"I do not fear computers.
I fear lack of them".

(Isaac Asimov)

ON APPLICABLE MODELS FOR DATA MANAGEMENT

Antoni Wolski

Finnish Meteorological Institute

Abstract

A historical review of models used in construction of data base management systems is presented. Capacity of the 3-level ANSI/SPARC model is analysed from the point of view of the ECMWF archive data handling problem. An effort to select a combined, data management/network model, for the purpose of the ECMWF data handling system, is undertaken.

1. Introduction

Models appeared in computer system design as a methodological aid. Once a system model is accepted, it performs the role of a "check list" for good design. Models put special stress on interfaces, both internal and external, on promoting modularity and on future growth and compatibility between systems. The very fact that these issues are so crucial in computer networking, contributed to the success of the Open System Interconnection model which is used in data communications. In the field of data management, the process of system model development has been somewhat slower. However, one of the latest efforts, the ANSI/SPARC model, is already a very good candidate for an "Open Data Management" model.

2. Goals of Data Base Management

Most of the progress in data management has been made within the field of data bases. While a current definition of a data base as a "... collection of inter-related data stored together ... to serve one or more applications in an optimal fashion (Martin, 1975) sounds very general, the fact of life has been that most of the solutions are applicable to commercial data bases only, and most of the available general-purpose systems for managing data bases are also aimed at the commercial processing market. Nevertheless, there are many concepts and ideas that may very well apply to, for instance, scientific data bases of the type needed at ECMWF

to handle all kinds of archives. In the course of the following discussion, we shall try to assimilate possibly usable concepts and experience.

A primary, general goal of using data base management techniques is

TO EXTEND INFORMATION SYSTEM LIFE TIME.

Life time of information systems (meaning applications and underlying support systems) is obstructed by a variety of problems including inability to maintain the system, modify it according to the changing needs, or survive a major change of the environment (e.g. new computer hardware, new storage media, etc.). The objective of extended life time is achieved by stressing the following characteristics of the system:

TRANSPARENCY - to hide, at any level of the system, actual operations performed by underlying layers;

FLEXIBILITY - to adapt quickly to needs arising, using inherent capabilities of the system;

RECONSTRUCTABILITY - to be able to remove and re-build major parts of the system;

MAINTAINABILITY - to perform error correction, testing and modifications efficiently;

EXPANDABILITY - to enhance the scope of the system applicability easily.

Approaching the mentioned characteristics leads, normally, to a shift of total costs, across the system life cycle, from labour (mainly maintenance of software) to machine (more computer resources required to absorb increased generality). Within labour costs, the shift is, again, from maintenance (unstructured, unmanageable) to design and administration (structured and well manageable tasks).

An impact of applying data base management methods can be followed in the examples shown in Fig. 1 and Fig. 2. In Fig. 1, two applications directly access data in storage. Problems appear if they access the same data. In addition to the fact that they are dependent on the data formats in storage, they become dependent on each other. Any change made to data formats because of the needs of one application will affect the other one. In a case like this, storage formats and operations are not transparent to the applications, flexibility of the whole system is very poor, and so on.

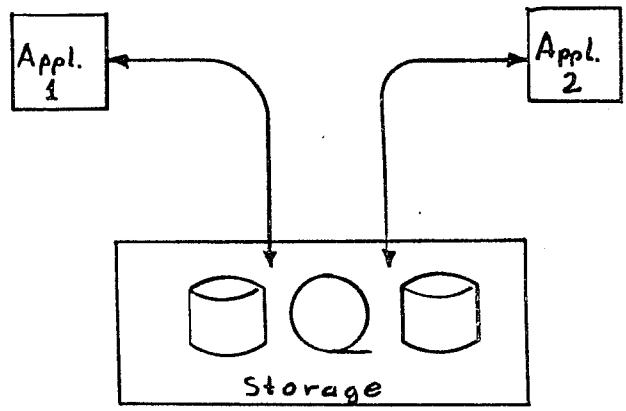
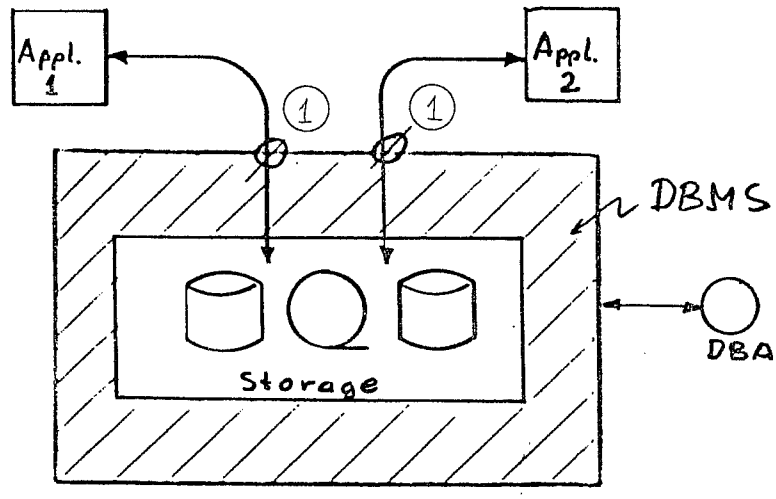


