DOI: 10.5281/zenodo.7217494

OPTIMIZING ECONOMIC APPLICATIONS BY OPTIMIZING DATA FLOWS PRESENT IN DIFFERENT MODULES INTEGRATED IN INFORMATICS TECHNOLOGY SYSTEMS

Dănuţ-Octavian SIMION, PhD Associate Professor
Athenaeum University, Bucharest, Romania
danut_so@yahoo.com

Emilia VASILE, PhD Professor
Athenaeum University, Bucharest, Romania
rector@univath.ro

Abstract*: The paper presents the solutions for Optimizing economic applications by optimizing data flows present in different modules integrated in informatics technology systems. The architecture of the information system represents the generic solution regarding the data processing processes that must be carried out and the way of data and processing integration. In other words, the architecture represents the constructive solution of the IT system and reflects the strategic managerial vision of how the organization (company) works. The company's global IT system is broken down into subsystems, each of which covers a distinct field of activity. In turn, each subsystem is broken down into applications, each of which covers a distinct activity within the domain. For example, the IT subsystem for the commercial domain will be broken down into distinct applications for each of the following activities: supply, sales, and marketing. The decomposition process also continues in the stage where procedures will be defined for each application performing distinct functions within the application, for example, procedures for directing processing, procedures for updating the database, procedures for consulting the database. In turn, the procedures are broken down into modules. They comprise code sequences, each performing a distinct function within the procedure. For example, a database update procedure will include: a module for adding records, a module for modifying records and a module for deleting* *records. The architecture promoted in the realization of decentralized systems is the client-server architecture characterized by the fact that the applications and data made available to users are dispersed on the various hardware components depending on the number of users who must have access and the required computing power.*

Keywords: *optimizing business modules, advanced solutions, data integration models, subsystems of business applications, procedures for processing economical data, advanced architecture in applications, business analyzes*

JEL Classification: *C23, C26, C38, C55, C81, C87*

1. Introduction

The links between data correspond to the associations that can be made between the objects of an IT application. Any DBMS must allow the definition and description of the data structure, as well as the links between them, according to a data model. Each type of data model allows certain connections between data. A DBMS, which implements a certain data model, will also have to ensure the creation of links between the corresponding data in accordance with the conceptual scheme.

Data sharability refers not only to the aspect of ensuring the access of several users to the same data, but also to the possibility of developing some applications without changing the structure of the database. The issue of sharability is raised to a higher level for DBMSs that allow networking (Schubert & Hütt, 2019; Livesey 2007).

The overall performance of the application is influenced by the DBMS. It must manage a large volume of data of high complexity, within a certain reasonable access time for different users. For all these things, the DBMS uses different access methods, optimization techniques, data types. Their implementation is done in specialized components of the DBMS. All the above objectives were taken into account in the evolution of DBMSs, each generation improving performance.

The achievement of the previously presented objectives is ensured by the database management systems through a series of components that allow the performance of specific operations. Depending on their nature and purpose, the operations can be grouped by activities. The activities also accept a grouping by functions so that one or more activities, relatively homogeneous, will perform a certain function (Sieja & Wach, 2019; Karimi & Khandani, 2020).

The DBMS allows defining the structure of the database using the Data Definition Language (LDD). Data definition can be done at a conceptual, logical and physical level. The attributes (fields) within the structure of the database are described, the links between the entities of the database or between the attributes of the same entity, the possible data validation criteria, the data access methods, the aspects related to ensuring the integrity of the data are defined. The embodiment of this function is the database schema, stored in internal code. The storage is done in a file, which allows the database structure to be displayed and updated at any time. This function has been greatly automated over time, with LDD now having few commands. LDD is specific to each DBMS, but it always performs the data description according to the elements of the data model that the respective DBMS implements. Thus, the definition and description of the entities and their characteristics, the definition of the links between objects (associations) and the specific integrity rules of the data model are achieved.

After performing the description function, in a DBMS, the database entities already exist created as files, but they do not contain the actual data, but only the database structure (database schema).

