

# The Study of Machine Translation Aspects Through Constructed Languages

Evangelos C. Papakitsos<sup>1, \*</sup>, Ioannis Giachos<sup>2</sup>

<sup>1</sup>Department of Education, School of Pedagogical and Technological Education, Iraklio Attikis, Greece

<sup>2</sup>Department of Linguistics, National & Kapodistrian University of Athens, Athens, Greece

## Abstract

The present paper describes a software system that performs bidirectional machine translation between two constructed languages. These languages are made by one or more persons, for various purposes. Such an important purpose is the development of easy and almost natural communication interfaces with robots. Despite the linguistic simplicity of the constructed languages, the automated translation from one to the other confronts some of the fundamental algorithmic challenges that are also encountered in the machine translation of natural languages. Hence, the usage of constructed languages can be an easier way both to train linguistic engineers in developing machine translation software and to study the linking of different robotic interfaces, as a novel field of research.

## Keywords

Natural Language Understanding, Constructed Language, Machine Translation, Human-Computer Interaction

Received: June 13, 2016 / Accepted: June 25, 2016 / Published online: July 21, 2016

© 2016 The Authors. Published by American Institute of Science. This Open Access article is under the CC BY license.

<http://creativecommons.org/licenses/by/4.0/>

## 1. Introduction

Machine translation still remains one of the most challenging applications of natural language processing [1]. In reality, we should better refer to it as computer assisted translation, because human intervention remains indispensable [2]. A significant source of difficulty is that various linguistic features are expressed differently by different natural languages, making the mapping of strings from one language to another a laborious task for linguistic computing engineers [3]. The existence of well-trained engineers is a necessity and a prerequisite for qualitative software applications. Thus, constructed languages may contribute significantly to this academic effort.

Another application of constructed languages is the development of a human-computer/robot interaction system in an almost natural-communication interface [4]. Such an interface presupposes that either the humans are trained in speaking the constructed language of their robot or,

alternatively, that a real-time machine translation system is developed, performing a bidirectional translation (interpretation) between a natural and a constructed language.

Consequently, novel machine translation applications may arise, covering the two novel translation modes:

- a natural language into a constructed one and vice-versa;
- one constructed language into another (constructed one) and vice-versa, in case of machines with different software installations.

Other research repercussions and possibilities, concerning augmented natural language understanding, will be discussed in the last section.

## 2. Constructed Languages

An *artificial language* (also known informally as *conlang* [5]), is a language that its phonology, grammar and vocabulary have been consciously devised by an individual

\* Corresponding author

E-mail address: [papakitsev@sch.gr](mailto:papakitsev@sch.gr) (E. C. Papakitsos)

or a group of them, instead of having evolved naturally. There are many possible reasons to create an artificial language: for facilitating human communication (e.g., Esperanto); linguistic experimentation; artistic creation and expression; language games. The term *glossopoeia* was coined by John Tolkien (who constructed Quenya in 1917), to indicate the construction of a language, especially an artistic one [6].

A *philosophical language* is any artificial language constructed by the first basic rules, such as a logical language, but can claim absolute perfection, transcendence or even the mystical truth and not the satisfaction of realistic goals [7]. Philosophical languages were popular in modern times, partly motivated by the goal of recovering the lost Adamic or Divine language (the language that according to the Jews, as recorded in Midrashim, and some Christians was spoken by Adam in Paradise [6]). The term *ideal language* sometimes is used almost synonymously, although more modern philosophical languages like *Toki Pona* [8] is less likely to achieve such a high requirement of perfection. The axioms and grammars of those languages differ from commonly spoken languages today. In most of the oldest philosophical languages, and some newer ones, words are constructed from a limited set of morphemes treated as *basic* or *fundamental*.

