ABOUT A METHODOLOGY TO SELECT A DBMS

M.F. ATAN

M. E. BRAGADO BRETANA

Institute Central de Investigacion Digital
Calle 198 1703
Siboney, C. de la Habana, Cuba

1. Introduction.

The rising use of database systems for the data management has resulted in an increasing number of systems entering the marketplace. The selection of a database system requires a structured, comprehensive investigation.

The following paper show a general methodology to select a DBMS in order to use it and also to select it as a pattern to be implemented in some hardware configuration.

2. Aims.

A complete evaluation methodology for database systems must integrate a feature analysis phase, human factors aspects /CNORT83/, and a performance analysis phase.

The objective of this evaluation is not only to choose a system for an application, but to take a DBMS as a pattern to be implemented. Because of this, several features of the systems do not have great importance; for example, the Operating System on which the system executes, arithmetic precision, error recovery, etc.; these are implementation characteristics that can be adapted to each necessities. Therefore, it complements the methodology with a phase about the characteristics of implementation of each system.

The main objetive in our methodology is to remark the integrity of these phases. Any phase itself can not be used in isolated form to determine which system must be selected. Each phase must be analyzed in complementary form to obtain a successful selection.

The figure 2.1 shows a summary of our methodology.

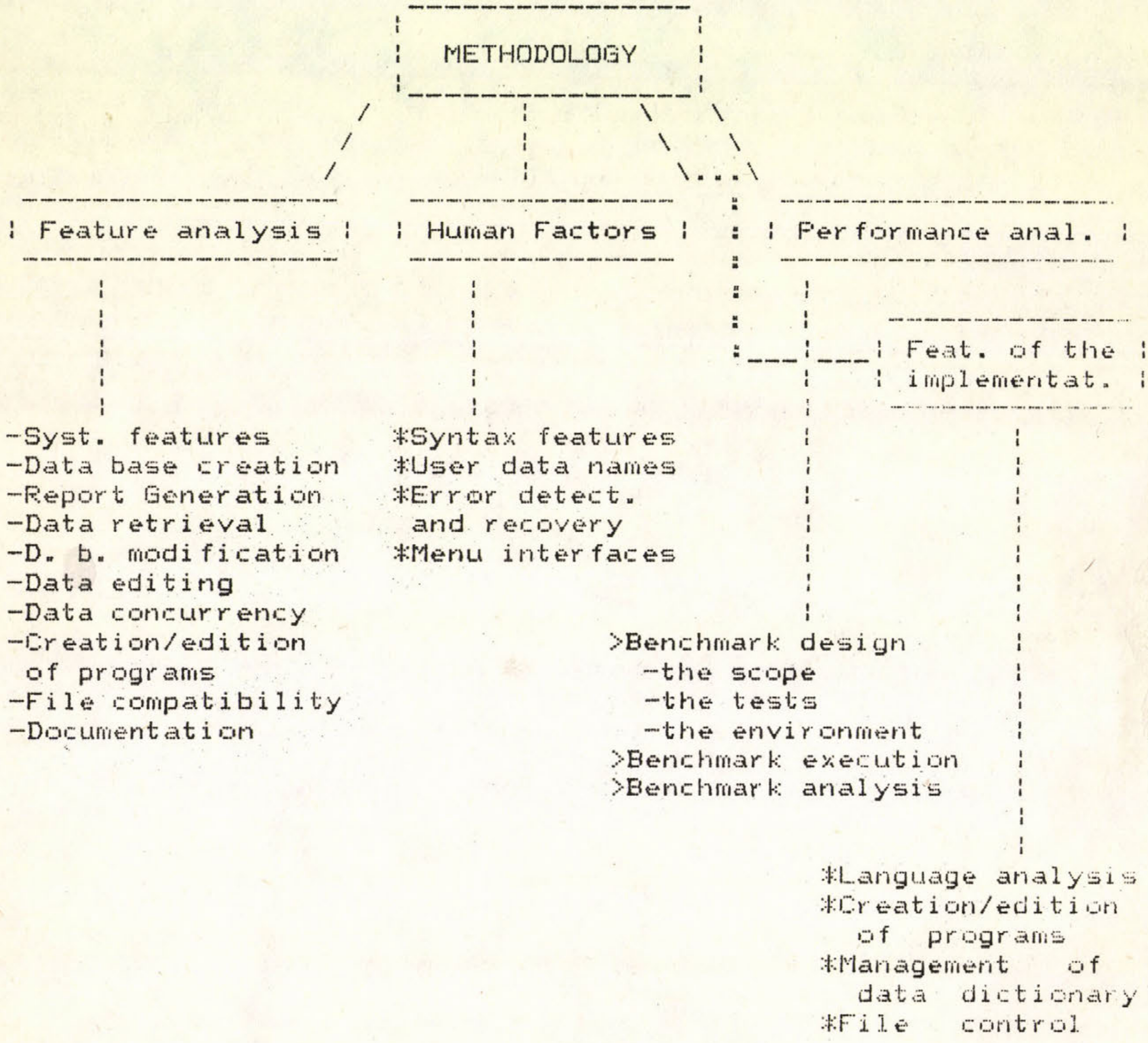

Figure 2.1. A summary of our methodology.

2.1. Feature analysis phase.

The features and capabilities that a database system may provide is very large. A feature analysis fulfill two functions; it first serves as a process to decide those systems that are completely unsuitable for answering the needs of a particular application and second, it provides a ranking of the surviving candidate systems.

Feature analysis has a number of advantages over other methods of system evaluation.

i) Featur analysis provides a structured first cut. The final result of a feature analysis should be a small number of andidate systems. Performance analysis, which is much more costly, can then be performed with only this small number of systems.

ii)There are qualitative aspects of a database system that cannot be quantified in terms of system performance; for example: vendor support, documentation quality, security, user friendliness, etc. Since benchmark analysis cannot directly test the performance of these features, feature analysis remains the best method for their analysis.

iii) Little or no system costs are involved in performing a feature analysis because a database implementation is not required.

