SIRIO: A DISTRIBUTED INFORMATION SYSTEM OVER A HETEROGENEOUS COMPUTER NETWORK

By: C.R. Costilla, M.J. Bas, J. Villamor [1]
Data and Knowledge Bases Research Group, Technical University of Madrid
e-mail: costilla@dit.upm.es

SUMMARY

This paper presents the SIRIO project, a commercial experience within Spanish industry, developed for Tecnatom S.A. by the Data and Knowledge Bases Research Group at the Technical University of Madrid. SIRIO runs over a heterogeneous local area network, with a *client-server* architecture, using the following tools: Oracle as RDBMS, running over a Unix server, TCP/IP as a communication protocol, Ethernet TOOLKIT for the distributed *client-server* architecture, and C as the host programming language for the distributed applications (every one of them is rather complex and very different from the rest). The system uses computers with MS-DOS, connected to the server over the LAN.

SIRIO is mainly based on the conceptual design of an Rdb, upon which several distributed applications are operational as big software modules. These applications are:
1.- The inspection programs, the management of their corresponding criteria and the automatic generation of queries. 2.- Graphics processing and interface definition. 3.- Interactive Rdb updating. 4.- Historical db management. 5.- Massive load of on-field obtained data. 6.- Report and query application.

The approach of the SIRIO integrated information system presented here is a pioneering one. There are about two dozen companies worldwide in this field and none has developed such an advanced system to this day. From 1992, SIRIO is totally operational in the Tecnatom S.A. industry. It constitutes an important tool to obtain the reports (from different plants) for the clients, for the State control organizations, and for the specialized analyst staff.

Keywords: Rdb (Relational database), design, integrity, consistency, integration, security, flexibility, friendly interface, graphics, local area networks, distributed database applications.

1.- INTRODUCTION.

In general, businesses and public administrations need computer resources to manage their information, in order to maintain their competitiveness. By their own nature and functionality, the technological solution required by these organizations is found, more and more, in the framework of the interconnection of nodes (their institutions) participating in an open system [Tane88] that sometimes is also distributed. The volume of data and their inherent heterogeneity warrant a considerable effort in the integration of techniques and technologies in order to obtain efficient, flexible and open cooperative information systems.

Tecnatom S.A. is a Spanish service-orientated enterprise specializing in non-destructive examinations and personal training. Its main activities are focused on nuclear and electrical plants in general. Tecnatom inspects the steam generators and heat exchangers of the various nuclear plants located throughout Spain. These inspections are expressed as a very large set of indications that specify failures detected in the tubes making up the components mentioned above. These failures are studied to learn about their evolution and, when one of them is serious, the plant is alerted so it can be repaired. The results of every inspection are passed on, in official reports, to the State Nuclear Security Council.

Tecnatom contracted a project with the Data and Knowledge Bases Group at ETSIT (Technical University of Madrid) to manage, in an integrated and flexible way, the information obtained through the inspection of tube bundles that make up a nuclear plant.

We can say that this project has been carried out with as good and durable computer technology as is now

[1] Their research interests are: Distributed databases, Cooperative Information Systems, db design, Distributed knowledge bases and Object-Oriented dbs. In addition to the authors, a total of 10 people, with varying degrees of involvement, have taken part in this project during 20 months. First author was the responsible for SIRIO project and she is the main researcher of this group. For the time being, a new project about "Distributed Knowledge Communication" is going on funded by the PRONTIC Spanish National Research Programme.

available in this technological field. We should stress the pioneering character of this integrated information system for the inspection of tube bundles in nuclear plants. There are about two dozen companies in this field worldwide, and none of them has developed such an advanced information system to this day. From 1992, SIRIO is totally operational in this industry. It constitutes an important tool to obtain the reports (from different plants) for the clients and State control organizations. The distributed information system here presented, now in industrial use, is obtaining highly satisfactory results.

2.- SCOPE OF THE SIRIO PROJECT.

