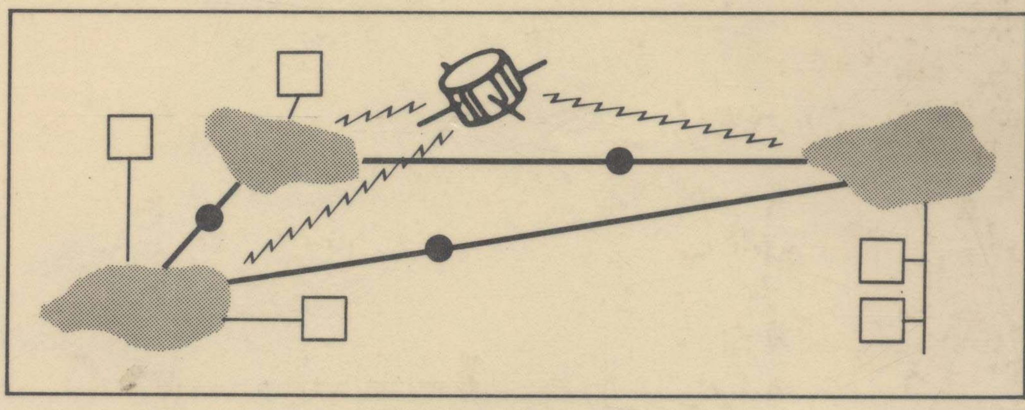
Network Database Systems

Philippe G. Lehot
Stephen H. Kaisler

3rd Edition



Published by:

ARCHITECTURE
TECHNOLOGY
CORPORATION
SPECIALISTS IN COMPUTER ARCHITECTURE

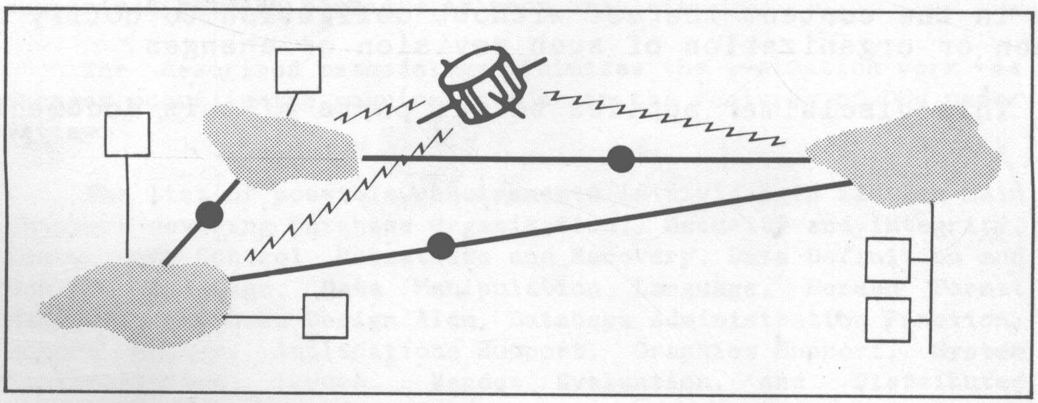
P.O. Box 24344, Minneapolis, MN 55424 (612) 935-2035

Network Database Systems

Philippe G. Lehot
Stephen H. Kaisler

3rd Edition

January 1986



Published by:

ARCHITECTURE
TECHNOLOGY
CORPORATION
SPECIALISTS IN COMPUTER ARCHITECTURE

P.O. Box 24344, Minneapolis, MN 55424 (612) 935-2035

DISCLAIMER

Architecture Technology Corporation makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose.

Further, reasonable care has been taken to insure the accuracy of the report, but errors and omissions could have occurred. Architecture Technology assumes no responsibility for any incidental or consequential damages caused thereby.

Further, Architecture Technology Corporation reserves the right to revise this document and to make changes from time to time in the content thereof without obligation to notify any person or organization of such revision or changes.

This disclaimer applies to all parts of this document.

FOREWORD

The analysis and selection of Database Systems (DSs) in a distributed environment is a corporate decision that competes in importance with the selections of hardware and operating systems. As users become more knowledgeable, this DS selection will even govern the hardware and operating system choices.

This report covers a comprehensive list of possible requirements for the DS evaluation and selection task, and offere an attractive methodology to accomplish the task.

A flexible approach provides the prospective DS user with the framework to make a choice among generalized DSs which are available off the shelf, even when requirements are changing.

Since DSs evolve as new features are constantly added to existing packages, any proper evaluation of one package or any comparison between packages is soon obsolete.

The described methodology minimizes the evaluation work as changes occur in the requirements or in the features of DSs under review.

The list of possible requirements is divided in fifteen main chapters covering Database Organization, Security and Integrity, Concurrency Control, Robustness and Recovery, Data Definition and Control Language, Data Manipulation Language, Screen Format Manager, Database Design Aids, Database Administration Function, Report Writer, Applications Support, Graphics Support, System Implementation Issues, Vendor Evaluation, and Distributed Processing Issues. Chapters are composed of sections, themselves divided into subsections, thus forming a tree structure.

