

**From a “well designed” database to  
AD/CYCLE tools:  
a reengineering experience**

**Oreste Signore - Francesco Celiano**  
*CNUCE - CNR via S. Maria, 36 56126 PISA*

**SHARE EUROPE**  
*Spring Meeting 1991*

*Lausanne*  
*April 8-12, 1991*

# Content

- **Introduction**
- **The application: an informal specification**
- **The manual methodology approach**
  - **The DATAID methodology**
  - **The database**
  - **The software architecture**
- **The CASE tool approach**
  - **IEW**
  - **The usage of the tool**
- **Evaluation of the experience**

---

\* This work has been partially supported by "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo" of C.N.R..

# Introduction

- **CNUCE is an Institute of CNR (National Research Council) of Italy**
- **his aim is to perform research activity**
- **it gives computing power to scientific community**
- **users' requests are on a non scheduled basis**
- **In 1983 a database approach was adopted**
- **In 1990, the decision was taken to migrate on a different DBMS**

## **The application: an informal specification**

- **Identified functions in the user management area:**
  - **Management of the library (sale of the manuals and other documentation).**
  - **Management of user access accounting codes and registration of people allowed to get access to the resources.**
  - **Management of resources (virtual machines, tapes, disks, etc.), that is, linking of resources to the appropriate user accounting codes, and recording relevant information about every transaction.**
  - **Management of invoices (every resource must be accounted to the customer).**
  - **Evaluation of the resource usage in order to plan the development of the computer center.**
- **Every user, in order to gain access to the CNUCE computer resources, must sign an agreement, and get one or more user codes. For each user code, a representative is identified, who can ask for every resource the code is allowed to ask for, and can grant this permission to other persons.**
- **Each request by the user can modify the number and the kind of allocated resources. In addition, even if a code is canceled, information must be kept alive for a minimum of two years (for accounting and possible claims), even if resources may be allocated to other customers.**

# **The manual methodology approach**

- **Introduction**
- **The application: an informal specification**
- **The manual methodology approach**
  - **The DATAID methodology**
  - **The database**
  - **The software architecture**
- **The CASE tool approach**
  - **IEW**
  - **The usage of the tool**
- **Evaluation of the experience**

## **DATAID methodology**

- **Defined by several groups in the context of the “*Progetto Finalizzato Informatica*”, a 5-year research project financed by the National Research Council of Italy.**
- **Essentially a manual methodology (some components were automated),**
- **Conceived as a methodology for database design.**
- **Analysis of the enterprise supposed as already done.**
- **Concentrating on tools for supporting the sequence of the phases for database design:**
  - **user requirement collection and analysis**
  - **conceptual design**
  - **logical design**
  - **physical design**

## **DATAID methodology (cont.)**

- ***User requirements collection and analysis***
  - getting information from user interviews,
  - collecting into glossaries and forms  
(Data Glossary, Operations Glossary, Event Glossary)
  - Constraint requirements also collected.
- ***Conceptual design***
  - based on the process of views integration
  - The data model:  
The Entity-Relationship model with some extensions:
    - ***Abstraction hierarchies:***
    - ***Aggregates:***
    - ***Repeating attributes:***
    - ***Identifiers:***
  - Dynamic aspects represented by means of Petri nets.
- **Basically data oriented**
- **Procedural aspects not emphasized.**
- **Specific tools, oriented toward specific DBMS families, were developed for logical and physical database design .**

# The database: conceptual schema

# The database: logical schema

# The software architecture

- **Two distinct aspects: report generation and update.**
- **Report generation:**
  - extensive usage of the SYSTEM 2000 strings
  - flexibility
  - easy implementation of new reports and of simple help.
- **Update transactions:**
  - Every application against the database could be seen as a navigation along the tree
  - Development of a general purpose, form based, software architecture.
  - A general purpose I/O routine was accessing an application dictionary (SYSTEM 2000 based, too) in order to build a standard 3270 map.
  - The application program had simply to ask for the appropriate schema record: the I/O routine was in charge of controlling every user/system interaction, and verify every constraint on the fields
  - The application main program had simply to control the procedural constraints and the logic.
- **Some constraints implicit in the DBMS data model**

- **Some traditional programs for the batch updates.**

# The CASE tool approach

- Introduction
- The application: an informal specification
- The manual methodology approach
  - The DATAID methodology
  - The database
  - The software architecture
- The CASE tool approach
  - IEW
  - The usage of the tool
- Evaluation of the experience

## **IEW: generalities**

- **IEW (Information Engineering Workbench)**
  - **one of the the CASE product leader in the market**
  - **selected by IBM in AD/Cycle.**
- **Implements the approach of Information Engineering**
- **Basic assumption:**  
**analysis and design for computing can be thought as having a four level, pyramid structure, concerned with the two aspects of *data and processing*.**

# IEW: the DP pyramid

- ***Information Strategy Planning:***
  - high level overview of an enterprise, its functions, and its information systems strategy.
  - A strategic overview of data and functions is taken.
- ***Business Area Analysis***
  - what fundamental processes are needed to run a business area, how they interrelate, what data are needed.
  - **Data:** fully normalized logical data model
  - **Processing:** description of the necessary processes and their interdependence.
- ***System Design :***
  - how the processes are implemented as procedures, how the procedures work, and how they interrelate.
  - **Data:** design of the records used by procedures
  - **Processing:** design of procedures and how they relate.
- ***Construction:***
  - design of programs, implementation of data structures.
  - **Data:** storage structure and program view of data
  - **Processing:** detailed program logic.
- **All tasks supported by a variety of diagramming techniques**  
(Action diagrams, Entity-relationship diagrams, Decomposition diagrams, Data flow diagrams, etc.).
- **For each level, a specific workbench.**
- **Data stored in the *Encyclopedia:***
  - central repository of knowledge
  - allows the diagrammer tools to be integrated.