Fig. 1 Directly accessed data



FEATURES

- DATA INDEPENDENCE
- DATA DEFINITION LANGUAGE
- DATA MANIPULATION LANGUAGE
- ADMINISTRATOR'S TOOLS

Fig. 2 Data Base Management System

And here, a data base management system (DBMS) comes (Fig. 2). It isolates all the applications from the storage providing a new interface (marked with number 1) to the data. Within the interface, logical structures of data are seen rather than the actual physical ones, and the data are manipulated within the defined structures. A feature of data independence appears whereby the applications are independent of the physical storage formats (physical independence) and, additionally, changes or enhancements made to the overall logical view of data need not affect the applications that are not concerned with the change (logical independence). Logical views (structures) are maintained using data definition language(s) (DDL), and the applications access data by means of an appropriate data manipulation language (DML). A new personnel function of the Data Base Administrator (DBA) appears in the system. He maintains usability and integrity of the data base using special tools provided by DBMS.

3. Models of data base management systems

A model graph is used whereby major system processes and interfaces are shown. A model graph may show different phases involved in data processing activities, like: execution (or run), preparation of applications, and preparation of the data base environment, i.e. data base administration.

Fig. 3 illustrates a model of a system where no data base software is used. Therefore, it corresponds to the situation shown in Fig. 1. Two phases are shown: the run-time phase where the application process performs direct accesses to the storage using direct access facilities (interface 2), and the preparation phase where a programmer encodes storage-dependent data structures in the program, and uses a corresponding access method (interface 1).

A DBMS appears in the model presented in Fig. 4. The system contains one level of logical data definition, whereby the data independence (both physical and logical) is significantly improved. Data definition is established during the administration phase. It is expressed in a Data Definition Language (interface 3) as a schema and is compiled into a run-time-system - legible descriptor (interface 4). The descriptor is used by a run-time transformation process in order to perform mapping between the logical view and the storage. The user is separated from storage by means of a system-supplied Data Manipulation Language and the schema listing showing the logical view of data. Most of the available generalised DBMS' conform with the presented model.

A two-level schema definition model is shown in Fig. 5. The degree of logical data independence is increased by adding a new, application-related logical view - expressed by means of a sub-schema. A corresponding transformation is added to the