In spite of these advantages features analysis should not be used in isolation to evaluate and select database systems. There are several reasons for this.

i) The feature importance coefficients and the system support ratings are given values by a knowledgeable design expert. However, no two experts may come up with same values given the same application environment, because the feature analysis is a subjective exercise.

ii) Feature analysis is a paper exercise that cannot truly evaluate how a system will perform in an organization's application environment.

The following are the points included in the feature analysis for each system.

1) System characteristics. It is included the general features of the system: the type of DBMS (relational, CODASYL, etc), Operating System, main and auxiliary memory necessary, type of file organisation, characteristics of the arithmetic operators, etc.

2) Data base creation. It is described the means for the data base creation.

3) Report generation. It is shown the commands or means to obtain a report of a file.

4) Data retrieval. It is included the analysis of tools for data retrival, the commands which allow to know the structure of a file, etc.

5) Data base modification. In any application it is necessary to modificate the database. Here, it is described the several means which allow to asimilate the modifications and changes in a database.

6) Data editing. To study the tools to data edition: edition and modification fields of a file.

7) Data concurrency. To analyze the several form of concurrency control included in the DBMS evaluated.

8) Creation and edition of programs.

9) File compatibility. The compatibility of files between different systems is a good characteristic which allows to transfer files and to make applications in connection with different systems.

10) Documentation. It is described the quality and characteristics of all the documentation of the system.

2.2 Human factors aspects.

Relational technology was provided to a new class of users through simplified terminology and a relational algebraic command language. These new users knew their application areas well, but their main tasks were nonprogramming tasks. To the correct evaluation of a DBMS it is necessary to make a heavy analysis of the human factors or psychologic aspects in order to accept a system. A few authors include these factors in the general evaluation of the systems.

The following phase highlights the human factors aspects, the benefits, and the limitations of each system evaluated.

i) Syntax features of the commands.

In this point it is evaluated the syntax features of the names of the commands: friendly language, relation between the name of the command and the corresponding data base operation, and the full command names and keywords without abbreviations.

