

IMANDC 1 (2) 53-94 (1978)

ISSN 0378-7206

# INFORMATION & MANAGEMENT

THE INTERNATIONAL JOURNAL OF MANAGEMENT PROCESSES AND SYSTEMS  
JOURNAL OF IFIP USERS GROUP (IAG)

Volume 1, Number 2  
February 1978

INFORMATION & MANAGEMENT VOL. 1, No. 2 (1978) P. 53-94

NORTH-HOLLAND PUBLISHING COMPANY

# A Conceptual Model of an Interfunctional Data Base System

Zygmunt Ryznar,  
30-147 ul. Na Blonie 9a/207, Cracow, Poland

The structuring of data resources is discussed with reference to both general and specialized data bases. A modular technique for developing computerized information systems is proposed as a way of meeting the changing needs of a business. Particular attention is given to the coding of business events, which is one of the key questions in data-driven processing. Most of the features of the described model are applicable to manufacturing companies, though some are more general.

*Keywords:* Data base, data structure, problem module, data module, task module, data-driven processing.



Dr. Zygmunt Ryznar studied systems analysis and design at the Institute of Economics and Statistics in Moscow (M.S. degree in 1962) and received a Ph.D. in 1974 from the Academy of Economics in Cracow. He is currently working for the Union of Aircraft and Engine Industry in Poland as a data base project leader. Previously he was involved on projects as senior and chief analyst in several computer centers. Dr. Ryznar is especially interested in developing modular data base systems for manufacturing companies.

## 1. A methodology for the design of data base systems

There is no comprehensive and widely accepted theory supporting the process of developing data base systems. Such a theory is needed because of the variety of existing data base packages and their potential impact on decision making, organizational structure, information flow and processing techniques. Since (in terms of data and procedures) a business is changeable, a data base system is required to have the flexibility, that allows both programs and data to change in accordance with object system situations. A data base system, like an operating system, should be a collection of many modules which can be united in a variety of combinations, according to the needs of the user, the complexity of the business and the constraints applied by the computer (mainly the storage and software facilities). To emphasize some methodological problems it may be useful to pose some questions that have occurred to analysts concerned in the design and use of data base oriented information systems:

- (i) What are the methods of breaking the system into subsystems and modules?
- (ii) What are the properties of the component necessary for assembling it with other modules or subsystems?
- (iii) How is the system expanded with the least possible cost and effort; i.e., how do we find the principles which provide the system both with stability and adaptability?
- (iv) What theory is available for choosing data structures?
- (v) How do we design a multi-database system (consisting of both central and local data bases), and how do we solve the problem of communication between them?

(vi) How do we code business events within one common input file to free the system from the necessity of processing many transaction files?

The above list of questions can easily be expanded. Some will be discussed later.

A method of system structuring, proposed in this paper, on one hand corresponds to the systems approach, while on the other it differs considerably from it. As is well known from the systems theory point of view, a system is a combination of interrelated parts forming the whole, i.e., having the same (common) objectives. The general scope of the business can be roughly obtained by a cross-table analysis (see fig. 1), that helps the project team translate user needs into a technically feasible system design, visualize company-wide functions, establish relationships between functions and resources, and finally provide the logical data macrostructure (in terms of data modules and data bases).

The development of a computerized information system is carried out by the successive design and implementation of particular components that must be relevant to the management requirements and the decision making process. Having only a broad picture of the system, the designer is often unable to provide all necessary relationships between modules, data, etc. This deviation from systems theory results mainly from the fact that the computerized system, in contrast to the object material system, is never to be completed and its components are continuously modified. There are some reasons for this situation:

- (a) Limited capability of the analyst and restrictions on time.
- (b) Absence of satisfactory models of the business.
- (c) Changeability of the business.
- (d) Deficiencies in existing methods of designing an information system.

