Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

Local and Global Parsing with Functional X-bar Theory and SCD Linguistic Strategy (II.)

Part II. Functional Generative Capacity, SCD Marker Classes, and Local / Global Segmentation / Parsing Algorithms

Neculai Curteanu

Abstract

This paper surveys latest developments of SCD (Segmentation-Cohesion-Dependency) linguistic strategy, with its basic components: FX-bar theory with local and (two extensions to) global structures, the hierarchy graph of SCD marker classes, and improved versions of SCD algorithms for segmentation and parsing of local and global text structures. Briefly, **Part I** brings theoretical support (predicational feature and semantic diathesis) for handing down the predication from syntactic to lexical level, introduces the new local / global FX-bar schemes (graphs) for clause-level and discourse-level, the (global extension of) dependency graph for SCD marker classes, the problem of (direct and inverse) local FX-bar projection of the verbal group (verbal complex), and the FX-bar global projections, with the special case of sub-clausal discourse segments. **Part II** discusses the implications of the functional generativity concept for local and global markers, with a novel understanding on the taxonomy of text parsing algorithms, specifies the SCD marker classes, both at clause and discourse level, and presents (variants of) SCD local and global segmentation / parsing algorithms, along with their latest running results.

Notice. This is a paper in two parts, preserving a unitary numbering of the sections, and the unitary set and system of references along both parts.

6 Functional Generativity of Local and Global Markers

The aim of Part II of the paper is to use the results of Part I for designing improved theoretical mechanisms and text segmentation / parsing techniques based on proper linguistic *marker classes* incorporated into (D)FX-bar theory and SCD linguistic strategy. We discuss the development of *segmentation* and (*dependency establishing*) *parsing algorithms*, especially for global, clause-level and discourse-level text structures, using the newly defined notions of *strong* (for lexical-level phrase markers) and *weak* (for class-level phrase markers) *functional generative capacity* [16].

A whole *class of segmentation / parsing algorithms* is described within the SCD strategy, by refining the SCD marker classes (and hierarchy) towards the lexical level of the contained markers as (weak vs. strong) *functional generativity* (using a generalization of lexical marker database in [10], both at the M3 and M4 levels of the SCD marker hierarchy [17]).

A comparison between (versions of) *SCD segmentation algorithms* [9], [13] and *Marcu*'s *segmentation algorithm* [25], [26] was realized in [14], [17], and two directions to approach the segmentation and parsing processes at the *global text structures* are explored: (G1) clause-level (syntactic) parsing, and (G2) discourse (rhetoric relation) parsing.

6.1 Functional Generativity for Classifying Parsing Algorithms

In [16], we defined the *functional generative capacity* for *phrase markers*, such as those in SCD or in Marcu's segmentation and / or parsing algorithms, as follows: when applied at *lexical level*, the phrase markers provide *strong functional generativity*; when applied at (marker) *class* *level*, they provide *weak functional generativity.* An observation is necessary: while a structure built from lexical preterminals N, V, and/or A is *strongly generated* and a sequence of lexical categories (words) is *weakly generated* in the classical sense of *categorial generativity* [22], [27], we consider *lexical markers* of *strong functional generativy* since the sharper *functional meaning* of *a lexical marker* entails (is stronger than) the *functional meaning* of a whole *class of markers.*

For example, the categorial strongly generated structure Det A N "implies" (its less informative meaning subsumes the one of) the weakly generated sequence *the beautiful flower*, and also the functional strongly generated *and*(XG_1, XG_2) implies (in the partial, reverse ordering of semantic meanings) the functional weakly generated *conjunction*(XG_1, XG_2), since the *more informative* meaning of the lexical conjunction "*and*" subsumes the inherently *less informative* meaning of the class-depending *conjunction* marker. To further support the proposed definition of functional generativity, we observe that the 'strong' lexical marker "*dacă*" (*if*) entails the 'weaker'-level phrase-marker *conjunction* (a marker class comprising several lexical conjunctions), since the information it holds is richer (e.g., in the sense of subordinate type determination) than the information held by the less informative *conjunction* class (which can only determine the subordinate, but not its type).

Thus, these definitions preserve the general entailment "*strong* implies *weak generativity*", but with the essential *difference* that while "strong categorial implies weak lexical generativity", we need a "strong *lexical marker*" to be *functionally* applied to an utterance to entail a "weak *marker class*" that is applied to the same utterance similarly.

Related to the manner in which markers or marker classes are applied to local or global text structures, the concept of *functional generativity* has immediate consequences on the FX-bar *projections* of local and global text structures, hence within the segmentation / parsing algorithms whose task is to handle the recognition / generation of these entities efficiently.

For instance, using clause markers at *lexical* level in a segmentation task entails a weaker *categorial* generativity and a higher complexity of the algorithms. The same task, worked with *classes* of clause markers, increases the expressional generativity and decreases the algorithm complexity.

At discourse level, and especially for the parsing (dependency-establishing) task, it is more profitably to use the *lexical* markers in order to obtain a stronger *functional* generativity. The usage of marker classes at this level, either for segmentation or parsing task, involves a certain degree of generality-ambiguity in determining the discourse units and rhetorical relations, resolved by the use of markers at lexical level (see [25] and the more general parsing tables proposed in Section 8.2).

In [17] we analyzed and classified several classes of local and global segmentation / parsing algorithms, based on such criteria as: **(a) categorial generative capacity** (or categorial generativity); the *strong* and *weak generativity* of major (N, V, A) preterminal and lexical categories [22], [27]; **(b) functional generative capacity** (or functional generativity); the *new concept*, introduced in [16], of *strong* and *weak functional generativity* of lexical and, respectively, classes of clause / discourse markers, as the *functional* counterpart to the corresponding notions of *categorial generative capacity* defined for major lexical categories [22], [27]; **(c) processing task**: *segmentation* or (dependency-establishing) *parsing*; **(d) the output structure targeted**: *clause* level or *discourse segment* (clause-like) level, *i.e.* local and/or global structures as outcome.

Finally, to notice that our concept of *functional generativity* differs essentially from what in [27; p.140-141] is called *derivational generative capacity*, a notion that is related to the derivation trees of functor-argument clause-type in TAGs (*e.g.* [21] and A. Joshi's previous papers on this well-known grammatical formalism). We see *derivational generativity* as a generalized form of strong categorial generativity applied to (derivation) trees instead of simpler categories (or lexical preterminals) N, V, A. The essential distinction between *categorial* (with the more general *derivational*) *generativity* and our concept of *functional generativity* is that they represent different components of the mathematical function object $f(X)$: the first concept corresponds to the argument X, while the second concept corresponds to the function name and role f.

6.2 SCD Variants for Local / Global Segmentation / Parsing

SCD algorithms are discussed at segmentation and parsing level. The segmentation is realized using inter-clausal marker classes (M3 class, see §7), to obtain finite clauses. Clause parsing is realized using lexical markers (of M3 class, described in a database which contains the information from each marker of the class, and a set of correction rules for the partial trees obtained in the first step of dependency determination).

Discourse segmentation and establishing the rhetorical relations between the discourse segments, as well as discourse tree building, is achieved using lexical markers from the M4 (discourse-level) class.

(1) In (SCD) *clause-level segmentation / parsing*, the following steps can be distinguished: SCD automated annotation (see §7.1.d and §7.2), clause segmentation, resolving the dependencies between clauses based on marker classes (superordinate – subordinate clause type) or on lexical markers (specifying the type of the subordination). **(2)** *Discourse-level parsing* can be done using marker classes of clause-like markers and structures [25], establishing of the discourse segments based on the lexical markers from the M4 level, determination of the rhetorical relations between the obtained discourse segments, based on the lexical discourse markers [25], [26], (extending lexical markers from M3 to M4-level lexical markers, Tables 8.1-8.2, §8).

Revealing the *discourse segments* is realized using clause segmentation and the lexical markers from the M3 level, while estimating the rhetorical relations between discourse segments is based on the discursive interpretation of the lexical discourse markers.

7 SCD Marker Classes and Algorithms

The SCD parsing strategy extends from *three* to *four* the representation level of marker classes, providing functions for setting the boundaries of the main syntactic structures, XG (X = N, V, A), clause, inter-clause, and discourse elementary unit (segment) (see Fig. 2.1-2.2, Part I).

The *first marker class*, denoted **M0** (or M00), is applied to the *word dictionary form*, is represented by the *functional role* of *morpho-grammatical inflection*, and corresponds to the lexical level of each word.

a) M1 Class = {*markers delimiting (introducing) XG structures*}.

The M1 class of markers *consists of* X1-*level markers*, (X = N, V, A), *i.e.* markers to be applied to the X1-*level syntactic constructions* (also denoted XG, and called X *groups*). These syntactic constructions consist basically of a *semantic head* (N, V, A category) surrounded by *modifiers* (adjectives or adverbs), and/or by (generalized) *quantifiers* (this includes determiners, negation, etc), *modal modifiers* of level 1-bar (*e.g.* the A1 adverb "*poate*" (*maybe*)) or 2-bar (*e.g.* the V2 modal verb "*a putea*"(*can-may*)), and/or functionally marked by *pre-positions* (in English, French, Romanian) or post-positions (in German or English) that express the *case* (for N), *aspect* or *meaning* (for V), etc. The main elements of an XG structure provide also the marker subclasses of M1. It is important to mention a certain linguistic (but not linear) *order* of these components of the XG, coming from the *distance* of these elements to the left or to the right of their semantic head, *e.g.* for the noun: the closest to the head are the modifiers, followed by quantifiers, the farthest to the head being the pre- or post-position functional particles. For VG (or verbal complex), the *predicational marking* and FX-bar *projections* are by far more elaborated operators and operations.

M1 can be split into subclasses of markers that are useful in delimiting the XG (X1) substructures, X = N, A, accordingly to criteria such as the above-mentioned *distance* to the X0 semantic head of the surrounding elements, a head which ultimately is always an (overt or covert) objectual common noun, proper noun, or personalized (no-named) noun.

M11 = {M11N, M11P}

M11N = {the occurrence of an objectual, non-predicational common noun, or of a proper noun}

M11P = {the occurrence of an accentuated or non-accentuated pronominal form}

M12 = {M12N, M12V}

M12N = {the occurrence of a noun modifier (adjective, pronominal adjective)}

M12V = {the occurrence of a verb modifier (adverb)}

M13 = {the occurrence of a (generalized) quantifier}

M14 = {pre-positions or post-positions expressing the case (for N), aspect or meaning (for V), etc.}

b) M2 Class = **{***markers that introduce a (finite or non-finite) clause, or a syntactic category group phrase with the semantic head N, V, A***}**. XG syntactic compound, (X = N, V, A), may be assimilated with a (degenerated) non-finite clause for X = N, A. M2 is split into the following subclasses (in decreasing order of priority when introducing dependency relations):

M25 = {markers that introduce the *relative clause*}.

The explanation for M25 *tag* (and its place in the dependency graph of Fig. 2.3, Part I) is that the relative clause represents the most complex syntactic compound playing the role of a modifier, to be applied to its NG head argument. The relative clause is an A2-level modifier in the FX-bar scheme, *i.e.* a modifier of 2-bar (clause) level of FX-bar projection.

M24 = {the occurrence of a *finite verbal group* (FVG) or, simply, the occurrence of the FINIte feature value assigned to a verb, introducing a *finite* VG, thus *clause*}.

The whole VG may inherit the FINIte feature value if its (predicational V) *semantic head*, or its (auxiliary V) *syntactic head* for complex tenses, bears this feature value.

M23 = {the occurrence of PREDF = PROCess non-lexical feature value assigned to any of the major categories N, V, A (since the lexicon encoding), thus introducing a clause}.

M22 = {the occurrence of the TENSe=NonFINIte feature value assigned to the category V}. See [2], [28], [30], [23], [18] for various analyses of the *verbal complex*, *i.e.* VG in FX-bar terms.

M25, M24, M23 and M22 marker classes introduce X2-level structures, *viz.* finite or non-finite clauses, made up of an X1 phrase (or XG group, X = N, V, A) that represents the semantic (either finite, non-finite or predicational) head of the X2-level structure, *followed* by the corresponding NG-type (including prepositional-headed) arguments and/or adjuncts within the same clause. Some of the arguments, such as the classical case of the *grammatical subject* (or *all* the arguments, as it is possible in German), may *precede* the X1-type semantic head of the clause to which they belong [12; p.73]. Note that there exists a *systemic* (*canonical*) *order* [37] of the clause compounds, or '*actants*' (Arguments and Adjuncts) in a (finite or non-finite) clause: ACT(or), PAT(ient), ADDR(essee), ORIG(ine), LOC(ation), etc. The systemic order of the arguments within a clause is (a *theta-order*) specific to each NL, being obtained as a result of a very careful linguistic and statistic research.

M21 = {markers that introduce JOIN-type relations, *i.e. conjunctions* of the type "*and*", "*or*", "*as_well_ as*", "*together_with*"}.

M20 = {COMMA}.

Classes M21 and M20 comprise markers with an important degree of ambiguity since they may introduce any structure of type X1 (XG groups, X = N, V, A) or X2 (finite or nonfinite clauses).

c) M3 Class = {*inter-clausal (discourse) markers***}**.

The M3 class markers are functions, or relations (when correlated), having as arguments two or several finite (some of them may be non-finite) clauses. These markers are what [25], [16], [17] and other approaches are calling inter-clause, 'clause-like', or discourse markers, and apply to the X2 = CL1 syntactic projections of clause-type in the FX-bar scheme(s).

M3 may be partitioned into the following subclasses (in decreasing order of priority when introducing dependency relations):

M34 = {punctuation (pragmatic) markers that separate clauses, *e.g.* ".","!","?"...}

M33 = {inter-clausal / discourse markers that introduce (unambiguous) *strict super-ordination* clausal dependency }. *Strict super-ordination* means the effective *raising* of (*at least*) *one level* of clausal dependency, and is represented by such markers as "*then*", "*else*", etc.

M32 = {inter-clausal / discourse markers that introduce *super-ordination* clausal dependency, including *punctuation marks* such as *colon*, *semi-colon*, *closed parenthesis, second-paired dash*, etc.}. *Super-ordination* means *raising* one (or several) level(s) of clausal dependency, or remaining on the same dependency level within a *coordination*-type dependency. Typical examples of markers from M32 class are: "*but*", "*therefore* (*thus*)", "*even*", "*equally_*(*to*)", "*in_comparison_with* (*compared_to*)", etc.

M31 = {inter-clausal (discourse) markers introducing one (or several) *sub-ordination* clausal dependency level(s), including *punctuation marks* such as *open parenthesis, first-paired dash*, etc.}. This is a large class of discourse markers bearing various types of relations between clauses: logical, syntactic (of several types), semantic, pragmatic, etc.

As mentioned above, each of the M33, M32, and M31 classes may, at their turn, be partitioned into sub-subclasses that contain relational-type markers (expressed by correlation) as relations on clauses, or as functions (with at least two arguments) on clauses.

d) M4 Class = {*discourse markers, which determine the rhetorical relations that can be established between discourse segments***}**.

The *elementary discourse units* (EDU*s*, or *segments*) are identical to clauses in most of the cases, but exceptions can be found, that is, some segments can be constituted of several clauses and, remarkably, sub-clausal segments (non-finite clauses or groups different from the verbal one, but which still contain a covert predication) can also exist (see §4.2, Part I). Some of the *discourse markers* are also M3 level markers, i.e. they also have an inter-clausal relation determination role.

The same rhetorical marker can introduce several types of rhetorical relations, the disambiguisation being resolved by additional methods (statistical results, anaphora resolution, and lexical chains).

The M4 level markers can be classified accordingly to several criteria:

i) *According to the type of rhetorical relation introduced*;

The M4 level markers determine certain types of rhetorical relations, similar to those described in [24]. The number of these relations is approximately 25 in [24], the list being extended in [25]. The M4-level markers can be classified by the type of the established relations, as follows:

Antithesis: *dar, însă, cu toate acestea, ci, dacă nu, numai nu*;
Concession: *deşi, cu toate că, cel puţin*;
Detail: *în acelaşi mod, la fel cum, cât despre*;
Duration: *niciodată, încă o dată, după ce, în tot acest timp*;
Elaboration: *pe deasupra, şi încă, în acea perioadă, la care*;
Justify: *dar şi, însă, de asemenea*;
Purpose: *pentru că, ca să, fiindcă, cu scopul*;

ii). *According to the type of the units introduced*

The discourse markers can be classified after the type of the discourse segments they introduce, in: markers that introduce nucleus-type discourse units (*dar, însă, atunci, altfel, în primul rând*) and discourse markers that introduce satellite units (*chiar dacă, cu toate că, din cauza, dacă*).

iii). *According to the complexity of the introduced relations*;

Applying these criteria, the M4-level markers can introduce: **(a)** binary relations – most of the relations between the discourse segments are binary; for example, the *Elaboration* relation, introduced by markers like "*în plus*", "*pe lângă acestea*", "*de asemenea*", "*în afară de acestea*"; **(b)** n-ary relations, ($n >= 3$).

There are some rhetorical relations which can have as arguments more than two discourse segments. Among these are the *Joint* relation (introduced by markers like "*şi*", "*sau*"), the *Contrast* relation (introduced by markers like "*dimpotrivă*", "*deşi*", "*ca şi cum*"), the *List* or *Sequence* relations.

An important aspect that has to be considered in establishing the rhetorical relations between discourse units is *marker correlation*. This is also used to establish dependencies between clauses, but at the discourse level it is essential if we want to build the discourse trees correctly.

An obvious example of *correlation* at the M4 level is the 3-uple (*dacă* S1) – (*atunci* S2) – (*altfel* S3) (*if-then-else* relation). In this case, the tree corresponding to the sentence must be built taking into consideration not only the relations between the S1, S2 and S3 segments, established on the markers, but also the relations between the markers, relations that determine the structure of the tree built from the discourse units.

Fig. 2.3 (Section 2.3, Part I) presents the hyper-graph hierarchy of the SCD marker classes. This hierarchy is considered to be valid for Romanian. Certain modifications could be necessary from a NL to another. When we situate within restrained field of Indo-European languages (such as French, English, German, possibly Russian), one can appreciate that the proposed marker classes and structures (in Fig. 2.3) remain very similar, possibly submitted to slight modifications from a particular NL to another.

7.1 SCD Segmentation / Parsing Algorithms

The SCD segmentation algorithm presented here has a *breadth-first* (or *sequential-linear*) processing form, using as input a morphologically tagged text, and obtaining the finite (and non-finite) clauses, and the XG-structures (*e.g.* [9], [13] for a *depth-first*, *recursive* version of the SCD segmentation-parsing algorithm). The XML standard is used for data representation, and the implementation of the algorithm is made in Java.

Steps of the SCD algorithm:

a) Marker recognition for local text structures (M1, M2, M3 classes)

This step is realized automatically, except for the *predicationality feature* PREDF := PROCess, which cannot be assigned in the same manner, and has to be done manually. A sample set of rules used to realize the automated SCD *annotation* is presented in [14; p.75].

M11N markers are associated to nouns; M11P markers to pronouns, M12N is being associated to adjectives and pronominal adjectives; M12V markers to adverbs; M13 subclass contains the quantifiers and negation, and M14 subclass contains prepositions and post-positions.

M20 marker is represented by comma, M21 is *"şi"* (*and*) coordinating conjunction; subclass M22 represents an occurrence of the NonFINite feature associated to verbs, M23 marker is associated to a V, N, or A that bears the *predicationality* feature PREDF := PROC, M24 marker is associated to the lexical elements tagged as verbs at a finite mode (TENSe = FINIte), and M25 subclass is assigned to relative pronouns.

M3 class markers are associated at this stage to lexical elements that have coordination, sub-ordination or super-ordination role, as well as sentence boundaries. The 'real' M3 markers, which can contain multiple lexical elements, as "*aşa cum*" (*such as*), "*chiar dacă*" (*even whether*) etc. are recognized in a subsequent stage.

b) Recognition and Structure of the Verbal Group Kernel [FVGIN tag]

The verbal group (VG), as XG structure in the BAR = 1 projection level of the FX-bar scheme, contains a *semantic head* verb, *around* which one can find pronouns (only in unaccentuated forms, *i.e.* clitics), special adverbs, auxiliaries, modal verbs (or adverbs), negation. VG is also better known under the label of *verbal complex* (see [28], [29], [2]), and constitutes what is traditionally called *verbal predicate* for the classical clause (proposition). The VG Kernel (VGK) was initially introduced in [17, p.175] (under the name of *default verbal kernel*), and represents a basic substructure in the VG parsing. The *typical difference* between VG and VGK is that VGK is missing the *proper adverb* of VG (that may syntactically commute with VGK to accomplish the VG).

c) Recognition of the inter-clause markers

M3 (M3n, $n = 1,\ldots,4$) and M25 class markers are recognized using the database described below (Table 8.1).

d) Text segmentation into finite clauses

Using the outcome of the precedent steps, the algorithm determines the clauses of every sentence in the text, based on the marker classes. A pseudo-code description of the algorithm is given below:

```
SCD Tagging
Input: morphological tagged text
```

Output: SCD tagged text
1.1 Recognition of the M1 class and M22 markers
Recognition of the FVGIN structure (M24 marker)
Recognition of the MRK structure (M3 class, M20, M21 and M25 markers)
Manual annotation of the M23 marker (predicational feature)
SCD meta-algorithms
2.1 Segmentation Algorithm
Input: SCD tagged text
Output: The syntactic structures like: finite-clauses, nonfinite-clauses, noun groups and verbal groups.
2.1.1 Finite-clause recognition
Input: sentence S with the SCD markers
Output: finite-clauses of the S sentence +

```
index_fvg1 := -1, index_fvg2 := -1;
index_mrk := -1;
nr_fvg = 1;
index_fvg1 := findFVG(S, nr_fvg);
while(index_fvg1 != -1)
{
   index_fvg2 := findFVG(S, nr_fvg+1);
   if(index_fvg2 != -1)
   {
       index_mrk = findMRK(index_fvg1, index_fgv2,
   "M3" OR "M25");
       if(index_mrk != -1)
       {
           insert_boundary(index_mrk);
           continue;
       }
       else
       {
           index_mrk = findMRK(index_fvg1,
   index_fvg2, "M20" OR "M21");
```

```
        if(index_mrk != -1)
        {
            insert_boundary(index_mrk);
            continue;
        }
        else
        {
            index_mrk = index_fvg2;
            insert_boundary(index_mrk);
        }
     }
  }
  index_fvg1 = index_fvg2;
  nr_vb ++;
}//end_while
```

7.2 Running Marcu's and SCD Algorithms

When working with SCD marker tags, *i.e.* Mpq identifiers ($p = 1 \div 4$, $q = 1 \div 5$) or X-p marker labels in Fig. 2.1-2.2 (Part I), it is necessary to transform a morphologic (or POS) *automatic tagging* into SCD *tagging*. SCD *annotation* can be performed with a small computational price, the only problem arising is the assignment of the *predicational* feature values PROC or EXIST to those major lexical categories N, V, or A that bear it (if it was not already assigned at the *lexicon level*). The *TexTag* C++ environment was developed for both the (manual) control of morphologic (POS) and SCD tagging, as well as for the automatic transformation of POS tagging into SCD annotation.

For segmentation / parsing tasks, we developed two main programs: the *ClauSEGM environment*, written in Visual C++ 6.0 [14], [17] and used to implement the Marcu's segmentation algorithm, and the *SCD-Segmentation environment*, written in Java and used to implement the SCD segmentation and parsing algorithms for Romanian sentence. Here it is the result of running an example with both programs.

Ex.7.2.1.Marcu_SEG. Marcu's (unstructured) segmentation-at-discourse algorithm within *ClauSEGM* environment (see Fig. 7.2.1)

[În toamna aceea n-a nins decât foarte târziu.]1 [Locuiam într-un *chalet* din lemn, aflat într-o pădure de pini de pe coasta unui munte şi noaptea totul îngheţa, încât dimineaţa cele două căni cu apă de pe bufet aveau o pojghiţă de gheaţă pe deasupra.]2 [Dimineaţa, devreme, Mrs. Guttingen intra în cameră]3 [ca să închidă ferestrele şi făcea focul în soba cea mare de folosinţă.]4 [Surcelele de brad pârâiau şi scoteau scântei şi focul începea]5 [să duduie în sobă.]6 [A doua oară, Mrs. Guttingen venea cu nişte butuci groşi de lemn pentru foc şi o cană cu apă fierbinte.]7

Ex.7.2.2.SCD_SEG. *SCD*-2004 (unstructured) segmentation-at-clause algorithm within *SCDSegmentation* environment

[În toamna aceea n-a nins decât foarte târziu.]1 [Locuiam într-un *chalet* din lemn, aflat într-o pădure de pini de pe coasta unui munte]2 [şi noaptea totul îngheţa,]3 [încât dimineaţa cele două căni cu apă de pe bufet aveau o pojghiţă de gheaţă pe deasupra.]4 [Dimineaţa, devreme, Mrs. Guttingen intra în cameră]5 [ca să închidă ferestrele]6 [şi făcea focul în soba cea mare de folosinţă.]7 [Surcelele de brad pârâiau]8 [şi scoteau scântei]9 [şi focul începea]10 [să duduie în sobă.]11 [A doua oară, Mrs. Guttingen venea cu nişte butuci groşi de lemn pentru foc şi o cană cu apă fierbinte.]12

Example 7.2.3. The SCD analyses, as tagged code, are as follows:

POS Tagged Input:

<TOK ID="**TOK111**" root="**Nu**" pv="**Particle**" Type="**negation**">**Nu**</TOK> <TOK ID="**TOK112**" root="**avea**" pv="**Verb**" Type="**main**" Mood="**indic.**" Tense="**imperfect**" Person="**third**" Number="**singular**">**avea**</TOK> <COMP ID="**COMP4**" pv= "**Determiner**" Person="**third**" Gender="**masculine**" Number="**singular**" Quantification="**existential**">**nici un**</COMP> <TOK ID= "**TOK113**" root="**rost**" pv="**Noun**" Type="**common**" Gender= "**masculine**" Number="**singular**" Definiteness="**no**">**rost**</TOK> <TOK ID="**TOK114**" root="**sã**" pv="**Particle**" Type="**subjunctive**">**sã**</TOK> <TOK ID="**TOK115**" root="**încerca**" pv= "**Verb**" Type="**main**" Mood="**subj.**" Tense="**present**" Person= "**third**">**încerce**</TOK> <TOK ID="**TOK116**" root="**la**" pv=

"**Adposition**" Type="**preposition**" Formation="**simple**"> **la** </TOK> <TOK ID="**TOK117**" root="**lift**" pv="**Noun**" Type= "**common**" Gender="**masculine**" Number="**singular**" Definiteness= "**no**">**lift**</TOK> <PTERM_P ID="**PTERM_P4**" type="**PERIOD**">**.**</PTERM_P>

SCD Tagging output:

<TOK ID="**TOK111**" root="**Nu**" Type="**negation**" pv="**Particle**" mark="**M13**">**Nu**</TOK> <FVGIN ID="**FVGIN_9**" mark= "**M24**"> <TOK Type="**main**" Mood="**indic.**" mark="**M24**" Number="**singular**" ID="**TOK112**" Person="**third**" root="**avea**" Tense ="**imperfect**" pv="**Verb**">**avea**</TOK> </FVGIN> <TOK ID= "**COMP4**" Gender="**masculine**" Person="**third**" Number="**singular**" pv="**Determiner**" Quantification="**existential**" mark="**M14**"> **nici un**</TOK> <TOK ID="**TOK113**" root="**rost**" Gender="**masculine**" Type="**common**" Number="**singular**" pv="**Noun**" Definiteness="**no**" mark="**M11N**">**rost**</TOK> <FVGIN ID= "**FVGIN_10**" mark="**M24**"> <TOK ID="**TOK114**" root="**sã**" Type="**subjunctive**" pv="**Particle**" mark="**M24**">**sã**</TOK> <TOK ID="**TOK115**" root="**încerca**" Person="**third**" Type= "**main**" pv="**Verb**" Mood="**subj.**" Tense="**present**" mark= "**M24**">**încerce**</TOK> </FVGIN> <TOK ID="**TOK116**" root= "**la**" Type="**preposition**" Formation="**simple**" pv="**Adposition**" mark="**M14**">**la**</TOK> <TOK ID="**TOK117**" root="**lift**" Gender="**masculine**" Type="**common**" Number="**singular**" pv= "**Noun**" Definiteness="**no**" mark="**M11N**">**lift**</TOK> <MRK ID="**MRK24**" mark="**M34**"> <TOK ID="**PTERM_P4**" type= "**PERIOD**" mark="**M34**">**.**</TOK> </MRK>

SCD Segmentation output:

<clauza id="**9**"> <TOK ID="**TOK111**" root="**Nu**" Type="**negation**" pv="**Particle**" mark="**M13**">**Nu**</TOK> <FVGIN ID= "**FVGIN_9**" mark="**M24**"> <TOK Type="**main**" Mood="**indic.**" mark="**M24**" Number="**singular**" ID="**TOK112**" Person="**third**" root="**avea**" Tense="**imperfect**" pv="**Verb**">**avea**</TOK> </FVGIN> <TOK ID="**COMP4**" Gender="**masculine**" Person= "**third**" Number="**singular**" pv="**Determiner**" Quantification="**ex-**

istential” mark=”**M14**”>**nici un**</TOK> <TOK ID=”**TOK113**” root=”**rost**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv=”**Noun**” Definiteness=”**no**” mark=”**M11N**”>**rost** </TOK> </clauza> <clauza id=”**10**”> <FVGIN ID=”**FVGIN_10**” mark=”**M24**”> <TOK ID=”**TOK114**” root=”**sã**” Type=”**subjunctive**” pv=”**Particle**” mark=”**M24**”>**sã**</TOK> <TOK ID= ”**TOK115**” root=”**încerca**” Person=”**third**” Type=”**main**” pv= ”**Verb**” Mood=”**subj.**” Tense=”**present**” mark=”**M24**”>**încerce** </TOK> </FVGIN> <TOK ID=”**TOK116**” root=”**la**” Type=”**preposition**” Formation=”**simple**” pv=”**Adposition**” mark=”**M14**”>**la** </TOK> ¡TOK ID=”**TOK117**” root=”**lift**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv=”**Noun**” Definiteness= ”**no**” mark=”**M11N**”>**lift**</TOK> <MRK ID=”**MRK24**” mark= ”**M34**”> <TOK ID=”**PTERM_P4**” type=”**PERIOD**” mark= ”**M34**”>**.**</TOK>

</MRK> </clauza>

SCD Parsing output:

<clauza id=”**9**”> <FVG id=”**9**”> <TOK ID=”**TOK111**” root= ”**Nu**” Type=”**negation**” pv=”**Particle**” mark=”**M13**”>**Nu**</TOK> <FVGIN ID=”**FVGIN_9**” mark=”**M24**”> <TOK Type=”**main**” Mood=”**indic.**” mark=”**M24**” Number=”**singular**” ID=”**TOK112**” Person=”**third**” root=”**avea**” Tense=”**imperfect**” pv=”**Verb**”>**avea** </TOK> </FVGIN> </FVG> <NG id=”**39**”> <TOK ID= ”**COMP4**” Gender=”**masculine**” Person=”**third**” Number=”**singular**” pv=”**Determiner**” Quantification=”**existential**” mark=”**M14**”> **nici un**</TOK> <TOK ID=”**TOK113**” root=”**rost**” Gender=”**masculine**” Type=”**common**” Number=”**singular**” pv=”**Noun**” Definiteness=”**no**” mark=”**M11N**”>**rost**</TOK> </NG> </clauza> <clauza id=”**10**”> <FVG id=”**10**”> <FVGIN ID=”**FVGIN_10**” mark=”**M24**”> <TOK ID=”**TOK114**” root=”**sã**” Type=”**subjunctive**” pv=”**Particle**” mark=”**M24**”>**sã**</TOK> <TOK ID= ”**TOK115**” root=”**încerca**” Person=”**third**” Type=”**main**” pv= ”**Verb**” Mood=”**subj.**” Tense=”**present**” mark=”**M24**”>**încerce** </TOK> </FVGIN> </FVG> <NG id=”**40**”> <TOK ID= ”**TOK116**” root=”**la**” Type=”**preposition**” Formation=”**simple**”

pv="**Adposition**" mark="**M14**">**la**</TOK> <TOK ID= "**TOK117**" root="**lift**" Gender="**masculine**" Type="**common**" Number="**singular**" pv="**Noun**" Definiteness="**no**" mark= "**M11N**">**lift**</TOK> </NG> <MRK ID="**MRK24**" mark= "**M34**"> <TOK ID="**PTERM_P4**" type="**PERIOD**" mark= "**M34**">**.**</TOK> </MRK> </clauza>

Example 7.2.4. Here there are *two snapshots* of running the *environments: ClauSEGM* and *SCDSegmentation*.

