An original application of ORACLE RDBMS as a data control tool for the computational fluid dynamics software DIODORE is presented

AUTHORS : Evelyne Sorasio / Corinne Milles / Jean Bougis

DIODORE Système

SUMMARY :

The originality of this application is based on the fact that ORACLE RDBMS is used as a "data storage structure" rather than a "data bank" wich is the usual application. This structure permits each user to optimise his storage memory space and data access for his particular application.

Computational fluid dynamics is used in many applications for different industrial fields (aeronautics, hydrodynamics, acoustics ...). The software DIODORE is a common computation tool to all these different fields. DIODORE is a user-friendly, hight performance tool portable on all systems. ORACLE RDBMS has been choosen to manage the data flow DIODORE in several levels :

— the input data
— the data control
— the recovery of data storage from external files
— the computation running from interactive session in interactive or batch
— the possibility the storage and the data access during the computation
— the management of the output files.

The data stored in the data base are different and indépendant for each project. All the users share the structure of the tables.

DIODORE Système PRESENTATION

Head office	:	Rue Albert Caquot BP 23 06901 - SOPHIA ANTIPOLIS Cedex Tel. 92.96.03.33
Creation date	:	Fin 1991
Capital	:	3.930.000 FF
Shareholdden	:	CONTROL DATA (F/USA) DORIS ENGINEERING (F) EUROPROJECT (Spain) GESTFIN (Italy) ISIS (F) PRINCIPIA (F) SEMA (F/UK)
Chairman	:	J.F. COUDERT
Managing Director	:	J. BOUGIS
Scientific Director	:	H. VIVIAND

Development team including highly experienced engineers in :

- Computational fluid dynamics
- Hydrodynamics
- Software developments
- Numerical Analysis

AIM

Develop, market, maintain and provide technical assistance on industrial softwares in all fields relating to Computational Fluid Dynamics.

OBJECTIVES

Sell

• industrial softwares licences on the international market.

Develop products

• adapted to the industries'needs.

Expand the products

• to new applications
• to follow the users evolution of needs.

Realize developments

• Out of :

- published results
- private results under marketing licence
- results from research works.

• From public or private research centers as well as inhouse research.

PARTNERSHIPS

Scientific and Marketing partnership may be set up with organisations interest by the development of the product, or by its computing architecture or for marketing purpose, or ...

INDUSTRIAL FIELDS

- petroleum
- energy
- environment
- aeronautics and space
- army equipment
- automobile and transport
- nuclear
- indutrial fluids
- process engineering
- buildings and big structures.

DESCRIPTION OF THE DIODORE SOFTWARE

In order to solve hydrodynamic problems, PRINCIPIA has developed the "Diodore" software package (version 2) which is technically the best on an international level. Diodore represents a national success which has been very difficult to repeat internationally solely in the marine hydrodynamics segment.

An increase in computer power and a better knowledge of digital calculation methods for fluid mechanics now enable problems previously reserved to experimental work to be solved by the computer. As with software for structure mechanics, the demand for industrial packages for fluid mechanics is becoming increasingly important.

This development has led to our decision to include industrial fluid mechanics in the "Diodore" package, which has required a complety revised version 3 to replace version 2.

THE ORACLE CHOICE

The architecture of Diodore version 3 was completely re-designed based on the different scientific disciplines to integrate. Experience has taught us the problems to be solved and the improvements to be made in the field of scientific software.

Definition of requirements and the list of functional specifications have resulted in the following conclusions with respect to data and file management:

- Data entry mode should be interactive and conversational with a built-in help menu.

- Particular care should be given to checking entries in order to limit, as far as possible, data errors before starting calculations.

- Data entry management should be open-ended to enable enhancements as developments occur and should be general enough to allow for the integration of data from different disciplines.

- Access to data in the computation subroutines by the scientific developers must be simple and user-friendly.

- In view of the considerable amount of data to be stored and handled, special attention should be given to access times.

These requirements led us to consider a general database solution for data processing. As development costs of an RDBMS with interactive I/O portable on all systems are prohibitive, we opted for off-the-shelf products which met our requirements. Among those available, the ORACLE product seemed the most suitable.

ANALYSIS

We used the MERISE methodology [1] as a basis for this analysis. This third-generation method is used to design and configure information systems. It is perfectly suitable for conversational processing and relational databases such as ORACLE.