The use of a language close to the natural is very important to the assimilation and learning of a system. The user does not feel the difference between the way usually he thinks and the way he works with the computer. This is important to decrease the debbuging time of applications. The use of "noise" words helps to improve the readability of a command. Also, these characteristics improve the self documentation of programs.

ii) User data names.

Here, it is analyzed the possibilities that the system provides in order to express the names of the user data in legible form. Also, this aspect has influence in the keyboard errors. From some observations, users desire conciseness, but this is overshadowed by the need to express and document ideas in meaningful phrases. Users frequently try to condense abbreviations or use meaningless names such as X or ABC that make errors typing them or cannot remember the precise names that were used. A good system must allow that syntax of data names be legible.

iii) Error detection and recovery.

The time lost when errors are not handled properly for the user indicate the importance of good error handling. Here, it is evaluated if the systems have a good error detection, recovery, and informative messages.

iv) Menu interfaces.

In the same form that increases the interactive way of work with the computer, it increases the use of menu interfaces between the man and the computer. The systems driven by menu are very easy to use. With the combination of menus defined with meaningful English phrases and availability of "help" messages, users have not much trouble, becoming effective users. Users do not have to learn or remember commands; they simply make choices from a menu. This is an important human aspect for the easy assimilation of a system, specially for the non-specialized user.

v) Learning.

It means how long it takes the user to learn how to work with a system. This is a very important human factor to accept a system by the user. A very efficient system but with difficulty when it shows the form of use, will be difficult to be accepted for the common user.

This aspect must be fulfill with some kind of statistical investigation between several user with no equal levels of technical experiences.

2.3. Performance analysis phase.

The major methods of performance evaluation are Analytic modelling, Simulation modelling, and Benchmarking.

Analytic modelling represents a system by defining equations that relate performance quantities to known system parameters. The use of these equations allows a fast and accurate means to evaluate system performance. The principal disadvantages are that the equations are inadequate to model the complete range of functionality found in a data base system and also they fail to account for the dynamic behavior of the data base system. For these reasons analytic modelling has failed to receive wide acceptance as a tool for modelling data base systems.

Simulation is the process of developing a computer program to approximate the behavior of a system over a period of time. Simulation modelling has been applied to data base systems /HULTE77/, /NAKAM75/. The major concern with using simulation is the time and expense that are often necessary to develop a simulation model. Stochastic simulation models also produce only estimates of a model's true performance and the large volume of results returned by a simulation often creates a tendency to place more confidence in the results than may actually be warranted.

Benchmarking is used when a few data base systems are to be evaluated and compared. Benchmarking requires that the systems be implemented so that experiments can be run under similar system environments. Benchmarks are costly and time-consuming but provide the most valid performance results upon which data base systems can be evaluated. While both simulation and analytic modelling are limited in the scope of their system testing, benchmarking offers the chance to evaluate the actual data base system /GOFF73/.

The benchmark experiments publicated concentrate on the comparison of candidate commercial systems for a particular application /GLESE81/, /ASTRA80/, /KEENA81/, /TEMPL/, etc.

While benchmarking can be a useful and important technique for data base system evaluation; designing, setting up, and running a benchmark is a difficult and time-consuming task. Benchmarking is problematic and at worst, a gross distortion of reality but it is possible to obtain good conclusions if these aspects are known and if specific features are analyzed.

In order to aid in the development and analysis of benchmarks it is essential to show the methodology used. No one methodology has provided the necessary robustness demanded from a generalized methodology. No benchmark methodology can expect to incorporate every aspect of every benchmark.

Our methodology has been divided into 3 principal parts: benchmark design, benchmark execution and benchmark analysis.

2.3.1. Benchmark design.

The design of a benchmark involves: a) the scope of the tests, b) the tests to be performed, and c) the environment of the data base system to be tested.

As it was shown above, the success of benchmarks depends on the objectives be exactly detailed. It has been proved that general benchmarks distort the results and mask the deficiencies /HOUST84/. In our case, it uses the benchmark to complement the other phases that are included in our evaluation. Besides, the current systems allow to perform several classic processes of DBMS in interactive way as: creation and modification of data bases, edition of programs and data bases, report generation, etc., which are not possible to apply to any classic benchmark test. These features are included into another phase of our general evaluation.