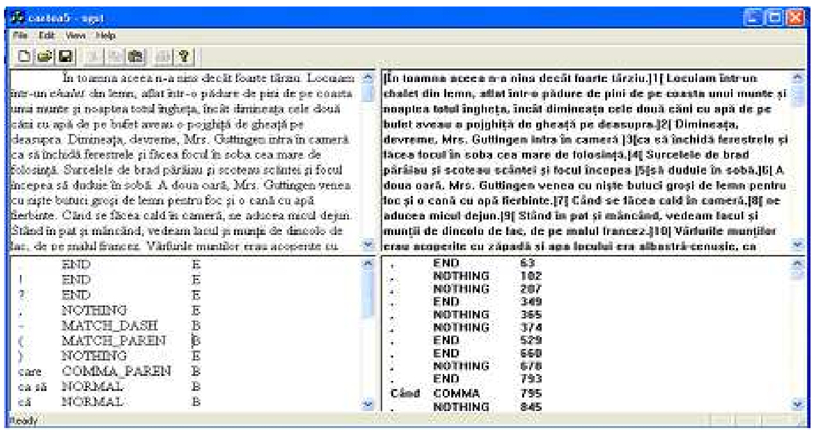

Figure 7.2.1. *ClauSEGM* run with Marcu's algorithm

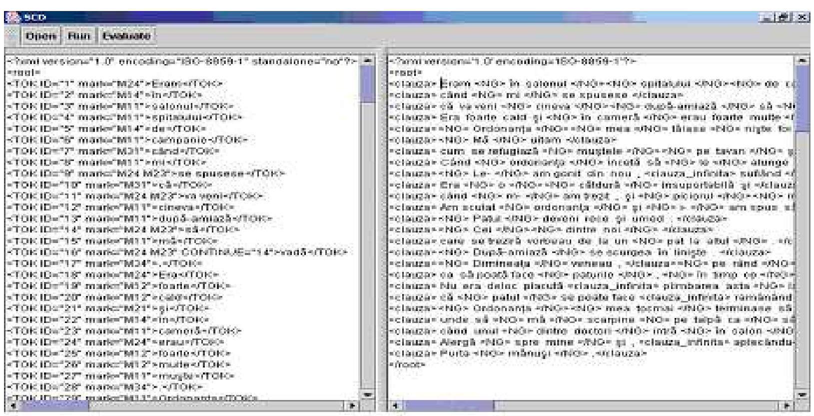

Figure 7.2.2. *SCDSegmentation* run with *SCD*-2004 algorithm

8 SCD Global Parsing Algorithms

8.1 SCD Clause-Level Global Parsing

The following step *after* the SCD *segmentation* is to use the resulted clauses for establishing the dependencies between them. In order to do this, there are two working possibilities: **(a)** dependency determination using the same marker classes as in the segmentation algorithm; **(b)** dependency resolving using lexical markers.

Both cases are using the information from a database containing, for each marker, the class to which it belongs, the type of the introduced relation (subordination, coordination, super-ordination), the name of the relation (Relative, Conditional etc.), the succession of the clauses in that relation, the place and distance to the clause of the related current marker, as well as information about the markers that correlate to the current marker (position and correlation distance).

For each sentence in the text, divided into clause, the marker sequence is being processed in several steps.

In the *first step* we establish a partial tree of the dependency relations using the information in the database described above.

In the *next step*, we use a set of rules in order to correct and complete the tree obtained in the previous step, by processing the whole marker sequence, corresponding to the current sentence.

In order to determine the dependencies between clauses, we must take under consideration some general cases of resolving, using the M3 level lexical markers.

The same operations will be used for determination of the rhetorical relations, in a subsequent step, the difference being determined only by the nature of the discourse units and the structure of the marker database.

The general case considers a clausal unit (Mi, Ci) (Marker, Clause) which contains a marker that can be found, with the information necessary to establish its super-ordinate, in the database.

i) Marker Correlation Processing

In the case of *marker correlation*, in order to ascertain the dependencies, the problem can be solved by structuring the information from

Table 8.1. SCD M3 Lexical/Class Marker Database for Clause-level

Mar-ker	**Class**	**Type**	**Rela-tion**	**Succ**	**Wh2-Lnk**	**Dist**	**Corre-late**	**Dist-2Corr**	**Wh-2C**	**Disc role**
Care	M25	Sub	Atr	R S	b(efore)	1	-			no
Dacă	M31	Sub Sub	Cond	R S	b	1	atunci altfel	2 1	b a	yes
Altfel	M33	Sub Coord	Cond	R R	b	2	dacă atunci	1 2	b a	yes
Şi	M21	Coord		R R	b	1	şi	1	b	no
a căror	M25	Sub	Atr	R S	b	1	-			no
aşa încât	M31 M12V	Sub	Cons	R S	b	1				yes
Aşa cum	M31 M12V	Sub	Mod	S R	a	1	tot aşa	2	b	yes
Acolo unde	M31	Sub	Loc	R S	b	1				yes
Cât	M31 M12V	Sub	Nspec		a	1				yes
Fie	M33	Coord	Coord	R R			fie	1	a	yes
Nici	M33	Coord	Coord	R R			nici	1	a	yes

correlated markers as in the database described above.

Thus, besides the information necessary to establish the dependencies in the general case, (Mi, Ci) (Mj Cj), the database contains several fields referring to the markers that correlate to the current one, namely the *lexical* marker(s), the correlation *distance* (number of marker units to correlate over), the correlation *direction* (before or after the current unit).

For instance, the necessary facts referring to correlation in case of "*dacă*" (*if*) marker are: the markers that *dacă*" is correlated with (*atunci*, *altfel*) (*then*, *else*), the correlation distance (2, 1), and the correlation direction (Before, After) respectively, for each of the markers that correlate with "*dacă*" (*if*).

dacă	Sub Sub	atunci altfel	2 1	b a	yes

In case at some parsing point one encounters the "*dacă*" (*if*) marker, the unit that contains it (Unit i) will be correlated in the *Sub*(*ordination*) relation to the $(i-2)$ Unit, which contains the "*atunci*" (*then*) marker, and to $(i+1)$ Unit, which contains the "*altfel*" (*else*) marker, as well in the *Sub* relation. In the same manner is solved the coordination, which represents a particular case of correlation.

Examples of correlation: *dacă... atunci... altfel*; *deşi... totuşi*;

Examples of coordination: *Fie... fie*; *Nici ... nici*; *Sau... sau*; *Ori ... ori*; *Nici ... ci*; *ci... şi... ci...* ; *ori... însă... ori*; *şi... dar ... şi*.

ii) Marker Sequence Processing

In order to establish the dependencies at the inter-clausal level, the marker sequences are treated as follows: any sequence of two or more markers is divided in other two subsequences, until one obtains sequences of two markers, treated analogously to the general case.

Therefore, for sequences like (Mi Ci) (Mj Cj), if the super-ordinate (regent) clause of Ci is established based on the information from database about the Mi marker, then the regent of Cj is established based on the information about the Mj marker.

For sequences like (Mi (Mj Cj) Ci), the regent clause of Ci is established according to the data about Mi, and Cj will have as regent the unit that includes it, that is Ci, rule specified in the database, at the Mj information.

If marker classes are utilized, the type of the relation (subordination, coordination, super-ordination) can be determined as well, but with a smaller precision, without being able to specify exactly the name of the relation, like using lexical markers. In this case, some important advantages would provide the increased processing speed and the reduced dimension of the database and dependency establishing rules.

After building the tree from the local relations (that can be found in the database), turn back to the obtained tree to correct the existing relations and to add some new ones, resulted as supplementary information from processing the whole sequence of markers ascribed to the entire sentence (e.g. verification of the last XG in the previous clause could give additional information about the type of the currently processed clause). Also, a set of rules is applied to the firstly resulted dependency tree.

Examples of rules:

Rule 1: "*dacă*" marker can correlate to a comma in the absence of "*atunci*".

Rule 2: If after a "*dacă*" follows a "*să*", "*care*", "*ca*" marker, the clause that contains "*dacă*" correlates to one (usually the first) after those clauses that contain the respective markers.

Rule 3: If there is an "*atunci*", but not preceded by "*dacă*", then one can correlate the "*atunci*" marker with "*când*", if any.

Rule 4: Repetition of the same marker (preceded or not by comma or "*şi*") represents a correlation.

8.2 SCD Discourse-Level Global Parsing

A type of parsing that presents a great deal of interest is *discourse parsing*. As noticed in Part I of this paper, discourse parsing can be done in various ways. The method proposed by the SCD strategy includes building the discourse tree using M4-level lexical markers, while discourse segments being obtained by clause parsing.

Using the results of the SCD clausal parsing and a database which contains information about the discourse markers, one can obtain the discourse structure of a text. The result is represented as a *discourse tree* whose terminal nodes are (almost always) clauses, having specified on the arcs the name of the involved rhetorical relations.

Discourse parsing can be done at sentence level [38], as well as between larger spans of text, like between sentences, paragraphs or sections of a text. For each type of text unit, separate sets of markers and rules are determined and applied.

Table 8.2. SCD M4 Lexical Marker Database for Discourse

Marker	**Rhet-Rel**	**Succ**	**Wh2L**	**Dist**	**Correlation**	**DisTo-Corr**	**Wh2C**
dacă	Cond	N S	B	1	atunci altfel	2 1	B A
altfel	Cond	N N	B	1	dac atunci	2 1	b b
în primul rând	Enum	N N	A	1	în al doilea rând	1	a
în al doilea rând	Enum Enum	N N N N	a b	1 1	în al treilea rând în primul rând	1 1	a b
în al treilea rând	Enum Enum	N N N N	a b	1 1	în al patru-lea rând în al doilea rând	1 1	a b

9 Evaluations and Conclusions

Part II of the paper outlines SCD variants of the segmentation/parsing algorithms, considering *criteria* of lexical and class levels of phrase markers, in the context of a general hierarchy of marker classes proposed for SCD (Fig. 2.3, Part I). The *newly defined* notions of strong and weak *functional generative capacity* are introduced for the lexical-level and class-level phrase markers, respectively.

In [14] and [17], SCD segmentation and parsing at the clause level are compared to several clause or clause-like segmentation and parsing algorithms such as those exposed in [25], [36], [31], [6] dealing with Romanian or English text segmentation / parsing. E.g., Marcu's clause-like (actually, discourse) segmentation is proved to be "embedded" into SCD (finite) clause segmentation, except the special situation of *subclausal discourse segments* [14]. Applied on several hundreds of (both Romanian and English) sentences, the implementation of [25] clause-like segmentation algorithm has a precision of 73% and a recall of 69% for Romanian, and a precision and recall of approx. 95% for English (Hemingway's "Farewell to Arms"). [36] and [31] provide similar segmentation methods for Romanian, respectively English, using lexical word-patterns around the clause boundary, obtaining 89.03% precision and 88.51% recall for Romanian, similar results for English, and increasing the precision / recall rate to approx. 95% when applied machine learning to the segmentation process.

For the current Java implementation of the SCD segmentation / parsing algorithm [17], tests have been effectuated from the whole "1984" (George Orwell) corpus, representing approximately 6.500 sentences and 15.000 clauses. The number of correctly recognized clauses rises up to 14.500, which means a precision of 96,6% and a recall of 95%. Comparing to the previous version of the parser, [17] obtained an improvement of approx. 10%, the previous test being done on 1500 sentences from the whole corpus.

In taking the tests on the SCD segmentation algorithm, several problems were identified, which led to an incorrect parsing of a number of approx. 500 sentences from the 15.000. The 3.4% error rate is due to some problems of lexical nature, to the lack of proper punctuation marks. Also, parsing errors can be found in the case of imbricate clauses, where there are no key words to indicate the continuation of a clause previously opened.

As a continuation of current topics, a *class of parsing algorithms* (at clausal and discourse level) is proposed, which can be described within the SCD strategy by refining the SCD marker classes (and hierarchy) towards the *lexical level* of the contained markers. A *generalization* of Marcu's *lexical marker database* was defined, both at the M3 and M4 classes of SCD marker hierarchy (Tables 8.1 and 8.2), which can be used in (inter-clausal and discourse rhetorical) parsing algorithms.

References

[1] Barbu, Ana-Maria; Emil Ionescu (1996): *Contemporary grammatical theories*: *grammars of the phrasal head.* in Limba româna, no. 45, (1-6) pp. 31–55 (in Romanian).

[2] Barbu, Ana-Maria (1999): *The Verbal Complex.* Studii si Cercetari Lingvistice, L, no.1, Bucureşti, pp. 39–84 (In Romanian).

[3] Chomsky, Noam (1981): *Lectures on Government and Binding*. Foris, Dordrecht.

[4] Chomsky, Noam (1986): *Barriers*. The MIT Press, Cambridge.

[5] Chomsky, Noam (1995): *The Minimalist Program*. The MIT Press, Cambridge, Massachusetts.

[6] Cristea, D., O. Postolache, I. Pistol (2005): *Summarisation through Discourse Structure*. In Proceedings of CiCling 2005, Springer LNSC, vol. 3406.

[7] Curteanu, Neculai (1988): *Augmented X-bar Schemes*. COLING'88 Proceedings, Budapest, pp. 130–132.

[8] Curteanu, Neculai (1990): *A Marker-Hierarchy-based Approach Supporting the SCD Parsing Strategy*. Research Report no. 18, Institute of Technical Cybernetics, Bratislava, Slovak Republik.

[9] Curteanu, Neculai (1994): *From Morphology to Discourse Through Marker Structures in the SCD Parsing Strategy. A Marker-Hierarchy Based Approach*. Language and Cybernetics, Akademia Libroservo, Prague, pp. 61–73.

[10] Curteanu, Neculai; G. Holban (1996): *SCD Linguistic Strategy Applied to the Analysis and Generation of Romanian*. In (Dan Tufi, Ed.) Language and Technology, Romanian Academy, Bucharest, pp. 169–176 (in Romanian).

[11] Curteanu, Neculai (2002): *Elements of a Functional X-bar Theory Within the SCD Linguistic Strategy*, ECIT2002 Conference, Iaşi, România.

[12] Curteanu, Neculai (2003): *Towards a Functional X-bar Theory*. In the volume "The Romanian Language in the Informational Society", Dan Tufiş, F. Filip (Eds.), Edited by the Romanian Academy, Research Institute for Artificial Intelligence, Bucharest, pp. 51–86 (in Romanian).

[13] Curteanu, Neculai; D. Gâlea; C. Linteş (2003): *Segmentation Algorithms for Clause-Type Textual Units.* In the volume "The Romanian Language in the Informational Society", Dan Tufiş, F. Filip (Eds.), Edited by the Romanian Academy, Research Institute for Artificial Intelligence, Bucharest, pp. 165–190 (in Romanian).

[14] Curteanu, Neculai; D. Gâlea; C. Butnariu; C. Bolea (2004): *Marcu's Clause-like Discourse Segmentation Algorithm and SCD Clause Segmentation-based Parsing*, In the volume "Intelligent Systems" (Ed. Horia-Nicolai Teodorescu). Selected Papers from ECIT2004 Conference, Iaşi, România, pp. 59–86.

[15] Curteanu, Neculai (2003-2004): *Contrastive Meanings of the Terms* "Predicative" *and* "Predicational" *in Various Linguistic Theories* (I, II). Computer Science Journal of Moldova (R. Moldova), Vol. 11, No. 4, 2003 (I); Vol. 12, No. 1, 2004 (II).

[16] Curteanu, Neculai (2005): *Functional FXbar Theory Extended to Discourse (Rhetorical) Structures.* In 'Intelligent Systems' Conference Volume, H.-N. Teodorescu *et al.* (Editors), Performantica Press, Iaşi (Romania), pp. 169–182.

[17] Curteanu, Neculai; E. Zlavog; C. Bolea (2005): *Sentence-Based and Discourse Segmentation / Parsing with SCD Linguistic Strategy.* In 'Intelligent Systems' Conference Volume, H.-N. Teodorescu *et al.* (Editors), Performantica Press, Iaşi (Romania), pp. 153–168.

[18] Curteanu, Neculai; Diana Trandabăţ (2006): *Functional (F)X-bar Projections for Local and Global Text Structures. The Anatomy of Predication.* Revue Roumaine de Linguistique, Bucharest (to appear).

[19] Dobrovie-Sorin, Carmen (1994): *The syntax of Romanian. Comparative Studies.* Berlin: Mouton de Gruyter.

[20] Irimia, Dumitru (1997): *The Morphosyntax of the Romanian Verb.* The Editorial House of the "Al. I. Cuza" Iaşi University (in Romanian).

[21] Joshi, Aravind K. and Ives Schabes (1997): *Tree Adjoining Grammars.* In “Handbook of Formal Languages and Automata” (A. Salomaa *et al.*, Eds.), Vol. 3, Heidelberg, Springer-Verlag.

[22] Kornai, András, Geoffrey Pullum (1990): *The X-bar Theory of Phrase Structure*, Language, Vol. 66, No. 1, pp. 24–50.

[23] Legendre, Géraldine (1999): *Optimal Romamian clitics: a cross-linguistic perspective.* In: V. Motapanyane (Ed.) Comparative Studies in Romanian Syntax. HAG, The Hague.

[24] Mann, William, Sandra Thompson (1988): *Rhetorical Structure Theory: A Theory of Text Organization.* Research Report RS-87-190, Information Sciences Institute, University of Southern California, Marina del Rey, California, 80 p.

[25] Marcu, Daniel (1997): *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*, Ph.D. Thesis, Univ. of Toronto, Canada, pp. 331.

[26] Marcu, Daniel (2000): *The Theory and Practice of Discourse Parsing and Summarization.* The MIT Press.

[27] Miller, Philip (1999): *Strong Generative Capacity. The Semantics of Linguistic Formalism.* CSLI Publications, Stanford, California, 1999.

[28] Monachesi, Paola (1998): *The Morphosyntax of Romanian Cliticization.* In: P.-A. Coppen *et al.* (Eds.), Proceedings of Computational Linguistics in The Netherlands 1997, pp. 99–118, Amsterdam-Atlanta: Rodopi.

[29] Monachesi, Paola (2000): *Clitic placement in the Romanian verbal complex.* In: B. Gerlach and J. Grijzenhout (eds.) Clitics in phonology, morphology, and syntax. Linguistik Aktuell. John Benjamins. Amsterdam.

[30] Monachesi, Paola (2005): *The Verbal Complex in Romance. A Case Study in Grammatical Interfaces.* Oxford University Press, Oxford Studies in Theoretical Linguistics.

[31] Orasan C.(2000): A hybrid method for clause splitting in unrestricted English texts Available at: http://www.wlv.ac.uk/sles/compling/papers/orasan-00.pdf

[32] Passonneau, Rebecca; Diane Litman (1997): *Intention-based segmentation: human reliability and correlation with linguistic cues*, in Proc. 31th Annual Meeting of ACL, Ohio, pp. 148–155.

[33] Gheorghe Păun: *Marcus Contextual Grammars*. Kluwer Academic Publishers, Dordrecht, 1997.

[34] Pollard, Carl; Ivan Sag (1994): *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago & London.

[35] Popârda, O.; N. Curteanu (2002): *L'évolution du discours juridique français analysé par la stratégie linguistique SCD*. In the volume "Representations du Sens Linguistique", Lagorgette, P. Larrivée (Eds.), LINCOM Europa, series Studies in Theoretical Linguistics, München, Germany, pp. 487–502.

[36] Puşcaşu, G. (2003): *Elementary discourse unit segmentation*. Dissertation thesis. "Al.I.Cuza" University of Iaşi.

[37] Sgall, Petr; E. Hajičova, J. Panevova (1986): *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Kluwer Academic Publishers, Dordrecht.

[38] Soricut, R. and Daniel Marcu (2003): Sentence Level Discourse Parsing using Syntactic and Lexical Information. *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, May 27-June 1, Edmonton, Canada.

N. Curteanu,

Received March 6, 2006

Institute for Computer Science,
Romanian Academy, Iaşi Branch
B-dul Carol I, nr. 22A, 6600 IAŞI, ROMÂNIA
E–mail: *ncurteanu@yahoo.com* and *curteanu@iit.tuiasi.ro*

Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

Text Classification Using Word-Based PPM Models

Victoria Bobicev

Abstract

Text classification is one of the most actual among the natural language processing problems. In this paper the application of word-based PPM (Prediction by Partial Matching) model for automatic content-based text classification is described. Our main idea is that words and especially word combinations are more relevant features for many text classification tasks. Key-words for a document in most cases are not just single words but combination of two or three words. The main result of the implemented experiments proved applicability of word-based PPM models for content-based text classification. Although in some cases the entropy difference which influenced the choice was rather small (several hundredths), most of the documents (up to 97%) were classified correctly.

1 Introduction

Text or document classification is the assignment of documents to predefined categories on the base of their content. It is one of the most actual natural language processing problems. Message classification is an every day problem for every person, using electronic mail; an adequate system for spam detecting has not been developed yet. Automatic text classification at the news tapes, automatic subject classifier in on-line libraries would be of much help for people supporting these services. The number of files, stored at a typical computer is also increasing rapidly; those collections will also need an automatic classification.

There are different types of text classification. Authorship attribution, spam filtering, dialect identification are just several of the purposes of text categorization. It is natural that for different types of categorization different methods are pertinent. The most common type is the content-based categorization which classifies texts by their topic, objects and events they describe.

In this paper the application of word-based PPM (Prediction by Partial Matching) model for automatic content-based text classification is described. Although the application of PPM model to the document classification is not new, the novelty in our approach is word-based model we support.

Our main idea is that words and especially word combinations are more relevant features for many text classification tasks. It is known that key-words for a document in most cases are not just single words but combinations of two or three words. Thus, sequences of words are quite representative for text classification task.

The following sections of this paper describe machine-learning and compression-based approaches to text classification. Several word-based models are considered and a particular word-based PPM model for content-based text categorization is presented. Finally, some experimental results for PPM models based on trigrams, bigrams and unigrams are reported.

2 Related works

Typical approaches to text classification extract "features" from documents and form feature vectors which are used as an input to a machine learning technique. The features are generally words. Because are so many of them, a selection process is applied to determine the most important ones and the remainder are discarded.

Lately the most wide spread machine learning techniques used for classification are based on the SVM (support vector machine). Almost any classification method can be reduced to separation hyper plain construction in terms of SVM method [3]. Although separation of vectors using plains appears rather simple, SVM classification methods are more effective than the others: decision tree approach [12], cluster classification [14], naïve Bayes classification [11], and neural nets [13].

SVM classification methods achieve high level of precision due to the separation of and adjustment to text "features", which are words of the text. The problem of SVM is the requirement of quadratic convex programming that demands time expenses and inevitable use of floating-point arithmetic. Text classification by SVM methods demands such large number of characteristics that the task becomes computationally not feasible. Of course, new methods of optimization are developed, but still SVM is capable of operating with not more than ten thousands characteristics.

A number of approaches that apply text compression models to text classification have been presented recently. The underlying idea of using compression methods for text classification was their ability to create the language model adapted to particular texts. It was supposed that this model captures individual features of the text being modeled.

3 Statistical Models for compression.

A number of powerful modelling techniques have been developed in recent years to compress natural language text. The best of these are adaptive models operating on character and word level, which are able to perform almost as well as humans at predicting text.

PPM (prediction by partial matching) is an adaptive finite-context method for compression. It is based on probabilities of the upcoming symbol in dependence of several previous symbols. Firstly this algorithm was described in [1], [2]. Lately the algorithm was modified and in [7] was described an optimized PPMC (Prediction by Partial Matching, escape method C) algorithm. PPM has set the performance standard for lossless compression of text throughout the past decade. In [10] it was shown that the PPM scheme can predict English text almost as well as humans. The PPM technique blends character context models of varying length to arrive at a final overall probability distribution for predicting upcoming characters in the text.

For example, the probability of character 'm' in context of the word 'algorithm' is calculated as a sum of conditional probabilities in dependence of different length context up to the limited maximal length:

P('m') = $\lambda_5\cdot$ P('m' | 'orith') + $\lambda_4\cdot$ P('m' | 'rith') + $\lambda_3\cdot$ P('m' | 'ith') + $\lambda_2\cdot$ P('m' | 'th') + $\lambda_1\cdot$ P('m' | 'h'),
where λ_i $(i = 1 \ldots 5)$ is normalization factor; 5 - maximal length of the context.

The models are adaptive: the counts for each context are updated progressively throughout the text. In this way, the models adapt to the specific statistical properties of the text being compressed. This particular feature of the model is used to sort documents.

4 Classification using PPM models.

Most of compression models are character-based. They treat the text as a string of characters. This method has several potential advantages. For example, it avoids the problem of defining word boundaries; it deals with different types of documents in a uniform way. It can work with text in any language and it can be applied to diverse types of classification.

In [6] the simplest way of compression-based categorization called off-the-shelf algorithm is used for authorship attribution. The main idea of this method is as follows. Anonymous text is attached to texts which characterize classes, and then it is compressed. A model, providing the best compression of document, is considered as having the same class with it. Among 16 compared compression algorithms, the best performance was obtained by **rarw** which uses PPMD (Prediction by Partial Matching, escape method D) compression program (71 correctly attributed texts for 82 test texts).

The other approach is direct measuring of text entropy using a certain text model. PPM is appropriate in this case, because text modeling and its statistic encoding are two different stages in this method. In [5] it was shown that results of this method were very similar to the results of the off-the-shelf algorithm. In their paper authors applied compression-based method to multi-class categorization problem in order to find duplicated documents in large collections. Comparing several compression algorithms, the authors found that the best performance was obtained by RAR and PPMD5 (84%-89% for different conditions).

In [9] similar approach was used for language identification, to identify the period of historical English texts, for dialect identification and for authorship attribution. All these problems may be viewed as text categorization problems and solved using minimum-entropy approach based on PPM model.

In [19] several compression schemes were used for source based text categorization. The result was not as satisfactory as the author desired. Furthermore, the word-based PPM model tested on the paper performed worse than the letter-based one. The author considered it happened due to the small training set. Performing a great number of different experiments of compression-based categorization author concluded that more work needs to be done to evaluate the technique.

In [20] extensive experiments on the use of compression models for categorization were performed. They reported some encouraging results; however they found that compression-based methods did not compete with the published state of the art in use of machine learning for text categorization. Authors considered that the results in this area should be evaluated more thoroughly.

In [15] the letter-based PPM models were used for spam detecting. In this task there existed two classes only: spam and legitimate email (ham). The created models were applied to TREC spam filtering task and exhibited strong performance in the official evaluation, indicating that data-compression models are well suited to the spam filtering problem.

5 Word-based models.

Word-based statistical model uses a number of previous words to predict the following one. For the first time, statistical models based on Markov's chains of words were successfully used in speech recognition [4].

It is necessary to mention that word-based models present a problem when implemented practically. Number of different words in a text is much greater than number of letters in alphabet. While there is no problem to create a letter-based model with the context of 6, 7, 8 letters, the creation of word-based model with the context of 3 words is time and memory consuming. Word-based Markov chains are practically implemented with context of one (bigrams) or two words (trigrams) because the longer context demands large training corpora, much time (sometimes more than 24 hours of training) and memory.

Nonetheless, a number of word-based text compression schemes have already been proposed [16][17]. In [18], four word-based compression algorithms were implemented in order to take advantage of longer-range correlations between words and thus achieve better compression. The performance of these algorithms was consistently better than UNIX *compress* program.

In [9] the adaptive word-based PPM bigram model was used to improve text compression. This model created the shorter code in comparison with letter-based model, because the code was created for the whole word at once, so less number of bits was used to code each letter. Besides, it provided faster compression than character-based models because fewer symbols were being processed.

Results with these models have shown that the word-based approach generally performs better when applied to compression.