The inspections that TECNATOM carries out in nuclear, coal and petrochemical plants are of great importance for their own security and that of the nearby population. An error or omission, when detecting a fault, can result in serious accidents. For this reason, using a reliable information system, as advanced as possible, will result in enhanced security.

Previous to SIRIO, Tecnatom had an automated information system based on files, with the following problems: lack of integration among programmers, little flexibility when making reports and difficulty in maintaining data integrity. Moreover, the system was single-user with a large amount of data, and information had to be scattered among many diskettes with the problems related to it: redundancy and inconsistency of information. This lack of cohesion in the old system required great efforts to get the necessary reports on the state of the components from different plants. And the reports must always be consistent and precise.

The above mentioned problems justified the building of a new information system, using state-of-the-art technology (computer networks, distributed systems and distributed relational technology) leading to the SIRIO project's complexity.

3.- SIRIO'S TECHNOLOGY: Communication Network, Hardware and Software Tools.

The hardware and software requirements for the SIRIO project are summed up in the following points:

*** Hardware requirements:**

1.- A data server, running under UNIX, and several PCs (under MS-DOS) as clients, connected through a LAN.

2.- Intelligent communication boards in each computer, and their corresponding communication processors in order to improve throughput.

3.- Possibility of offering a graphical interface.

4.- Support for using a plotter.

5.- Flexibility for future extensions.

6.- Possibility of supporting a scanner and optical disks.

*** Software requirements:**

1. Ability of running over a computer network, letting a set of users access the information concurrently from node's applications with a relatively good performance.

2.- The use of a communications TOOLKIT, sold by the RDBMS vendor (Oracle). For the given configuration, the optimal TOOLKIT makes use of TCP/IP protocol.

3.- Capacity of managing databases with a very large volume of heterogeneous data (graphics, texts and maps).

4.- Concurrency control and failure recovery.

5.- Security: setting up access keys and privileges (user levels). System administrator.

6.- User friendly interfaces: graphics, maps and menus.

7.- Capacity of producing customized reports.

8.- Ability to schedule jobs in the absence of an operator.

9.- Integrity of information, using tools to maintain it (key and cross reference integrity, field validation conditions).

10.- Graphical tools to display component maps on a screen.

11.- Functions for operation tracing (System auditing).

For the development of system applications the following tools were selected:

- SQL*Plus: Oracle SQL command interpreter.
- SQL*Forms: Oracle application generator through the definition of data entry screens.
- Pro*C: Oracle SQL pre-compiler for accessing the database through SQL embedded in C code.
- Microsoft C compiler version 5.1.
- Graphics and Utilities library from Essential Software.

Following these requirements, SIRIO has been implemented on a mixed environment Ethernet network, with a client-server architecture. It has a main node operating as a database server, running an Oracle

RDBMS server under Unix. Several clients (PCs) run the database applications under MS-DOS, and are connected to the server with the TCP/IP communication protocol (see Fig.1).

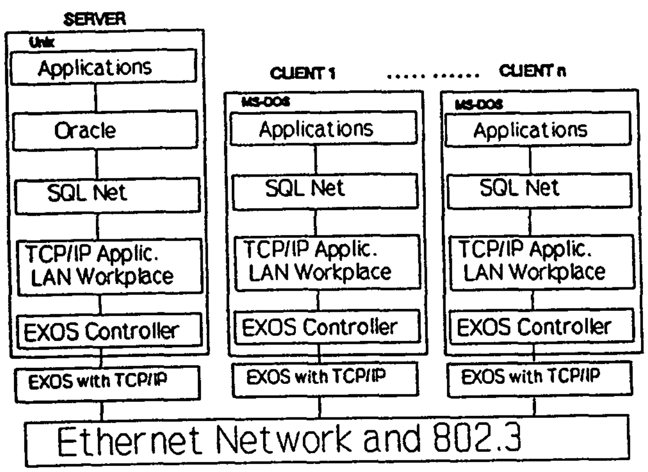

Fig 1: *SIRIO, communication system between server and clients.*

4.- DATABASE CONCEPTUAL SCHEMA

As we know, a database design is of critical importance for an information system. The successful completion of this step will determine the capacity to manage easily the information of a part of the real world, and it will also determine the flexibility when making queries and the simplicity when modifying (or extending) the schema. To fulfil this design task in SIRIO, the next steps were followed:

