Title : *Modular architecture for CAD software*

Author : *Jacques LE QUERE*
CENTRE SCIENTIFIQUE ET TECHNIQUE
DU BATIMENT
FRANCE

Summary :

Building construction information systems were turned upside down last years with the concept of data base. Technical CAD softwares must be designed today regarding this concept. This paper presents an general design method for technical CAD softwares, suited to data bases use.

1. INTRODUCTION

This paper concerns software engineering for technical CAD systems for building construction projects. Hardware et software tools have evolved, and some software quality factors become dominating :
- software correction,
- software maintenance,
- opening towards other softwares, like data bases, to share or exchange data.

We presents some methods suited to improve these quality factors :
- general architecture of software,
- data dictionnary
- data structure
- dialog module -data base
- module architecture

2. General architecture

In order to reduce costs of development, it is desirable to use existing softwares such as :

- data base managment systems (DBMS)
*- data editing*
- man computer interface
The general architecture presented on illustration 2.1 is suited to use such software tools.

3. Data dictionary

The first result of analysis the data dictionary of the software. This is the most important tool to realize a software and to exchange data. The data dictionary describes all the significant objects that the computer will manipulate. The description can look like this :
- object name
- abreviated name
- definition
- characteristics for computer
- characteristics for man computer interface
- constraints