6 Classification using word-based PPM model.

As usual, PPM based classification methods use symbol-based models. As mentioned above, experiments showed that given classification methods achieved results, competitive to those obtained by classical techniques. PPM classification methods are based on text fragments consisting of a number of symbols. This number should not be higher than a certain value which is called maximal context. As usual, maximal context is five symbols long, because it was proved, that this maximal context value provides best performance for PPM [9]. Taking into consideration, that PPM models based on 5 or less symbol text fragments have best achievements in document classification, one can assume that those fragments characterize texts good enough for text classification.

However, if texts are classified by the contents, they are better characterized by words and word combinations than by fragments consisting of five letters. We believe that words are more indicative text features for content-based text classification. That's why we decided to use a model based on words for PPM text classification.

It is obvious that 5-word contexts are impossible to use. As it was mentioned above, in case of words, one or two word context usually is used. Therefore we applied PPM model based on two, one and zero word contexts. In case of zero context, words without context were used. In this case, the same information about the text was available as in common classification methods. As it is known, classical methods of documents categorization are based in most cases on frequency dictionary.

We used minimum cross-entropy as a text classifier [22], using models, created on the base of certain classes of documents. As it was mentioned above, PPM is the most convenient for these purposes, because it has text modelling and its statistical encoding separated in two different stages.

In informational theory [21], the fundamental coding theorem states that the lower bound to the average number of bits per symbol needed to encode a message (text) is given by its entropy:

$$H = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{1}$$

where

$p(x_i)$ - probability of symbol x_i in the message, for all symbols in the message, $i = 1 \ldots n$;

PPM compression algorithm can be used to estimate the entropy of text as its modelling part estimates the probabilities of text symbols. Thus, one can estimate the probabilities of symbols in the text and then calculate their entropy using equation (1). The entropy provides a measure of how well the probabilities were estimated; the lower entropy is, the better probabilities are estimated.

Cross-entropy is the entropy calculated for a text if the probabilities of its symbols have been estimated on another text:

$$H_d^m = -\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i) \qquad (2)$$

were

H_d^m – text d entropy obtained using model m;

$p^m(x_i)$ - probability of symbol x_i using model m for all symbols in the text d $(i = 1 \ldots n)$;

m – a statistic model created on the base of another text.

Usually, the cross-entropy is greater than the entropy, because probabilities of symbols in diverse texts are different. The cross-entropy can be used as a measure for document similarity; the lower cross-entropy for two texts is, the more similar they are. It is considered that texts with lowest cross-entropy belong to the same class. Hence, if several statistic models had been created using documents that belong to different classes and cross-entropies are calculated for an unknown text on the base of each model, the lowest value of cross-entropy will indicate the class of the unknown text. In this way cross-entropy is used for text classification.

Thus, two steps were realized: (1) creation of PPM models for every class of documents; (2) entropy estimation for unknown document using models for each class of documents.

The unknown document considered to be of the same class with the model providing the lowest value of entropy.

In the experiments the entropy per word was calculated in order to avoid influence of document's size on the entropy value:

$$H_d^m/n = (-\sum_{i=1}^{n} p^m(x_i) \log p^m(x_i))/n \qquad (3)$$

where n is number of words in document d.

The aim of the experiments was twofold:

- to see how distinct the values of entropies on different models for the same document are;
- to evaluate quality of document classification by this method.

7 Experiments

To check the word based classification method using PPM, a set of experiments was made. The corpus of newspaper articles from the Romanian electronic newspaper “Evenimentul zilei” (Event of The Day) was used in the experiments. All the articles in this newspaper belonged to one of the 7 categories:

- editorial;
- money, business;
- politics;
- investigations;
- quotidian;
- in the world;
- sport.

Thus there were documents of seven categories. Each category was considered a class of documents in the classification task. To verify the documents classification quality, word-based trigram PPM models, based on groups of documents from each category separately, were created firstly.

As the result, seven models were created, each of them reflecting features of a certain category.

Then the entropies of a number of test documents were calculated using the created models (we took 10 test documents from each category, total - 70 documents). It was supposed that texts from one category had similar lexicon and differed from other texts. The entropy of texts from the same category as well as of those used to create the model must be less than in texts from other categories. So, having the entropy calculated on the base of seven models, we attributed the text to the category for which its entropy was minimal. In the table 1, the average entropy value per word for seven types of test documents is shown. Columns show seven models based on each text category, rows refer to test files of the given category. Figures in the table cells show average entropy per word for test documents of the row calculated on base of the model in the column.

Table 1. Average entropy value for test documents for seven categories (trigram model).

categories	Money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	**9,60**	10,30	10,39	10,33	10,23	10,12	10,34
quotidian	10,25	**10,02**	10,32	10,23	10,07	10,14	10,20
editorial	10,35	10,19	**9,59**	10,29	10,13	9,86	10,14
in the world	10,28	10,19	10,40	**9,38**	10,20	10,11	10,23
investigations	10,21	<u>**10,00**</u>	10,30	10,18	**9,62**	10,02	10,17
politics	10,09	10,18	10,07	10,11	10,03	**9,32**	10,16
sport	10,41	10,29	10,39	10,32	10,19	10,17	**9,06**

Minimal entropy obtained on each model is shown with bold. As it can be seen, articles from the same category which was used for the model creation have minimal entropy. It means that entropy, calculated by this way, can be used for the document classification. But it must be mentioned that there is a very small difference in values. Such a small difference in values increases the risk of errors.

The only mismatch is for the test articles from 'investigations' and the model for 'quotidian'. The figure is underlined.

In the table 2 the files are given separately. The entropy for each test file has been calculated. Each test document was classified to a category for which the entropy of given document was minimal.

Again columns show seven models accordingly to the categories, rows refer to test files of the given category. Figures in the table cells show number of test files classified to the category of the column.

Table 2. Test documents classification (trigram model).

categories	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	10	**10**						
quotidian	10	3	**5**			2		
editorial	10			**10**				
in the world	10				**10**			
investigations	10					**10**		
politics	10						**10**	
sport	10							**10**

Almost all the documents were classified correctly. But in some cases the difference in entropy values, that influenced the decision, was equal to one hundredth. The same can be said about documents that were classified incorrectly. Documents of only one category were classified wrongly: quotidian. It is obvious that the errors in classification were influenced by the category.

The category 'quotidian' is not a well-defined class of documents; it contains topical articles. Accordingly to the errors in classification, in most cases those were articles about finances and investments. Thus in this case errors are not due to the system imperfection, the category itself doesn't differ considerably from the other categories. This can explain the wrong minimal value in the previous table for 'quotidian' test files and 'investigations' model.

The next experiment was made using PPM based on word bigram models. The conditions were the same as in the previous one. In table 3 the results of the experiment are presented.

Table 3. Average entropy value for test documents (bigram model).

categories	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	**9,60**	10,34	10,50	10,42	10,27	10,19	10,50
quotidian	10,26	**10,05**	10,42	10,30	10,09	10,18	10,30
editorial	10,31	10,22	**9,58**	10,36	10,18	9,89	10,19
in the world	10,27	10,26	10,51	**9,40**	10,23	10,16	10,31
investigations	10,21	<u>**10,03**</u>	10,38	10,23	**9,59**	10,03	10,25
politics	10,10	10,23	10,06	10,16	10,05	**9,31**	10,23
sport	10,41	10,37	10,49	10,40	10,23	10,23	**9,02**

Again bold font shows the minimal entropy values. Similar to the two words context model all the categories were classified correctly except 'quotidian'. Bigram model can be as well used for documents classifications.

It must be said that bigram model took less computer memory and worked faster than trigram model. Thus more training texts could be used for this model. About 400-500 Kb of training files for each category were used in the experiment for the trigram model. Almost 1 Mb of training files for each category were used for the bigram model. Indeed, comparing the tables, one can see that the difference in entropy values in table 3 is a bit bigger than in 1. The increase of difference may occur for two reasons. Either bigram model better fitted the task of classification, or volume of the training texts influenced the results.

We can see that the results obtained with the bigram and the trigram models are similar. The category 'quotidian' remains the biggest problem here as well. It is interesting that the given model didn't relate

Table 4. Test documents classification (bigram model).

categories	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	10	**10**						
quotidian	10	1	**5**			4		
editorial	10			**10**				
in the world	10				**10**			
investigations	10					**10**		
politics	10						**10**	
sport	10							**10**

the questionable articles to "money, business" but selected "investigations". These two categories are very close, so the misunderstanding is easy to explain.

The next experiment was made using word-based unigram PPM model i.e. without any context. In fact, the classification was performed using frequency dictionaries. The other conditions remained the same. Test results of the model without context are presented in the table 5.

It is seen, that the results obtained using this model are worse in comparison with two previous experiments. Problem category 'quotidian' was mixed with categories 'money, business', 'investigations', 'politics' and 'editorial' (figures in italic).

It must be mentioned that if the size of training texts was enlarged when changing from trigram to bigram model, no changes of the texts size were produced when changing from bigram to unigram model. Probably, enlarging of the training texts size for the last model would improve its result. Thus the next step was to increase the training texts volume, to train and then classify using unigram model. In table

6 the results of this experiment are presented.

Table 5. Average entropy value for test documents (unigram model).

categories	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	**10,47**	10,92	10,91	10,87	10,79	10,75	10,91
quotidian	*10,73*	**10,78**	10,80	10,79	*10,70*	*10,72*	10,79
editorial	10,70	*10,78*	**10,37**	10,79	10,67	10,53	10,69
in the world	10,77	10,94	10,94	**10,73**	10,82	10,75	10,88
investigations	10,73	10,81	10,82	10,79	**10,54**	10,68	10,81
politics	10,72	10,91	10,69	10,78	10,74	**10,35**	10,80
sport	10,85	10,95	10,93	10,91	10,81	10,78	**10,53**

Table 6. Average entropy value for test documents (unigram model).

categories	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	**10,60**	11,00	11,13	11,10	10,95	10,95	11,07
quotidian	10,93	**10,85**	11,01	11,00	10,86	10,91	10,94
editorial	10,89	*10,84*	**10,57**	*10,97*	10,80	10,68	10,82
in the world	10,99	11,02	11,15	**10,98**	10,99	***10,96***	11,05
investigations	10,94	10,87	11,02	*10,97*	**10,63**	10,84	10,94
politics	10,90	10,97	10,84	*10,97*	10,86	**10,45**	10,93
sport	11,05	11,04	11,15	11,12	10,98	10,97	**10,62**

The results didn't improve. On the contrary, the categories were mixed even more. Of course it could be explained by the fact that in category 'in the world' there can be articles about 'politics' and 'investigations', thus their lexicons intersect. However we were interested in accurate classification in accordance with predefined categories. Table 7 presents classification results using unigram model.

Table 7. Test documents classification (unigram model).

categories	Total number of test documents	money, business	quotidian	editorial	in the world	investigations	politics	sport
money, business	10	***10*** **10**						
quotidian	10	22	***0*** **4**			7 4	1	
editorial	10			***9*** **8**			1 2	
in the world	10	1			***6*** **3**	2	4 4	
investigations	10					***9*** **10**	1	
politics	10						***10*** **10**	
sport	10							***10*** **10**

We used italic to show the classification results of the first experiment with unigram and bold for the results of the second experiment with unigram and enlarged size of training texts. As it is seen from the table, the increase of the text size gave rather arguable result. In some cases the classification quality improved, while in the others it became worse. It can be argued that the articles from the category 'in the world' speak about 'politics' and so on. On the other hand we didn't do the category division and our task was just to classify documents according to the initial classification.

Thus we can conclude that to classify the documents properly we would rather use bigram and trigram models, while the model with zero contexts does not fit here.

Also several experiments were made to check the influence of the training texts size over the classification quality. The volume of training texts for the bigram model was doubled. Test results are shown in table 8.

Comparing values in this table and table 3 for the bigram models, it can be seen that the difference in the entropy values of the given category texts and the texts from other categories increased. Although the

Table 8. Average entropy value for test documents (bigram model, doubled training set).

categories	money, business	quotidian	editorial	in the world	investigations	politics	sport
Money, business	**9,56**	10,36	10,65	10,51	10,33	10,27	10,61
quotidian	10,37	**10,05**	10,54	10,37	10,16	10,29	10,38
editorial	10,41	10,22	**6,32**	10,41	10,21	9,95	10,26
in the world	10,39	10,29	10,64	**6,14**	10,31	10,27	10,40
investigations	10,32	10,03	10,47	10,27	**9,53**	10,06	10,31
politics	10,18	10,23	10,07	10,22	10,05	**9,25**	10,28
sport	10,54	10,39	10,61	10,50	10,33	10,34	**8,98**

value changes are small, the training texts volume increase influenced the classification quality positively. In the previous experiment with the bigram model, five test documents from 'cotidian' category were not classified correctly. In the last experiment, eight of ten documents from this category were placed correctly and two were attributed to 'investigations'.

Thus, in the first experiment 93% of documents were classified correctly (65 of 70), in the last experiment 97% of documents were classified correctly (68 of 70).

8 Conclusion

In this paper, the application of compression word-based PPM language models to text classification has been described. The results of the implemented experiments proved that word-based PPM models are relevant for content-based text classification.

The PPM models based on trigrams, bigrams and unigrams have been compared. The results are quite promising for bigram and trigram models. Though trigram model performed slightly better, it required much more memory than bigram model.

Unigram model was not good enough. This showed that single words were not so characteristic features as word combinations for context-based text classification.

Although in some cases the entropy difference that influenced the choice was rather small (several hundredths), most of the documents (up to 97%) were classified correctly. Unfortunately we can not directly compare our results with other approaches because we tested our method on another corpus.

It should be mentioned that initially document categories in our experiments were not defined exactly, which produced difficulties while classifying.

We consider that the proposed method is appropriate for context-based text classification and it should be evaluated on standard test-bed for the evaluation of text categorization methods.

References

[1] J.G.Cleary and I.H.Witten. *A comparison of enumerative and adaptive codes.* IEEE Trans. Inf. Theory, IT-30, 2 (Mar.), 1984, pp. 306–315.

[2] J.G.Cleary and I.H.Witten. *Data compression using adaptive coding and partial string matching.* IEEE Trans. Commun. COM-32, 4 (Apr.), 1984, pp. 396–402.

[3] Thorsten Joachim. *Learning to Classify Text using Support Vector Mashine.* Methods, Theory, and Algorithms. Kluwer Academic Publishers, May 2002.

[4] Xuedong Huang, Fileno Alleva, Hsiao-wuen Hon, Mei-Yuh Hwang, Kai-Fu Lee and Ronald Rosenfeld. *The SPHINX-II Speech Recognition System: An Overview. Computer, Speech and Language,* volume 2, pp. 137–148, 1993.

[5] D.V.Khmelev, W.J.Teahan. *Verification of text collections for text categorization and natural language processing*: Tech. Rep. AIIA 03.1: School of Informatics, University of Wales, Bangor, 2003.

[6] O.Kukushkina, A.Polikarpov, D.Khmelev. *Using Letters and Grammatical Statistics for Authorship Attribution.* Problems of Information Transmission. 2001. Vol. 37, no. 2. pp. 172–184.

[7] A.Moffat. *Implementing the PPM data compression scheme.* IEEE Transaction on Communications, 38(11): pp. 1917-1921, 1990.

[8] Ronald Rosenfeld. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach,* Ph.D. thesis, Computer Science Department, Carnegie Mellon University,TR CMU-CS-94-138, April 1994.

[9] William John Teahan 1998. *Modelling English text.* PhD thesis, University of Waikato, 1998.

[10] Teahan and J.G.Cleary. *The entropy of English using PPM-based models.* In IEEE Data Compression Conference. IEEE Computer Society Press, 1996.

[11] S.Dumais, J.Platt, D.Heckermann, and M.Sahami. *Inductive learning algorithms and representations for text categorization.* In Proc. Intl. Conf. on Info. and Knowledge Management, pp. 148–155, 1998.

[12] D.D.Lewis and M.Ringuette. *A comparison of two learning algorithms for text categorization.* In Proc. Annual Symposium on Document Analysis and Information Retrieval, pp. 37–50, 1994.

[13] H.T.Ng, W.B.Goh, and K.L.Low. *Feature selection, perceptron learning, and a usability: case study for text categorization.* In Proc. ACM SIGIR, pp. 67–73, 1997.

[14] Ravi Kannan, Santosh Vempala, and Adrian Vetta. *On clusterings: good, bad and spectral.* In Proc. 41th IEEE Symp. on Foundations of Comp. Science, 2000.

[15] Andrej Bratko and Bogdan Filipic. *Spam Filtering Using Compression Models Technical Report IJS-DP 9227.* Department of Intelligent Systems, Jozef Stefan Institute, Ljubljana, Slovenia. December, 2005

[16] J.L.Bentley, D.D.Sleator, R.E.Tarjan, and V.K.Wei. *A locally adaptive data compression scheme.* Comm. of the ACM, 29(4):320-330, April 1986.

[17] A.Moffat. *Word based text compression.* Software - Practice and Experience, 19(2), pp. 185–198, 1989.

[18] R. Nigel Horspool and Gordon V. Cormack. *Constructing Word-Based Text Compression Algorithms.* Proceedings of Data Compression Conference (DCC'92), Snowbird, UT, March 1992, pp. 62–71.

[19] Nitin Thaper *Using Compression For Source Based Classification Of Text.* Bachelor of Technology (Computer Science and Engineering), Indian Institute of Technology, Delhi, India. 1996.

[20] Eibe Frank, Chang Chui and Ian H. Witten. *Text categorisation using compression models,* Proceedings of DCC-00, IEEE Data Compression Conference. 2000.

[21] C.E.Shannon. *A mathematical theory of communication.* Bell System Technical Journal, 27, 1948.

[22] W.J.Teahan. *Text classification and segmentation using minimum cross-entropy.* In Proceeding of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur", Paris, FR. 2000.

V. Bobicev, Received August 30, 2006

Technical University of Moldova
Phone: (373+2) 31–91–82
E–mail: *vika@rol.md*

Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

Traveling Salesman Problem with Transportation*

Valeriu Ungureanu

Abstract

Traveling Salesman Problem (TSP) is a generic name that includes diverse practical models. Motivated by applications, a new model of TSP is examined — a synthesis of classical TSP and classical Transportation Problem. Algorithms based on Integer Programming cutting-plane methods and Branch and Bound Techniques are obvious.

Mathematics Subject Classification 2000: 91A05, 91A06, 91A10, 91A44, 90C05, 90C31, 90C90.

Keywords and phrases: Traveling Salesman Problem with Transportation and Fixed Additional Payments (TSPT), TSP, algorithm, computational complexity.

1 Introduction

The TSP gained notoriety over the past century as the prototype of problem that is easy to state and hard to solve practically. It is simply formulated: *a traveling salesman has to visit exactly once each of n cities and to return to the start city, but in such order that the respective tour (Hamiltonian cycle) has the minimal total cost (it is supposed that the cost c_{ij} of traveling from every city i to every city j is known).* There are other related formulations of this problem and a lot of methods for their solving [1-5].

*The research is partially supported by the Award CERIM-1006-06 of the Moldovan Research and Development Association (MRDA) and US Civilian Research Development Foundation for the Independent States of the Former Soviet Union (CRDF).

TSP is representative for a large class of problems known as NP-complete combinatorial optimization problems. This class (of NP-complete problems) has an important property that all of them simultaneously have or don't have polynomial-time algorithms [6]. To date, no one has found efficient (polynomial-time) algorithm for the TSP solving. But, over the past few years many practical problems of really large size are solved. Thus, at present, a record exactly solved problem (with branch-and-cut algorithm) has 24 978 cities (Applegate, Bixby, Chvatal, and Cook - 2004).

The Transportation Problem is the well known classical problem [7]. There are several efficient methods for its solving [8, 9, 10].

TSP with Transportation and Fixed Additional Payments [11] generalizes these two problems.

2 Formulations of the TSPT.

Given an orgraf $G = (V, E), |V| = n, |E| = m$. Each node $j \in V$ has its own capacity δ_j (demand, if $\delta_j < 0$, supply, if $\delta_j > 0$), such that $\sum_{j=1}^{n} \delta_j = 0$. Salesman starts his traveling from node $k \in V$ with $\delta_k > 0$. A unit transportation cost through arc $(i, j) \in E$ is c_{ij}. If the arc (i, j) is active, then additional payment d_{ij} is demanded. If $(i, j) \notin E$, then $c_{ij} = d_{ij} = \infty$. Find the Hamiltonian cycle and the starting node $k \in V$ with the property that the respective salesman traveling satisfies all demands with minimal cost.

A TSPT may be formulated as integer programming problem. Let x_{ij} denotes the quantity of product which is transported via (i, j). Let $y_{ij} \in \{0; 1\}$ be equal 1 if $x_{ij} > 0$, and $x_{ij} = 0$ if $y_{ij} = 0$. In such notations the TSPT, as stated above, is equivalent to the following problem:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (c_{ij} x_{ij} + d_{ij} y_{ij}) \to \min, \tag{1}$$

$$\sum_{i=1}^{n} y_{ij} = 1, \quad j = \overline{1, n}, \tag{2}$$

$$\sum_{j=1}^{n} y_{ij} = 1, \quad i = \overline{2,n}, \tag{3}$$

$$\sum_{k=1}^{n} x_{jk} - \sum_{i=1}^{n} x_{ij} = \delta_j, \quad j = \overline{1,n}, \tag{4}$$

$$u_i - u_j + n y_{ij} \leq n - 1, \quad i,j = \overline{2,n},\ i \neq j, \tag{5}$$

$$x_{ij} \leq M y_{ij}, \quad i,j = \overline{1,n}, \tag{6}$$

$$x_{ij} \geq 0,\ y_{ij} \in \{0;1\},\ u_j \geq 0, \quad i,j = \overline{1,n}, \tag{7}$$

where $M = \sum_{j=1}^{n} |\delta_j|$.

If $c_{ij} \equiv 0,\ \forall i,j$, then (1)-(7) is the classical TSP. If $d_{ij} \equiv 0,\ \forall i,j$, then (1)-(7) is the classical Transportation Problem.

Theorem. *TSPT and problem (1)-(7) are equivalent.*

Proof. (1)-(3), (5) and (7) define Hamiltonian cycle [8]. (1), (4), (7) state the transportation problem [7]. (6) realizes the connection between both *"facets"* of the TSPT. The starting node $k \in V$ is determined by elementary searching. □

If each arc $(i,j) \in E$ has upper bound capacity $u_{ij} > 0$, we obtain another capacitated model of the problem. In this case inequalities

$$x_{ij} \leq u_{ij} y_{ij},\ (i,j) \in E,$$

are substitute for (6).

If each arc $(i,j) \in E$ has also a lower bound capacity $l_{ij} > 0$, then inequalities

$$l_{ij} y_{ij} \leq x_{ij} \leq u_{ij} y_{ij},\ (i,j) \in E,$$

are substitute for (6).

Kuhn [8] restrictions (5) may be substituted by equivalent restrictions:

$$\sum_{i \in K} \sum_{j \in K} y_{ij} = |K| - 1,\ \forall K \subset V,$$

where K is any proper subset of V.

3 Algorithms

It is obvious that the solution of the classical TSP does not solve TSPT.

The branch-and-bound algorithm may be constructed on backtracking technique for branch generation and the value of 1-tree plus T_0 for lower bound estimation, where T_0 is calculated at first step and represents the value of minimal cost flow problem obtained in relaxed problem without Hamiltonian cycle requirement. For efficient bounding, T_0 may be substituted at every step with exact cost of transportation through respective fragment of the cycle.

Direct solving (1)-(7) with Gomory type cutting-plane algorithms is rational for problem with modest size. In recent vogue opinion, the branch-and-cut super-algorithm [9, 10] may be much more recommended for TSPT.

Finally, note that dynamic programming approach comports difficulty as the TSPT optimal value depends really (practically) on first node from which the travel is starting. This fact may be simply taken in consideration in previous methods, but not in dynamic programming method.

References

[1] Lawler E.L., Lenstra J.K., Rinooy Kan A.H.G., and Shmoys D.B., eds., *The Traveling Salesman Problem,* John Wiley, Chichester, UK, 1985.

[2] Reinelt G., *The Traveling Salesman: Computational Solutions for TSP Applications,* Springer-Verlag, Berlin, 1994.

[3] Fleishner H., *Traversing graphs: The Eulerian and Hamiltonian theme,* M. Dror (ed.), Arc Routing: Theory, Solutions, and Applications, Kluwer Acad. Publ., 2000.

[4] Golden B.L. and Assad A.A., eds., *Vehicle Routing: Methods and Studies,* Elsevier Science, Amsterdam, 1988.

[5] Ball M.O., Magnanti T.L., Monma C.L., and Nemhauser G.L., eds., *Network Routing,* Elsevier, Amsterdam, 1995.

[6] Garey M.R. and Johnson D.S., *Computers and Intractability: a Guide to the Theory of NP Completeness,* W.H. Freeman, San Francisco, 1979.

[7] Hitchcock F.L., *The Distribution of Product from Several Sources to Numerous Localities,* J. Math. Phys., 20, No. 2, 1941, 217–224.

[8] Papadimitriou C.H. and Steiglitz K., *Combinatorial Optimization: Algorithms and Complexity,* Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982.

[9] Hoffman K.L. and Padberg M., *LP-based Combinatorial Problem Solving,* Annals Operations Research, 4, 1991, 145–194.

[10] Padberg M. and Rinaldi G., *A branch-and-cut algorithm for the resolution of large-scale traveling salesman problem,* SIAM Review, 33, 1991, 60–100.

[11] Ungureanu V., *TSP with transportation and fixed additional payments,* Achievements and Perspectives of Research in the Economic Field, Abstracts of International Conference on Economic Cybernetics, Bucharest, 2002.

V. Ungureanu
Received May 11, 2006

State University of Moldova
60, A. Mateevici str.
Chişinău, MD−2009, Moldova
E–mail: *valungureanu@usm.md*

Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

On quasi-stability of the vector Boolean problem of minimizing absolute deviations of linear functions from zero*

Vladimir A. Emelichev, Evgeny E. Gurevsky

Abstract

We consider a multi-criterion Boolean programming problem with partial criteria of the kind MIN MODUL of linear functions. We investigate such type of stability which can be understood as a discrete analogue of the Hausdorff lower semi-continuity. A formula of the quasi-stability radius is obtained.

Mathematics Subject Classification 2000: 90C09, 90C29, 90C31.

Keywords and phrases: vector Boolean programming problem, Pareto set, quasi-stability, quasi-stability radius.

1 Introduction

Let us consider a vector (m-criterion) Boolean programming problem with a finite solution set X

$$Z^m(A,b): \ \min\{f(x,A,b): \ x \in X\},$$

where

$$f(x,A,b) = (|A_1x+b_1|, |A_2x+b_2|, \dots, |A_mx+b_m|),$$

$X \subseteq \mathbf{E}^n = \{0,1\}^n, \ |X| \geq 2, \ n \geq 2, \ A_i$ denotes the i–th row of matrix $A = [a_{ij}]_{m\times n} \in \mathbf{R}^{m\times n}, \ i \in N_m = \{1,2,\dots,m\}, \ m \geq 1, \ b = (b_1,b_2,\dots,b_m)^T \in \mathbf{R}^m, \ x = (x_1,x_2,\dots,x_n)^T$.

*Support by State program of fundamental research "Mathematical structures" (Grant 913/28) and program of the Ministry of Education "Fundamental and application studies" (Grant 492/28) of the Republic of Belarus.

By problem $Z^m(A,b)$ we understand the problem of finding the set of efficient solutions (Pareto set)

$$P^m(A,b) = \{x \in X : X_x(A,b) = \emptyset\},$$

where

$$X_x(A,b) = \{x' \in X : f(x,A,b) \geq f(x',A,b) \ \& \ f(x,A,b) \neq f(x',A,b)\}.$$

By virtue of inequalities $1 < |X| < \infty$, the set $P^m(A,b)$ is nonempty for any $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$.

Note, that the vector function $f(x,A,b)$ shows a measure of inconsistency of the following system of linear boolean equalities

$$Ax + b = \mathbf{0}^{(m)}, \quad x \in X. \tag{1}$$

where $\mathbf{0}^{(m)} = (0,0,\ldots,0) \in \mathbf{R}^m$.

Thus, minimization of function $|A_i x + b_i|$ on the finite set X is equivalent to the minimization of absolute deviations of linear functions $A_i x + b_i$ from zero. Therefore, problem $Z^m(A,b)$ is reduced to finding the set of all solutions to system (1) under the condition that this system is consistent. Otherwise, Pareto set $P^m(A,b)$ can be considered as the set of quasi solutions to system (1). It is easy to see that the system of equalities is consistent if and only if the set of efficient vector estimates

$$f(P^m(A,b)) = \{y \in \mathbf{R}^m : \ y = f(x,A,b), \ x \in P^m(A,b)\}$$

contains only zero vector $\mathbf{0}^{(m)}$.

For each number $k \in \mathbf{N}$, we endow the space $\mathbf{R}^k$ with two metrics l_1 and l_∞, i.e. we define the norms of a vector $z = (z_1, z_2, \ldots, z_k) \in \mathbf{R}^k$ as follows:

$$||z||_1 = \sum_{j \in N_k} |z_j|, \quad ||z||_\infty = \max_{j \in N_k} |z_j|.$$

Under a norm of matrix we understand the norm of vector, composed from all its elements.

For any positive number $\varepsilon > 0$, we define the set of perturbing pairs

$$\Omega(\varepsilon) = \{(A', b') \in \mathbf{R}^{m\times(n+1)} : \max\{||A'||_\infty, ||b'||_\infty\} < \varepsilon\}.$$

As usual [1 - 6], problem $Z^m(A, b)$ is called quasi-stable, if the set

$$\Xi = \{\varepsilon > 0 : \forall\ (A', b') \in \Omega(\varepsilon)\ (P^m(A, b) \subseteq P^m(A + A', b + b'))\} \neq \emptyset.$$

Therefore, problem $Z^m(A, b)$ is quasi-stable, if there exists a neighborhood of pair (A, b) in the space of perturbing parameters $\mathbf{R}^{m\times(n+1)}$ with metric l_∞, in which Pareto optimality of all solutions is preserved. Note, that the quasi-stability of the problem is a discrete analogue of the Hausdorff lower semi-continuity [7] of the optimal mapping at point (A, b)

$$P^m :\ \mathbf{R}^{m\times(n+1)} \to 2^X,$$

i.e. of the point-to-set (many-valued) mapping, that puts in correspondence the Pareto set $P^m(A, b)$ to the collection of parameters (pair (A, b)) of the problem.