There is general feeling that considerable improve-

| FUNCTIONS and PROCESSES              | 1                      | 2           | 3         | 4     | 5             | 6         | 7         | 8     | 9             | 10                | 11      | 12                   | ... | 20                                 |
|--------------------------------------|------------------------|-------------|-----------|-------|---------------|-----------|-----------|-------|---------------|-------------------|---------|----------------------|-----|------------------------------------|
|                                      | Codes and descriptions | Forecasting | Purchases | Sales | Manufacturing | Inventory | Purchases | Sales | Manufacturing | Inventory control | Costing | Financial accounting | ... | Activity of the company as a whole |
| 1 Financial assets                   | X                      |             |           |       |               |           |           | X     |               | X                 |         | X                    |     |                                    |
| 2 Raw materials                      |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |
| 3 Components and finished goods      | X                      | X           |           | X     |               |           |           | X     |               | X                 |         | X                    |     |                                    |
| 4 .....                              |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |
| 5 Work in progress                   |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |
| 6 Power and water                    |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |
| 7 Equipment                          |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |
| 8 Personnel and labour               |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |
| 9 Intangible assets                  |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |
| 10 Factory engineering documentation |                        |             |           |       |               |           |           |       |               |                   |         |                      |     |                                    |

Logically segmented data-base record

functional vector

coffer

main logical segment

associated logical segment

Fig. 1. An example of the fact cross-table for the manufacturing company. The problem module: selling finished goods and components.

ments could be achieved by structured design methodology and particularly through the use of data base systems. Broadly speaking, three approaches may be distinguished in a data base system design: external, intermediate, and internal. An *external approach* refers to the manner in which the user sees the information system. It answers the question, "What are the user requirements?" An *intermediate approach* refers to the analyst point of view and deals with data structures definition and procedure specification. An *internal approach* refers to the data processing technology, including DBMS facilities. It is required to answer the question, "How is it to be done?"

These approaches are related to one another and all represent a structural point of view. An external approach leads to *structural* elements called problem modules, an intermediate approach results in data modules, and an internal approach gives the requirements of task modules. A task module is a block of coding for performing a specific operation, e.g., data validation, etc. (see fig. 2). The data module is a collection of the logical data segments required to solve a given problem. A system is set up successively by implementing problem-oriented packages, which are the combination of problem modules, data modules and task modules. The problem module is responsible for providing additional data description and user procedures and should be formalized in a system specification language like PSL (ref. [1]).

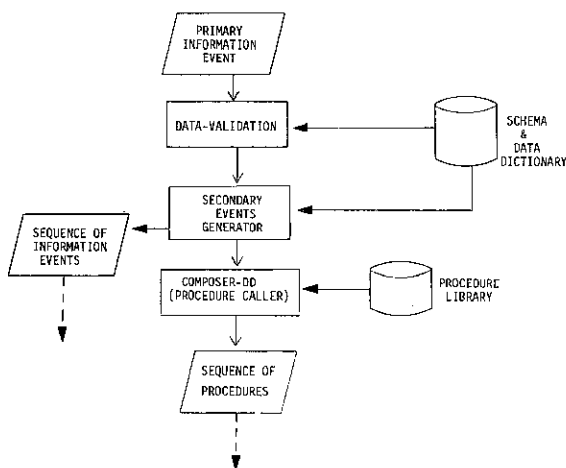


Fig. 2a. Interrelationship of FACT and data driven processing. The front-end.

## 2. Functional analysis for establishing data modules

A business information system is closely related to the processes and management functions carried out on materials, personnel and financial resources. All these elements may be considered as essential factors in every business activity, and in order to get an overall view of the relationships between resources and functions (processes) a special cross-table is proposed (see fig. 1). The square in this table is the intersection of a resource line and a function column; it will be called a "coffer". A *function* implies a management function: forecasting, planning, costing, etc., while *process* marks out an activity involving materials; e.g., manufacturing, selling, purchasing, etc. Generally, the layout of the cross-table is not fixed and should be established according to company needs. Particularly, the ordering of functions in the table affects the "clustering" facility, that consists in choosing some functions (such as short-term planning) as the primary or superior, and others as the secondary. This cluster corresponds to a *subsystem* as used in general classification, but differs from it by being more sensitive to the business environment, and in giving more detailed specification of elements and relationships.

