An Algorithm for Automated Creation of a PC Database Storing Related Text Objects

D. Dimova, K. Onkov

Key Words: *Related text objects; coding system; algorithm; PC database, phytopharmac.*

Abstract: *An algorithm for automated creation of a relational PC database based on primary text documents is developed. The suggested coding system assigns a natural number to each explicit text object. Two numbers form a pair when the text objects they represent relate to each other. Input data, basic algorithmic operations and general database structure are described. The developed algorithm is applied for creation of the „Phytopharmacy" database that stores and structures the data referring to the cultures, pests and basic groups of pesticides. The presented algorithm and software would be useful methodology and tool for automated creation of databases storing related text objects.*

1. Introduction

Primary documents containing unformatted text with complex relations between the text objects (words, phrases) lead to research of applicability of relational design. Text documents are often stored unsystematically in a rather confusing file structure with an inscrutable hierarchy and little access control [1]. The databases store data in a systematic way and allow multi-user, multi-site, user-/role-specific controlled access [1]. Relational database seems like a good candidate for storage of related text objects. An information retrieval system that can search and display any unit of a text such as a sentence, a paragraph or a chapter and therefore supports user's use of the retrieved documents using the text-level structure of documents is presented in [2]. A technique for storing compound keywords is developed in [3]. This work also discusses the importance of preparing efficient methods for extraction keywords with information about the relationships between them. In this case it is expedient to build a coding system of the text objects [4]. This system facilitates the specialists in automatic input and the structuring of the data in database and subsequently their processing. At the same time the coding system can remain relatively hidden because of easier daily work of end-users.

The information about permitted pesticides for plant protection in Bulgaria is available either as printed booklet [5] or electronically in MS Word tables [6]. The Word tables are practically a full copy of the booklet pages and do not offer new data organization. The phytopharmaceutical products in the mentioned two forms are arranged in ascending order of the active substance or in alphabetic order of the three groups of pesticides. Currently, relationships between pesticides, pests and culture are visible but in different MS Word tables respectively in different sheets of the booklet. This layout does not provide for easy search of products for treating a culture against certain pest, which is the main query against this data.

This work proposes the solution to the problems of this nature by creating of the PC based relational database that stores and integrates related text objects. This type of database provides not only fast searching due to the effective relationships but also preparing queries for data integration from different tables.

The aims of this paper are:

- To present the developed algorithm for extracting text objects from primary documents, their coding, relating and storing automatically into a PC database.
- To show the application of the suggested algorithm for automated creation of a relational PC database storing the data referring to cultures, pests and basic groups of pesticides used in agriculture – fungicides, herbicides and insecticides.

2. A System for Coding Text Objects

Problem definition. Two sets M_1 and M_2 containing text objects related to one another are studied in this work. Each text object is one word or sequence of words (phrase). The set M_2 is divided into q-subsets ($q > 1$). Let assume $q=3$, i.e. the set M_2 will contain three subsets P_{M21}, P_{M22} and P_{M23}. The disjunction of the subsets is the empty set $\varnothing$. The same assumption would be applied to other values of q. All types of relationships between the sets' elements („one to one", „one to many" and „many to many") are taken into account (*figure 1*).

Building of the coding system. A specific coding system of the text objects of the sets M_1 and M_2 needs to be built because of the following reasons:

- The numbers (codes) are indexed more easily than text. It leads to faster searching of the objects.
- The coding facilitates the work of the programmer in the development of specialized software for processing text objects.

The text objects from M_1 and M_2 are used as „key" words for the purpose of creating a relational database. The input data is arranged in rows into symbol strings with different length i.e. they are inhomogeneous. Each row contains sequence of words, a part of which are keys or entities for the database and they have to be found and coded depending on their belonging to the set M_1 or M_2.

The codes of the text objects are defined in the following way. Codes-natural numbers belonging to a definite segment in advance are defined for coding the objects from M_1, P_{M21}, P_{M22} and P_{M23}. The number of the elements in the subsets P_{M21}, P_{M22} and P_{M23} is respectively p_{21}, p_{22} and p_{23}, where: $p_{21} \in 1 \div n$; $p_{22} \in n+1 \div m$; $p_{23} \in m+1 \div q$; $n, m, q \in \mathbb{N}$. Exactly one natural number (code) corresponds to each element of the sets in order to avoid ambiguity.

