CONFERENCE PAPER

NO. 15 (1990)
BRANZ DATABASE PROGRAMME

C.C Gan and A.H. Dechapunya

Presented at DECUS 1990 Conference
14-16 August 1990
Wellington
New Zealand

BRANZ DATABASE PROGRAMME

C.C. Gan and A.H. Dechapunya
BRANZ
Private Bag, Porirua, New Zealand

The aim of the BRANZ Integrated Information Technology programme has been to investigate the potential of computing technology to improve the management and transfer of data and knowledge.

This paper describes the BRANZ database programme which looks at the application of information technologies such as relational database and advanced programming tools for representing data. The programme also look at the human interactions with the data stored in the computers.

At present, BRANZ computing environment is a VAX/VMS based local area network with RDB, RALLY, C, PROLOG, FORTRAN, DECWindows, and CLASS. The CLASS language has been developed by a joint project between BRANZ and Auckland University.

The paper outlines a number of database systems including: REFERENCE, FIREPROP(MP9), PUBLICATION, SLIDE and SITE VISIT.

So far, BRANZ has concentrated on delivery of textual information in databases and knowledge-based systems. A desirable next step is the inclusion of graphics presentation and windowing in such systems.

The paper also discusses various issues involved in the interfacing between RDB databases and graphics files.

INTRODUCTION

BRANZ is an industry Research Association funded primarily by the New Zealand building industry by means of a levy on new building. The BRANZ mission statement is: "To promote better building through the application of acquired knowledge, technology, and expertise".

BRANZ is in the information business and has the daunting task of providing information to a large and diverse industry. It is an industry which is often disinterested because like most other industries it is already swamped by information. It is necessary therefore to be able to provide the information that is needed, when it is needed, in an acceptable form.

BRANZ has been accumulating a large resource of building knowledge expertise and information, as well as an understanding of the market needs for this information.

Database technology has come of age. Database systems and supporting software have become more widely available. The greatest flexibility is offered by Relational Database Management Systems (RDBMS). It has only been in recent years that advances in hardware and software technology have made the implementation of relational databases viable.

BRANZ Database work has been established with the aims to evaluate database tools, and to apply these to develop an integrated database environment for use by the building industry and the public. It has produced two database systems available to the building industry, REFERENCE and FIREPROP(MP9) and a few others for BRANZ internal use. REFERENCE provides references and online ordering of technical information to the building industry. It covers reports from building research and information bodies worldwide. FIREPROP(MP9) is a database containing the SANZ document "MP9:1987 Fire properties of Building Materials and Elements of Structures". Great care has been put in the development of users interfaces for these two systems.

BRANZ computing environment consists of high-speed communication equipment connecting computers and printers, terminals and plotters. The network provides user-oriented computing requirements to all BRANZ staff. It contains:

PDP 11/24	- Real time machine; primarily for experimental work.
VAX 11/750	- Information machine; provides on-line information systems.
MicroVAX 2	- Management machine; used for general management and planning activities.
MicroVAX 2	- Computing machine; used for scientific work and statistical analysis.
VAXStation 3100	- Research and Development machine.

The established hardware environment comprises terminals connected to the BRANZ local area network.

Databases, once properly developed, provide the following advantages over the paper version:

* Reliable retrieval of the information needed
* Tremendous flexibility as to how the information can be accessed

The users of databases include: builders; consulting engineers in the civil, structural and mechanical disciplines; architects; local government engineering and building inspectors; manufacturers of building products; universities and other tertiary educational institutes; libraries; and BRANZ staff members.

This paper discusses the various issues in the application of database technology to the dissemination of information.

DATABASE APPLICATIONS

REFERENCE

CAIRS (Computer Assisted Information Retrieval System) is a commercial software package designed for maintaining bibliographic databases. It provides a comprehensive range of search and retrieval functions. BRANZ required a library search and loan request system suitable for access by untrained users. The system had to be easy to use, presenting options and providing assistance when needed. Direct use of the CAIRS system was found unsuitable. A user who wishes to obtain data from the database has to go through the process of formulating a query as a series of CAIRS search and retrieval commands.

Because no work was being undertaken elsewhere on providing a user friendly search and retrieval environment for accessing CAIRS with VAX/VMS, or any other machine, an in-house project was initiated.