A coffer may be considered as a logical data region resulting from the relationship between a given resource and a chosen function. If the region makes

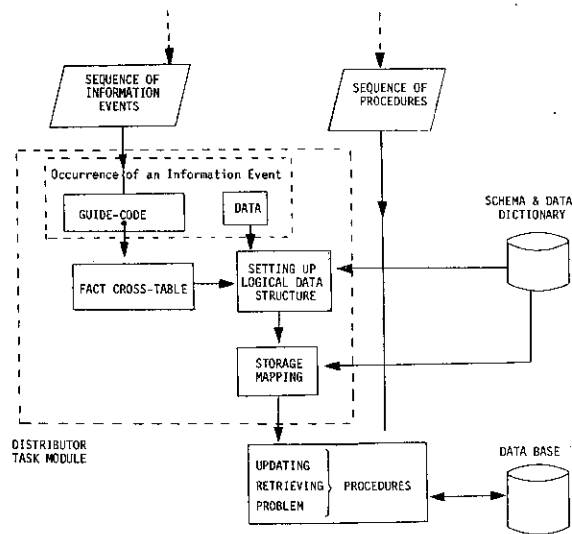


Fig. 2b. Interrelationship of FACT and data driven processing. The back-end.

sense and thus contains data, it is the subject of the description in a data definition (or schema) as a logical segment. A collection of all logical segments related to a given problem is a *data module*. One method of depicting logical segments is demonstrated in fig. 1. The complexity of the system can be easily estimated by overlaying several problem-oriented cross-tables and the most used coffers will be found.

It is necessary, however, to distinguish between data base organization and data base usage. Firstly, the data-base organized under cross-table rules is considered as a collection of logically segmented records. Secondly, the same base may be processed as a collection of virtual (logical) vectors. A *functional vector* is a set of one type segments related to a given function. Thus, in a resource-oriented data-base organization, we can derive many functional sub-data-bases.

The cross-table technique helps in breaking information resources of a company into data-bases. However, due to data compaction methods, access methods, and other internal aspects of the system, the storage segments need not be implemented to coincide with the logical segments.

The method of structuring as discussed here, facilitates an interfunctional usage of information within the company. It helps also in establishing data base boundaries, thereby providing file stability. The FACT (Functional Analysis Cross Table) method (ref. [2]) may be useful as an aid in the early-stages of systems analysis, particularly in supporting top-down decomposition and in selection of a pilot function for initial implementation.

### 3. Information events and data-driven processing

There are many DBMS packages for handling data located in data bases. However, this activity is only the final step of data base implementation. In most cases a successful loading and updating of the data base requires careful attention to data capture. Input transaction entered in random sequence and at random times have to be recognized and processed. One essential principle of *data-driven processing* is that procedures are initiated because of the data entries, and thus one input stream is used instead of involving many transaction files. A data entry, termed

here an *information event*, consists (at least) of primary keys and attributes. Primary keys are necessary to recognize data and distribute them. Attributes (secondary keys) are used for establishing data relationships by indexing and inversion techniques. Some primary keys are grouped into a "guide-code", that governs the business.

The components of a guide-code are: activity, resources, operation, and event category. An example list of primary keys for a manufacturing company is shown in table 1. This proposed way of the coding business events is closely related to the cross-table of fig. 2. Each information event has to be recognized by the DISTRIBUTOR task module, which establishes logical segments and maps the storage, by having access to the guide-code and complementary keys (e.g. part number) located in the DATA portion of a record. These operations are supported by the data schema containing descriptions of storage structure and data relationships.

An input stream consists of information events that correspond directly to the actions performed in the business (e.g., customer order receipt, finished product expedition, material receipt). These events are referred to as primary events. There is no pre-defined limit to the number of different types of events that can be accommodated in one input stream, but they should be described in the guide-code. The number of events can be internally expanded by the GENERATOR task module, if several procedures are required to process some primary events. In this case, a given primary event triggers the process of generation and a sequence of secondary events is established. Each of them indicates what problem procedure or multidirectional operation should be invoked. Assembling problem procedures and task procedures into program units is the activity of the COMPOSER-DD module. An example of primary events is the customer order receipt, which causes the generation of secondary events necessary for calling the following procedures: product specification (breakdown), gross requirement, netting, man-hour estimation, material consuming estimation, manufacturing schedule, purchasing, and pricing and quotation procedures.

