MULTI-DOMAIN INFORMATION MODEL

Krassimir Markov

Abstract: The Multi-Domain Information Model for organisation of the information bases is presented.

Keywords: Multi Domain Information Model, Information Bases, Knowledge Representation.

1. Introduction

The "Multi-Domain Information Model" (MDIM) has been established twenty years ago. For a long period it has been used as a basis for organisation of various information bases. The first publication containing some details of MDIM is [Markov, 1984] but the model has not been fully presented till now. In addition, over the years, the model has been extended with some new concepts like "information space", "metaindex", "polyindexation", etc. which we will introduce in this paper.

The present paper aims to present MDIM as a coherent whole.

2. Information Domain

Definition 1. ***Basic information element*** "e" of MDIM is an arbitrary long string of indivisible information fragments (bytes in the version for IBM PC; symbols; etc.). ■

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 a set C_1 of natural numbers: $C_1 = \{c_i \mid c_i \in N, i:=1,\ldots, m_1\}$, i.e.

$$\mu_1 : E_1 \leftrightarrow C_1$$

Definition 2. The elements of C_1 are said to be ***co-ordinates*** of the elements of E_1. ■

Definition 3. The triple $S_1 = (E_1, \mu_1, C_1)$ is said to be an ***information domain*** of range one ***(one-dimensional information space)*** . ■

Remark: In the previous publications, the information domain S_1 was denoted by D and the co-ordinates c_i were called "codes" of the corresponded information elements.

3. Information Spaces

Definition 4. The triple $S_n = (E_n, \mu_n, C_n)$, $n \geq 2$, is said to be an (***complex*** or ***multi-domain***) ***information space*** of range n iff 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 E_n and elements of the set C_n of natural numbers (co-ordinates of range n):

$$C_n = \{c_k \mid c_k \in N, k:=1,\ldots,m_n\}, \text{ i.e.}$$

$$\mu_n : E_n \leftrightarrow C_n \ ■$$

Definition 5. Every basic information element "e" is considered as an ***information space*** S_0 of range 0. ■

It is clear that the information space $S_0 = (E_0, \mu_0, C_0)$, is constructed in the same manner as all others:

- the indivisible information fragments (bytes) b_i, $i=1,\ldots,m_0$ are considered as elements of E_0,
- the position p_i (natural number) of b_i in the string e is considered as co-ordinate of b_i, i.e. $C_0 = \{p_k \mid p_k \in N, k:=1,\ldots,m_0\}$,
- function μ_0 is defined by the physical order of b_i in e and we have: $\mu_0 : E_0 \leftrightarrow C_0$

When it is necessary the string S_0 may be considered as a set of ***sub-elements (sub-strings)*** which may contain one or more indivisible information fragments (bytes). The number and length of the sub-elements may be variable. This option is very helpful but it closely depends on the concrete realizations and is considered as none standard characteristic of MDIM.

Definition 6. The information space S_n of range n is called ***information base*** of range n. ■

Usually, the concept information base without indication of the range is used as generalized concept to denote all information spaces in use during given time period.

4. Indexes

Definition 7. The sequence $A = (c_n, c_{n-1}, \ldots, c_1)$ where $c_i \in C_i$, $i=1,\ldots,n$, is called ***space address*** of range n of an 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 co-ordinates.

Definition 8. Every sequence of space addresses $A_1, A_2, \ldots, A_k$, where k is arbitrary natural number, is said to be an (address) ***index***. ■

Definition 9. Every ordered subset I_i, $I_i \subset C_i \subset N$ of co-ordinates (i – arbitrary natural number) is said to be a (space) ***index***. ■

It is clear that space index is a kind of address index.

5. Polyindexation

Every index may be considered as basic information element (i.e. as a string) and may be stored in a point of any information domain. In such case, it will have a space address which may be pointed again.

Definition 10. Every ***index*** which point only to indexes is said to be a ***metaindex***. ■

Every metaindex may be considered as basic information element (i.e. as a string) and may be stored in a point of any information domain, too. So, it will have a space address which may be pointed again, etc. This way, we may build a hierarchy of metaindexes.

Definition 11. The approach of representing the interconnections between elements of the information domains as well as between spaces using hierarchies of metaindexes is called ***polyindexation***. ■

6. Aggregates

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

Let $\tau=\{\nu_{ij} \mid \nu_{ij} : S_i \rightarrow S_j,\ i=const,\ j=1,\ldots m\}$ is a set of mappings of one "main" information space $S_i \subset G$, $i=const$, into the others $S_j \subset G$, $j=1,\ldots m$, and, in particular, into itself.

Definition 12. The couple $Đ = (G, \tau)$ is said to be an "***aggregate***". ■

It is clear we can build m aggregates using the set *G* because every information space $S_i \subset G$, $j=1,\ldots m$, may be chosen to be a main information space.

Remark: In the previous publications, the aggregate Đ was called generalized domain.

7. Operations in MDIM

After defining the information structures we need to present the operations which are admissible in the model.