a).- *Design of SIRIO conceptual schema.* As a starting point, the first design level was completed using a hierarchical semantic model based on SHM [SS77] and ACM/PCM [BrRi84]. Later, we mapped it to the E-R model [Chen76] and; finally, its conversion to a relational model was straightforward. A general part of the database design is organized according to a common information shared by all plants (common area tables), and then, every plant has its own particular design (concise area plant tables). In this way, even though the total conceptual schema is stored now only in a server, in the future, if necessary, conversion to a distributed database with a global conceptual schema (among several servers) will be straightforward [OzVa91]. For space reasons, we omit more explanation about it, and we summarize below the more relevant modeled concepts.

b).- *Conversion to the relational model.* The conversion mechanism can be summed up in the following points: **entity sets** transform themselves into **relations**; M:N **associations** into **relations** that include the keys of the associating entities; and finally, 1:N and 1:1 **associations** are implemented through foreign keys. To give an idea of the size of this design, 58 relations with 406 attributes were defined.

c).- *Relation description and semantic data control.* Every relation is described with its primary key and, sometimes, for practical reasons, some functional and multivalued dependencies (FD and MVD) are explicitly included. Most of the schema relations are in 3NF or BCNF.

d).- *Relational normalization.* When a careful design of a database schema is carried out, the aim is to embed FDs in the primary keys of all the relations according to relational database theory [Maie83]. But the problem is not so simple, because the nature of the main applications plays an important role in the design. If queries are more frequent than updates in a database, it is advisable not to fragment relations too much, even though we have to maintain a large set of integrity constraints. When updating predominate, a higher fragmentation is advisable in exchange for a reduction in the number of integrity rules [Ullm82].

e).- *Integrity rules definition.* The Oracle RDBMS (version 6.0 and lower) only supports the entities' integrity rule as part of the database schema. So, applications have to enforce the rest of integrity rules themselves. For this reason we defined *triggers*, processes that are executed when a condition arises during the execution of a program. The semantic data control in SIRIO consists of a total of 120 intra-relational and 250 inter-relational integrity rules, most of them implemented through fairly large and complex triggers.

In Fig. 2 the final relational schema is represented without much depth. Each square represents a relation and the links are relations associated through foreign keys (when the edge is directed, the association is 1:N).

The security mechanisms (privileges and user passwords) are of great importance in order to avoid the possibility of a person updating the tables through tools or applications (as the Oracle-provided SQL) that do not maintain integrity rules.