The vocabularies of *oligosynthetic languages* [9] are made of compound words, which were devised by a small (theoretically minimum) number of morphemes. Similarly, *oligoisolating languages* like Toki Pona use a limited set of root-words, but produce sentences that are series of distinct words. Toki Pona is based on minimalistic simplicity, incorporating elements of Taoism. *Láadan* has been designed to lexicalize concepts and distinctions that are important to women, based on the *muted group theory* [10]. The *a priori* (from the beginning) is a constructed language, where the vocabulary is created from scratch (e.g., Dama Diwan [11]), rather than from other existing languages (like Esperanto or Interlingua). Philosophical languages are almost all *a priori*, but most *a priori* languages are not philosophical. For instance, Quenya is an *a priori* but not a philosophical language. Its goal is to look like a natural language, even if it has no genetic relationship with any natural language.

Historically, philosophical languages appeared in 1647 with the pioneer Francis Lodwick. It is noteworthy that in 1678 Gottfried Leibniz, in order to create a dictionary of characters in which the user can perform calculations that will give automatically real proposals, developed the *binary calculus* as a side effect. In those years, projects were created that were designed not only to reduce or model grammar, but also to organize all human knowledge into “characters” or hierarchies. This idea eventually led to the Encyclopaedia of

Enlightenment. Leibniz and the Encyclopaedists realized that it is impossible to organize human knowledge unequivocally as a tree-structure, and so it is impossible to construct an *a priori* language based on such a classification of concepts. After Encyclopaedia, plans for an *a priori* language were marginalized more and more [6].

### 3. Natural Semantic Metalanguage

*Natural Semantic Metalanguage* (NSM [12]), is a linguistic theory and practice, which aims to eliminate all the confusion of cross-cultural communication. This is achieved by using a set of basic and universal concepts, known as *semantic primes*, which can be expressed in words or other linguistic expressions in all languages. The theory of NSM debuted in 1972 in the book *Semantic Primitives* of the Polish-Australian linguist Anna Wierzbicka [13]. It is based on a centuries-old idea for a language of the mind. It is nowadays recognized internationally as one of the leading theories in the world of language and meaning.

The use of NSM allows us to develop tests that are clear, precise, cross-translatable, non-Anglo-centered and understood by people without specialized language training. The method has applications in intercultural communication, lexicography, language teaching, child language acquisition and other areas. Below are some NSM concepts (primes), coded as English words. These concepts are supposed to be linguistically universal. Most of them have been tested in a wide variety of languages without causing ambiguities. The English exponents of some semantic primitives are given below [14]:

- Substantives: I, YOU, SOMEONE, PEOPLE, SOMETHING, BODY;
- Determiners: THIS, THE SAME, OTHER/ELSE;
- Quantifiers: ONE, TWO, MUCH/MANY, SOME, ALL;
- Evaluators: GOOD, BAD;
- Descriptors: BIG, SMALL;
- Speech: SAY, WORDS, TRUE;
- Time: WHEN, NOW, BEFORE, AFTER, MOMENT;
- Space: WHERE, HERE, ABOVE, BELOW, FAR, NEAR, SIDE, INSIDE;
- Similarity: LIKE/WAY.

As an example, we can give one sentence in English without the use of universal concepts (non-prime concept), followed by the respective sentences that make up the same meaning written in semantic primes [15]:

“Someone X killed someone Y”,

- someone X did something to someone else Y;
- because of this, something happened to Y at the same time;
- because of this, something happened to Y's body;
- because of this, after this Y was not living anymore.

The term *metalanguage* is used not only in linguistics, but in sciences as well, especially in computing. In linguistics, *metalanguage* is called a set of words, phrases, terms, signs and symbols that describes or analyses the language itself. Thus, a metalanguage may be regarded as a language for languages. Such a metalanguage can be used to formulate rules, theories or relations regarding the actual language. Terms of this metalanguage are: subject, determiner, verb, object, adverb, etc. In linguistic computing (natural language processing), these terms or equivalent ones are parts of the grammar formalisms that describe a language in a mathematically rigorous manner. The relation of NSM to constructed languages will be discussed in the last section, especially considering the two constructed languages, processed herein: Toki Pona and Minimal Extent Free Greek (MEFG).

