

THE STUDY OF CURRICULAR DEPENDENCY

Dragos ANASTASIU¹

Academy of Economic Studies, Bucharest, Romania
Student of "Informatics Security" Master Program

Published articles: Structured Entities Structures Building (co-author), Workshop Information Systems & Operations Systems Proceedings, 1 - 2 March 2006

E-mail: dragos.anastasiu@gmail.com



Abstract: *In the educational process, the disciplines D_1, D_2, \dots, D_n have a succession which is generated by the content and the final objective – the student formation. In this work, the disciplines are presented as structured text entities. The graph associated to the disciplines is established. A method for dependencies evaluation is proposed. The testing is done with 9 sets of representative input data.*

Key words: *Text entity, Vocabulary, Graph, Data Structure, Dependency, Independency, Graph Drawing, Curricular, Metrics, Documentation, Progression, Consistency, Uniformity, Proportionality, Intraorthogonality, Interorthogonality, Software, Statistical, Inclusion*

1. Structured text entities

Text entities are used for storing and organizing texts representing very diversified information. The naming of entity indicates the generality of the concept, as the text represents information, which is able to be structured according to the origin and the scope which is associated to this.

The following base concepts regarding text entities are defined:

- **the alphabet** A is a finite multitude formed of N symbols: a_1, a_2, \dots, a_N ;
- **the separator** is a symbol which does not belong to the alphabet A , having the role of delimitating two words that form a words sequence;
- **the word** is a succession formed out of symbols which follow one another. A word c_i is characterized by its length, $lg(c_i)$ expressed as a number of characters that participate to the formation of word;
- **the vocabulary** V_A is a multitude of different words. The length of the vocabulary V_A , noted as $Lgv(V_A)$ indicates the number of words which participate to the formation of the vocabulary;
- **the text vocabulary** is constructed of a multitude of different words which appear in the text. The text vocabulary, V_T , is included in the vocabulary V_A . Sometimes, V_T is identical with V_A ;
- **the frequency of apparition** for the word c_i , noted as f_i shows the number of apparitions for the word c_i in the text T . The frequency of apparition for the symbol a_i in the text T shows the number of apparitions for this symbol and it is noted as g_i ;

- **the sub-vocabulary** is a part of the vocabulary constructed in such way that the intersection of any sub-vocabulary pair leads to void multitude of elements. Sub-vocabularies are disjoint multitudes of elements;
- **the text T** is a succession of words from vocabulary V_A separated by special symbols which are called separators. The text length $Lgt(T)$ states the number of words which form the text. The text length $Lgts(T)$ states the number of symbols which goes into the formation of the text T.

The entities which are based on texts are actually constructions formed out of word sequences characterized by the positions of words within the text, by the grouping of words for the purpose of defining a context, by the making of a correspondence between the words and the elements, actions and real world elements, with qualitative attributes that groups concrete aspects from the reality in homogenous collectivities taking in consideration predefined criterion.

In [IVAN05]², there are presented elements which must taken in consideration for constructing a text entity:

- clear delimitation of the tackled field;
- the defining of key words for the field; for a text entity developed in a field, there must be identified those words that describe it in the most synthetic way;
- the usage of the vocabulary in which the key words vocabulary is also included;
- the knowledge of the concepts, techniques, methods, methodologies, technologies which are specific for the field;
- documentation regarding the detail elements and those connected to other fields of activity;
- following the rules of the syntax for each language;
- following the rules regarding text entities structure, gradual tackle of the problem, usage of standard formats for representation of text typed information.

Some representative examples of text entities are: scientific, literal, cultural words, web pages which are found on the Internet, the source code of the software products, dictionaries, phone books and any other grouping entity that exists as a text or is able to be structured as a text.

A particular case of text entity is represented by the list of disciplines within the learning system. A discipline is a text entity, formed out of a multitude of concepts that belong to it along with the definitions, demonstrations and corresponding examples.

Each text entity E_T is formed out of NT components noted as $SET_1, SET_2, \dots, SET_{NT}$ and each component has its own vocabulary, which represents a sub-vocabulary for the parent entity. The text entities structures are established based on the relations between entities, relations that are formed taking into consideration the connections between the component vocabularies, meaning the entities sub-vocabularies. In this way, for describing the entities structures and also the connections which are formed between these ones, it is necessary to analyze the base level, of words and words vocabulary.

Considering the vocabularies V_1, V_2, \dots, V_{NV} , with $V_i \cap V_j = \phi$, $\forall i, j \in \{1, 2, \dots, NV\}$ and $i \neq j$, it results that between the announced vocabularies a **independency relation** is established, meaning that the concepts have nothing in common. The graphic for this type of relation is presented in figure 1.

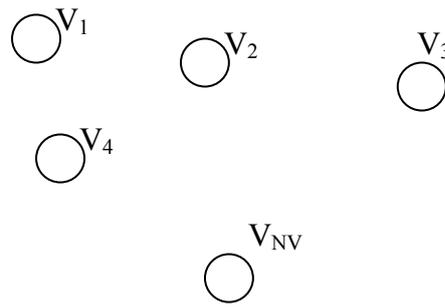


Figure 1. Independence relation between vocabularies

Linear dependency relations between vocabularies, as opposed to independency relations presented previously, indicate the presence of some connections between vocabularies. These connections are condensed in two categories:

- **full dependency**, when the vocabularies are fully included one in the other: $V_1 \subset V_2 \subset V_3 \subset \dots \subset V_{NV}$, case in which the concepts contained in the vocabularies with lower index are assumed and further extended in vocabularies with higher index. Such type of relation between vocabularies is presented in figure 2;

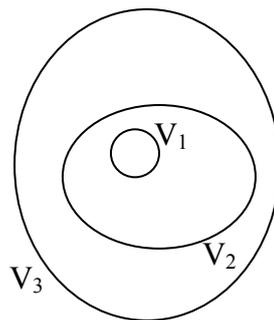


Figure 2. Full dependency relation between vocabularies

- **partial dependency** which forms when the vocabularies are not totally included one in the other, but some concepts that are assumed exist so that $V_1 \cap V_2 \neq \emptyset$, $V_2 \cap V_3 \neq \emptyset$, $V_1 \neq V_3$, $V_1 \not\subset V_2$ and $V_3 \not\subset V_2$. The partial dependency relation between vocabularies is presented in figure 3.

