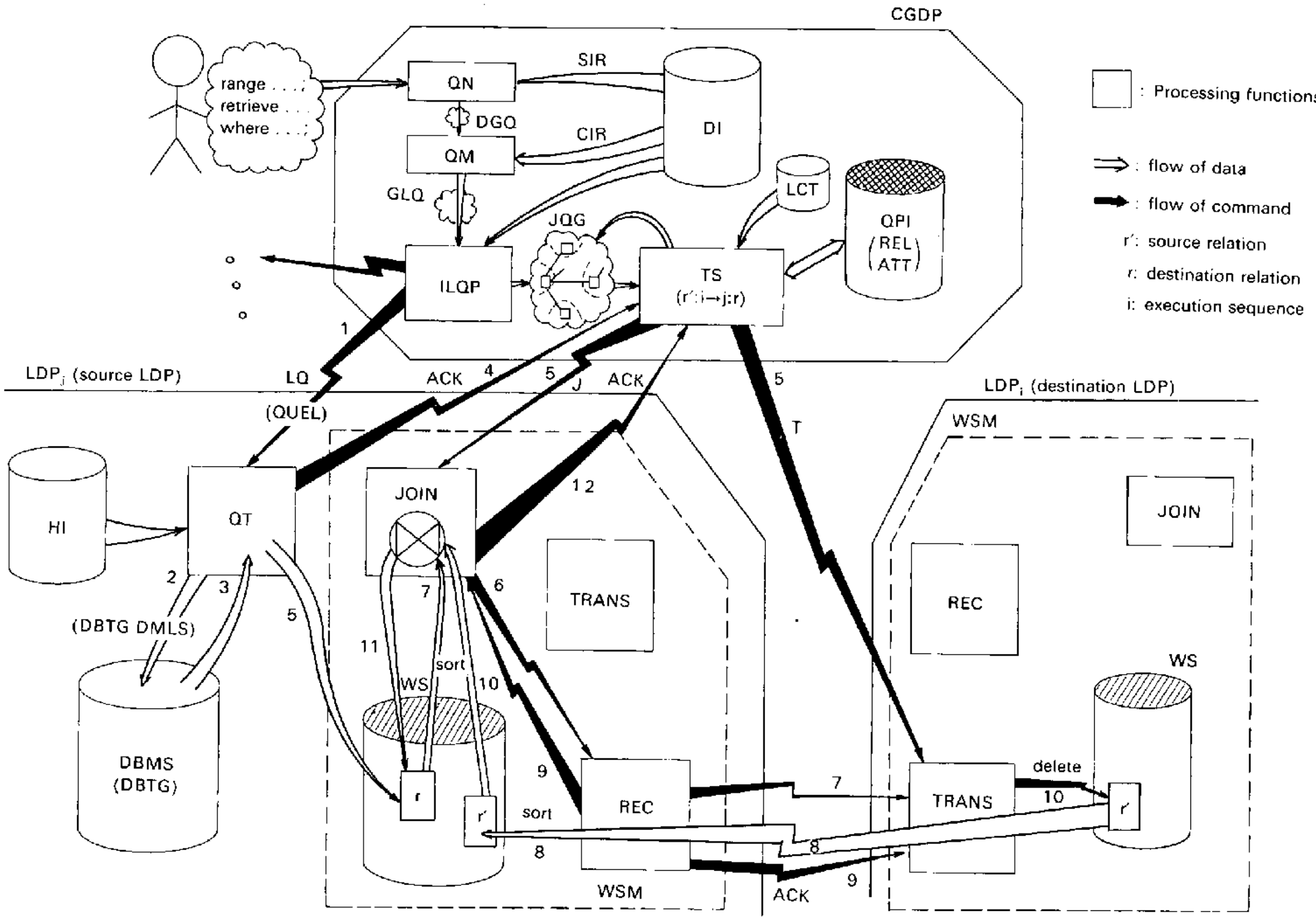


Fig. 19 The Architectures of the CGDP and LBPs

VI. Concluding Remarks

This paper presents our results in JDDBS-I project. In this project, we believed that we could make clear the basic concept of the distributed database systems which are characterized by global packet switching networks and existing large database systems. Furthermore, the common relational interface system to the CODASYL database systems is implemented. This system is useful not only for the distributed database systems but also the single system, because it can provide users with a means of accessing and describing database in a high level and non-procedural manner.