## 4. Toki Pona & MEFG

Toki Pona, as mentioned before [8], is a constructed language that was introduced in 2001, designed by the translator and linguist Sonja Lang (formerly Sonja Elen Kisa), from Toronto. This is a minimal language that focuses on simple concepts and the related commonalities between cultures. It was designed to express maximum meaning with minimum complexity and claims to be the easiest language in the world, yet ideal for conveying basic concepts. The name means “simple / good language”, constructed with Zen style, according to Sapir-Whorf Hypothesis. For creating the words of that language, measurements provided by the Department of Psychology of the University of Ghent were utilized, so that the most frequently occurring words (in English meaning always) have the shortest length of characters. This process resulted to the use of the minimum number of letters. It has 14 phonemes and 124 words. The grammar of Toki Pona is simple. The rules apply equally to all the words and there is no exception. This language does not contain all parts of speech. The names of persons are the same as those of the natural languages. An example of a sentence is given below, in English, in Toki Pona and in the literal translation of Toki Pona back to English:

- I love this fruit.
- Pito loki wikute.

- I love fruit.

All the words are incorporated in the vocabulary of ROILA (ROBot Interaction LAnguage), which is an open international project in progress [4], for constructing a language exclusively for robots.

MEFG (Minimal Extent Free Greek) is also an artificial language, similar to Toki Pona, with 137 words of Greek origin written in Greek alphabet, designed by Ioannis Kenanidis [16]. The conception of the idea came in 1993 that led in 2007 to the construction of the *Free Greek Language*, which used a small (minimal) grammar but the entire Greek vocabulary. The evolution process led initially to MEFG and then to SostiMatiko [17] in 2013. SostiMatiko is also a constructed language. Its vocabulary consists of 222 words, written in the Latin alphabet as the first choice, with the Cyrillic alphabet as a second one and the third option is the Japanese Katakana. SostiMatiko has been applied for enhancing the artificial intelligence of a machine, especially regarding natural language understanding [6]. In MEFG, although the order of the words can be defined by the user and should not be rigorous, Kenanidis argues that in a minimal language the word order should be SVO (subject - verb - object) and AN. “AN” means that the words that characterize a noun/verb (adjectives or adverbs) as well as their supplements should precede. Generally, all the words that characterize or complete any other word, including referential sentences, must precede. The reason why we must follow SVO and AN is that otherwise the language will not be minimal because we need additional indicators. Compare when we say “easy work” to the equivalent “the work that is easy”.

## 5. Methodology

Efficient machine translation (or almost natural communication with a robot) is mainly considered to be a problem of encoding semantics [25]. One possible way to accomplish such a task could be firstly the encoding of a NSM (section 3). Then, the words and the syntactic structures of the source language can be mapped to the NSM equivalent ones. Subsequently, the NSM structures are mapped to the equivalent ones of the target language. The first goal for exploring the relevant computational procedures is to minimize complexity. Therefore, the usage of constructed languages (section 2) may facilitate the implementation of this goal. Especially constructed oligosynthetic languages offer minimal vocabularies and simple syntactic structures that can easily simulate the previous mapping process. Moreover, the meanings of their (limited in number) words are very close to the semantic primes of NSM, thus making them the ideal candidate for such experimentation.

An important consideration is which pair of constructed languages to choose for a similar project. It would be better (less complex) if the two languages have comparable expression potentialities. For example, ROILA and SostiMatiko is not such a suitable pair, because the former has a vocabulary of 800 words while the latter has 222 words. This difference requires composite word-mapping from one language to the other. On the contrary, Toki Pona (having 124 words) and MEFG (having 137 words) is a suitable pair of languages (section 4).