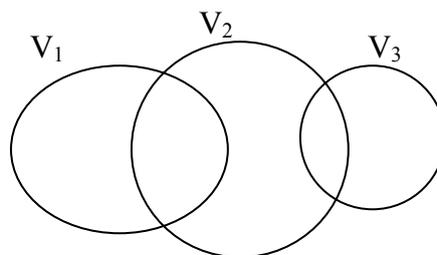


Fig. 3. Partial dependency relation between vocabularies

Other types of relations form by combining the already defined ones, so that in figure 4, a vocabulary which holds in its composition two independent vocabularies is presented. It is the case in which $V_2 \subset V_1$ and $V_3 \subset V_1$ and $V_2 \cap V_3 = \emptyset$. The graphic representation is presented in figure 4.

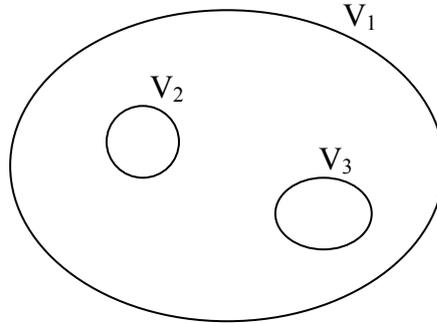


Figure 4. Combination of dependency and independency relations between vocabularies

When between two partial dependent vocabularies, the connection section between them is fully dependent of other two vocabularies which are partially dependent between themselves, $V_1 \cap V_2 \neq \emptyset$, $V_3, V_4 \subset V_{INT12}$, $V_3 \cap V_4 \neq \emptyset$, $V_3 \not\subset V_4, V_4 \not\subset V_3$ and the resulted structure is graphically displayed in figure 5.

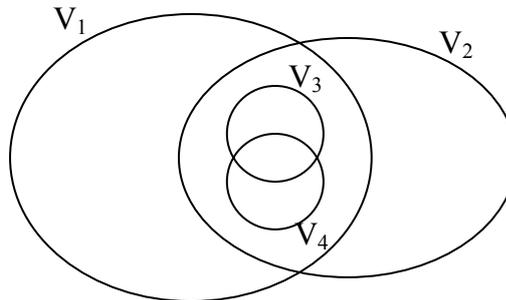


Figure 5. Combination of partial and full dependency between entities which are partial dependent and have another partial dependent relation between entities in the connection section

For a practical implementation of the presented concepts, three text entities are considered: the first one represented by the current work and the two others extracted from the bibliography included in it, [IANA06] and [IVAN05]. The graphical display of the structure that forms is presented in figure 6.

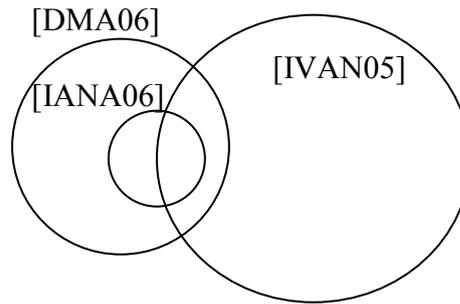


Figure 6. Structure showing the relations between the current work and two works papers included in its bibliography

According to the structure, the current work [DMA06], completely includes the work [IANA06], meaning that it assumes all the concepts which are further tackled and extended, but it also includes a part of the work [IVAN05], part that is also divided in two categories:

- the first category is the one included in the current work, and which also exists in both [IANA06] and [IVAN05];
- the second category is the one included in the current work [DMA06] just from [IVAN05], without being presented the concepts in [IANA06].

2. The graph associated to the entities

A dependency graph is a graph whose nodes are represented by different types of entities among which there are distinguished some dependency relation by using arcs. The precedence is a dependency which is transposed to the time line. Thus, some operations which take place have precedence while the concepts within a learning domain depend one on another in such way so that they must be preceded in the approach.

The graph node represents the text entity, or a component of it having the corresponding vocabulary. The precedence is established either by using directional arcs, when the resulted graph is directional or by using simple arcs with priority decreasing from left to right and from up to down when the graph is not directional.

For the structures presented in chapter 1, the dependency graphs associated are presented:

- for the structure presented in figure 1, the associated graph is the one from the figure 7;

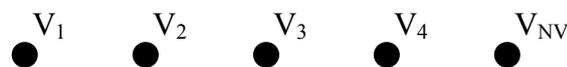


Figure 7. Graph associated to the structure presented in figure 1

- for the structure that has the diagram in figure 2, the graph from the figure 8 results;

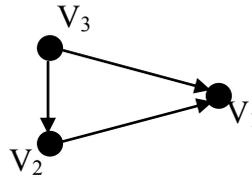


Figure 8. Graph associated to the structure presented in figure 2

- in figure 9, it is presented the graph associated to the structure presented in figure 3;

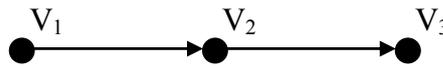


Figure 9. Graph associated to the structure presented in figure 3

- the graph for the structure in figure 4 is the one drawn in figure 10. V_1 is dependent both on V_2 and V_3 , which means that in V_1 all the concepts from V_2 and V_3 are assumed and extended;

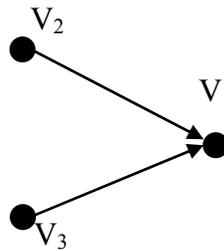


Figure 10. Graph associated to the structure presented in figure 4

- the dependency graphs for the structure in figure 5, with the elimination of the intersections between arcs, and without the elimination of these are presented in figure 11 and 12 respectively. V_3 and V_4 are partially dependent and they represent the base for entities V_1 and V_2 being completely included in their common section.