2.3.2. Benchmarks execution.

When the experiment has been formally defined, the next step is to implement the design for each of the candidate systems.

2.3.3. Benchmark analysis.

The final phase of benchmarking is the analysis of results. Evaluation of the data generated during benchmarking must begin before the tests have been completed. It provides feedback during the testing by suggesting which types of experiments need to be repeated in more detail, or should be extended in some way. Summarizing the meaningful information from these results and discussing them in a report form is a key step in the benchmark testing.

2.4 Characteristic of the implementations.

As it has shown above, the objective of our methodology is to select a system to implement it in some configuration. Therefore, it is necessary that our methodology contains a phase about the difficulty to implement one or another DBMS.

In this phase the following modules will be analyzed: language, creation and edition of programs, management of data dictionary, and file control system. Each modules must be analyzed from the point of view of the difficulties of implementation.

i) Language.

The characteristics of implementation of the languages must be analyzed. It is necessary to analyze the structure of the language, characteristic of the syntax and semantic analysis, type of compiler (interpreter, compiler, etc), language ambiguity, inter-relation between the language and other modules, etc.

ii) Creation and edition of programs.

The creation and edition of programs are the means included in the DBMS to develop programs. It is necessary to analyze:

-level of full-screen editing of command files

-special features

-relation with other modules of the system

-if it has included some function of syntax analysis it is important to evaluate the level of relation with other modules of the system.

iii) Management of data dictionary.

This module includes all the software means necessary to control the operations with data dictionary. Here, it is included the analysis of the following aspects:

-structure of the data dictionary

-the means to create/maintenance of data dictionary

-the software to handle the dictionary

-level of complexity of the data dictionary

iv) File control system.

The difficulty of programming the file organization and its commands are shown in this section. The analysis must include the following:

-features of the organization used (indexed, sequential, etc)

-characteristics of the commands that have relation with the files

-characteristics of the data protection.

3. Conclusions.

It is shown an integrated methodology to select a DBMS in order to take it as a pattern to be implemented. This methodology must be used as a whole and complemented with particular analysis. The conclusion of the evaluation process must be shown with several summary tables which explain the result in each phase. This methodology is used in our Institute with succesful results.

4. Bibliography.

/ASTRA80/ Astrahan, M. et al. Performance of the System R Access Path Selection Mechanism. Proc. IFIP, 1980

/BENWE75/ Benwell, N. Benchmarking. Toronto, J. Wiley and Sons, 1975.

/BING85/ Bing, Yao S.,and A. Hevner. Performance Evaluation of Database Systems: a Benchmark Methodology. c N.J. USA Datapro Research Corp.,AS80-100-101,Aug.1985.

/BITTO83/ Bitton, H. et al. Benchmarking Database Systems: A Systematic Approach. Computer Sciences Department, University of Wisconsin, Technical Report #526.

/BOAR84/ Boar, B. H. Ten Criteria For Selecting Mature DBMSs. c Auerbach Publishers Inc, Data Base Management, 1984.

/BOGDA83/ Bogdanowicz, R. et al. Experiments in Benchmarking Relational Database Machines. Munich, Proc. of the third International Workshop on Database Machines, Sept. 1983.

/BOIES74/ Boies, S.J. User Behavior on an Interactive System. IBM System Journal 13,1-18 1974.

/BOND84/ Bond, G. A Database Catalog. BYTE, Oct. 1984.

/BORAL84/ Boral, H. and D. DeWitt. A Methodology for Database System Performance Evaluation. Univ. of Wisconsin, Computer Sciences Department, Technical Report #532, January 1984.

/BOYLE84/ Boyle, B. Software Performance Evaluation. BYTE, 9(2): Feb. 1984.

/CARRO84/ Carrol, J.M., M.B. Rosson Beyond MIPS. Performance Is Not Quality. BYTE, 9(2): Feb. 1984.

/CNORT83/ c North-Holland Pub. Company. Human Factors Aspects of a Modern Data Base System. Information and Management, 6(1): Feb. 1983.

