Data Base Engineering

Vol. 1 No. 4 Contents December 1977

A Quarterly Bulletin published by
the IEEE Computer Society Technical Committee on
Data Base Engineering

Data Base Engineering Bulletin

A Quarterly Publication of the IEEE Computer Society
Technical Committee on Data Base Engineering

Data Base Engineering Bulletin is a quarterly publication of the IEEE Computer Society Technical Committee on Data Base Engineering. Its scope of interest includes: data structures and models, access strategies, access control techniques, data base architecture, data base machines, intelligent front ends, mass storage for very large data bases, distributed data base problems and techniques, data base software design and implementation, data base utilities, etc.

Contribution to the Bulletin is hereby solicited. News items, letters, technical papers, book reviews, meeting previews and summaries, etc., should be sent to the Editor. All letters to the Editor will be considered for publication unless accompanied by a request to the contrary. Technical papers are unrefereed.

Opinions expressed in contributions are those of the individual author rather than the official position of the TC on Data Base Engineering, the IEEE Computer Society, or organizations with which the author may be affiliated.

Membership in Data Base Engineering Technical Committee is open to IEEE Computer Society members, student members, and associate members. (Application form in this issue.)

IEEE COMPUTER SOCIETY

December 12, 1977

TO: TC/DBE MEMBERS

FROM: S. Bing Yao
Roles Committee Chairman
Department of Computer Science
Purdue University
West Lafayette, IN 47907

The Roles Committee was charged by the Chairman of TC/DBE to prepare a Bylaw proposal for this new Technical Committee. A meeting is planned for June 2, 1978 to finalize this proposal. A preliminary draft of this document is included in this issue of the Data Base Engineering Bulletin. The Roles Committee would appreciate your comments and suggestions. You may send your comments to me or to any of the Roles Committee members:

Murray Edelberg
Sperry Research Center
100 North Road
Sudbury, MA 01776

Michael Hammer
Dept. of Electrical Engineering
M.I.T.
Cambridge, MA 02139

Kai Hwang
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

Barry Housel
IBM Research Laboratory
San Jose, CA 95193

David Hsiao
Dept. of Computer and
Information Science
The Ohio State University
2036 Neil Avenue Mall
Columbus, OH 43210

Sham Navathe
Dept. of Computer Applications
and Information Systems
New York University
625 Tisch Hall
New York, NY 10003

LIMITED SELF-GOVERNMENT

Proposed Bylaws
# of the
Technical Committee
on
Data Base Engineering
# of the
Institute of Electrical and Electronics Engineers
Computer Society

Article I. Name

This organization shall be called the Technical Committee on Data Base Engineering of the Institute of Electrical and Electronics Engineers-Computer Society (IEEE-CS), hereinafter referred to as TC/DBE.

Article II. Purpose

1. This Technical Committee is organized and will be operated for educational and scientific purposes in the subject areas of design, application, and management of information handling systems.

2. The organization will promote the interest of professionals by:

   a. Affording opportunity for discussion of problems of common concern.

   b. Encouraging presentation of papers of special interest to this committee at national meetings of the IEEE-CS and at other special meetings organized by this committee.

   c. Publishing a bulletin containing information of interest to this committee.

   d. Other appropriate means. No addition may contradict the main purpose as stated in Article II-1 above.

Article III. Membership

Membership in TC/DBE is open to all IEEE members, student members, and associate members. Non-IEEE members may belong to TC/DBE. A non-IEEE member who is a member of TC/DBE may vote within TC/DBE. However, on any ballot, the effective value of the vote of the non-IEEE members eligible to vote shall not exceed that of the IEEE members eligible to vote at that time.

Article IV. Officers

1. The elective officers of this committee shall be:

   a. Chairman

   b. Vice-Chairman

2. The appointed officers of this committee shall be:

   a. Secretary-Treasurer

   b. Newsletter Editor

3. Each elective officer shall normally serve for a one-year term beginning July 1 of each year.

Article V. Duties of Officers

1. The duties of the Chairman shall include presiding at all meetings, appointing all standing and ad hoc committees, and such other duties as normally fall to this office.

2. The duties of the Vice-Chairman shall be to plan the program for all meetings and to assume any duties delegated by the Chairman.

3. The duties of the Secretary-Treasurer shall be to keep minutes of business meetings, maintain the roster of members, prepare budgets, report the committee's finances annually, file such other reports as are required by IEEE-CS, and maintain a knowledge of the financial situation of the committee. The Secretary-Treasurer shall file, at least annually, a roster of members with the IEEE-CS.

4. The duties of the Newsletter Editor shall be to organize and publish a bulletin periodically for the committee.

Article VI. Appointment of Officers

1. The Chairman shall appoint an election committee by December 1 in each year. This committee will nominate for each elective office at least two candidates who consent to serve. All candidates must be members of the TC/DBE. Candidates may also be nominated by petition of at least two TC/DBE members. The election committee shall certify that each candidate meets the requirements for the office for which he is nominated.

2. By March 1 of each year, the vote shall be initiated by mail ballot. An alternative procedure may be used if prior announcement is made by the election committee.

3. Of the ballots returned by the specified date, a plurality of votes for each office shall determine the winner. In the event of a tie vote, the office will be filled by appointment by the Board of Directors of the TC/DBE.

4. The appointed officers shall be appointed by the Chairman.

Article VII. Meetings

1. Meetings of national or regional character will be convened by the Chairman as he sees fit in order to serve the needs of the committee.

2. The committee may hold meetings only in places that are open to all members of the TC/DBE.

Article VIII. Board of Directors

1. There shall be a Board of Directors consisting of all officers and the previous Chairman.

2. The members of the Board of Directors shall serve for one-year terms, starting July 1 of each year.

3. The duties of the Board of Directors will be to advise the Chairman on all matters of interest to the committee, and to approve proposed annual dues and submit proposed bylaw amendments to the members of TC/DBE.

Article IX. Amendments

1. A resolution of the majority of the Board of Directors of the TC/DBE shall be sufficient to cause a bylaw amendment to be voted on by the TC/DBE members.

2. Any member of TC/DBE may submit an amendment to the Chairman; the Secretary shall determine whether a majority of the Board of Directors is in favor of proposing the amendment, and shall announce this determination.

3. A petition of 10% of the TC/DBE members shall be sufficient to cause a bylaw amendment to be voted on by the TC/DBE members. The right to petition shall be independent of any decision taken with respect to the procedures provided in paragraphs 1 and 2 in Article X.

4. The proposed amendment shall be voted on by the following mail balloting procedure:

a. The ballots shall be mailed out and returned to the TC/DBE Secretary. The ballot shall include (i) a copy of the proposed amendment including a specification of the date on which it will become effective; (ii) a copy of the article(s) in the existing bylaws that is (are) being proposed for amendment.

b. No ballot received by the TC/DBE Secretary postmarked later than thirty calendar days after the postmark of the last ballot mailed out shall be valid.

5. The amendment shall become effective if and only if the following two conditions are satisfied:

a. The effective number of valid ballots returned is greater than or equal to 25% of the total possible effective vote at the time the last ballot mailed out is postmarked.

b. A majority of valid ballots returned approve the proposed amendment.