As a result of the said above, under the quasi-stability radius of $Z^m(A, b)$ we will understand the number

$$\rho^m(A, b) = \begin{cases} \sup\ \Xi, & \text{if } \Xi \neq \emptyset, \\ 0, & \text{if } \Xi = \emptyset. \end{cases}$$

For any $x, x' \in X$ and $i \in N_m$, put

$$\alpha_i(x, x') = \min\{\beta_i(x, x', s) :\ s \in \{-1; 1\}\},$$

$$\beta_i(x, x', s) = \frac{|A_i(sx + x') + b_i(s + 1)|}{||sx + x'||_1 + |s + 1|},$$

$$K(x, x') = \{i \in N_m :\ |A_i x + b_i| \leq |A_i x' + b_i|\}.$$

It is evident, that $K(x, x') \neq \emptyset$, if $x \in P^m(A, b)$. For any number $z \in \mathbf{R}$, denote

$$\text{sg } z = \begin{cases} 1, & \text{if } z \geq 0, \\ -1, & \text{if } z < 0. \end{cases}$$

We will also use the following implication:

$$\exists s \in \{-1;1\} \quad \forall s' \in \{-1;1\} \quad (sz > s'z') \Rightarrow |z| > |z'|, \tag{2}$$

which holds for any numbers $z, z' \in \mathbf{R}$.

2 The Results

Theorem. *The quasi-stability radius of problem $Z^m(A,b)$ is expressed by the formula*

$$\rho^m(A,b) = \min_{x \in P^m(A,b)} \min_{x' \in X \setminus \{x\}} \max_{i \in K(x,x')} \alpha_i(x,x'). \tag{3}$$

Proof. Denote by φ the right side of (3). It is easy to see, that $\varphi \geq 0$.

At first we will prove the inequality $\rho^m(A,b) \geq \varphi$. Suppose $\varphi > 0$ (this inequality is evident where $\varphi = 0$). Let $(A',b') \in \Omega(\varphi)$. Then by definition of φ, for any solutions $x \in P^m(A,b)$ and $x' \in X\setminus\{x\}$, there exists $k \in K(x,x')$ such that

$$\max\{||A'||_\infty, ||b'||_\infty\} < \varphi \leq \alpha_k(x,x'). \tag{4}$$

Taking into account $\alpha_k(x,x') > 0$, we have

$$|A_k x + b_k| < |A_k x' + b_k|.$$

From this, assuming $\sigma = \text{sg}\ (A_k x' + b_k)$, we obtain that the following equalities hold:

$$A_k(sx + \sigma x') + b_k(\sigma + s) = |A_k(\sigma s x + x') + (1 + \sigma s) b_k|, \ s \in \{-1;1\}.$$

Therefore, applying (4), we derive

$$s((A_k + A'_k)x + (b_k + b'_k)) + \sigma((A_k + A'_k)x' + (b_k + b'_k)) =$$

$$= |A_k(\sigma s x + x') + (1 + \sigma s) b_k| + \sigma(A'_k(\sigma s x + x') + b'_k(1 + \sigma s)) \geq$$

$$\geq |A_k(\sigma s x + x') + (1 + \sigma s) b_k| - (||A'||_\infty ||\sigma s x + x'||_1 +$$

$$+||b'||_\infty|1+\sigma s|) \geq |A_k(\sigma sx + x') + (1+\sigma s)b_k| -$$

$$-\max(||A'||_\infty, ||b'||_\infty)(||\sigma sx + x'||_1 + |1+\sigma s|) >$$

$$> |A_k(\sigma sx + x') + (1+\sigma s)b_k| -$$

$$-\alpha_k(x,x')(||\sigma sx + x'||_1 + |1+\sigma s|) \geq$$

$$\geq |A_k(\sigma sx + x') + (1+\sigma s)b_k| -$$

$$-\beta_k(x,x',\sigma s)(||\sigma sx + x'||_1 + |1+\sigma s|) = 0.$$

Thus we have

$$\sigma((A_k + A'_k)x' + (b_k + b'_k)) > s((A_k + A'_k)x + (b_k + b'_k)),\ s \in \{-1; 1\}.$$

Therefore, taking into account (2), we obtain

$$|(A_k + A'_k)x' + (b_k + b'_k)| > |(A_k + A'_k)x + (b_k + b'_k)|,$$

which implies $x \in P^m(A + A', b + b')$.

Summarizing the said above, we conclude that for any $x \in P^m(A, b)$ and for any perturbing pair $(A', b') \in \Omega(\varphi)$ the solution $x \in P^m(A + A', b + b')$. Hence, $\rho^m(A, b) \geq \varphi$.

It remains to prove inequality $\rho^m(A, b) \leq \varphi$. Using the definition of number φ we conclude that there exist such solutions $x^0 \in P^m(A, b),\ x^* \in X\backslash\{x^0\}$, that for any index $i \in K(x^0, x^*)$ the following inequality holds:

$$\alpha_i(x^0, x^*) \leq \varphi. \tag{5}$$

Let $\varepsilon > \varphi$. Let us show, that there exists perturbing pair $(A', b') \in \Omega(\varepsilon)$ satisfying the condition $x^0 \notin P^m(A + A', b + b')$.

Suppose

$$N(x^0, x^*) = |\{j \in N_n :\ x^0_j = 1 \ \&\ x^*_j = 0\}|,$$

$$\sigma^0 = \mathrm{sign}(A_i x^0 + b_i),$$

$$\sigma^* = \mathrm{sign}(A_i x^* + b_i).$$

It is easy to see that at least one of the numbers $N(x^0, x^*)$ or $N(x^*, x^0)$ is positive and

$$\max\{N(x^*, x^0),\ N(x^0, x^*)\} \leq ||x^0 + x^*||_1. \tag{6}$$

To build all the rows

$$(A'_i, b'_i), \quad i \in N_m$$

of the needed perturbing pair (A', b'), consider four possible cases.

Case 1: $i \in K(x^0, x^*)$, $\beta_i(x^0, x^*, -1) < \beta_i(x^0, x^*, 1)$. Then

$$0 < \beta_i(x^0, x^*, 1) \leq \frac{|A_i x^0 + b_i| + |A_i x^* + b_i|}{||x^0 + x^*||_1 + 2} \tag{7}$$

and under inequality (5) there exists a number $\delta_i < \varepsilon$ such that

$$0 \leq \beta_i(x^0, x^*, -1) < \delta_i < \beta_i(x^0, x^*, 1). \tag{8}$$

From this, assuming

$$A'_i = (a'_{i1}, a'_{i2}, \ldots, a'_{in}),\ b'_i = 0,$$

where

$$a'_{ij} = \begin{cases} \sigma^0 \delta_i, & \text{if } x^0_j = 1,\ x^*_j = 0, \\ -\sigma^* \delta_i, & \text{if } x^0_j = 0,\ x^*_j = 1, \\ 0 & \text{in the other cases,} \end{cases} \tag{9}$$

we have $\max\{||A'_i||_\infty, |b'_i|\} = \delta_i$ and

$$\begin{aligned} &\sigma^0((A_i + A'_i)x^0 + (b_i + b'_i)) - \sigma^*((A_i + A'_i)x^* + (b_i + b'_i)) = \\ &= |A_i x^0 + b_i| - |A_i x^* + b_i| + \delta_i(N(x^0, x^*) + N(x^*, x^0)) \geq \\ &\geq -|A_i(x^* - x^0)| + \delta_i ||x^* - x^0||_1 > \\ &> -|A_i(x^* - x^0)| + \beta_i(x^0, x^*, -1)||x^* - x^0||_1 = 0, \end{aligned} \tag{10}$$

$$\begin{aligned} &\sigma^0((A_i + A'_i)x^0 + (b_i + b'_i)) + \sigma^*((A_i + A'_i)x^* + (b_i + b'_i)) = \\ &= |A_i x^0 + b_i| + |A_i x^* + b_i| + \delta_i(N(x^0, x^*) - N(x^*, x^0)). \end{aligned} \tag{11}$$

The right side of the last equality is denoted by ψ. If $N(x^*, x^0) = 0$, then in view of (7) we have $\psi > 0$. If $N(x^*, x^0) > 0$, then taking into account (6), (7) and (8), we have

$$\delta_i N(x^*, x^0) < |A_i x^0 + b_i| + |A_i x^* + b_i|.$$

Therefore we obtain

$$\psi > \delta_i N(x^0, x^*) \geq 0.$$

Thus $\psi > 0$ and we have

$$\sigma^0((A_i + A_i')x^0 + (b_i + b_i')) > s\sigma^*((A_i + A_i')x^* + (b_i + b_i')),\ s \in \{-1; 1\}.$$

From this, applying (2) we find, that

$$|(A_i + A_i')x^0 + (b_i + b_i')| > |(A_i + A_i')x^* + (b_i + b_i')|. \qquad (12)$$

Case 2: $i \in K(x^0, x^*),\ \beta_i(x^0, x^*, -1) > \beta_i(x^0, x^*, 1)$. Then under (5) there exists a number $\delta_i < \varepsilon$ with conditions

$$0 \leq \beta_i(x^0, x^*, 1) < \delta_i < \beta_i(x^0, x^*, -1).$$

From this, denoting

$$A_i' = (-\sigma^*\delta, -\sigma^*\delta, \ldots, -\sigma^*\delta) \in \mathbf{R}^n,\ b_i' = -\sigma^*\delta,$$

we have $\max\{||A_i'||_\infty, |b_i'|\} = \delta_i$ and

$$-\sigma^*((A_i + A_i')x^0 + (b_i + b_i')) - \sigma^*((A_i + A_i')x^* + (b_i + b_i')) =$$
$$= -\sigma^*(A_i(x^0 + x^*) + 2b_i) + \delta_i(||x^0||_1 + ||x^*||_1 + 2) >$$
$$> -|A_i(x^0 + x^*) + 2b_i| + \beta_i(x^0, x^*, 1)(||x^0 + x^*||_1 + 2) = 0,$$
$$-\sigma^*((A_i + A_i')x^0 + (b_i + b_i')) + \sigma^*((A_i + A_i')x^* + (b_i + b_i')) =$$
$$= \sigma^* A_i(x^* - x^0) - \delta_i(||x^*||_1 - ||x^0||_1) =$$
$$= |A_i(x^* - x^0)| - \delta_i(||x^*||_1 - ||x^0||_1) >$$
$$> |A_i(x^* - x^0)| - \beta_i(x^0, x^*, -1)||x^* - x^0||_1 = 0.$$

Thus the following equalities hold

$$-\sigma^*((A_i+A_i')x^0+(b_i+b_i')) > s\sigma^*((A_i+A_i')x^*+(b_i+b_i')),\ s\in\{-1;1\}$$

Then, applying (2), we obtain (12).

Case 3: $i \in K(x^0,x^*)$, $\beta_i := \beta_i(x^0,x^*,-1) = \beta_i(x^0,x^*,1) = \alpha_i(x^0,x^*)$. Then

$$\beta_i \le \frac{|A_ix^0+b_i|+|A_ix^*+b_i|}{||x^0+x^*||_1+2}. \tag{13}$$

Consider two variants.

At first let $\beta_i = 0$. Then it is easy to see, that

$$A_ix^0+b_i = A_ix^*+b_i = 0. \tag{14}$$

If $N(x^0,x^*)>0$, then, specifying components of the vector

$$A_i' = (a_{i1}', a_{i2}', \dots, a_{in}')$$

by the rule

$$a_{ij}' = \begin{cases} \delta_i, & \text{if } x_j^0=1,\ x_j^*=0,\\ 0 & \text{in the other cases,}\end{cases}$$

$$b_i' = 0,$$

where

$$0 \le \varphi < \delta_i < \varepsilon,$$

we have $\max\{||A_i'||_\infty, |b_i'|\} = \delta_i$ and from (14), inequality (12) holds.

If $N(x^0,x^*)=0$, then $N(x^*,x^0)>0$, i.e. we may choose index $k\in N_n$ such that

$$x_k^*=1,\quad x_k^0=0.$$

Then, assuming

$$A_i' = (a_{i1}', a_{i2}', \dots, a_{in}'),\ b_i' = \delta_i$$

where

$$a_{ij}' = \begin{cases} -\delta_i/2, & \text{if } j=k,\\ 0 & \text{in the other cases ,}\end{cases}$$

$$0 \leq \varphi < \delta_i < \varepsilon,$$

we have, that from (14) the inequality (12) holds, and also $\max\{||A'_i||_\infty, |b'_i|\} = \delta_i$.

Now let $\beta_i > 0$. Assuming,

$$\beta_i < \delta_i < \varepsilon, \tag{15}$$

we build the row A'_i by formula (9), and assume b'_i to equal zero. Then $\max\{||A'_i||_\infty, |b'_i|\} = \delta_i$ and the relations (10) and (11) hold. As in the case 1, we can show, that $\psi > 0$. If $N(x^*, x^0) = 0$, then from (13) we have $\psi > 0$. If $N(x^*, x^0) > 0$, then taking into account (6) and (13) we may additionally to condition (15) impose the following condition on the number δ_i:

$$\delta_i N(x^*, x^0) < |A_i x^0 + b_i| + |A_i x^* + b_i|.$$

Therefore (see case 1) $\psi > \delta_i N(x^0, x^*) \geq 0$. Hence, we have (12).

Case 4: $i \in N_m \backslash K(x^0, x^*)$. Then, assuming $A'_i = \mathbf{0}^{(n)},\ b'_i = 0$, we have (12).

As a result of the above reasoning, we obtain a pair $(A', b') \in \Omega(\varepsilon)$ such that $x^* \in X_{x^0}(A+A', b+b')$. Thus for solution $x^0 \in P^m(A, b)$ and for any $\varepsilon > \varphi$ there exists a perturbing pair $(A', b') \in \Omega(\varepsilon)$ such that $x^0 \notin P^m(A+A', b+b')$. Consequently $\rho^m(A, b) \leq \varphi$.

That completes the proof.

Let $S^m(A, b)$ (otherwise – the set of strictly efficient solutions) be Smale set [8] defined by the rule

$$x \in S^m(A, b) \ \Leftrightarrow \ \{x' \in X \backslash \{x\} : \ f(x, A, b) \geq f(x', A, b)\} = \emptyset.$$

It is evident that the inclusion $S^m(A, b) \subseteq P^m(A, b)$ holds for any A, b.

Corollary. *The problem $Z^m(A, b)$ is quasi-stable if and only if*

$$P^m(A, b) = S^m(A, b). \tag{16}$$

Proof. Sufficiency. Let equality (16) holds. Then, for any solutions $x \in P^m(A,b)$ and $x' \in X\backslash\{x\}$ there exists an index $k \in N_m$, such that

$$|A_k x + b_k| < |A_k x' + b_k|,$$

which implies $k \in K(x,x')$.
Therefore, taking into account evident relations

$$\beta_k(x,x',s) = \frac{|A_k(sx + x') + b_k(s+1)|}{||sx + x'||_1 + |s+1|} \geq$$

$$\geq \frac{|A_k x' + b_k| - |A_k x + b_k|}{||sx + x'||_1 + |s+1|} > 0, \quad s \in \{-1;1\},$$

we conclude that

$$\max\{\alpha_i(x,x') : \ i \in K(x,x')\} > 0.$$

On the basis of the theorem we have $\rho^m(A,b) > 0$, i.e. problem $Z^m(A,b)$ is quasi-stable.

We prove the **necessity** by contradiction. Let there exists a solution $x^0 \in P^m(A,b)\backslash S^m(A,b)$ under the assumption, that $Z^m(A,b)$ is quasi-stable. Then there exists a solution $x^* \neq x^0$, such that

$$|A_i x^0 + b_i| = |A_i x^* + b_i|, \quad i \in N_m.$$

Therefore

$$\forall i \in N_m \ \ \exists s \in \{-1;1\} \ \ \beta_i(x^0,x^*,s) = \frac{|A_i(sx^0 + x^*) + b_i(s+1)|}{||sx^0 + x^*||_1 + |s+1|} = 0.$$

From this we conclude

$$\max\{\alpha_i(x^0,x^*) : \ i \in K(x^0,x^*)\} = 0.$$

Consequently, on the basis of the theorem we obtain $\rho^m(A,b) = 0$, i.e. problem is not quasi-stable.

Corollary 1 implies

Corollary 2. *The scalar problem* $Z^1(A,b)$, *where* $A \in \mathbf{R}^n, b \in \mathbf{R}$, *is quasi-stable if and only if*

$$|P^1(A,b)| = |\text{Argmin}\{|Ax+b| : \ x \in X\}| = 1.$$

References

[1] Emelichev V. A., Kravtsov M. K., *Stability in vector optimization path problems*, Cybernetics and System Analyses, 1995, vol. 31, pp. 599 – 604.

[2] Emelichev V. A., Nikulin Yu. V. *On stability and quasi-stability of a vector lexicographic quadratic boolean programming problem*, Revue d'Analyse Numérique et de Théorie de l'Approximation, Cluj-Napoca, Romania, 2001, t. XXX, no. 1, pp. 35 – 46.

[3] Emelichev V. A., Girlich E., Nikulin Yu. V., Podkopaev D. P. *Stability and regularization of vector problems of integer linear programming*, Optimization, 2002, vol. 51, no. 4, pp. 645 – 676.

[4] Emelichev V. A., Kuz'min K. G., Leonovich A. M. *Stability in the combinatorial vector optimization problems*, Automatic and Remote Control, 2004, vol. 65, no. 2, pp. 227 – 240.

[5] Emelichev V. A., Kuz'min K. G. *The stability radius of efficient solution to a vector problem of boolean programming in the* l_1 *metric*, Doclady Mathematics, 2005, vol. 71, no. 2, pp. 266 – 268.

[6] Emelichev V. A., Karelkina O. V., Kuz'min K. G. *Measure of stability of a Pareto optimal solution to a vector integer programming problem with fixed surcharges in the* l_1 *and* l_∞ *metrics*, Computer Science Journal of Moldova, 2005, vol. 13, no. 2, pp. 177 – 192.

[7] Tanino T., Sawaragi Y. *Stability of nondominated solutions in multicriteria decision-making*, Journal of Optimization Theory and Applications, 1980, vol. 30, no. 2, pp. 229 – 253.

[8] Smale S. *Global analysis and economics, V. Pareto theory with constraints*, J. Math. Econom., 1974, vol. 1, no. 3, pp. 213 – 221.

V. A. Emelichev, E. E. Gurevsky,
Received January 24, 2006

Belarussian State University,
ave. Independence, 4,
Minsk, 220050, Belarus.
E–mails: *emelichev@bsu.by*, *emelichev@tut.by*, *Eugen_Eugen@tut.by*

Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

Query by Constraint Propagation in the Concept-Oriented Data Model

Alexandr Savinov

Abstract

The paper describes an approach to query processing in the concept-oriented data model. This approach is based on imposing constraints and specifying the result type. The constraints are then automatically propagated over the model and the result contains all related data items. The simplest constraint propagation strategy consists of two steps: propagating down to the most specific level using de- projection and propagating up to the target concept using projection. A more complex strategy described in the paper may consist of many de-projection/projection steps passing through some intermediate concepts. An advantage of the described query mechanism is that it does not need any join conditions because it uses the structure of the model for propagation. Moreover, this mechanism does not require specifying an access path using dimension names. Thus even rather complex queries can be expressed in simple and natural form because they are expressed by specifying what information is available and what related data we want to get.

1 Introduction

Let us consider the following problem domain and design of its data model. Customers order goods using orders. One order consists of several parts where each part is one product item. This can be represented by three tables `Orders`, `OrderParts` and `Products` as shown in Fig. 1. Each record from `OrderParts` references one order it belongs to and one product that has been ordered in this item (denoted by arrows). For example, order part `<#23, #16>` references order `<#23>` and product `<#16>` (descriptive fields such as product name or order date are not shown).

One general problem that arises in the context of data modelling consists in getting *related* records given some other records and additional constraints. For example, we might want to get all orders related to one or more products or, vice versa, all products related to a selected set of orders. Notice that related records are not directly connected because there is an intermediate table `OrderParts` between tables `Orders` and `Products`. In the real world data models the structure of relationships is much more complex and normally there exist more than one intermediate table. In this case it is difficult to automatically retrieve related records because the constraints need to be unambiguously propagated over the model according to the assumed semantics of its relationships.

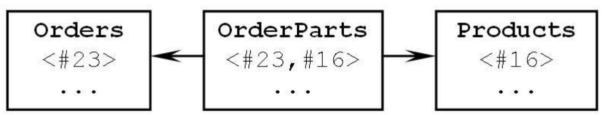

Figure 1. An example of orders and products

A wide spread approach to this problem is based on using the relational data model (RM) [4]. It consists in manually specifying all the involved tables and all their join conditions. Essentially in this case we precisely specify what kind of data we want to retrieve and how this data is related to other data in the model. Notice that all this information is provided within one query. For large data models such a method is not only tedious but also error-prone because writing long queries with complex join conditions requires high expertise. In our almost primitive example it is necessary to join two pairs of tables by choosing among many possible query options. The database cannot help us too much because it is unaware of the model semantics and what actually we want to get and hence all the peculiarities and details of the query have to be specified in an explicit form.

Another problem of such an approach is that many queries will contain the same information. For example, joining the two pairs of tables will be normally done using identical fragments of the query. These fragments will be then repeated in the same or similar form in many other queries. If in future we change a relationship between some tables then all these fragments need to be updated (the problem of code maintenance). Because of this necessity to provide a very detailed manual description for each query intended to retrieve related records, RM is not very successful for solving the problem of *logical* navigation. (Its main achievement is that it successfully solved the problem of *physical* navigation eliminating the need to know and specify physical location of records what was necessary in the preceding data models).

An alternative solution of the problem of logical navigation is provided within the concept-oriented model (CoM) [13, 15]. This model is based on *ordering* its elements, i.e., elements are positioned one under another without cycles rather than compose an arbitrary graph like in entity-relationship model (ERM) [3]. In other words, it is of primary importance if an element is above or below another element while other properties are derived from this ordered structure. In particular, if a table references some other table then it is positioned below it. For example, since table `OrderParts` references `Orders` and `Products`, we put it below the both of these tables as shown in Fig. 2. In addition, we provide names for the arrows connecting tables, which are called dimensions (all dimensions have upward direction). Thus CoM is not only aware of the structure but its direct responsibility consists in managing and using it for querying data, logical navigation and other operations with data.

Using ordered graphs it is possible to get related data by specifying an exact access path from constraints to target elements as dimension names. A direct advantage is that we avoid any join conditions which can be thought of as encoded in the model dimension structure. Such queries are much simpler than SQL queries because they do not involve repeated fragments. For example, a set of orders related to products `P` is obtained as follows:

`P` $\rightarrow$ {`OrderParts` $\rightarrow$ `product`} $\rightarrow$ `order`

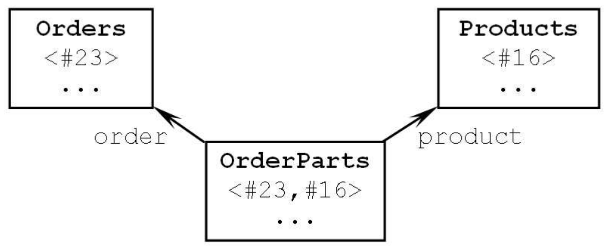

Figure 2. In CoM all elements are ordered

If there is a set of orders `O` then related products are obtained as follows:

$\texttt{O} \to \{\texttt{OrderParts} \to \texttt{order}\} \to \texttt{product}$

In both queries we de-project the initial set to `OrderParts` (denoted by dimension path in curly brackets). And then the result is projected to the target table. For more complex structures such an access path is longer and may consist of many upward (projection) and downward (de-projection) segments in the ordered table structure (zigzags). But in any case the use of dimensions effectively removes the need for explicit joins. It is the responsibility of the database to translate the access path into appropriate operations with records.

It can be noticed that a similar approach is used in other navigational data models such as the network model (NM), object-oriented model (OOM) or functional data model (FDM) [17, 7, 8]. However, the big difference is that CoM uses an *ordered* structure rather than an arbitrary graph and this fact has significant consequences. One of them is that the access path is not simply a sequence of dimensions but rather a sequence of projection and de-projection operations. If we go up in the ordered set then it is projection while if we go down then it is de-projection. These operations can also be effectively used for grouping and aggregation [16] what makes it similar to multidimensional models [1, 9, 11], OLAP [2] and formal concept analysis (FCA) [6].

Above we shortly described how related records can be retrieved in CoM by specifying an access path. However, an amazing property of this model is that in many cases the access path can be built automatically using for that purpose the model structure. In this case we need to only provide a set of initial items and some target from where related records have to be retrieved. In our example this means that it is enough to specify initial products (say, by imposing some constraints on table `Products`) and then say that we want to get related orders. Or, we can specify some initial orders and then say that it is necessary to get related products. In both cases the system will be able to unambiguously build the correct access path and retrieve related records with no additional information. Such ability is again based on using order of elements. Particularly, the system knows that `OrderParts` is a common subtable for both `Orders` and `Products`. Hence it builds the access path as consisting of one de-projection and one projection. This principle can be generalized and applied to more complex data models where it is frequently possible to automatically propagate constraints and get related records expected by the user.

This problem was considered within the universal relation model (URM) where all relations are assumed to be projections of a single relation [10, 5, 12]. However, this direction did not result in an acceptable solution because some things become simpler while others become more complex. One reason is that an assumption of universal relation was shown to be incompatible with many aspects of the relational model. In contrast, CoM possesses a number of unique properties which make the solution not only possible but very natural. One of them is that CoM has canonical semantics (an analogue of universal relation) as its intrinsic feature, which allows us to define natural constraint propagation rules.

The main goal of this paper consists in describing how the mechanism of automatic constraint propagation works in CoM and how it can be used for getting related records. In Section 2 CoM is shortly described. Section 3 describes operations of projection and de-projection and how they are used for constraint propagation. Section 4 describes some more complex cases where it is not possible to find an access path for constraint propagation. Section 5 provides concluding remarks.

2 Model Structure

One of the main principles of the concept-oriented paradigm is that of *duality*, which means that any element has two sides or flavours. For example, in the concept-oriented programming (CoP), concept is a programming construct consisting of one object class *and* one reference class [14]. An object in CoP is then a combination of its fields *and* a collection of internal objects. In CoM the principle of duality can be formulated as follows: any element participates in two separate structures called *physical* and *logical* (Fig. 3). The physical structure has a hierarchical form where any element has a single parent element (dotted lines in Fig. 3). It is called physical because elements are assumed to be included by value in their parents and the element position in the parent element is thought of as its (local) identifier or reference. Thus the physical structure is responsible for object representation and access (ORA) by providing permanent references and some access procedure. For example, all product records are created within a table where they have some unique and permanent references. The table itself is included in its database.

In CoM any element, such as product, may have its internal (physical) elements however in this paper we consider only a *two-level model* which consists of one *root*, a set of *concepts* included in the root and a set of data *items* included in the concepts. (Further in the paper we will use these terms instead of their analogues like tables and records.) In Fig. 3 the physical structure is drawn along the horizontal axis and consists of one root, which includes three concepts `OrderParts`, `Orders` and `Products`, which in turn consist of some data items.

The logical structure has a form of directed acyclic graph where an element may have many parents by storing their references in its fields. In other words a reference stored in a field is interpreted as a pointer to one of the parent elements. This allows us to bring order into the logical structure by placing an attribute value above the element it characterizes. For example, order part item `<#23, #16>` references order item `<#23>` and product item `<#16>`. Hence it is positioned below both of them. Here we do not need even to know that these items represent an

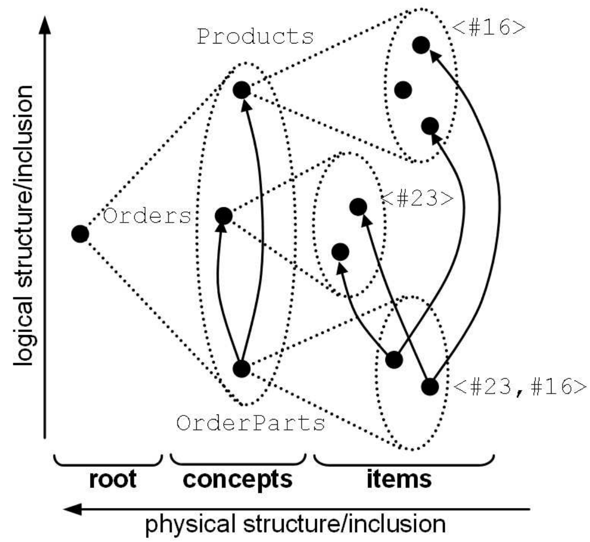

Figure 3. Physical and logical structures

order part, order and product – we simply interpret references (arrows in Fig. 3) as the logical membership relation leading from a member to its group. Such an interpretation of references (and attribute-value relation) as a membership is a very important characteristic property of CoM which allows us to consider a value as a logical set or collection for the objects it characterizes (an item is a logical collection for its subitems).

Concept S referenced by concept C is referred to as *superconcept* (and C is then referred to as *subconcept*). For example, concept `OrderParts` has two superconcepts `Orders` and `Products` positioned above their common subconcept. A named link from a subconcept to a superconcept is referred to as *dimension*. For example, concept `OrderParts` in Fig. 2 has two dimensions `order` and `product`. If a concept does not have a subconcept explicitly defined then formally introduced *bottom* concept is used for that. Analogously, if a concept does not have a superconcept then formally introduced *top* concept is used for that. Top concept is the most abstract (general) concept and bottom concept is the most specific one. All other concepts in the model are positioned between top and bottom concepts. Direct subconcepts of top concept are referred to as *primitive* concepts.

In the logical structure any element references a number of parent elements. Theoretically an element can reference any other element provided that this structure does not have cycles. (Cycles do not allow us to define order because it is not possible to determine if an element is above or below another element from this cycle.) In particular, concepts can reference items, items can reference concepts and any item can reference any other item from the model. However, such a freedom is not desirable and CoM introduces so called *syntactic constraints*, which mean that an item can reference only items from concepts referenced by its parent concept. For example, an order part item cannot reference any other item in the model under syntactic constraints. Its domains are restricted by items from `Orders` and `Products` because they are referenced by concept `OrderParts`.