By assigning a weight to each node in this tree structure, the user quickly defines his requirements specifications.

Examining each candidate DS, the user may then allocate a "grade" for each requirement. The overall "figure-of-merit" for a candidate DS - with respect to the specified requirements - is the weighted sum of the grades.

If the user's requirements change after the beginning of the evaluation task, the evaluation work is not invalidated by these changes. Rather, the overall "figure-of-merit" of each candidate DS may be recalculated automatically by a small computer program.

Making use of this property, the methodology extends to the study of the best DS choices as a function of different alternatives for the requirements.

TABLE OF CONTENTS

Ø. INTRODUCTION

Ø.Ø Generalities

This document illustrates a methodology for the comprehensive evaluation of generalized Database Systems (DSs). Our intention is to provide a flexible approach that will help the knowledgeable prospective DS user

(1) Establish his requirements in a rapid, flexible and refined manner,

(2) Make a choice among the commercially available generalized DSs, and make that choice a true function of the expressed expressed requirements.

Many Database Systems (DSs) are commercially available, often running on more than one type of processor under different operating systems.

More DSs become available with each succeeding month, although the differences between them are often very subtle. Choosing among the DSs that run on a given processor can be a lengthy, difficult and aggravating process. The advent of inexpensive, powerful microprocessors has compounded the problem by stimulating the development of local area networks. In turn, this has stimulated the development of distributed databases across two or more of the network nodes.

To assist prospective users in selecting a database system, we have developed a comprehensive list of evaluation criteria (CLEC) as a guide to evaluating competing DSs.

The CLEC is first divided into main "categories" or classes of features that reflect the main "viewpoints" from which DSs are evaluated.

The CLEC is designed to cover a wide range of features and implementations. Thus, many of the classes or tcategories" of features defined in the CLEC overlap, because features often contribute to several areas of performance. In fact, due to tradeoffs, a given feature may have a desirable effect from one "category" viewpoint, and simultaneously an undesirable effect from a different viewpoint.

Accompanying the CLEC is a methodology for assigning "weights" to individual features, and aggregating "grades" assigned in judgement of a candidate DS to derive a "figure-of-merit" for each DS under evaluation.

Computing the figure-of-merit is greatly simplified by arranging the CLEC in a tree structure. Users may examine the impact of different database system requirements by modifying the weights assigned to leaves in the tree.

Since database systems evolve as new features are added to existing packages, any evaluation of one package or any comparison between packages soon becomes obsolete. However, our evaluation methodology permits a quick update with a minimum amount of impact on the work already done, namely the requirements specifications and the evaluation work.

Hence, we believe the CLEC serves equally well the prospective DS user or the DS user who is enhancing his data management capabilities.

Ø.1 Why a database system?

This monograph discusses the selection of database systems for a distributed data processing environment (either locally or geographically distributed). It proceeds under the assumption that the user has already decided that a DS is the proper tool for managing his data. Such a decision should be made after proper reflection upon the objectives for data management. Sometimes, a DS is not required where a file management system would be quite adequate for the job. With this thought in mind, we briefly discuss the differences between the database environment and the file environment to set the stage for the rest of the monograph.

Ø.1.1 Objectives of FIle Management Systems

a The objectives of toe file environment are:

* Regroup and standardize all input-output and handle overflow automatically to relieve the programmer from these lower tasks. The latter is often performed by the file access method.
* Standardize the typical query and update operations against these files, by providing a language or a set of functions or subroutines for a Host Language Interface (HLI).
* Provide utilities to accomplish typical "file operations" such as sorting, merging, copying, or creating and deleting files.
* Maintain the file structure definition in order to permit the user to refer to the different fields in the relevant file without any precaution. The system knows about their position, type, format and name.

Ø.1.2 Objectives of Database Management Systems

The objectives of the database system include all of the objectives for the file management system, plus the following additional objectives:

* Provide a common base of all data elements in the database, which are relevant to the activities of the users' organization. The structure for these data elements must be of a more refined nature than the file.

* Make these data elements easily available to all appropriate users, both current and future.

* Make these data elements easily accessible to all applications that will depend on the database.

* Ensure the security and integrity of all data elements in a multi-user environment.