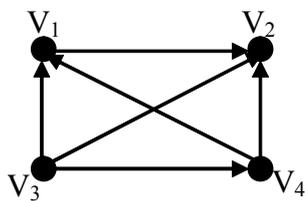


Fig. 11. Graph associated to the structure presented in figure 5

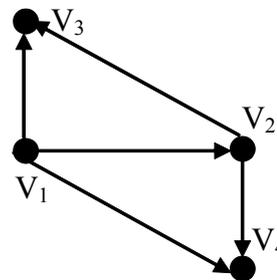


Fig. 12. The version without the arcs intersections for the graph presented in figure 11

- for the structure presented in figure 6, the constructed dependency graph is the one presented in figure 13..

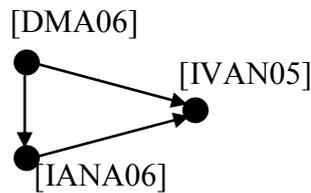


Figure 13. Graph associated to the structure presented in figure 6

The dependency graph is the same with the one in figure 8, this fact showing that even if the dependency is represented, the quantity of elements in connection does not have a specific meaning within the drawing.

3. Associated metrics

In [IVAN05] a metric is defined as being a mathematical model with the following form:

$$y = f(x_1, x_2, \dots, x_{nfi})$$

where:

- y is a model which depends on the values x_1, x_2, \dots, x_{nfi} for the factors $Ft_1, Ft_2, \dots, Ft_{nfi}$;
- x_i is the numerical value for the influential factor Ft_i ;
- Ft_i represents the influential factor I from the multitude of factors that determine the variable which shows the result y .

Through the mathematical model, the quantification of the characteristics for the analyzed entity is obtained.

The work [BOJIO04] mentions that the metrics have the following functions:

- measuring – values for the elements from the text entity structure are distinguished;
- comparing – the resemblances and differences between two or many analyzed entities for classification or hierarchically categorizing are pointed out;
- analysis – has the role of distinguishing the quality characteristics of the analyzed entities;
- synthesis – consists in extraction of what is essential for an analyzed text entity collectively;
- estimation – future evolutions of the behaviour for the analyzed text entity are established;
- verification – implies the validation of mathematical models associated to the metrics.

For measuring purposes, quality characteristics for text entities are taken from [IANA06] and presented:

Documentation is a very important quality characteristic. A text entity is defined by using an expert vocabulary V_T . to justify the documentation, the following must be taken into consideration so that:

- the bibliography's article titles must have as a base a vocabulary V_B included in vocabulary V_T ;
- the article words written by the entity authors must form a vocabulary V_A included in the vocabulary V_T .

The quality of progression refers to the gradual nature of the approach. The concepts are treated from closer to closer so that:

- primary concepts that are clarified by examples are considered;
- based on primary concepts, new concepts are defined, some of them being derived from others;
- the connections between concepts are assured by formulas, examples and diagrams;
- the synthesis and analysis determine the particularization and aggregation of all the presented elements;
- the particularization is obtained from a definition of concepts to another.

The consistency implies the existence of some definitions, relations and presentations so that a logical succession is possible to be obtained. The particularization level grows with the growing text.

The uniformity consists in the usage with the same intensity of the bibliographic sources in the development of the concepts as well.

If a text entity E_T aims to tackle a specific field as a synthesis, this implies that a bibliography formed out of titles G_1, G_2, \dots, G_h , where h represents the length of the bibliography expressed as a number of used works, exists.

The uniformity means that the references of the works are done in equal measure for each of them. The analysis of this quality characteristic implies the parsing of E_T entity for frequencies f_i computing for which the work G_i from the bibliography is referenced, $i = 1, 2, \dots$

For the quality analysis of the entities, the following must be obtained:

- the length of the entities E_T in total number of words;
- the length of the entity E_T in total number of essential words, which is obtained by eliminating the connection words and the words which are not modified regardless the context;
- the length of the entity vocabulary as a number of maximum orthogonal words, by regrouping the results obtained by deriving a root word;
- the frequencies for the essential words which form a vocabulary;
- the frequencies f_i for the totals G_i references, with $i = 1, 2, \dots, h$;
- the indicators used in determining the quality of the references, such as:
 - the quantity of bibliography titles, introduced without being quoted, I_{pn} , that it is calculated using the relation:

$$I_{pn} = \frac{\sum_{i=1}^h \alpha(G_i)}{h}$$

$$\text{where } \alpha(G_i) = \begin{cases} 1, & \text{when } f_i \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

- the quantity of the quoted titles, I_{pc} , being calculated using the relation:

$$I_{pc} = 1 - I_{pn}$$

- the quantity of excessively quoted titles, I_{pex} , calculated with the relation:

$$I_{pex} = \frac{\sum_{i=1}^h \beta(G_i)}{h}$$

$$\text{where } \beta(G_i) = \begin{cases} 1, & \text{when } |f_i - \max_{i \leq l \leq h} \{f_l\}| < \varepsilon \\ 0, & \text{otherwise} \end{cases}$$

- the list of excessively quoted titles, $G_{i1} G_{i2} \dots G_{i0}$ for which $|f_i - \max\{f_i\}| < \varepsilon$;
- the list of unquoted titles, $G_{i1} G_{i2} \dots G_{ir}$ for which $f_{is} = 0, js = \{i_1, i_2, \dots, i_r\}$.

Proportionality is a very important characteristic by the sense and especially the effects which are determined during each entity reference process.

As well as all the reality is formed out of objects, processes, phenomena, beings characterized by structures formed out of components which interact, the text entities, as a reflection of the reality, even if they are artificially constructions, also consist of interacting components.

Proportionality is represented through the attention for the real world analysis. To a complex subsystem, a subtext with a bigger length than the length of a simple subsystem must correspond. Proportionality is represented by the relation within the text entity, itself.

Intraorthogonality is the quality characteristic that marks the differences among some text entity components.

The E_T text entity having the tree structure from figure 14 is considered.