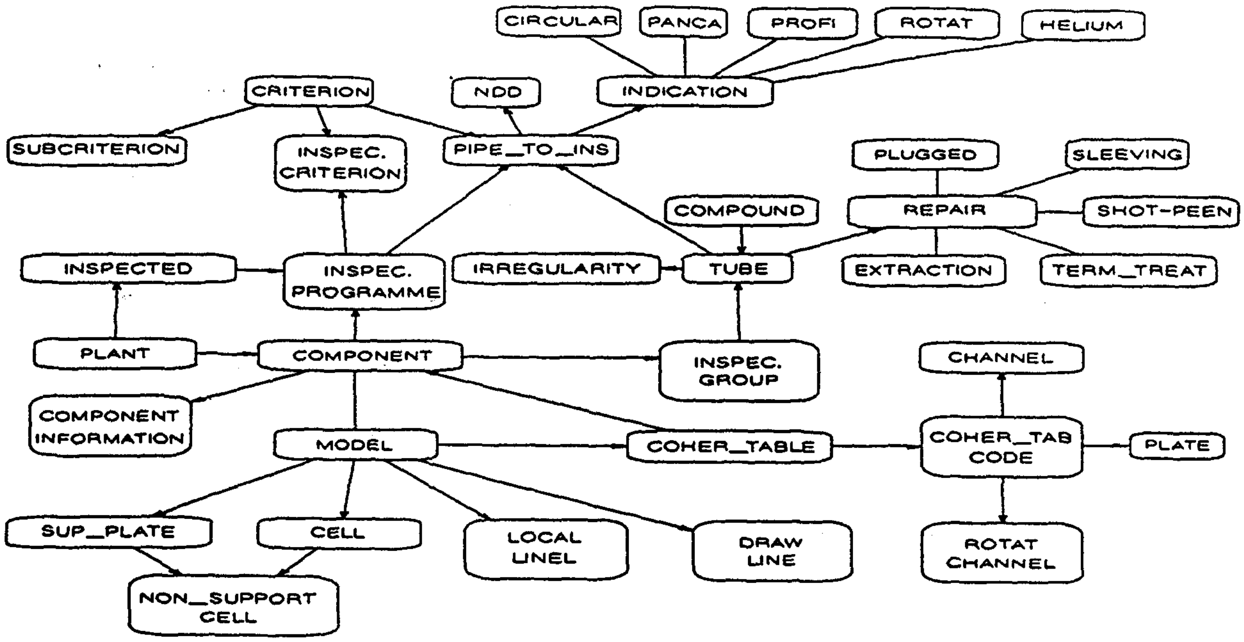

Fig 2: *Overview of SIRIO Conceptual Schema*

5.- HISTORICAL DBs MANAGEMENT.

Historical data in SIRIO results from indications when inspecting the components of a nuclear plant, and it is of primary importance to manage their evolution through time. An indication is inserted into the database when an inspection detects a defect in a given geometric position of a component. This defect will be checked in later inspections, and if the value measured differs considerably (more than a given range) this new measurement will constitute a new version of the indication, with a "timestamp" identifying the current version. We use the pair "month/year" corresponding to the moment when the inspection is carried out. The set of versions of the indication will constitute its history. Inspections take place approximately once a year.

Among many versioning proposal approaches [AhSn88] [AdQu86] [DLW84], we considered two possible solutions for SIRIO to keep historical data: store only the changes from a given version that is chosen as a reference (this option takes up minimum space), or keep all the data of every version (it simplifies the historical data look up mechanism).

From the brief preceding discussion we can infer the existence of two different areas for keeping the history of an object. One area, called **actual area**, where the complete information of the object (usually corresponding to the last version) is kept. Another area, **historical area**, is where the historical information related to the object is stored.

The solution adopted in SIRIO was to break up the information contained in *defect indication* in two parts, *global_ind* and *historical_ind*. The first one keeps the last version of a defect, so it contains the most up to date state of each component, while the second one contains the previous versions of a defect, so we can learn about the evolution of the defect through time.

6.- SIRIO'S DISTRIBUTED APPLICATIONS.

Figure 3 represents the set of applications that make up the SIRIO system. Each one can be located at any node of the interconnecting network. They are concurrent with a good degree of parallelism on the server. Their relationships are presented in Fig.4, and we briefly describe them below.

6.1.- Interactive updating. Its objective is the on line updating of data present in databases, guaranteeing their integrity and consistency. For this application, the potential user is the administrator of the system. It has been developed using Oracle SQL*Forms.