The approach chosen was to write a user interface to CAIRS. A front end was implemented in VAX/VMS FORTRAN. The front end accepts commands and queries at a high level of abstraction. Lower level commands are issued to drive CAIRS which is running as a background slave process. Communication is two-way. The output from CAIRS is monitored and information is indirectly or directly passed back to the user.

The inter-task communication inherent in this approach presented problems in VMS because of the need to co-ordinate execution of the front end and CAIRS. Asynchronous buffering and processing of output from CAIRS is required to maintain an acceptable level of performance for the front end.

The present version of the front end is being used for search and retrieval of the BRANZ bibliographic database, REFERENCE, which contains some 28,500 records (June 1990). It is run on a VAX-11/750 information computer for use by both BRANZ staff and external users. Initial response to the front end has been very encouraging. Novice users can quickly browse the database to locate and request publications. Even experienced CAIRS users prefer to make use of the front end for search and retrieval.

Some of the major features of the front end are:

* An interface that allows a user to concentrate on search concepts, rather than the details of search and retrieval commands.
* The menu presentation of options and on-line help to provide guidance for users.
* Automatic selection and presentation of information from the inverted index file and thesaurus to assist in searches. This results in considerable savings by eliminating the time required to manually browse these information sources.
* A Display mode that allows users to select only relevant titles located by a search.
* An electronic loan request and library messaging system which allows immediate request of titles.

FIREPROP(MP9)

The FIREPROP(MP9) was the result of the co-operative effort between BRANZ and the Standards Association of New Zealand (SANZ). It is an on-line implementation of document "MP9:1989 Fire Properties of Building Materials and Elements of Structure". The MP9 document contains tables of approvals, text, numerical data and diagrams.

MP9 had until recently been manually maintained and distributed as a hardcopy document. Subscribers to MP9 were posted hardcopy supplements prepared by SANZ. They followed updating instructions, adding, replacing and removing pages to merge the supplement and create a new edition of MP9.

Each approval in a table in MP9 document typically contains fields giving an approval code, approval and expiry years, fire ratings, product names, manufacturers, and product descriptions.

The representation must maintain an integrated mixture of text and tables of approvals. These tables occur in a wide range of formats. Moreover, the structure of the document (defined through the table of contents) must itself be changeable.

To summarise, the physical data implementation had to support the following objects, subobjects, and operations:

Objects
* Tables of approvals
* Paragraphs of text
* The table of contents

Subobjects:
* text
* approvals
* fire resistance ratings
* early fire hazard indices
* proprietors
* table headers and footers
* items from the table of contents

Operations:
* Browsing and look-up. The table of contents can be browsed and the corresponding data retrieved.
* Search and Retrieval. A selection can be set up to look for data.
* Construction of approval tables. When formatting approvals for display or printing we must be able to collect all the data associated with a table and present the data in a suitable tabular format.
* Ordering, i.e. preserving the overall structure and sequence of MP9
* Update. It must be easy to create, alter or delete any objects or subobjects in the database.
* Security measures; to guard against accidental erasure or corruption of the data, and to restrict access to authorised personal only.

A relational database model expresses a data model in terms of relations and the relationships between them. Approvals, Fire Resistance ratings, etc have been described as basic objects.These basic objects can be represented as relations in a relational database. Thus, basic objects of the MP9 Document can be represented in a relational database system. Relationships can be used to express composite objects and the structure of the MP9 database. For example, Proprietors and Fire Resistance Ratings are related to an Approval. The relationships between basic objects in a common section are used as the basic building blocks for constructing an approval table.

Text in the database was stored in free format and without indexing. This was adequate for our purposes, however a keywording or indexing system would be required in a situation where large volumes of text have to be searched.

The system was implemented using Rdb, Rally, DCL, and C.

FIREPROP(MP9) will benefit members of the building industry who want fast, efficient and flexible browsing and retrieval of up-to-date information from the MP9 document.

One of achievements of the work is the development of techniques for representing a structured document. The knowledge gained could be applied to other structured documents, for example the proposed New Zealand Building Code.

PUBLICATION

The PUBLICATION database provides stock control of publications to enable more effective scheduling of printing resources, and to make available to BRANZ staff in general, information relating to the building industry.