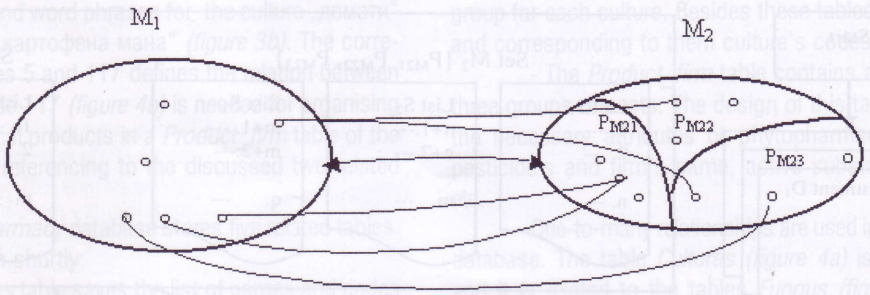


Figure 1. Presentation of the sets and relations

The names of the objects and their codes are stored in lists. Finally, the aim of this coding system is to define the relations between contextually related text objects. In all these cases a pair of their codes (combination) is formed providing that the first code belongs to the text object of M_1 while the second one corresponds to the text object of M_2. These combinations have to be saved and used as keys of table schemes of the database.

There are cases where an element code from one set and an element code from the other one are consecutively placed in a row of the in put document. The number of possible combinations between the elements of M_1 and P_{M21} or P_{M22} or P_{M23} depends on the product of number of the elements from the two sets. There is an exception when an element from set M_1 is not possibly related to all elements from P_{M21} or P_{M22} or P_{M23} of the examined row and vice versa. In this case combinations between the elements of the two sets are not correct. Therefore the user has to check and delete incorrect combinations in the dialogue mode.

3. Algorithmic Solutions

The flowchart of the algorithm for automated creation of a PC database that stores and structures related text objects is presented on *figure 2*.

Input data: Each row of the input document D_1 is interpreted as a string. The searched text objects are placed in these strings. The lists S_1, S_2 and S_3 consist of the elements of the subsets P_{M21}, P_{M22} and P_{M23} respectively and their codes The list R_1 presents the elements from M_1 and their codes. These codes are predefined on the base of expected text objects in the input document D_1. Exactly, these coded objects are the keys in the tables of the relational database.

Basic operations: The subalgorithm SA1 is applied on each row of the input document D_1. In case that a word or word phrases in a row of the input document D_1 are found as an element of the list of the sets M_1 or M_2, then the respective code from the list is inserted in the row of the document with codes D_2. The number of the received codes from the lists of the sets M_1 or M_2 in a row of document D_2 defines the number of the combinations between the elements of these two sets. The subalgorithm SA2 combines the codes and saves them in D_3. When a code from one of the two sets is duplicated SA2 automatically deletes it. The subalgorithm SA3 provides the user with the information necessary for him to establish and delete the incorrect combinations mentioned in previous section in dialogue mode. Getting the incorrect combinations depends on the text presentation of the explicit object in the input document. In these cases one has to work with an expert in the scientific field where the database is built. The data from the corrected document D_4 is an the input in the database. The number of the true combinations between the elements of M_1 and M_2 defines the number of the records stored in the tables of the relational database.

Relational database structure: The number of the database tables is determined by the number of the subsets of the set M_2 and two tables corresponding to the set M_1 and the input document D_1, that is to say q+2 tables. The table containing the objects of the set M_1 is referenced table. If an element of the set M_1 is related to many elements of the different subsets of M_2 *(figure 1)* then the elements of the subsets are stored in separated referencing tables. The table concerning the input document is related to each of the tables containing the elements of the subsets and it is the side „many" of the relationship.

The software needed for applying the suggested algorithm is implemented in the environment of Visual Basic. It also uses Excel form for input data and the DBMS „Access" for management of relational objects in PC database. The PC database updating can be accomplished by relatively small modifications of this software. The changes in the database structure are not necessary.

4. Application of the Algorithm for Automated Creation of the Phytopharmacy Database

The elements of the sets M_1 and M_2 are generally interpreted as cultures and respectively their pests. Let specify:

- The list R_1 consists of the names of the cultures from M_1 set and their codes.
- The names of all funguses from the set $P_{M21} \subset M_2$ and their codes are in the list S_1.
- All weeds from $P_{M22} \subset M_2$ and their codes make the list S_2.
- And all insects from $P_{M23} \subset M_2$ and their codes are in the list S_3.

