# SOFTWARE

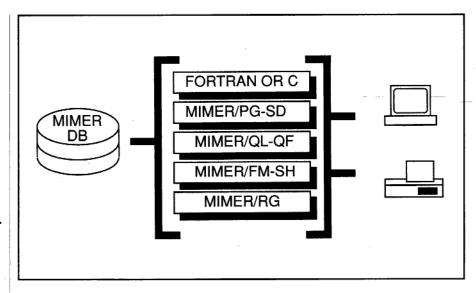
# System development tools

**■ MIMER VERSION 4** 

MIMER is an advanced database system featuring efficient facilities for application development, data presentation and text management.

MIMER/DB database handler MIMER/QL query language MIMER/PG program generator MIMER/FM screen form handler MIMER/QF query-by-form MIMER/RG report generator MIMER/SD system development MIMER/SH screen handler

By using MIMER/DB as a development aid system, solutions can be produced much faster than with conventional technology with a result that is user-friendly, flexible and easy to maintain.



# MIMER/DB

The MIMER/DB database handler is based on the relational data model. The system's function is, among other things, to transfer data between the database and other part systems. The database structure can be defined and modified without changing existing application programs. The system allows independence of computer type and programming language and the system is simple to learn and easy to use. Furthermore, MIMER/DB allows more than one program to use the database at the same time.

#### Relational database

In a relational database, all data are stored in tables.

Other forms of data structures rely on hierarchies and/or networks of pointers between records. There are no such pointers or hierarchies in a relational database, thereby making it much faster than other models.

Each table in a relational database is a collection of records, where each record has the same group of data items. The data items form the columns of the table and each row is a distinct record. To uniquely identify each record one field is chosen as the primary key (but, where necessary, several fields can be combined to form a unique key).

Any particular data item in any column can always be accessed by searching the whole table. However, this can be slow and to avoid this an additional index – a secondary index – can be created for any column.

In short, the relational approach to a database structure features the following major advantages over other database structures:

- Ease of use. The tables are simple to handle and understand.
- Flexibility. Tables can easily be added and extended. The data base software provides full independence between the tables and programs.
   Old programs do not need revision when tables are revised.
- High degree of relatability. Several choices are available to relate one table to another, with no need to predetermine likely combinations.
- Ease of design. Other database system require highly skilled staff to create the design and the designs themselves are subject to revision as the usage of the data develops. Relational databases do not need redesigning.

# **Objects**

In a database environment, there are certain kinds of 'base data' that combines into a description of the database normal data. These data are denoted by the term objects. MIMER features the following objects:

- idents,
- o databanks,
- o tables,
- o views,
- o synonyms,
- o domains,
- o reports.



#### Ident

An ident in MIMER is:

- o a user = user ident,
- a protection facility for a program or parts of a program = program ident
- o a group = group ident

#### **Databanks**

MIMER/DB works with storage units called databanks and a database may contain an arbitrary number of databanks. As far as D-NIX is concerned, a databank corresponds to an ordinary direct access file consisting of commonly sized pages addressable by a relative page number.

#### **Tables**

A databank contains an arbitrary number of tablest The content of a table is stored in one databank, whereas one databank may contain several tables.

# View

A view is a specified part of a table. A view can comprise a few columns, certain rows, or in the most extreme case, just a single field. A view can be created on the basis of another view, which in turn has been created as part of another view. The views are not created as tables but always go back to an underlying table.

## **Synonyms**

A synonym is an alternative name for a table or view.

#### **Domains**

Domains or term descriptions are primarily concerned with the control of data. The domain definition includes many different ways to specify values, ranges and masks for control of data.

### Reports

A report is a description of a report form definition in MIMER's report generator.

## **Data dictionary**

The active data dictionary in MIMER is updated through menu-driven application programs. All definitions for databanks, tables, views, index, reports, synonyms and domains are stored in the data dictionary.

# **Database security**

Any user, group of user or programs that will communicate with MIMER/DB must first provide a user identification and a pass word. Authorized user id's and passwords for accessing MIMER/DB are stored in the system table. Initial entries in the system table are made when creating the database.

