Effective Results Analysis for the Similar Software Products' Orthogonality

Ion Ivan
The Bucharest Academy of Economic Studies
ionivan@ase.ro
Daniel Milodin
The Bucharest Academy of Economic Studies
daniel.milodin@ase.ro

Abstract. *It is defined the concept of similar software. There are established conditions of archiving the software components. It is carried out the orthogonality evaluation and the correlation between the orthogonality and the complexity of the homogenous software components is analyzed. Shall proceed to build groups of similar software products, belonging to the orthogonality intervals. There are presented in graphical form the results of the analysis. There are detailed aspects of the functioning of the software product allocated for the orthogonality.*

Keywords: uniform software product; orthogonality; software complexity.

JEL Codes: C88.
REL Codes: 9A.

1. Similar software components

At present, as a result of the existing applications to solve various theoretical and practical problems, there is a wide variety of software. There are defined criteria and build multitudes of software products that meet these criteria.

For the programming language criterion one defines the multitude of programs written in C++, the multitude of programs written in C#, the multitude of programs written in Java.

For the criterion of working with databases there are identified applications that use relational databases, object oriented database and classic databases.

For the criterion of the operating system the software components are determined that deal with the organization and the management of a computer, respectively MS-DOS, Linux, Unix, Windows, Mac OS, Solaris.

For the criterion of programs allocated to protect the computers from some codes' malicious action to affect its proper functioning, there are identified the following components: BitDefender, Norton Antivirus, Kaspersky Antivirus, McAfee, avast!, Eset Nod32.

Let us consider the criteria C_1, C_2, ..., C_m, and the P multitude of programs.

If the C_i criterion is applied the P programs are divided into sub-multitude P_{i1}, P_{i2}, ..., P_{ih}.

If for the P_{ij} sub-multitude, obtained by applying the C_i criterion, the C_h criterion is applied the sub-multitude P_{ij1}, P_{ij2}, ..., P_{ijh} will result.

It is noticed that the criteria application is done step by step either on the P multitude line, or one after another on sub-multitudes (on top down criteria).

Further, by applying a different criterion C_s, it is obtained a new sub-multitude, composed by the elements of the old sub-multitude, but which respects the new criterion.

The general condition that must be respected for the application of new criteria is that the main multitude to have a large number of components, and the resulted sub-multitude to be large enough to allow the application of new criteria.

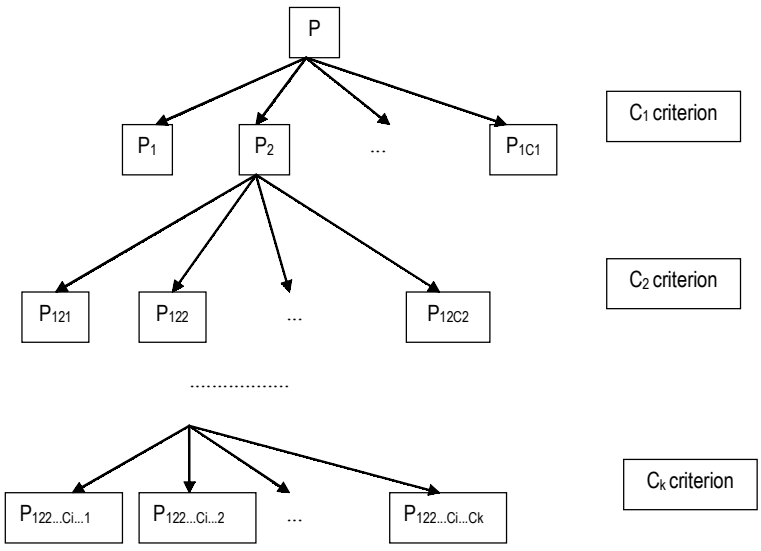

Figure 1. *Criteria implementation on multitude*

After applying the top down criteria, the obtained sub-multitude has achieved a level of similarity increasingly high.

It is considered the multitude of students who sign up to take the admission to the Faculty of Cybernetics, Statistics and Economic Informatics. A first criterion, C_1, which applies to this lot is that they must be high school graduates. Another criterion is that of mathematics knowledge. Since the admission has two test papers of mathematics, algebra and mathematical analysis, the students must be well prepared for this subject.

