# Data Storage and Processing in Real-Time Information Systems Using Multidimensional Technologies

Juri S. Kabalnov
Department of Computer Science and Robotics
Ufa State Aviation Technical University
Head of the Chair of Computer Science
Professor, D.Sc. (Eng.)
12. K.Marx St. Ufa, 450000, Russia
e-mail: informatic@ugatu.ac.ru
phone: +7(3472)237876

Olga I. Christodulo
Department of Computer Science and Robotics
Ufa State Aviation Technical University
Assistant professor, D.Sc.
12. K.Marx St. Ufa, 450000, Russia
e-mail: postmaster@tc.ugatu.ac.ru
phone: +7(3472)237835

Dmitry V. Ivlev
Department of Computer Science and Robotics
Ufa State Aviation Technical University
Master of science
12. K.Marx St. Ufa, 450000, Russia
e-mail: postmaster@tc.ugatu.ac.ru
phone: +7(3472)237835

Alexander A. Levkov
Department of Computer Science and Robotics
Ufa State Aviation Technical University
Master of science
12. K.Marx St. Ufa, 450000, Russia
e-mail: postmaster@tc.ugatu.ac.ru
phone: +7(3472)237835

## Abstract

The work deals with the problem of building data storage and processing systems for real time information systems (RTIS). The authors suggest considering this problem as a development of a pair of connected information objects with clearly defined levels of abstraction. The first object is a physical, and the second, a logical model of data storage. The physical model may be built using a particular type of indexed structure based on binary irregular trees, providing a more efficient and optimal structure and sparing losses in speed as compared with classical structures. Using this structure we resolve the problem of disperse data storage. The logical model of data representation offers an opportunity to qualitatively upgrade the level of information visualization with respect to available products. The model is based on the algebra of binary multi-index variables. This permitted us to create a flexible and powerful language of query for work with large amounts of information.

**Proceedings of the Workshop on Computer Science and Information Technologies CSIT'2000**
**Ufa, Russia, 2000**

## 1. Introduction

RTIS are widely used nowadays. Besides simply storing information, they provide active continuous work with it. Simple RTIS, which are characterized by limited amount of information ($10^4$ – $10^8$ bytes) and relatively simple queries involving no complex analysis of data, admit conventional methods of data storage and processing (such as the well-known relational approach) [1]. Large, complex RTIS require methods that would provide good operation speed.

Complex RTIS are characterized by:

1. Large amounts of information stored ($10^8$ – $10^{13}$ bytes);
2. High standards of storage reliability and data integrity;
3. Multi-user work mode;
4. Free use of complex analytical queries involving complex processing of large data arrays. Analytical queries are queries aimed at aggregation of data, prognosis of indices, search by fuzzy conditions, etc.;

5. Work in real time.
6. Advanced dynamic structure and content (which means that the system's structure at a given moment cannot be predicted).

Multidimensional presentation of information is one of the most successfully and actively developed technologies for solving the problems above. This technology underlies multidimensional data bases (MDB), which allow processing of large data arrays due to a flexible query language that permits simple formalized expression of involved operations. MDB were originally intended for parallel work in advanced multi-user systems [1] and, moreover, they offer advanced support of complex analytical queries that are characteristic of data processing methods, as compared with relational data bases (RDB).

At present, however, this approach suffers from a number of disadvantages that put constraint on its use in large complex real-time systems. The multidimensional approach lacks reliability of data storage and ensuring of its integrity; this is characteristic of all logically distributed structures used in the concept of MDB polycube_construction. Also, MDB are unable to function in the mode of real time [2], since their speed strongly depends on the queries processed. The larger part of them comes unforeseen, as in real-time systems, and is processed with considerable loss in speed, due to the RISC-architecture of MDB.

Modern MDB solve the problem of data storage by using algorithms of data backup and archiving, by which achievement of required reliability is possible but is unreasonably costly and resource-consuming. At the same time the RISC-architecture of MDB does not allow their work in real time [3].

In our opinion, all the above difficulties associated with multidimensional data storage systems result from inadequate division of labor between the logical and the physical data structures. Both the physical and the logical data base management system structures must be assigned their own strictly typified functions, admitting no intersection except for support of data storage reliability.

Proceeding from the above considerations, we propose that a multidimensional data storage system should be clearly subdivided into at least two levels of abstraction (LOA):

1. The level of logical representation of data (and the logical model thereof);
2. The level of physical representation of data (the physical model).

Development of any data storage system should start, in the first place, with distribution of functional demands placed on the physical and the logical data models. For each level, we must immediately define all liabilities of the abstraction that are assigned to the interface of the corresponding data models. These liabilities may be prescribed arbitrarily but such as to preserve their consistency and stability.