/CURNO76/ Curnow, H. J., B. A. Wichman A Synthetic Benchmark. Computer Journal, 19(1): Feb. 1976.

/DAVIE81/ Davies, D.J.M. Benchmarking in Selection of Timesharing Systems. Proceedings of the 14th. meeting of the CPEUG, Nov. 1981.

/DEARN78/ Dearnley, P. Monitoring Database System Performance.

The Computer Journal, 21(1): 1978.

/DEMET84/ Demetrovics, J., et al Some Remarks on Statical Data Processing. Hungary, MTA SZTAKI Kozlemenyek, 30: 37-51 1984.

/FERRE78/ Ferreri D. Computers Systems Performance Evaluation. Prentice-Hall, Inc., 1978.

/GILBR81/ Gilbreath, J. A High-Level Language Benchmark. BYTE, 180-198 Sept. 1981.

/GLESE81/ Gleser, M. et al. Benchmarking for the Best. Datamation, May 1981.

/GOFF73/ Goff, N.S. The case for Benchmarking. Computer and Automation, May 1973.

/HOUST84/ Houston, J. Don't Bench me in. BYTE, 9(2): Feb. 1984.

/HULTE77/ Hulten, C. and L. Soderlund. A simulation Model for Performance Analysis of Large Shared Data Bases. Proc. Third VLDB Conf., Tokio, 1977.

/KEENA81/ Keenan, M. A Comparative Performance Evaluation of Database Management Systems. Berkeley, EECS Dept., University of California, 1981.

/MARVI84/ Marvit, P., M. Nair. Benchmark Confessions. BYTE, 9(2): Feb. 1984

/NAKAM75/ Nakamura, F., et al. A Simulation for Data Base System Performance Evaluation. Procc. NCC, 1975.

/REISN81/ Reisner, P. Human Factors Studies of Database Query Languages: a Survey and Assesment. ACM Computing Surveys, 13(1):

13-32, 1981.

/ROBER84/ Roberts, B. Benchmarks and Performance Evaluation. BYTE, 9(2): Feb. 1984.

/RODRI75/ Rodriguez-Rosell, J. and D. Hilderbrand. A Framework for Evaluation of Data Base Systems. Research Report RJ 1587, IBM, San Jose, 1975.

/SIBLE84/ Sibley, E. H. DBMS Evaluation and Selection. c Auerbach Publishers Inc, Data Base Management,22-04-01.

/SPOON/ Spooner, C.R. Benchmarking Interactive Systems: Modeling the Application. Proceedings of the 15th. Meeting of the Computer Performance Evaluation Users Group (CPEUG), pp.53-63.

/SU81a/ Su, S. et al. A DMS Cost/Benefit Decision Model: Cost and Preference Parameters. National Bureau of Standards,Report NBS- GCR-82-373,1981.

/SU81b/ Su, S. et al. A DMS Cost/Benefit Decision Model: Analysis, Comparison, and Selection of DBMSs. National Bureau of Standards,Report NBS-GCR-82-375,1981.

/TEMPL/ Templeton, M. et al. Evaluation of 10 Data Management Systems. SDC document TM-7817/000/00.

/WALTE76/ Walters, R.E. Benchmark Techniques: A Constructive Approach. The Computer Journal, 19(1): Feb. 1976.

/WEISS81b/ Weiss, H. Which DBMS is Right for You. Mini-Micro Systems, October 1981.

Одна методолония для выборки DBMS

М. Фонфрия Атан, М.Е. Брагадо Бретана

Резюме

Уже существует много различных пакетов для разработки базы данных. Для того чтобы выбрать один из них, надо применить очень много тщательных и структурированных методов. В статье показана общая методология такой выборки.

A DBMS KIVÁLASZTÁSÁNAK EGY MÓDSZERTANA

M. Fonfria Atan, M.E. Bragadó Bretana

Összefoglaló

A piacon már rengeteg különböző adatbázis-kezelő programcsomag létezik. Egynek a kiválasztásához igen sok, alapos és részletes vizsgálat szükséges. A cikk a kiválasztásnak egy lehetséges metodológiáját irja le.