Article X. Dissolution

In the event of the dissolution of TC/DBE, all assets and/or liabilities of the TC/DBE will be transferred to the IEEE-CS.

VERY LARGE INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES*

David K. Hsiao
Ohio State University

The Third International Conference on Very Large Data Bases was held in Tokyo, Japan, on October 6 through 8, 1977, and had a very large attendance representing twenty countries and five continents. The host country provided 275 attendees. There were sixty from the U.S., eleven from Germany, six from Sweden, three each from Belgium, France, and Malaysia, and one or two each from the other thirteen countries. It was truly an international eventwhich also coincided with Japan's Information Week.

The 3rd VLDB Conference also had a very large number of sponsors including ACM's SIGBDP, SIGIR, and SIGMOD, IEEE Computer Society's TC on Data Base Engineering, IECE (The Institute of Electronics and Communications Engineers of Japan), IFIP, IPS (Information Processing Society of Japan), Japan Special Research Project on Scientific Information Systems, and SMIS (Society for Management Information Systems).

The attractiveness and success of the Conference was largely due to the range and quality of its program. It was apparent that papers on very large data bases covered a broad range of computer science and engineering. There were papers on hardware for very large data base management and store (e.g., data base machines, terabit memory, and tertiary store), on software architectural issues (e.g., security, integrity, and recovery), on experimental software systems (distributed, interactive, multi-leveled and self-descriptive), on data base design tools, on physical data base structuring, manipulation, restructuring, on natural and user-oriented language interfaces, on theoretical treatments (e.g., study of imprecise data, precise dependencies and data models), and on applications (for medicine, music, statistics). There were over 120 submitted papers. With the help of nine program committee members and 130 referees, 44 papers were included in the program, of which seven were selected by the ACM Transactions on Database Systems (TODS) for publication consideration. The remaining 37 were published in the Proceedings of the Conference (570 pages) which may be obtained from the sponsoring societies. The editors and referees of TODS had some difficulty in restricting the number of selected papers

* With an apology to Howard L. Morgan (see CACM 18, 11; November, 1975, p. 670).

since there were many high-quality papers from the Conference. TODS accepted seven because it does not intend to devote more than one issue to the Conference and seven papers usually make up an issue of TODS.

In addition to contributed technical papers, there were sessions for survey and tutorial papers whose quality and range of coverage were good. There were two papers on data base machines, four on distributed data bases and systems, three on design aids and methodology, one on theory, one on directions in data base research. The selections of the panelists were first rate.

If one must single out a few individuals to credit the attractiveness and success of the Conference, this observer would attributed Stuart E. Madnick for his untiring organizational skill as the Conference Chairman in putting the Conference together and in getting a strong showing of U.S. participation; Tosio Kitagawa, the Honorary Chairman, for his prestige and for inciting his colleagues at Fujitsu to make local arrangements and schedules; Alan G. Merten for his care in handling and processing the contributed papers as the Program Committee Chairman; Tosiyasu L. Kunii for his promotion of surveys and tutorials as the General Chairman; and Hermann Schmutz for his leadership as the European Coordinator in contributing a heavy European participation. There were, of course, many known and unknown heroes in a large undertaking such as this one. Whether as an attendee or as an organizer, everyone associated with the Conference learned something. There was only one displeasure aired by some of the U.S. participants--the travel service for the Conference was ill-prepared and badly coordinated.

In echoing a successful conclusion for a promising beginning, Janis Bubenko, of Chalmers University of Technology in Sweden, and Bing Yao, of Purdue University welcomed the participants at the closing of the Conference to come to West Berlin in September of 1978. As organizers of the 4th VLDB Conference, they made it very obvious that the next VLDB Conference would be as attractive and successful as this one was.

A KERNEL DESIGN FOR A SECURE DATA BASE MANAGEMENT SYSTEM *

Deborah Downs and Gerald J. Popek

Computer Science Department
University of California at Los Angeles
Los Angeles, California 90024

ABSTRACT

The need for reliable protection facilities, which allow controlled sharing of data in multi-user data base management systems, is steadily growing. This paper first discusses concepts relevant to such protection facilities, including data security, object granularity, data independence, and software certification. Those system characteristics required for reliable security and suitable functionality are listed. The facilities which an operating system must provide in support of a such data base management system also are outlined. A kernel based data base management system architecture is then presented which supports value independent security and allows various grains of protection down to the size of domains in relations. It is shown that the proposed structure can substantially improve the reliability of protection in data bases.

INTRODUCTION

It is now becoming important to provide reliable, convenient protection facilities in multi-user data base management systems. This recently perceived need has been motivated by several developing trends. Data handling, rather than scientific calculations or language processing, now accounts for the bulk of computer activity, an estimated 85%.[24] Accompanying that growth has been a recognition of the desirability of integrated systems which maintain information and provide services for a heterogeneous group of users. The development of these systems will be further spurred by underlying technological growth in computer networks, decreasing processing costs, and large scale intelligent storage facilities. Data management systems have already been so successful that the proper functioning and sometimes the very existence of organizations depends on their continued operation. This fact is being reinforced by the developments mentioned above.

In short, large amounts of data are now stored in machine processable form. Some of the stored data is sensitive, in that unauthorized reference or modification would be quite undesirable. Further, the amount of such sensitive data together with the number of access facilities are increasing. For these reasons, protection in data management is of growing concern. In particular, the reliability of the protection facilities in preventing unauthorized access will be of increasing importance.

However, even if reliable data management protection facilities were not of immediate interest, it would still be important to consider the issues now, for reasons illustrated by the work in operating system protection.[18][19][2] There it was found that the conditions under which it was practical to retrofit reliable controls into existing systems were very limited [22]. In most cases either massive changes were necessary or it was advisable to start over with the design and implementation of the next generation operating system.

Further, in light of the trends already mentioned, it would not be surprising if the investment in data management software exceeded that in operating systems. Compatibility requirements, together with the large and growing investment in data management software, may make it rather costly to add reliable protection facilities later.

Today, there are few data management systems in existence that provide significant access control. In those few which do, the control often depends entirely on the supporting operating system. This current state of affairs is due to several factors: previous lack of a strong need for data base protection, insufficient understanding of how to provide reliable security, and the lack of a secure operating system on which to run a secure data base management system.

In the discussion which follows, we first examine some basic concepts important to data base security and then give a list of security related design requirements for a data base management system. Finally we present a design for a secure data base management system and discuss how it fulfills those design requirements.

CONCEPTS IMPORTANT TO SECURITY

Data Security

It is important to define what we mean by a secure data base management system. Basically a secure data base management system

----

*This research was supported by the Advanced Research Projects Agency of the Department of Defense under Contract DAHC-73-C-0368.