We find it reasonable to impose the following tasks on the logical model: support of data integrity on the logical level; arrangement of a dialog with the user via the calculus or the algebra of virtual data structure interaction; support of complex queries by simple linguistic constructions.

As for the physical model, it is supposed to provide, besides physical data storage reliability and fail-safety, also optimal storage of disperse data and quick navigation using this data, as well as quick algorithms of data addition and removal. Note also that these demands should be applicable to not only single elementary units of information (as an RDB attribute), but also to rather large lists and sets of information.

## 2. A logical model of data representation

Our research in this field is aimed at building a logical model of data representation that should be applicable to complex analytical queries expressed in ultimately simplified natural language constructions, and in minimum time. This will require the following:

1. Development of a new concept of data representation;
2. Development of a mathematical formalism for handling the elements in this model;
3. Formulation of the main postulates of an algebra allowing operation with complex information objects on the basis of the proposed mathematical formalism;
4. Construction of a query language, flexible enough to provide processing of complex queries taking advantage of the newly built model concept.

With this in view, we present a number of original proposals.

A novel (bit multidimensional) model of data representation is based on the traditional MDB concept (i.e. the data model used by Oracle) [4], but with a few important modifications issuing from the fact that it does not employ the notion of indices. Instead, the so-called flags (indications of presence) are used. The novelty is essentially in that each of the indexed groups found in a traditional multidimensional model of data representation is reduced to a separate dimension. As a consequence, the proposed data model has hypercube rather than polycube architecture. All hypercubes are united in one, because their "content" is no longer typified (comprising nulls and units) [5]. The general scheme of the bit multidimensional model of data representation is shown in Fig. 1.

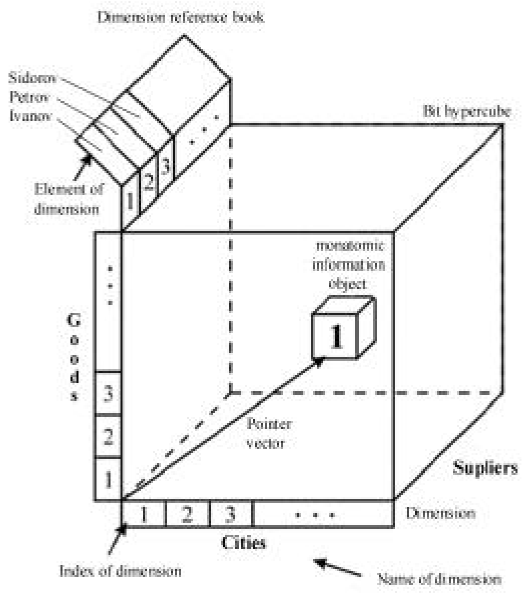

**Figure 1. Multidimensional bit model of data representation**

The interface of the model is subdivided into three levels of abstraction.

The first LOA is a set of elementary operations over monatomic objects of a bit hypercube. Monatomic objects are indivisible units of information, that is, in terms of the bit model, the nulls and units located in the hypercube cells. Besides, we consider the associated hypercube elements that perform addressing of monatomic units, i.e. the dimension and the element of dimension. On this LOA, the basic operations of variation, addition and removal of the bit hypercube monatomic objects are defined. The first LOA is entirely concealed both from the users and the database manager, and serves as a basis for defining the second LOA operations over complex information objects.

The second LOA is an interface for interaction between formalized syntactical structures and elementary operations over monatomic objects. The methods used on this level are based upon the algebra of multi-index variables, which, in its turn, is built on a set of elementary operations over the bit hypercube monatomic objects. Complex information objects, representing multidimensional bit spaces, serve as arguments of functions and methods of the second LOA. This level may be said to be the language of machine representation and handling of data and is therefore inconvenient for the end user.

In order to establish interaction of the system with its users, we complete the second LOA with the third, higher LOA, which is the direct formalized language of queries addressed to the MDB. This language belongs to the group of data control languages (informally called “the fourth generation languages”), and offers the user a powerful, flexible and handy tool of data control.

As we use a multidimensional logical model of data representation, we can formulate the major tasks to be fulfilled in order to build the desired physical model:

1. Achievement of a highly reliable and least redundant structure of disperse information storage and navigation;
2. Implementation of mechanisms for mapping of multidimensional structure information on the 2D structure of physical storage devices;
3. Support of multidimensional queries based on this physical model;
4. Adaptation of the storage structure to the given formalism basis.