## 6. The Machine Translation System

The purpose of this project was the development of a software system that automatically translates texts from MEFG into Toki Pona and vice versa [18]. The program called Mini Translator is developed in Visual C# (e.g., see [19]). The translation is conducted by a simple mapping of words, where four cases are distinguished:

- Some words (or affixes) of MEFG have no corresponding translation in Toki Pona. In that case, the translation is done with words of English enclosed in square brackets. If the translation refers to a grammatical attribute, then inside the square brackets the equal sign (=) precedes or the property appears in capital letters: e.g., “[PASSIVE]” (voice).
- Some words of MEFG are translated periphrastically into two of Toki Pona words, separated by a hyphen.
- The proper names in Toki Pona remain unchanged in MEFG.
- The proper names in MEFG are transcribed with the Latin alphabet into Toki Pona, according to the rules of the latter’s phonetics.

The mapping of syntax rules is not followed exactly. In Toki Pona the determiner follows the designated, while in MEFG it is the opposite. For the sake of computational simplicity, the developer of the software considered that MEFG, being “Free”, can respond to this change in order to maintain the syntactic structure of Toki Pona. The unique cases of adaptation are the three prefixes-prepositions of Toki Pona, which are converted into suffixes in MEFG. Namely, the movement of these affixes is executed from the previous position to the left of the word in Toki Pona, on the right after the end of the corresponding word in MEFG. The process is reversed when the role between the source and the target language is also reversed.

## 7. Program Structure

The physical structure of the software system is a folder that includes the following parts:

- 1) The input text file with the text of the source language for translation.
- 2) The output file with the translated text in the target language. This file is not initially present but created after each execution of the translation program.
- 3) The executable file with the translation program.
- 4) The database folder, containing the lexicon text files (see No 5) and the alphabet text file (see No 6).
- 5) The lexicon text file contains the mapping of words between Toki Pona (first column) and MEFG (second column), separated by blank characters (SPACE). The entries are sorted alphabetically according to the words of Toki Pona, with one match per line.
- 6) The alphabet text file contains the letter assignment (and diphthongs) between MEFG (first column) and Toki Pona (second column), separated by the character {-}.

The software package is complemented by an optional folder, containing examples of texts and translations.

The program (source files) consists of five classes (C0-C4). Classes C0 and C1 implement the interface of the software system. Class C2 implements the data management module, accessing the database folder (see No 4). Classes C3 and C4 implement the processing subsystem, which performs automatic translation using the input text file (see No 1) and the output text file (see No 2). In a few details:

- Class C0 displays the initial interface form. It contains the user instructions, the selection buttons for the source language and the command key for the translation. It also initializes the program’s data structures by transferring the data in the database. Then, depending on the activation of the selection buttons (bidirectional ability), it calls the corresponding processing function in order to perform the translation. If there is a delay due to the large size of the input file then it calls the activation of C1 form, with the delay message. Once the translation is extracted in the output file, it displays a window with the completion message.
- Class C1 shows the window with the delay message, when the completion of processing delays, until the end of the translation.
- Class C2 initializes the data structures of the program by uploading the data from the external database (see No 4). The entries of the lexicon file (see No 5) are sorted firstly by alphabetical order and secondly by descending order of

size (number of characters), according to the words of Toki Pona in the relevant table. A similar table is also initialized according to the words of MEFG. Each table is accessed according to which language is the source one, respectively. Another table is also filled with an alphabetical mapping between the Greek and Latin letters. This table is used for the transcription of MEFG proper names into Toki Pona.

- Class C3 translates text from Toki Pona into MEFG. A function performs the movement of Toki Pona prefixes, which were previously translated into MEFG (la = α; e = ν; li = ει), to become suffixes in MEFG, as shown in the following example (in italics):

i) *la mi > α εμέ > εμέ α* (= “I [sub-clause separator]”).

ii) *e toki > ν λόγο > λόγο ν* (= “speech [direct object indicator]”).

iii) *li lukin > ει βλέπ- > βλέπ-ει* (= “see [predicate indicator]”).