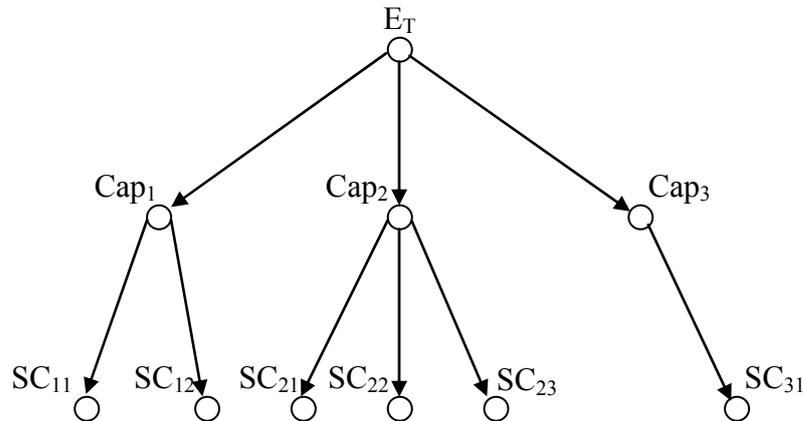


Figure 14. Tree structure corresponding to a text entity

The subtexts SC_{ij} are orthogonal between themselves if and only if $H(SC_{ij}; SC_{kl}) \rightarrow 1$, for $\forall i, k \in \{1, 2, 3\}$ with $i \neq k$ with $j, l \in \{1, 2, 3\}$ and $j \neq l$

Intraorthogonality reveals how different the parts that form a text entity are.

In case in which some concepts are treated again, the intraorthogonality declines.

When at each subchapter level the concepts and concepts connections graph are planned, the result consists in sub-graphs with nodes arranged on levels.

The nodes are connected only with simple arcs that connect nodes from adjacent levels, as showed in figure 15.

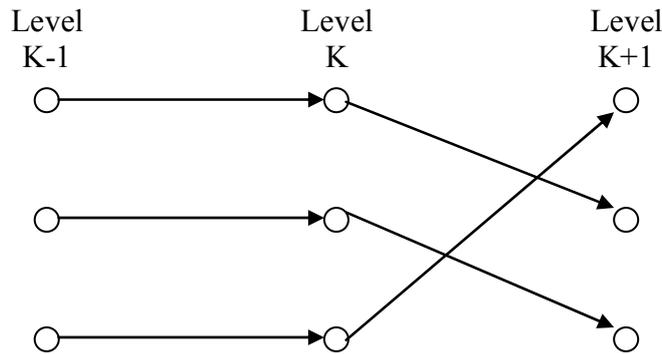


Figure 15. Graph structure with arcs connecting adjacent nodes

Interorthogonality refers to the entities differences and reveals the measure in which they differ as presentation form or content.

Considering two text entities, they tend to be orthogonal, as they do not treat similar concepts.

For example, the works elaborated in a specific field, are orthogonal if their texts have in common only the words that are characteristic for the field with the other words being very different as apparition frequency and position [IVAN05].

4. Software for evaluating structured entities

With the purpose of dependency graphs elaboration and analysis performing based on showed structures for obtaining the desired different forms of output data, SESE application is developed – Software for Evaluating Structured Entities. The behaviour of the application in different contexts is analyzed within this chapter. The product development is done in .NET platform, by using the C# language, according to the diagram presented in figure 16.

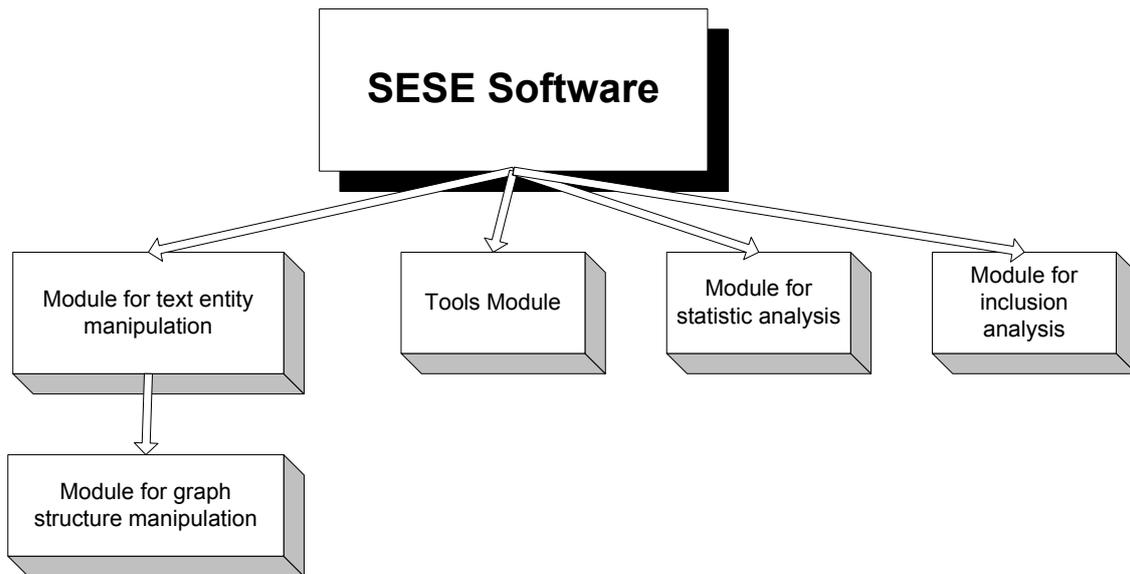


Figure 16. Software structure

The *module for graph structure manipulation* contains the graph structure definition that represents the most important part of the whole application..

The *module for text entity manipulation* refers to the structure which is defined within the module according to the concepts presented in chapter 1.

The *tools module* is composed of different objects used within the application for many types of operations such as: windows and XML files manipulation.

The *statistical analysis module* elaborates the dependency graph for some entities based on statistic indicators.

The *module for inclusion analysis* contains functions and algorithms for dependency analysis among the entities based on their vocabularies.