Of the above tasks, we believe the former two are most important. The adaptation problem is mostly a problem of application rather than of conception, and may be reduced essentially to a closer following of the tasks 1 and 2 by the development engineer. As for the multidimensional query support, it is largely dependent on the language employed by the logical model and is in fact the problem resolved within the interface between the logical and the physical models. This is of course an important problem, but its solution is taken on at the final stage of development of data base management systems and is technical by nature.

## 3. Physical model of data storing

When we undertake the task of building a highly reliable and minimally redundant structure of disperse information storage and navigation, we must give a more clear definition to the main demands placed on it:

1. *Ensuring of enhanced reliability of information storage*. This implies both fail-safety and restorability of the structure;
2. *Optimal efficiency of using memory resources*. This structure must provide maximum compactness of information storage;
3. *High-speed information retrieval, removal, addition and variation*. Note the additional demand of the *ability to process large data arrays* in one transaction, i.e. one entry of the structure.

In the course of extensive investigation in the field of physical storage of information [1,5,6,7], we have developed a structure *based on looped binary irregular trees* as the means of securing enhanced reliability and optimality of data storage (under *irregularity of a binary* *tree we understand different capacity of coupled lists entering into different parent nodes*)

The general scheme of this structure is given in Fig.2.

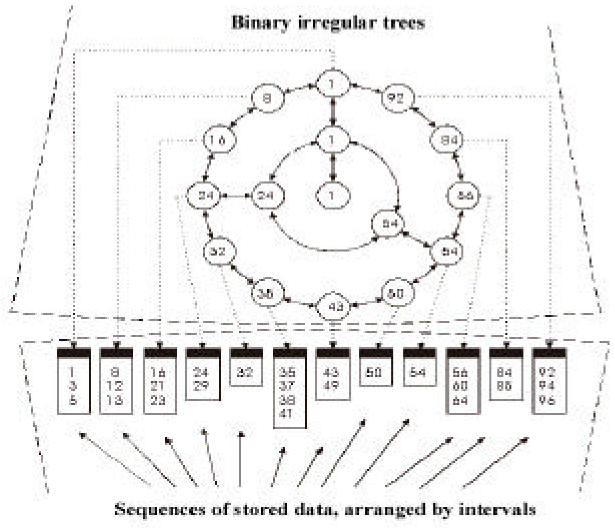

**Figure 2. Binary irregular tree**

The counter-directed structure of references and the multilevel looping allow considerable enhancement of the structure reliability. This redundant structure ensures data protection unless a sufficiently large group of references arranged in a particular manner is lost.

The counter-directed backup allows us to avoid the structure crash in case of losing an element. In contrast to single-direction lists where all data that follows the lost element is destroyed, in the proposed structure this error would result in nothing more than a double run through the structure in search of the desired record: the first run goes in a standard direction (counter-clockwise in Fig.2) until a failure element is discovered, and the second run goes clockwise until the desired element is found.

Multidimensional looping ensures that the lower-level elements storage security is independent of their parent elements. It may be seen that, provided the access to at least one path from the central No.0 node down to any lower-level element (i.e. the full-length path), all the lower-level data becomes available due to counter-parallel references.

The structure also makes use of dynamic redundancy. Each element may contain an arbitrary number of duplicate references. This arrangement allows enhancement of fail-safety of the most important elements in the structure. Thus, for instance, we may assign a number of full-length paths by providing their elements with enhanced redundancy to ensure preservation of at least one such path. Real-time systems require, along with fail-safety and restorability, also high speed, particularly when a damaged structure is concerned, therefore creation of such a way guarantees the best time of search for a damaged structure element with preset probability.

Another important task concerns optimal utilization of the memory resources by the physical structure of data storage. It has been shown that the minimum redundancy in ubiquitous structures exceeds 50% [4]. Thus a fully indexed data base (up to 200 MB) occupies 3 or 4 times the room occupied by an non-indexed small or medium-size structure [5]. Such redundancy is unacceptable for large and superlarge systems due to exceedingly high overhead expenses. The complex structure of data and inability to apply query-expected indices prevents the fast and effective retrieval in such data bases [2]. In this case, the problem is in determination of the optimal balance of the speed of retrieval and the amount of redundant information. These two quantities are inversely proportional, i.e. the higher the speed of retrieval, the lower is the optimality of storage, and vice versa. The proposed model offers the most flexible solution to this problem. Irregularity of the structure allows assignment of a different number of daughter elements to different elements of the higher levels; hence regulation of both the speed and redundancy of retrieval. The intelligent system of irregularity formation makes it possible to assign to frequently used elements a faster time than to those rarely used, due to the loss in speed when addressing to the latter. In addition, we may apply a number of templates describing probability distribution for the presence of an element in a list. The processing speed may be significantly increased by indicating such a template in the parent list, for determination of the probable location of the element in the daughter list. This is particularly useful when the parent element possesses more than 100, rather than 2 to 10, daughter elements. In this case, the speed may increase 2- to 20-fold, depending on the adequacy of the prescribed templates to the actual probability distributions. Generation of such templates is done by the structure itself and is dynamic, which allows the structure to react flexibly and rapidly to the variation in its own content. Moreover, by using these templates we can cut the redundancy of the structure down to 1 to 3%, retaining at the same time the performance as in structures with 50 to 100% redundancy..