# Access to databanks and tables

In MIMER, there are essentially unlimited ways to build up a security system where each identity has defined access rights and privileges. The access rights give the identy the right to read, update and delete data and rows in the table.

# MIMER/QL has over 30 commands, the most common ones being:

COPY TO/FROM, DEFINE INDEX, DEFINE TABLE, DELETE, DESCRIBE, EDIT, EXEC, EXIT, GET, HELP, INSERT, PRINT, REDEFINE, REMOVE, UPDATE.

The privileges given are divided into two groups:

- System privileges which give an identity the right to create data banks, reports or new identities.
- Object privileges which give an identity access to a program, an identity or membership of a group, and the right to create tables in a specified databank.

## Transaction management

In order to preserve the integrity during concurrent access to a shared database, operations are grouped in units called transactions.

When using the MIMER/DB transaction facilities, a databank has to be defined for that purpose. If a logging facility is required, a databank has to be defined for that purpose as well and if that bank exists, MIMER/DB will automatically log every committed update transaction. During a transaction, read operations are logged on a local 'read-set' and update operations are put on a local 'write-set', thus leaving the question of integrity and consistency until 'end-of-transaction' is executed.

# **Database recovery**

MIMER/DB provides an automatic restart facility, that will redo all unfinished committed transactions, thus keeping the database consistent. The careful commitment used in MIMER/DB has eliminated the need for any recovery procedure using checkpoints and log files in order to keep consistency.

When the databank transaction logging exists it contains a log of all changes completed by each transaction. For each change a log record of the updated object is written, thus making it possible to redo all transactions during disk crash recovery.

MIMER/DB provides a method of recovering from a system failure that has caused a loss of disk storage integrity. This method entails periodically generating a 'backup' copy of the data. When starting MIMER/DB after a

When starting MIMER/DB after a media failure, the data is restored from the backup and the changes recorded on the log-file, if any, are applied.

# **MIMER** programming interface

PI is a programming interface that makes it possible to use all functions in MIMER/DB from programs written in traditional programming languages, provided the language can make external calls to the Fortran or C languages.

## MIMER/QL

MIMER/QL is an interactive query langauge, which is used to define and build databases as well as to search and manipulate data.

MIMER/QL organizes data in databanks. A databank may contain one or more tables where each table consists of rows and columns. There must be a way of uniquely identifying each row in a table and such an identifier is called a primary key. A primary key can consist of one or more columns.

A table can be compared with a card index system where each row corresponds to a card and each colum to the structured information that is on each card. By making connections between different tables and by specifying conditions for the data that are in the column, it is possible to find the desired information.

MIMER/QL is used, among other things, to:

- o define new databanks
- o define new tables
- o change table definitions
- o load tables with data
- o retrieve desired data
- o change or remove data
- o insert data
- predefine a number of MIMER/QL commands (procedures) for later regular use.

In a procedure, in addition to the usual MIMER/QL commands, there can be special functions, such as substitutions, assignments, conditional jumps, looping and menu handling.

# MIMER/PG

MIMER/PG is a system allowing the user to develop and generate application programs in FORTRAN or COBOL without conventional programming.

MIMER/PG is an application generator, a 'fourth generation language', that makes it possible to develop and test applications in a very short time. This can often be done by the end users themselves.

In fact, MIMER/PG is an interactive system including modules for:

- database operations (an expanded MIMER/QL)
- specification of application systems
- o library functions (storing, retrieving

# MIMER/QF features the following commands:

NEXT, CLEAR, GET, UPDATE, DELETE, INSERT, UTILITY, EXIT

- and modifying defined applications)
- program generation (transfers specified applications to FORTRAN or COBOL code)
- installation of new functions in MIMER/PG

While specifying an application, a model is defined which can be tested directly. The application can easily be redefined if the expected or desired result is not obtained. Once the user is satisfied with the application definition, the program generator functions can be executed to obtain COBOL or FORTRAN programs which more efficiently produce the same result.

Generated applications can be transferred to another computer system, since MIMER/PG is capable of generating code suited to the compiler on a specific computer. This means that program development can take place on one computer while the production system may be run on completely different computers.