For **statistical analysis** module testing, the sets containing input data are shaped by collecting options tables from the terminal year students, in which they state the disciplines order that they consider as optimal for all concepts within the disciplines understanding. The results generated by the application with 9 representative input sets of data are presented:

- the set 1 with input data is presented in table 1:

Table 1. The set 1 with input data

Discipline Name	S ₁	S ₂	S ₃	S ₄
Alg – Algebra	Alg	Ec	Stat	Stat
Ec – Economics	Ec	Alg	Alg	Ec
Stat – Statistics	Stat	Stat	Ec	Alg
Micro – Micro-economics	Micro	Micro	Micro	Micro
Macro – Macro-economics	Macro	Macro	Macro	Macro

The dependency graph for the disciplines presented in table 1 is the one in figure 17. which shows three independent disciplines and another discipline, *Micro-Economics*, which is dependent of all three, and also *Macro-economics* which is dependent of *Micro-economics*.

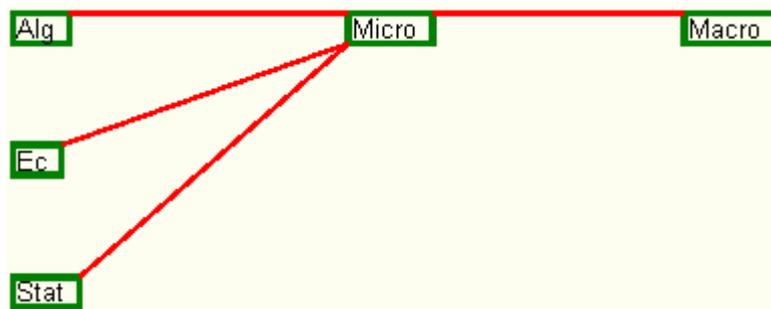


Figure 17. The dependency graph for set 1 of input data

- the set 2 with input data, which is presented in table number 2, has as result, as figure 18 shows, a graph in shape of a triangle, in which OOP and Networks are independent between each other, but both dependent on PE, and PM dependent both on OOP and on Networks.

Table 2. The set 2 with input data

Discipline Name	S₁	S₂
PE – Programming Engineering	PE	PE
OOP – Object Oriented Programming	OOP	NET
Net - Networks	Net	OOP
PM – Project Management	PM	PM

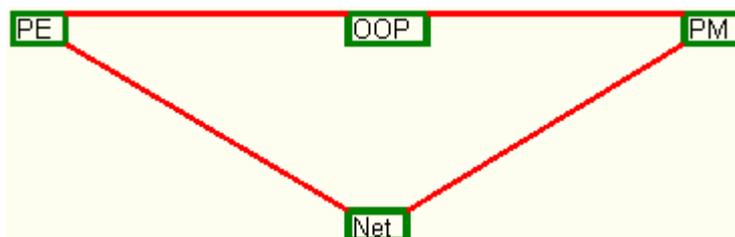


Figure 18. The dependency graph for set 2 of input data

- the set 3 of input data according to table number 3, leads after applying the algorithm to the graph presented in figure 19.

Table 3. The set 3 with input data

Discipline Name	S₁	S₂	S₃	S₄
OS – Operating Systems	OS	OS	OS	OS
BTI - Business Technologies on Internet	BTI	MM	DBMS	DBMS
MM – Multimedia	MM	BTI	BTI	MM
DBMS – Database Management Systems	DBMS	DBMS	MM	BTI
PM – Project Management	PM	PM	PM	PM
SDU – Software Development and Usage	SDU	APL	SDU	APL
APL – Advanced Programming Languages	APL	SDU	APL	SDU

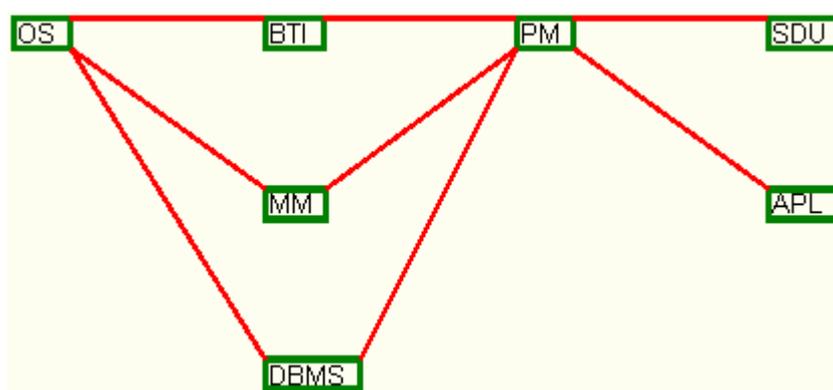


Figure 19. The dependency graph for set 3 of input data

Figure 19 presents a combination of precedent cases, in which dependencies as well as independencies are presented together.

- the set 4 of input data is represented by the table number 4:

Table 4. The set 4 with input data

Discipline Name	S ₁	S ₂	S ₃	S ₄
DS – Data Structures	DS	PE	PA	PA
PE – Programming Engineering	PE	DS	DS	PE
PA - Programming Algorithms	PA	PA	PE	DS
DBMS – Database Management Systems	DBMS	DBMS	DBMS	DBMS
OS – Operating Systems	OS	CS	OS	CS
CS – Computing Systems	CS	OS	CS	OS
PM – Project Management	PM	PM	PM	PM

The resulted graph for data in table number 4 is drawn in figure 20, which is also a combination of the precedent test cases.

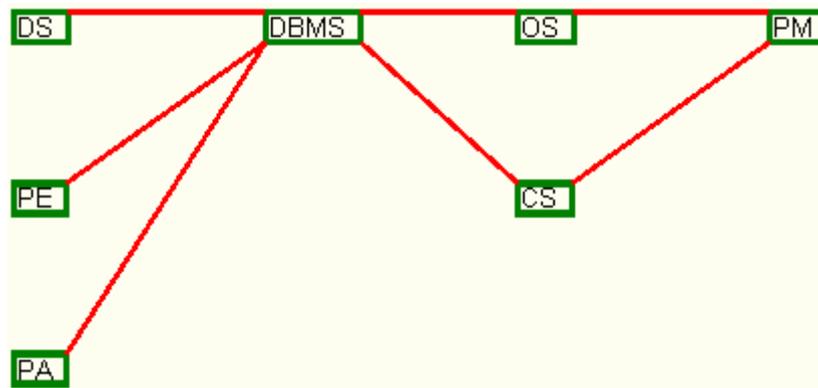


Figure 20. The dependency graph for set 4 of input data

- the set 5 of input data consists of data from the table number 5. According to it, all the students have chosen the disciplines in the same order, which means that they are dependent between each other based on the order in which they were chosen. The resulted graph is the one from figure 21.