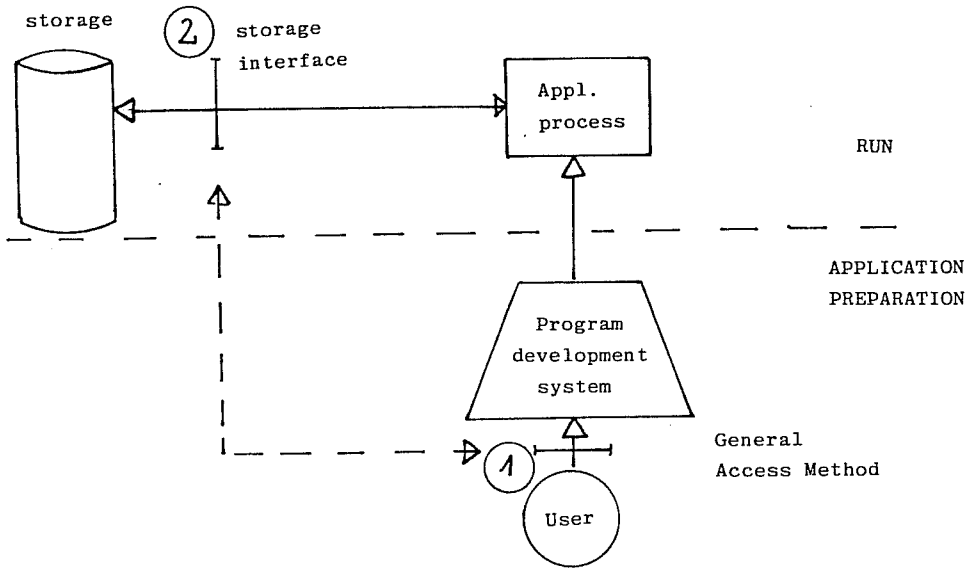


Fig. 3 Ø-level model (hard-coded data description)

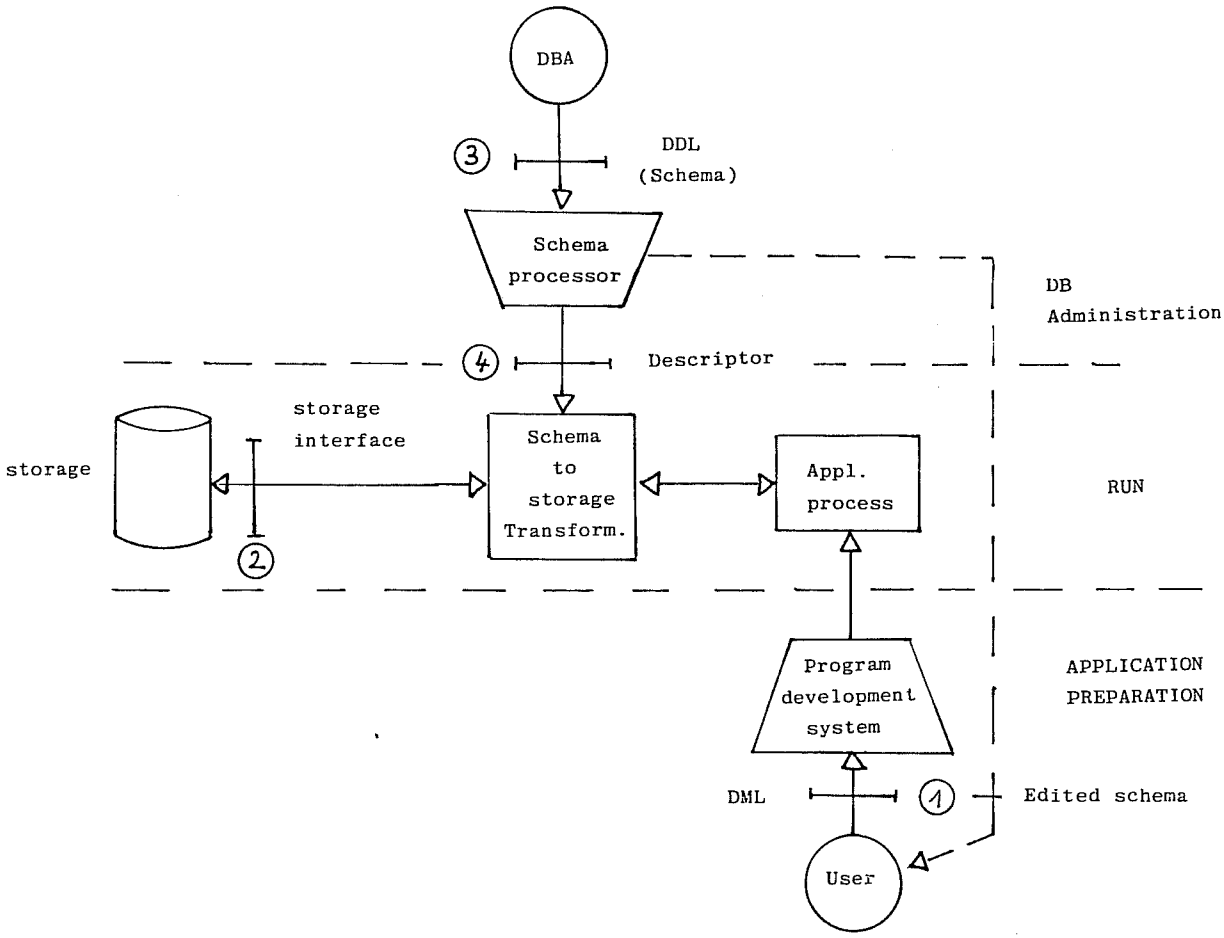


Fig. 4 1-level model (one schema)