There exist two operations that can be used to get related items given some other items: projection and de-projection [16]. *Projection* is applied to items from a subconcept and returns items from some its superconcept along a dimension path. For example, `OrderParts` is a subconcept of `Products` and hence each order part item can be projected on some product. Given a subset of order parts `OP` it is possible to get all related products by projecting this subset into its superconcept `Products`:

`OP` $\rightarrow$ `product`

De-projection is an opposite operation. It is applied to items from a superconcept and returns related items from some its subconcept. A path in the case is specified using dimension names which however are interpreted in the opposite direction. For example, given a set of products `P` we can get all related order parts by de-projecting them into concept `OrderParts`:

`P` $\rightarrow \{$`OrderParts` $\rightarrow$ `product`$\}$

Notice that here we use *inversion operator* $\{\,\}$ in order to change the direction of dimension path. Thus projection can be viewed as moving up in the concept graph along a dimension path while de-projection is viewed as moving down along some *inverse dimension*. A sequence of projection and de-projection operators is referred to as an *access path*.

3 Constraint Propagation

In the previous section we described a procedure for retrieving related items using the mechanism of access path. This method assumes that some initial set of items is selected and then operations of projection and de-projection are applied. This method is simpler and more natural than the existing approaches but it still requires a complete specification of the access path.

Let us now look at this process differently. The set of initial items has to be somehow specified and for this purpose some constraints are normally used. These initial constraints are imposed on items of some concept and are somehow propagated over the model changing its semantics. Finally we want to get some part of this modified (constrained) model by specifying the target concept. Due to constraint propagation this target concept will include only items which are related to the source items. An advantage of this approach is that we do not need to specify an access path because related items are retrieved as a result of constraint propagation procedure carried out automatically. For example, to get products related to some orders it is enough to specify constraints for the source orders and then the target concept – the constraint propagation path can be computed automatically.

The main question in this method is *how* initial constraints have to be propagated over the model in order to get natural and meaningful results. The simplest strategy consists of two operations:

1. Initial constraints imposed on concepts $X_1, \dots, X_n$ are propagated down to and imposed on the most specific concept Z using de-projection.

2. The constrained semantics of concept Z is propagated up to the target concept Y using projection.

Constraints are specified using some predicate that has to be true for each selected data item:

$\bar{X}_i = \{x_i \in X_i | f_i(x_i) = \text{true}\} \subseteq X_i,\ i = 1, \dots, n$

Here $\bar{X}_i$ is a subset of items from X_i satisfying condition f_i (bar stands for constrained).

Let us now consider how these constraints are propagated down to the subconcept Z. The idea is that if some item from X_i does not satisfy the imposed condition f_i then all its subitems from the target subconcept Z also do not satisfy this condition. In the case of many constraints the result is intersection of individual de-projections:

$\bar{Z} = \bar{Z}_1 \cap \dots \cap \bar{Z}_n,$

$\bar{Z}_i = \bar{X}_i \to \{d_i\},\ i = 1, \dots, n$

Here d_i is a dimension from concept Z to its superconcept X_i. (In the case of many dimensions all of them are used independently.) Equivalently, subset $\bar{Z}$ can be defined as a multidimensional de-projection [16] of items $\langle x_1, \dots, x_n \rangle \in \bar{X}_1 \times \dots \times \bar{X}_n$ to Z along n dimensions $d_1, \dots, d_n$:

$\bar{Z} = \langle x_1, \dots, x_n \rangle \to \{d_1, \dots, d_n\}$

Thus an item from Z satisfies all the constraints $f_1, \dots, f_n$ imposed on concepts $X_1, \dots, X_n$ if it is projected in $\bar{X}_1, \dots, \bar{X}_n$ along the chosen dimension paths $d_1, \dots, d_n$:

$\bar{Z} = \{z \in Z \,|\, z \to d_i \in \bar{X}_i, i = 1, \dots, n\} \subseteq Z$

Subset $\bar{Z}$ contains the semantics satisfying all the imposed constrains. However, we do not need it in such a detailed form. Instead, we want to get related items from concept Y. In order to decrease the level of details the data from $\bar{Z}$ has to be projected to the target concept Y:

$\bar{Y} = \bar{Z} \to m$

Here m is a dimension from Z to Y. Selected items $\bar{Y} \subseteq Y$ are related to items $\bar{X}_1, \dots, \bar{X}_n$ chosen in $X_1, \dots, X_n$ and are returned as a result of the constraint propagation procedure.

An example of this procedure is shown in Fig. 4. Constraints are imposed on concept `Products` by selecting a number of product items. In order to get related orders from `Orders` these products have to be propagated down to `ProductParts` using de-projection. All the selected order parts (shown in dashed line in Fig. 4) reference only the specified products. On the second step these order parts are projected on concept `Orders`.

This procedure is analogous to inference in multi-dimensional space (Fig. 4 right). Constraints are imposed by selecting a subset of values along axis X. The available dependencies are represented as a subset of points from space Z (in circle). Intersection $\bar{Z} = \bar{X} \cap Z$ is then projected onto target axis Y and gives result $\bar{Y}$. Dependence Z can be represented using different techniques such as linear equations, differential equations, neural networks and so on. In the world of databases it is represented as a set of data items which select points from the space. In other words, a subconcept with its data items encodes a dependence between its superconcepts.

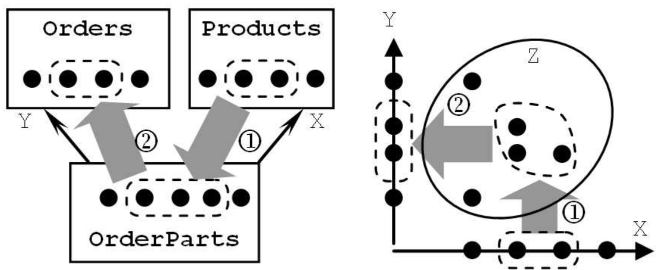

Figure 4. Constraint propagation procedure

Notice that related orders can be always obtained by manually specifying an appropriate access path. However, a property of the described procedure is that it does not require any access path and propagates constraints automatically. We need only to impose constraints and then indicate a target concept. The query for getting related orders can be written as follows:

```
SELECT * FROM Orders
WHERE Products.size > 10
```

This query will return records from table `Orders`. However its `WHERE` clause imposes additional restrictions on table `Products` by selecting only those with big size. Notice that there is no indication in this query how these two tables are connected and therefore the imposed restriction will be propagated automatically.

Fig. 5 provides an example of the described procedure. Assume that the question is what orders are related to beer and chips. This means that it is necessary to find all orders where either beer or chips are product items. The selected two products are shown in bold in Fig. 5. These two products are de-projected to concept `OrderParts` which will contain only three items. Then these three items are projected to concept `Orders`. They reference only two orders `<# 23>` and `<# 24>` which are the result of this query.

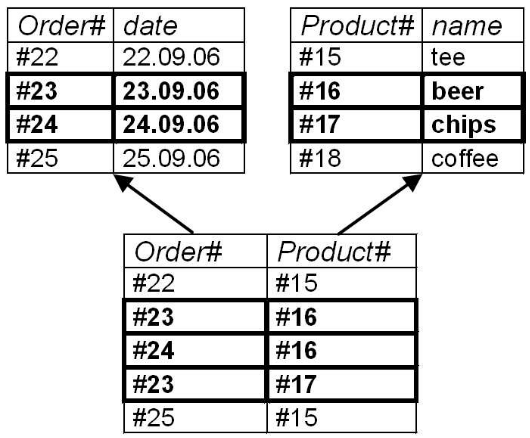

Figure 5. An example of constraints

In the case of many constraints in different parts of the concept graph they are propagated down to the most specific concept along their individual dimension paths. On the second step their intersection is propagated up to the target concept. A model in Fig. 6 consists of three already described concepts `Orders`, `Products` and `OrderParts`. However, concepts `Orders` and `Products` have their own superconcepts. In particular, each order is characterized by a customer (who made this order) and a date (when this order was made). Each customer belongs to some country from concept `Countries` and each product has a category from concept `Categories`. Let us now assume that we want to get all countries related to some product category (say, 'cars') and during some period of time (say, in 'June'). In other words, we want to learn in what countries cars were sold in June. Here again it is possible to write this query using concrete access paths for projection and de-projection. However, the method described in this paper does not need it and it is enough to simply impose our constraints and indicate what kind of result we want to get – all the rest will be done automatically.

The first constraint consists in selecting only cars. As a consequence all non-car items are effectively removed from the model. This means that all non-car products are removed and all order parts with non-car products are also removed. Thus we get some subset of all available order parts. The second constraint consists in selecting only items characterized by June as their date (concept `Months`). When this constraint is propagated down, all non-June items from its subconcepts are effectively removed. In particular, all non-June dates, all non-June orders and all non-June order parts do not satisfy this constraint. After that concept `OrderParts` will contain only items characterized by cars as its product category and by June as its date.

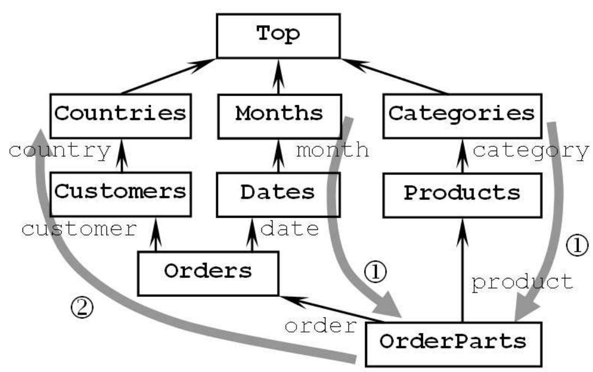

Figure 6. Many source concepts with constraints

The last step in this procedure consists in propagating the selected order parts up to the target concept `Countries`. The whole query can be written as follows:

```
SELECT * FROM Countries
WHERE Categories.name=='cars'
AND Months.name=='June'
```

This query selects records from one table while imposing restrictions on other tables which are not explicitly connected to the first one. In SQL such a query would need to join 8 pairs of tables.

The described procedure for automatic constraint propagation works only if an access path can be unambiguously restored. However there are models where many possible constraint propagation paths exist between source and target concepts. For example, products can be characterized by a country of origin (Fig. 7) and then concept `Countries` is a domain for two subconcepts `Customers` and `Products`. Let us assume that we want to get all related product categories for a selected country. Obviously, in this case there exist two options for constraint propagation from `Countries` to `Categories`. The first path goes through concept `OrderParts` and this propagation strategy will result in all product categories *ordered* by customers from the specified country (path ①). The second path goes through concept `Products` and it will return all categories for products *made* in the specified country (path ②).

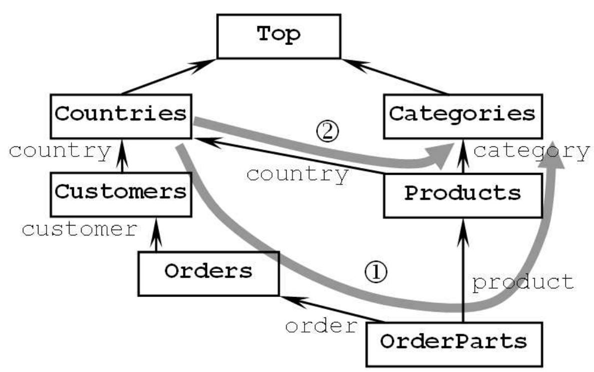

Figure 7. Multiple constraint propagation paths

In order to resolve this ambiguity and avoid the necessity to specify the complete path it is possible to use hints which help to choose one of many alternatives. One approach consists in using 'VIA' keyword followed by concept name. This keyword means that constraint propagation path has to include the specified concept(s), i.e., the reconstructed access path has to pass through this concept(s). For example, the following query will return all categories for products made in Germany:

```
SELECT * FROM Categories
WHERE Country.name=='Germany'
VIA Products
```

However, we can change this query and get all categories for orders from Germany:

```
SELECT * FROM Categories
WHERE Country.name=='Germany'
VIA OrderParts
```

By default in the case of no additional information the imposed constraints will be propagated along all possible paths. In the above example this would produce a set of categories for products ordered *and* made in Germany.

4 Implicit Constraints

In the previous section we considered an automatic constraint propagation procedure consisting of two steps: propagating down via deprojection and propagating up via projection. In this section we consider a more complex case where this procedure does not work because the source constraints do not directly influence the target concept. Let us consider an example shown in Fig. 8. Here coaches (concept `Coaches`) train teams (concept `Teams`) while one team consists of a number of players (concept `Players`). Concepts `Trains` and `Plays` store pairs of coach-team and player-team, respectively. (It is assumed that a player may play for many teams and a coach may train many teams.) Now let us ask the following question: find players related to a selected coach. For this model this question means that we want to get all players who have ever played for a team trained by this coach.

If we select the coach and propagate this constraint down according to the 2-step procedure then we get a subset of items from concept `Trains`. However, here this procedure stops because there is no path leading to the target concept `Players`. Indeed, we can project `Trains` to its superconcept `Teams` but not to `Players` because players are not directly connected with coaches.

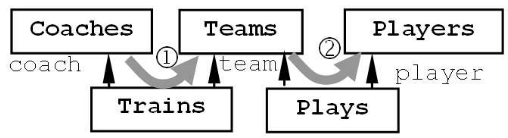

Figure 8. Multi-step inference

One simple solution to this problem consists in performing multiple de-projection/projection zigzags in concept graph. In this example, an obvious strategy is (1) to find a set of teams trained by the coach and then (2) all players from these teams. Here again we can use keyword `'VIA'` in order to specify more precisely a point in the concept graph used for constraint propagation:

```
SELECT * FROM Players
WHERE Coaches.name=='Smith'
VIA Teams
```

Another solution consists in applying the same 2-step procedure as described in the previous section but adding additional constraints which are implicitly assumed in the initial query. For example, when we want to get all players trained by some coach then it is implicitly assumed that the coach trains a team of this player. In other words, the team trained by the coach must be the same team where the player plays:

```
Trains.team == Plays.team
```

If this assumption can be explicitly formulated in the query then we can correctly propagate the initial constraints directly from coaches to players.

The concept-oriented model always has bottom concept which is a subconcept for any other concept. If it does not exist then it is added formally. In particular, the model in Fig. 8 does not have bottom concept so we formally add it and get the model shown in Fig. 9. The semantics of bottom concept is the canonical semantics of the whole model. It is equal to the Cartesian product of all its direct superconcepts: $B = S_1 \times \ldots \times S_n$. In our example it contains all combinations of items from concepts `Trains` and `Plays`. Effectively this means that there are no dependencies between these concepts and hence we will not be able to derive any consequence in one concept given constraints in another concept.

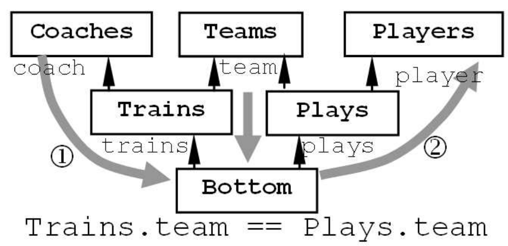

Figure 9. Inference under additional constraints

Bottom concept has a number of dimensions with domains in all its superconcepts. In our example it has 2 dimensions with domains in direct superconcepts `Trains` and `Plays` and 4 dimensions of rank 2 with domains in concepts `Coaches, Teams` and `Players` (so it is a 4-dimensional model). Notice however that two dimensions `Bottom.trains.team` and `Bottom.plays.team` have the same domain in concept `Teams`. We know that a coach is related to a player only if he trains the same team where this player plays. Thus we need to consider only data items from `Bottom` which satisfy this condition

`Bottom` → `trains` → `team` == `Bottom` → `plays` → `team`

It is precisely the additional constraint that has to be taken into account when carrying out inference procedure. (The first constraint is as usual a coach for which we want to get related players.) Both types of constraints (explicit and implicit) are propagated down to bottom concept and then the result is propagated up to the target concept. The whole query can be written as follows:

```
SELECT * FROM Players
WHERE Coaches.name == 'Smith'
AND Trains → team == Plays → team
```

The first line selects records from the target table. The second line imposes explicit restrictions by choosing one coach. And the third line specifies an additional implicit condition according to our understanding of the word *related* (records). In other situations this additional constraint could express some other semantics of the term related items. For example, a player might be treated as related to a coach if he has played for the team for more than some time or more than some number of games. These constraints encode dependencies in the model which are implicitly assumed when making a query.

5 Conclusions

In the paper we presented an approach to getting related data using automatic constraint propagation. This mechanism is based on specific properties of the concept-oriented data model. In particular, it assumes that elements of the model are ordered and this order is used to implement operations of projection and de-projection. This method is easy to use because in many situations it is enough to impose constraints and indicate the type of result we want to obtain. However, it makes it possible to express rather complex queries using additional hints in the case of ambiguity or additional constraints in the case of the absence of direct dependencies.

References

[1] R.Agrawal, A.Gupta, and S.Sarawagi. *Modeling multidimensional databases.* In 13th International Conference on Data Engineering (ICDE'97), 1997, pp. 232–243.

[2] A.Berson, and S.J.Smith. *Data warehousing, data mining, and OLAP.* New York, McGraw-Hill, 1997.

[3] P.Chen. *The Entity-Relationship Model. Toward a Unified View of Data.* In: ACM Transactions on Database Systems 1/1, 1976.

[4] E.F.Codd. *A relational model of data for large shared data banks.* Communications of the ACM, 13(6), 1970, pp. 377–387.

[5] R.Fagin, A.O.Mendelzon, J.D.Ullman . *A Simplified Universal Relation Assumption and Its Properties.* ACM Trans. Database Syst., 7(3), 1982, pp. 343–360.

[6] B.Ganter, and R.Wille. *Formal Concept Analysis: Mathematical Foundations.* Springer, 1999.

[7] P.M.D.Gray, P.J.H.King, and L.Kerschberg. (eds.) *Functional Approach to Intelligent Information Systems.* J. of Intelligent Information Systems, 12, 1999, pp. 107–111.

[8] P.M.D.Gray, L.Kerschberg, P.King, and A.Poulovassilis. (eds.) *The Functional Approach to Data Management: Modeling, Analyzing, and Integrating Heterogeneous Data.* Heidelberg, Germany, Springer, 2004.

[9] M.Gyssens, and L.V.S.Lakshmanan. *A foundation for multidimensional databases,* In VLDB'97, 1997, pp. 106–115.

[10] W.Kent. *Consequences of assuming a universal relation,* ACM Trans. Database Syst., 6(4), 1981, pp. 539–556.

[11] C.Li, and X.S.Wang. *A data model for supporting on-line analytical processing,* Proc. Conference on Information and Knowledge Management, Baltimore, MD, 1996, pp. 81–88.

[12] D.Maier, J.D.Ullman, and M.Y.Vardi. *On the foundation of the universal relation model.* ACM Trans. on Database System (TODS), 9(2), 1984, pp. 283–308.

[13] A.Savinov. *Hierarchical Multidimensional Modelling in the Concept-Oriented Data Model,* 3rd Intl. Conference on Concept

Lattices and Their Applications (CLA'05), Olomouc, Czech Republic, 2005, pp. 123–134.

[14] A.Savinov. *Concept as a Generalization of Class and Principles of the Concept-Oriented Programming,* Computer Science Journal of Moldova, 13(3), 2005, pp. 292–335.

[15] A.Savinov. *Logical Navigation in the Concept- Oriented Data Model,* Journal of Conceptual Modeling, Issue 36, 2005, August, http://www.inconcept.com/jcm.

[16] A.Savinov. *Grouping and Aggregation in the Concept-Oriented Data Model.* In 21st Annual ACM Symposium on Applied Computing (SAC'06), Dijon, France, 2006, pp. 482–486.

[17] D.W.Shipman. *The Functional Data Model and the Data Language DAPLEX.* ACM Transactions on Database Systems, 6(1), 1981, pp. 140–173.

Dr. Alexandr Savinov,
Institute of Mathematics and Informatics,
Academy of Sciences of Moldova
str. Academiei 5,
MD-2028 Chisinau, Moldova
E–mail: *savinov@conceptoriented.com*
Home page: *http://conceptoriented.com/savinov*

Received August 24, 2006

Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

The Development Methodology of the UML Electronic Guide

N.A. Magariu, L.L. Nigreţcaia-Croitor, M.V. Croitor

Abstract

A technological model for realization of the electronic guide to UML language is considered. This model includes description of peculiarities of using the special graphic editor for constructing the UML diagrams, XML vocabularies (XMI, DocBook, SVG, XSLT) for representing the text and diagrams and JavaScript code for constructing the tests.

1 About the problem

The high capacities of modern computers and networks of computers on the one hand and the needs of the big number of users on the other hand push developers of information technologies to create new approaches and means for the Complex Programming Systems (CPS) development. One of such means is the UML (Unified Modeling Language). The UML represents a graphic language for specifying various aspects of Object Oriented (OO) Programming Systems (PS) design [1]. It is already the input language for some CASE (Computer Aided Software Engineering) systems [2]. The methodology of applying the UML and methods of its realization are the subjects of modern scientific research. Our research of the tendencies in CPS elaboration methodology using OO CASE systems demonstrated that on the one hand one needs a lot of time to master these systems, and on the other hand, these systems themselves should be more effective [3]. In Software Design Research Laboratory, Faculty of Mathematics and Computer Science, State University of Moldova, an improved architecture of a CASE system named "MD Case" is being elaborated. General functional requirements to "MD Case" system are: - support for the forward and reverse engineering; - support for round-trip engineering which facilitates the obtaining complete program's code in a concrete OO language; - exporting and importing the UML models, represented in XMI language; - support for wide possibilities of using design patterns; - providing the possibility of automated project evaluation; - providing the possibility to master this system very quickly [3]. One of the "MD Case" system components is the "Electronic Guide" subsystem, which must help the user to become familiar with UML language constructions and methodology of their applying. The problem of learning and mastering the UML language is very important, that's why it had been decided to develop the independent electronic guide to UML language in Romanian, Russian and English.

2 Guide's concept

Modern technologies for the development of the electronic guides are usually being reduced to writing the static web-applications, dynamic web-applications or the complex interactive system, allowing carrying out user's mastering and examination concerning all aspects of the discipline [4]. On analyzing these technologies it has been found that the method of electronic guide construction based on representing the informational and programming components in HTML, XML, Java, PHP, etc. is the most commonly used. Such guides could be realized as web pages, which are executed by browser. Static web pages usually are generated automatically from .doc files with the help of MS Word or another editor. For the most part they support only sequential viewing and studying information. Dynamic web pages are interactive. They allow user to enter some information and to receive information depending on entered data. Such applications usually contain a database, an interface for maintaining and a user interface. The following tools can be listed among technological means, necessary for building such type of application: a HTML or XML editor, a data manager like MySQL, a programming system like JDK, servers Apache, PHP, etc. Sometimes special tools for automatic construction of e-Learning systems are used. One of these is the "Combinator" – an instrumental system for guide development [5]. Often the necessary tools for electronic guides' development and maintenance require additional licensing costs. Such technologies often use texts and multimedia files (pictures, audio, video), which are represented in different formats. This fact implies heterogeneity of data representation forms: text is usually represented in HTML format, graphics – in JPEG or GIF formats, audio – in MIDI or MP3 formats, and video – in AVI or MPEG. The heterogeneity of the data representation forms limits the information search, because it takes place within text format only.

The proposed technological model solves this problem by using the XML language to represent both text and other types of data. Since XML format represents a structured text, which describes the content of the document, but not its representation, the searching doesn't depend on the type of information (text, graphics, audio information). In addition such method of information representation is very useful for native databases creation.

Having analyzed the contents of the UML guide, it has been found, that this guide should contain following types of information: texts, UML diagrams and tests. Texts can be in Romanian, Russian or English languages. UML diagrams should be imported from CASE systems with UML as input language. After reading and learning any part of the guide, the user can test his knowledge immediately. Tests can be complex enough and can contain UML diagrams.

Taking into account the above-enumerated requirements it had been decided to realize the guide as a dynamic web-application, which uses XML to represent all the information. This had required analyses of above mentioned types of information representation in XML, and an effective method of their combination in a web-application. As a result of the analysis of web-applications creation methods, the method of the electronic guide realization, based on using of various XML vocabularies, had been chosen [6]. These vocabularies are: XMI (XML-based Metadata Interchange), SVG (Scalable Vector Graphics), DocBook, and XSLT (XSL for Transformations) [7,8,9,10]. The JavaScript language had been chosen for tests implementation.

3 The proposed technological model

To represent the formatted text in the guide, it had been decided to use DocBook language, because DocBook documents can be processed in the same way as XML documents, and because of the possibility of their simple transformation into PDF or CHM documents [9, 10]. Representation of the text in DocBook format can be obtained using the AbiWord processor, which reads Word Document (*.doc) files and generates the DocBook (*.dbk) files [11]. Thus, the guide's texts can be prepared by means of the MS Word-like system, and later transformed into DocBook format.

To represent UML diagrams in the guide, it had been decided to use XMI, a language for describing and interchanging metadata and, particularly, UML models. For the UML diagrams' construction it had been decided to use the "MD Case" system's graphic editor, which allowed to build UML models and to receive their descriptions in XMI language. There is an important property of the "MD Case" system that distinguishes it from the most of the similar systems. This property consists in the possibility to store the coordinates and dimensions of UML diagram elements in the generated XMI document [6,12].

The modern web-browsers are not adapted to display UML diagrams represented in XMI language. However, two-dimensional graphics can be rather easily displayed by browsers if they are represented in the SVG language. The problem of transforming the XMI document into SVG-document was solved by using the XSLT technology. For the purpose of viewing SVG graphics the SVG Viewer must be integrated in the Web browser as a plug-in.

For tests realization the JavaScript language is being used. The scripts can be included in XSLT templates or in DocBook document.

The guide content is being developed as an XML document, where the texts are represented in DocBook format, the UML diagrams – in XMI format, and the tests are realized in JavaScript. The browser visualizes the XML document, using the corresponding XSLT transformation and SVG Viewer.

The general logical scheme of our technological model is presented in the Figure 1.

On analyzing this model one can observe that the XMI descriptions of UML diagrams (*.xmi) must be included in the DocBook document (*.dbk). The *.dbk file is processed in accordance to XSLT transformation templates (*.xslt) and using special XSLT processor, integrated in Web browser. These XSLT templates are used to transform DocBook document into HTML document with SVG images descriptions included.

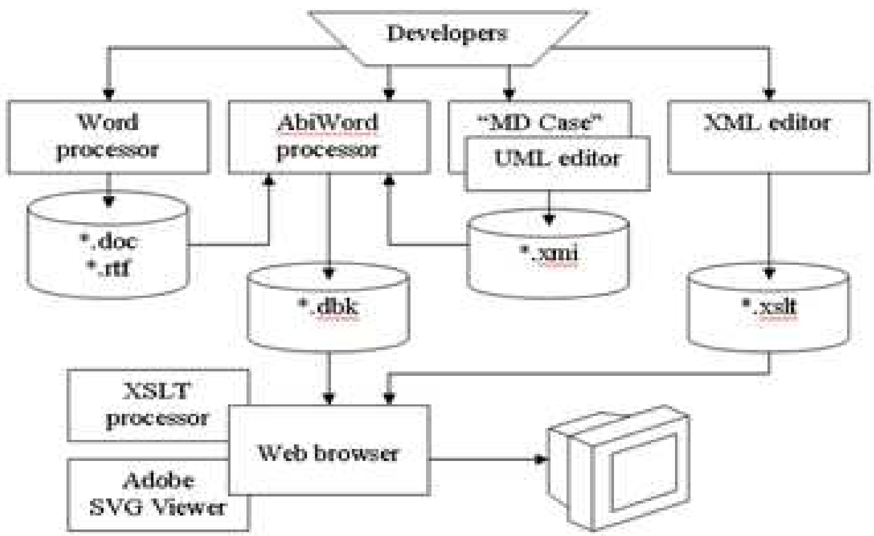

Figure 1. The general scheme of the technological model

4 Instruments application particularities

The common structure of the UML diagram description in XMI format looks in much the same way as the example, which contains XMI description of the UML class diagram:

```
<XMI version = "1.1" xmlns:UML = "http://omg.org/UML">
  <XMI.header>
    <XMI.documentation>
      <XMI.exporter xmi.extender = "MD Case"/>
    </XMI.documentation>
    <XMI.model xmi.name = ""/>
    <XMI.metamodel xmi.name = "UML"/>
  </XMI.header>
  <XMI.content>
    <UML:Namespace.ownedElement>
      <UML:Class name = "NewClass1" xmi.id = "NewClass1"
      isAbstract = "false"/>
    </UML:Namespace.ownedElement>
  </XMI.content>
  <XMI.extensions xmi.extender = "MD Case">
    <UML:Diagrams>
      <UML:Diagram kind = "Class Diagram">
        <UML:Diagram.elements>
          <UML:DiagramElement type = "clasa" name =
          "NewClass1" originx = "74" originy = "73"
          width = "110" height = "110"/>
       </UML:Diagram.elements>
      </UML:Diagram>
    </UML:Diagrams>
  </XMI.extensions>
</XMI>
```

Because the standard XMI description of the UML model doesn't contain the information about representation of the model as a set of diagrams, it had been decided to extend this description by adding special elements: **UML:Diagrams**, **UML:Diagram**, **UML:Diagram.elements**, **UML:DiagramElement**. New elements had been added inside the **XMI.extensions** element that was specially destined for describing the extensions of the XMI format. The attributes of above-mentioned new elements contain all the information needed for visual representation of the UML diagrams. Particularly, they contain coordinates and dimensions of the diagram elements. For example, in above-stated XMI-document the element **UML:DiagramElement** contains attributes: **originx="74", originy="73", width="110", height="110"**. These attributes are being used while transforming the XMI specifications into SVG specifications. The values of these attributes specify the rectangle in the following SVG-document, which represents specification for the graphical notation of the UML class described in the above XMI document:

```
...
<svg:svg width="400" height="400">
   <svg:g>
      <svg:rect x="74" y="73" height="110" width="110"
      style="stroke:rgb(0,0,0);fill:rgb(255,255,255);"/>
      <svg:line style="stroke:rgb(0,0,0); stroke-width:1;
      " x1="74" y1="93" x2="184" y2="93"/>
      <svg:line style="stroke:rgb(0,0,0); stroke-width:1;
      " x1="74" y1="113" x2="184" y2="113"/>
      <svg:text x="94" y="93" style="font-size:12;
      font-weight:12; font-family: Courier;
      font-style:italic">NewClass1</svg:text>
   </svg:g>
</svg:svg>
...
```

The **svg** is the root element in this description. The size of the image is specified with the help of its attributes. The **g** element serves as a container of graphic elements. The **rect** element describes a rectangular. Its attributes set the coordinates of the top left corner of a rectangular, the size of a rectangular, its color and the color of the borders. The **line** element describes a segment. Its attributes specify coordinates of the segment endings, color and thickness of the line. It is necessary to accentuate, that the SVG-document was obtained as the result of the XSLT transformation, applied to the XMI document.