Table 1 An example guide code: Primary keys for a typical manufacturing company.

**I ACTIVITY**

- 0 out of classification
- 1 manufacturing
- 2 purchasing
- 3 selling
- 4 marketing
- 5 maintenance
- 6 research and development
- 7 production engineering
- 8 administration activity
- 9 activity of the company as a whole

**II RESOURCES**

- 1X financial assets
  - 11 share capital
  - 12 operating profit
  - ...
- 2X raw materials
- 3X components and finished goods
  - ...
- 5X work in progress
- 6X power and water
- 7X equipment
- 8X personnel and labour
- 9X intangible assets
  - 91 patents
  - 92 copyrights
- 10X factory engineering documentation

**III OPERATION**

- (a) direction of operation
  - 0 insertion
  - 1 increase
  - 2 decrease
  - 3 multidirectional operation
  - 4 stock entry
  - 5 replacing
  - 6 deleting
  - 7 retrieval
  - 8 problem procedure called by data-driven rules
  - 9 other procedure
- (b) operation code
  - 1X receipts of resources
    - 11 purchase
    - 12 receipt for deposit
    - 13 receipt for regeneration
    - 14 receipt from production departments
    - ...
    - 19 gift received
  - 2X external outgoing of resources
    - 21 sale
    - 22 issue for deposit
    - ...
    - 29 gift issued

**3X issues of resources forming the cost of production**

- 31 direct (prime) cost
- 32 factory overheads
- 33 administrative overheads
- 34 sale-distribution expenses
- 35 purchase expenses
- 4X expenses
  - 41 accrued expenses
  - 42 prepaid expenses
  - 43 expenses not costed
- 5X income
  - 51 accrued revenue
  - 52 prepaid revenue
  - 53 income not included as profit
- 6X } some documentary events
- 7X }
  - 61 customer order receipt
  - 62 purchase order issue
  - 63 job order issue
  - 64 invoice receipt
  - ...
- 8X } problem procedures
- 9X }
  - 81 pricing and quotation
  - 82 product breakdown
  - 83 manufacturing scheduling
  - ...

**IV EVENT CATEGORY**

- 0 unrecognized event (outside data-driven processing)
- 1 bookkeeping events (materials movements and finance)
- 2 forecasting and long-term planning
- 3 short-term planning
- 4 government directives
- 5 top management disposition
- 6 operating management disposition
- 9 event in clarification (temporarily unclassified).

**4. High-level data structuring**

The level of data structuring discussed here deals with data bases and sub-data-bases. The starting point is a cross-table functional analysis, that leads to resource-oriented data bases. Each resource line is usually assigned as the sub-database (i.e., the area within the database having a homogenous data structure), but it may be set up as a data base, depending on the data volume, storage capability and complexity of elementary data structures defined in low-level data structuring.

The idea of one common data-base serving all

applications within company is very attractive, because it provides good conditions for minimizing data redundancy and simplifying data access, however, it is almost impossible to implement this concept in large companies. Firstly, the data volume in the total system exceeds the storage capability of most medium-size computers. Secondly, some specialized packages like BOMP (Bill Of Material Processor) have often already been implemented. They use their own data files and are efficient for a particular application. Secondly, a business often has several data groupings which requires different data structures and procedures. Some such examples are

- (1) material resources registration (e.g., stock records),
- (2) employee (complex records and specific retrieving based on many attributes; this sometimes requires an inverted file capability),
- (3) active objects (e.g., customers and suppliers records),
- (4) selected attributes (e.g., bad debts, departments, professions, stock location),
- (5) events (e.g., orders, sales, purchases, etc.),
- (6) calculations (e.g., merchandise turnover).

Regarding the above factors, the following data-bases are proposed and shown in fig. 3:

- (i) RSS (RESOURCES base) – contains information about material resources; links to BOMP (PNF-

Part Number File and WCF-Work Center File) and other bases.