- Class C4 translates text from MEFG into Toki Pona. A function performs the movement of MEFG suffixes to become Toki Pona prefixes, in the reverse manner to Class C3 (previously), as shown in the following examples (in italics):

iv) *εμέ α > mi la > la mi* (= “[sub-clause separator] I”).

v) *λόγο ν > toki e > e toki* (= “[direct object indicator] speech”).

vi) *βλέπ-ει > βλέπ- li > lukin li > li lukin* (= “[predicate indicator] see”).

Finally, a transliteration function performs the transcription of words (proper names) that were not already translated, from the Greek alphabet to the corresponding ones of Toki Pona. The transfer is executed by means of the alphabet file (see No 6).

## 8. Function & Results

The use of the machine translator begins with the activation of the executable file (see No 3). With the start of the program, the user reads the initial form instructions and selects the target language between the two available languages (MEFG, Toki Pona). The text for translation must be written in the input text file (see No 1). Then, the button TRANSLATION is clicked. If the text is too large then a small form with a delay message may appear. Once the translation is completed, a window appears with the completion message. Upon acceptance of the completion message, the translation process can be continued with another piece of text. The translated text is placed into the

output text file (see No 2). An example of the results is presented below. It is the first verse of the Lord’s Prayer in Toki Pona [8] translated into MEFG, with a literal translation of MEFG into English (in italics):

- Lord’s Prayer: *Our Father in heaven, hallowed be your name.*
- Toki Pona: *mama pi mi mute o, sina lon sewi kon.*
- MEFG: *γονιό απο εμε πολύ ώω, εσε είναι άνω πνοή.*
- English: *Parent from me very (salute), you be upper puff.*

Even in a machine translation program of such a small size (355 lines of code) and purpose, bidirectional asymmetry is presented among the results [20]. The test was conducted by re-translating the output text, originally translated into MEFG, back again into Toki Pona:

Toki Pona > MEFG > Toki Pona.

When returned, the following changes occur:

- Because there are two versions of the Toki Pona word “all” in the dictionary, only one is returned: *ali > όλο > ale.*
- In this version of the software system, during the course of action: {MEFG > Toki Pona > MEFG}, the returned MEFG text is not the original one, because of the extra features available in MEFG (genders, collocations [21], grammatical features). Two examples of these extra features are: the passive voice indicator “εται” ([PASSIVE] εται) and the plural number indicator “ς” ([=plural] ς) that are absent in Toki Pona and therefore are substituted by the content of the respective square brackets.

Additionally, the determiners that in Toki Pona follow the designated word still follow in MEFG, as well, while it is preferable to precede. Corrections that will deal with some of the previous results require extra layers of processing, with an increased degree of complexity.

## 9. Discussion & Conclusions

It has been previously presented (section 4) that constructed languages have a very limited vocabulary. The two processed herein languages have 124 (Toki Pona) to 137 (MEFG) words and affixes. These words function both as roots and morphemes for creating compounds and collocations. Thus, by being minimal forms, they express very basic meanings that are called *sememes* [22] [23]. The exact manner of expressing more composite meanings from sememes, and consequently of creating composite lexical structures from the minimal forms, is left to the imagination of the speaker. For example [24], the word “wheel” could be expressed as “round leg” or as

“moving circle”. Regardless of the particular choice, this process is similar, if not identical, to the creation of a NSM (section 3), where the semantic primes are encoded as the sememes of the minimal forms (words / morphemes).