XSLT transformations represent instructions of kind "if node X is found then the Y action is to be executed". The root element of the XSLT transformation may contain such child elements as **xsl:template** which sets a rule of transformation, **xsl:apply-templates** which is used to apply the rules, **xsl:element** – to create element nodes, etc. To demonstrate the common structure of XSLT template let us consider a template, transforming some parts of the mentioned above XMI document.

The XSLT template that transforms the above-stated XMI class description to SVG format (using **svg** namespace) is represented in APPENDIX 1.

Thus, for all XMI-descriptions of the elements of UML-diagrams we can define the corresponding templates. On applying these templates, the XSLT processor, which is the part of web browser, generates documents with both HTML and SVG elements. Now it is required to display simultaneously both the text and figures. Usually, browsers ignore all tags that are not the standard HTML tags. Adobe SVG Viewer plug-in – a special COM-component used to process SVG elements, could be integrated in web-browser [13]. Adobe SVG Viewer plug-in processes elements of the SVG namespace, therefore the SVG namespace should be declared in the root template. The plug-in can be activated in the following way:

```
...
<html xmlns:svg = "http://www.w3.org/2000/svg">
   <object id = "AdobeSVG" CLASSID = "clsid:78156a80-c6a1-
   4bbf-8e6a-3cd390eeb4e2"/>
   <xsl:processing-instruction name = "import"> namespace =
   "svg" implementation = "#AdobeSVG" </xsl:processing-
   instruction>
...
</html>
...
```

This fragment of code means that all tags, belonging to **svg** namespace will be processed by Adobe SVG Viewer plug-in.

For processing and transforming XML documents the special programs – XML processors are used. Modern browsers come with XML processors, implemented as COM components. The XML processor allows using JavaScript code simultaneously with XSLT. That can essentially increase the capabilities of the guide. For example, it can allow the inclusion in guide some knowledge evaluation system. Besides this, taking into consideration that all the data is stored in a single file, per page representation of the information is required. JavaScript allows achieving this goal.

The proposed model of guide should have a simple knowledge evaluation system, meant for self-control. In such a system the security for user unauthorized data access is not a primary goal because he is interested in an adequate evaluation of himself. Therefore the information about tests could be stored in an XML-oriented database. One of the methods for tests representation in XML documents is the method, corresponding to QTI (Question and Test Interoperability) specification from IMS (Instruction Management System) standard [14]. At the same time the testing process itself can be implemented in JavaScript language.

The guide which is elaborated in conformance to this technological model represents an XML-oriented database. The access and representation of the data from this database, as well as interaction with user is realized by browser according to XSLT templates and JavaScript programs. The content of database could be easily modified without making any modifications in XSLT templates.

Replacing „UML editor" by „SVG editor" makes this technological model suitable for construction of any other electronic technical literature which contains a lot of graphics.

Implementation of this model doesn't require large financial costs for buying expensive tools and allows the creation of the platform independent applications, which can run both online and offline.

References

[1] OMG's UML Resource Page http://www.uml.org/

[2] OMG's List of UML Tools http://www.uml.org/#Links-Tools

[3] Magariu N. *Complex programs development by means of CASE systems.* Second Conference of the Mathematical Society of the Republic of Moldova: Communications, Chisinau, August, 17-19, 2004. Ch: Tipogr. Acad. Şt. RM – 2004, pp. 207–210.

[4] Carmocanu Gh., Magariu N. *E-Learning applications development.* The scientifical annals of the State University of Moldova. "Physical and Mathematical Sciences" series. Chisinau, CEP USM, 2005, pp.161–170. (Romanian)

[5] The technology of dictionary, guide and manual construction. http://www.elcode.ru/other_tech.html

[6] Magariu N. A. *The technological model for the realization of the electronic guide to UML language.* The proceedings of the 1-st scientific-practical conference "The Modern Problems of the Applied Computer Science", May 23-25, 2005, St.-Petersburg Engineering and Economics University. INJECON, 2005, pp. 73–76. (Russian)

[7] OMG's XMI Specification http://www.omg.org/technology/documents/modeling_spec_catalog.htm#XMI

[8] W3C's SVG Resource Page http://www.w3.org/Graphics/SVG/

[9] DocBook Specifications http://www.docbook.org/specs/index.html

[10] W3C's XSLT Resource Page http://www.w3.org/TR/xslt20/

[11] AbiWord Web Page http://www.abisource.com/

[12] Buga T., Magariu N. *The development of the ,,MD Case" system's graphic editor.* The scientifical annals of the USM. "Students works" series. Natural sciences. Chişinău, CEP USM, 2005, pp. 61–65. (Romanian)

[13] Adobe SVG Viewer Web Page http://www.adobe.com/svg/viewer/install/

[14] IMS Question & Test Interoperability Specification http://www.imsglobal.org/question/index.html

APPENDIX 1. The XSLT template for XMI class transformation into SVG format

```
... <xsl:template match="UML:Diagram">
   <xsl:if test="@kind='Class Diagram' and ./UML:Diagram.
   elements/UML:DiagramElement/@type='clasa'">
      <xsl:variable name="x" select="./UML:Diagram.
      elements/UML:DiagramElement/@originx"/>
      <xsl:variable name="y" select="./UML:Diagram.
      elements/UML:DiagramElement/@originy"/>
      <xsl:variable name="h" select="./UML:Diagram.
      elements/UML:DiagramElement/@height"/>
      <xsl:variable name="w" select="./UML:Diagram.
      elements/UML:DiagramElement/@width"/>
      <xsl:variable name="name" select="./UML:Diagram.
      elements/UML:DiagramElement/@name"/>
      <svg:rect x="{$x}" y="{$y}" height="{$h}"
      width="{$w}"
      style="stroke:rgb(0,0,0);fill:rgb(255,255,255);"/>
      <svg:line style="stroke:rgb(0,0,0); stroke-width:1;">
         <xsl:attribute name="x1">
            <xsl:value-of select="$x"/>
         </xsl:attribute>
         <xsl:attribute name="y1">
            <xsl:value-of select="$y + 20"/>
         </xsl:attribute>
         <xsl:attribute name="x2">
            <xsl:value-of select="$x + $w"/>
         </xsl:attribute>
```

```
         <xsl:attribute name="y2">
            <xsl:value-of select="$y + 20"/>
         </xsl:attribute>
      </svg:line>
      <svg:line style="stroke:rgb(0,0,0); stroke-width:1;">
         <xsl:attribute name="x1">
            <xsl:value-of select="$x"/>
         </xsl:attribute>
         <xsl:attribute name="y1">
            <xsl:value-of select="$y + 40"/>
         </xsl:attribute>
         <xsl:attribute name="x2">
            <xsl:value-of select="$x + $w"/>
         </xsl:attribute>
         <xsl:attribute name="y2">
            <xsl:value-of select="$y + 40"/>
         </xsl:attribute>
      </svg:line>
      <svg:text x="{$x + 20}" y="{$y + 16}" style="font-
      size:12; font-weight:12; font-family: Courier;
      font-style:italic">
         <xsl:value-of select="$name"/>
      </svg:text>
   </xsl:if>
   <xsl:apply-templates/>
</xsl:template>
...
```

N.A.Magariu, L.L.Nigreţcaia-Croitor, M.V.Croitor
Received August 23, 2006

State University of Moldova
60, A. Mateevici str.
Chişinău, MD−2009, Moldova
Phone: (373+2) 57-77-33
E–mail: *magariu@usm.md*, *liudmile@softhome.net*, *krey@usm.md*

Computer Science Journal of Moldova, vol.14, no.2(41), 2006

The effects of Ethernet LANs' fragmentation

Ion Bolun, Victor Andronatiev

Abstract

Three configurations of Ethernet networks that operate in duplex mode at relatively low load of transmission media are investigated. Configurations differ by number of collision domains and switches used in the network. To investigate the dependences among basic parameters, analytic models of the functioning of such networks are elaborated. Calculations made for a relatively large set of values of initial data confirm the necessity of case dependent special investigations on the physical configuration of concrete Ethernet networks. It has been determined, for example, that the fragmentation doesn't necessarily permit to increase the number of stations in the network; it may be even vice versa.

1 Introduction

Currently the Ethernet, Fast Ethernet and Gigabit Ethernet local networks are largely used. Moreover, the efficient capitalization of their resources is important. In many cases one of the reserves consists in the fragmentation of networks - their construction from many interconnected segments [3, 4]. Such a fragmentation usually permits:

- grouping in a fragment of stations with higher traffic interchange;
- to increase the total data traffic in the network;
- to decrease the mean time of messages delay in the network;
- to increase the reliability of services offered etc.

At the same time, fragmentation implies additional costs with equipment to interconnect the network's segments. Therefore, for a rational networks' fragmentation, it is necessary to make a quantitative appreciation of the respective effects. Some aspects of such appreciation are discussed in this paper.

2 Preliminary considerations

Investigations are based on networks or network fragments with only one collision domain. Some results of the research of networks with only one collision domain and homogenous stations are published in [1-3]. In these papers the boundaries of networks performance parameters, such as the mean delay time of packets and the mean utilization of transmission media are estimated.

At the same time, data packets are considered of fixed length, which, as mentioned in [3], sometimes causes significant deviations from the real situation, improving the real network performance. Moreover, in paper [2] the nature of flow of messages generated by network stations is not taken into account; in calculations only the mean value of respective parameters is used, such as: the mean time that a station doesn't have packets to transmit and the mean waiting time until the beginning of packet transmission. In [3] the flows of packets generated by stations are considered of Poisson type, but at the same time it is supposed that each station transmits each packet before the next one arrives; this corresponds to the reality only in the case of messages which do not exceed the maximum length of one packet; in addition it is considered that the network contains an infinite number of stations.

Thus, investigations in [1-3] are oriented to marginal appreciations of the performance parameters of local networks with only one collision domain. In the present paper an attempt is made to use models that would allow the evaluation of performance parameters for the cases of relatively low transmission media load ρ, aiming to appreciate the effects of fragmentation of local networks in many collision domains. It is well known that the Ethernet technology is working efficiently at a load that doesn't exceed 30-40 % [3]. At such a load, there may be signals' collisions in the transmission media, but the probability that there will be more than one collision per packet is very small.

There are investigated Ethernet local networks containing N homogeneous stations, each of which generates requests to other stations in the network with the transmission of a Poisson flow of messages of rate β; the repartition of messages' size is exponential and its average is L. To be mentioned that the exponential repartition of messages' size and data packets' size are accepted in many investigations [3-5, etc.]. Stations of the same collision domain are distributed uniformly between marginal stations. The data transmission speed in the media is V. It is required to investigate the network's basic operation features when the following constraints are observed:

- the load ρ of the network transmission media doesn't exceed a given value ρ_0
$$\rho \leq \rho_0; \tag{1}$$
- the mean delay time θ of a message in the network doesn't exceed a given value θ_0
$$\theta \leq \theta_0. \tag{2}$$

Messages are transmitted through the network by packets, the size of which is considered of exponential repartition with the average value ν and the heading of length s. Thus, the number of packets per message is equal to $L/(\nu-s)$. The processing time for disassembling messages in packets and for the inverse operation of assembling packets in messages at stations can be considered proportional with the number of packets per message, that is equal to $aL/(\nu - s)$, where a is a proportionality coefficient.

Let T be the mean delay time of a packet in the network (the mean waiting time + the mean transmission time), then

$$\theta = \frac{L}{\nu - s}\left(T + a\right). \tag{3}$$

Taking into consideration relation (3), the constraint (2) can be represented, at packets' level, as

$$T \leq T_0, \tag{4}$$

where T_0 is the upper limit of the mean time of a packet delay in the network; the following relation takes place between constants θ_0 and T_0:

$$\theta_0 = \frac{L}{\nu - s}\left(T_0 + a\right). \tag{5}$$

Switches (bridges) are used to interconnect different network fragments. It is considered that the network operates in the duplex mode with the memorization of frames by switches and each collision domain of it forms a separate fragment. Thus, the network's physical configuration is determined by the content of its fragments and the topology of their interconnection through switches.

From multiple possible variants, the investigations in this paper will confine to networks of three configurations:

1) networks with only one collision domain;

2) networks with a switch to which m homogeneous fragments-collision domains, with n stations each, are connected

$$n = \frac{N}{m}; \tag{6}$$

3) networks with two switches, to each of which m homogeneous fragments-collision domains, with n stations each, are connected

$$n = \frac{N}{2m}. \tag{7}$$

Obviously, the rate μ of packets transmission through network media is calculated as

$$\mu = \frac{V}{\nu}. \tag{8}$$

There are of interest such aspects of the problem as:

- determination of parameters θ and ρ for a given network;
- finding the maximum number N_{max} of stations for a given network;

- comparison of the performance of networks of different configurations.

These aspects will be investigated in sections 3–6. The analytic models of networks of each of these three configurations are made up in sections 3–5, and some results of calculations for a set of real values of the respective parameters are described in section 6.

3 Networks of configuration 1

Let there be a network with a single collision domain in the frame of which all stations are interconnected. Two analytic models of such networks will be investigated: (1) model 1a, which doesn't take into account the time of signal propagation σ through the transmission media, nor the mean delay time ω caused by a possible collision and (2) model 1b, which takes into account parameters σ and ω. The quantitative comparison of these two models is made in subsection 6.2.

3.1 Model 1a – an elementary analytic model.

When constraint (4) is observed and ρ_0 has reasonable values, it is acceptable, in practical calculations, to not take into account the influence of collisions on packets' delay time in the network. The time of signal propagation through the transmission media will be not taken into account, as well. The opportunity of such suppositions is confirmed for a large set of real values of initial data in subsection 6.2.

So, the network can be represented as a queue system with one server – the transmission media. The mean time of a packet delay T_1 in the network, taking into account the Poisson nature of packets flow (see [1]) and the exponential repartition of packets' size, are determined as [5]

$$T_1 = \frac{1}{\mu - \lambda_1}, \tag{9}$$

where the rate λ_1 of packets flow in the transmission media is calculated like

$$\lambda_1 = N\gamma. \tag{10}$$

Taking into account relations (8) and (10), formula (9) takes the following form

$$T_1 = \frac{\nu}{V - N\nu\gamma}. \tag{11}$$

At the same time, load ρ_1 of the transmission media, taking into account relations (8) and (10), is

$$\rho_1 = \frac{\lambda_1}{\mu} = \frac{N\nu\gamma}{V}. \tag{12}$$

The analytic model of the network, from the viewpoint of aspects suggested in section 2, is determined by relations (11) and (12) for parameters T_1 and ρ_1.

Determination of the maximum number N_{1max} *of stations in the network.*

The observance of constraints (1) and (3) or (4) imposes the limitation of number N of stations in the network. Proceeding from the condition not to exceed the mean delay time T_0 of packets in the network (see constraint (4)) and taking into account formula (11), one has

$$\frac{\nu}{V - N\nu\gamma} \leq T_0 \tag{13}$$

or

$$N_{1\,\max}(T_0) = \frac{VT_0 - \nu}{\nu\gamma T_0}. \tag{14}$$

As well, proceeding from the condition that the load mustn't exceed ρ_0 (see constraint (1)), taking into account the formula (12), one has

$$\frac{N\nu\gamma}{V} \leq \rho_0 \tag{15}$$

or

$$N_{1\,\max}(\rho_0) = \frac{\rho_0 V}{\nu\gamma}. \tag{16}$$

Combining constraints (14) and (16) with regard to the maximum number of stations in the network, one has

$$N_{1\,\max} = \min\left\{\frac{\rho_0 V}{\nu\gamma}; \qquad \frac{VT_0 - \nu}{\nu\gamma T_0}\right\}. \tag{17}$$

3.2 Model 1b – a more exact analytic model.

In order to consider the mean time σ of signal propagation through the media between the source station and the destination station and also the mean delay time ω caused by a possible collision, one adds values σ and ω to expression (9) for T_1

$$T_1 = \frac{1}{\mu - \lambda_1} + \sigma + \omega. \tag{18}$$

Let's determine the values of σ and ω. For a station placed at a distance l (in time) from one of the two marginal stations (Fig. 1) and a uniform distribution of stations between the two marginal stations, the mean distance r (in time) to one of the other $n-1$ stations of the collision domain is

$$r = \frac{l}{\tau}\int_0^l t\frac{1}{\tau}dt + \frac{\tau - l}{\tau}\int_0^{\tau - l} t\frac{1}{\tau}dt = \frac{l}{2} + \frac{(\tau - l)^3}{2\tau^2}. \tag{19}$$

Here τ is the network diameter in time units – the signal propagation time through the media between the marginal stations. Then, in average on the set of n stations, one has

$$\sigma = \int_0^\tau r\frac{1}{\tau}dl = \int_0^\tau \left[\frac{l}{2} + \frac{(\tau - l)^3}{2\tau^2}\right]\frac{1}{\tau}dl = \frac{3}{8}\tau. \tag{20}$$

Obviously,

$$\omega = p(\sigma + \eta), \tag{21}$$

where p is the probability of one collision during the transmission of the current packet, the mean transmission time before the collision is numerically equal to σ, and η is the mean time before the beginning of packet retransmission after the collision. The following relation takes place:

$$p = \rho \frac{\sigma}{\frac{\nu}{V}} = \frac{3}{8}\rho\tau\mu = \frac{3}{8}\lambda_1\tau. \tag{22}$$

Here $\rho = \lambda_1/\mu$ is the media load (without taking into consideration the collisions) and, at the same time, the probability that when a new packet arrives the transmission media is busy. Taking into consideration that the ratio ν/V is the mean packet transmission time, the ratio $\sigma/(\nu/V)$ is the probability that the source station of the new packet doesn't identify the busy state of the media and so it will start the transmission, which will, in the ultimate respect, lead to a collision.

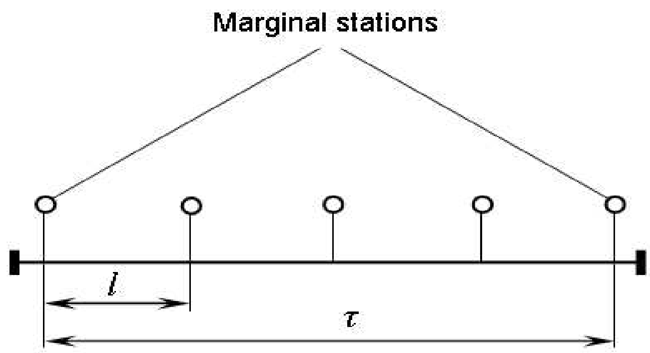

Figure 1. A collision domain.

After the collision, each of the two stations will begin the transmission after a random period of time which is uniformly distributed in the interval $[0; b]$. So, the probability that at least one of the two stations will begin the transmission after the period of time t is

$$P(x_1 \leq t \text{ or } x_2 \leq t) = 1 - \left(\frac{b-t}{b}\right)^2 = \frac{t(2b-t)}{b^2}, \tag{23}$$

from which the repartition $f(t)$ is

$$f(t) = \frac{dP(x_1 \leq t \text{ or } x_2 \leq t)}{dt} == \frac{2}{b^2}(b-t). \tag{24}$$

So,

$$\eta = \int_0^b tf(t)dt = \int_0^b \frac{2t(b-t)}{b^2}dt = \frac{b}{3}. \qquad (25)$$

By substituting (22) and (25) in (21), one can obtain

$$\omega = \frac{3}{8}\rho\tau\mu\left(\frac{3}{8}\tau + \frac{b}{3}\right) = \frac{3}{8}\lambda_1\tau\left(\frac{3}{8}\tau + \frac{b}{3}\right) = h\lambda_1, \qquad (26)$$

where

$$h = \sigma(\sigma+\eta) = \frac{3}{8}\tau\left(\frac{3}{8}\tau + \frac{b}{3}\right). \qquad (27)$$

Taking into account relations (20) and (26), formula (18) takes the form

$$T_1 = \frac{\nu}{V - N\nu\gamma} + \frac{3}{8}\tau\left[1 + N\gamma\left(\frac{3}{8}\tau + \frac{b}{3}\right)\right]. \qquad (28)$$

The load ρ_1 of the transmission media, taking into account relation (12), is

$$\rho_1 = \frac{\lambda_1}{\mu} + \lambda_1\omega = \frac{\lambda_1}{\mu} + \lambda_1 h\lambda_1 = h\lambda_1^2 + \frac{\lambda_1}{\mu} = N\gamma\left(N\gamma h + \frac{\nu}{V}\right). \quad (29)$$

The network analytic model 1b, from the point of view of the aspects proposed in section 2, is determined by relations (28) and (29) for parameters T_1 and ρ_1.

Determining the maximum number N_{1max} of stations in the network.

From the condition that the mean delay time of packets in the network should not exceed T_0 (see constraint (4)), taking into account formula (28), one has

$$\frac{\nu}{V - N\nu\gamma} + \frac{3}{8}\tau\left[1 + N\gamma\left(\frac{3}{8}\tau + \frac{b}{3}\right)\right] \le T_0. \qquad (30)$$

Then from (30) one has

$$N_{1\,\max}(T_0) = \frac{hV + T_0'\nu + \sqrt{(hV + T_0'\nu)^2 + 4h\nu(\nu - T_0'V)}}{2h\nu\gamma}, \qquad (31)$$

where

$$T_0' = T_0 - \frac{3}{8}\tau. \tag{32}$$

To be mentioned that in formula (31) the solution obtained using the sign "-" (minus) before the square root doesn't have sense (the value of $N_{1\max}(T_0)$ would be negative).

In a similar mode, from the condition that the load of media should not exceed ρ_0 (see constraint (1)), taking into account formula (29), one has

$$N\gamma\left(N\gamma h + \frac{\nu}{V}\right) \le \rho_0 \tag{33}$$

or

$$N_{1\max}(\rho_0) = \frac{-\nu + \sqrt{\nu^2 + 4h\rho_0 V^2}}{2\gamma h V}. \tag{34}$$

As in the case of formula (31), in formula (34) the solution obtained using the sign "-" (minus) before the square root doesn't have sense, because the value of $N_{1\max}(\rho_0)$ would be negative.

Combining expressions (31) and (34) with regard to the maximum number of stations in the network, one has

$$N_{1\max} = \min\left\{\frac{-\nu + \sqrt{\nu^2 + 4h\rho_0 V^2}}{2\gamma h V};\right. \left.\frac{hV + T_0'\nu + \sqrt{(hV + T_0'\nu)^2 + 4h\nu(\nu - T_0'V)}}{2h\nu\gamma}\right\}. \tag{35}$$

4 Networks of configuration 2

4.1 An analytic model.

Let's consider a network with one switch to which $m$homogenous fragments-collision domains of n stations each are connected. The mean delay time T_2 of a packet in such a network is determined as

$$T_2 = \frac{\lambda_{2int}T_2' + \lambda_{2ext}2T_2'}{\lambda_{2int} + \lambda_{2ext}} = T_2'\frac{\lambda_{2int} + 2\lambda_{2ext}}{n\gamma}, \tag{36}$$

where:

$T_2^{'}$ - the mean delay time of a packet in a fragment-collision domain;

λ_{2int} - the rate of packets flow generated by the n stations of the same collision domain and destined to stations belonging to it;

λ_{2ext} - the rate of packets flow generated by the n stations of the same collision domain and destined to another $N-n$ stations (stations that belong to the other collision domains) of the network. Obviously,

$$\lambda_{2int} + \lambda_{2ext} = n\gamma. \tag{37}$$

In relation (37), the product $n\gamma$ determines the rate of requests flow generated by the n stations of the investigated collision domain. Taking into account that the rate of packets flow generated by one of the stations and destined to a different concrete station is $\gamma/(N-1)$ the following relations take place:

$$\lambda_{2int} = n\frac{\gamma}{N-1}(n-1) = n\gamma\frac{n-1}{N-1}; \tag{38}$$

$$\lambda_{2ext} = n\frac{\gamma}{N-1}(N-n) = n\gamma\frac{N-n}{N-1}. \tag{39}$$

The quantitative comparison of models 1a and 1b for networks of configuration 1 described in subsections 3.1 and 3.2 show (see section 6.2) that, at reasonable loads of the transmission media, these do not differ significantly. Therefore, in models for networks of configurations 2 and 3, the effects caused by signals collisions shall not be taken into consideration, nor will the duration of signal propagation through the transmission media. So, like in formula (9), the following relation takes place

$$T_2^{'} = \frac{1}{\mu - \lambda_2}, \tag{40}$$

where the rate λ_2 of packets flow in the transmission media of one collision domain is calculated according to formula

$$\lambda_2 = n\gamma + (N-n)\frac{\gamma}{N-1}n = n\gamma\frac{2N-n-1}{N-1}. \tag{41}$$

In expression (41), product $n\gamma$ determines, as was mentioned above, the rate of the flow of requests generated by the n stations of this collision domain, and expression $n\gamma(N-n)/(N-1)$ – the rate of the flow of requests generated by the other $N-n$ stations of the network and addressed to stations of the investigated collision domain (external requests). From (41) it is easy to notice that

$$\lambda_2 < 2n\gamma, \tag{42}$$

and at $n << N$ or, more exactly, at $m \to \infty$ relation (41) turns to

$$\lim_{m\to\infty} \lambda_2 = 2n\gamma, \tag{43}$$

meaning that λ_2 depends, practically, not on the total number N of stations in the network, but on the number $n = N/m$ of stations in a collisions domain.

From (38), (39) and (41) one can notice that the following relation takes place

$$\lambda_2 = \lambda_{2int} + 2\lambda_{2ext}, \tag{44}$$

therefore and, also, taking into consideration relation (40), formula (36) can be replaced with

$$T_2 = \frac{1}{\mu - \lambda_2} \cdot \frac{\lambda_2}{n\gamma} = \frac{2N-n-1}{\mu\,(N-1) - n\gamma\,(2N-n-1)}. \tag{45}$$

To be mentioned that at $m = 1$ the equality $n = N$ takes place and formula (45) for T_2 is reduced to formula (9) for T_1; thus

$$T_2|_{m=1} = T_1 \tag{46}$$

and, respectively, see (3),

$$\theta_2|_{m=1} = \theta_1. \tag{47}$$

As far as it concerns load ρ_2 of the transmission media, it is determined as

$$\rho_2 = \frac{\lambda_2}{\mu} = \frac{n\nu\gamma\,(2N-n-1)}{V\,(N-1)}. \tag{48}$$

In a manner similar to (46), at $m = 1$, the equality $n = N$ takes place, and formula (48) for ρ_2 is reduced to formula (12) for ρ_1, therefore

$$\rho_2|_{m=1} = \rho_1. \tag{49}$$

So, T_2 is calculated according to the expressions (45) and (48) (taking into account (3), and $\theta_2 = \theta$) and, respectively, ρ_2 which constitutes, from the viewpoint of the research aspects proposed in section 2, the analytic model of networks of configuration 2.

4.2 An asymptotic analytic model.

The analytic model determined by relations (45) and (48) can be considerably simplified at $n << N$ or, more exactly, at m $\to \infty$. Indeed, from (45) it is easy to notice that

$$T_2 < \frac{2}{\mu - 2n\gamma}, \tag{50}$$

and at $m \to \infty$ relation (45) turns to an asymptotic, simpler one

$$\lim_{m\to\infty} T_2 = \frac{2}{\mu - 2n\gamma}, \tag{51}$$

meaning that T_2 depends, practically, not on the total number N of stations in the network, but only on the number $n = N/m$ of stations in a collision domain.

In a similar way, from (48) it is easy to notice that

$$\rho_2 < \frac{2n\gamma}{\mu} \tag{52}$$

and at $m \to \infty$ expression (48) is reduced to a simpler one

$$\lim_{m\to\infty} \rho_2 = \frac{2n\gamma}{\mu} = \frac{2n\nu\gamma}{V}, \tag{53}$$

meaning that ρ_2 depends, practically, not on the total number N of stations in the network, but only on the number $n = N/m$ of stations in the collision domain.

Thus, since usually $n << N$, both T_2 (see formula (51)) and ρ_2 (see formula (53)) depend, practically, not on the total number N of stations in the network, but only on the number $n = N/m$ of stations in the collision domain. Therefore it is reasonably to calculate the maximum number n_{2max} of stations in the collision domain, imposed by restrictions (1) and (4).

4.3 Calculation of n_{2max}.

Taking into account relation (45), the condition (4) of not exceeding the mean delay time T_0 of packets in the network takes, for this case, the following form

$$\frac{2N - n - 1}{\mu\,(N-1) - n\gamma\,(2N - n - 1)} \le T_0, \tag{54}$$

whence, in the result of some transformations,

$$\gamma T_0 n^2 - [\gamma T_0\,(2N-1) - 1]n - [\mu T_0\,(N-1) + 2N - 1] \ge 0. \tag{55}$$

From this square inequation it is easy to obtain the solution for the maximum number $n_{2max}(T_0)$ of stations in a collision domain, calculated taking into account separately just the restriction (4) regarding the limit mean time $T_2 = T_0$ of a packet in a network of configuration 2

$$\begin{aligned} n_{2\max}(T_0)_{1,2} \; &= \; N - \frac{1}{2} - \frac{1}{2\gamma T_0} \pm \\ &\pm \; \sqrt{\left[N - \frac{1}{2} - \frac{1}{2\gamma T_0}\right]^2 + \frac{\mu T_0\,(N-1) + 2N - 1}{\gamma T_0}}. \end{aligned} \tag{56}$$

In a similar way, taking into account relation (48), the condition (1) that the load ρ_2 of the transmission media of the respective collision domain mustn't exceed the limit value ρ_0 takes, for this case, the form

$$\frac{n\nu\gamma\,(2N - n - 1)}{V\,(N-1)} \le \rho_0 \tag{57}$$

or, after some simple transformations,

$$\nu\gamma n^2 - \nu\gamma(2N-1)n + \rho_0 V(N-1) \geq 0. \tag{58}$$

From here it is easy to obtain the expression for the maximum number $n_{2max}(\rho_2)$ of stations in a collision domain, calculated taking into account just the restriction (1) regarding the limit load $\rho_2 = \rho_0$ of the transmission media in this domain,

$$n_{2\max}(\rho_0)_{1,2} = N - \frac{1}{2} \pm \sqrt{\left[N - \frac{1}{2}\right]^2 - \frac{\rho_0 V(N-1)}{\nu\gamma}}. \tag{59}$$

Combining solutions (56) and (59), one can obtain the expression for the maximum number n_{2max} of stations in the collision domain calculated taking into account restriction (1) regarding the limit load $\rho_2 = \rho_0$ of the transmission media in this domain, as well as restriction (4) regarding the limit mean time $T_2 = T_0$ of the packet transfer in the network

$$\begin{aligned} n_{2\max} = N - \frac{1}{2} + &\\ + \min\Bigg\{ -\frac{1}{2\gamma T_0} \pm \sqrt{\left[N - \frac{1}{2} - \frac{1}{2\gamma T_0}\right]^2 + \frac{\mu T_0\,(N-1) - 2N + 1}{\gamma T_0}}; &\\ \pm\sqrt{\left[N - \frac{1}{2}\right]^2 - \frac{aV(N-1)}{\nu\gamma}}\Bigg\}. & \end{aligned} \tag{60}$$

Obviously, upon using formulas (56), (59) and (60) there should be taken into account the direct relation among parameters n and N (see relation (6)).