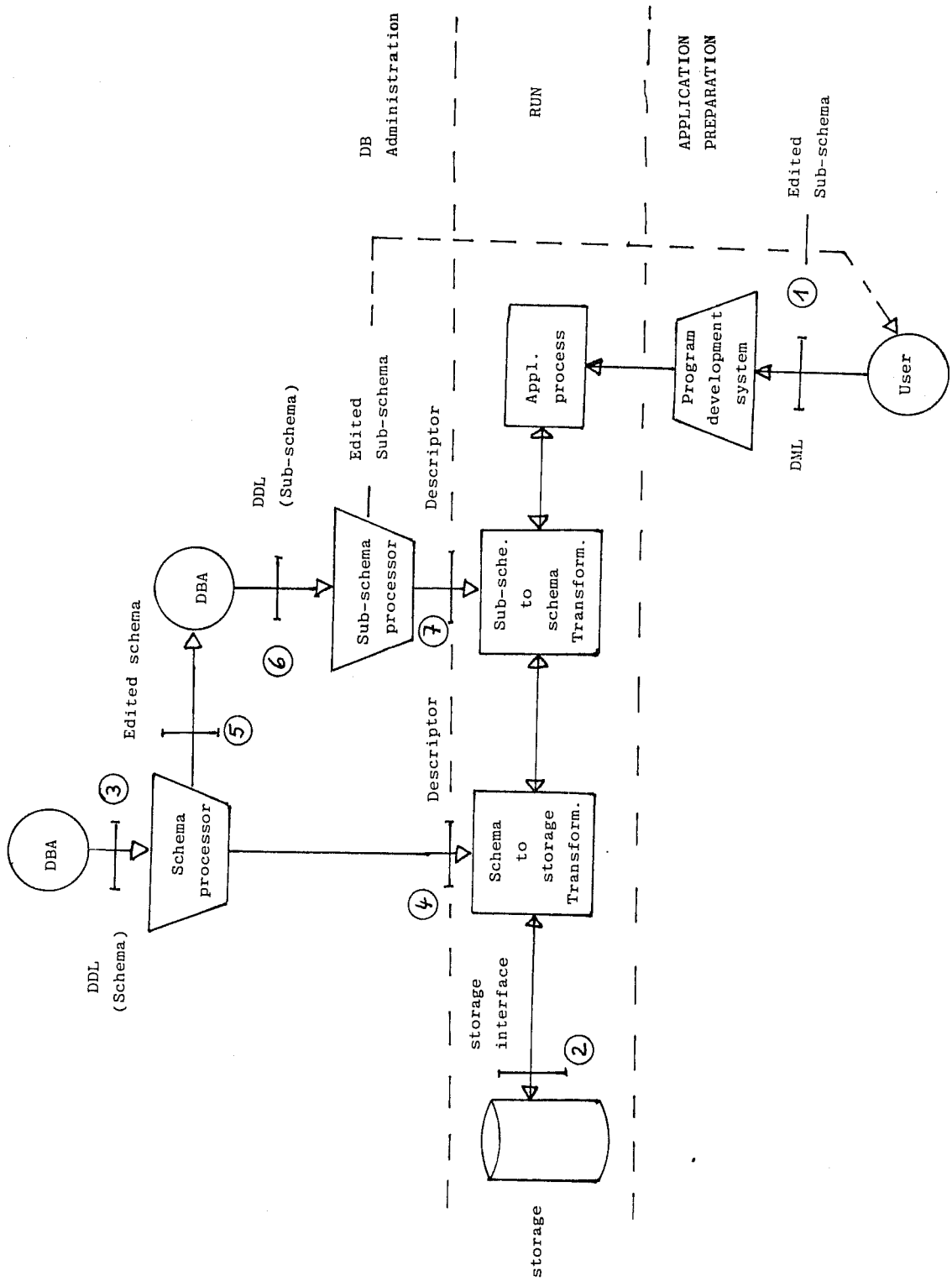


Fig. 5 2-level model (CODASYL)

run-time phase, and the user is presented with an extracted view of data only. As a result, a change made to the overall logical data view (schema) need not affect the application/user view (sub-schema). Quite a few available systems conform with the model.