The encoding of meanings, as semantic primes or sememes and their combinations, is the ultimate prerequisite for Natural Language Understanding (NLU). Only NLU applications may provide human-computer interaction (HCI) in natural language or efficient machine translation for natural language pairs of very different typology, like Spanish and Japanese. In fact, the latter case constitutes a problem of “meaning preservation” from one language to the other, where the encoding of semantics is considered to be the “ultimate level of cross-lingual representation” [25]. Therefore, the works on how to encode semantics, as a prime goal of artificial intelligence, are evident at least since the early seventies [26] and continuously up to now [27], not only as computational experimentations but also as formal theoretical constructs [28].

Two possible methods of encoding meaning will be discussed herein. The first one encodes the meaning of words as a semantic network. A representative application of this method is the WordNet lexical-semantics database [29]. It is a large lexical database, where all the words of a natural language are grouped in tree data-structures according to their semantic relationship, contrary to the traditional alphabetical grouping. For example, words like “oak” and “fir” are grouped together, since they are both trees. The WordNet method has been implemented in many languages, especially for machine translation applications [30]. The mapping between the same word of the source to the target language is conducted through their same position (node) in the semantic tree (data-structure). Yet, the processing of collocations, like “credit card”, is not a trivial matter at all [31]. The second method of encoding meaning, which is proposed herein for experimentation, is through NSM and constructed languages. Namely, the semantic database is initially composed of the semantic primes or the sememes of a constructed language. Then, every other word of a more composite meaning is a combination of sememes. This concept is equivalent to the RISC CPU-architecture [32], compared to the WordNet method that is equivalent to the CISC CPU-architecture [33].

In this respect, a HCI natural language interface may consist of a machine translation system that processes a natural language (for the person) and a constructed language (for the machine), which is probably easier than having a person to learn a constructed language for full scale communication. Consequently, the contribution of constructed languages to machine translation and particularly to semantic encoding seems worth exploring further.

## References

- [1] Baljinder Kaur & Brahmaleen Kaur Sidhu (2014). Machine Translation: An Analytical Study. *International Journal of Engineering Research and Applications*, 4(5), Version 7: 168-175.
- [2] Raybaud S., Langlois D. & Smaïli K. (2011). “This sentence is wrong.” Detecting errors in machine-translated sentences. *Machine Translation*, 25: 1–34.
- [3] Dorr B. J. (1994). Machine Translation Divergences: A Formal Description and Proposed Solution. *Computational Linguistics*, 20(4): 597-633.
- [4] <http://roila.org>.
- [5] [https://en.wikipedia.org/?title=Constructed\\_language](https://en.wikipedia.org/?title=Constructed_language).
- [6] Giachos I. (2015). *Implementation of OMAS-III as a Grammar Formalism for Robotic Applications*. Postgraduate Dissertation, National & Kapodistrian University of Athens and National Technical University of Athens, pp. 2-3 (in Greek).
- [7] [https://en.wikipedia.org/wiki/Philosophical\\_language](https://en.wikipedia.org/wiki/Philosophical_language).
- [8] [http://en.wikipedia.org/wiki/Toki\\_Pona](http://en.wikipedia.org/wiki/Toki_Pona).
- [9] [https://en.wikipedia.org/wiki/Oligosynthetic\\_language](https://en.wikipedia.org/wiki/Oligosynthetic_language).
- [10] <http://www.laadanlanguage.org>.
- [11] [http://linguifex.com/wiki/Dama\\_Diwan](http://linguifex.com/wiki/Dama_Diwan).
- [12] [https://en.wikipedia.org/wiki/Natural\\_semantic\\_metalinguage](https://en.wikipedia.org/wiki/Natural_semantic_metalinguage).
- [13] Wierzbicka A. (1972). *Semantic Primitives*. Frankfurt: Athenäum.
- [14] Goddard C. (2008). Natural Semantic Metalanguage: The state of the art. In C. Goddard (ed.), *Cross-Linguistic Semantics*. Amsterdam: John Benjamins, p.33.
- [15] Goddard C. (2010). The Natural Semantic Metalanguage approach. In Bernd Heine and Heiko Narrog (eds.), *The Oxford Handbook of Linguistic Analysis*. Oxford: Oxford University Press, pp. 459-484.
- [16] <http://independent.academia.edu/GiannhsKenanidhs/Teaching-Documents>.
- [17] [www.unilang.org/blog/SostiMatiko](http://www.unilang.org/blog/SostiMatiko).
- [18] Papakitsos E. (2013). *Mini Translator: Software of bidirectional machine translation between the artificial languages of Toki Pona and Minimal Extent Free Greek*. Athens: National Library of Greece (in Greek).
- [19] Foxall J. (2009). *Visual C# 2008 in 24 Hours: Complete Starter Kit*. SAMS Teach Yourself, 2<sup>nd</sup> Printing. Indianapolis IN: Pearson Education.
- [20] Simova I. & Kordoni V. (2013). Improving English-Bulgarian Statistical Machine Translation by Phrasal Verb Treatment. In J. Monti, R. Mitkov, G. Corpas Pastor & V. Seretan (Eds.), *Workshop Proceedings for: Multi-word Units in Machine Translation and Translation Technologies* (Organized at the 14th Machine Translation Summit, 2-6 September 2013, Nice, France). Switzerland: The European Association for Machine Translation, p. 64.

