Bibliography

1. A.E. Gay, S.V. Mostovoi, V.S. Mostovoi, A.E. Osadchuk. Model and Experimental Studies of the Identification of Oil/Gas Deposits, Using Dynamic Parameters of Active Seismic Monitoring, Geophys. J., 2001, Vol. 20, pp. 895-9009.
2. S. V. Mostovoi, A.E. Gui, V. S. Mostovoi and A. E. Osadchuk Model of Active Structural Monitoring and decision-making for Dynamic Identification of buildings, monuments and engineering facilities. KDS 2003, Varna 2003, p. 97-102
3. Kondra M., Lebedich I., Mostovoi S. Pavlovsky R., Rogozenko V. Modern approaches to assurance of dynamic stability of the pillar type monument with an application of the wind tunnel assisted research and the site measuring of the dynamic characteristics. Eurodyn 2002, Swets & Zeitlinger, Lisse, 2002, p. 1511 - 1515.
4. Mostovoi S., Mostovoi V. et al. Comprehensive aerodynamic and dynamic study of independance of Ukraine monument. Proceedings of the national Aviation University. 2' 2003, pp. 100 - 104.

Authors' Information

Sergey V. Mostovoi – Institute of Geophysics of the National Academy of Sciences, Kiev, Ukraine. e-mail: smost@i.com.ua; most@igph.kiev.ua

Vasiliy S. Mostovoi – Institute of Geophysics of the National Academy of Sciences, Kiev, Ukraine. e-mail: vasmost@i.com.ua; most@igph.kiev.ua

BUILDING DATA WAREHOUSES USING NUMBERED INFORMATION SPACES

Krassimir Markov

Abstract: An approach for organizing the information in the data warehouses is presented in the paper. The possibilities of the numbered information spaces for building data warehouses are discussed. An application is outlined in the paper.

Keywords: Data Warehouses, Operational Data Stores, Numbered Information Spaces

ACM Classification Keywords: E.1 Data structures, E.2 Data storage representations

Introduction

The origin of the Data Warehouses (DW) can be traced to studies at MIT in the 1970s which were targeted at developing an optimal technical architecture [Haisten, 2003]. The initial conception of DW had been proposed by the specialists of IBM using the concept "information warehouses" and its goal was to ensure the access to data stored in no relational systems. In 1988, Barry Devlin and Paul Murphy of IBM Ireland tackled the problem of enterprise integration head-on. They used the term "business data warehouse" and defined it as: "a repository of all required business information" or "the single logical storehouse of all the information used to report on the business" [Devlin and Murphy, 1988]. At present, the conception of "data warehouse" becomes popular mainly due to activity of Bill Inmon. In 1991, he published his first book on data warehousing.

W.H. Inmon's definition is: "Data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision making process" [Inmon, 1991]. Let remember, the data warehouses allow long term information about an enterprise to be recorded, summarized and presented. Usually the data warehouse is a passive observer object that takes no part in business processes, and is not part of the business model. The axes of a multidimensional data warehouse are not arbitrary, but represent real aspects of the business. Axes should represent the purpose, process, resource and organization aspects. The summary hierarchies on each of these axes should parallel the fractal structures in the business model. Roll up and drill down to zoom from summary to detail information is therefore based on the structure of the business, so is meaningful to management and other users. [Marshall, 1997].

As a rule, the typical enterprise has many different systems for operative processing with very incompatible data. In such case, the main task is to convert the existing archives of data into a source for new knowledge which will give to the users a uniform integrated and consolidated notion of the corporate data. The old systems for operative information processing have been developed without foreseeing the support of the requirements of modern business and the need of automated support of decision making. Because of this, the converting the usual systems for online transaction processing (OLTP) in the systems for decision support (resp. – DW) were very complicated task. To solve this problem, an intermediate level has been proposed – the "operational data stores". The Operational Data Store (ODS) is a database designed to integrate data from multiple sources to facilitate operations, analysis and reporting. Because the data originates from multiple sources, the integration often involves cleaning, redundancy resolution and business rule enforcement. An ODS is usually designed to contain low level or atomic (indivisible) data such as transactions and prices as opposed to aggregated or summarized data such as net contributions. Aggregated data is usually stored in the DW [Wikipedia, ODS].