At present, we are developing the 2nd phase of JDDBS (called JDDBS-II). The characteristics of the JDDBS-II are as follows:

a) a local contention network like ethernet is used,
b) its main component is a small personal database system, in which allmost all of the data are dedicated to the user's application of the local site,
c) its components are heterogeneous, say CODASYL 78 and relational database system,
d) high intelligent DD/D facilities are provided for the system and users, and
e) the office information system (OIS) is considered as its most important application.

We are trying to realize such a system by 1985.

Acknowledgement

We would like to thank Mr. Makoto Suzuki, Systech Corp., for his most helpful and useful discussions. We would also like to thank Ms. K. Yamamoto, Mr. S. Koseki, and Mr. M. Ogawa, JIPDEC, for their helpful consultations and advices.

References

[ADIBM78]
Adiba, M. and Euzet, C., "A Distributed Data Base System Using Logical Relational machines," Proc. of 4th International Conf. on VLDB, Berlin, Sept. 1978, pp. 450–461.

[BERNP79]
Bernstein, P. A. and Goodman, N., "Full Reducers for Relational Queries, Using Multi-Attribute Semi-Joins," Proc. of the IEEE Computer Networking Symposium, Gaithersburg, Maryland, Dec. 1979, pp. 206–215.

[CHU W79]
Chu, W. W. and Hurley, P., "A Model for Optimal Query Processing for Distributed Data Bases," Proc. of the IEEE Compcon 79 Spring, Feb. 1979, pp. 116–122.

[CHENP76]
Chen, P. P., "Entity-Relationship Model—Toward a Unified View of Data," ACM TODS, Vol. 1, No. 1, Mar. 1976, pp. 9–36.

[CODDE70]
Codd, E. F., "A Relational Model of Data for Large Shared Data Bank," CACM, Vol. 13, No. 6, June 1970, pp. 337–387.

[EPSTR78]
Epstein, R., Stonebraker, M. and Wong, E., "Distributed Query Processing in a Relational Data Base System," UCB/ERL M79/18, Electronics Research Lab., UC. Berkeley, April 1978.

[HELDG75]
Held, G. D., Stonebraker, M. and Wong, E., "INGRES—A Relational Data Base System," AFIPS Conf. Proc., May 1976, pp. 409–416.

[HEVNA78]
Hevner, A. R. and Yao, S. B., "Query Processing on a Distributed Database," Proc. of the 3rd Berkeley Workshop on Distributed Data Management and Computer Networks, San Francisco, CA., Aug. 1978, pp. 91–107.

[ROTHJ77]
Rothine, J. B. and Goodman, N., "An Overview of the Design of SDD-1: A System for Distributed Databases," Proc. of the 2nd Berkeley Workshop on Distributed Data Management and Computer Networks, UC Berkeley, California, May 1977, pp. 39–57.

[SELIP79]
Selinger, P. G., et al., "Access Path Selection in a Relational Database Management System," Proc. of the ACM SIGMOD, Boston, MA., May 1979, pp. 23–24.

[STONM76]
Stonebraker, M., Wong, E., Kreps, P. and Held, G., "The Design and Implementation of INGRES," ACM TODS, Vol. 1, No. 3, Sept. 1976, pp. 189–222.

[TAKIM78]
Takizawa, M., Hamanaka, E. and Ito, T., "Resource Integration and Data Sharing on Heterogeneous Resource Sharing System," Proc. of the ICCC'78, Kyoto, Japan, Sept. 1978, pp. 253–258.