Table 5. The set 5 with input data

Discipline Name	S ₁	S ₂	S ₃
DS – Data Structures	DS	DS	DS
PE – Programming Engineering	PE	PE	PE
PA - Programming Algorithms	PA	PA	PA
DBMS – Database Management Systems	DBMS	DBMS	DBMS
OS – Operating Systems	OS	OS	OS
CS – Computing Systems	CS	CS	CS
PM – Project Management	PM	PM	PM
SDU – Software Development and Usage	SDU	SDU	SDU



Figure 21. The dependency graph for set 5 of input data

The resulted graph is a linearly structured one, in which every node has a connection with a single node which also comes-next-after.

For testing the **inclusion analysis module**, instead of establishing the dependencies based on the statistic calculation, the connections between entities sub-vocabularies are considered. Thus, for the entity noted as E_T that has the components $SET_1, SET_2, \dots, SET_{NT}$, between them, relations like the ones seen in chapter 2 are established, these being essential for text entities structure creation. The application with various sets of input data returns the following results:

- the set 6 of input data refers to words dictionaries analysis and it is represented in table number 6 containing the definitions for words *football, game, activity and multitude*.

Table 6. The set 6 with input data

Word	Definition
Football	Game between 2 teams with 11 persons
Game	Fun activity for humans beings
Activity	Multitude of actions to reach a purpose
Multitude	Resulted from unification

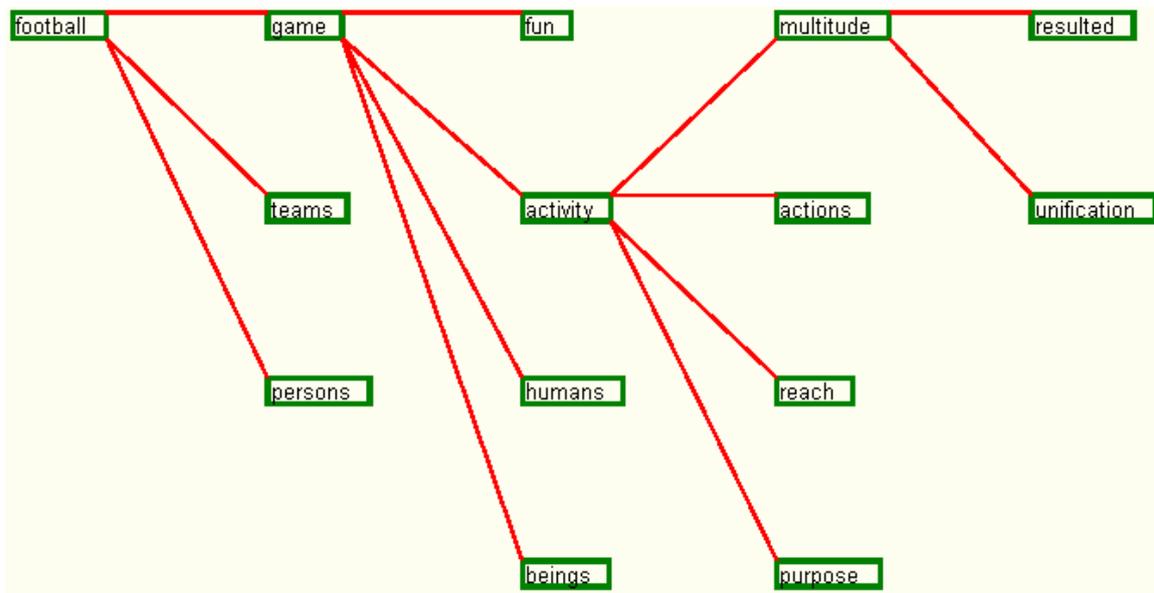


Figure 22. The dependency graph for set 6 of input data

After selecting the graph drawing option for the word *football*, the application returns the representation from figure 22 which shows all the necessary words for the understanding of word *football*, along with the connections between them.

After removing the intermediary words, the graph modifies, so that it contains just the root and the leaves of the tree represented in figure 22. The structure so modified is represented in figure 23 and it actually shows the necessary concepts to understand the word, without including the other words which have only been necessary to reach this level.

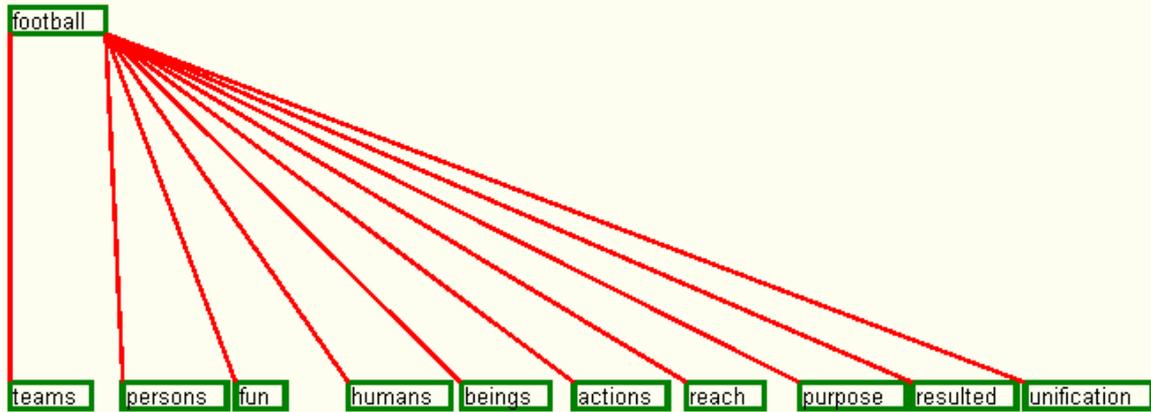


Figure 23. The dependency graph for set 1 of input data without intermediary nodes

- the set 7 of input data is represented by the activity fields like the ones seen in table number 7, that have associated specific concepts, instead of definitions according to a dictionary of words as previously.

