Exploring natural language understanding in robotic interfaces

Ioannis Giachos[a,1], Evangelos C. Papakitsos[b,2,*], Georgios Chorozoglou[c,3]

[a] *"Technoglossia" Postgraduate Computational Linguistics Programme, University of Athens, Zografou 15784, Greece*
[b] *School of Pedagogical and Technological Education, ASPAITE, Iraklio Attikis 141 21, Greece*
[c] *1st Centre of Informatics and Novel Technologies of Athens, Kifisias 16, Athens 11526, Greece*
[1] *giachosg@yahoo.gr; [2] papakitsev@sch.gr*; [3] geochoro@sch.gr*
* corresponding author

ARTICLE INFO

ABSTRACT

Natural Language Understanding is a major aspect of the intelligence of robotic systems. A main goal of improving their artificial intelligence is to allow a robot to ask questions, whenever the given instructions are not complete, and also by using implicit information. These enhanced communicational abilities can be based on the voids of an output data structure that corresponds to a systemic-semantic model of language communication, as grammar formalism. In addition, the enhancing process also improves the learning abilities of a robot. Accordingly, the presented herein experimental project was conducted by using a simulated (by a plain PC) robot and a simple constructed language that facilitated semantic orientation.

I. Introduction

The enhancement of the capability of communication between humans and robots is an extremely complex task. It requires the understanding of natural language by robots, which is a function that has to be carried out in all linguistic-processing levels: the phonetic/phonological, the morphological, the syntactic, the semantic and the pragmatic level [1], [2]. In order to simulate the function of a natural language by a robot, a way of simplification is to select a level of language processing, which is the research target, and reduce the rest of the levels to a minimum or exclude them, if possible. Accordingly, this work focuses on presenting methods and algorithms related to a simplified natural language understanding (NLU) by simulated robots that includes mainly the semantic and secondary the pragmatic level of processing. For this purpose, a constructed language [3] that simulates natural language has been selected (i.e., SostiMatiko), with a minimized morphological and syntactic level (see section: II. Related Works). The phonetic-phonological level has been excluded, since the communicating interaction has been conducted through the input/output devices of a PC (keyboard & screen). Specifically, a systemic communication model has been used, as semantic grammar formalism, which will be presented subsequently (see section: IV. OMAS-III).

II. Related Works

Similar projects have been conducted in the past, the most notable being ROILA [4], which focuses on the phonetic/phonological level. The ROILA project (RObot Interaction LAnguage) is an international open work in progress that aims at developing a language exclusively for robots. Robots often misinterpret words or cannot understand what it is said, because the current speech-recognition technology has not yet reached a satisfactory level of phonetic comprehension. So the easiest way is to construct a new language that addresses the associated problems of speech that the usage of a natural language introduces. Thus ROILA has a simple grammar without irregularities and its words are composed of phonemes (sounds) that are common among the majority of natural languages. Moreover, a genetic algorithm creates new words in a way that is easy to pronounce, while ensuring that each of these words sounds differently from others, as far as possible. This helps the robot's voice-recognition system to understand accurately the human speaker. ROILA

incorporates all the vocabulary of the artificial language Toki Pona [5], which was designed by the translator and linguist Sonja Lang. It is a minimalistic language that includes only 14 phonemes and 120 root-words. In the vocabulary of ROILA there are 850 words in total.

SostiMatiko [6] is also an artificial language of Greek origin, designed by the philologist-linguist Ioannis K. Kenanidis (completed at the end of 2013), with words that are derived from Greek roots. It is a very easy language to learn and through this it can express by combination a large number of concepts. The vocabulary consists of 222 words, but it has unlimited possibilities in the production of new words in a simple way. It features an extremely simplified grammar. It was chosen to replace ROILA, for its two main attributes: simplicity (smaller vocabulary and grammar) and the Greek origin of the vocabulary, since this research project has been implemented at two Greek universities [7].