The Publication Database Consists of:

* Comprehensive records of about 15,000 organisations and individuals who work in areas related to Building and Construction in New Zealand.
* A list of the BRANZ publications available for distribution.
* Stock inventory, and distribution information on these publications.

The Publication Database System provides:

* A database for storing information on BRANZ audiences.
* The ability to search and identify sectors of the building industry on a variety of criteria.
* The ability to construct mailing lists for a selected audience.
* The ability to monitor production and distribution of BRANZ publications.
* A data entry system that allows authorised staff members to update the database.

The database is used to select mailing lists of the targeted audiences by vocation, postal zone, city etc., according to the information required. It will then record against individual entries what publications were supplied and when. This information will relate directly to the stock control inventory, so that we may have an accurate record of the number of units used, how many remain in stock and what re-order level to set to maintain stock. It may also record orders from users for multiple copies, where a charge is involved.

The system was implemented using Rdb, Rally, DCL, and C.

SLIDE AND SITE VISIT

The application developed enables users to browse, search, retrieve slide cataloguing and site visit report data and let selected staff update data. The system has a keyword dictionary and an author list. These are used to maintain keyword indexing and photographer/author indexing of slides and site visit reports.

Staff in the Advisory group spend a significant amount of time managing their information resources. The members of the group are often acting in a librarian role. They retrieve information for others because they do not have a system that can be reliably used by other staff. The problem is worsening as their information resources increase.

Site visit reports are prepared by a staff member following a site inspection. The "Memorial" copy is held in the Advisory Office. The slide library collection is in constant use as a staff resource for talks, presentation, displays and publications requiring methodical storage and cataloguing.

Consider the site visit reports. There are currently about 850 on file and they are being filed at rate of about 10 per week. Examples of the type of information which could be extracted include: a history of all problems seen by BRANZ at a site; all visits relating to a building material; finding problems with a common cause.

The information should be instantly available to all staff who have the need. It should be up to date, and there should be no need to involve Advisory staff members in a librarian role.

The system was implemented using Rdb, Rally, DCL, and C.

WHAT HAVE WE LEARNT?

Environment

The computing environment must be flexible and capable of high performance.

A flexible environment is one in which the products can be distributed easily and conveniently throughout the BRANZ local area network. In a high performance system, the combination of hardware and software must be such that it will not be a bottleneck in the development.

BRANZ environment has evolved to become multidisciplinary due to the nature of our work. For each project, a working group (programmer, subject expert, user representative, and project manager) was formed to monitor the progress of the project and to provide a forum for discussion and exchange of information.

Hardware

The hardware provides the framework for teamwork, where all people involved in the development of applications have access to the systems at any time. This is especially useful for incremental checking of all phases of system development. Compatibility and networking capability between the development and delivery systems is essential, so that the end product can be transferred to the delivery system without modification.

The present research and development environment comprises terminals connected to the BRANZ local area network. It is being proposed that the terminals should be replaced with more intelligent stations. There are two basic reasons for this change. The first is that databases development requires large computing resources and the sharing of central processing units has created some problems both for programmers and other users of the system. The second is that there is a need to improve the users interfaces of the applications by the use of graphics and windowing. Thus, clearly, there is a need for workstation technology.

Software

VAX Rdb has been used to successfully implement databases at BRANZ. It was however necessary to keep the number of indices small, as each index consumes copious quantities of disk storage.

The search/browse/retrieval systems and data entry system can be implemented by Rdb built-in commands. However, the tool is not friendly enough for the user to perform searching, retrieving and updating operations.

VAX Rally has proven to be a good tool for developing the data entry system. It provides a prepackaged set of solutions to interface to the database, to enforce constraints, to design menus, report and screen handling, but it does not provide a flexible environment for searching. RALLY does not, as of yet, support a sufficiently powerful query language (procedural or non-procedural) to handle the range of queries that the search and retrieval systems have to be capable of supporting.

Conventional third generation language 'C' is a satisfactory tool to develop the search and retrieval system.

Database Content

The FIREPROP(MP9) project has highlighted the need to have a subject domain with a well defined boundary. A subject domain should also have a high degree of technical information reliability, a good presentation level, available human experts, and a reasonable life expectancy.