In 1975, the ANSI/SPARC committee came up with a new model for data management (Jardine, 1977). It contains two new concepts (Fig. 6). The first one is the conceptual schema (interface 3). It is supposed to represent a "best data model" of the entities and their properties in the real world, without any reference to the world of machines. The main idea about this approach is that, once the conceptual schema is successfully constructed, any application-related view (here: an external schema) can be extracted, even if it has not been anticipated before. The conceptual schema may be expressed in terms of a special language conforming with the task of modelling. On the other hand, the external schema and related manipulation language have to be oriented towards the environment of a programming language.

The second new concept, the internal schema (interface 8), increases the degree of physical data independence. It provides the means to change storage-level representations of the conceptual structures without even affecting the conceptual schema. The change affects only the lowest level descriptor and the corresponding transformation.

4. Applicability of the model

The so-called "ECMWF Data Handling System" is supposed to maintain huge amounts of meteorological reports, fields and miscellaneous files in a physically separate "Data Handling Processor" (DHP), to be accessible at users' host computers connected via a high-speed computer interconnection (data highway) to the DHP. The problem is that of a combined data management/networking system.

In the data management part, a data base methodology could be considered. The system objectives listed in paragraph 1 fully conform with the expectations associated with the Data Handling System. Another list of objectives, cited by James Martin (1975), is presented in Fig. 7. The list is fully acceptable as well.

A question may be asked: what are the differences between the Data Handling System and a typical data base? One significant difference is the nature of entities and related data elements. Size of the elements (fields) itself implies special ways of handling. Data Languages, if used, would have to be different. Transformations would be different, as well as data manipulation facilities. The differences are significant at some levels of system design, very likely to the point where no ready-made software can be applied.