This paper is reprinted from the Third International Conference on Very Large Data Bases held in Tokyo, Japan, October 6-8, 1977, with the permission of the conference sponsors including the Association for Computing Machinery, 1133 Avenue of the Americas, New York, N.Y., 10036.

needs to support data security, the protection of data against unauthorized reference or modification. This support has two phases: authorization, defining who has access to which items; and enforcement, making sure the authorization rules are followed. The statement of authorization takes the form of protection data which defines the "users" and the security "objects" which contain the data to be protected, and also defines the possible actions the users may take with the objects. The enforcement is done by the security mechanism and usually takes place at the time the user attempts an access upon an object. The security mechanism checks the protection data and then allows or disallows the access.[10][21]

This basic data security facility should be flexible enough to support a variety of policies. A policy is a definition of the general security rules that the protection mechanism should support.[14] For example, a policy could be as simple as: "allow only users in the personnel office to access employee records" or it can be as complicated as the following medical data base example. There the policy may permit each patient to see his own records except for special annotations by the doctor. The dietitian is allowed to see only those domains pertinent to diet, the accounting department can see what lab tests and operations were performed but no medical history or results from tests, the research personnel can see statistics over all the patients but no individual patient's data and so forth. By setting the protection data in a particular manner many policies could be implemented in a "discretionary" manner although some policies may also require some non-discretionary changes to the security mechanism. The term discretionary here means that changes can be made without recoding; that a mechanism for changing the policy is provided as part of the protection system.

Granularity

The grain of protection wanted in a data base management system can be rather fine.[4] In operating systems the grain of protection is usually in terms of pages or files. In data base management systems one may want to protect a person's salary and rating while giving more free access to the rest of his employment records. This means that, in relational terms, we wish to protect at the level of domains. In data base management systems we may further wish to provide statistical access and value dependent control. Statistical access gives the user access to aggregates of the data without having access to the data itself. For example, the user may be able to have access to information on what percentage of the patients in a hospital are there for elective surgery but not be able to see individual patients' medical history. Value dependent control is based on the value of the data.[5] One authorization for value dependent protection might be "user A may see employee records of those employees earning less than $10,000". This type of control requires protection at the level of data elements in domains. Since the operating system does not provide protection at these low levels it is necessary for the data base management system to supply its own support.

Data Independence

Data base management systems are usually not static mechanisms. They must be adapted to new types of data, new semantic views of data by the users, new storage hardware, new access methods and so forth. In order to provide this adaptability without being forced to recreate the data base management system software, data independence is a recognized necessity.[6][8] This concept divides the data base management system into levels, based on increasingly abstract views of the data.[26] For example, a three level system may have a logical level, which is the users view of the data; a internal level which may be a system view of the data unconnected to any physical representation; and a physical view that is the actual physical organization. Stored mapping tables are used to translate requests from one level to another. When changes are made at one level only the mappings to the other adjacent level(s) are affected. Data independence therefore allows the user to look at the data as logical entities and not worry about their physical representation.

For the protection mechanism data independence means that the specification of the protection data must be done in terms of logical entities, since the user knows nothing about their physical counterparts. But the enforcement is most reliably done at access time on the physical entities, since for certification the checking of an operation should be done as close as possible to its action. Therefore the protection data must be translated through the mappings to specify physical objects. But this translation process can involve most of the data base management system, a very large amount of code. To provide a reliably secure data base management system it is clearly necessary to assure the correct operation of that mapping system, or provide some other more direct path from the logical to the physical representation of data entities.

Certification

Certification gives some form of guarantee that the modules being certified work correctly. This might in principle be done, for example, by exhaustively debugging every possible data flow path or by proving the program correct. As of now one of the most reliable ways of assuring the correct operation of a program is to use program verification methods, but currently this task is very difficult. When certification was taken into consideration during the design process, modules up to 2000 lines of code have been verified, but with considerable difficulty and months of work.[23]

If the security mechanism is not certified there is no guarantee that it works correctly, and therefore little confidence that enforcement is assured. Therefore, a systematic form of assurance is highly desirable.

In operating systems the concept of a kernel design was introduced in order to improve the reliability of the system. It had the added advantage of reducing the relevant code for certification. It was noticed that previously, because the security relevant code was scattered throughout the system, the correct behavior of many operations had to be assured which really were not security relevant. So all security relevant code was separated out and isolated in an operating system nucleus, or security kernel. In operating systems the security kernel is the nucleus of the system and runs in the most privileged state.

In data base management systems the role of a security kernel can be played by several separate modules containing all security relevant code which must be certified. The design that follows is based on the concept of a security kernel, to allow certification and increase the reliability of the protection system.

SECURITY REQUIREMENTS FOR A DATA BASE MANAGEMENT SYSTEM

It is important in any design to set down a list of requirements for the system so that when the design is finished it is possible to make some measure of the validity of the design. The list of requirements presented here are only those relevant for security.

1. The protection architecture must be simple and involve only a limited amount of software for its proper functioning.

2. The protection constraints must be described at the logical level in order that data independence be maintained.

3. The desired data base management system functions, such as integrity, locking, recovery, performance measurements, and so forth, must be adequately performed.

4. The operational cost of protection enforcement must be low; degradation of the system must be minimal.

5. The reliability of the protection system must be capable of being certified.

Virtually any secure data base management system suggested by this list of requirements clearly requires a secure operating system on which to run. The operating system must support data security, and in particular:

1. provide authentification of the user,

2. allow only the data base management system access to the database, and

3. allow no unauthorized alteration of the data base management system code.

Without the support of a secure operating system it is almost impossible to have a secure data base management system.

A SECURE DATA BASE MANAGEMENT SYSTEM KERNEL DESIGN

The following design, which is under development at U.C.L.A. on a secure Unix testbed, supports various grains of data security protection down to the size of domains in relations. The problems in reliably supporting value dependent control appear considerably more difficult and therefore the design only supports value independent control as an important first step.

Let us first look at a high level view of the system. Aside from the user software there are three main modules in the system, the Kernel Input Controller (KIC), the base kernel, and the Data Management Module (DMM). The DMM need not be certified but must run isolated in its own execution environment with no direct contact with the user of the data base. The DMM contains all the non-security related portions of the data base management system and presumably with only a few changes could run alone as a non-secure data base management system. The KIC and the base kernel are really two pieces of the security kernel and therefore must be certified. A high level description of the functions of such data base management system kernel modules include:

1. The kernel must assure at all levels of the data base management system a secure association between the data and its identification. If this association is not maintained there is no way to reliably determine access rights to the data.

2. Since the protection data is stated at the logical level and the actual access is taking place at the physical level, some secure translation process is necessary in order to compare the physical object to be accessed to the logical object specified in the protection data.

3. The kernel must enforce the security policy on each access based upon the values in the protection data.

In order to see how the kernel design performs these functions, the operation of both retrieval and update are presented. Figure 1 shows a retrieval operation. In this figure the three main elements of the design can be seen: the Kernel Input Controller, the base kernel, and the Data Management Module. As already outlined, in order for the protection mechanism to be able to check a user's authorization for access, it is important that the identification of the data be correct. It is the KIC's function to supply this correct identification to the base kernel.

A user makes a request for retrieval of a certain piece of data and the KIC parses the request and retains the request type and the logical name. But the logical name specified may not be the system wide logical name but merely a part of the user's schema. To reduce the number of logical entity names used to identify the data the KIC also parses the data definition language and builds the tables necessary to translate a user schema name to a system wide logical entity name. The KIC then passes to the base kernel the type of command, a system wide logical entity name and the user identification supplied by the secure operating system. Meanwhile the request also is sent to the DMM. The DMM performs all the normal functions of a data base management system: it makes the translations from one level to the next, follows access paths, chooses the access methods, records performance and usage statistics, does any necessary locking, and so forth.