* Stabilize the organization's information management environment

Ø.1.3 Comparison of Databse Systems vs. File Management Systems

Database Systems (DSs) are a logical evolution of File Management Systems (FMSs). The following table compares FMS and DS characteristics:

Characteristic	FMS	DS
Data Integrity	Programmer control	DS enforcement
Redundancy	Uncontrolled	Often DS maintained
Data Independence	None or very little, at the field level, e. g. field position	Both Physical and Logical
Data Definition	Proliferation of non-uniform structures	Standardization along the DS "data model"
Access Methods	Often Complex, and handled by programmer, sometimes OS provided	Often provided by DS
Data Vulnerability	Depends on programmer's integrity and user's coordination	Protection integrated in the DS against loss, misuse, damage
Data Control	Distributed among application programmers	Centralized in DD
Access Control	File level only, by OS	At different levels, by DS
Data Recovery	User's responsibility	Automatic or manual, by DS and/or DA

Tuning	Programmer's rewrite, and/or utilities	Depends on Logical and Physical data structures. Some DSs provide automatic tuning
Data Sharing	No concurrent sharing	Concurrent sharing with a range of access controls
Record structure (Data independence, Data Definition)	Defined by programmer	Independent of application programmer
Record Lifetime	Short; frequent changes	Long; Logical Data Structure
Architecture	File definition in multiple sources	Centralized in DD

Ø.2 STEP 1: What are your Database System (DS) requirements?

First, you must define your requirements. These requirements are defined by scanning the CLEC, and assigning a "weight of importance" and a "minimum performance" to each feature corresponding to a node of the CLEC's tree structure. These two numbers are between Ø and 1ØØ inclusive.

Consider a CLEC which which has been abbreviated for the clarity of an example. It would be represented by the following tree-structure:

```
                          root
                          (Ø)
                        g(Ø)=44.6
                          /   \
                        /       \
                      /           \
        data model                  language
        (1)                         (2)
        w(1)=30%                    w(2)=70%
        g(1)=60                     g(2)=38
                                      /    \
                                    /        \
                                  /            \
                     definition            manipulation
                     language              language
                     (2.1)                 (2.2)
                     w(2.1)=20%            w(2.2)=80%
                     g(2.1)=25             g(2.2)=90
```

Let us denote w(n) to be the "weight" of node numbered n.

The grade of node (n) for a candidate DS is the "figure-of-merit" attributed to the candidate DS with respect to the requirement associated with that node. In other words, the grade is a quantitative measure of the manner in which the DS satisfies the requirement.

By definition, the grade of a node is equal to the weighted sum of the grades of its children, unless the node is a "leaf" of the tree (i. e. there is no child-node). In the above example, there are three leaves which are nodes (1), (2.1), and (2.2). In the case of a leaf-node, the grade is assigned by the DS expert who evaluates the candidate DS, and reflects the judgement of this expert with respect to the manner in which the DS fulfills the requirement corresponding to the leaf-node in the tree.

Let us denote g(n) to be the "grade" of node n.

The weight or "weight of importance" is the percentage of influence that a particular node grade has with respect to siblings in the tree-structure composed of the father node and all siblings.

In the example above, we may assign the weights for nodes (2.1) and (2.2) as follows:

w(2.1) = 80%, and w(2.2) = 20%.

Note that 80% + 20% = 100%, that is to say the grade for node (2) is made up from node (2.1) and (2.2), and nothing else. Let us assign the weights for nodes (1) and (2) as well to be

w(1) = 30%   and   w(2) = 70%.

Let us assume now that the grade for an imaginary Database System called "DS1" is 25 for node (2.1), and 90 for node (2.2). We note:

g(2.1) = 25 and g(2.2) = 90.

The grade for node (2) is calculated as the weighted sum of the grades of its children nodes. Thus:

w(2) = w(2.1) * g(2.1) + w(2.2) * g(2.2)

In this case, this becomes:

w(2) = 80% * 25 + 20% * 90, or

w(2) = 38.

Repeating the previous operation, we obtain for an overall figure-of-merit, i.e. w(0), the value 44.6 for this Database System called "DS1".

The "minimum performance" is simply the minimum grade acceptable for the feature associated with the node. The assignment of a minimum grade is a way of stating a mandatory requirement. If no minimum grade is allocated for a node, 0 is assumed.

The CLEC together with the "weights" and "minimum performances" then represent your requirements specifications.

Ø.3 STEP 2: Evaluate a DS with respect to your requirements

The next step is to take a proposed DS and answer the questions expressed in the CLEC. The CLEC and the answers represent the evaluation document.

Comments may be written below each question and discussion in the document, thereby clarifying the answer and the evaluation process for the DS under consideration.

You evaluate a DS in its satisfaction of each criterion. The result will be the assignment of a "grade" for each leaf-node in the tree-structure of the CLEC.

For a DS under evaluation, if a "grade" for any node is less than the minimum prescribed (minimum performance) for that node in the customer's requirements, the DS under scrutiny may be eliminated from the list of DS candidates.

It may be of interest to pursue the grading process to its completion in spite of an elimination, because of the possibility of a change in the customer's requirements. In the eventuality of such a change, it is possible for the minimum performance to be satisfied, in which case the grades would be ready for a new calculation of an overall "figure-of-merit".

Ø.4 STEP 3: Calculation of an overall "figure-of-merit" for a DS

The definition of the "figure-of-merit" for a DS under evaluation is the weighted sum of the grades for all level 1 nodes in the tree-structure. Recursively, the grade for any node in the tree is the weighted sum of the grades for all its children in the tree.

This calculation is easily implemented by a program, written for instance in some DS language.