3.4
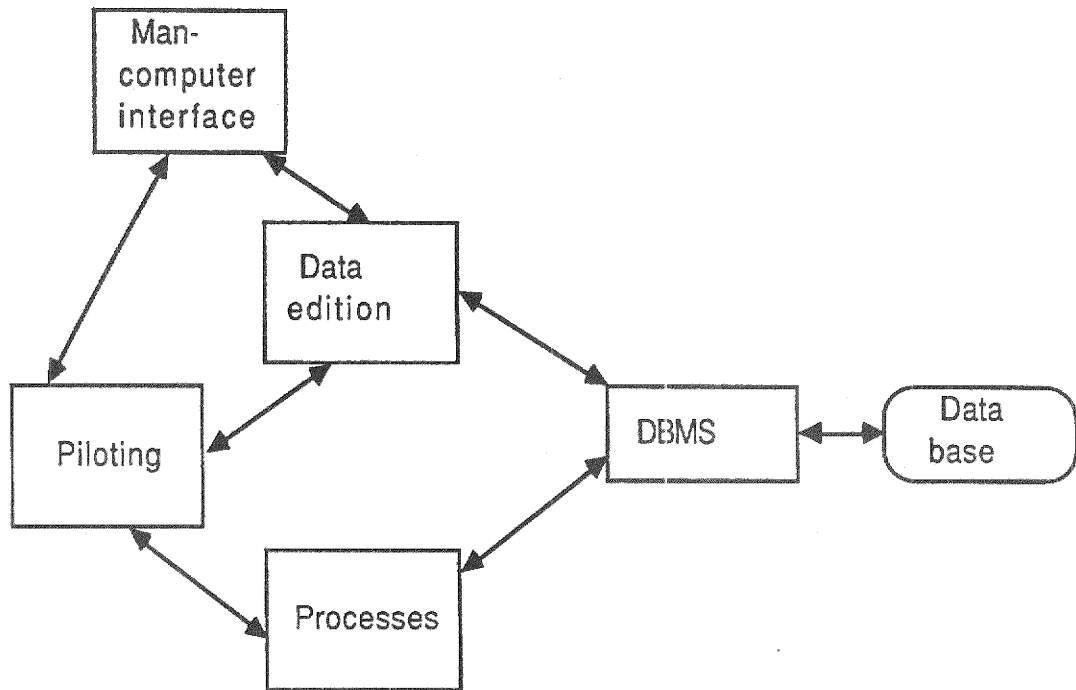
Illustration   2.1

General   architecture

4. Data structure

Data structure is an element of a data base. The structure regroups data into entities and links entities.This data structure can help to design a software related to the data. One can create module related to entity ; one can declare variables in a programming language using the data dictionary. The illustration 4.1 shows and example of a data structure an of an entity.
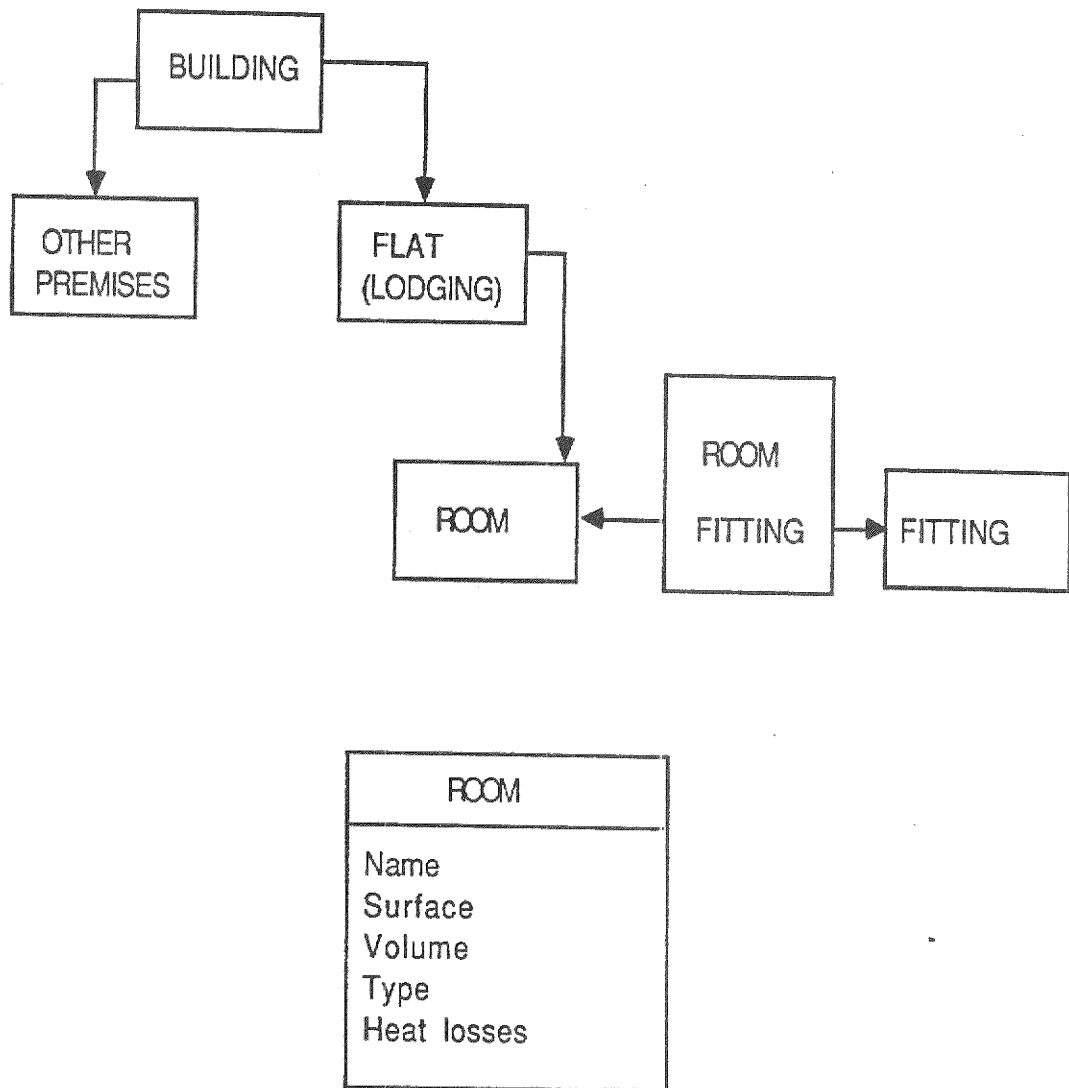
5. Dialog module-data base

So, we have seen that :

- it is useful to use software tools as a DBMS,
- the data structure of a data base is helpful when designing software.