It is considered the multitude of students from the 3rd year – Economic Informatics specialization.

A first criterion to be applied to this multitude is given by the level of the students' preparedness in terms of programming languages.

In the 3rd year, the degree of similarity of the multitude of students from the specialization Economic Informatics is very high because:

- they have the same age;
- they took the same exams;
- they have the same level of experience;
- they have attended the same courses;
- the obtained marks are top level, because the selection for the Economic Informatics section is made on the average criterion.

It is required the construction of similar classes of software products, because the high quality derives from:

- qualitative members aggregation;
- further easy development;
- keeping the requirements imposed by the realization process;
- ensuring comparability, as human and financial effort;
- identifying the bad components, as in the case of distributed applications, regardless of their inferior quality processes, this reflects on dozens or thousands of users.

Citizen-oriented applications that are specific to knowledge-based economies must meet strict requirements of similarity, such as:

- technologies used for elaboration;
- style of elaboration;
- high quality level;
- high degree of compatibility.

Therefore it is proposed to conduct an analysis of similar software products, as citizen-oriented applications must be considered under this criterion.

2. Lots of applications

It is formulated a set of specifications for:

- problems to be solved (each stated problem to be solved by implementing the software products);
- used resources (programming language used);
- the characteristics of quality (reliability, simultaneity, orthogonality) and content (the source lines number, occupied space on the hard disk by the source file, contained key words).

The formulated specifications receive individual solutions from the community members.

Developers form a similar multitude, characterized by value, experience, knowledge level, age level.

It is considered the D_1, D_2, ..., D_n developers multitude, characterized by:

- close level of preparation from one user to another;
- level of knowledge which is not different from one user to another, fact proved by the covered qualifying stages;
- same level of experience in solving the mentioned problems;
- the compulsoriness to respect the deadline for formulating the solutions;
- the common source of processing the specifications for building solutions;
- the unique way of transmitting the generated solutions for evaluation.

In this context it is defined a first strict filter that ensures the construction of similar solutions:

- vocabulary based on a thesaurus and keywords from the C++ language;
- the length obtained by respecting the restriction of efficiency in programming;
- complexity defined through restrictions regarding the C++ procedures construction;
- correctness given by the compulsoriness of checking the C++ operations functioning;
- completeness achieved by the mandatory inclusion of a set of procedures, the orthogonality being the assessment objective.

There are considered the S_1, S_2, ..., S_6 specifications.

Based on these specifications homogenous lots of programs are constructed. Getting the lots of programs from the users is done using the ORTOES application.

The multitude of developers is composed of a number of 118 members. Of these, only 114 met the conditions imposed for loading the personal solutions for listed specifications.

Table 1 shows the users' arrangement according to the six types of problems that must be solved:

Table 1

The users' arrangements according to the specifications of the problems that must be solved

Specification	No. of the members who loaded solution	No. of the members who loaded no solution
S_1	96	18
S_2	96	18
S_3	93	21
S_4	91	23
S_5	81	33
S_6	103	11

As it results from Table 1, for any specification no solutions were submitted by all members of the community, the smallest value of the number of students who have introduced a theme is 81, according to S_5 specification and the highest value for the number of solutions corresponding to a specification is 103, corresponding to the S_6 specification.

3. The homogenous applications orthogonality

Orthogonality studies the degree of similarity between two or more entities. Through this quality characteristics it is determined the extent to which the entities differ from each other.

To study orthogonality it is defined an indicator of orthogonality contained within the interval [0, 1], which takes the following values:

- 1, if the elements are orthogonal, i.e. they have nothing in common;
- 0, the elements are identical, i.e. they don't have different values for any feature.

If the indicator tends to 1 it means that the provided data sets tend to orthogonality, and if the indicator's value is close to 0 it means that the data sets have many identical elements.

The orthogonality of the solutions is determined as internal orthogonality and external orthogonality. Internal orthogonality sets the measure in which, within the solution, the used words repeat, the existent links between words within it.

The external orthogonality shows the differences or similarities that exist between texts.