The data referring to permitted products for plant protection and fertilizers in Bulgaria are presented in booklet [5]. „ABBYY FineReader" software system is used to transform this data [7] to Excel tables, stored in the input document D_1 of the developed algorithm *(figure 2)*. All words found in the rows of the tables from D_1 that belong to the sets M_1 and M_2 are „keys" in the database structure. *Figure 3* presents fragments of

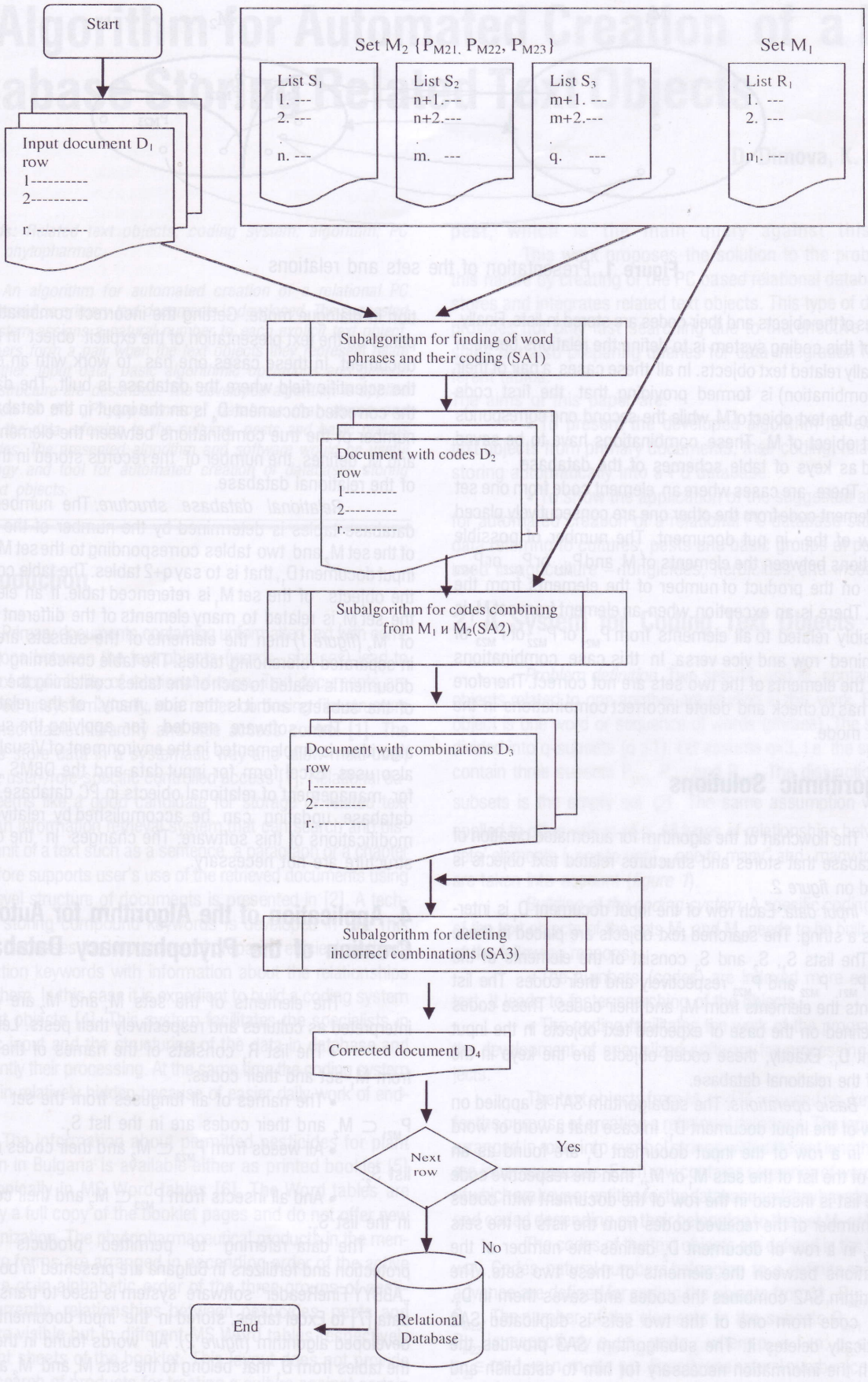


Figure 2. Flowchart of the algorithm

Excel tables with found word phrases for the culture „домати" *(figure 3a)* and pest „картофена мана" *(figure 3b)*. The corresponding pair of codes 5 and 117 defines the relation between them. The second code 117 *(figure 4b)* is needed for organising all phytopharmaceutical products in a *Product_firm* table of the database *(figure 4c)* referencing to the discussed two related objects.