More, a data base can obviously help to share data.

The illustration 5 shows how one can organize the dialog between a module and the data base. It is interesting to have data with the following structure :

- reading data in the data base
- processing data
- writing results in the data base.

```
                    ┌──────────────┐
                    │   BUILDING   │
                    └──────────────┘
               ┌───────┘        └───────┐
               ▼                        ▼
     ┌──────────────┐          ┌──────────────┐
     │   OTHER      │          │    FLAT      │
     │  PREMISES    │          │  (LODGING)   │
     └──────────────┘          └──────────────┘
                                       └────────┐
                                                ▼
                              ┌──────────┐  ┌──────────┐  ┌──────────┐
                              │          │  │  ROOM    │  │          │
                              │  ROOM    │◄─│ FITTING  │─►│ FITTING  │
                              └──────────┘  └──────────┘  └──────────┘
```

ROOM
Name
Surface
Volume
Type
Heat losses

Illustration   4

Example   of data structure

6. Module architecture

Hows to organize and structure the modules when using a data base. We examine the following standard cases ?

6.1 Module- entity relation.

In this case, all data read or written by a module belong to a single entity, as presented on illustration 6.1.

The same relation applies when the module read or write data in entities which have only one occurence in the data base.
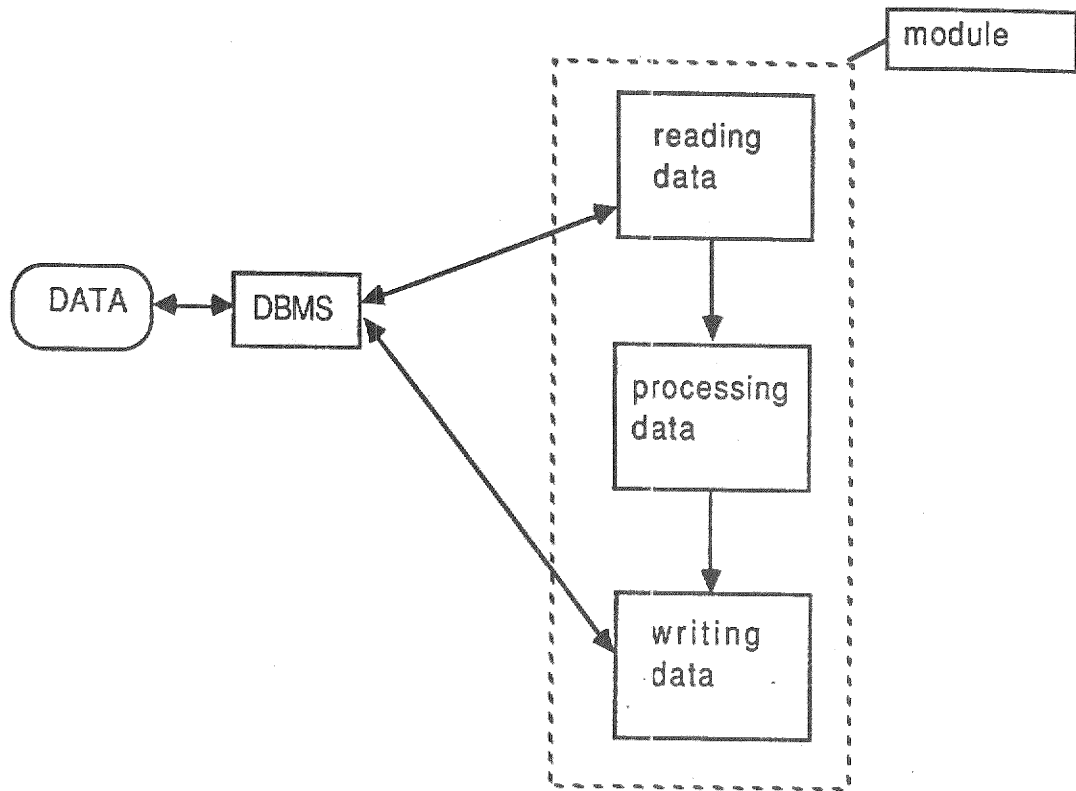
6.2 Hierarchic structure

When the data base has a hierarchic structure, as "flat is composed of rooms", one can consider the organization of illustration 6.2 where the function of summation is introduced.