The orthogonality of the provided solutions is studied on the basis of the following criteria:

- the size of file that stores the solutions, expressed as the number of needed bits;
- the frequencies of occurrence of words within the solutions, calculated by identifying the number of words occurrences within them;
- the frequencies of occurrence of linking words, calculated by identifying the number of occurrences of the linking words within the solutions;
- the size of solutions considered as the number of words or number of letters constituting solutions;
- the words arrangement within the solutions, determining the position of each word and identifying the similarities;
- the distances between words defined as the number of words which interpose between two considered words;
- the number of sections for the solutions.

Two provided solutions are orthogonal if they do not have common words.

The orthogonality of two texts is calculated by taking stock of common words.

The orthogonality study is given by:

- the need to establish the level of similarity between two solutions;
- the need to identify the identical solutions;
- establishing the source for the given solutions, depending on the similarity level existing between them;
- the need to determine each user's contribution to generate original solutions.

The orthogonality is defined depending on the solution's component which is studied:

- orthogonality throughout the all solution identifies the level of similarity between the two provided solutions, based on the common words, the procedures of their disposal and the belonging of the solutions to a studied field;
- orthogonality on the fingerprints identifies the level of similarity between certain parts of the solutions, comparatively with portions from the same solutions or from different solutions;
- the orthogonality through vocabulary calculates the similarity between two or more solutions based on the frequencies of occurrence within the solutions of the words from the vocabularies devoted to the studied field.

There are situations in which on a T solution are performed a series of transformations, achieving a T' solution by:

- replacing the words with synonyms;
- establishing pairs of words (a_i-, b_i) and replacing the word a_i wherever appearing with the b_i pair to create the impression of originality;
- constructing paragraphs identical in meaning (p_i, q_i), so that the p_i paragraph will be replaced by the q_i paragraph; the instruction *for* is replaced with the instruction *while*, the instruction *if... else* is replaced by two conditions *if*, the metering formula is being replaced by i = i + 1 with i++, the procedure type is replaced with the function type that returns certain type of data.

It is reasonable that for the orthogonality analysis to be considered the similar aspects between terms and instructions, to identify the relationship of transformation between two solutions.

When T' is transformed by an orthogonality application results T", in this situation the orthogonality between T and T" is being analyzed, and depending on the orthogonality level it is verified the fact that T' is the product of a transformation applied to T. To establish the orthogonality of two source files, S_1 and S_2, it is used the indicator:

$$H(S_1, S_2) = 1 - \frac{NCC}{\max\{LG(S_1), LG(S_2)\}}$$

where:

NCC – the number of common words;

LG() – the solution's length, represented as a number of component words.

The members of the community must meet certain deadlines imposed for loading solutions to the mentioned problems. After overcoming the limits, the application no longer allows access for the users. After running the module for establishing the level of orthogonality it is determined the orthogonality of each solution provided by the members of the community. In Table 2 is presented a centralized situation of the results:

Table 2

The weight of the orthogonality level on intervals

Problem specification	Orthogonality within the interval [0;0.75)	Orthogonality within the interval [0.75;0.85)	Orthogonality within the interval [0.85;1.00]
S_1	0	3	93
S_2	0	0	96
S_3	0	0	93
S_4	0	6	85
S_5	0	5	76
S_6	0	0	103

It is observed a high level of the orthogonality, fact based on:

- the enouncement of problems with multiple solutions solving;
- a low level re-using modules to solve the problems;
- the use of a wide range of variables;
- the use of various data structures for the application's implementation;
- the building of instructions by varying the orders of language.

Table 3 shows the minimum and maximum values of the orthogonality indicator corresponding to the six specifications:

Table 3

Minimum and maximum values of the orthogonality

Problem specification	Maximum value	Minimum value
S_1	0.996	0.799
S_2	0.996	0.861
S_3	0.996	0.906
S_4	0.996	0.822
S_5	0.995	0.81
S_6	1	0.799

From Table 3 results that only for the 6th specification was obtained a high level of the orthogonality, value obtained because the solutions corresponding to these specifications are more elaborated, their complexity being high.

4. ORTOES application

It is implemented to allow to a number of users to establish the orthogonality of the structured entities that are generated.