We can describe a complex procedure to a robot or have a simple conversation with it through SostiMatiko. The achievement of such an objective, having also a semantic enhancement of our communication with the robot, requires the usage of a semantic grammar formalism for the designing of the related procedures. The major semantic formalisms are: the semantic grammar (CF-grammar), the Case Grammars (decision tree describing semantic relationships), the feature-based grammars (group of unifying formalisms) [8]; Conceptual Dependency (CD) [9], [10]; and the Constraint-based grammars (environment-dependent rules) [11], [12]. In these formalisms, the linguistic relations between verbs and their arguments are declared and decision trees for semantic analysis are produced, as appropriate.

III. Research Objective

The current research objective was to investigate the ability of a robot to understand commands by receiving simple sentences and being fully aware of the words included in them. The demonstration of artificial intelligence (AI) through language by a robot is not only accomplished by the recording of static linguistic structures and relations, as they are expressed in the previous semantic grammar formalisms, but mainly by the ability of the robot to ask questions. This ability was a major research challenge in this work as a key criterion of language comprehension (NLU) by the robot. Moreover, in order to reduce the conceptual complexity of the development process, it was desirable to use a unified conceptual framework both for the software development process and for the description of the semantic grammar. For this reason, a systemic analysis method was chosen to be tested as a semantic grammar formalism for the designing of the communication software. This method is the 3rd version of the Organizational Method for Analyzing Systems (OMAS-III).

IV. OMAS-III

OMAS-III [13] is a designing evolution of the family of SADT [14] and IDEFx [15] techniques, which are well-established analysis techniques for Information Systems that are compatible to the General Systems Model (GSM) [16]. Moreover it is semantically augmented to become also compatible with similar established models that describe human communication from a social point of view [17], [18].

According to OMAS-III, the full understanding of a system requires answers to the seven key questions (*journalist's questions*) that are related to:

- the causality of the system (*Why?*);
- the output, including feedback (*What?*);
- the input, including feedback (*Which?*)
- the functioning conditions/regulations (*How?*);
- the monitors that guide its operation (*Who?*);
- the spatial aspects of functionality (*Where?*);
- the temporal aspects of functionality (*When?*).

It will be demonstrated that the implementation of OMAS-III as a semantic grammar formalism enhanced the communication abilities of the robot and it was discovered that it can acquire knowledge too. The reason for this outcome is because the particular formalism has been built on the seven fundamental questions, the same ones that people use in their daily lives. When somebody knows what should be done, by whom, why, how, when, where and which available resources with, then s/he just knows well. If a robot stores this knowledge and combines it with people and places then a big step towards NLU and AI is accomplished.

V. Software Design

The software design aims at matching the elements of grammar/syntax of each incoming sentence (input) to the seven questions of OMAS-III. The robot should take all the information that can be provided by answering these questions through the processed input. If this is not possible, then it externalizes questions accordingly. The output of the system is not a new sentence but the input data organized in such an order that will allow the robot to do what is requested by the sentence. Namely, the processing creates a temporary one-dimensional table, in which the data of a sentence provide a sequence that dictates: who will do what action, how, with what intensity or duration, when and where. In case of many input sentences, when the repeating process is completed, then the data are given to the outside world. Sentences lacking a subject and/or an object are not discarded but instead they trigger a clarification procedure. For example, in the case of an Imperative, the word "come" means "you come here" but the word "go" means "you go somewhere". This "somewhere" is not meant and the robot will have to look for it. If no answer is found in the existing data, then a question is submitted by the robot. It is necessary to use default values, which will mark the position of implied grammatical elements. So through this way of recursion and completion of each sentence, the temporary table is finalized and the complete data are transferred to the final list of actions, which are placed in the correct temporal order.

A. Matching Grammar Elements to Questions

The question *What?* is the output of the model. The answer is: "what the verb says". That is the action to be taken, either in active or passive voice or in Imperative. The robot detects the verb in the incoming sentence. If it does not find a verb in the current sentence but finds a subject or an object, then it must submit a question about what action is supposed to do.

In the question *Which?* every object of the sentences is allocated, along with all quantitative determinations and generally anything that holds the object's position and is not an adverb or determination of manner, time or space. Objects are semantically placed in this position, although they may not have any relationship with any quantity.

The question *How?* indicates the manner that the action of the verb will be taken. It is answered by the adverbs of manner, present participles, infinitives as adverbials and any other determination indicating manner. If the manner is not given, the robot searches it by asking.