The first phase of our work consisted in identifying real life objects and sorting them into groups of interdisciplinary entities. This synthesis work was performed using the definition of data specific to each discipline. It can be split into the following categories:

- Identification of "identifying" objects (project title, etc);

- Identification of objects used to define the geometry (structure, limit, fluid area, reference, etc);

- Identification of objects representing the physical medium (fluid, substances, etc);

- Identification of objects representing the system of equations to solve (Laplace, Helmots, Euler, Navier-Stokes, etc);

-Identification of computation objects (problem, simulation, symmetry, universal set, etc).

Once this summary was completed, we drew up the exhaustive list of fields i.e. properties specific to each entity or to relations between two or more entities. We then defined the graph of functional dependencies from which we deduced all the tables with their access keys.

APPLICATION DEVELOPMENT

Following the analysis, the form-based interactive applications and interactive menus were developed respectively using SQL*FORMS V3 and SQL*MENU V5.

Figure 1 shows the diagram of the main menu in the form of "POP UP" menus. This main menu gives an overall idea of our application. The application is run as follows:

- The user must first create a new project or load an existing one.

- He must then enter all data on its geometry. Depending on specific criteria, such as mesh size or user's own facilities, he has the option of re-entering the charastics of his meshing either interactively or externally by loading an ASCII file (mesh generator output or neutral format).

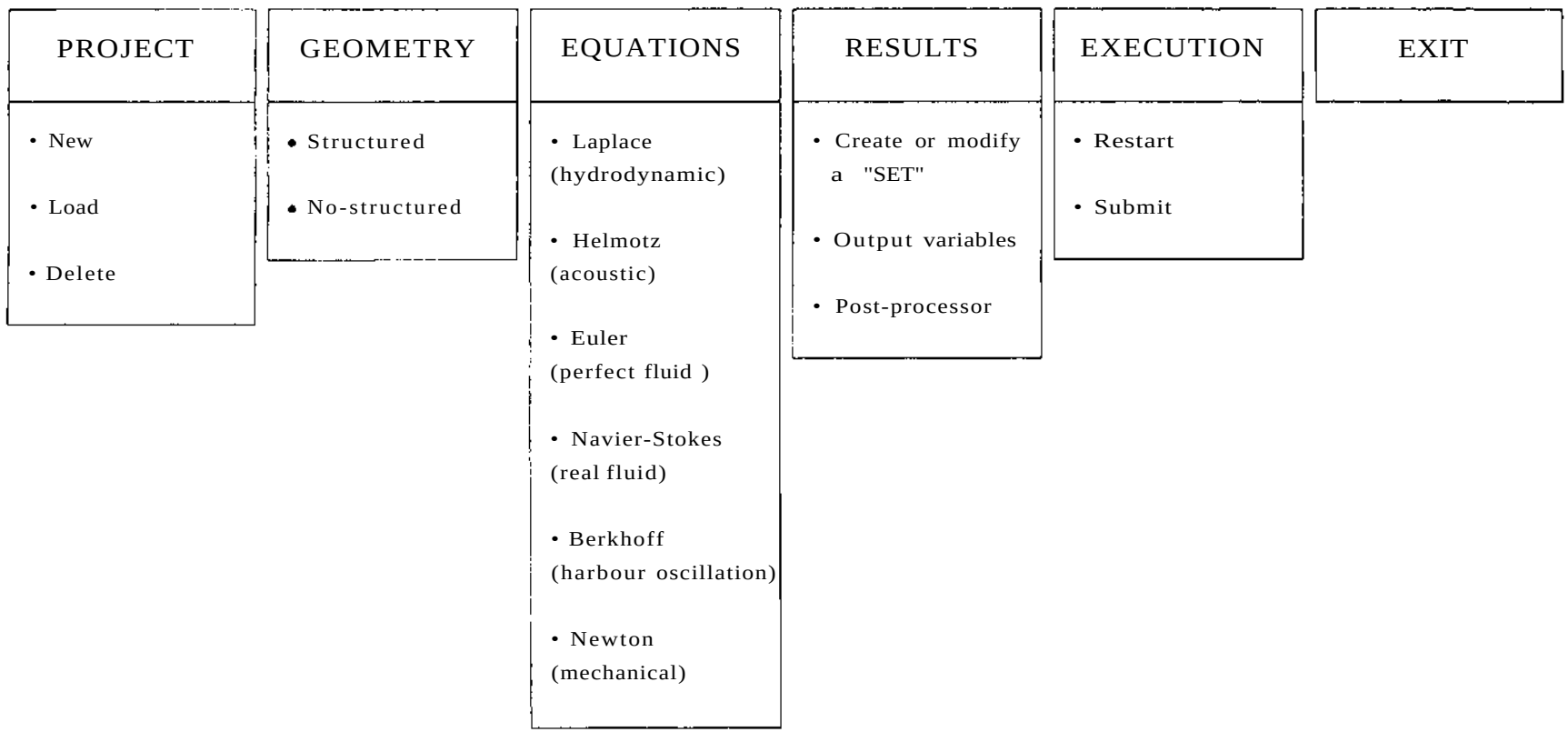
- Then he must select the type of system of equations to solve, which requires entering a certain number of data specific to the selected type. At this point, all data on the problem are defined.