The ORTOES application is built for automation of the interaction with users. In the first phase, the application's manager loads the users that have access to the application's functions. Based on these records the users create their accounts and passwords for access, load the solutions, envision the orthogonality level.

A first problem about the application's management was the loss of passwords or accounts for access by the users. The problem was solved by querying the database, finding the user account in question and generating a new password.

Passwords are protected by encryption, so that the application's administrator can not view password's text, the only solution in this case being the passwords rewriting.

Application's functions:

- building the list of users;
- building access data allocated to each user (username and password);
- taking-over the solutions generated by each user for each specification;
- measuring the level of orthogonality on categories of solutions built by users:
- $H(T_{ki},T_{kj})$, where k is the category of solution proposed for the analysis, $k = 1,2,3$, and i and j represent the users whose solutions are analyzed;
- determining the aggregate orthogonality corresponding to the categories of processed solutions;
- determining the medium orthogonality for each user on the basis of the aggregated orthogonality calculated for each category of solutions;
- displaying the partial, final, individual and general results;
- re-introducing solutions generated with the detention of the last generated solutions, because the program product aims to generate the increasing of the orthogonality generated over a period of time, has the character to assist the process of generating solutions.

Application's administration includes:

- introducing the list of users;
- introducing the accounts and passwords for the users list, selecting each user's account and password through which they access the application;
- allocating the unique numeric codes for each user; the allocation is made when user's data are inserted and not when he creates his account;
- retrieving txt files containing solutions provided;
- launching the implementation of procedures which analyze the orthogonality of the provided solutions;
- displaying the introduced solutions and their orthogonality level so that users whose solution does not meet the criterion of orthogonality to reintroduce their solutions modified;
- displaying the conditions to be followed to load txt files containing the solutions provided;

- managing the different versions of the application for intervals of time when the user accesses the application in order to send generated solutions;
- managing the messages sent by users, and display them either individually or in the form of lists containing the final results;
- managing the files placed by users and prepare them to be accessed by the module which examines the solutions' orthogonality; the files are renamed and given the code name assigned to the user who uploaded them;
- storing the results concerning the solutions' orthogonality and ensuring their matching with the module it shows;
- storing the information about the solutions that have the orthogonality level lower than the threshold;
- generating the results relating the average and aggregated orthogonality, results provided for each user or for all;
- managing the user's accounts;
- generating new passwords in a situation when this is desirable;
- deleting a user account;
- identifying and viewing the solution provided by users.

Administration performed by users includes:

- defining the username and the password;
- uploading the files with solutions provided;
- viewing the individual and the final results.

For the solutions collected from community members through direct input from the keyboard, orthogonality is calculated depending on when they are loaded using the ORTOES application: first user who loads the solution will have the maximum orthogonality level because there are no similar solutions for comparison. Solutions which are loaded later will be reported to the solutions already taken, fact based on evidence and analysis of results, arranged according to the time of loading.

For the solutions generated by users verification is performed after the deadline is exceeded for loading txt files containing solutions, by running a C+ + module which incorporates txt source files and analyzes them individually.

To ensure the security of the on-line application which offers the facility of establishing the orthogonality of the solutions generated by the users, it was resorted to the facilities offered implicitly by the used MySQL data basis (the creation of users with unique credentials, stocking their passwords in an encrypted format) and, on the other hand, by using the work sessions.

For each user that opens online the application it begins an individual work session, this thing offering protection for the attempts of accessing unauthorized the application's resources. So, the work session is closed when the user quits the currents account or when more than 5 minutes passed without the user to use the application for his personal project.

Another implemented method for ensuring the application's security is represented by the *mysql_real_escape_string (string unescaped_string[, resource link_identifier]*) function. It is used to ensure the data before sending them as a request towards the MySQL server, through eliminating the characters potential dangerous from the row.

The elimination is made by adding backslashes (\x00, \n, \r, \, ‘, “ and \x1a), thing that forces the MySQL to interpret the characters rather as characters with a single apostrophe than as part of the declaration SQL.

5. Correlation between the complexity of solutions provided and their orthogonality level

The complexity of the solutions provided by community members implies details involving the size of solutions. The size of solutions provided is expressed as the number of words contained in each txt file source and size in bytes occupied on the drive file.