- (ii) PRS (PERSONNEL base) – provides information about each employee and summarized data (PSS) ordered against departments, professions, etc.; equipped with fast retrieving facility based on specialized inverted files (SIF) connected with BOMP files by means of temporary extracts.
- (iii) VRT (VARIETY base) – contains information about active objects, events, financial resources, etc.
- (iv) BOMP (Bill of material files) – in proposed multibase environment these files contain only data necessary to chain files and obtain some reports; complementary data are stored in the RESOURCES base.
- (v) MOD (MODELS base) – includes mathematical (linear and dynamic) models of manufacturing processes; most input data are derived from other bases.
- (vi) CIF (Central Inverted File) – is set up to provide fast retrieval of special information like unacknowledged orders, inactive stocks, inactive orders, bad debts, etc.

These above mentioned bases may decrease their volumes by:

- (a) transferring inactive records to ARCHIVE (ARCH) files,

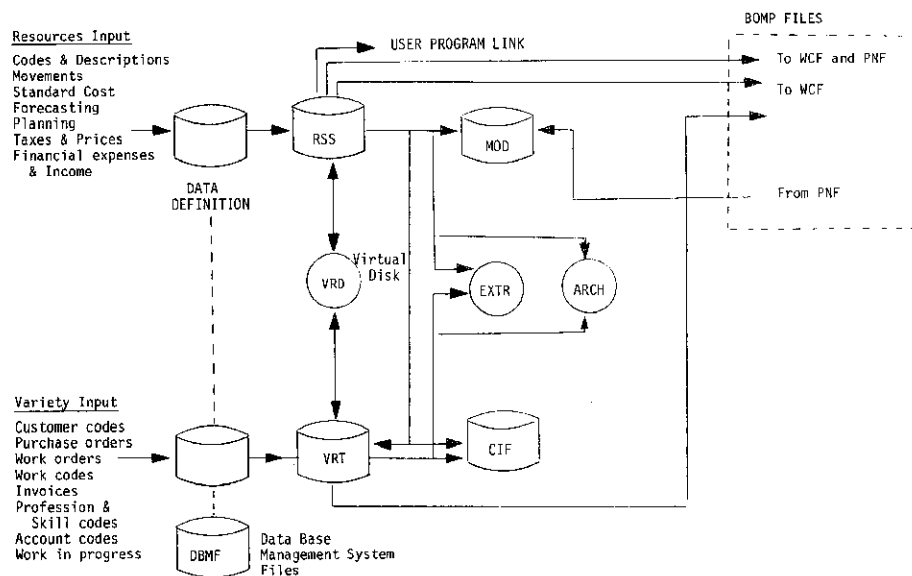


Fig. 3a. Data flows between various data bases. Common interfunctional data bases and files.

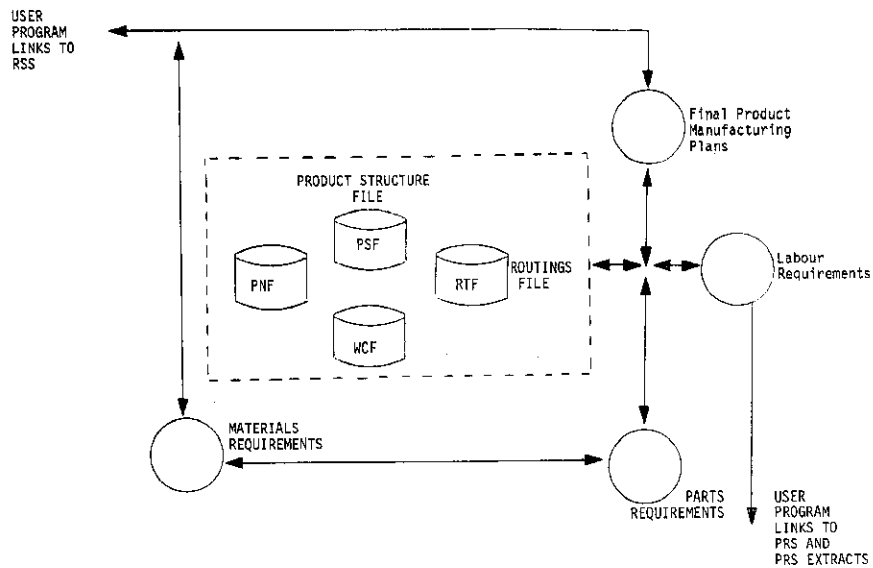


Fig. 3b. Data flows between various data bases. BOMP files.