RETRIEVAL OPERATION

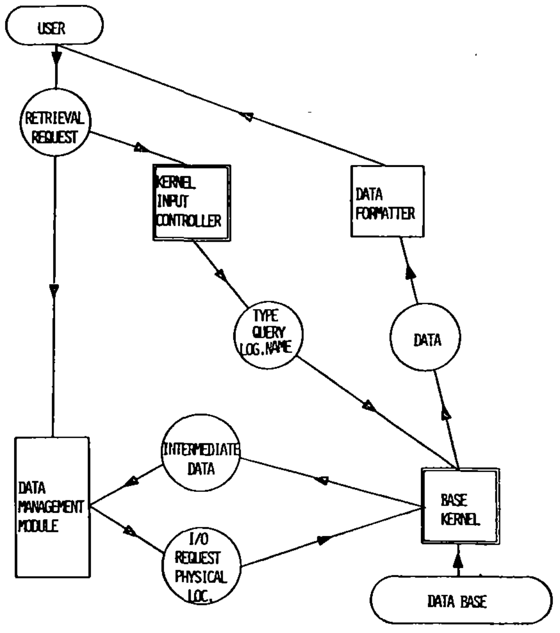

FIGURE 1
NOTE: RECTANGLES ARE MODULES & THOSE WITH TWO OUTLINES ARE TO BE CERTIFIED
CIRCLES ARE DATA.

Eventually the DMM needs to access the data base itself, but the DMM will not be allowed direct access to the data base since it cannot be assumed correct. Instead, it prepares a read command specifying the physical location that is to be accessed and that command is passed to the base kernel.

The base kernel is the only module that actually accesses the data base. Before allowing access to the information in the data base the base kernel must check to see if the user is allowed to access this particular data entity. The request from the DMM can include both physical location parameters and the logical entity name. Of course none of these parameters are trusted so far as security enforcement is concerned, since to do so would imply that the DMM is trustworthy. Recall that the protection data maintained by the base kernel identifies the objects by their logical entity name. Therefore some form of reliable mapping is still necessary between the DMM's physical identification and the protection data's logical identification.

In this design a physical to logical mapping is used which takes the form of a tag on each piece of separately protectable data in the data base.[11] This tag is maintained entirely by the base kernel and the DMM has no knowledge of it. The tag is an encoded form of the logical entity name of the data and its size depends on the number of distinct logical entities to be protected. The information received from the KIC: the logical entity name, the command type and the user identification, is used by the base kernel to check if the specified user is allowed access to the logical entity. When the read command is received from the DMM the base kernel reads the data along with its tag and then, using that tag, checks to see if the logical entity is the same as was specified by the KIC. The data requested by the DMM may be merely an intermediate result, such as information necessary for a relational "join"; if so the tags may not agree and the data is given only to the DMM. But if the information is a final result for the user and the tag retrieved from the data base matches the tag prepared from the information supplied by the KIC, the base kernel returns the results to the user without allowing the DMM access to the data.

The data to be returned to the user may have to be translated from the form used in the data base to the user schema's format. The correct operation of this transformation is not necessary for data security purposes, but it is necessary that no other users' data influence the operation of the data formatter. To achieve this isolation it should be confined.[19] Confinement provides an environment where only data authorized by the protection data is available to the process. In other words the data formatter will run in its own isolated execution environment. Note that the operation that parses the data definition language and prepares the data which the data formatter uses must also be run under the same constraints. After the data is formatted in terms of the users schema, it is delivered to the user workspace.

Now let us look at an update operation, as shown in Figure 2.

UPDATE OPERATION

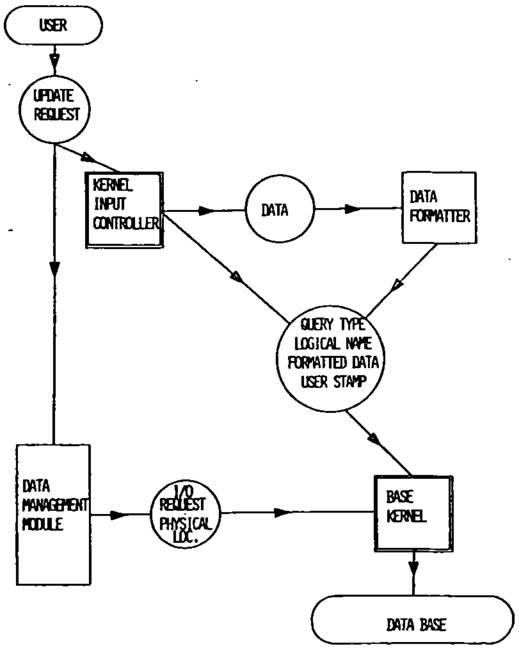

FIGURE 2
NOTE: RECTANGLES ARE MODULES & THOSE WITH TWO OUTLINES ARE TO BE CERTIFIED.
CIRCLES ARE DATA.

When the request comes from the user the KIC parses it, looking at this stage for any requests which would change the protection state of the data base, such as creates, updates or deletes. Whenever the KIC recognizes such a request it copies the logical name, the data, and the command type. As stated before in retrieval, the logical name must be translated to the system wide logical name. This translation must be done correctly for security to be maintained.

The data in general also must be translated from the user schema view to the form maintained in the data base. Packing is just one such example. This step is done by the untrusted data formatter in a manner similar to that described above in retrieval, but in reverse. Thus the KIC passes directly to the base kernel the system wide logical entity name, the translated data, the command type, and finally the user identification as supplied by the operating system.

Meanwhile the request has also been handled by the DMM analogously to the retrieval case. The DMM eventually is ready to execute a write on the data base and it prepares a request specifying the physical location. The base kernel has already used the information from the KIC to check the user's update access to the logical entity name in the protection data. When the base kernel receives the request from the DMM selecting the location to be updated, it reads the data location and checks the tag to be sure that the DMM has pointed to the correct logical entity or on a create to see if the space is really available. If the location is correct and if the user is allowed update access to the logical entity, the data that was passed by the KIC is written with the correct tag attached. On a create request the base kernel would have created the tag from the information received from the KIC.

This design supports value independent data security because access authorization is checked on any attempted access to the data base and the mechanism doing the checking can be certified. Also the identification of the data is always secure since the kernel maintains the association between the data and its identification through all levels.

The addition of the KIC in retrieval increases the reliability of the system but is not necessary for data security since without the KIC the only possible mistake is for the user to retrieve data which was not requested but to which the user had access. The untrusted DMM selects which data is to be retrieved and without the KIC the base kernel can only check for legal access. But if the KIC is used to pass the logical entity name to the base kernel, it can also check if the access is to the correct logical entity.

It should be recognized that despite the importance of data security, and the belief on the part of many that more complex forms of security control are unnecessary, a policy such as value dependent protection can be useful.