User

Before development of an application begins, it is important to have a clear idea of who the users will be and how they will use the system. The system must be seen to offer a significant benefit to users. The possible benefits are an enhancement in performance over other methods with respect to:

* speed
* cost
* effort
* reliability
* problem-solving capability
* flexibility

Communication between users and developers of information systems is important. The MP9 project illustrated the need for a clear idea of the objectives of an application and its intended audience.

THE PROBLEM AND THE NEED

Based on users' experiences, there are some limitations in the present database systems, such as:

* Users need to go to a source other than the computer for diagrams

  For example in the FIREPROP(MP9), all diagrams are stored in the FIREPROP(MP9) user manual. If users want information about the fire resistance rating of the design stress ratio for a particular beam, they have to look up sheets of paper in the manual for the Design Stress Ratios For Solid Timber Beams graphs.

  The same problem occurs in the SLIDE system. It only stores pointers to the appropriate slides. The users may have to go through boxes of slides in order to view the particular slides.

* No update facility for diagrams and graphs.

  For example, in the FIREPROP(MP9), at present, all diagrams and graphs are stored in hardcopy. The whole diagram or graph has to be redrawn when SANZ updates information in the diagram or the graph. Subscribers to the FIREPROP(MP9) are posted hardcopy supplements prepared by SANZ. They follow updating instructions to merge the supplements and create a new edition of user manual.

Text is only adequate for collecting and generating simple information, eg: printed texts, letter, and figures. It is not adequate for spatial and graphical information, for example , maps or plans. BRANZ database systems could be improved through the use of :

* Pre-prepared graphic images

  Static graphics or pre-prepared graphic images will do away with the process of having to look for supplementary diagrams on papers, or slides. The system will automatically select and display the appropriate diagrams.

* Generating graphic images

  Dynamic graphic images will allow users to draw diagrams by entering information about the diagrams. It will also allow users to make sure that diagrams have been drawn correctly and to view the diagrams.

WHAT ARE WE DOING?

It is clear that the next step in our work is to investigate the integration of graphics into the BRANZ database environment.

As an initial phase of our study, we want to develop a prototype that can provide communication between relational database and graphics files, and a relational database that can represent a building element.

Some Previous Work on Graphic Database

Database Management Systems Integrate With Autocad Drawing (Van Der Roest, 1989)

Within Autocad, drawings are viewed as files with objects and their characteristics. Within database management system (DBMS), data files are viewed as a collection of records. Each record is composed of data fields. To map a CAD drawing file into a database file format with records and fields, is to view all drawing file objects as database file records and the object characteristics as the record's fields. So, a LINE is viewed as a record with its X axis position, Y axis position and LAYER values serving as fields.

This concept offers an ability for CAD drawing data to be a part of some common data pool. As a result, other applications could take advantage of the drawing data, and vice versa. Besides this benefit, users could also manipulate, modify, and analysis the drawing data.

Indexing of drawings and images in ARC/INFO (ESRI,1989)

A Geographical Information System (GIS) can be thought of as a sophisticated extension of a DBMS. A DBMS, with its simplest form, has the ability to define, analyse and retrieve textual and numeric data. GIS extends the ability (of a DBMS) by extending these data types (textual and numeric) to include spatial and locational data and to manipulate and analyse map data in a graphical form.

To extend GIS ability by allowing users to bring up a map, point at a street, and see the detailed engineering drawing, scanned document or video images of the street, ARC/INFO (a commercial GIS) stores the references of these files (CADD drawing, scanned document, video images) in its DBMS.

Tools Available

The tools chosen must be compatible with industry standard and well integrated with the BRANZ computing environment.

The chosen tools are C, DECWindows/compound document architecture (CDA), and Rdb. Figure 1 illustrates the relations of the chosen tools. The application will be written in C language in conjunction with CDA routines. Rdb will be used for storing the prototype's database.