It starts from the premise of multi-criteria dependency between the orthogonality of the provided solutions, the number of words and the size of solutions, independent features, which stressed the constructed graphs.

To establish dependencies one uses statistical concepts, like:

- correlation, used to determine the degree in which the outcome variable is dependent on the independent variables;
- regression, used to determined how the variables influence each other.

From Table 4 results that the independent variables are strongly correlated, and the introduction of both variables in the model of dependence leads to its invalidation.

Table 4

Correlation between variables

	HT1	CUV1	DIM1	HT2	CUV2	DIM2	HT3	CUV3	DIM3	HT4	CUV4	DIM4	HT5	CUV5	DIM5	HP	CUVP	DIMP
HT1	1.00																	
CUV1	0.48	1.00																
DIM1	0.44	0.96	1.00															
HT2	1.00	0.47	0.43	1.00														
CUV2	0.41	0.66	0.70	0.41	1.00													
DIM2	0.35	0.65	0.72	0.35	0.98	1.00												
HT3	0.91	0.42	0.39	0.91	0.41	0.36	1.00											
CUV3	0.49	0.47	0.43	0.49	0.70	0.65	0.53	1.00										
DIM3	0.48	0.44	0.41	0.47	0.69	0.66	0.52	0.99	1.00									
HT4	0.85	0.45	0.41	0.85	0.39	0.34	0.88	0.50	0.49	1.00								
CUV4	0.34	0.32	0.30	0.33	0.55	0.48	0.36	0.55	0.53	0.41	1.00							
DIM4	0.32	0.31	0.30	0.31	0.54	0.47	0.34	0.53	0.52	0.39	1.00	1.00						
HT5	0.67	0.35	0.33	0.67	0.30	0.26	0.69	0.35	0.33	0.78	0.27	0.25	1.00					
CUV5	0.35	0.47	0.42	0.35	0.55	0.50	0.38	0.64	0.62	0.42	0.38	0.36	0.52	1.00				
DIM5	0.35	0.49	0.45	0.34	0.56	0.52	0.37	0.65	0.64	0.41	0.38	0.36	0.52	0.99	1.00			
HP	0.16	0.09	0.09	0.17	0.05	0.04	0.21	0.05	0.04	0.17	0.06	0.07	0.12	-0.06	-0.07	1.00		
CUVP	0.06	0.11	0.20	0.06	0.23	0.24	0.10	0.05	0.06	0.08	0.01	0.02	0.02	-0.04	-0.03	0.33	1.00	
DIMP	0.09	0.18	0.27	0.10	0.27	0.30	0.13	0.10	0.10	0.11	0.03	0.04	0.03	-0.05	-0.03	0.34	0.96	1.00

Taking into account the results concerning the dependence between the orthogonality level corresponding to the 5 provided specifications and the project, it has been concluded that between the orthogonality level and the number of words of each solution provided there is a stronger connection than between the orthogonality level and the size of the source file and are presented the results of the dependence analysis.

Specifying an econometric model involves the choice of using mathematical functions which can describe the relationship

between the two variables. For single factor regression model, the method most often used is the graphical representation with scatter plot.

Figure 2 shows that between the orthogonality of first themes and the source file size expressed as number of words is a linear relation, single factor regression (Y = a×x + b) directly proportional.

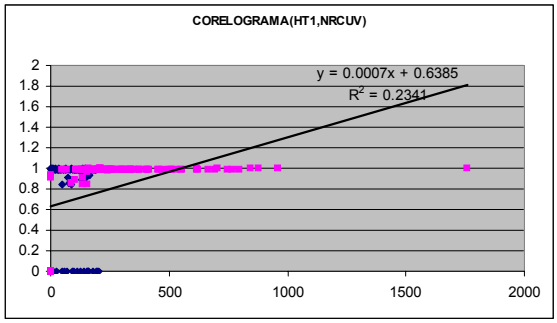

Figure 2. *Scatter plot orthogonality – words number for S_1 specification*

Using Excel it is build the linear regression model of addiction, for a total of 112 cases with a 95% probability and 110 degrees of freedom resulting model HT_1 = 0.0007 × NrCuvinte + 0.638. Parameter *a* has a positive sign and confirms our hypothesis that the link between the two variables is directly proportional.