The definition of ODS given by Bill Inmon is: "an ODS is a subject-oriented, integrated, volatile, current-valued, detailed-only collection of data in support of an organization's need for up-to-thesecond, operational, integrated, collective information". [Inmon, 1995]

At first glance the ODS appears to be very similar to the data warehouse in structure and content. In some respects there are strong similarities between the two types of architectural constructs. But the ODS has some very different characteristics from the data warehouse. Both the ODS and the data warehouse are subject-oriented and integrated. In that regard, the two environments are identical. Both environments require that data be integrated and transformed as it passes into the ODS and/or the data warehouse. But here the similarities between the ODS and the data warehouse end. The ODS contains volatile data while the data warehouse contains non-volatile data. Data is updated in the ODS while data is not updated in the data warehouse. Another important difference between the two environments is that the ODS contains only very current data while the data warehouse contains both current data and historical data. The data in the data warehouse is not nearly as fresh as the data in the ODS. The data warehouse contains data that is no more current than the last 24 hours. The ODS contains data that may be only seconds old. Another major difference between the two architectural constructs is that the ODS contains detailed data only. The data warehouse contains both detailed and summary data. There are then some major differences between the types of data found in the two environments. One of the most important features of the ODS is the system of record. The system of record is the formal identification of the data in the legacy environment that feeds the ODS. (Figure 1) [Inmon, 1995]

So, an operational data store (ODS) is a type of database often used as an interim area for a data warehouse. Unlike a data warehouse, which contains static data, the contents of the ODS are updated through the course of business operations. An ODS is designed to quickly perform relatively simple queries on small amounts of data (such as finding the status of a customer order), rather than the complex queries on large amounts of data

are typical of the data warehouse. An ODS is similar to your short term memory in that it stores only very recent information; in comparison, the data warehouse is more like long term memory in that it stores relatively permanent information.

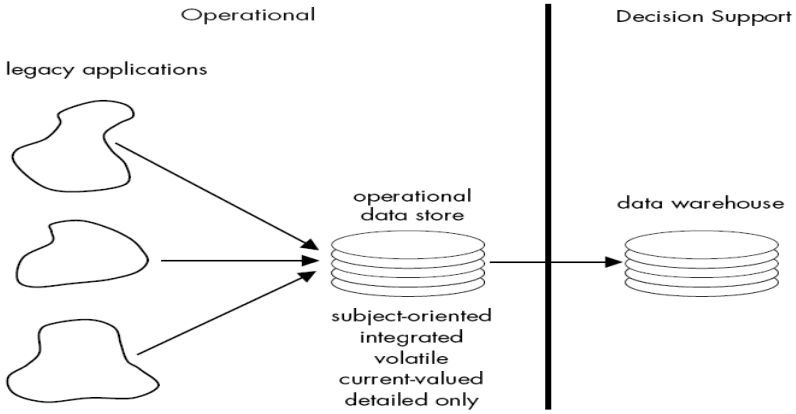

Figure 1. The Operational Data Store [Inmon, 1995]

In the early 1990s, the original ODS systems were developed as a reporting tool for administrative purposes. They were usually updated daily and provided reports about business transactions for that day, such as sales totals or orders filled. This type of system is now referred to as a Class III ODS. With changes in technology and business needs, the Class II ODS evolved to track more complex information such as product and location codes, and to update the database more frequently (perhaps hourly) to reflect changes. Class I ODS systems arose from the development of customer relationship management (CRM). In Class I systems, synchronous or near-synchronous updates are used to provide customers with consistently valid and organized information. Another version, the Class IV ODS, was recently developed with an added capacity for more interaction between the data warehouse or data mart and the ODS. [Oracle ODS]

The milestone for the work presented in this paper is the simple idea that we may use a special kind of organization of the information and this way to develop easy to use and compact ODS of Class I with facilities of DW with very high speed for response which enables *the real-time analytical processing* (RTAP). (The RTAP multithreaded processing engine needs to support extremely large volumes of data in real time. The analytics performed are composed of combinations of algorithmic, statistical and logical functions. [B-Jensen 2002])

The investigation presented in this paper is based on the fact that a specialized form of data warehouse is the corporate financial ledger. The segments of an account code serve the same purpose as the values on the axes of a data warehouse [Marshall, 1997]. In the same time, there exist a lot of account codes in a financial ledger and it is needed to operate with great complex of tables, descriptions, reports, etc. This leads to very complicated realizations which in the most cases are paid by more and more external memory for hundreds files as well as by growing quantity of processing operations.

In other hand, well-known considerable information complexes are offered by “SAP” (Germany), “Oracle” and “PeopleSoft” (USA), “Baan” (The Netherlands), etc., but the prices of such software are very high. This is serious problem for the middle and small enterprises, especially in Bulgaria, which will bankrupt if decide to implement so rich automated systems. Because of this the narrow versions of such software are offered at the market.