- [21] Barreiro A., Monti J., Orliac B. & Batista F. (2013). When Multiwords Go Bad in Machine Translation. In J. Monti, R. Mitkov, G. Corpas Pastor & V. Seretan (Eds.), *Workshop Proceedings for: Multi-word Units in Machine Translation and Translation Technologies* (Organized at the 14th Machine Translation Summit, 2-6 September 2013, Nice, France). Switzerland: The European Association for Machine Translation, pp. 26-33.
- [22] <https://en.wikipedia.org/wiki/Sememe>.
- [23] Babiniotis G. (1985). *Introduction to Semantics*. Athens: G. Gkelbasis, p. 47 (in Greek).
- [24] Giachos I. (2015). *Implementation of OMAS-III as a Grammar Formalism for Robotic Applications*. Postgraduate Dissertation, National & Kapodistrian University of Athens and National Technical University of Athens, p. 58 (in Greek).
- [25] Bond F., Oepen S., Nichols E., Flickinger D., Velldal E. & Haugereid P. (2011). Deep open-source machine translation. *Machine Translation*, 25: 87–105.
- [26] Carbonell J. R. & Collina A. M. (1973). Natural semantics in artificial intelligence. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, San Francisco, CA: Morgan Kaufmann, pp. 344-351.
- [27] Gonzalez M. (2015). Artificial Intelligence Semantics. *SDSU Student Research Symposium (SRS)*, San Diego State University.
- [28] Rapaport W. J. (2013). Meinongian Semantics and Artificial Intelligence. *Humana. Mente Journal of Philosophical Studies*, 25: 25-52.
- [29] Fellbaum C. (ed.) (1998). *Word Net: An Electronic Lexical Database*. Cambridge, MA: MIT Press.
- [30] Kornilakis H., Grigoriadou M., Galiotou E. & Papakitsos E. (2003). Aligning, Annotating and Lemmatizing a Corpus for the Validation of Balkan Wordnets. *Workshop on Balkan Language Resources and Tools*. Thessaloniki, November 2003.
- [31] Abedin Md J. & Purkayastha B. S. (2013). Detection of Multiword from a WordNet is Complex. *International Journal of Research in Engineering and Technology*, 02 (Special Issue: 02): 89-91.
- [32] Garidis P. K. & Deligiannakis E. N. (1993). *Dictionary of Computing (English-Greek / Greek-English)*. (6<sup>th</sup> Edition), Athens: diaulos, pp. 481-482.
- [33] Garidis P. K. & Deligiannakis E. N. (1993). *Dictionary of Computing (English-Greek / Greek-English)*. (6<sup>th</sup> Edition), Athens: diaulos, pp. 106-107.