[TAKIM79]
Takizawa, M. and Hamanaka, E., "The Four-Schema Structure Concept as the Gross Architecture of Distributed Databases and Heterogeneity Problems," Journal of Information Processing (JIP), Information Processing Society of Japan (IPSJ), Vol. 2, No. 3, Nov. 1979, pp. 134–142.

[TAKIM80a]
Takizawa, M. and Hamanaka, E., "Query Translation in Distributed Databases," Proc. of the IFIP'80, Tokyo-Melbourne, Oct. 1980, pp. 451–456

[TAKIM80b]
Takizawa, M., "Distribution Problems in Distributed Database Systems: Integration and Query Decomposition," submitted for presentation, 1980.

[TSICD78]
Tsichritzis, D. and Klug, A. (eds.), "The ANSI/X3/SPARC DBMS Framework," Information Systems, Vol. 3, No. 3, pp. 173–191.

[WONGE77]
Wong, E., "Retrieving Dispersed Data from SDD-1: A System for Distributed Databases," Proc. of the 2nd Berkeley Workshop on Distributed Data Management and Computer Networks, Berkeley, May 1977, pp. 217–235.

[YAMAK75]
Yamamoto, K., "Computer Network System JIPNET," Proc. of the PACNET Symposium, Sendai, Japan, Aug. 1975, pp. 199–206.

News in Briet

Government and Information Organizations

JULY 1980

Nippon Telegraph and Telephone Public Corp. (NTT) carries out successful tests of its comprehensive digital satellite transmission system. "Sakura," a transmission satellite for experimental purposes, was used to transmit en bloc telephone, data and picture signals. NTT also successfully carried out experiments on undirectional multiple address transmissions.

CDC Japan started its Cybernet service on the 23rd in the Sunshine 60 building in Tokyo's Ikebukuro district, which represents a full-scale move into the field of computation for scientific and technical purposes.

CDC Japan has started work on developing a Japanese-language version of Control Data's (U.S.) educational service, PLATO.

NTT plans to use VLSI memories throughout for its computers, from 1981, and for its electronic switchboard, from 1982.

Maruzen starts sales of COMrac, an engineering design and analysis software package developed by Engineering Science's Data Unit (U.K.).

SDC Japan plans to offer, from August, aircraft and automobile information put out by America's Society of Automotive Engineers.

Marubeni Electronics has decided to invest capital in Applicon Corp. (U.S.), in order to strengthen its ties with that company.

NTT's packet switching service started on the 29th. Applications have been received from 15 users for a total of 33 lines.

AUGUST 1980

NTT, which together with the Captain System Development Research Institute, has been progressing with trial services of the Captain video information system, has started development of a system capable of retrieving information written in kana syllabaries.

Business International and Information Services International Dentsu on the 7th announced that from the 8th they would be starting the Bi-Metrics service on an on-line basis. This will be Japan's first foreign exchange market forecasting information service.

China's telecommunications organization and a KDD delegation invited to China for the 8-day period beginning July 30, and headed by Mr Masuda, KDD president, reached accord on six items for the improvement of Japan-China international telecommunication services.

NTT's Musashino Electrical Communications

Laboratory succeeds in developing the nation's first Japanese-language question answering system, allowing Japanese-language interaction with a computer. The ultimate aim for the future is an artificial intelligence computer.

Nihon Keizai Shimbun (Japan Economic Journal) started providing, in Japan, the news information retrieval service of the New York Times information bank. This service was started on September 8, which was when KDD started its international computer access service (ICAS).

NTT intends to apply in September or October to the Ministry of Posts and Telecommunications for permission to start Denwa-Fax Mini, its home facsimile service; NTT intends to start the service in the fall.

NTT is expanding the area covered by its digital data switched packet network. Within the next fiscal year it is intended to increase the number of cities covered to 25.

SEPTEMBER 1980

Kinokuniya, the book-store chain, started providing, on the 8th, Information Science's (U.S.) Chemical Information Systems, a data base of information related to toxicity, etc., the 8th being the day KDD's International Computer Access Service commenced.