In a similar way, to asymptotic solutions for T_2 and ρ_2 (see relations (51) and (53), respectively) correspond asymptotic expressions for $n_{2max}(T_0)$, $n_{2max}(\rho_0)$ and n_{2max}.

At $m \to \infty$, condition (4), considering relation (50), takes the form

$$\frac{2}{\mu - 2n\gamma} \leq T_0, \tag{61}$$

from which one can obtain the asymptotic, simpler expression for $n_{2max}(T_0)$

$$\lim_{m\to\infty} n_{2max}(T_0) = \frac{\mu T_0 - 2}{2\gamma T_0} \tag{62}$$

Likewise, at $m \to \infty$, condition (1), considering relation (53), takes the form

$$\frac{2n\nu\gamma}{V} \leq \rho_0, \tag{63}$$

from which one can obtain the asymptotic, simpler expression for $n_{2max}(\rho_0)$

$$\lim_{m\to\infty} n_{2\,\max}(\rho_0) = \frac{\rho_0 V}{2\nu\gamma}. \tag{64}$$

Combining solutions (62) and (64), one can obtain the asymptotic expression for the maximum number n_{2max} of stations in a collision domain

$$\lim_{m\to\infty} n_{2\,\max} = \min\left\{\frac{\mu T_0 - 2}{2\gamma T_0}; \frac{\rho_0 V}{2\nu\gamma}\right\}. \tag{65}$$

4.4 Calculation of N_{2max}.

In many practical cases, it is necessary to know, in addition to n_{2max}, the maximal number N_{2max} of stations that can be in the network. Examples of such cases include: creating a new network; expanding the number of stations in a functioning network; determining the cause of low efficiency of a functioning network, etc.

Taking into consideration relation (6) among N, n and m, restriction (54) for the value of T_2 can be turned, after some simple transformations, to the form:

$$\gamma T_0 \left(2 - \frac{1}{m}\right) N^2 - [\gamma T_0 + 1 - m\,(2 - \mu T_0)]N + m[\mu T_0 - 1] \leq 0, \tag{66}$$

whence

$$N_{2\,\max}(T_0)_{1,2} = \frac{m}{2\gamma T_0\,(2m-1)} \Big[\gamma T_0 + 1 - m(2 - \mu T_0) \pm$$
$$\pm \sqrt{[\gamma T_0 + 1 - m\,(2 - \mu T_0)]^2 - 4\gamma T_0\,(2m-1)\,(\mu T_0 - 1)}\,\Big]. \tag{67}$$

In a similar way, considering relation (6), restriction (57) for ρ_2, can be turned, after some simple transformations, to the form:

$$\nu\gamma\left(2-\frac{1}{m}\right)N^2-(\nu\gamma+\rho_0 mV)N+\rho_0 mV\leq 0, \qquad (68)$$

whence

$$N_{2\max}(\rho_0)_{1,2}=\frac{m}{2\nu\gamma(2m-1)}\Big[\nu\gamma+\rho_0 mV\pm \pm\sqrt{(\nu\gamma+\rho_0 mV)^2-4\rho_0\nu\gamma V(2m-1)}\Big]. \qquad (69)$$

Combining solutions (67) and (69), one can obtain

$$N_{2\max}=\min\{N_{2\max}(T_0);N_{2\max}(\rho_0)\}. \qquad (70)$$

To be mentioned that at $m=1$ expressions (67) and (69) can be reduced to (14) and (16) ones for networks of configuration 1, therefore:

$$N_{2\max}(T_0)|_{m=1} = N_{1\max}(T_0)=\frac{\mu T_0-1}{\gamma T_0}=\frac{VT_0-\nu}{\nu\gamma T_0}; \qquad (71)$$

$$N_{2\max}(\rho_0)|_{m=1} = N_{1\max}(\rho_0)=\frac{\rho_0\mu}{\gamma}=\frac{\rho_o V}{\nu\gamma} \qquad (72)$$

and

$$N_{2\max}|_{m=1}=\min\{N_{1\max}(T_0);N_{1\max}(\rho_0)\}= =\min\left\{\frac{\rho_0 V}{\nu\gamma};\frac{VT_0-\nu}{\nu\gamma T_0}\right\}. \qquad (73)$$

5 A model for networks of configuration 3

Let's consider a network with two switches, to each of which m homogeneous fragments-collision domains of n stations each are connected. The mean time T_3 of a packet delay in such a network is determined as

$$T_3=\frac{\lambda_{3int}T_3^{'}+\lambda_{31}T_{31}+\lambda_{32}T_{32}}{\lambda_{3int}+\lambda_{31}+\lambda_{32}}=\frac{\lambda_{3int}T_3^{'}+\lambda_{31}T_{31}+\lambda_{32}T_{32}}{n\gamma}, \qquad (74)$$

where:

$T_3^{'}$ - the mean time of a packet delay in a fragment-collision domain;

T_{31} - the mean delay time of a packet transferred between two stations connected to different collision domains of the same switch. Obviously,

$$T_{31} = 2T_3^{'}; \tag{75}$$

T_{32} - the mean delay time of a packet transferred between two stations connected to collision domains of different switches. Of course

$$T_{32} = 2T_3^{'} + T_3^{''}, \tag{76}$$

where $T_3^{''}$ is the mean delay time of a packet in the fragment-collision domain that interconnects the two switches;

λ_{3int} - the rate of packets flow generated by the n stations of a collision domain and destined to stations of the same domain;

λ_{31} - the rate of packets flow generated by the n stations of a collision domain and destined to the other $N/2 - n$ stations (stations connected to the other collision domains) of the same switch;

λ_{32} - the rate of packets flow generated by the n stations of a collision domain and destined to the $N/2$ stations connected to the another switch. Obviously,

$$\lambda_{3int} + \lambda_{31} + \lambda_{32} = n\gamma. \tag{77}$$

Let λ_{3c} is the rate of packets flow in media c that interconnects the two switches. Taking into account that the rate of packets flow generated by one station and destined to certain another station of the network is $\gamma/(N-1)$, the following relations take place:

$$\lambda_{3int} = n\frac{\gamma}{N-1}(n-1) = n\gamma\frac{n-1}{N-1} = \lambda_{2int}; \tag{78}$$

$$\lambda_{31} = n\frac{\gamma}{N-1}\left(\frac{N}{2} - n\right) = n\gamma\frac{N-2n}{2(N-1)}; \tag{79}$$

$$\lambda_{32} = n\frac{\gamma}{N-1}\frac{N}{2} = n\gamma\frac{N}{2(N-1)}; \tag{80}$$

$$\lambda_{3c} = 2\frac{N}{2}\cdot\frac{\gamma}{N-1}\cdot\frac{N}{2} = \gamma\frac{N^2}{2(N-1)} \approx \frac{\gamma N}{2}. \tag{81}$$

Relation (81) shows that the rate of flow through media c that interconnects the two switches is approximately equal to half of the rate of the summary packets flow generated by all N stations. Thus, regarding the restriction not to exceed the load ρ_0 of the transmission media, c is the critical segment. At the same time this segment can be easily used in full duplex mode and then the ρ_0 value can be bigger in comparison with the one for the collision domains; in this case, for segment c it is sufficient to consider only restriction (4) regarding the value of $T = T_3$.

To be mentioned that the rate of packets flow in the transmission media of a collision domain and, respectively, the load of this media, are the same as for the network of configuration 2 (see formula (41)). Therefore

$$T_3^{'} = T_2^{'}. \tag{82}$$

We will consider that in the transmission through the c media, the duplex operation mode is used. Then, in a similar way as (9), the following takes place

$$T_3^{''} = \frac{1}{\mu - \frac{\lambda_{3c}}{2}} = \frac{2}{2\mu - \lambda_{3c}} \tag{83}$$

and, considering relation (69), one can obtain

$$T_3^{''} = \frac{4(N-1)}{4(N-1)\mu - \gamma N^2}. \tag{84}$$

Thus, considering relations (75), (76), (78)-(82) and (84), the formula (74) can be replaced with

$$T_3 = \frac{T_2^{'}(n\gamma + \lambda_{31} + \lambda_{32}) + T_3^{''}\lambda_{32}}{n\gamma} = T_2 + T_3^{''}\frac{N}{2(N-1)}. \tag{85}$$

So, the use of configuration 3 instead of configuration 2, at preserving the total number of collision domains, results in the growth of mean delay time by value

$$T_3^{''}\frac{N}{2(N-1)} = \frac{2N}{4\mu(N-1) - \gamma N^2} \approx \frac{2}{4\mu - \gamma N} \tag{86}$$

in the case of full duplex operation mode in the c transmission media and - by value

$$T_3^{''}\frac{N}{2(N-1)} = \frac{N}{2\mu(N-1)-\gamma N^2} \approx \frac{1}{2\mu-\gamma N} \qquad (87)$$

in the case of duplex operation mode in the c transmission media.

Substituting T_2 and $T_3^{''}$ in formula (85) with expressions, (45) and (84), respectively, one can obtain

$$T_3 = \frac{2N-n-1}{\mu(N-1)-n\gamma(2N-n-1)} + \frac{2N}{4\mu(N-1)-\gamma N^2}. \qquad (88)$$

Condition (4) not to exceed the mean delay time T_0 of packets in the net for the full duplex operation mode in the c media, taking into account relation (70), takes the form

$$\frac{2N-n-1}{\mu(N-1)-n\gamma(2N-n-1)} + \frac{2N}{4\mu(N-1)-\gamma N^2} \leq T_0 \qquad (89)$$

At the same time it should be mentioned that the condition not to exceed load ρ_0 of the transmission media of a collision domain is the same as for configuration 2 (see (57)). Obviously, it is supplementary required the observance of the restriction

$$\rho_{3c} < 1, \qquad (90)$$

but this is, indirectly, taken into account by condition (88) for the T_3 value.

6 Calculations concerning the performance of some networks

6.1 Initial data.

In the aim of practical research of dependences among the parameters of networks of different configurations, calculations were made for a concrete set of initial data, some of which are the following:

$V = 10$ Mbps;

$\beta = 0,01$ mes/s;
$L = 1$ Mbit;
$\nu = 0,006392$ Mbits;
$s = 0,000208$ Mbits;
$\rho_0 = 0,3$;
$a = 0$.

The average length v of a packet is calculated, in concordance with the specifications of the standard IEEE802.3 [6], as (12784 + 576)/2 bits = 6392 bits, where 12784 bits is the maximum size and 576 bits - the minimum size of an Ethernet packet. The control information per an Ethernet packet constitutes $(18 + 8) \cdot 8$ bits = 208 bits.

6.2 Comparison of models 1a and 1b for networks of configuration 1.

In calculations, for models described in subsections 3.1 and 3.2 the following supplementary initial data are used:
$\tau = b = 285$ bt;
$\theta_0 = 0,35$ s.

As criteria for the comparison of the two models, the following relative differences are used: of mean time of a packet delay δT_1, of load $\delta\rho_1$ of the transmission media and of maximum number δN_{1max} of stations in the net:

$$\delta T_1 = \frac{T_{1b} - T_{1a}}{T_{1b}} \cdot 100\%; \quad (91)$$

$$\delta\rho_1 = \frac{\rho_{1b} - \rho_{1a}}{\rho_{1b}} \cdot 100\%; \quad (92)$$

$$\delta N_{1\max} = \frac{N_{1a\max} - N_{1b\max}}{N_{1a\max}} \cdot 100\% \quad (93)$$

The results of calculations of these indicators, using the respective expressions from subsections 3.1 and 3.2, are systemized in Table 1.

From this table one can notice that by these three indicators models 1a and 1b differ only slightly from each another. The biggest difference refers to the mean time T_1 of a packet delay, but it doesn't exceed 1,7%.

Table 1. Dependence of parameters δT_1, $\delta\rho_1$ and δN_{1max} on the number N of stations

N, stations	Message size 1000 bits			Message size 1 Mbit		
	$\delta\rho$, %	δT, %	δN_{max}, %	$\delta\rho$, %	δT, %	δN_{max}, %
50	2.73E-06	1.644	0.0158	0.0027	1.563	0.0158
100	5.46E-06	1.644	0.0158	0.0055	1.482	0.0158
150	8.19E-06	1.644	0.0158	0.0082	1.400	0.0158
200	1.09E-05	1.644	0.0158	0.0109	1.317	0.0158
250	1.36E-05	1.644	0.0158	0.0136	1.235	0.0158
300	1.64E-05	1.644	0.0158	0.0164	1.151	0.0158
350	1.91E-05	1.644	0.0158	0.0191	1.068	0.0158
400	2.18E-05	1.644	0.0158	0.0218	0.984	0.0158
450	2.46E-05	1.644	0.0158	0.0246	0.899	0.0158
500	2.73E-05	1.644	0.0158	0.0273	0.814	0.0158

This confirms the opportunity of using, at low loads of transmission media, a reductive model without taking into consideration the time of signal propagation through media and the supplementary delay of packets caused by possible collisions.

6.3 Results of other calculations.

In this section, data regarding the networks of configuration 1 refer to model 1a. In Figure 2, for networks of configuration 1, there are shown some results regarding the dependence of the maximum number $N_{1max}(T)$ of stations in the network, calculated using constraint (12) regarding the limit mean time T_0 of a packet delay. One can notice that, although the value of $N_{1max}(T)$ grows with the increase of θ_0 value, this increase is not linear.

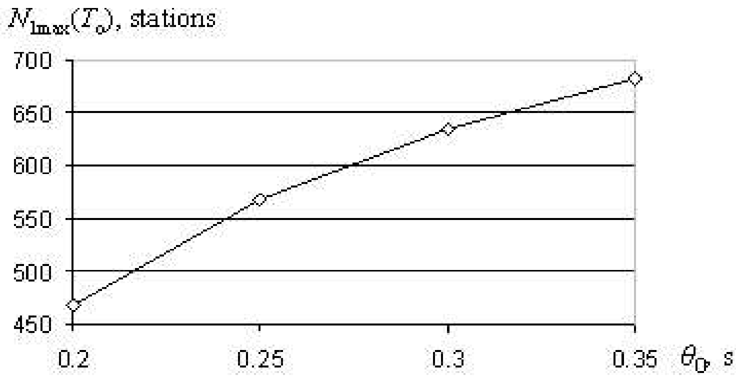

Figure 2. Dependence of $N_{1max}(T_0)$ on θ_0

Dependences of $N_{max}(T_0)$ and $N_{max}(\rho_0)$ on θ_0 and the number m of fragments-collision domains in the network, for configurations 1 and 2, are shown in Figure 3. Calculations were made according to formulas (67) and (69).

Obviously, $N_{max}(\rho_0)$ doesn't depend on θ_0. Depending on the case, there takes place $N_{max}(T_0) > N_{max}(\rho_0)$ or $N_{max}(T_0) < N_{max}(\rho_0)$. Already at relatively not big values of m, the respective dependences are almost linear. At the same time if in the case of $\theta_0 = 0,35$ s the value of $N_{max}(T_0)$ grows with the increase of m, then in cases $\theta_0 = 0,3$ s and $\theta_0 = 0,25$ s the value of $N_{max}(T_0)$ decreases at first and only afterwards it grows once with the increase of m. Moreover, in the case of $\theta_0 = 0,2$ s, the growth of the number m of fragments results in the decrease of the maximum number $N_{max}(T_0)$ of stations in the network. Thus, the fragmentation doesn't necessarily permit to increase the number of stations in the network; it may be even vice versa.

The two criteria $N_{max}(T_0)$ and $N_{max}(\rho_0)$ are combined into a single one by formula (70) for N_{max}. The dependence of N_{max} on θ_0 and m, at the same initial data as for cases from Figure 3, are shown in Figure 4.

It is important to estimate the dependence of the mean time θ of a message delay by the number N of stations and the number m

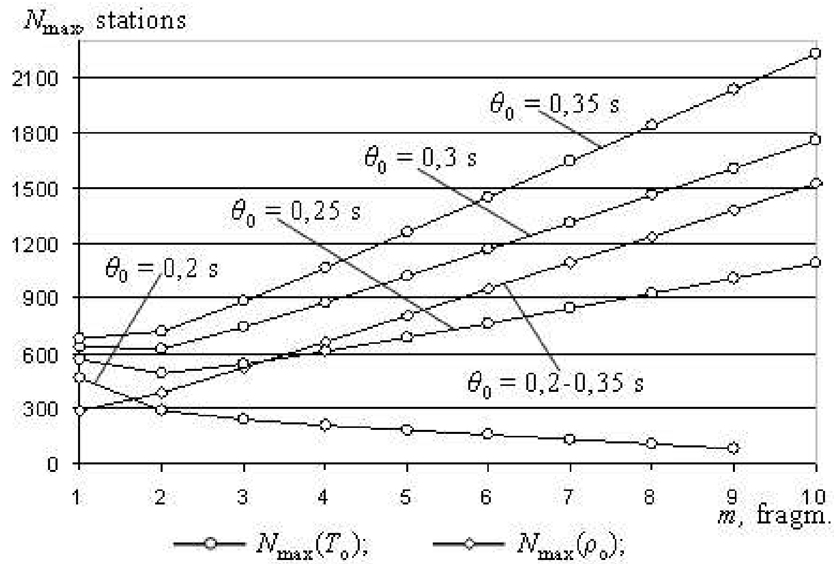

Figure 3. Dependence of $N_{max}(T_0)$ and $N_{max}(\rho_0)$ on θ_0 and m, configurations 1 and 2

of network fragments. This dependence, for configuration 2 (at $m = 1$, configuration 1) for cases encompassed by initial data described in subsection 6.1, can be traced from Figure 5. The respective calculations were made according to formula (3), in which T was substituted by T_2 which, in turn, was determined according expression (45).

From Figure 5 there can be noticed that, although at a small number m of fragments the θ value differ considerably for networks of different number N of stations, with the increase of m value these differences gradually decrease. Also, there are cases, for which the increase of m results in the increase of the mean time θ of messages delay – see, for example, the case for $N = 200$ from Figure 5. Thus, the increase of the number of network fragments doesn't necessarily result in the decrease of the mean delay time of messages transferred among stations.

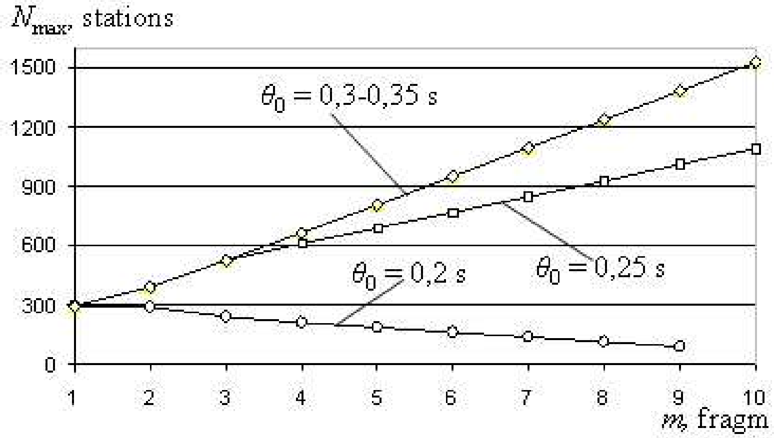

Figure 4. Dependence of N_{max} on θ_0 and m, configurations 1 and 2

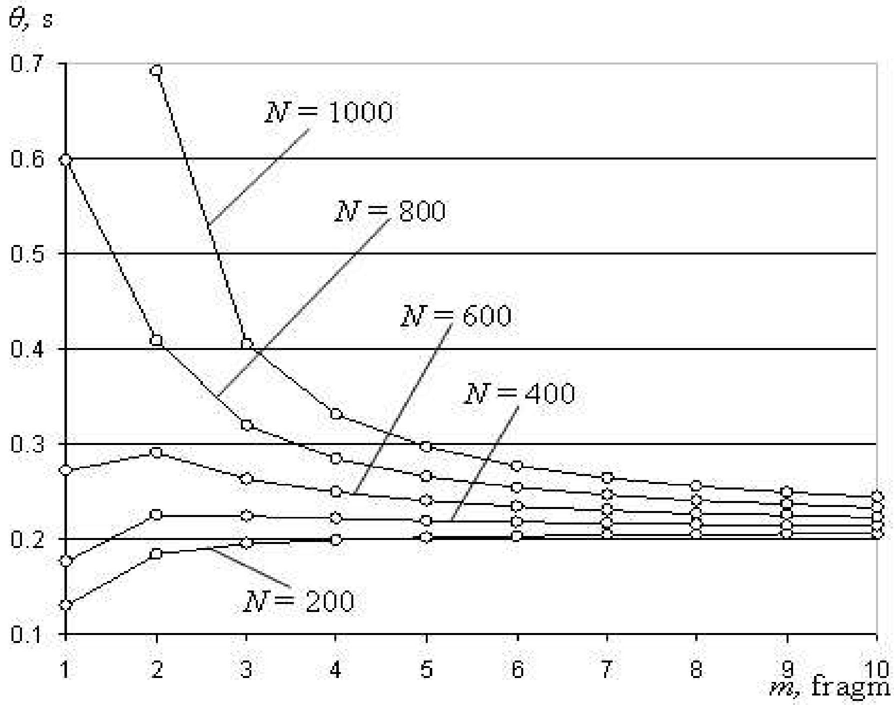

Figure 5. Dependence of θ on N and m at $\theta_0 = 0,25$s, configurations 1 and 2

Data similar to those from Figure 5, but for networks of configuration 3, are shown in Figure 6. The respective calculations are made according to formula (3), taking into account that, in this case, $T = T_3$ and T_3 is determined according to expression (88).

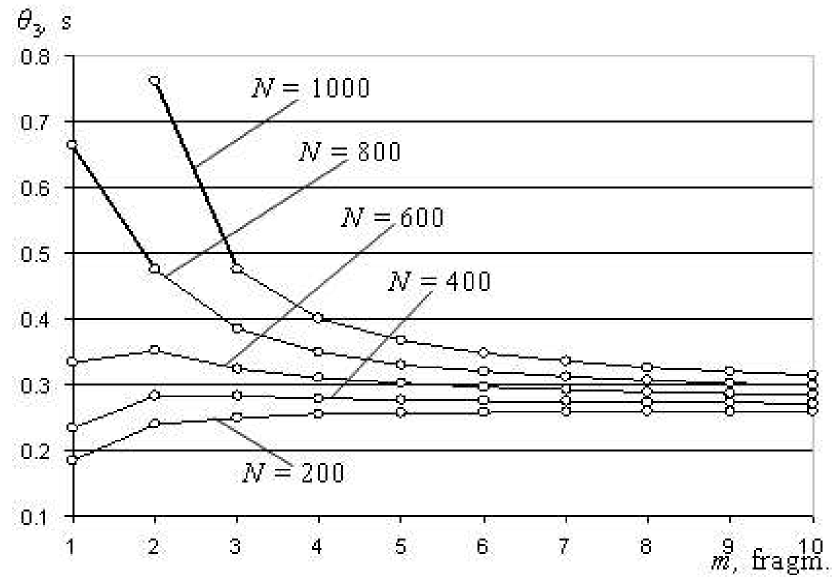

Figure 6. Dependence of θ_3 on N and m at $\theta_0 = 0,25$s, configurations 3

Graphics from Figure 6 show that these dependences have the same character with those for the respective networks of configuration 2 (see Fig. 5), only the value of the mean time θ_3 of messages delay is bigger. So, if the adding of switches results in the decrease of the load of some separate fragments, then the time of messages delay can be, at the same time, longer.

Of special interest is the comparison of configurations 1 and 2 in terms of the mean time θ of messages delay. For this purpose calculations were made for the relative increase $\delta\theta_{12}$ of the mean delay time θ_1

of messages in networks of configuration 1 comparatively to the mean delay time θ_2 of messages in networks of configuration 2:

$$\delta\theta_{12} = \frac{\theta_1 - \theta_2}{\theta_1} \cdot 100\%. \quad (94)$$

The results obtained are shown in Figure 7. Data from Figure 7 show that the difference between the θ_1 and θ_2 values can be considerable. At a small number of stations in the network, the mean time of message transfer is smaller for networks of configuration 1 comparatively to those of configuration 2 (in Figure 7, at $N = 200$ stations and $N = 400$ stations), and at a relatively large number of stations in the network (in Figure 7, at $N = 600$ stations and $N = 800$ stations) vice versa: $\theta_1 > \theta_2$. Also, in some cases (in Figure 7, at $N = 200$ stations) the increase of the number of fragments results with the decrease of the parameter $\delta\theta_{12}$, and in other cases (in Figure 7, at $N = 400$ stations, $N = 600$ stations and $N = 800$ stations) on the contrary leads to the growth of parameter $\delta\theta_{12}$.

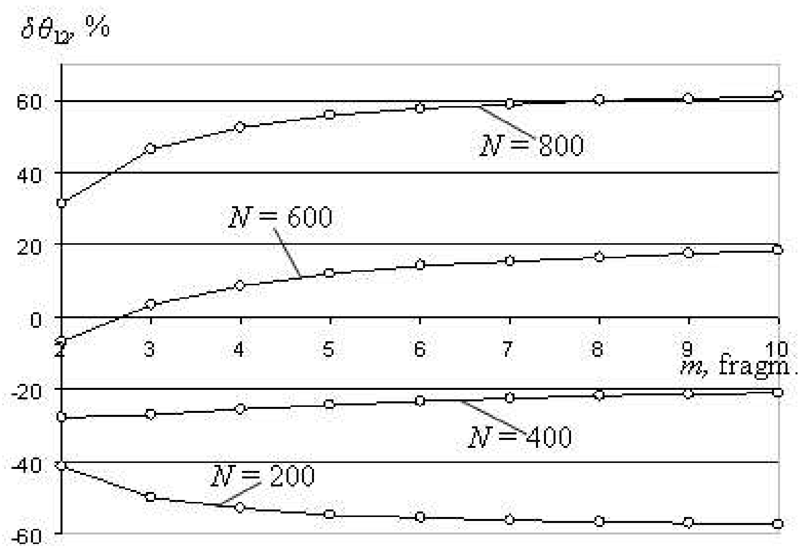

Figure 7. Dependence of $\delta\theta_{12}$ on N and m at $\theta_0 = 0,25$s

In a similar mode, it is of interest the comparison of configurations 2 and 3 by the mean time θ of message delay. For this purpose calculations were made for the relative difference $\delta\theta_{32}$ of the mean time θ_3 of message delay for configuration 3 comparatively to the mean time θ_2 of messages delay for configuration 2:

$$\delta\theta_{32} = \frac{\theta_3 - \theta_2}{\theta_3} \cdot 100\%. \tag{95}$$

The results obtained are shown in Figure 8. Data from Figure 8

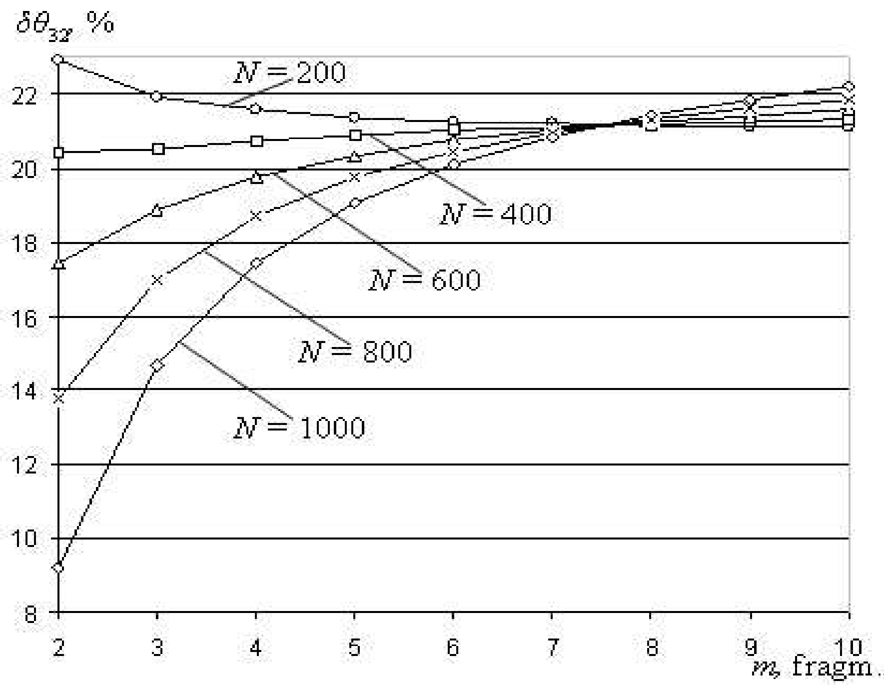

Figure 8. Dependence of $\delta\theta_{32}$ on N and m at $\theta_0 = 0,25$s

show that there exists an m value at which the value of parameter $\delta\theta_{32}$ doesn't depend of the number N of stations in the network (in Figure 8 this is $\approx 7-8$ fragments). The increase of the number of fragments results in the decrease of differences between the values of

θ_2 and θ_3 at a relatively small number of stations in the network (in Figure 8, at $N = 200$ stations) and in the increase of these differences at a relatively large number of stations in the network (in Figure 8, at $N \geq 400$ stations). At a certain number of stations in the network, the number m of fragments in the network doesn't influence or have little influence on the value of parameter $\delta\theta_{32}$ (in Figure 8, $N \approx 350$ stations).