Unfortunately those versions are not as convenient as they are advertised and provoke many additional problems during the implementation process and exploitation.

Our approach is to build information complexes for information service of business accounting and decision making based on numbered information spaces [Markov, 2004a], which may support RTAP on the level of ODS Class I and this way to reduce the expenses for maintenance separate DW. This goal may be achieved using the FOI Archive Manager (ArM) ®.

FOI Archive Manager (ArM) ®

The FOI Archive Manager (ArM) ® is a tool for building numbered information spaces. ArM is based on the "Multi-Domain Information Model" (MDIM). It has been established more than twenty years ago. For a long period it has been used as a basis for organization of the information bases. The first publication which contains some details from MDIM is [Markov, 1984] but as a whole the model was presented in [Markov, 2004a]. There exist several realizations of FOI Archive Manager (ArM) ® for different hardware and/or software platforms. The newest ArM Version No.:9 for IBM PC developed using DELPHI for MS Windows XP is called ArM32.

Let remember the main possibilities of ArM32 [Markov, 2004b] using some definitions of MDIM.

Basic information element of MDIM is an arbitrary long string of machine codes (bytes). When it is necessary the string may be parceled out by lines. The length of the lines may be variable. In ArM32 the length of the string may vary from 0 (zero) up to 2^{30} (1G) bytes. There is no limit for the number of strings in an archive but theirs total length plus internal indexes could not exceed 4G bytes in a single file.

Let E_1 is a set of basic information elements: $E_1 = \{e_i \mid e_i \in E_1,\ i=1,\ldots, m_1\}$.

Let μ_1 is a function which defines a biunique correspondence between elements of the set E_1 and elements of the set C_1 of positive integer numbers: $C_1 = \{c_i \mid c_i \in N, i:=1,\ldots, m_1\}$, i.e. $\mu_1 : E_1 \leftrightarrow C_1$. The elements of C_1 are said to be number codes of the elements of E_1. The triple $\mathcal{S}_1 = (E_1, \mu_1, C_1)$ is said to be a *numbered information space of range 1*.

The triple $\mathcal{S}_2 = (\mathcal{E}_2, \mu_2, C_2)$ is said to be a *numbered information space of range 2* iff $\mathcal{E}_2$ is a set which elements are numbered information spaces of range 1 and μ_2 is a function which defines a biunique correspondence between elements of $\mathcal{E}_2$ and elements of the set C_2 of positive integer numbers: $C_2 = \{c_j \mid c_j \in N,\ j:=1,\ldots,m_2\}$, i.e. $\mu_2 : \mathcal{E}_2 \leftrightarrow C_2$.

The triple $\mathcal{S}_n = (\mathcal{E}_n, \mu_n, C_n)$ is said to be a *numbered information space of range n* iff $\mathcal{E}_n$ is a set which elements are information spaces of range *n-1* and μ_n is a function which defines a biunique correspondence between elements of $\mathcal{E}_n$ and elements of the set C_n of positive integer numbers: $C_n = \{c_k \mid c_j \in N,\ k:=1,\ldots,m_n\}$, i.e. $\mu_n : \mathcal{E}_n \leftrightarrow C_n$.

The sequence $A = (c_n, c_{n-1}, \ldots, c_1)$ where $c_i \in C_i$, i=1,…,n is called *multidimensional space address* of range n of a basic information element. Every space address of range m, m<n, may be extended to space address of range n by adding leading n-m zero codes. Every sequence of space addresses $A_1, A_2, \ldots, A_k$, where k is arbitrary positive number, is said to be a *space index*.

Every index may be considered as basic information element, i.e. as a string, and may be stored in a point of any information space. In such case it will have a multidimensional space address which may be pointed in the other indexes and, this way, we may build a hierarchy of indexes. So, every index which points only to indexes is called *metaindex*.

Let $G = \{S_i \mid i=1,...,m\}$ is a set of numbered information spaces.

Let $\tau=\{\nu_{ij} : S_i \rightarrow S_j \mid i=const, j=1,\ldots m\}$ is a set of mappings of one "main" numbered information space $S_i \subset G$, i=*const*, into the others $S_j \subset G$, j=1,...m, and, in particular, into itself. The couple: $Đ = (G, \tau)$ is said to be an "*aggregate*".

The ArM32 elements are organized in numbered information spaces with variable ranges. There is no limit for the ranges the spaces. Every element may be accessed by correspond multidimensional space address (coordinates) given via coordinate array of type cardinal. At the first place of this array the space range needs to be given. So, we have two main constructs of the physical organizations of ArM32 – numbered information spaces and elements.