Another problem is development of mechanisms of mapping information in multidimensional structures on the planar linear structure of physical storage. One of the methods to translate multidimensional to planar structures is to directly transform the addresses (as in multidimensional arrays) by the formula: Alin = $\sum_{i=1}^{N} A_i *$ $li$, where $N$ is the number of dimensions in a multidimensional structure, $A_i$ is the element's address in the $i$-th dimension, $l$ is the number of elements in the $i$-th dimension. This addressing has the advantage of easy mutual translation of addresses, due to the extremely high performance of modern computers when calculating integer expressions (and the above formula pertains to such expressions). The approach has a disadvantage, too, as it is applicable to static multidimensional structures alone, in which $l_i$ depends on dynamic qualities. By adding a new element to the dimension, we create a situation when all indices in an already formed tree-like structure become incorrect and require transformation. We propose a more optimal method of mapping, based on splitting all dimension ranges into definite sets, and arranging them into all possible combinations in the form of a looped irregular binary tree. This will ensure easy retrieval of data in this structure by any dimension or a group of dimensions.

Disperse multidimensional data creates one of the most serious problems during their storage [4]. In order to give the most optimal solution to this problem, we suggest that the so-called refusal nodes, i.e. data-free range peaks, should be introduced into the tree structure. During retrieval of a large amount of information, this would provide elimination of the most of the 'false' (i.e. non-existing in the data base) elements on the higher levels by range, rather than on the last level by individual elements, thus leading to 5- or 6-fold gain in time when retrieving the whole data array.

## 4. Conclusions

The paper presents an object-oriented approach to RTIS development. Authors offer the whole concept, which is based on multidimensional technologies of data storage and processing and allows building complex analytical queries and an optimal, highly reliable model of data storage. The work also makes use of the subdivision of the logical model into levels of abstraction (which facilitates the building and the functioning of the language of queries addressed to data bases) and the effective method of mapping multidimensional data on the structure of physical data storage. The obtained system is characterized by high operation speed, optimality and reliability of work as compared with the available products, providing, at the same time, a considerably easier, convenient and intuitively clear working language for data handling.

During the trial period of our DB working in oil obtaining industry authors has executed some tests of several DB's. This performance benchmark is represented on the Figure 3 (MD bit DB is the internal name of our database).

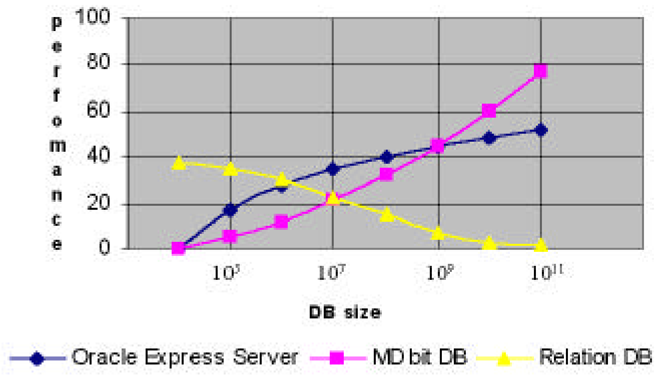

**Figure 3. Performance of databases with various information storing models**

## References

1. Bobrovsky S. ORACLE 7: Client-Server Calculations. "Lori", 1996 (In Russian).
2. Chaudhuri S., Dayal U. An Overflow of Data Warehousing and OLAP Technology. "SIGMOD Record", March 1997,
3. Inmon W.H. Building the Data Warehouse. "Prism". 1995.
4. Sakharov A.A. Principles of Design and Use of Multidimensional Databases Illustrated by Oracle Express Server. "Data Base Management Systems". 1996. No.3 (In Russian).
5. Ramodin D. D3 Server 7.0: Postrelational Modern Database Management Systems. "Mir PC". 1997. No.3.
6. Shulenin A. Microsoft Technologies for Data Analysis. http://www.citforum.ru/seminars/cis99/ms.shtml
7. Kogalovsky M.R. Abstractions and Models in Database Systems. "Database Management Systems". 1998. No.4,5.