2. Optimizing business flows for different types of data

The data manipulation function is the most complex and performs the updating and retrieval of data from the database, using the data manipulation language (DML). The following activities are carried out: uploading, updating, processing and retrieving data.

a) Uploading data into the database is done through automated or programmed operations that also ensure the necessary validation criteria.

b) Updating the database consists of adding, modifying and deleting records. For addition and modification operations, the same validation criteria that were used for the data loading activity are kept. The update is carried out only authorized, by ensuring an appropriate protection of the data, in order to preserve the coherence of the database.

c) Data processing is carried out through selection, ordering, interclassing (composition) and ventilation (decomposition) operations performed on the database entities. These are usually operations preparatory to the data retrieval activity. Many of the processing operations are carried out with the help of operators from the data model that the DBMS implements.

d) Retrieving (querying) the data consists of visualization operations (display on the screen, printing on paper), browsing, editing some output situations. The output situations can be intermediate or final and can be obtained on different technical information media (screen, paper, magnetic medium, optical medium). They can have the most different forms (points, lists, reports, graphics, images, sound, video) and can be obtained according to the most different retrieval criteria.

The data manipulation function is the most used in databases and is better supported by the DBMS than any other external memory data management system that exists so far.

The DBMS manipulates data in an efficient way, using for this purpose a series of methods and techniques to optimize access and the allocation of memory space in the computer. Every DBMS from one generation to another and even from one version to another has sought to improve this aspect. The manipulation function is provided in the DBMS by an LMD. It must respect the data integrity restrictions and implement the operators from the data model on which the DBMS to which it belongs is based.

LMD can be host language or own language. Those with a host language are developed by adapting universal programming languages to the work requirements of the DBMS. In this way, the power of a universal language is combined with the needs of data retrieval (example: ORACLE). Those with their own language are developed through a specific language capable of combining the power of the procedural with retrieval from a certain type of database (Korab 2021; Riechmann 2021).

For retrieval activation there are specialized query languages that can be included in LMD or exist as such. They appeared due to the widening of the range of database users, which increasingly includes non-IT people. For them, simple and friendly retrieval facilities were needed, close to the human way of working and thinking. In this sense, relational languages like SQL have the best performances.

The database user function provides the set of interfaces necessary for all users to communicate with the database. In order to achieve this function, the DBMS must provide facilities for several categories of database users: non-computer scientists, specialists (computer scientists), the administrator.

Non-IT users represent the main category of information beneficiaries (end and intensive users) from the database. The DBMS offers them non-procedural languages and other query facilities (generators, utilities, etc.) of the database in a simple and interactive form. These users do not need to know the structure of the database and do not need to know how to program, the DBMS interactively helping them to use the database. In this sense, the DBMS offers: menus with suggestive options, windows, and templates for different forms, wizard-type assistants, self-documentation or helps, explanatory messages/windows.

Computer specialist users create the database structure and perform complex database exploitation procedures. The DBMS offers these users the description language and the data manipulation language as well as interfaces with universal languages. These are of different complexity and power, from one DBMS to another, offering both non-procedural and procedural elements to the IT specialist. With these elements it describes the database schema and provides complex data manipulation. To create the database, the DBMS offers the specialist elements of CASE (Computer Aided Software Engineering). They help him in the various activities involved in the database design stages.

The database administrator, who is a special user and has a decisive role in the optimal functioning of the entire system. Due to the importance of this category of users, the DBMS has a distinct function in this regard.

The administration function is complex and the competence of the database administrator. The administrator, who has a rich experience of analysis, design and programming, organizes and administers the database in all stages of its creation.

Thus, he organizes the database according to a certain methodology, makes the (conceptual) scheme of the database, and coordinates the design of the database. For all these things, the DBMS offers a series of CASE elements, as well as a series of specialized utilities.

In the database exploitation stage, the administrator has the role of authorizing data access (granting accounts, passwords, etc.), restoring the database in case of incidents (by journaling, copies), efficiently using the storage space internal and external memory (through organization, optimization routines), to perform a series of statistical analyzes from the database (number and type of users, number of accesses, number of updates, etc.). For each of these activities, the DBMS offers tools and working techniques.

In the case of working in a network of computers with distributed databases, the DBMS has very developed components intended for the administrator. This is determined by the fact that the database is, in this case, of great complexity, the data is distributed on the computers in the network, and the users are of all types and in large numbers.