Maruzen decides to reduce the charges for its Marunet service, which allows on-line utilization of Lockheed Corp.'s (U.S.) document information retrieval service, DLALOG; at the same time it was decided to continue the Marunet service even after KDD's International Computer Access Service starts up, and to operate it in parallel with ICAS.

KDD starts its International Computer Access Service on the 8th.

Computer Systems

JULY 1980

Fujitsu announced on the 30th the establishment of a semiconductor sales company in West Germany.

Toray joins hands with Apple Computer Corp. (U.S.) and enters the Japanese personal computer market.

Hitachi has been engaged in talks with BASF Corp. (W. Germany) on OEM exports of large computers. Basic accord has now been reached and an agreement signed.

Fujitsu has received an order from Australia's central bank, the Federal Reserve Bank of Australia, for a comprehensive on-line system having as its host machine a large computer.

NEC announced on the 7th that it has suceeded in obtaining an order from Indonesia Asahan Aluminium for the ACOS 350 system, NEC's advanced, medium-sized computer. It will be shipped in January 1981, and will be in operation from April.

Olivetti Corp. (Italy) announced that it will sell, in Italy, the OH5560 series, a new, high-performance computer designed and produced by Hitachi.

BASF's data technology division in the third quarter starts sales in Europe and South America of Hitachi's large computers.

Toshiba and NEC have reached agreement on the establishment of a specialized, jointly-financed sales company, Japan Office Automation. This move is designed to streng-then the two companies' office equipment business.

Hitachi reaches agreement with BASF (W. Germany) on OEM sales of Hitachi's M-170 medium-sized computer. This will be sold in Europe and South America as the BASF 7–60.

The computer industry is facing an ever greater shortage of software engineers, as computers become more widely used with the advent of the office computer, the microprocessor, etc.

Fujitsu's consolidated financial statement for fiscal 1979 shows sales of ¥599 billion.

Commodore Japan decides to market personal computers.

AUGUST 1980

Hitachi announced, on the 8th, orders received in the first quarter of fiscal 1980. These amounted to ¥573 billion, the highest total in the company's history.

Cray Japan reveals its plan to develop super computers, and also details its strategy for Japan. Development is proceeding with the Cray-2 and Cray-1S, the new machines, and there is a strong likelihood that completion can be announced around 1984.

Toshiba and Mitsubishi Electric announced, on the 11th, their orders received and sales for the first quarter of fiscal 1980. Toshiba had orders totaling ¥355 billion, a drop of 2% compared with the same period one year earlier, but Mitsubishi Electric's total of ¥374 billion represented a 25% increase.

Hitachi has decided to put a giant machine on the U.S. market, one equivalent to its giant Hitac M-200H generalpurpose computer, the largest and fastest such computer in the world. In line with this, Hitachi has started negotiations with America's National Advanced Systems.

Hitachi has reached agreement to supply BASF (W. Germany) with two machines, the M-200H and M-180, following on from the M-170.

Fujitsu will provide Edisa Corp. with technical assistance as part of its cooperation with Brazil's move to produce computers domestically. Assembly production of small computers will begin next January in Brazil.

NEC and NEC-TOSHIBA Information System start sales in early September of the ACOS-1000, the world's largest and fastest general-purpose computer.

NEC and NEC Honeywell Information Systems have reached accord on the dissolution of their joint-venture company, so the HIS will stop its management participation in NEC Honeywell.

SEPTEMBER 1980

IBM Japan announces on the 24th the 4341–2, a machine with 1.8 times the internal processing capability of IBM's mainline 4341–1 computer.

Toray announces on the 2nd the sales from the 8th of Apple Computer Corp.'s (U.S.)

Apple II J-Plus personal computer.

HIS, Japan moves into the Japanese-language information processing field. It plans to announce Japanese-language processing systems using the medium-sized computers level 6 Model 47 and downwards.

The Japanese office computer industry's market is certain to reach ¥220 billion in fiscal 1980.