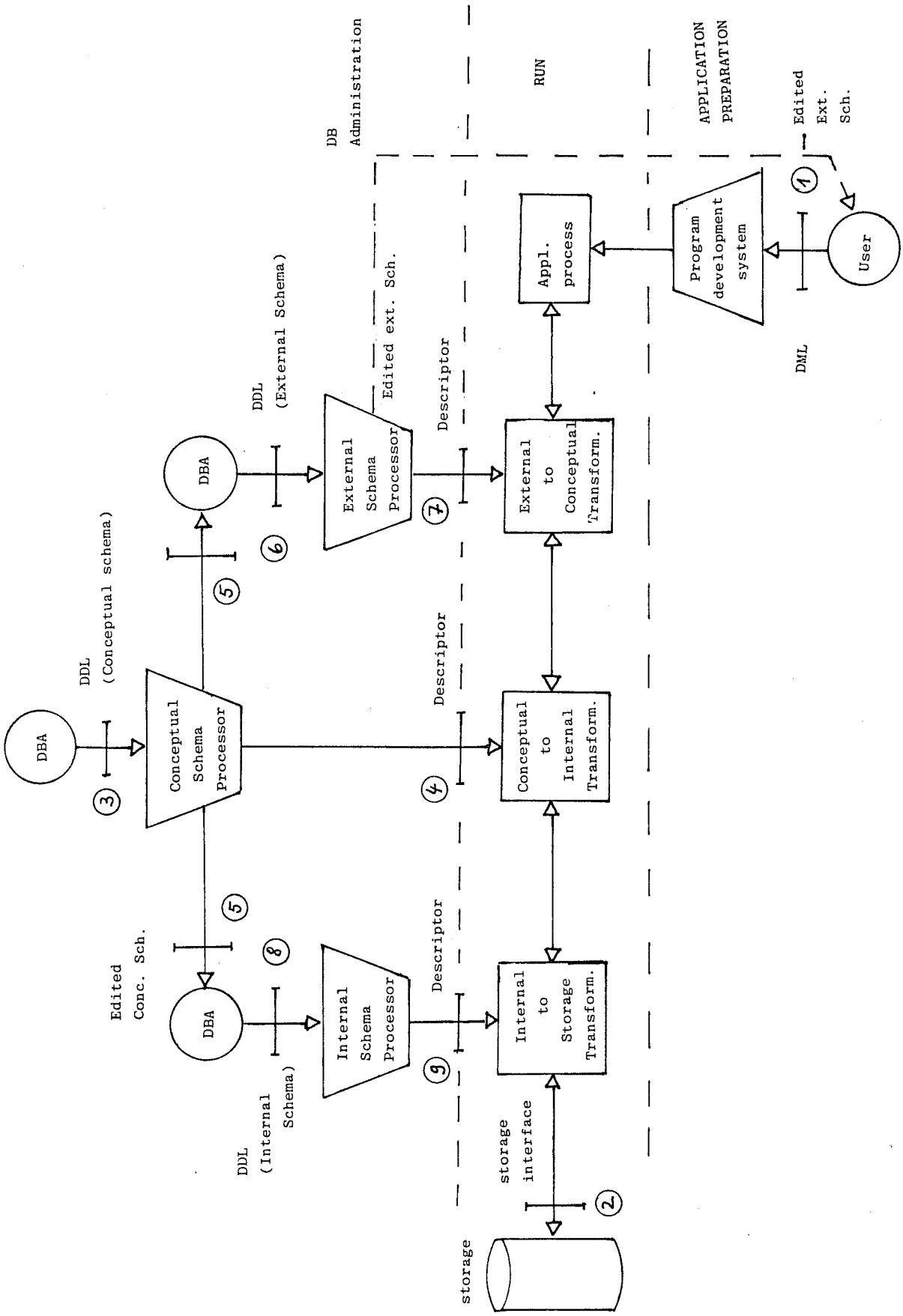


Fig. 6 3-level model (ANSI/SPARC)

PRIMARY OBJECTIVES OF DATA-BASE ORGANIZATION

- **The Data Base is the Foundation Stone of Future Application Development**
  It should make application development easier, cheaper, faster, and more flexible.
- **The Data Can Have Multiple Uses**
  Different users who perceive the same data differently can employ them in different ways.
- **Intellectual Investmentment Is Protected**
  Existing programs and logical data structures (representing many man-years) will not have to be redone when changes are made to the data base.
- **Clarity**
  Users can easily know and understand what data are available to them.
- **Ease of Use**
  Users can gain access to data in a simple fashion. Complexity is hidden from the users by the data-base management system.
- **Flexible Usage**
  The data can be used or searched in flexible ways with different access paths.
- **Unanticipated Requests for Data Can Be Handled Quickly**
  Spontaneous requests for data can be handled without application programs having to be written (a time-consuming bottleneck), by means of high-level query or report generation languages.
- **Change Is Easy**
  The data base can grow and change without interferring with established ways of using the data.
- **Low Cost**
  Low cost of storing and using data, and minimization of the high cost of making changes.
- **Less Data Proliferation**
  New application needs may be met with existing data rather creating new files, thus avoiding the excessive data proliferation in today's tape libraries.
- **Performance**
  Data requests can be satisfied with speed suitable to the usage of the data.
- **Accuracy and Consistency**
  Accuracy controls will be used. The system will avoid having multiple versions of the same data item available to users in different stages of updating.
- **Privacy**
  Unauthorized access to the data will be prevented. The same data may be restricted in different ways from different uses.
- **Protection from Loss or Damage**
  Data will be protected from failures and catastrophes, and from criminals, vandals, incompetents, and persons who might falsely update them.
- **Availability**
  Data are quickly available to users at almost all times when they need them.

Fig. 7 Objectives of Data-Base Organisation (Martin, 1975)

When a system model, such as the 3-level ANSI/SPARC one, is taken into consideration, the differences are not seen at the gross model level. The model remains valid as a methodological framework and can be used successfully.