The dependence of load ρ on N and m presents interest as well. Some results of the calculations referring to these dependences for the set of initial data described in subsection 6.1 are shown in Figure 9.

From Figure 9 there can be noticed, as expected, the growth of load ρ once with the increase of number N of stations and the decrease of this load once with the increase of the number m of collision domains in the network. The increase of m results also in the decrease of differences between load ρ for networks with a different number N of stations.

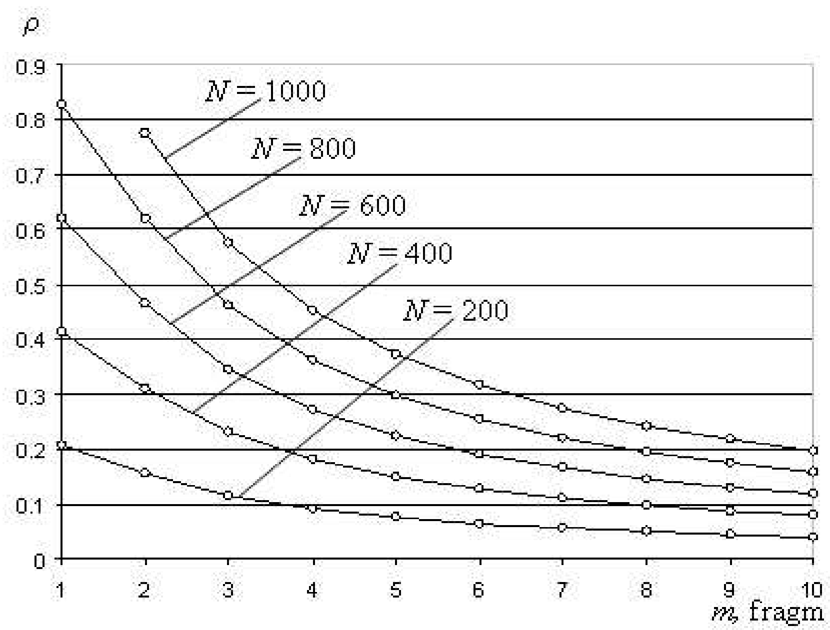

Figure 9. Dependence of ρ on N and m at $\theta_0 = 0,25$s

It is of interest the estimation of differences between load ρ_1 of the transmission media for networks of configuration 1 and load ρ_2 of the transmission media for networks of configuration 2 (for networks of configuration 3, excluding the case of the transmission media c between the two switches, the equality $\rho_3 = \rho_2$ takes place). In this purpose, there were made calculations for parameter $\delta\rho_{12}$, which is determined as

$$\delta\rho_{12} = \frac{\rho_1 - \rho_2}{\rho_1} \cdot 100\%. \tag{96}$$

Some results of calculation of $\delta\rho_{12}$ are systemized in Table 2.

Table 2. The dependence of $\delta\rho_{12}$ on N and m, %

m, fragm.	N, stations			
	200	400	600	800
2	25,12	24,94	25,00	25,03
3	44,44	44,31	44,35	44,38
4	56,04	56,17	56,13	56,23
5	63,77	63,92	64,03	63,97
6	69,57	69,49	69,35	69,41
7	73,43	73,37	73,39	73,40
8	76,33	76,51	76,61	76,54
9	79,23	78,93	79,03	78,96
10	81,16	80,87	80,97	81,02

Data from Table 2 show that the value of $\delta\rho_{12}$ very little depends on the total number N of stations, but depends significantly on the number m of network fragments. In a graphic form, this dependence for the case of $N = 800$ stations is shown in Figure 10.

Calculations were made for Ethernet networks with data transfer speed of 10 Mbps, but the elaborated models and procedure can be applied to FastEthernet and Gigabit Ethernet networks, as well.

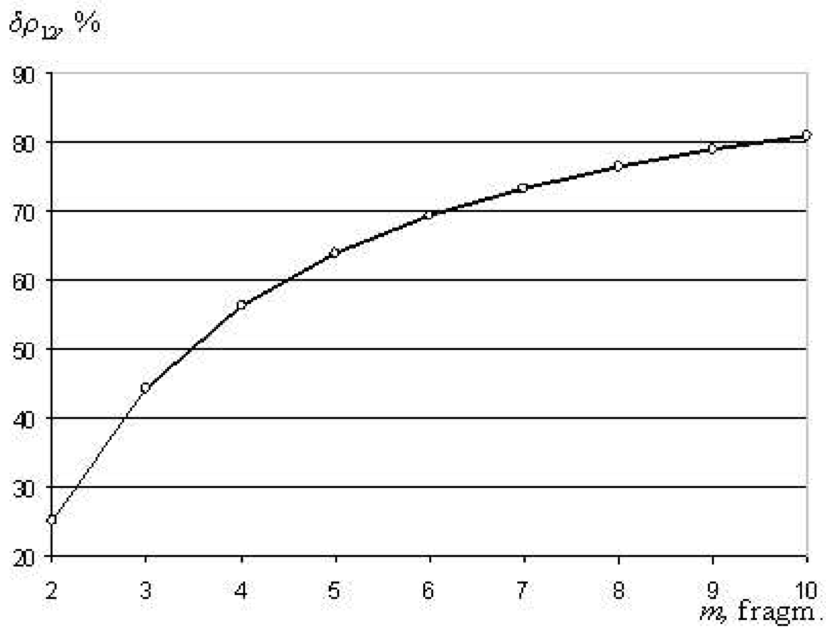

Figure 10. Dependence of $\sigma\rho_{12}$ on m at $N = 800$ stations and $\delta_0 = 0,25$s.

7 Conclusions

The analysis of performance features of different networks with frame memorization by switches confirms the necessity of case dependent special investigations on the configuration of concrete Ethernet networks. It was determined, for example, that:

1) the fragmentation doesn't necessarily permit the increase of the number of stations in the network, in may be even vice versa;

2) the increase of the number of network fragments doesn't necessarily results in the decrease of the mean time of messages transfer among stations;

3) if the adding of switches results in the load decrease for some fragments, then the mean time of messages transfer in the network

can be, at the same time, longer;

4) there exists a value of number m of fragments for which the relative difference $\delta\theta_{32}$ of the mean time θ_3 of messages delay in a network with two switches comparatively to the mean time θ_2 of messages delay in a network with one switch doesn't depend, practically, on the number N of stations in the network;

5) at certain number N of stations in the network, the number m of network fragments doesn't influence or little influences the value of parameter $\delta\theta_{32}$;

6) the relative difference $\delta\rho_{12}$ of load ρ_1 of the transmission media of a network with one collision domain from load ρ_2 of the transmission media of a network with one switch little depends on total number N of stations, but depends significantly on number m of network fragments.

References

[1] F.Tobagi, V.Hunt. *Performance Analysis of Carrier Sense Multiple Access with Collision Detection.* Computer Networks, October-November 1980.

[2] B.Stuck, E.A. Arthurs. *Computer Communications Network Performance Analysis Primer.* Englewood Cliffs, NJ: Prentice Hall, 1985.

[3] W. Stallings. *Local and Metropolitan Area Networks*, 5th Edition, New York, NY, Macmillan Publishing Co., 1997.

[4] G. Thomas. *Improving the Performance of Ethernet Networks*, http://ethernet.industrial-networking.com/articles/i06articleimproving.asp

[5] T.G. Robertazzi. *Computer networks and systems: queuing theory and performance evaluation.* – New York: Springer-Verlag, 1994.

[6] IEEE Std 802.3-2002. *Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.* New York: IEEE, 8 March 2002.

Ion Bolun, Victor Andronatiev

Received July 13, 2006

Academy of Economic Studies of Moldova
59, Banulescu Bodoni str., off. 401
Chişinău, MD−2005, Moldova
Phone: (373+2) 22-27-65
E–mail: *bolun@ase.md*

Computer Science Journal of Moldova, vol.**14**, no.2(41), 2006

Macroeconomic growth determining

E.Naval

Abstract

In the offered work two approaches to the problem of economic growth rate determining is considered. The first approach assumes, that economic growth rate is set exogenously, so its value is known while at the second approach economic growth rate is determined endogenously, i.e. its value is unknown. In the model there are included such components of the economic growth, which are determined by the technical progress containing the human factor during training, and also sector of research and development ($R\&D$). Research and development are estimated in two ways: one, supposing development of new technologies and another, using already available developed technologies, and on their basis developing new manufactures. Estimations of expenses both on the first, and on the second way are resulted.

Introduction

Work consists of three sections. In the first section the Solow model [11] with the exogenously set growth rate of technical progress is investigated. The equation of dynamics is brought and the golden rule of accumulation is written. Using dynamics equation for Solow model, the optimization problem is formulated and it is solved by the maximum principle. The solution of this problem determines an optimal trajectory for consumption function and, accordingly, an optimal rate of accumulation.

In the second section Aghion-Howitt [1-2] model with endogenous growth caused by uncertainty in which researches provide qualitatively improved innovations is discussed. The dynamics equation of the model is brought, and economic growth rate is determined, proceeding from the known values of the model parameters offering basic characteristics of economic development.

In the third section the Romer [13-15] endogenous growth model is examined. Estimation cost of the original invention which underlies qualitatively improved innovational processes and the cost of the license invention providing qualitatively improved innovations are brought.

1. Optimal rate of accumulation in exogenous growth economic model

Let's consider Solow exogenous growth economic model with technical progress.

$$\begin{aligned} Y &= K^{\alpha}(EL)^{1-\alpha} && (1)\\ Y &= C + I && (2)\\ C &= (1-s)Y, \qquad 0 < (1-s) < 1 && (3)\\ \Delta K &= I - \delta K, \qquad 0 < \delta < 1 && (4)\\ \Delta L/L &= n, \qquad n > 0 && (5)\\ \Delta E/E &= g, \qquad g > 0 && (6) \end{aligned}$$

Here Y, C, I, K are endogenous variables, L, E are exogenous variables. Y is gross domestic product; C is total consumption; I is total investments; K is fixed capital; L is total workers quantity occupied in manufacture; E is some variable named work efficiency. s, δ, n, g are parameters determining rate of accumulation, fixed capital amortization rate, growth rate of the labor occupied in manufacture, growth rate of the labor caused by technological progress, it takes into account quantity of workers L and efficiency of everyone working E. Production function (1) determines, that total output Y depends on the available capital quantity K and on the quantity of effective workers $L \cdot E$. The increase of efficiency of work E is equivalent to increase in a labor L. Using substitution, $\frac{Y}{EL} = y$; $\frac{K}{EL} = k$, we shall transform the first equation to the following one $y = k^{\alpha}$. From the equation (4) we shall receive

$$\dot{k} = i - \delta k,$$

where $i = \frac{I}{EL}$. Further, using (2) and (3), we shall transform (4) to the form

$$\dot{k} = sy - \delta k; \quad \dot{k} = sk^{\alpha} - \delta k.$$

After simple transformations we shall receive, that

$$\frac{\dot{K}}{EL} = \frac{I - \delta K}{EL} = \frac{sY - \delta K}{EL} = sk^{\alpha} - \delta k. \qquad (7)$$

And, finally,

$$\frac{\dot{k}}{k} = sk^{\alpha-1} - \delta. \qquad (8)$$

The stationary condition takes the form of:

$$\frac{\dot{k}}{k} = \frac{\dot{K}}{K} - \frac{\dot{L}}{L} - \frac{\dot{E}}{E} = 0. \qquad (9)$$

And, substituting (8) in (9), we shall receive

$$\frac{\dot{k}}{k} = sk^{\alpha-1} - (\delta + n + g).$$

Solving this equation in respect to k we shall obtain

$$k^{*} = \Big(\frac{s}{\delta + n + g}\Big)^{1/(1-\alpha)}. \qquad (10)$$

Let's notice, that in steady state the value k is as greater as the values s and α (rate of accumulation and the interchangeability constant of work and capital) are greater and as smaller as the values δ, n and g are smaller. So the growth rate of the capital per capita is positive and is equal to $g + n$.

$$\frac{\Delta k}{k} = \frac{\Delta K}{K} - \frac{\Delta L}{L} - \frac{\Delta E}{E} = 0; \quad \frac{\Delta K}{K} = g + n > 0. \qquad (11)$$

From the equation (3) we shall receive equation for consumption function in a steady state

$$c^{*} = y - i; \quad c^{*} = f(k^{*}) - k^{*}(n + g + \delta). \qquad (12)$$

As $\frac{I}{EL} = \frac{\Delta K}{K} + \delta$ from (11) it is received

$$i^* = k^*(n + g + \delta). \quad (13)$$

From (12) follows, that the maximal value c^* is achieved when

$$\frac{dc^*}{dk^*} = f'(k^*) - (n + g + \delta) = 0 \Longrightarrow f'(k^*) = n + g + \delta$$

$$f'(k^*)k^* = sf(k^*); \quad s^* = \frac{f'(k^*)k^*}{f(k^*)}. \quad (14)$$

Expression (14) determines "golden rule" of accumulation. Such premises, as the constant returns to scale production function, competitive markets, exogenous savings and exogenous growth of technology, made this model an extremely convenient one to solve the case for steady state.

Let's take advantage of a maximum principle for determination of the maximal rate of accumulation which satisfies to the "golden rule". We shall consider the problem of optimal control in which the utility function on infinite orison is maximized and the restriction is the dynamics equation for the Solow model. As a control variable the consumption function takes place.

$$\int_0^\infty \frac{c(t)^{1-\theta}}{1-\theta} e^{-\rho t} dt \implies max \quad (15)$$

$$k' = [f(k) - k(n + g + \delta)] - c(t) \quad (16)$$

$$k(0) = k_0 \quad (17)$$

Profit function is written as $V = f(k) - rk - w$ where V is the maximal profit rent equal to zero and r is characterized by the following ratio: $\frac{\partial f(k)}{\partial k} = r, \quad f(k) = rk + w.$

So, let's write down the Hamiltonian of the system (15) - (17)

$$H(c(t), k(t), \pi(t)) = \frac{c(t)^{1-\theta}}{1-\theta} e^{-\rho t} + \pi(t) \cdot [f(k) - c(t) - k(n+g+\delta)]. \quad (18)$$

Further the conditions of the maximum principle follow:

$$\frac{\partial H}{\partial c} = 0; \qquad \frac{\partial H}{\partial \pi} = \dot{k}; \qquad \dot{\pi} = \rho\pi - \frac{\partial H}{\partial k}. \tag{19}$$

And the transversality condition is written down as

$$\lim_{t\to\infty} \pi(t)\cdot k(t)\cdot e^{-\rho t} = 0$$

After simple transformations we receive the following three equations:

$$e^{-\rho t}\cdot c(t)^{-\theta} - \pi(t) = 0 \tag{20}$$

$$\dot{k} = s\cdot f(k) - k\cdot(n+g+\delta) \tag{21}$$

$$\dot{\pi} = \pi\cdot(\rho - \dot{f}(k) + (n+g+\delta)). \tag{22}$$

And, since $\frac{\partial f(k)}{\partial k} = r$, the equation (21) will become $\dot{\pi} = \pi\cdot[\rho - r + (n + g + \delta]$. In consequence, we receive system of two differential equations which are necessary to solve:

$$\begin{aligned} \dot{k}(t) &= w - k(t)\cdot(n+g+\delta) - c(t) \\ \dot{\pi}(t) &= \pi(t)[\rho - r + (n+g+\delta)] \qquad (23) \\ k(0) &= 0; \quad \lim_{t\to\infty} \pi(t)\cdot k(t)\cdot e^{-\rho t} = 0. \qquad (24) \end{aligned}$$

Introducing some notations, we obtain

$$\begin{aligned} \dot{k}(t) &= -\beta k(t) - c(t) + w \\ \dot{\pi}(t) &= \pi(t)\cdot\gamma \qquad (25) \\ k(0) &= 0; \quad \lim_{t\to\infty} \pi(t)\cdot k(t)\cdot e^{-\rho t} = 0. \qquad (26) \end{aligned}$$

In order to solve this problem, the equation (22) should be differentiated on t

$$\frac{\dot{c}(t)}{c(t)} = -\frac{\gamma}{\theta} = \frac{r - (n+g+\delta) - \rho}{\theta} = \dot{\gamma}.$$

Hence, $c(t)$ grows with a constant exponential growth rate. To receive the solution in $k(t)$ we act as follows. To resolve equation (22) in accordance with $k(t)$ we shall take an integrating multiplier $e^{\beta t}$, and we shall receive

$$\dot{k}(t) \cdot e^{\beta t} + \beta \cdot e^{\beta t} \cdot k(t) = c_0 \cdot e^{-\grave{\gamma} t} + w.$$

Common solution of this equation depends on two constants. Taking into account the transversallity condition one of the two integration constants in the common solution for $k(t)$ is determined. The second constant is determined from the condition $k(0) = k_0$. As a result, the following is received:

$$k(t) = k_0 \cdot e^{\grave{\gamma} t} \quad (27)$$

$$c(t) = k_0 \cdot [\beta - \grave{\gamma}] \cdot e^{\grave{\gamma} t} + w. \quad (28)$$

2. Endogenous growth economic model

Solow neoclassical model was first modified by Cass [3] and (independently) by Koopmans in [7] following a mathematical version of Frank Ramseys model [12](RCK). The endogenous saving rate in it is deeply different from the Solow model.

The Aghion-Howitt [1-2] endogenous growth economic model represents such approach when the endogenous economic growth is examined under uncertainty conditions. In this model the economic growth is generated by a sequence of qualitatively improved innovations which are the investigations connected to economic activities in a random way. This model has natural property when the last innovation is presented out-of-date in comparison with the present innovation. In other words this model is the model of destructive innovation creation. Starting from Romers [13] the basic idea of endogenous technological progress is obtained

$$u(y) = \int_0^{\infty} y(\tau) e^{-r\tau} d\tau.$$

Here y is the production volume of the final consumer goods, r is the rate of preference in time equal to the interest rate.

$$y = Ax^{\alpha}, \quad 0 < \alpha < 1. \quad (1)$$

A is the parameter of technological progress. Innovations increase A by size of a constant factor $\gamma = \frac{A_{t+1}}{A_t}$, here t is an innovational index but not the time.

$$L = x + n. \tag{2}$$

L is the volume of labor, x are the works used by manufacture of the intermediate goods (one for one). It means that for production of one unit of the intermediate goods one unit of labor is used. If the price of intermediate goods is denoted by P, then from the profit maximization condition the demand function for the produced intermediate goods is deduced.

$$Ax^{\alpha} - Px = \pi; \quad \frac{\partial \pi}{\partial x} = \alpha A x^{\alpha-1} - P = 0; \quad P = \alpha A x^{\alpha-1}, \tag{3}$$

hence, it follows that the manufacturer will produce final goods until the marginal product will be equal to its price. Innovations appear casually with the Poisson rate of appearance equal to λn where λ indicates productivity of research technologies. If there is one additional unit of labor, the probability of new innovations appearance grows by value λ. And because Poisson distribution is additive, the rate expectation of n researchers appearance is equal to λn. V_{t+1} is a profit function caused by innovation $t+1$; λV_{t+1} is an expected gross revenue of each researcher for the period before appearance of the following innovation. The net expected profit from the use of n units of labor in research area for production of an innovation t is equal to:

$$\lambda n V_{t+1} - w_t n = 0, \tag{4}$$

here w_t is a wages rate resulted at the innovation t.

$$\lambda V_{t+1} = w_t, \tag{5}$$

and this equation settles that expected cost of one unit used in scientific researches with necessity is equal to its price. Value V_{t+1} is determined from the following equations

$$rV_{t+1} = \pi_{t+1} - \lambda n_{t+1} V_{t+1}, \tag{6}$$

$$V_{t+1} = \frac{\pi_{t+1}}{(r + \lambda n_{t+1})}. \tag{7}$$

The denominator from (7) can be interpreted as the interest rate adapted to disappearance of an innovation, and illustrates it creative-destructive effect. It is necessary to note, that from (7) it follows that the innovator can not improve $(R\&D)$ since, actually, λn_{t+1} determines the probability that innovator will lose exclusive innovational rent

$$\pi_t = \max_x [P(x) \cdot x - w \cdot x]. \tag{8}$$

Equation (8) defines income flow, which can be received from the following innovation. Here P and x are interconnected by means of the equation (3). As expression $\alpha A \cdot x^\alpha - w \cdot x$ is maximized, then $\alpha^2 \cdot A_t x_t^{\alpha-1} = w_t$ or

$$x_t = \Big(\frac{\alpha^2}{w_t/A}\Big)^{1/(1-\alpha)}. \tag{9}$$

Defining $W = \frac{w_t}{A_t}$ as the wages referred to productivity, we can express x as decreasing function of w_t, $x_t = \tilde{x}(W_t)$, $\tilde{x}' < 0$. Substituting x_t in expression for income function, we receive

$$\pi_t = A_t \cdot \alpha \cdot x_t^\alpha - w_t \cdot x_t = \Big(\frac{1}{\alpha} - 1\Big) \cdot w_t \cdot x_t = A_t \tilde{\pi}\Big(\frac{w_t}{A_t}\Big) \tag{10}$$

or $\pi_t = \tilde{\pi}(W_t), \ \tilde{\pi}' < 0$.

Two basic equations.

The first equation is an arbitrary one. It turns out from (5) that $w_t = \lambda \cdot V_{t+1}$, further from (7) it is received the following:

$$\frac{w_t}{A_t} = \lambda[\pi_{t+1}/(r + \lambda \cdot n_{t+1})]/A_t$$

$$W_t = \lambda \frac{A_{t+1}}{A_t} \cdot \tilde{\pi}(W_{t+1}) \cdot (r + \lambda \cdot n_{t+1})^{-1}.$$

And, since $\frac{A_{t+1}}{A_t} = \gamma$, we have

$$W_t = \lambda \frac{\gamma \cdot \tilde{\pi}(W_{t+1})}{r + \lambda \cdot n_{t+1}}. \tag{11}$$

The second equation is the equation for labor market clearing.

$$L = n_t + \tilde{x}(W_t). \tag{12}$$

The balanced growth SS (stable state) or growth in condition of stability is defined as the stationary solution of the system (11) - (12) at $n = n_{+1} = n$ and $W = W_{+1} = W$. That gives two equations with two unknown variables n and W. In other words, SS means, that both attraction of an additional labor n in sphere of researches, and $\tilde{x}(W)$ in sphere of manufacture remain constants in time until salary, income and volume of manufacture remain at the same level $\gamma < 1$ each time when innovations are made. In SS the equations (11) and (12) are reduced to:

$$W = \lambda \frac{\gamma \cdot \tilde{\pi}(W)}{r + \lambda \cdot n} \qquad (11(a))$$

$$L = n + \tilde{x}(W) \qquad (12(a))$$

Market forces.

It is easy to show, that from (12(a)) it follows

$$\tilde{\pi} = \frac{1-\alpha}{\alpha} \cdot Wx = \frac{1-\alpha}{\alpha} \cdot W(L-n)$$

and, having substituted this equation in (11(a)), dividing by W, we shall receive

$$1 = \frac{\lambda \cdot \dfrac{\gamma \cdot (1-\alpha)}{\alpha} \cdot (L-n)}{r + \lambda \cdot n}.$$

From this equation the statement for $\hat{n}$ in SS is obtained:

$$\hat{n} = \frac{\lambda \cdot \gamma \cdot (1-\alpha) \cdot L - \alpha \cdot r}{(\alpha \cdot \lambda + \lambda \cdot \gamma \cdot (1-\alpha))}.$$

Growth in the stable state.

In SS we have $y = A \cdot (\hat{x})^{\alpha} = A \cdot (L - \hat{n})^{\alpha}$, which implies $y_{+1} = \gamma \cdot y$.

3. P. Romer growth endogenous economic model

The other spark in growth theory has been provided by the advent of "New Growth Theory" attributed to the efforts of Romer [13] and Lucas [7]. New growth theory casts doubt on many results and assumptions of the neoclassical model. Romer's model included increasing returns, imperfect competition and endogenous technological growth. P. Romer in [14] determines technological progress as finding of new varieties and increasing amount of capital goods.

$$Y_i = A \cdot L_i^{1-\alpha} \cdot \sum (X_{ij})^{\alpha},$$

here X_{ij} is the implication of the specialized intermediate goods of the type j. At any point of time, the technology exists to create N specialized intermediate goods. Research sector in the economy uses the human capital (labor skills and labor capacities) and accumulates knowledge to produce designs for new intermediate capital goods. The growth rate in such a model is determined from the following equation:

$$\gamma = \frac{1}{\theta} \cdot \Big[\frac{L}{\eta} \cdot A^{\frac{1}{1-\alpha}} \cdot \Big(\frac{1-\alpha}{\alpha}\Big) \cdot \alpha^{\frac{2}{1-\alpha}} - \rho\Big],$$

where η is a cost of new product creation. Ii is supposed that total human capital accessible to use in economy is constant. If $\gamma < 0$, firms have insufficient incentive to expend resources on $R\&D$, N remains constant, and γ ultimately equals 0. The increase in growth rates can be provided by the following actions:

- desire to raise the rate of accumulation, which lowers ρ and θ;
- reducing the cost of a new product inventing, which lowers η;
- advanced technology, which is represented by expansions of N resulted from $(R\&D)$.

However, constant increase in N results in the tendency of income reduction. The size of innovation income – profit value for the j-th innovator, is equal to π_j/r_j, where r_j is the rate of return. Let r_1 be the rate of return, for example, in the USA, then

$$r_1 = \Big(\frac{L_1}{\eta}\Big) \cdot \frac{(1-\alpha)}{\alpha} \cdot (A_1)^{\frac{1}{(1-\alpha)}} \cdot \alpha^{2/(1-\alpha)}.$$

Here η is the invention cost; L_1 is the working amount. While imitation or adaptation of innovational products is equal to ν and is less expensive than expenditure for an innovation η, the rate of return r_2 in this country is approximately constant

$$r_2 = \left(\frac{L_2}{\nu}\right) \cdot \frac{(1-\alpha)}{\alpha} \cdot (A_2)^{\frac{1}{(1-\alpha)}} \cdot \alpha^{2/(1-\alpha)}.$$

The conclusion

Macroeconomic models of economic growth [1-15] are investigated. Such production factors as capital and labor are examined. The last is subdivided into a manpower used in industrial sector and in ($R\&D$) sector. Opportunities of each sector for maintenance of steady economic growth are studied.

We shall start from the fact, that the gain of capital is defined, basically, as a level of accumulation in national economy, particularly in private sector. By using the Solow model it is possible to determine optimal rate of accumulation corresponding to the given economic growth rate. Thus there are two problems, the first is connected with attraction of additional investments, and the second concerns a manpower which grows presumably with constant rates. The last can be restrictive for Moldova, taking into account the outflow of labor from the country and the fact, that the government has not developed a clear policy in the problem of professional training of high quality specialists. It can happen, that a bit later the country will face labor shortage in industry and agriculture. And also, as assumed, the economic growth in a physical world is constrained by diminishing returns and scarcity of resources. Such state policies as maintenance of services infrastructure, property rights protection and taxation of economic activities influences technological level and, simultaneously with the desire to accumulate, influence the growth rate in the long-term prospects.

And, as in [13], two basic fundaments of economic growth are the new ideas and advances in technology; and by creating appropriate economic incentives, the government can increase the rate of growth in a way that can make all citizens better off. In this connection in our country these ideas may by interpreted as clear definition of key economy development directions (whether it be the sphere of services, tourism, information technologies, etc.) and elaboration of professional training strategy in these brunches. Therefore the state should provide for specified branches the preferential development, stimulating investments, giving tax privileges and other.

In endogenous growth models [1-2, 13-15] the economic growth rate, proceeding from economic development in examined period, is determined. The problem of alternative use of scientific innovations is considered:

- whether to create scientific innovations, which for the certain period monopolize appropriate sphere of production, involving essential additional income until the new innovation in this area will appear, thus depriving a country of existing monopoly and additional profits,

- or to simulate and adapt advanced technologies, the application of which, at other equal conditions (capital and labor), is cheaper and, to a certain extent, more trustful.

References

[1] P.Aghion, and P.Howitt. *A model of growth through creative destruction. Econometric,* 1992, 60, pp.323–351.

[2] Aghion, Phillippe and Peter Howitt. *Endogenous Growth Theory.* Cambridge: MIT Press, 1998.

[3] Cass, David. *Optimal Growth in an Aggregate Model of Capital Accumulation,* Review of Economic Studies, July 1965, 32, pp. 233–240.

[4] D.Cass. *On capital overaccumulation in the aggregate, neoclassical model of economic growth: a complete characterization,* Journal of Economic Theory, 4, 1972, pp. 200–223.

[5] Jordi Caballe and Manuel S. Santos. *On Endogenous Growth with Physical and Human Capital.* Journal of Political Economy, 101, 1993, pp. 1042–1067.

[6] C.Jones. *R&D-Based Models of Economic Growth,* Journal of Political Economy, 103, 1995, pp. 759–784.

[7] T.C.Koopmans, *On the Concept of Optimal Economic Growth,* In: The Econometric Approach to Development Planning, Amsterdam: North Holland, 1965.

[8] Robert E. Lucas, Jr. *On the Mechanics of Economic Development,* Journal of Monetary Economics, 22, 1998, pp.3–42 (Chapter 1 in Lucas (2002)).

[9] E.S.Phelps, and R.A. Pollak. *On Second-best National Savings and Game-equilibrium Growth,* Review of Economic Studies, 1968, 35, pp. 185–199.

[10] Joseph Schumpeter. *The Process of Creative Destruction in Capitalism, Socialism, and Democracy,* New York: Harper and Row, 1975.

[11] R.M. Solow. *Growth Theory: An exposition,* 1988 edition, Oxford: Oxford University Press.

[12] Frank Ramsey. *A Mathematical Theory of Saving,* Economic Journal, 1928, 38, pp. 543–559.

[13] P.Romer. *Increasing returns and long-run growth,* Journal of Political Economy 94, 1986, pp. 1002–1036.

[14] P.Romer. *Endogenous Technical Change,* Journal of Political Economy, (October 1990): pp. 71–102.

[15] P.Romer. *Two Strategies for Economic Development: Using Ideas and Producing Ideas,* Proceedings of the World Bank Annual Research Conference 1992, supplement to the World Bank Economic Review, March 1993, pp. 63–91.

E.Naval
Institute of Mathematics and Computer Sciences,
Academy of Sciences, Moldova
5, Academiei Str., Kishinev,
MD-2028, Moldova
E–mail: *nvelvira@math.md*

Received May 30, 2006