For example, the kernel design presented here enforces data security. However, value dependent errors can occur. During updates and creates it is still possible for the DMM to point to the wrong location, such as the wrong tuple but the right domain. The base kernel cannot prevent such errors. For example, the user might ask to update the Salaries of all employees in Department 44, but instead the DMM selects the Salaries of all Departments. If the user is permitted read access to Departments and write access to Salaries the update would be legal. But although writing in the wrong location destroys or damages data it does not breach value independent data security, either in this example or in subsequent uses of the data. Access to the data by any unauthorized user, would still not be allowed since the data is still identified by its correct tag. In fact, to support data security the base kernel need not check to see if the DMM even selects the correct logical entity, although to do so does increase reliability. This principle also applies to the base kernel correctly matching the request from the DMM to the information from the KIC. Since the data and the tag come from the secure KIC a mismatch does not result in a loss of data security because the correct tag is attached to the data by the base kernel before any update or create is stored in the data base. But such a mismatch would decrease the reliability of the system.

Details of the Design

Consider the Kernel Input Controller and the base kernel in more detail. Figure 3 shows the KIC with its main functions being parsing of the query language and parsing of the data definition language in order to maintain the name translation tables.

KERNEL INPUT CONTROLLER

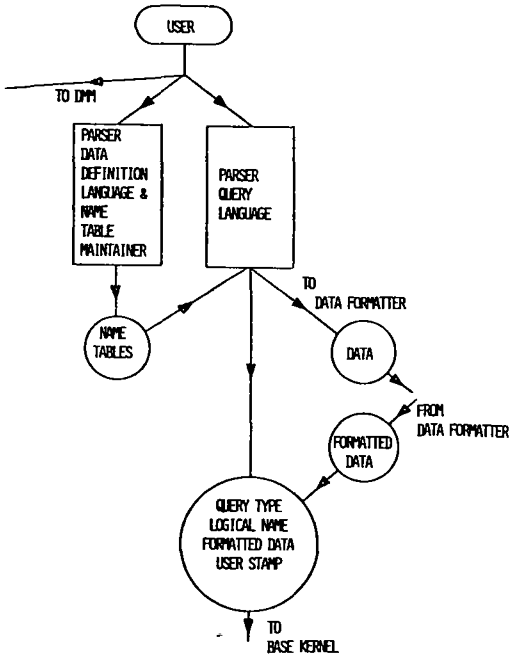

FIGURE 3
NOTE: RECTANGLES ARE MODULES TO BE CERTIFIED AND CIRCLES ARE DATA.

The functions of a secure updater for the protection data are also managed by the KIC. The commands to update the protection data could be normal update commands with the base kernel doing some extra checking, or it could be a separate language with the KIC doing the extra parsing. The complexity of the KIC, and therefore the ease of its certification, depend on the complexity of the user interface.

Figure 4 shows the base kernel with the data that it must handle. In the data base the protection data will be maintained in a manner similar to all the other data. The DMM can keep usage statistics about the protection data as it does with all other data and it can set the protection data's organization and do reorganizations whenever the DMM decides it is needed. Each entry of the protection data will also have a logical entity name tag with some kind of designation as protection data. The protection data specifies the user and the accesses allowed, and could include a standard length list of users who are allowed to update this piece of protection data.

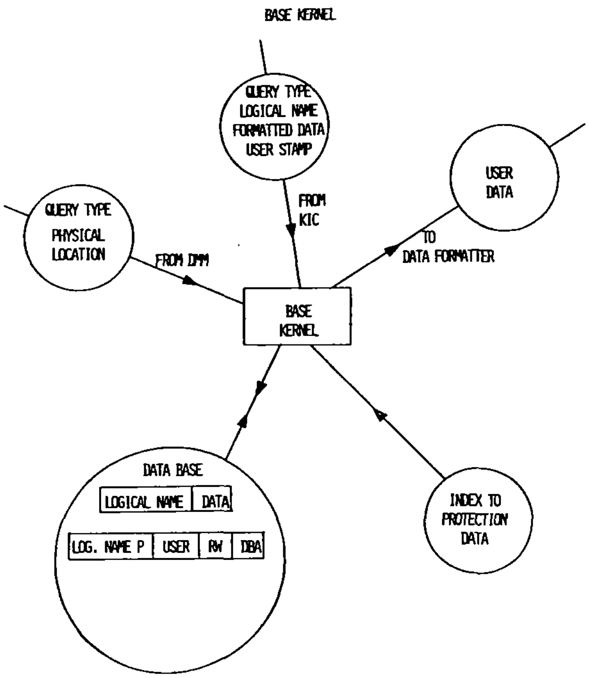

FIGURE 4
RECTANGLES ARE MODULES TO BE CERTIFIED AND CIRCLES ARE DATA.

A given user ("owner") and the data base administrator would be common entries in that list. This design of the protection data could support either a centralized or a decentralized control of the data.

It is also necessary for the kernel to maintain an index to the protection data both for quick access purposes, since the protection data must be checked on each access to the data base; and also to maintain general data security. The data security problem is illustrated by the following example. Suppose a piece of protection data exists stating what access user A has to object 1 and a user wishes to create a new piece of protection data stating new access rights for user A to object 1. The DMM could select a new location for the create instead of the location where the piece of protection data now exists. If the base kernel could not detect the error, two copies of the protection data for user A to object 1 would exist and there would be no guarantee that the newest one would be used for authorization of access requests. But the index allows the base kernel to check the existing protection data and prevent any such duplicates from being created.

DISCUSSION

It is important to see if the requirements specified for the design have been satisfied and what the cost is for doing so. First, the size of the kernel depends on the complexity of the user interface for the KIC, since the base kernel itself should be small. Both of their operations are relatively simple and straight forward. The kernel as a whole, if the user interface is at the level of complexity of the user interface of most present data base management systems, is small and simple enough to be reliably certified. The protection data is stated at the logical level and physical accesses are securely translated to the logical level for checking, so data independence is maintained. The security kernel should not affect the operation of the other desired functions of a data base management system.

The cost of the enforcement should be minimal, since it only requires some extra parsing, which could also be used by the DMM, and some extra accesses to the protection data. If computation must be done on the data internal to the data base management system, the computation modules must also run in their own isolated execution environment. Lastly, the storage for the data base has been increased by the size of the logical entity name tags and by the protection data.

Thus the kernel design detailed above does appear to fit the list of requirements previously specified. An added advantage to this design is that it should be able to be added to some existing data base management systems. The base kernel replaces all I/O to the data base and the KIC is merely a monitor on communication from the user. The original data base management system would largely be run isolated in its own execution environment. Of course the format of the data would have to be changed to include tags and the protection data created.

Limitations: Confinement

The protection system detailed above does not prohibit the existence of certain covert channels. Since the DMM may retain statistics and data from many users, it may be possible to manipulate the DMM in such a way as to reveal another user's data through covert channels.[19] For example, while the protection system does assure that only legal data will be directly given to a user, it cannot assure that the DMM will not vary the amount of legal data and leak bits of information. Suppose a malicious user had found an error in the uncertified DMM that he could manipulate. If the query is made by a user for the names of employees who have brown hair, the DMM could, in order to leak some illegal information such as psychiatric records retained from a previous legal query by another user, return 20 names to indicate a 0 bit and return 21 names to indicate a 1 bit. The kernel checks if the malicious user is allowed to see employee names and hair color but the kernel cannot know exactly how many correct answers there are without duplicating much of the DMM code.