- The user can then select the presentation mode of his results (report or chart).

- The last phase is the initiating of calculations. Batch processing is required for the high volume of calculation.

Figure 1 : DIODORE V3RO APPLICATION

Main menu



Any user selection results in a based-form interactive application. A screen can be composed of one or more blocks, in turn composed of one or more records. At different times during data update, checks can be made using triggers, interactively tripped either by the user (TRIGGER by key) or automatically by the application (TRIGGER by event).

The example below shows the development of the form which corresponds to the exact definition of the "ELEMENT" entity and of its relations.

ELEMENT **EXAMPLE**

* DEFINITION: The set of nodes which delimits a known geometrical form (element type) to which a certain number of values are attached. The number of element nodes depends on its geometrical type. The set of elements corresponding to a physical entity (structure, fluid area, etc) is a mesh. An element always belongs to a mesh and has an input order number in this mesh. The elements can be grouped into geometrical entities (for printout of results) called a "SET". One element can belong to several sets.

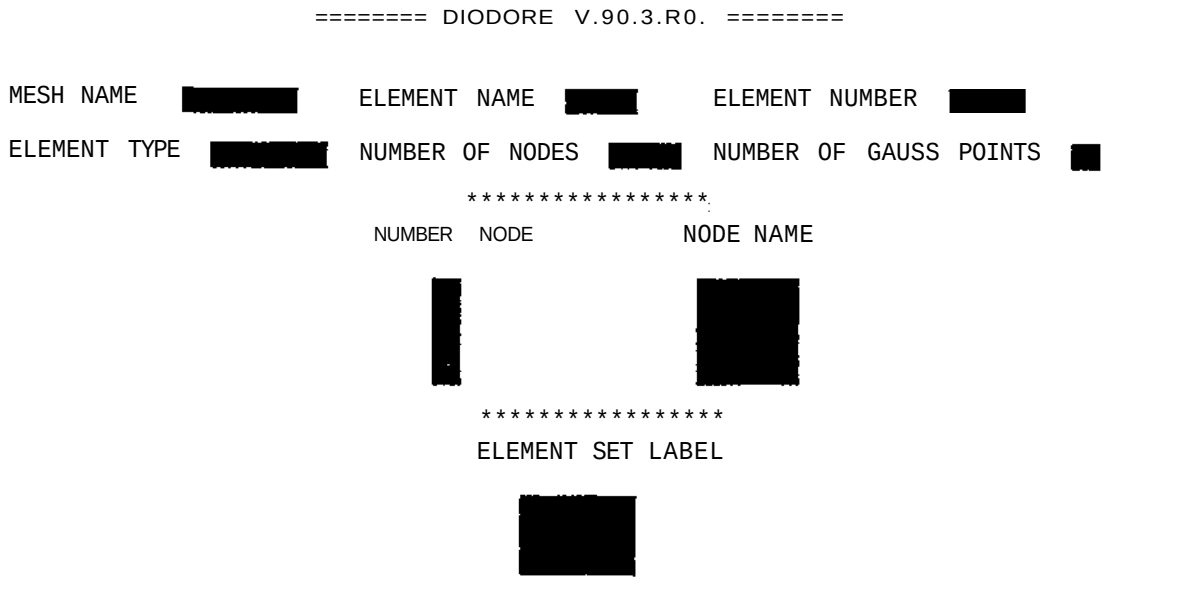
* NAME OF THE FORM: ELEMENT.