The *Phytopharmacy* database stores five related tables. Let characterize them shortly:

- The *Cultures* table saves the list of names and codes of all cultures in Bulgaria.
- The tables *Fungus* (funguses), *Herba* (weeds) and *Insectum* (insects) save lists of the pests from the respective group for each culture. Besides these tables store pests' codes and corresponding to them culture's codes.
- The *Product_firm* table contains all pesticides for the three groups of pests. The design of this table includes also all the necessary attributes of phytopharmaceutical products - pesticide's and firm's name, active substance, concentration etc.

One-to-many relationships are used in the *Phytopharmacy* database. The table *Cultures (figure 4a)* is a referenced table and it is related to the tables *Fungus (figure 4b)*, *Herba* and *Insectum*. Every record of the *Cultures* table corresponds to the table of its pests from the indicated group.

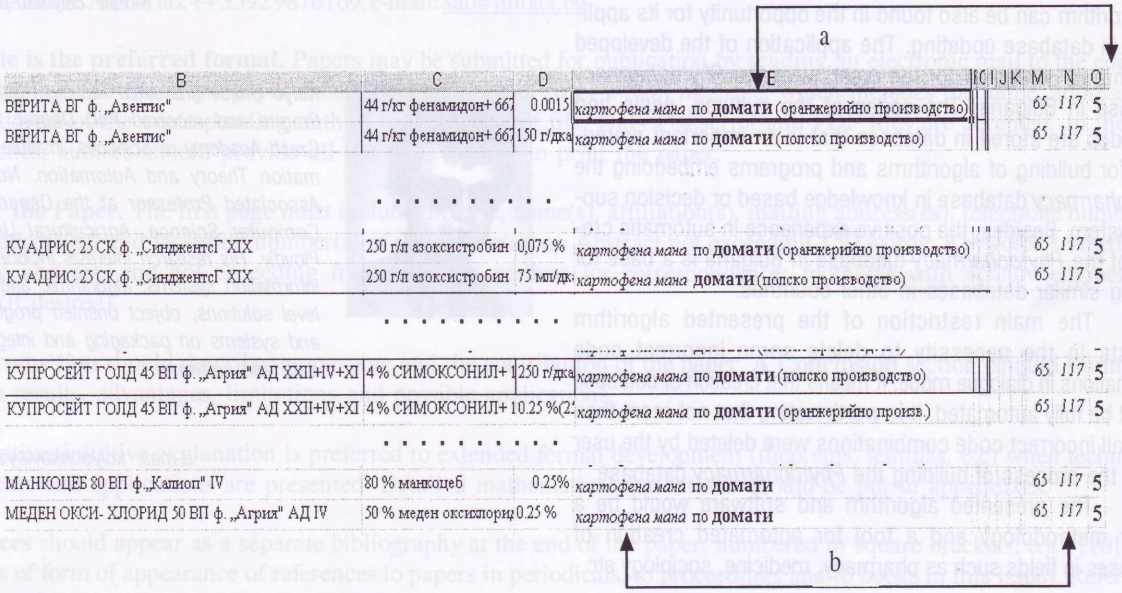


Figure 3. Fragments of Excel tables with found word phrases

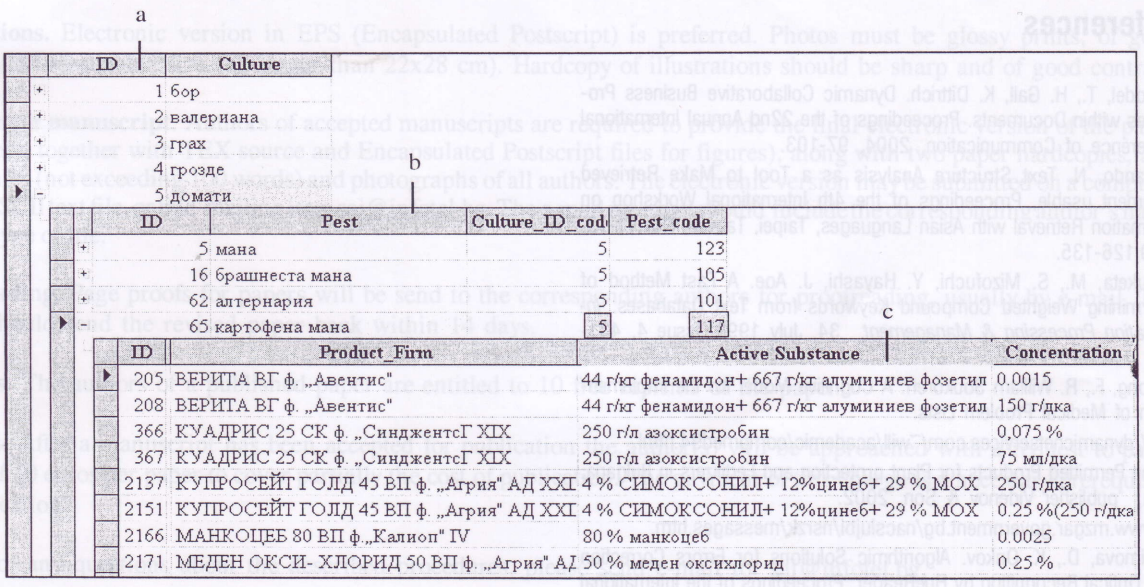


Figure 4. Relations between the objects in the „Phytopharmacy" database

Each of the three tables *Fungus, Herba* and *Insectum* is related with the respective records referring to the phytopharmaceutical products and their attributes from the *Product_firm* table *(figure 4c)*.

The storing and relating the text objects in the *Phytopharmacy* database facilitates the agricultural specialists in the fast searching and presenting needed data concerning pesticides for treating cultures against pests.

5. Conclusion

The innovation novelty of the algorithm consists in integration of the suggested coding system and required operations for automated creation of a relational PC database that stores and structures related text objects. The effectiveness of the algorithm can be also found in the opportunity for its application in database updating. The application of the developed algorithm is presented for the creation of the *Phytopharmacy* database in Bulgaria. The codes of the cultures, pests and pesticides are stored in database that is an important prerequisite for building of algorithms and programs embedding the *Phytopharmacy* database in knowledge based or decision support system. Besides, the positive experience in automatic creation of the *Phytopharmacy* database in Bulgaria is a base for building similar databases in other countries.

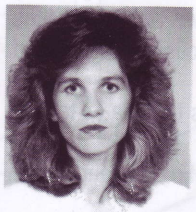