Tandem Computers (U.S.) has started sales activities in the Japanese market, through Japan Tandem Computers. The emphasis is on the company's Tandem Non-stop System.

NEC announced on the 18th its decision to build a new, integrated IC production facility in Scotland. Within the year a new company, NEC Semiconductors, will be set up, a site aquired and work begun.

Hitachi intends to ship, to the European and U.S. markets, perhaps within this year, a giant computer with a performance equivalent to the company's giant M-200H computer.

Software and Computer Services

JULY 1980

Ministry of Posts and Telecommunications (MPT) decides to lower, from July 1, KDD's international charges, in line with the report of the Posts and Telecommunications Council.

Industrial Structure Council's Information Committee has started deliberations on a report, the first for six years. The members for three sub-committees, "Informationalization vision," "Computer Industry" and Information Processing Industry," have been decided.

MPT approved, on the 8th, NTT's application to use a collect call system, on a trial basis, form telegraphs for emergency purposes, and its public facsimile service, also on a trial basis.

NTT achieved a world's first with its successful digital satellite communications experiment, revealed on the 11th.

MPT approved, on the 15th, KDD's application related to its International Computer Access Service. Service will start in early September.

MITI made an announcement on the 28th concerning the development of a super-high-performance scientific and technological computer. Completion by 1990 will be the objective, and the computer will employ gallium-arsenic semiconductor elements and Josephson devices.

MITI, on the 29th, convened the general committee of its Industrial Structure Council, at Tokyo's Hilton Hotel. The Ministry's basic policies for fiscal 1981 were explained and opinions invited.

The Science & Technology and Economy Association has received general research and development organization aid and is to start studying the social and international effect of microelectronics.

AUGUST 1980

NTT is taking measures designed to allow American makers of communications equipment to participate in the Japanese private-sector market for telephones, facsimile equipment, in-house exchanges and other such privately run equipment. It was therefore decided, on the 31st, to make the technical standards involved easier to understand, and to publish English translations of such standards.

NTT's material procurements discussed in three days of U.S.-Japan working-level talks starting on the 4th. Taking part in the talks were U.S. Trade Representative Kirkland and GATT trade representatives in a party visiting Japan; the discussions were with MPT and NTT officials.

The Software Industry Association, in early September, are sending upper-echelon members to CSA, the U.K.'s industrial organization for software firms and computing centers. The policy decided on is to work towards regular exchanges of opinions in order to realize joint undertakings, in Southeast Asia, between the U.K. and Japan.

MPT, in fiscal 1981, plans to expand the scope of the trial operation of the Captains system, which was started last December. The number of terminals and screens will be increased.

MITI's Agency of Industrial Science and Technology has decided to develop a super-high-performance computer for scientific and technological applications. With a target completion date of 1988, the computer will be several thousand times faster than the largest general-purpose computer now in use.

The Software Industry Association inaugurates from the end of the month a special committee to study legal protective measures, in order to promote the legal protection of software.

SEPTEMBER 1980

NTT's procurement methods are the subject of U.S.-Japan working-level talks beginning from the 1st in Washington. The talks will last till the 4th and Japanese participants include Mr. Ugawa, Ministry of Foreign Affairs, Mr. Yamaguchi, NTT's Director of General Affairs, and Mr. Ikeda, head of the International Procurements Countermeasures Section. These are preliminaries to cabinet talks, and are between Mr Okita, the governments representative for external economic affairs, and Mr. Askew, the U.S. trade representative.

The Japan International Cooperation Agency decided on the 3rd, as part of Japan-China economic cooperation, in cooperation with China's National Science and Technology Committee to invite Chinese computer engineers to Japan for education and training. It was decided to have Fujitsu carry out this education and training.

MITI submits its outline budget relating to information processing to the Ministry of Finance. Included are ① research and development work on 5th generation computers, ② research and development into a high-speed scientific and technological calculation system, a major project of the Agency for Industrial Science and Technology, and ③ as a new financial measure, a leasing scheme to promote information processing.

JIPDEC

Japan Information Processing Development Center