It should be noted that these types of covert channels would be very difficult for the user to manipulate. Since the DMM itself is protected by the operating system, the DMM code is assumed unalterable and therefore an existing flaw in the system would have to be used. If one can assume that the programmers who originally coded the DMM were trustworthy the flaw would have to be an accidental bug. Further, the user has a rather restricted interface into the DMM: normally only certain commands to retrieve, delete, update, create, and so forth. These conditions are likely to make any covert channel difficult to use and of small bandwidth.

There is however a possible method for restricting even these covert channels through an analysis of data flow.[7] Using a compile time analysis the DMM code could be scanned for instances of actions based on a previous user's data. It may be possible to recode to remove these channels.

Other Approaches

Other approaches to providing a secure data base management system are taken by System R and Ingres.[1][13][27] They are both examples of protection mechanisms implemented and enforced strictly at the logical level instead of the physical level. In System R protection constraints may be stated in terms of relations and domains or in terms of "views". Authorization to these constructs is checked at the logical level and then the data base management does the necessary translations to implement the query. In Ingres the security mechanism is in the form of query modifications. The security constraints pertinent to a request are "anded" to the particular query and then the request is handled by the data base management system. In both of these systems the reliability of the security mechanism depends on the correct operation of much of the data base management system and would require certification of most of it.

The advantage of System R and Ingres is that they support value dependent control. Since value dependent control is based on the value of the data any portion of the data base management system that could affect that value is security relevant, which is most of the data base management system. Therefore System R and Ingres's control at the logical level does not appear to make the certification problem any more difficult in the value dependent case.

CONCLUSION

Until recently, little attention has been given to the issue of how best to integrate protection facilities into data management systems. Reliability considerations especially seem to have received inadequate attention but the potential architectural impact could be significant. It is believed that the preceding discussion supplies a solution to the important value independent form of security in data base management systems, and is a steo toward a more complete solution.

Bibliography

1. Asthrahan, M.M., et.al. "System R: Relational Approach to Database Mnagement," ACM Transactions on Database Systems, Vol. 1, No. 2, pp. 97-137, June 1976.

2. Browne, Peter S., Dennis D. Steinauer. "A Model for Access Control," Proceedings. 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, San Diego, Ca., November 1971.

3. Chamberlin, Donald D., Raymond F. Boyce, Irving L. Traiger. "A Deadlock-Free Scheme for Resource Locking in a Data-Base Environment" , IBM Research RJ 1329, December 1973.

4. Chamberlin, D.D., J.N. Gray, I.L. Traiger. "Views, Authorization, and Locking in a Relational Data Base System," Proceedings of National Computer Conference, Anaheim, California, May 1975.

5. Conway, R.W., W.L. Maxwell, H.L. Morgan. "On the Implementation of Security Measures in Information Systems," Communications of the ACM, Vol. 15, No. 4, April 1972.

6. Codd, E.F. "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Vol. 13, No. 6, pp. 337-387, June 1970.

7. Denning, Dorothy E. "A Lattice Model of Secure Information Flow," Communications of the ACM, Vol. 19, No. 5, pp. 236-242, May 1976.

8. Date, C.J., P. Hopewell. "File Definition and Logical Data Independence," Proceedings. 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, San Diego, Ca., November 1971.

9. Feistel, H. "Cryptography and Computer Privacy," Scientific American, Vol. 228, No. 5, May 1973.

10. Fernandez, E.B., R.C. Summers, C.D. Coleman. "An Authorization Model for a Shared Data Base," Proceedings. of ACM-SIGMOD Workshop on Data Description, Access and Control, San Jose, California, May 1975.

11. Friedman, T.D. "The Authorization Problem in Shared Files," IBM Systems Journal, Vol. 9, No. 4, 1970.

12. Graham, G. Scott, P.J. Denning. "Protection - Principles and Practice," AFIPS Conference Proceedings, SJCC, Vol. 40, pp. 417-429, 1972.

13. Griffiths, Patrica P., Bradford W. Wade. "An Authorization Mechanism for a Relational Database System," ACM Transactions on Database Systems, Vol. 1, No. 3, pp. 242-255, September 1976.

14. Hartson, H.R., D.K. Hsiao. "A Semantic

Model for Data Base Protection Languages," Systems for Large Data Bases, pp. 27-42, September 1976.

15. Held, Gerald, Michael Stonebraker. "Storage Structures and Access Methods in the Relational Data Base Management System Ingress," Proceedings. ACM-PACIFIC-75 San Francisco, Ca., November 1971.

16. Hinkf, T.H., M. Schaefer. "Secure Data Management System," working paper TM-(L)-5407/007/00, System Development Corporation, June 1975.

17. IMS/360, General Information Manual, GH20-0765, IBM Program Product, 1974.

18. Lampson, B.W. "Protection," Proceedings. Fifth Princeton Symposium on Information Sciences and Systems, Princeton University, pp. 437-443, March 1971.

19. Lampson, B.W. "A Note on the Confinement Problem," Communications of the ACM, Vol. 16, No. 10, pp. 613-615, October 1973.

20. Minker, Jack. "Performing Inferences over Relation Data Bases," Proceedings. of ACM-SIGMOD Workshop on Data Description, Access and Control, San Jose, California, May 1975.

21. Owens, Richard C. "Evaluation of Access Authorization Characteristics of Derived Data Sets," Proceedings. 1971 ACM-SIGFIDET Workshop on Data Description, Access and Control, San Diego, Ca., November 1971.

22. Popek, G.J. "Protection Structures," IEEE Computer, pp. 22-33, June 1974.

23. Ragland, L.C. "A verified Program Verifier," Ph.D. Thesis, University of Texas, Austin, 1973.

24. Senko, M.E. "Information Storage and Retrieval Systems," Advances in Information Systems Science, Vol. 2, edited by J.T. Ton, Plenum Press, New York, p. 230, 1969.

25. Senko, M.E., et.al. "Data Structures and Accessing in Data-base Systems," Vol. 12, No. 1, pp. 30-93, 1973.

26. Snuggs, M., G. Popek, R. Peterson. "Data Base System Objectives as Design Constraints," Proceedings of the ACM, San Diego, California, pp. 641-647, November 1974.

27. Stonebraker, M., E. Wong. "Access Control in a Relational Data Base Management System by Query Modification," Proceedings of the ACM, San Diego, California, pp.180-186, November 1974.

TC/DBE MEMBERSHIP APPLICATION/RENEWAL FORM

To become a member of the TC/DBE and be on the mailing list for the Data Base Engineering Bulletin, please return this form or a copy of it to:

IEEE TC/DBE
Department of Computer Science
University of Illinois
Urbana, IL 61801

NAME ______________________________
(please print)

INSTITUTION ______________________________

ADDRESS ______________________________

____

Areas of Interest: ______________________________

Level of
Participation: ( ) Read newsletter only. ( ) Work on workshops and Symposia.