Table 7. The set 7 with input data

Terms	Associations
Computer-Operations	Computer, Display, Keyboard, Windows
Programming	Computer-Operations, Algorithms
Project-Management	Programming, OOP

The resulted dependency graph for the *Project-Management* activity is described in figure 24.

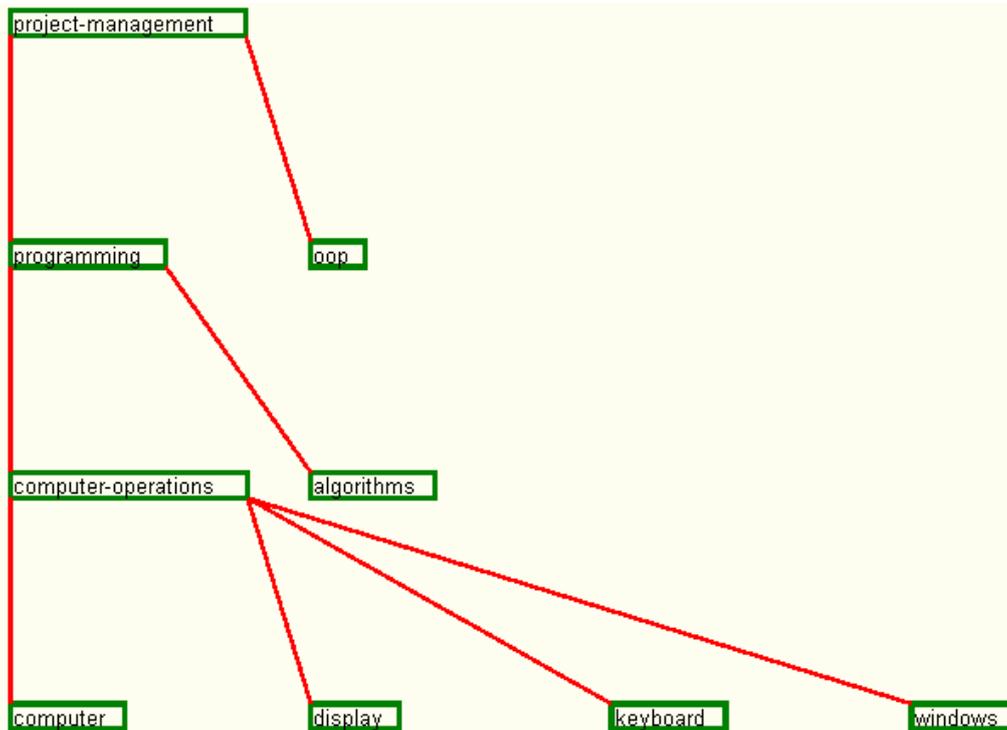


Figure 24. The dependency graph for set 7 of input data

Thus, all the activities and corresponding concepts are structured as a graph, in which the arcs between nodes show the dependencies. After removing the intermediary nodes, the resulted graph presented in figure 25 shows in a more simple form all the concepts which are necessary for the *Project-Management* activity.

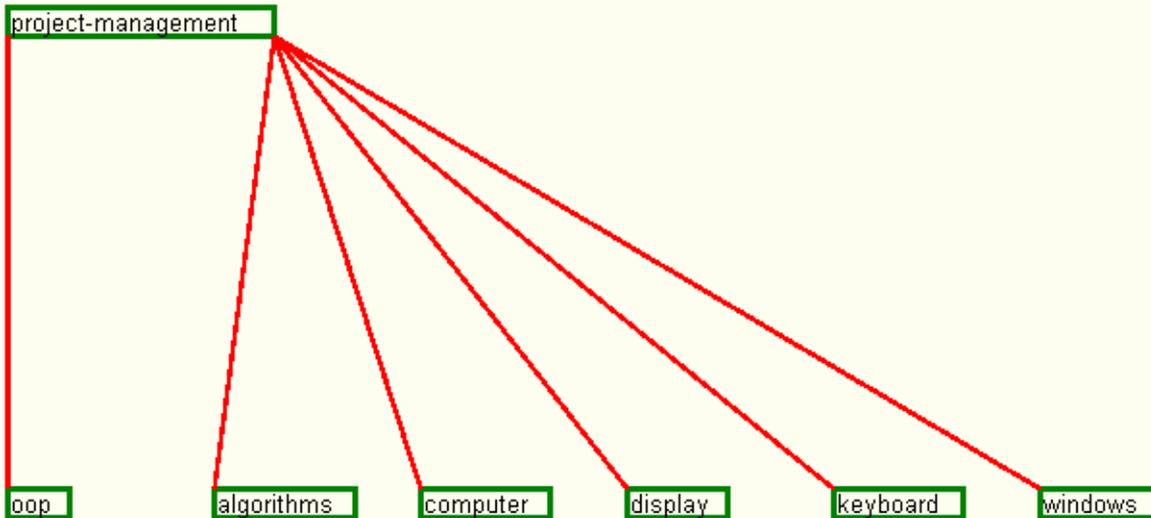


Figure 25. The dependency graph without intermediary nodes for the project-management activity according to set 7 of input data

- the set 8 of input data represents a grade one error from a dictionary of words. Thus, considering the definition from table number 8, the resulted graph is the one drawn in figure 27.

Table 8. The set 8 with input data

Word	Definition
Flower	Plant with shape as a flower

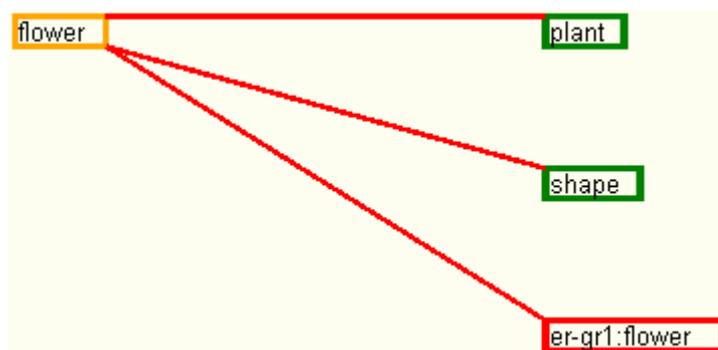


Figure 27. The dependency graph for set 8 of input data

The grade one error is stated when a recurrence within the definition occurs. In this way, the concept of *flower* is not fully understood because of the use of the same concept within the same definition.