6.2.- Historical data management application. It allows

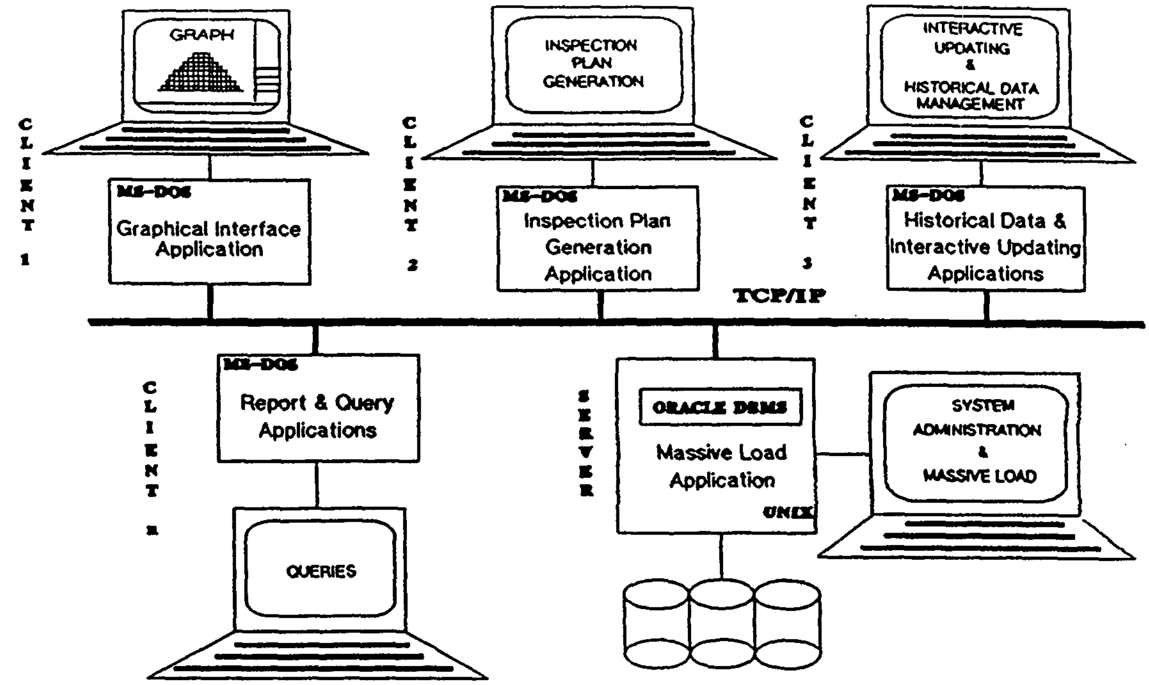

Fig 3: *SIRIO's Distributed Applications*

changes in registers belonging to tables that contain historical data and, automatically, the system spreads changes through the history of the modified object. The historical evolution is used by the specialized analyst staff, in a consultative way, when preparing an inspection of a nuclear plant (see 6.4).

6.3.- Graphical interface application. It provides the following services:

* Generation and storage of the field and screen coordinates that define a component's structure, and definition of the lines that make up a drawing on screen. The generated coordinates can be cartesian or polar.
* Retrieval of a component's graphical information for on-screen display.
* Graphical mode display of the ground (cell map) and elevation plan of a component; the tracking mouse clicks any place on the map, and defines, as the user wants, different examination groups on each map.
* Definition, using the mouse, of location lines and labels on maps drawn on the screen, and their insertion in the database.
* Marking of a group of cells (square or circular) and insertion of the group in the db with different identifiers.
* Colored representation of damaged tubes. Each type of fault stored in the server database is represented on the maps of this application by a different color.
* Zoom and unzoom (nested) of any part of the cell map.

6.4.- Inspection plan generation application. The goal of this application is the automatic selection of the tubes that must be inspected in each component of a plant. This selection is done according to prearranged and newly created criteria, defined following the indication history of

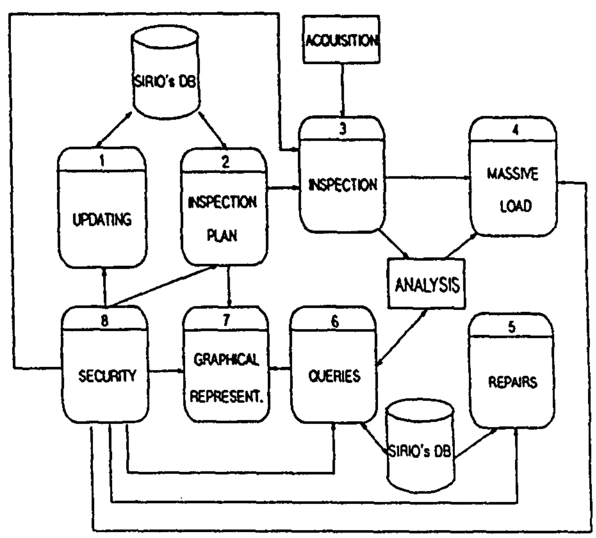

Fig 4: *Inter-relationships of SIRIO Distributed Applications.*

each component, using the graphical services provided by the already mentioned graphics module.