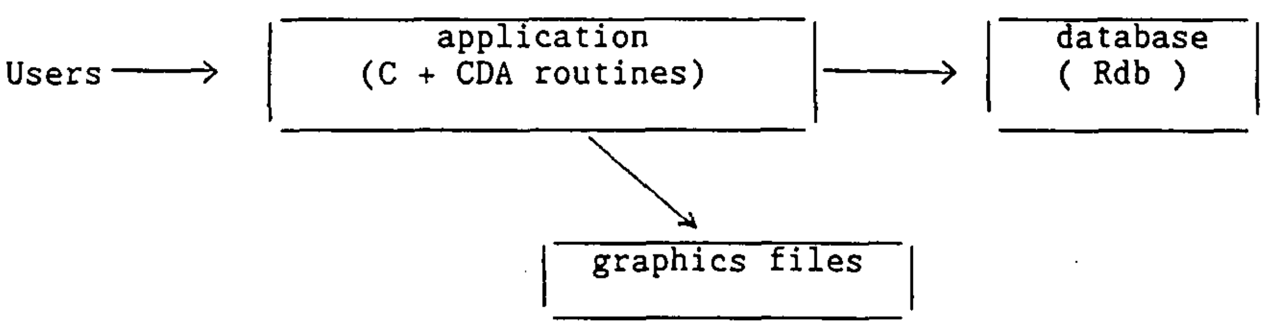

Figure 1: The use of chosen tool in the prototype

DECwindows/Compound Document Architecture (CDA)

Compound Document Architecture (CDA) is the architecture which provides the framework for document processing and interchange between heterogeneous systems. CDA comes with DECWindows. It provides a set of tools and utilities for creating, storing, and interchanging compound documents. A Compound document is a collection of data that can be edited, formatted, and processed as a document. In fact, it is a document that can contain not only text, but other integrated components such as numerical data, text, business graphics, logo, and images.

CDA toolkit is a collection of routines that support the creation and manipulation of compound document data. The routines provide functions for file management, stream management, compound document data management, document conversion, and item access.

Digital Document Interchange Format (DDIF) is the format used for the creation, storage and interchange of compound document. It is application independent, system independent, and compatible with ISO standards. It is also supported by Digital Equipment Corporation as interchange file format.

C - Third Generation Language

BRANZ has considerable experience with C language. C was used in conjunction with relational database management language, to embed database query commands for implementing the search and retrieval systems for our databases.

Interfaces between C, and Rdb, and DECWindows are available. C is well integrated with the CDA toolkit routines. It provides low level operations, in particular, memory management, which is needed in the scanning and the manipulating of compound document data.

Rdb - Relational Database Management System

Rdb is a relational database management system for VAX computer. Rdb uses the relational model of data access and storage, and provides a flexible, interactive, multi-user environment.

Development Of Prototype

Bjork and co-workers (1988,1989) proposed a prototype of building product model by using relational database. Product models are conceptual structures specifying what kind of information is used to describe buildings and how such information is structured.

We chose 'door' as a sample of a building object. The development was divided into three parts. The first part was to understand the DOOR.DDIF file structure. The second part was to implement a database that contained information describing the door. The third part was to write an application that would allow users to view the DOOR.DDIF files, to browse the information stored in the database, and to change the contents of the DOOR.DDIF files without having to redraw.

Understanding The Digital Document Interchange Format (DDIF)

To be able to communicate with the DDIF file, there is a need to understand its structure. Every DDIF document has the same general structure. It has a root, a descriptor, a header and a content. Figure 2 below illustrates the typical DDIF document.

This general structure makes it possible to communicate with the document.