The main ArM32 operations with basic information elements are: ArmRead (reading a part or a whole element); ArmWrite (writing a part or a whole element); ArmAppend (appending a string to an element); ArmInsert (inserting a string into an element); ArmCut (removing a part of an element); ArmReplace (replacing a part of an element); ArmDelete (deleting an element); ArmLength (returns the length of the element in bytes).

The ArM32 numbered information spaces are ordered and main operations within spaces take in account this order. So, from given space point (element or subspace) we may search the previous or next empty or non empty point (element or subspace). In is convenient to have operation for deleting the space as well as for count its nonempty elements or subspaces.

The ArM32 logical operations defined in the multi-domain information model are based on the classical logical operations - intersection, union and supplement, but these operations are not so trivial. Because of complexity of the structure of the spaces these operations have at least two principally different realizations based on codes of information spaces' elements and on contents of those elements.

The ArM32 information operations can be grouped into four sets corresponding to the main information structures: elements, spaces, aggregates, and indexes. Information operations are context depended and need special realizations for concrete purposes. Such well known operations are, for instance, transferring from one structure to another, information search, sorting, making reports, etc.

At the end there exist several operations which serve information exchange between ArM32 archives (files) such as copying and moving spaces from one to another archive.

ArM32 engine supports multithreaded concurrent access to the information base in real time.

Very important feature of ArM32 is possibility not to occupy disk space for empty structures (elements or spaces). Really, only non empty structures need to be saved on external memory.

Complex FOI®

Complex FOI® is an integrated software environment for economical information processing and business analysis. The main features of Complex FOI [Markov et al, 1994] are built on three levels, which correspond to the Pyramidal Information Model (PIM) presented in [Markov et al, 1993]. The levels of this model are "Strategy", "Analysis", and "Service". Every level contains three parts, which correspond to "Human Resources", "Materials", and "Finances" of the enterprise. It easy to see that there exist correspondence between PIM and ODS and DW.

The main set of concrete systems for information processing is included on "Service" level. They are aimed to service the operative work and control. For instance, there exist systems for service the enterprise financial tasks such as computing of salaries [Markov et al, 1996a], systems for managing different material stores using appropriate information access - by names or by numbers of goods [Markov et al, 1995a], systems for maintenance of fixed assets [Markov et al, 1996b], etc. An example of another class of service systems is one

for automated payment of consumption of water and other communal services in a town as well as the specialized service systems, such as one for computing the price of building of some architectural object. It is clear, *the legacy applications* of the enterprise are assumed to be on this level too.

All these systems are integrated with the upper level ("Analysis") via very convenient interface – the natural language standard accounting records which are the usual transaction form for accounting process. Furthermore, the information in Complex FOI is distributed in correspond numbered information spaces in accordance to usual every day financial accounting information structures. This make integration possible and automated information exchange is simple and comprehensible.

There is only one system on level "Analysis". It is an ODS with possibilities for accounting as well as for account analysis [Markov et al, 1995b]. This is the main tool for enterprise financial control and managing which support automated day-to-day operations (purchasing, banking etc), transactions access and modifying a few records at a time, application oriented database design, and metric: transactions/sec. The main structure of this level is the financial ledger - usually it is a numbered information space of range up to 10. Its subspaces represent accounting divisions, groups and accounts, as well as sub-accounts on several sub-levels. Every space may contain operational and historical data in the same time.

The main feature of the level "Strategy" is the decision support. All information from low levels can be used for supporting the processes of business decisions in the group of leaders of the enterprise. The functionality of this level covers the usual understanding of data warehouse but it is realized as distributed RTAP engine which support complex queries that access records with operational and/or historical data for trend analysis.

Because of special multidimensional organization, in Complex FOI the analytical pre-computation can be provided in real time during the operative work and its results (elements, spaces, aggregates, and indexes) can be stored in corresponded structures of the multidimensional hierarchical information base. So, in query response time, it is easy to process *multidimensional modeling* (for instance - compute total *sales* volume per *product* and *store*); *operating with dimensions and hierarchies* (for instance - roll-up: move up the hierarchy e.g. given total salaries per department, we can roll-up to get salaries per enterprise; drill-down: move down the hierarchy more fine-grained aggregation; pivoting: aggregate on selected dimensions usually 2 dims (cross-tabulation)); *comparisons* (for instance - this period vs. last period - show me the sales per store for this year and compare it to that of the previous year to identify discrepancies); *ranking and statistical profiles* (for instance – top N / bottom N - show me sales, profit and average call volume per day for my 10 most profitable salespeople); *custom consolidation* (for instance - market segments, ad hoc groups - show me an abbreviated income statement by quarter for the last four quarters for my northeast region operations); etc.