6.3 Relational structure

In this case, a relation is set between two entity. One have the organization shown on illustration 6.3, where data can be read from entities, but can only be written in the relation.

7. CONCLUSION

We have presented general methods to design software in relation with a data base. These methods lead to a software architecture linked with the data base structure. The modularity of software can improve the correction of sofwares and make the maintenance easier. The use of a data base enables the sharing of data and the exchange of data.
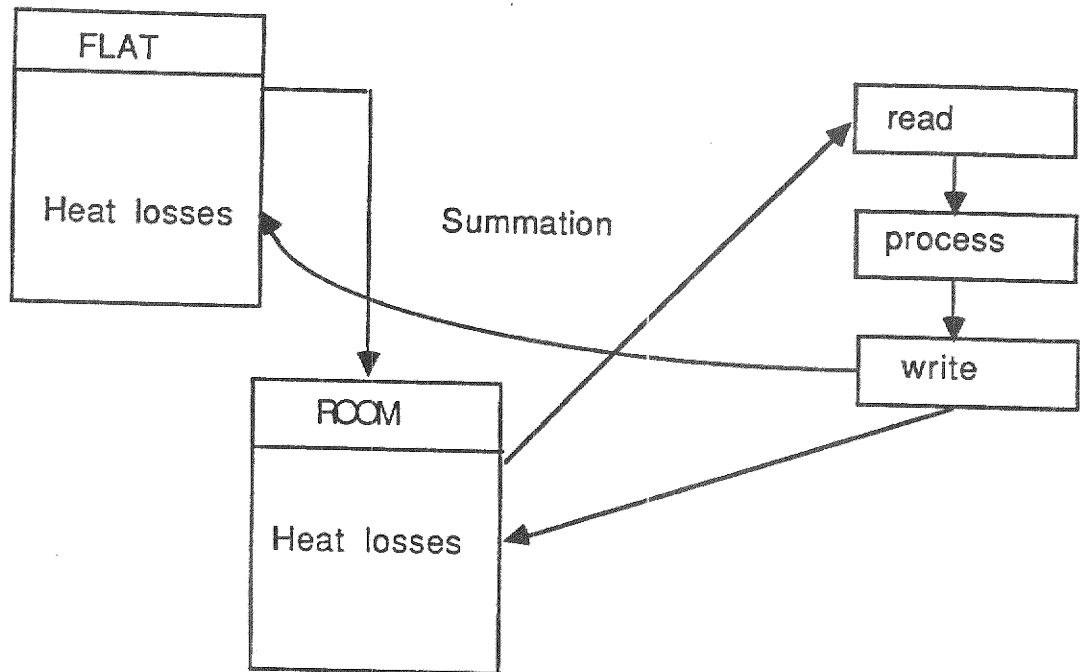
Illustration 5

Structure of the dialog module - data base
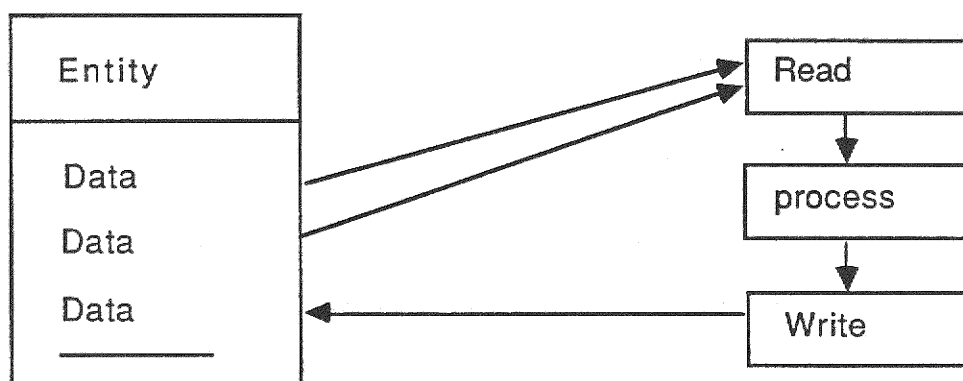
Illustration   6.1