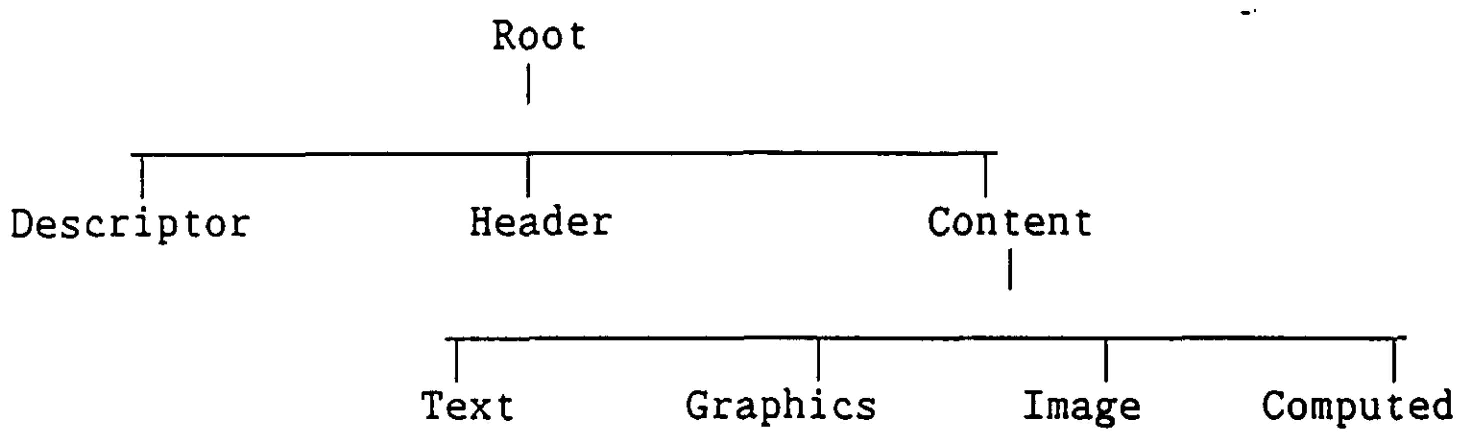

Figure 2: Typical DDIF document

In the DDIF document structure, it is the content part that differentiates one document from another, and it is the content that appears on the screen. Therefore it is the content that we want to store.

Consider a single-panelled door. A single-panelled door consists of framing or a rim which is grooved on the inside edges to receive a panel. If you analyse the door DDIF document, it gives you a lot of information, including the descriptor content and header content. But the most important information is the value of the height and the value of the width of the door and the coordinates of the panel. These values can be found in the document content. It is these values that make the difference between one drawing (of a door) and the other.

Implementing The Door Database

The database is required to support the following operations and to store the door attributes.

Operations:

* Browsing and lookup - The DOOR.DDIF can be displayed and the door data can be retrieved.
* Update - The graphical data and the corresponding standard text data can be retrieved and modified.

Representation:

* Graphical data - e.g: the coordinates of the door panels.
* Graphical file - e.g: DOOR.DDIF
* Door data - the standard text character. The information which describes a door.

A classification scheme has been used to represent objects in relational database. We classified a small set of the doors by how they are structured:

1. Panelled - consists of framing or a rim which is grooved on the inside edges to receive one or more panels.
2. Flush - This is the most popular type of door, particularly for internal use.
3. Ledged and battened - consists of vertical boards or battens which are secured to horizontal pieces called ledges.

Consider the inheritance hierarchy diagram of the doors in Figure 3. The DOORS is the superclass. The subclasses of the DOORS inherit all the attributes in DOORS.

Figure 4 describes the relations in DOOR database. Every relation is treated as a class. For example, DR1 in Figure 5 is a door which belongs to the DOORS class.

Door, DR1, has 7 attributes: IDENTIFIER, TYPE, HEIGHT, WIDTH, THICKNESS, MANUFACTURER, FILEHANDLE. The MANUFACTURER attribute holds the identity of a manufacturer, and the TYPE attribute is used to classify the door, DR1. The rest are self-identifying attributes.

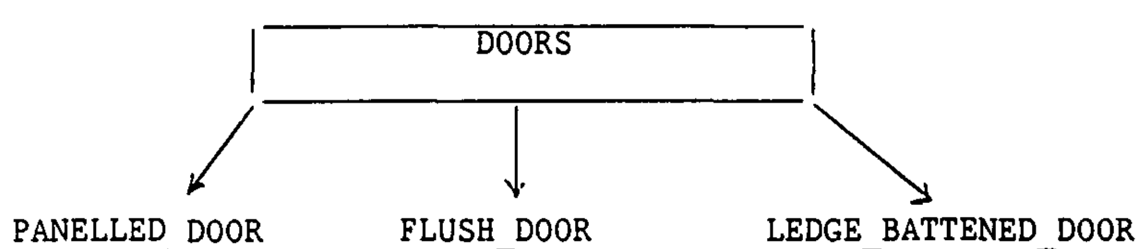

Figure 3: An inheritance hierarchy of the doors.