* FORM DEFINITION: Element definition.

An element definition screen is composed of three blocks. The first block identifies an element. The second block indicates its composition (nodes). The third block indicates the set to which it may belong (optional).

Name of tables: ELEMENT (for block 1)
ELNOEUD (for block 2)
ELSET (for block 3).

—"ELEMENT" FORM



* VALIDATION TESTS UNDERTAKEN ON FORM DATA:

For block 1 (ELEMENT):

- NOMAIL (mesh name) field: tests if mesh name already exists in database. If not, mesh name validation is refused.

- NOEL (element name) field: tests if element name already exists in database. If yes, element name validation is refused.

- ELTYP (element type) field: tests if type name exists in database. If not, type name validation is refused.

For block 2 (ELNOEUD):

- NOND (node name) field: tests if NOND field exists in NODE table. If not, node name validation is refused.

* OPERATIONS UNDERTAKEN ON FORM DATA:

ELEMENT **block:**

1 - If the user selects the "insert" mode, the mesh name to which the previously defined element belongs is automatically displayed.

2 - When an element is created, the element number in the mesh (NUEL) is automatically calculated.

3 - The number of nodes corresponding to the element type in the NBND field is automatically retrieved from the TYPEL table upon validation of the ELTYP field.

4 - If an element is deleted, the subsequent element numbers in the mesh are updated.

ELNOEUD block:

1 - When an element is created or an existing element type modified, the node internal number in the element is automatically calculated using the number of nodes of the element (NBND). Consequently, the user can and should only enter the exact number of nodes given according to the element type.

ELSET block:

1 - If the set exists: update of SET-ELEMENT relational table in insert or delete mode. If the deleted element is the last in the set, the set will be deleted.

2 - If the set does not exist: set creation and update of SET-ELEMENT relational table.

When calculations to be made on the form data are far too complex or require using mathematical functions not available in the SQL*PLUS language, we have used USER-EXIT functions. Following

a TRIGGER, these functions very simply and quickly run a third-generation language program (FORTRAN in our case). The USER-EXIT accesses the data on the form from where it was initiated, processes them and restores the results on the same form or on the database.

The USER-EXIT system has enabled us to retrieve external data from a sequential file into the base (e.g.: retrieving a mesh).

Calculations are triggered through the SQL*FORMS capability to run operating system commands.

The DIODORE code can handle problems with a wide ranging quantity of data, mesh and calculation results.

It was decided to create several TABLESPACE of different sizes whose parameters are adjusted in order to optimize table management according to the data volume stored in these tables. An ORACLE account is associated with each TABLESPACE. To allow for using EXPORT/IMPORT products for data exchange between different systems and for optimized and simple filing, this account corresponds to a project.

CONCLUSION

In DIODORE, version 2, the data set was designed in the form of a sequential file by means of an editor [2], as is the case with most scientific software packages. The sequential file was analyzed by a "batch" pre-processor, which returned a list of errors to the user [3]. The user had to edit the file again to make any necessary corrections. Version 3, developed with ORACLE, provides the following enhancements of scientific software:

- user-friendly, interactive data entry
- consistency check and data entry help menu levels almost equivalent to an expert system
- portability of the application to almost all systems
- portability of the form-based interactive application to a very wide range of terminals
— reduced number of operations through a lesser risk of submitting calculations with inconsistent data
- data sorting capability and power.

Available as of July 1992, V3.RO will provide a far-reaching context in limit conditions to solve a wide range of problems concerning the mechanics of incompressible, steady and unsteady fluids with a K-e type turbulence model and wall effect.

BIBLIOGRAPHY

[1] J.P. Matheron
"Comprendre Merise"
Editions Eyrolles
Paris /1990/

[2] PRINCIPIA RECHERCHE DEVELOPPEMENT S.A.
"Manuel d'utilisation V2R3"
Sophia Antipolis /1991/

[3] PRINCIPIA RECHERCHE DEVELOPPEMENT S.A.
"Manuel du système V2R3"
Sophia Antipolis /199l/