The main restriction of the presented algorithm consists in the necessity to delete some incorrect code combinations in dialogue mode. It means that creation of database cannot be fully automated. It is worth noting that not more than 4% of all incorrect code combinations were deleted by the user during the process of building the *Phytopharmacy* database.

The presented algorithm and software would be a proper methodology and a tool for automated creation of databases in fields such as pharmacy, medicine, sociology etc., where relations between text objects can be defined.

References

1. Hodel, T., H. Gall, K. Dittrich. Dynamic Collaborative Business Processes within Documents. Proceedings of the 22nd Annual International Conference of Communication, 2004, 97-103.
2. Kando, N. Text Structure Analysis as a Tool to Make Retrieved document usable, Proceedings of the 4th International Workshop on Information Retrieval with Asian Languages, Taipei, Taiwan, Nov.11-12, 1999,126-135.
3. Fuketa, M., S. Mizofuchi, Y. Hayashi, J. Aoe. A Fast Method of Determining Weighted Compound Keywords from Text Databases. *Information Processing & Management,* 34, July 1998, Issue 4, 431-442.
4. Song, F., R. William Soukoreff. A Cognitive Model for the Implementation of Medical Problem Lists.
http://dynamicnetservices.com/~will/academic/compmed94.html.
5. The Permitted Products for Plant protection and Fertilizers in Bulgaria. Sofia, publisher Videnov & Son, 2002.
6. www.mzgar.government.bg/nacslujbi/nsrzk/messages.htm.
7. Dimova, D., K. Onkov. Algorithmic Solutions for Errors Correction after Optical Recognition by FineReader. Proceedings of the International Conference „Automatics and Informatics'04", 6-8 October 2004, Sofia, Bulgaria, 251-253.

Manuscript received on 08.05.2006



Delyana Dimova *(born in 1974) was awarded Master of Science Degree in Mathematics and Informatics from the University of Plovdiv in 1997. She has been employed as an assistant in the Department of Computer Science, Statistics and Accounting at Plovdiv Agricultural University since 2001. She has been doing a Ph.D. Degree since 2005. Her research interests are in the field of databases, information systems and algorithmic solutions.*

Contacts:
e-mail: dely@au-plovdiv.bg

Kolyo Onkov *graduated Technical University in Prague and received PhD Degree from the Czech Academy of Sciences, Institute of Information Theory and Automation. Now he is Associated Professor at the Department of Computer Science, Agricultural University, Plovdiv. His research interests include building information systems, algorithmic and system-level solutions, object oriented programming, and systems on packaging and integration of data and software.*

Contacts:
e-mail: kolyoonkov@yahoo.com