- the set 9 of input data presents a grade two error for a dictionary of words. Two definitions for concepts of *flower* and *plant* are presented in table number 9.

Table 9. The set 9 with input data

Word	Definition
Flower	Nature plant
Plant	Flower that grows from soil

The resulted dependency graph is the one drawn in figure 28.

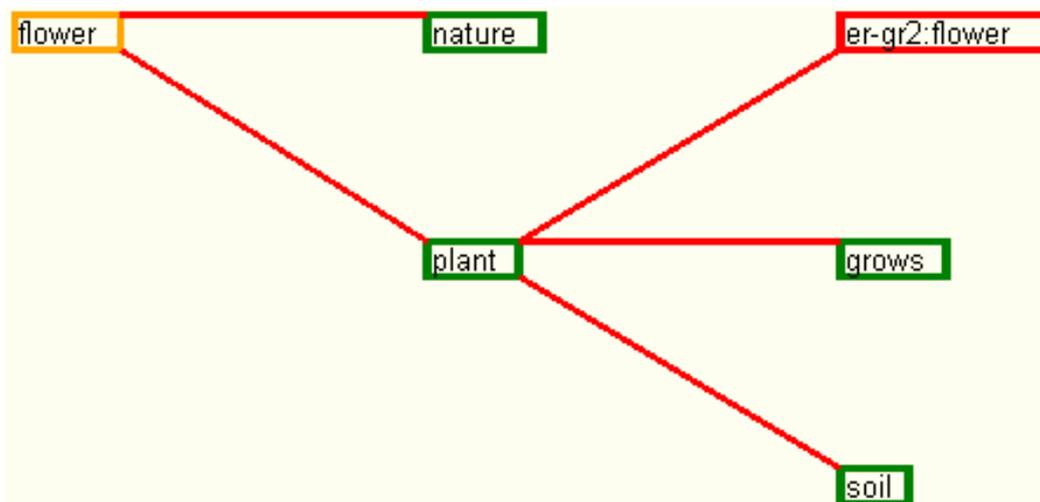


Figure 28. The dependency graph for set 9 of input data

Thus, for understanding the concept of *flower*, the word *plant* used within the definition presents its own definition, which contains the initial word *flower*. In this way, the dictionary contains an error, being qualified as grade two error because of the level at which the recurrence occurs.

Conclusions

The fields in which the interdependencies between different types of entities are able to be shaped and structured based on dependency graph are countless. The purpose of developing a software product to generate these types of graphs is to go deeply into the process of obtaining quality structures for precedence analysis of different types of entities.

The graph is a very malleable structure with possibilities to reach, especially by using recursive mechanisms, levels of advanced complexity. This structure, applied to text entity, offers the possibility of processing analysis or other operations on texts by using the computer. Text entity also offers a very big coverage. In this way, the linking of the two concepts leads to notable results.

Bibliography

1. Ion IVAN, Cătălin BOJA **Metode statistice în analiza software**, Bucharest, Publisher ASE, 2004
2. Ion IVAN, Dragoș ANASTASIU **Structured Entities Structures Building**, Workshop Information Systems & Operations Systems Proceedings, 1 - 2 March 2006, Universul Juridic Printing House, Bucharest 2006, pp. 298 – 305
3. Ion IVAN, Marius POPA **ENTITĂȚI TEXT, dezvoltare, evaluare, analiză**, Publisher ASE, 2005, 587pg
4. Ion IVAN, Marius POPA **Reprezentativitatea entităților text**, work accepted at International Conference "The Impact of European Integration on the National Economy", University "Babeș-Bolyai" Cluj-Napoca, 2005
5. Ion IVAN, Marius POPA **Consistența entităților text**, Studii si Cercetări de Calcul Economic și Cibernetică Economică, 2005
6. Donald KNUTH **The art of computing programming**, Volume 1
7. Tom ARCHER, Andrew WHITECHAPEL **Inside C#**
8. Jesse Liberty **Programming C#**, Publisher O'Reilly, 2005

¹ Dragos ANASTASIU is 24 years old. He graduated Academy of Economic Studies, Faculty of Cybernetics, Statistics and Economic Informatics, section Economic Informatics.
He follows the Informatics Security master courses.
He also carries on research in the field of text entities structures.
He published an article in "Workshop Information Systems & Operations Systems Proceedings" regarding structured entities structures building.
He programs using C++, C# and ASP.NET for web pages development.

² Codification of references:

[BOJIO04]	Ion IVAN, Cătălin BOJA Metode statistice în analiza software , Bucharest, Publisher ASE, 2004
[IANA06]	Ion IVAN, Dragoș ANASTASIU Structured Entities Structures Building , Workshop Information Systems & Operations Systems Proceedings, 1 - 2 March 2006, Universul Juridic Printing House, Bucharest 2006, pp. 298 – 305
[IVAN05]	Ion IVAN, Marius POPA ENTITĂȚI TEXT, dezvoltare, evaluare, analiză , Publisher ASE, 2005, 587pg
[IOPMA05]	Ion IVAN, Marius POPA Reprezentativitatea entităților text , work accepted at International Conference "The Impact of European Integration on the National Economy", University "Babeș-Bolyai" Cluj-Napoca, 2005
[IIVAN05]	Ion IVAN, Marius POPA Consistența entităților text , Studii si Cercetări de Calcul Economic și Cibernetică Economică, 2005
[KNUTH01]	Donald KNUTH The art of computing programming , Volume 1
[MICRO02]	Tom ARCHER, Andrew WHITECHAPEL Inside C#
[LIBER01]	Jesse Liberty Programming C# , Publisher O'Reilly, 2005