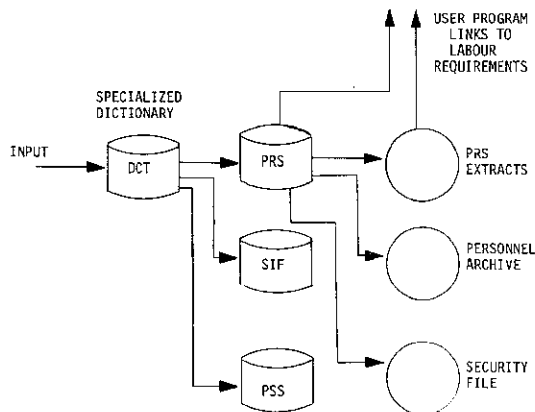


Fig. 3c. Data flows between various data bases. Specialized personnel data bases.

### 5. Low-level data structuring

We shall not review in this paper methods of data structuring. We shall mention only some complementary data structures, which are needed in the context of our data base model. The "macrolinks" established by crosstables technique require complex data structure. To choose the data structures one must consider the following factors:

- complexity of processes and objects acting in the business (e.g., a product structure expressed as a tree),
- specific needs of the user (for a very fast answer, inverted files may be imperative),
- demand to minimize data redundancy, computer time, mass storage, etc.

It is difficult to find a structure which meets all the above requirements, and simultaneous use of different data structures in multibase system seems to be the only solution. In order to achieve a good solution, special attention should be given to atomic and vector data structures. An *atomic structure* deals with elementary sets, which have no links between data fields. We call these sets *atoms*, an example is the data group: raw material number (as the nucleus), material description, and unit of measure (as the associated elements). The aim of atomizing is to decrease the

- (b) recording summary data (potential answers to enquiries),
- (c) sending less active records to "virtual" base on tapes,
- (d) extracting some data to an extractions (EXRT) file to allow processing of a given base together with another large base.

The use of several data bases causes some complications to the Data Bases Management System, since it should provide links between bases and it has to operate on many data structures types, defined in the low-level data structuring stage.



number of relationships and, thus, reduce computer time. In a *vector structure* data are stored in a transposed form, i.e., one vector is used for storing occurrences of each data field (instead of traditional records, which groups different fields). The vector structure comprises data vectors, link vectors, and key vectors. In data vectors, the key is virtual (i.e., it does not exist) and is indicated by a relative position of value in a data vector. A key vector is a collection of key values and, optionally, a relative position of each value (to facilitate a data access). A vector structure is suitable for making many data relationships by means of linkage vectors maintained by DBMS software. This structure is attribute oriented and therefore is applicable when the user, having the value of the attribute, wants to know the objects to which this value belongs. Low-level structuring is the final stage of top-down design of the data base oriented information system.

## 6. Conclusions

- (i) A modular problem-oriented technique for developing data base systems seems to be a way to meet changing requirements of business.
- (ii) There are reasons for designing both common and specialized data bases within one information system for large manufacturing companies.
- (iii) A proper technique of coding business events is a basis for data-driven processing.
- (iv) Simultaneous use of various data structures within one data base system is one way of expressing business complexity and improving the data processing function.
- (v) A system should be set up by first implementing problem-oriented packages, which are the combination of problem modules, data modules and task modules.

## References

- [1] D. Teichroew and E.A. Hershey III: "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems." IEEE Transactions on Software Engineering, Vol. SE-3 No. 1, (Jan. 1977) pp. 41-48.
- [2] Z. Ryznar, "Structure Description Language in the Structured Design", paper submitted for publication, available on request from author.