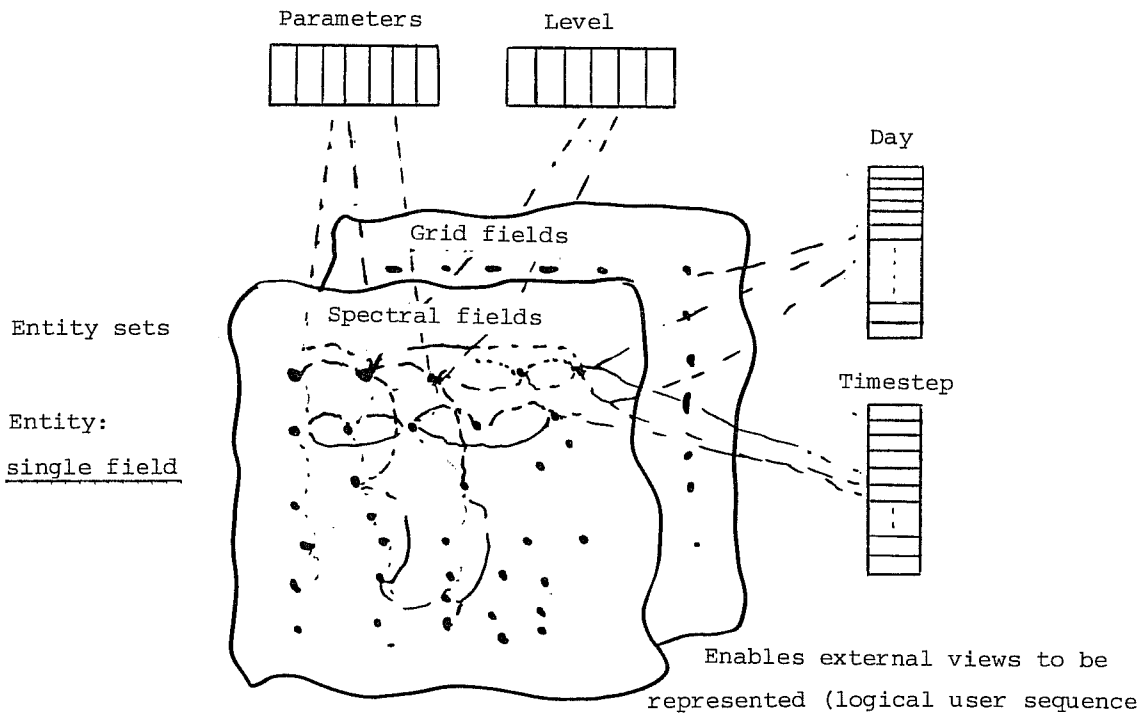
As an example, a suggestion for a conceptual schema (or general data model) of the fields data base is presented in Fig. 8. It is a network data structure where the elements (fields) are organised into superimposed sequences, the sequences being meaningful from the usage point of view (e.g. all the fields pertaining to a specific level or day, etc.). As a result, it may be anticipated that most of the retrieval requests can be matched to the conceptual view structures.

5. Combined model

Because the Data Handling System will function in the data communications environment, there is a need for a combined data management/network model. A basic representation presented in Fig. 9 is suggested. The right side of the picture takes the form of the standard layered representation of the Open System Interconnection (OSI) model. The left side is a simplified, 3-level data base management model shown in the environment of DHP. Both parts are connected by means of the application-level processes (data handling servers).

A model like the one presented provides a base for further systems study. One role is to provide questions. Here are, in place of conclusions, some resulting questions:

What is the required scope of model generality?
How many data management transformations needed?
What data languages should be applied?
Which data management levels are located in DHP?
How is the network interfaced to data handling?
Are the network transformations necessary?
Will the archive data base be distributed?