Mainly for the administration function, but partially also for the other functions, the DBMS ensures the protection of the database. The essential aspects will be presented here, and then for each type of DBMS the specific aspects that appear will be specified through examples.

The diversity of DBMSs, which were and are in operation on different computers and under different operating systems, requires their classification according to different criteria:

1) According to the computing systems on which they are implemented:

- DBMS for large computers are used for very complex and very large databases.

- DBMS for minicomputers are used for complex and large databases and experienced strong development in the 80s (example: Oracle).

- DBMS for microcomputers are used for databases of small and medium complexity and size. They are very widespread at the moment.

The current trend is for the DBMS to be compatible on as many computing systems under as many operating systems as possible. This is dictated by the new technology of creating open systems type applications, which is also taken into account by the new (third) generation of DBMSs.

2) According to the programming language used:

- The host language DBMS is the one that has a data manipulation language based on a high-level (universal) one. The host language can even be a universal language or an extension (adaptation) of such a language. The advantage of this solution is that complex program procedures can be developed, very good human-machine interfaces can be created, and programming experience from high-level languages can be used (all result from the advantages of procedural programming). The major disadvantage is that the formulation of retrieval requests is made more difficult, often in a way that is inaccessible to end users. This is supplemented by SGBD through other specific components: generators, utilities, etc.

- The DBMS with its own (autonomous) language is the one that has a specific data manipulation language. This own programming language is procedural and has the great advantage of allowing the implementation of all the facilities offered by the DBMS. In it, complex procedures and powerful interfaces can be programmed as in a universal language, but in addition easy and optimized access to the database is achieved. The disadvantage is that such a language can only be used by IT specialists (eg the language from Visual FoxPro).

The current trend is for the DBMS to have implemented, in addition to a procedural language, a non-procedural retrieval language, which allows the formulation of retrieval requests easily, by all users of the database. In this sense, most DBMSs for microcomputers have implemented, partially or totally, the SQL language, which is also internationally standardized.

3) According to the logical data model implemented:

- Hierarchical DBMSs are those that implement the tree (hierarchical) data model and were the first to be used for database management. They have a number of advantages for precise domains in the surrounding real world, for example machine building technology, but have limitations for other domains (eg: IMS).

- Network DBMS are those that implement the network data model and have removed many of the limitations of the hierarchical ones. They have wide applicability to many real-world problems, but are difficult to use due to their high complexity (example: IDMS).

- Relational DBMSs are those that implement the relational data model and have applicability in most domains in the real world. They can be used by a wide range of users thanks to the facilities offered (generators, non-procedural language, etc.) (examples: Oracle, Visual FoxPro, Paradox, Access, Informix, Progress) (Schubert & Hütt, 2019; Karimi & Khandani, 2020).

- Object-oriented DBMSs are those that implement the object-oriented data model. They lend themselves well to very large problems, of high complexity, as well as to new types of applications (aided design, multimedia, open systems) (examples: O2, Orion, Jasmin).

Most current DBMSs have implemented, in the latest versions, facilities for handling objects. The above types of DBMS, based on the implemented data model, are basic (fundamental). Starting from these, through the extension of information technology, there are other types of DBMS: deductive, distributed, multimedia, spatial, etc. There are also types of DBMSs (such as functional ones) that are based on other data models than the ones above. These, however, have a narrower spread, having performances only for well-specified domains.

4) After locating the database

- Centralized DBMSs are those that manage data located in a single central database. All authorized users have access to them to perform various data manipulation operations. All computers that are not networked and work with databases have a centralized DBMS installed. A centralized DBMS, but with networking facilities, must also be installed in computer networks that have placed the database on a single computer (usually on the server). (example: Visual FoxPro, Access)

- Distributed DBMS are those that manage the data located on several computers in a network, treating them as a unitary whole. The complexity of these DBMSs is high, having special components for making connections and distributed data processing.

3. Usage of different types of data in encapsulated modules

Encapsulated data require specialized algorithms and techniques for their use in economic applications that incorporate specific economic flows of companies. Users of business applications need easy and fast access to data, and application optimization through encapsulation and standardization provides competitive advantages compared to classical data types (Sieja & Wach, 2019; Riechmann 2021).