# MIMER/FM

MIMER/FM (Forms manager) is a tool for creating, editing and using screen forms for MIMER database applications in terminal environments. Forms are defined and edited using a full-screen editor. Form definitions are stored in a user-defined library in MIMER databanks, and may be retrieved for modification with the editor or for use in application programs. Facilities are provided in the context of the editor for testing the appearance and behaviour of forms.

Forms are used in application programs written in conventional programming languages through calls to the FM Interface. In addition, the use of FM from MIMER/QL and MIMER/PG is supported.

Many of the features provided by MIMER/SH Screen Handler are included in FM, in a form suited to use with synchronous terminal protocols also. Forms defined with FM may be converted to input files for the MIMER/SH Compiler in systems where SH is available. Facilities for converting SH screen pictures to a FM form definition are provided.

### MIMER/QF

MIMER/QF is a tool for manipulating the contents of a MIMER database. It is designed for entry, updating and removal of data, and allows simple data retrieval operations. One prominent

feature of MIMER/QF is that it is very easy to use – everything is done via a screen formula that the user just fills out with appropriate data.

MIMER/QF works with one table at a time. The user is first asked for the name of the table to use. By typing a question mark instead of the table name, the system will list all tables to which the user has access. Once the table is specified a screen picture showing the names of all columns in the table are shown. Each column name is followed by a number of characters appropriate to the width of the column as defined in the table. The user can change the appearance of the standard screen formula by using the MIMER/SH screen handler. Moving between the fields is performed by using the terminal arrow keys. A command line displayed at the bottom of the screen lists all commands available.

By using one or more of QF's eight commands, the user is able to manipulate the data in the active table.

#### MIMER/RG

MIMER/RG report generator provides a simple tool for defining a neat output format for the data in a MIMER table.

MIMER/RG consists of the following programs:

- RGFORM supporting screenoriented definitions of reports,
- RGCOMP compiling report definitions for execution
- RGEXEC executing a previously prepared report definition and generating a completed report in the form of an output table
- RGVIEW displaying a report on the user's terminal
- RGPRT printing a report on a printer or a sequential file

All RG programs are available as routines that can be called up from the user's own program.

Reports can be defined in two ways – either by using the screen-oriented definition program RGFORM, or directly in the report generation language RGL, which is written in string in a calling program or in a sequential file. After compiling, reports can then be executed using RGEXEC, either from the query language MIMER/QL or from the user's own programs. The formatted output can be reviewed on the screen with RGVIEW and printed with RGPRT.

# MIMER/SD

MIMER/SD (System Development) turns MIMER into a complete 4th generation programming tool.

Typical characteristics of MIMER/SD are:

- Create basic applications directly from a data model
- Develop prototypes in short time for end user review
- Generate source code for a working

application

 Finalize application by including high level language routines through MIMER/PG

MIMER/SD comprises a number of routines that will help in handling an advanced data dictionary for development of the database as well as applications operating against the data base.

The database component of the system is specified through a data model while applications are specified through descriptions of links, user commands, etc. MIMER/SD is also used to define user access authorities and terminal environments. Thoughout this process, MIMER/SD aids in structurizing the specifications. In fact, MIMER/SD suggests standard routines which either can be accepted, modified or discarded.

## MIMER/SH

MIMER/SH is a screen handling system that is independent of computer and terminal type. It is employed to define screens and use them in application programs.

The screen consists of a number of fields and descriptive text. The fields are used for the input and output of information. The user can move between different fields using function keys.

MIMER/SH can perform checks on the values that have been entered, for example:

# Field checks

A field

- must consist of numbers or letters,
- o cannot be skipped if not filled out,
- o date check.

# Validation

Checks of entered values against predefined values and/or interval.

#### User defined checks

- personal registration number checks,
- checks against matching or nonmatching occurence in the user's files.
- checks of the user's own personal registration number.

# Group checks

Checking of a group of fields that are linked together.

MIMER/SH makes it possible to modify the screen's appearance, controls and dialogue texts without any modification to the application program. Thus, expensive modification and testing are avoided and the application program remains unchanged.

MIMER/SH together with MIMER/DB makes the application program independent of computer and terminal type. The application programs can easily be moved from one computer configuration to another and, from the user's point of view, the program will behave in the same way on this machine. In addition, high conversion costs are eliminated.