It is clear; the operations are closely connected to the defined structures. So, we have operations with:
- basic information elements (**BIE**)
- spaces
- indexes
- metaindexes

In MDIM, we assume that all information elements of all information spaces exist. If for any S_i : $E_i=\varnothing \wedge C_i=\varnothing$, than it is called *empty.* Usually, most of the information elements and spaces are empty. This is very important for practical realizations.

7.1. Operation with basic information elements

Because of the rule for existing of the all structures given above we have need of only two operations:
- updating the BIE
- getting the value of BIE

For both types of operations we need two service operations:
- getting length of BIE
- positioning in the BIE

Updating, or simply – writing the element, has several modifications with obvious meaning:
- writing of a BIE as a whole
- appending a BIE
- inserting in a BIE
- cutting a part of BIE
- replacing a part of BIE
- deleting a BIE

The operation for getting the value of BIE is only one – **Read** a portion from BIE starting from given position. We may receive the whole BIE if the starting position is the beginning of BIE and the length of the portion is equal to the BIE length.

7.2. Operation with spaces

With a single space we may do only one operation – clearing (deleting) the space, i.e. replacing the all BIE of the space with ∅. After this operation the BIE of the space will have zero length.

With two spaces we may provide two operations with two modifications every:
- copying the first space in the second
- moving the first space in the second

The modifications concern the type of processing the BIE of the recipient space. We may have:
- copy with clear
- move with clear
- copy with merge
- move with merge

The “clear” modifications first clear the recipient space and after that provide copy or move operation.

The merge modifications may have two types of processing:
- destructive
- constructive

The *destructive merging* may be “conservative” or “alternative”. In the conservative approach the recipient space BIE remain in the result if it is with none zero length. In the other approach – the donor space BIE remain in the result.

In the *constructive merging* the result is any composition of the corresponded BIE of the two spaces.

Of course, the move operation deletes the donor space after the operation.

7.3. Operation with indexes and metaindexes

The indexes are the main approach for describing the interconnections between the BIE.

At the first place, we may operate with and in the indexes C_i, $i=1,2,\ldots,n$ of the spaces. We may receive the co-ordinates of the next or previous empty or none empty elements of the space starting from any given co-ordinate. The possibility to count the number of none empty elements is useful for practical realisations.

The operations with indexes and metaindexes may be classified in two main types:
- logical operations
- information operations

The first type is content independent operations based on usual logical operations between sets. The difference from usual sets is that the information spaces are build by interconnection between two main sets:
- set of co-ordinates
- set of information elements

The logical operations defined in the MDIM 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 information spaces these operations have at least two principally different realizations based on:
- co-ordinates
- information elements

The operations based on co-ordinates are determined by the existence of the corresponding space information elements. So, the values of the co-ordinates of the existing information elements determine the operations.

In the other case, the values of the BIE determine the logical operations.

In both cases the result of the logical operations is any index, respectively – metaindex.

The information operations are context depended and need special realizations for concrete purposes.

The main information operation is creating the indexes and metaindexes. This may be very complicated processes and could not be given in advance. The main purpose of the MDIM is to give up possibility for access to the practically unlimited information space and easy approach for building interconnection between its elements. The goal of the concrete applications is to build tools for creating and operating with the indexes and metaindexes and to implement these tools in the realization of user requested systems.

For instance such tools may realize the transfer from one structure to another, information search, sorting, making reports, more complicated information processing, etc.

The information operations can be grouped into four sets corresponding to the main information structures:
- basic information elements
- information domains
- information spaces
- index or metaindex structures

8. Discussion

Usually, the submission of any new information model needs to be discussed in connection to already existing models and theories. We have no place in this paper to analyze all known models. Because of this we will point only two of them we assume as more important:
- theory of the named sets [Burgin, 1984]
- relation model of Codd [Codd, 1970]

Our proposition is that the MDIM has the same and more modeling possibilities than named sets and relation model.

8.1. Theory of the named sets

For our further discussion we need some information from [Burgin and Gladun, 1989] .

If α is a relation of X with Y i.e. $\alpha\subseteq X \times Y$, $A\subseteq X$, $B\subseteq Y$ then

$\alpha(A)=\{y|\exists x\in A\ ((x,y)\in\alpha)\}$, $\alpha^{-1}(B)=\{x|\exists y\in B\ ((x,y)\in\alpha)\}$,

$\alpha|_{(A,B)}=\{(x,y)\in\alpha \mid x\in A\ y\in B\}$.

The empty set is denoted by $\varnothing$.

Definition B&G-1. A named by $\mathcal{M}$ set (an N-set) is a triple

$$\mathcal{X}=(X,\alpha,I)$$

where X is a set from some fixed class of sets and is called the support set of the named set $\mathcal{X}$. I is a set from some (may be another) fixed class of sets and is called the set of names of the named set $\mathcal{X}$. $\alpha:X\rightarrow Y$ is a map or a correspondence (a relation) from X to I and belongs to a given class of relations $\mathcal{M}$.