DOORS	
IDENTIFIER	STRING
FILE_HANDLE	STRING
TYPE	STRING
HEIGHT	INTEGER
WIDTH	INTEGER
THICKNESS	INTEGER
MANUFACTURER	STRING
PANELLED_DOOR	
IDENTIFIER	STRING
NUM_OF_PANEL	INTEGER
PANEL_EDGES	STRING
FRAME_MATERIAL	STRING
VISUAL_TREATMENT	STRING
RAIL_MATERIAL	STRING
FLUSH_DOOR	
IDENTIFIER	STRING
FLUSH_TYPE	STRING
LEDGE_BATTENED_DOOR	
IDENTIFIER	STRING
JOINT	STRING
LATCH	STRING
T_HINGES_WIDTH	INTEGER
LEDGE_THICK	INTEGER
MID_BOTTOM_LEDGE_THICK	INTEGER
BATTEN_THICK	INTEGER
MANUFACTURER	
NAME	STRING
ADDRESS	STRING

Figure 4: The Skeleton of the DOOR Database

DOORS	
IDENTIFIER	DR1
FILE_HANDLE	DOOR.DDIF
TYPE	PANELLED_DOOR
HEIGHT	2400
WIDTH	1500
THICKNESS	30
MANUFACTURER	ABC Ltd
PANELLED_DOOR	
IDENTIFIER	DR1
NUM_OF_PANEL	6
PANEL_EDGES	0,500,1000,1500 etc
FRAME_MATERIAL	SOLID
VISUAL_TREATMENT	SQUARE
RAIL_MATERIAL	SOLID
MANUFACTURER	
NAME	ABC Ltd
ADDRESS	P.O. BOX 777, WELLINGTON

FIGURE 5: The run-time of the DOOR database

Application

The door application allows users to search for a particular door in the database, and display the information as well as the drawing of the door. It also enables users to modify door attributes in the drawing files.

The door program was divided into three modules, Searching, Viewing and Creating. It was designed and implemented by using structure programming techniques. In the program, Rdb commands were embedded in C to attach to the database, start the transaction, detach from the database, retrieve records from the database, and use CDA routines to create new graphics files.

The graphics file's name is an attribute of the door database. It provides a link between the door and the door's drawing file. Graphics file's name is stored in the FILEHANDLE field of the door relation. Figure 6 illustrates the link between door DR1 and its drawing file.

```
DOOR
    IDENTIFIER:  DR1                    DOOR.DDIF
    FILE_HANDLE: DOOR.DDIF  ------>
       .
       .
       .
```

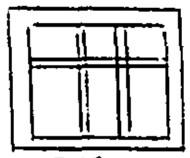

Figure 6: the link between door, DR1, and its drawing file

Searching the database is achieved by converting the user's search requests into internal database query. Door data which matches the query are retrieved, formatted and displayed on the screen. Viewing a drawing file is achieved by retrieving the file name of the drawing file, invoking the CDA Viewer, and passing the file name to the Viewer.

We only concentrated on the interfacing between the DDIF files and relational database, and not on the development of the user interface. Accordingly we did not develop a multi-windows application that would allow users to view the door data on one window and the DOOR.DDIF file on another window.

After the Creating module has received the commitment of changes of a drawing file's attributes, it retrieves the drawing file name and its new attributes from the database. Because the drawing file is a compound document, the module has to redraw the file. This is done by passing the file name, the descriptor, the header and the new content (new attributes) of the drawing file to the CDA routines.

Discussion And Future Work

We have developed techniques for interfacing Rdb database and graphics files, and the representation of objects using relational database. Techniques used in this project would be applied to the BRANZ database systems.

The prototype meets the initial aim. Beyond the development of the interface between database and graphics files, the work also provides experience in developing database based on graphical information and has given an opportunity to evaluate the tools used.

Once we understood the structure of the DDIF document, the writing of the code for the application appeared straightforward. Techniques used in communicating with the DDIF document, could be applied to other compound documents, eg: PostScript.

The C programming language in conjunction with Rdb commands offered a flexible environment for writing the search and retrieval program. CDA satisfies the requirements by providing tools and utilities which make the application development easier.

We plan to continue our investigation into the development of consistent user interface using DECWindows, the development of intelligent database, and the development of an integrated database environment.

CONCLUSION

Computerised information systems have the potential to play an important role in the management and transfer of information. The BRANZ programme has used university research expertise, knowledge from building technologists and BRANZ communication channels to the building industry, to ensure that the industry benefits from new information technologies.