6.5.- Massive load application. It allows the validation (using a coherent-table) and the insertion in the db of data from inspections carried out in each nuclear plant. The name "massive load" is due to the fact that insertions are processed in packages with hundreds of thousands of tuples, without operator intervention. It is coded in C, with embedded SQL sentences. It is the only application running in the server, for maximum performance.

6.6.- Report and query application. It provides a friendly interface for definition and execution of queries over some tables of the database. The data output can be in map form (using graphics functions), in table form or as statistics, with the user deciding whether to use a screen, plotter or printer.

The applications from 6.2 to 6.4 and 6.6 are coded in C, with embedded SQL statements, using the Utilities library from Essential.

7.- CONCLUSIONS.

This paper presented the SIRIO project, which has been commercialized by the Spanish industry. The system is based on a distributed information system running over a local area network. This project has been designed and implemented for Tecnatom enterprise at Technical University of Madrid.

The hardware and software tools used (KDBMS, Unix, C, Ethernet, MS-DOS) and the techniques applied to the implementation of SIRIO (described in this paper), guarantee two important aspects: its portability to other platforms, and its flexibility for future expansions in a wider rang of networks and/or computer nodes.

The SIRIO information system integrates the design of a rather complex relational database and a set of distributed applications: historical data management, graphics and maps, inspection plan generation, "massive" load, friendly interface for querying and report generation.

We would like to emphasize the advanced technology used throughout the project, and the pioneering character of this information system for the inspection of tube bundles in nuclear plants. There are about two dozen companies in this field worldwide, and none has developed such an advanced system up to now. For these reasons, we consider this work an experience *de facto* of certain relevance, both in the context of technologically advanced information systems and in the framework of interconnection techniques for open systems.

The SIRIO system is obtaining highly satisfactory results in Tecnatom S.A. Spanish industry, the present owner of this commercialized information system.

We are grateful to Prof. A. Segev for his suggestions that improved the quality of the paper.

8.- BIBLIOGRAPHY.

[AdQu86] Adiba, M. and Quang, N.B., "Historical Multi-Media Databases", Proc 12th International Conference on Very Large Data Bases, Kyoto, 1986, pp. 63-70.

[AhSn88] Ahn, I., and Snodgrass, R., "Partitioned Storage for Temporal Databases", Information Systems Vol.13 No.4, 1988, pp.369-391.

[BrRi84] Brodie, M.L., and Ridjanovic, D., "On the design and Specification of Database Transactions", in "On Conceptual Modelling", Brodie,M.L., Mylopoulos, J., and Schmidt, J.W. (eds.), Springer-Verlag, 1984, pp.277-306.

[Chen76] Chen, P.P., "The entity-Relationship Model: toward a unified view of data", ACM TODS, Vol.1 No.1, 1976, pp. 9-36.

[DLW84] Dadam, P., Lum. V., and Werner, H-D., "Integrating of Time Versions into a Relational Database System", Proc. 10th VLDB, Singapore, 1984, pp. 509-522.

[KeRi88] Kernighan, B.W. and Ritchie D.M., "The C Programming Language", 2nd. ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[Maie83] Maier, D., "The theory of relational databases", Computer Science Press, 1983.

[ÖzVa91] Özsu, M.T. y Valduriez, P., "Principles of Distributed Database Systems", Prentice-Hall International, New Jersey, 1991.

[SS77] Smith, J.M. and Smith, D.C.P., "Database Abstractions: aggregation and generalization", ACM Transactions on Database Systems, Vol.2. N.2, 1977, pp.105-133.

[Tane88] Tanenbaum, A.S., "Computer Networks", 2nd. ed., Prentice Hall, 1988.

[Ullm82] Ullman, J.D., "Principles of Database Systems", 2nd. ed., Pitman, 1982.