- Network model

- Data independence

Fig. 8 Conceptual view (basic data model of ECMWF)

(permanent) archives

(fields only)

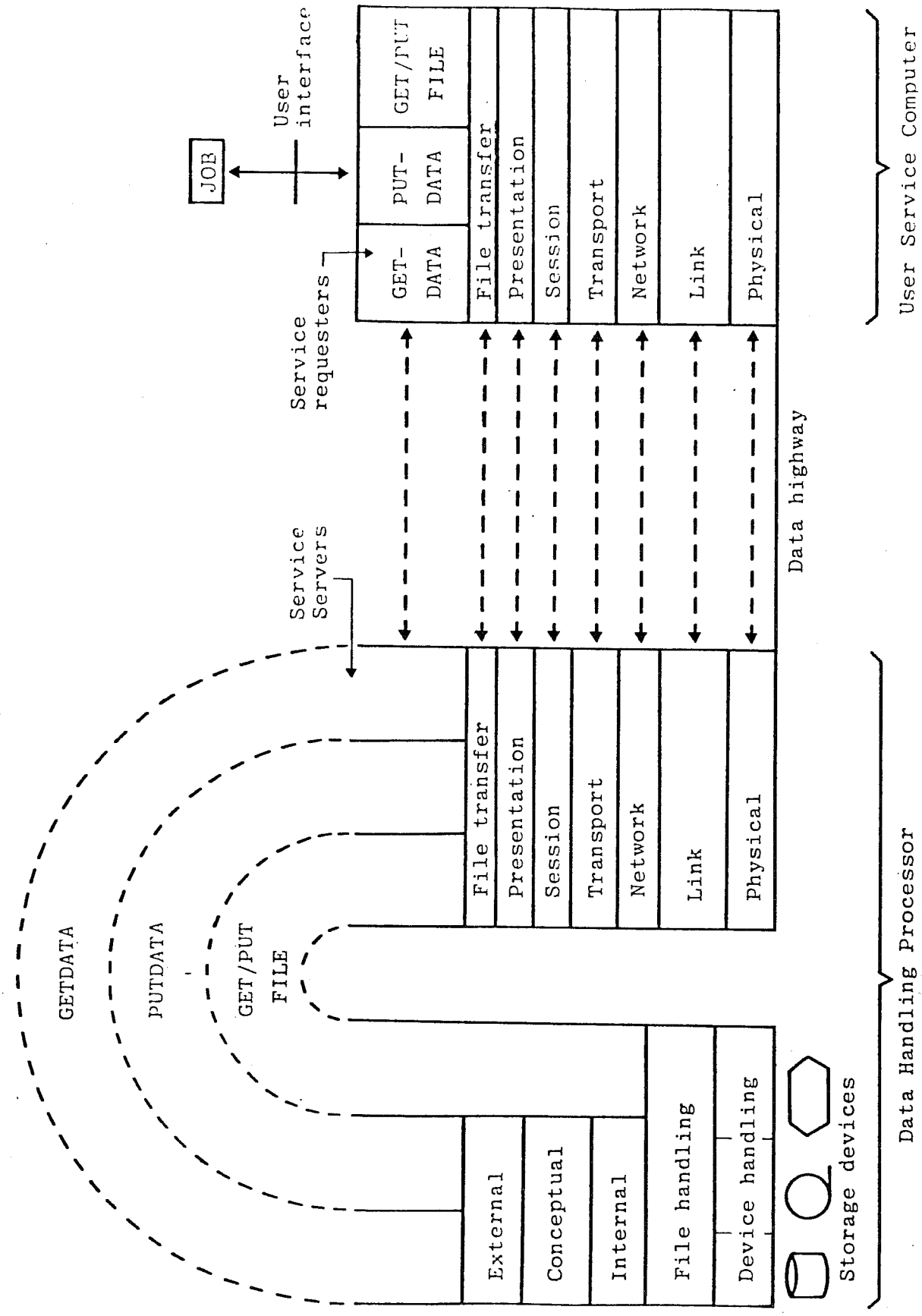


Fig. 9 Representation of combined data management/network model

REFERENCES

Jardine, D.A. (ed.), 1977: The ANSI/SPARC DBMS model. Proceedings of the second SHARE Working Conference on Data Base Management Systems, Montreal, Canada, 26-30 April 1976, North-Holland Publishing Co.

Martin, J., 1975: Computer Data-Base Organisation, Prentice-Hall.

Nijssen, G.M. (ed.), 1978: Modelling in data base management systems. Proceedings of IFIP Working Conference, Freudenstadt, Germany, 5-8 January 1976. North Holland Publishing Co.

Tsichritzis, D.C. and Lochovsky, F.H., 1977: Data Base Management Systems, Academic Press.