A combination of the right subject domains and the right environment will dictate the success of the database applications.

Menu-based systems are more acceptable by casual users of database systems for searching. Professional searchers prefer command-based systems.

An ability to store structured documents using relational database technology creates more efficient techniques for electronic publishing.

The interfacing between database and graphics will allow easier access of data stored in the databases. The knowledge gained will also provide understanding in the development of the interface between relational database and graphics files.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contribution made by their colleagues at BRANZ , in particular David Warring, Roger Fowkes, David Sharp, Denis Bastings, Chris Price, John Duncan and Barry Holmes, and at University of Auckland, in particular John Hosking, Rick Mugridge and John Hamer.

BIBLIOGRAPHIES

Bjork, B., 1989.
Basic structure of a proposed building product model
Computer-aided Design, Volume 21 number 2-March-1989

Bjork, B., and Penttila, H., 1989.
A scenerio for the Development and Implementation of
a Building Product Model Standard
Paper presented at the symposium on current research and development
in Integrated Design, Construction and Facility Management
Standard University, California, 28-29, March, 1989

Bjork, B., 1988.
RATAS - a Proposed Finnish Building Product Model
Helsinki University of Technology, Otaniemi, Finland

Clausen, J., 1989.
User-Interface and Control Issues in Expert Systems
Auckland University

Date, C.J., 1985.
Relational Database selected writings
Addison-Wesley Publishing Company, Inc

ESRI, 1989.
Intergration of Geographics Information Technologies
ARC News Winter 1989 Issue Vol.11, No.1

Enkovaara, E., Salmi, M., Sarja, A., 1988.
RATAS Project - Computer Aided Design For Construction
Building Book Ltd, Helsinki

Fowkes, A.H.R.,Dechapunya A.H. and Smith, A.G., 1986
The Development of the BRANZ Computerised Information Service.
CIB Conference

Garvey M.A. and Jackson M.S., 1989.
Introduction To Object-Oriented Database
Information and Software Technology Vol 31, No 10 Dec 1989

Grant, J., 1987.
Logical Introduction to Databases
Harcourt Brace Jovanovich, Publishers

Hosking J.G, Mugridge W.B., Hamer J. and Dechapunya A.H., 1989.
A Programme for Research into Knowledge-based Systems for the Building Industry
Procs. AI'89, Melbourne Australia, 29-41
BRANZ Reprint No. 96

Robson D., 1981.
Object-Oriented Software Systems
Byte August 1981

Smith K.E., Stanley B.Z., 1987.
Intermedia: A Case Study Of The Differences Between Relational
and Object-Oriented Database Systems
OOPSLA 87 Proceeding

Stein J., 1988.
Object-Oriented Programming and Databases
Dr. Dobb's Journal, March 1988

Van Der Roest, M., 1989.
Databases Management Systems Integrate with AutoCAD Drawings
CADENCE November, September 1989

VMS Compound Document Architecture Manual Dec 1988
Digital Equipment Corporation, Maynard, Massachusetts

Whitney, R.S. and Dechapunya, A.H., 1985
Building of a System
Interface: Journal of the NZ Computer Society Inc.
September Issue

BUILDING RESEARCH ASSOCIATION OF NEW ZEALAND

To promote better building through the application of acquired knowledge, technology and expertise.

HEAD OFFICE AND RESEARCH STATION

Moonshine Road, Judgeford
Postal Address - Private Bag, Porirua
Telephone - (04) 357-600, FAX 356-070

REGIONAL ADVISORY OFFICES

AUCKLAND
Telephone - (09) 5247-018
FAX - (09) 5247-069
290 Great South Road
PO Box 17 -214
Greenlane

WELLINGTON
Telephone - (04) 357-600
FAX 356-070
Moonshine Road, Judgeford

CHRISTCHURCH
Telephone - (03) 663-435
FAX - (03) 668-552
GRE Building
79-83 Hereford Street
PO Box 496

DUNEDIN
Telephone - (024) 740-454
FAX - (024) 778-436
6 Hanover Street
PO Box 1635

BRANZ

THE RESOURCE CENTRE FOR BUILDING EXCELLENCE