The model shows that the variation of orthogonality is determined by the number of words in the proportion of 23.40%, the rest being affected by other factors, such as words belonging to a domain, the uniqueness of the words. Also, based on the tests performed using Excel results that the model and the regression parameters are valid.

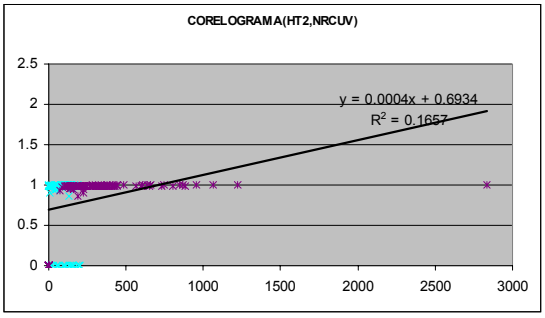

Figure 3. *Scatter plot orthogonality – words number for S_2 specification*

Figure 3 shows that for S_2 specifications there is a direct link between the level of orthogonality and the number of words used to solve the specification, the regression model being HT_2 = 0.0004 × NrCuvinte + 0.693, the level of orthogonality being explained in the proportion of 16.57% by the number of file words.

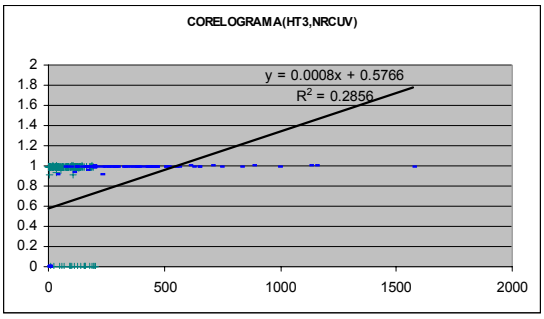

Figure 4. *Scatter plot orthogonality – words number for S_3 specification*

For S_3 specifications, dependence is linear, the regression model being HT_3 = 0.0008 × NrCuvinte + 0.576, for this case orthogonality is explained in proportion of 28.55% by the variation of the number of words as shown in Figure 4.

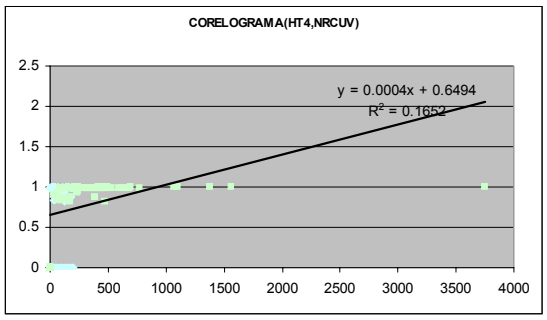

Figure 5. *Scatter plot orthogonality – words number for S_4 specification*

The link existing for S_4 specifications is shown in Figure 5, this link is a single factor regression, the regression model for this case is $HT_4 = 0.0004 \times NrCuvinte + 0.649$, variation of orthogonality is explained in proportion of 16.5% by the number of words variation.

Solving S_5 specifications led to a linear single factor regression model, represented in Figure 6, the regression model being $HT_5 = 0.0007 \times NrCuvinte + 0.503$, the variation of orthogonality is explained in the proportion of 27.47% by the variation of the number of words.

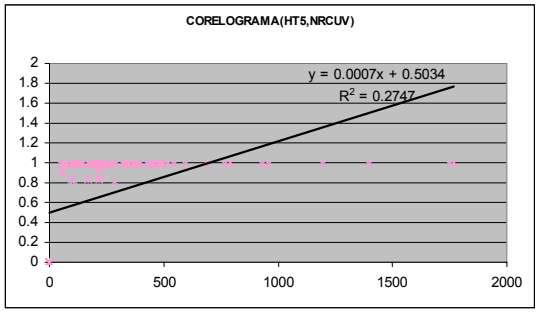

Figure 6. *Scatter plot orthogonality – words number for S_5 specification*

The S_6 specifications are designed to allow a larger approach from the community members, the size of files expressed as the number of words being larger.