module-entity   relation

Illustration 6.2

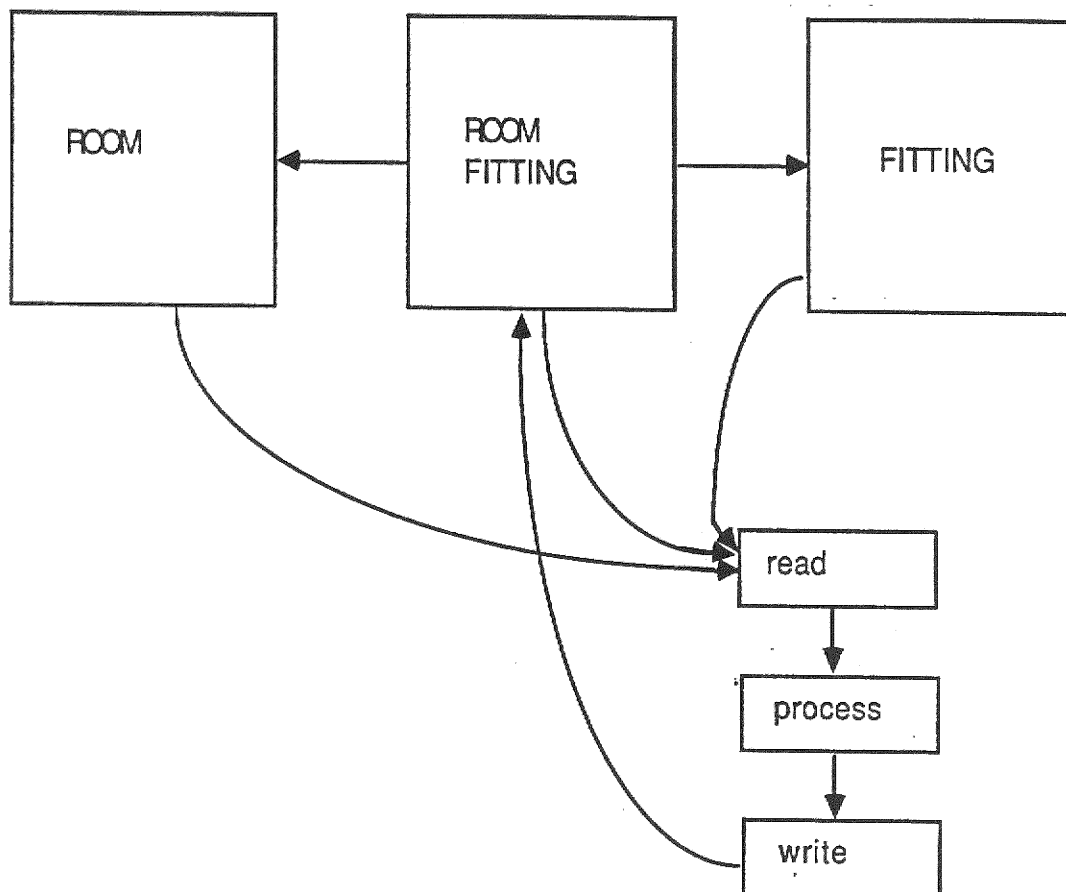**Module-entity relation in a hierarchic data base**

Illustration 6.3

Module-entity relation in a relational data base