## **The usage of the tool: Planning Workstation**

- **Major emphasis on the organization structure analysis,**
- **Size of the application was not so large to constitute a really effective test-bed.**
- **Analysis of the organization was lacking in the manual methodology.**

# **Planning Workstation: a Decomposition Diagram**

## The usage of the tool: Analysis Workstation

- The conceptual schema was modified
- Programmers, in charge of developing the software, were involved (an overhead projector was used)  
Their contribution resulted very effective.
- The interactive Entity-Relationship editor resulted very useful,

... but ...

the Entity-Relationship model adopted by IEW semantically poorer than we were used to have in the DATAID methodology.

- The *generalization* concept is not supported: decisions that could be postponed up to the phase of logical design must be taken at higher level.  
Possible drawbacks:
  - resulting conceptual schema graphically more complicated (hierarchy splitted into many entities)
  - the designer must guarantee semantic constraints implicit in the richer model (hierarchy implemented as a single entity)  
But there is no way to formally state the constraints!
- The *n-ary relationships* are not supported.

**Associative entities may be used for this purpose and in order to represent relationship with attributes.**

## The usage of the tool: Analysis Workstation (cont)

- ***Multiple attributes are available***  
(this avoid the presence of useless entities in the schema).
- **Attributive entities may be used.**  
This feature is particularly useful in the aim of representing *aggregate attributes* (repetitive or non repetitive).
- ***Procedural aspects* represented by means of:**
  - data flow diagrams
  - mini-specs.
- **Data Flow Diagrams not suitable for the representation of the temporal sequence of events and procedures.**  
The concept of Data Store may be overloaded, but this practice should not be encouraged.

# **Analysis Workstation: Entity-Relationship Diagram**

# **Analysis Workstation: Entity Description**

# **Analysis Workstation: Decomposition Diagram**

# **Analysis Workstation: Mini-Specs**

# **Analysis Workstation: Data Flow Diagram**

## **The usage of the tool: Design Workstation**

- **The logical database schema was generated**
- **DDL was easily modified for usage of SQL/DS**
- **Mapping between the conceptual schema and the logical schema is fully automatic.  
There is no way the user may affect this process**

# **Design Workstation: Database Structure**

# **Design Workstation: a Relation**

## **The usage of the tool: Construction Workstation**

- **No use has been made of this workstation**

## **The usage of the tool: General remarks**

- **Documentation:**
  - graphical representation of data is very effective
  - textual reports verbose
  
- **Integration:**
  - the vendor asserts that IEW is a fully integrated tool
  - few cases (i.e. data types) where the vertical integration between the workstations appears to be scarce, as the modifications made at lower levels are not automatically exported toward higher levels.
  - In the MS-DOS version, it is necessary to quit one workstation before activating another one. May be this aspect will be greatly affected by the migration in OS/2 environment.

# Evaluation of the experience

- ***Analysis of the organization.***

This aspect was not considered in the manual methodology. The use of the tools supplied by the Planning Workstation gave a valuable help in considering some aspects that were not so much emphasized in the first implementation of the application.

- ***Data Analysis and constraints representation.***

- **DATAID:**

- **Semantically richer data model**
- **(Many implicit constraints. + free text for other constraints).**

- **IEW:**

- **excellent user interface**
- **increased participation of the users**
- **lacking of some aspects peculiar to the semantic data models**
- **resulting conceptual schema less readable.**

## Evaluation of the experience (cont.)

- ***Dynamic aspects***

- **DATAID:**

- theoretically solid formalism (Petri nets)
- concerned more with data than with operations
- filling of the forms and the glossaries is not an easy task.
- usage suggested at a too low level of atomicity.

- **IEW:**

- Decomposition Diagrams, Data Flows e Action Diagrams.
- widely used
- temporal dependencies and alternatives.

- ***Mapping.***

- **DATAID:**

- task performed in several steps
- restructuring of the conceptual schema
- general rules, plus qualitative choices

- **IEW:**

- fully automatic
- user action only on conceptual schema

## Evaluation of the experience (cont.)

- ***Development time.***

For a project of this size, the development time has not been considerably reduced.

- ***Quality.***

- Does not seem to have been considerably improved.

- **Anyway, the adoption of a CASE tool leads to:**

- improvement of the understanding of the application
- accuracy of the data structures design
- clear software architecture
- detailed documentation.

- **Improvements more relevant if:**

- projects of significant size
- mean for introducing the usage of a more formal design methodology.