( ) Help with newletter.

CALL FOR ABSTRACTS IN DATA BASE ENGINEERING

★★★★★★★★★★★★★★★★★★★★★★

DBE, as a service to its readership, is now publishing short statements from members of the research and development community about what they and/or their groups are doing in Data Base Engineering and what they consider important topics for future research and development.

The purpose of this activity is to promote a general awareness within the community served by DBE of what is being done and/or what is planned, where it is being done, who is involved, and how to get additional information.

Individual project abstracts will be published on a continuing basis. Priority in publishing the abstracts will generally be determined by the date of receipt of the material, at the discretion of the Editor, and as space available within each issue of DBE allows.

Each individual Abstract should include at least the following information presented if possible within these named categories:

1. <u>NAME OF PROJECT</u>: Use an existing name. If it is not self-descriptive append more explanatory terms.

2. <u>ORGANIZATION(S)</u>: State the organization and/or sponsor of the project. Include mailing address.

3. <u>PERSONNEL</u>: Give the name of the Principal Investigator and/or other technical personnel, and include mailing addresses if different that above. Telephone numbers are optional.

4. <u>KEYWORD(S)</u>: Give short phrases or keywords that indicate subareas within Data Base Engineering.

5. <u>DESCRIPTION</u>: A paragraph or two describing the project, its objectives, novelity longevity, technical approach, outcomes, current status, etc.

6. <u>IMPLICATIONS</u>: A paragraph or two indicating what's expected, what the future may bring, what problems there might be,etc.

7. <u>REFERENCES</u>: A few key citations as appropriate to (5) and (6).

Each statement should be fitted into a 6.5" X 10.0" page frame minimizing white space from the bottom of the form upward. The photo-ready copy should be sent flat to the Editor of DBE: Dr. Jane Liu, Department of Computer Science, University of Illinois, Urbana, Illinois 61801.

Please circulate this Call for Abstracts in Data Base Engineering among your colleagues and others who should contribute.

PLEASE DUPLICATE AND POST

★★★★★★★★★★★★★

• RESEARCH PROJECT ABSTRACT •

1. NAME OF PROJECT: Data Structure Architectures (DSA)

2. ORGANIZATION: Department of Computer Science, University of Minnesota,
114 Lind Hall, Minneapolis, Minnesota 55455.

3. PERSONNEL: Wolfgang K. Giloi
Helmut K. Berg

4. KEYWORDS: Computer architecture, general-purpose computing, data structures, machine data model, ordered sets, parallel processing.

5. DESCRIPTION: The DSA concept defines a class of general-purpose architectures which provide increased hardware support for the implementation and manipulation of data structures.
Operational principle: An intermediate data structure, called the basis and managed by hardware, is defined by a machine data model. Arbitrary data structures can be defined in terms of basis elements (multi-dimensionally ordered sets). Basis elements are represented by variable descriptors. They are referenced through descriptor identifiers and can be processed in response to single machine instructions.
A representative DSA and a respective operating system have been designed as a generalization of the STARLET system. The next stage of the project includes the development of a simulator for the measurement and evaluation of the designed DSA.

6. IMPLEMENTATIONS: With the DSA concept, the programmer is only concerned with the mapping of a data structure onto the basis, whereas the mapping of the basis elements onto consecutively stored data vectors is handled by hardware. The impact of this principle on software generation is being investigated.
The Modular internal DSA information structure lends itself in a natural way to parallel processing, implies a high degree of hardware orthogonality, and facilitates hardware support to the operating system. Implementations of the concept through different architectural concepts will be explored.

7. REFERENCES: Giloi, W. K., "STARLET - Das Konzept eines interaktiven Kleinrechners für Array Verarbeitung", Rechnerstrukturen, Proc. IBM Symposium on Computer Architecture in Wildbad, 1973, R. Oldenburg Verlag, Munich, 1974, pp. 172-198.

Giloi, W. K., Berg, H., "STARLET - A Computer Architecture Based on Ordered Sets as Primitive Data Types", Proc. 2nd Annual Symposium on Computer Architecture, Houston, 1975, CAN, Vol. 3, No. 4, Dec. 1974, pp. 201-206.

Giloi, W. K., Berg, H., "Data Structure Architectures - A New Approach to Parallel Processing", Proc. Int. GI-IMACS Symposium "Parallel Processing - Parallel Mathematics", Munich, March, 1977, North-Holland Publ. Co., 1977.

Berg, H. K., "A Computer Architecture Based on Ordered Sets as Primitive Data Entities', Ph.D. Thesis, Department of Computer Science, University of Minnesota, June, 1977.

Giloi, W. K., Berg, H., "Introducing the Concept of Data Structure Architectures", Proc. Int. Conference on Parallel Processing, Detroit, August, 1977.

• MEETINGS OF INTEREST •

♦ ACM SIGMOD International Conference on Management of Data
May 31 - June 2, 1978
University of Texas at Austin

Sponsor: ACM SIGMOD

General Chairmen: Alfred G. Dale
Nell B. Dale
Dept. of Computer Science
University of Texas
Austin, TX 78712
(512) 471-4353

♦ Third Berkeley Worshop on Distributed Data
Management and Computer Networks
May 31 - June 2, 1978
Berkeley, California

Submit papers (max. 5000 words), short papers or abstracts (max. 1500 words) by February 15, 1978 to the program co-chairman:

Steve Kimbleton
National Bureau of Standards
B-212 Technology Building
Washington D.C. 20234

♦ International Conferences on Data Bases:
Improving Usability and Responses
August 2-4, 1978
Technion, Haifa, Israel

Sponsor: Technion, ACM

Program Chairman: Ben Shneiderman
Dept. of Information Systems Management
University of Maryland
College Park, MD 20742
(301) 454-2548

CALL FOR PAPERS

&

REGISTRATION FORM

FOURTH WORKSHOP ON COMPUTER ARCHITECTURE FOR NON-NUMERIC PROCESSING

to be held at

Syracuse University Minnowbrook Conference Center
Blue Mountain Lake, New York
in the Adirondack Mountains of New York State (135 miles from Syracuse)

August 1-4, 1978

SPONSORED BY: ACM: SIGARCH, SIGIR, SIGMOD
in cooperation with
IEEE-CS: TCARCH, TCDBE, and
Syracuse University

WORKSHOP DESCRIPTION:

In the past, this Workshop has been a primary avenue for those actively engaged in research and development of a variety of specialized non-numeric systems to present information regarding their current activities and to discuss future directions.

To continue this activity, we invite papers on current or proposed work in all areas of computer architecture for non-numeric processing, such as: data communications, information storage and retrieval, data base management, command and control, artificial intelligence, distributed processing, searching and sorting, and text processing.

Presentations will last from 30 to 45 minutes, with half the time devoted to giving the paper and half to discussion. In addition, the final session will consist of a number of short presentations for those wishing to present or discuss a concept, but unable to prepare a full length paper.

INSTRUCTIONS TO AUTHORS:

Authors are invited to submit four (4) copies of their paper to the Program Chairman by March 31, 1978. Papers should be approximately 20 pages in length and include a 100-word abstract. Consideration will also be given to extended abstracts of at least 1,000 words and which include appropriate references and figures. All submissions will be acknowledged and authors will be notified of acceptance no later than May 20, 1978.