In Figure 7 is represented the relation between the orthogonality of solutions according to S_6 specifications and the number of words used:

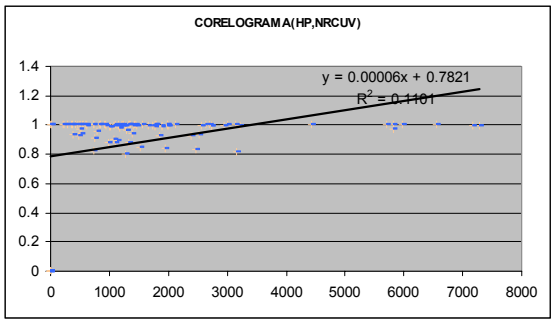

Figure 7. *Scatter plot orthogonality – words number for S_6 specification*

The relationship between variables is a linear single factor regression model, HT6 = 0.00006 × NrCuvinte + 0.782, the orthogonality variation being explained in proportion of 11% by the variation of the number of words.

For all presented models their validity and the regression parameters are tested with the Ficher test and Student test, models being constructed for a number of 112 cases with a 95% probability and 110 freedom degrees.

6. Conclusions

Defining the top down criteria for determining the sub-multitudes leads to a high degree of simi'larity of the sub-multitude's

components. Top down criteria should be defined so as to provide a filtrated resulting sub-multitude which contains components that strictly meet the criteria.

By building some similar components there are respected the quality criteria required by the citizen oriented applications. The use of similar applications in working with people helps to increase work efficiency and a higher data processing.

The ORTOES application ensures the source texts of the applications processing in order to determine their orthogonality degree. Building applications with a high orthogonality level and imposing certain restrictions with regard to their domain affiliation, same as the conformation to precise specifications, imposed to develop informatics' applications contributes to increase the applications homogenous degree, to make a better use of them by citizens.

Based on the built models it is observed that only approximately 20% of the orthogonality's value is influenced by the number of words. Turns out that except from the number of words there are other factors with powerful impact on the orthogonality level: originality, uniqueness of the used words, the newness which the defined concepts involve.

For the available data sets the linear models have a low relevance. The proposed technique represents an opening for the use of other classes of models and aims to establish the proportion in which the number of words influences the orthogonality level of a source file.

The application has an open character, allowing the introduction of new models and increasing the number of factors needed for the orthogonality's study, by introducing variables such as the distance between words belonging to the same class and the distance between identical words.

References

Feng-Jen, Yang, „The orthogonality in C++", *Journal of Computing Sciences in Colleges*, Volume 23, Issue 6, 2008, pp. 213-219

Ivan, I., Boja, C., Milodin, D., „Orthogonality level optimization for distributed informatics applications", 4th International Conference on Applied Statistics, Bucharest, November, 20 – 22, 2008, Bucharest University of Economics, National Institute of Statistics, *Special Issue of Romanian Statistical Review* (CNCSIS B+), ISBN 1018-046x

Ivan, I., Dumitrascu, E., Palaghita, D., Milodin, D., Popa, M., "Optimum criteria for developing defined structures", *Economic Informatics*, vol. 12, nr. 2, 2008, pp. 43-54

Ivan, I., Milodin, D., Dumitru, S., Palaghita, D., „C# Entities Quality Analysis", *Journal of Applied Quantitative Methods*, vol. 3, nr. 1, martie 2008, pp. 20-31

Ivan, I., Milodin, D., Popa, M., „Operation on text entities", *Economic Informatics*, vol. 11, nr. 1, 2007, pp. 14-20

Ivan, I., Surcel, T., Milodin, D., „Turism management information system based on mobile tehnology", *The 2009 international conference on tourism and workshop on "Sustainable tourism within High Risk areas of environmental crisis"*, 22-25 April 2009, Messina, Italy

Linoff, G., Berry, M. (2002). *Mining the Web: Transforming Customer Data*, John Wiley, New York

Milodin, D., Dumitru, S., „The security of the application for evaluating the text entities' orthogonality", *International Conference on Informatics*, 9th, Bucharest, 2009