A name $a\in I$ is called empty if $\alpha^{-1}(B) = \varnothing$.

Named sets as special cases include: usual sets, fuzzy sets, multisets, enumerations, sequences (countable as well as uncountable), etc. A lot of examples of named sets we may find in linguistics studying semantic aspects that are connected with applying different elements of a language (words, phrases, texts) with their meaning. [Burgin and Gladun, 1989, p.121-122].

The Theory of named sets (TNS) has been established about 1982 [Burgin 1984]. Independently, the MDIM has been developed in the period from 1980-1982 and its first publication was [Markov 1984] .

We may find many common ideas in the two approaches. Here we will point at two main characteristic of MDIM.

Proposition 1. Every information space is a named set.

Proof: By definition, the set E is the support set, C is the set of the names and μ is a function of naming.

Proposition 2. Every named set may be represented by an aggregate.

Proof: It is simple to build the named set by an aggregate using:

- two information spaces: one for the names and one for the elements of the named set,
- aggregation mapping which is identical to the named set mapping. ■

This way all possibilities of the TNS exist in the MDIM. In other hand, the polyindexation does not exist as theoretical base in the TNS. The aggregates are more general constructs than named sets. At the end, MDIM is designed to support practical realizations whereas the TNS is a theoretic logical construction for reasoning.

The conclusion is that the MDIM has the same and more modeling possibilities than named sets.

8.2. Relation model of Codd

The Cood's Relation theory [Codd 1970] is so popular that we do not need to explain it here. For our discussion we will proof one very important proposition.

Proposition 3. The relation in the sense of the model of Codd may be represented by an aggregate.

Proof: It is easy to see that if the aggregation mappings of the generalized domain are one-one mappings it will be relation in the sense of the model of Codd. ■

In the same time many possibilities of MDIM could not be represented by the relation model or this is very expensive work. Especially, the polyindexation could not be represented by relations. The representation of the information spaces of range more than three is very expensive for the practical realizations.

So, we may say that MDIM is more universal and convenient for practical realizations than the relation model.

9. Conclusion

The Multi-Domain Information Model (MDIM) for organisation of the information bases has been presented in this paper. The information structures and operations of MDIM have been presented.

The correspondences between MDIM and named sets (Propositions 1 and 2) as well as the relation model (Proposition 3) were shown. Our conclusion is that the MDIM has the same and more modeling possibilities than named sets and relation model.

At the end, we need to discuss some more general conclusions.

We consider *the real world* as a space of *entities.* The entities are built by other entities, connected with *relationships*. The entities and relationships between them form the internal *structure* of the entity they build. To create the entity of a certain structural level of the world, it is necessary to have:

– the entities of the lower structural level;
– establishing of the forming relationship.

The entity can dialectically be considered as a relationship between its entities of all internal structural levels. [Markov et al 2003]. Every entity may be considered as relationship between "*atoms*" which are entities on the lowest structural level where there exists another relationship and so on.

This way we may distinguish three types of relationships: explicit (forming relationships), implicit (forming relationships at lower levels) and mixed (in case we distinguish the relationships from lower levels as elements of the forming relationship of given level).

In our model, the information atoms are the basic information elements. It is easy to see that they may contain more complex structures such as domains, spaces, generalized domains, indexes, metaindexes, etc.

This means: *the complexity of the real word can be reflected by the complexity of the MDIM realizations.*

This inference gives us one very fruitful idea – to use MDIM as a model for memory structuring in intelligent systems [Gladun 2003].

Finally, we need to point out that for more than twenty years the MDIM realizations have shown the power of this model. The concrete systems based on MDIM information bases now work on more than one thousand installations all over the Bulgaria.

Acknowledgments

Author is indebted to Ms. Krassimira Ivanova and Mr. Ilia Mitov for support and collaboration during the latest 15 years. Due to theirs hard work the model presented in this paper has been implemented in practice.

Special thanks to Mr. Boicho Kokinov for fruitful remarks and advices.

Author owes a dept of gratitude to Prof. S.S. Lavrov for his important support and recommendations given in the beginning of work presented in this paper.

Bibliography

[Burgin 1984] Burgin M.S. Named Sets and Information representation. VII Allunion conference on mathematical logic, Novosibirsk, 1984. (in Russian)

[Burgin and Gladun, 1989] Mathematical Foundations of Semantic Networks Theory. Lecture Notes in Computer Science, No. 364, pp. 117-135.

[Codd 1970] Codd E.F. A Relation Model of Data for Large Shared Data Banks. CACM 13, No.6 (June 1970).

[Gladun 2003] Gladun V. Intelligent Systems Memory Structuring // International Journal: Information Theories and Applications.-2003, V.10, №1.

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

[Markov et al 2003] Kr.Markov, Kr.Ivanova, I.Mitov. General Information Theory. Basic Formulations. // FOI-COMMERCE, Sofia, 2003.

Author Information

Krassimir Markov – Institute of Mathematics and Informatics, BAS, P.O. Box: 775, Sofia-1090, Bulgaria; e-mail: foi@nlcv.net