PUBLICATION:

Full length papers of exceptional quality may be considered for publication in ACM Transactions of Database Systems. It is anticipated that the proceedings will be published as a special joint issue of the Newsletters of the sponsoring groups.

REGISTRATION FORM

Fourth Workshop on Computer Architecture
for Non-Numeric Processing, August 1-4, 1978

NAME________________________________

AFFILIATION_________________________

ADDRESS_____________________________

CITY________________________________

STATE____________________ZIP_______

TELEPHONE___________________________

MAIL THIS FORM AND YOUR CHECK TO:

Mary Jo Fairbanks
Department of Electrical & Computer Engineering
111A Link Hall of Engineering
Syracuse University
Syracuse, New York 13210

Minnowbrook Conference Center
Blue Mountain Lake, New York

REGISTRATION DEADLINE IS:

July 15, 1978

PLEASE CHECK ONE:

___Enclosed is $50.00 deposit

___Full Payment Enclosed ($150)

PLEASE CHECK BELOW:

___I will need transportation to Minnowbrook from Syracuse.

___Travel arrangements are not complete. I will notify you.

___I am driving and can take passengers. Please specify number._____

REGISTRATION - There will be no on-site registration for this Workshop. Everyone must pre-register. The total fee for this Workshop is $150.00. This includes registration fee ($30), meals, lodging, and use of all Minnowbrook facilities. A deposit of $50 is payable with the return of your registration form. The balance is payable on August 1, 1978. Refunds for cancellation will be honored until July 15, 1978. Please make checks payable to: FOURTH WORKSHOP ON COMPUTER ARCHITECTURE FOR NON-NUMERIC PROCESSING.

MEALS, LODGING, & CLOTHING - The first meal will be Dinner on Tuesday, August 1, at 6:00 p.m. Please try to arrange your schedule to arrive at Minnowbrook by that time. The last meal will be served at Noon on Friday, August 4th. IF YOU HAVE ANY SPECIAL DIETARY REQUIREMENTS, LET US KNOW AT YOUR EARLIEST CONVENIENCE.

We regret that due to the tax exempt status of Minnowbrook, families or friends of registrants cannot be accommodated. A list of nearby commercial resorts will be provided upon request.

Dress informally and comfortably. Nights are often chilly so bring a sweater.

TRAVEL - Transportation will be provided between Syracuse and Minnowbrook for those who need it and arrive in Syracuse before 2:00 p.m. on August 1st. Notify M. J. Fairbanks, (315) 423-3511, as soon as your travel plans are complete so a ride can be scheduled for you.

If you drive, a brochure will be sent to you with your letter of confirmation which will provide sufficient directions to the Minnowbrook Conference Center.

Those departing from Syracuse after the Workshop should arrange their travel schedules to allow for departure after 5:00 p.m. on Friday, August 4th.

COMMITTEE:

General Chairman: P. Bruce Berra, Industrial Engineering & Operations Research, Syracuse University, Syracuse, New York 13210. (315) 423-2826.

Program Chairman: Lee A. Hollaar, Department of Computer Science, University of Illinois, Urbana, Illinois 61820. (217) 333-3162.

Program Committee: David K. Hsiao, The Ohio State University
Robert Korfhage, Southern Methodist University
Glen L. Langdon, International Business Machines-San Jose
G. Jack Lipovski, University of Texas
Stewart A. Schuster, University of Toronto

Publications Chairman: Michael McGill, Information Studies, Syracuse University, Syracuse, New York 13210. (315) 423-4522.

Local Arrangements: Mary Jo Fairbanks, Electrical & Computer Engineering, Syracuse University, Syracuse, New York 13210. (315) 423-3511.

FOR FURTHER INFORMATION PLEASE CONTACT COMMITTEE MEMBERS LISTED ABOVE.

Very Large Data Bases

CALL FOR PAPERS

FOURTH INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES

BERLIN, GERMANY SEPTEMBER 13-15, 1978

This conference is the fourth in a series. Like its predecessor, the conference is intended to identify areas of general interest for research, development, and applications of data base systems. It objectives are to promote an understanding of current undertakings, to further the exchange of experience gained in the construction and use of data base systems, and to serve as a forum for future research and development. We are especially interested in high quality papers in the areas of data base design, data base software engineering, distributed data base systems, and the impact of new technologies.

The VLDB conference has been scheduled to coincide with the International Congress of Data Processing. As a result it can be expected that many practitioners will attend the VLDB conference. Thus practically oriented survey and tutorial papers are especially welcome. Intensive debates among practitioners and researchers will be organized.

TOPICS OF INTEREST

Suitable topics include, but are not limited to:

DATA BASE DESIGN
system analysis
requirement specification
logical data base design and integration

DATA BASE SOFTWARE ENGINEERING
formal specification
design methodology
development tools
verification

DISTRIBUTED DATA BASES
network architecture
resource management
program conversion
data migration

DATA BASE COMPUTER ARCHITECTURE
back-end machines
microprocessors
memory organization
storage technologies

DATA SEMANTICS AND MODELING
concepts
formalism
consistency

SYSTEM IMPLEMENTATION
query evaluation
integrity and recovery
concurrent access
access control mechanisms

USER INTERFACE
graphic interface
very high level languages
natural language interface

PERFORMANCE EVALUATION
measurement
monitoring
simulation and analytic modeling

DATA BASE APPLICATIONS
in decision making
in health and environmental systems
in office automation

WHERE TO SUBMIT PAPERS

Send *five* copies of each full paper by March 1, 1978 to one of the following persons:

U.S. Program Committee Chairman
Prof. S. Bing Yao
c/o Computer Applications and Information Systems
New York University
40 West 4th Street
New York, New York 10003

European Program Committee Chairman
Prof. Janis A. Bubenko, Jr.
Chalmers University of Technology
Department of Computer Sciences
Fack, S-402 20 Goteborg, Sweden

IMPORTANT DATES

March 1, 1978	Papers due
May 18, 1978	Authors notified of acceptance of the papers
June 15, 1978	Final revisions of papers due
September 13-15, 1978	Conference meets in Berlin

PUBLICATION

A conference proceedings will be published, and selected conference papers will be published in the *ACM Transactions on Database Systems* (TODS). All papers accepted for presentation will be available to the participants at the conference.

MORE INFORMATION AND TRAVEL GRANTS

Suggestions for panel or tutorial sessions should be directed to one of the two program chairmen. Requests for other information should be directed to one of the conference chairmen. It is hoped that some partial travel grants will become available to help support the travel of conference participants.

General Conference Chairman
Dr. Herbert Weber
Hahn-Meitner-Institut
Bereich Datenverarbeitung und Elektronik
Glienicker Str. 100
1000 Berlin 39

European Conference Chairman
Professor Claude Delobel
Computer Laboratory,
University I of Grenoble
Boite Postale 53
38041 Grenoble Cedex
Grenoble, FRANCE

or

U.S. Conference Chairman
Professor Anthony I Wasserman
Section of Medical Information Science
University of California
San Francisco, CA 94143