Conclusion

The approach to build information complexes for information service of business accounting and decision making based on numbered information spaces which may support RTAP on the level of ODS Class I and this way to reduce the expenses for maintenance separate DW has been presented in the paper. This goal may be achieved using the FOI Archive Manager (ArM) ® and "Multi-Domain Information Model" (MDIM). An application of presented approach named "Complex FOI" was outlined.

Acknowledgments

Author is indebted to Ilia Mitov and Krassimira Ivanova for support and collaboration. Due to theirs hard work the approach presented in this paper has been widely implemented in practice.

This work is a part of the project "ITHEA XXI", partially financed by the Consortium FOI Bulgaria.

Bibliography

[B-Jensen 2002] M.T. B-Jensen. High Tower Software's Tower View is the Odds-On Favorite of International Game Technology for Real-Time Data Management. Product Review published in DM Review Magazine July 2002 Issue. http://www.dmreview.com/article_sub.cfm?articleId=5403

[Devlin and Murphy, 1988] B.A. Devlin and P.T. Murphy. An Architecture for a Business and Information System. IBM Systems Journal. Volume 27, No. 1, 1988. http://www.research.ibm.com/journal/sj/271/ibmsj2701G.pdf

[Haisten, 2003] M. Haisten. The Real-Time Data Warehouse: The Next Stage in Data Warehouse Evolution http://www.damanconsulting.com/company/articles/dwrealtime.htm

[Inmon, 1991] W.H. Inmon. Building the Data Warehouse, QED/Wiley, 1991.

[Inmon, 1995] W.H. Inmon. The Operational Data Store. InfoDB February 1995 http://www.evaltech.com/wpapers/ODS2.pdf

[Markov 1984] K. Markov. A Multi-domain Access Method. // Proceedings of the International Conference on Computer Based Scientific Research. Plovdiv, 1984. pp. 558-563.

[Markov et al, 1993] K. Markov, K. Ivanova, I. Mitov, J. Ikonomov. Pyramidal Model of the Firm Information Activities. IJ ITA, 1993, Vol. 1, No. 2. (in Russian)

[Markov et al, 1994] K. Markov, K. Ivanova, I. Mitov. Basic concepts and main information structures of the Complex FOI. FOI-COMMERCE, Sofia, 1994. (in Bulgarian)

[Markov et al, 1995a] K. Markov, K. Ivanova, I. Mitov. Automated service of the storehouses. FOI-COMMERCE, Sofia, 1995. (in Bulgarian)

[Markov et al, 1995b] K. Markov, K. Ivanova, I. Mitov. Automated service of the financial accounting using System "ANALYSE". FOI-COMMERCE, Sofia, 1995. (in Bulgarian)

[Markov et al, 1996a] K. Markov, K. Ivanova, I. Mitov. Automated service of the accounting the staff and salaries FOI-COMMERCE, Sofia, 1996. (in Bulgarian)

[Markov et al, 1996b] K. Markov, K. Ivanova, I. Mitov. Automated service of the accounting of the fixed assets. FOI-COMMERCE, Sofia, 1996. (in Bulgarian)

[Markov, 2004a] K. Markov. *Multi-Domain Information Model*. Proceedings of the ITC&P-2004 - International Conference "Information Technologies and Communications & Programming", Varna. FOI-COMMERCE, 2004, pp. 79-88. Int. Journal "Information Theories and Applications", 2004, Vol. 11, No. 4, pp. 303-308

[Markov, 2004b] K. Markov. *Coordinate Based Physical Organization of Computer Representation of Information Spaces*. Proceedings of the Second International Conference "Information Research, Applications and Education" i.TECH 2004, Varna, Bulgaria. Sofia, FOI-COMMERCE – 2004, стр.163-172 (in Bulgarian).

[Marshall, 1997] Cr. Marshall. *Business Object Management Architecture*. OOPSLA'96 Workshop Business Object Design and Implementation II: Business Objects as Distributed Application Components - the enterprise solution? http://jeffsutherland.com/oopsla97/marshall.html

[Oracle ODS] - whatis.com - http://searchoracle.techtarget.com/sDefinition/0,,sid41_gci786730,00.htm

[Wikipedia, ODS] http://en.wikipedia.org/wiki/Operational_data_store

Authors' Information

Krassimir Markov – Institute of Mathematics and Informatics, BAS, e-mail: foi@nlcv.net