The following example shows a route optimization method for a distribution company:

```
Sub optimized_route()
 Dim k As Integer
 k = 1
 init k, st
 While k > 0
  Do
   suc_col hv_suc, st, k
   If hv_suc = True Then
    valid_col ev, st, k
   End If
  Loop Until (Not hv_suc) Or (hv_suc and ev)
  If hv_suc Then
   If sol(k) Then
    tip_col
   Else
    k = k + 1
    init k, st
   End If
  Else
   k = k - 1
  End If
 Wend
End Sub

Sub suc_route(hv_suc As Boolean, st As stiva,
k As Integer)
 If st.ss(k) < n Then
  hv_suc = True
  st.ss(k) = st.ss(k) + 1
 Else
  hv_suc = False
 End If
End Sub
Sub valid_route(ev As Boolean, st As stiva, k
As Integer)
 ev = True
 For i = 1 To k - 1
   If (st.ss(i) = st.ss(k)) And (mat.m(i, k) =
1) Then
   ev = False
  End If
 Next
```

```
End Sub

Sub tip_route()
 Dim i As Integer, b As String
 b = „ „
 For i = 1 To n
 b = b + „Loc = „ + Str$(i) + „; vizited „ +
Str$(st.ss(i)) + „ „
 Next
 MsgBox b
End Sub
```

Another method of use in modules specific to economic applications is described in the following program source:

```
Private Sub App_sub1()
 Dim n As Integer, k As Integer, a As vector
 r_n „n = „, n
 r_data „a”, n, a
 template „string is : „, n, a
 k = n
 Do
  k = k / 2
  Do
   b = 1
   For i = 1 To n - k
    If a.v(i) > a.v(i + k) Then
     x = a.v(i)
     a.v(i) = a.v(i + k)
     a.v(i + k) = x
     b = 0
    End If
   Next
  Loop Until Not (b = 0)
 Loop Until Not (k <> 1)
 ,MsgBox „String sorted for accesing „
 template „String sorted is”, n, a
End Sub
```

Data sources require special integrations and their use through different templates, and the complexity of operations can be reduced by encapsulations and modular uses in the application logic or in the activity of describing economic flows (Korab 2021; Livesey 2007).

4. Conclusions

The main objective pursued through the introduction of an economic IT system is the selective and timely provision of all levels of management with necessary and real information for the substantiation and operative elaboration of decisions regarding the most efficient performance of all activities in the economic unit. The main objective therefore refers to the entire activity in the economic unit. In order to get to know the activity more closely, and to carry it out in the best conditions, other secondary objectives can be defined, which are called „conditions" for the achievement of the main objective. There must be compatibility between the main objective and the secondary objectives, in the sense that the achievement of the secondary objectives must lead to the achievement of the main objective. (Sieja & Wach, 2019; Korab 2021). Economic IT systems require successive modeling of data to ensure easy integration and optimal encapsulation in objects that can be used in application modules that model economic flows specific to business activities.

References

Karimi, N, Khandani, K. (2020). Social optimization algorithm with application to economic dispatch problem. *International Transactions on Electrical Energy Systems* (*Wiley Online Library*), 30(11), e12593. https://doi.org/10.1002/2050-7038.12593.

Korab, P. (2021). Use of Machine Learning in Economic Research: What the Literature Tells Us. *Towards Data Science,* https://towardsdatascience.com/use-of-machine-learning-in-economic-research-what-the-literature-tells-us-28b473f26043.

Livesey, D. A. (2007). The importance of numerical algorithms for solving economic optimization problems, *International Journal of Systems Science*, 5:5, 435-451, DOI: 10.1080/00207727408920112.

Riechmann, T. (2021). An Exemplary Introduction to Structure and Application of Genetic Algorithms in Economic Research, *Researchgate.net,* DOI:10.1007/978-3-642-57612-6_3.

Schubert, C., & Hütt, M.-T. (2019). Economy-on-demand and the fairness of algorithms. *European Labour Law Journal,* 10(1), 3–16. https://doi.org/10.1177/2031952519829082.

Sieja, M., & Wach, K. (2019). The Use of Evolutionary Algorithms for Optimization in the Modern Entrepreneurial Economy: Interdisciplinary Perspective. *Entrepreneurial Business and Economics Review*, 7(4), 117-130. https://doi.org/10.15678/EBER.2019.070407.