The question *Who?* corresponds to the subject of the sentence. The subject may be incorporated into the verb or even omitted, especially in highly inflected languages (like Greek). If none of the above cases apply, then the robot submit the question "who (?)"! In addition to the subject's identification process, there is also some information gathering about it.

The question *Where?* defines the place of the activity. When not specified, then it is understood as the current location of the robot. The place may be set by a previous or a next sentence of the current text. It can be also defined in the current sentence, either as an absolute position or as a range "from-to", with or without a movement command. If by interpreting the verb the machine concludes that it is a static point, then the beginning and end coincide. If the starting point is not specified, then the machine will take the destination of the previous action as the new beginning. If we do not give any points or elements of movement but request it to be positioned in a point where someone else, known to the robot, is placed, then it has the corresponding information and will use it. If the location is required and not given or cannot be determined, then the robot asks for it ("where?").

The question *When?* determines the time or the moment of the action. It can be the tense of the verb (this case concerns the broadest sense of time: present, future and past), a temporal adverb or a temporal determining phrase. The time can be made more specific by determining the hour or the day. By integrating a RTC (Real Time Clock) to a robot, along with the associated software for

time-handling, can make it experience virtually every moment. The robot itself would require absolute time values more often than it was meant to. For the time being, time is tested and given hourly. Unless stated explicitly by the incoming data, the robot approximates time through the verb tense, as best as it can.

The question *Why?* represents mainly a slot for the causal and explanatory conjunctions like "because". Likewise "to", as a conjunction, may indicate an explanation or justification. In any case we are talking about secondary sentences, where a recursive procedure will be applied to, as well. The answer is a complementary sentence introduced by an explanatory-causative conjunction. It is not always given but the robot must be able to recognize and accept it if so.

B. The Specifications of Processing

In general, the computational functions of this robotic system should allow it to:

- manipulate incoming data in the form of a sentence or set of sentences;

- make clarification questions, whenever necessary and feasible;

- present a list of actions placed in chronological order and followed by full details of "who", "where", "how" and "when".

Each word is analyzed and the results are inserted in a temporary table. The incoming data pass a six-options filter:

- If it is verb (responds to "What").

- If it is an explanation (responds to "Why").

- If it is an adverb or another determination of manner (responds to "How").

- If it is an object or another quantitative determination (responds to "Which").

- If it is a subject (responds to "Who").

- If place or time is explicitly stated (responds to "Where" or "When").

If we refer to an explanation or an indication of supplementary sentence, then this information triggers a corresponding process in order to receive this new sentence, word for word. Place and time do not concern the structure of the sentence, but the level of executing the ordered action, in relation to where and when. They are both essential even in vague situations and thus they can be determined by explicit statements or from default values that will be later added in the final output-line. Additional details of the process are the following:

- If a sentence doesn't conform to the grammatical rules, according to the syntax of the language, then a relevant message is forwarded to the outside world.

- For filling voids in a sentence (such as a subject), a two-way communication with the outside world takes place.

- Cases of structure voids are externalized, so as to be decided whether to terminate or continue the syntax process.

If there are missing terms, then a structure procedure for searching data is updated. This is repeated until the answer is negative, so as to permit the placing of the output-line in a two-dimensional output-table. An example of application is demonstrated after the next section.

VI. Implementation

The system's software is encoded with the ECLIPSE platform [19], which is the most popular development package for Java and is free to download and use. Java is a high-level object-oriented programming language and runs on all computer systems, regardless of operating system.

The code of our program is implemented through a series of Classes organized and grouped according to the type of operation that are called to serve. Each class is divided into individual procedures, which are independent small units that undertake an operating section of the code. Each procedure calls and answers calls from other similar procedures, under the continuous monitoring of

the basic procedure "main". "Main" is read by the compiler when the code is initially run and the recursion of each sentence is implemented. The software includes an Interface, while the basic algorithm is shown in Fig. 1. The presented elements will be briefly described to exhibit how the code handles a sentence, how it analyzes the incoming words and how it forms the final output-line/command to the robot.
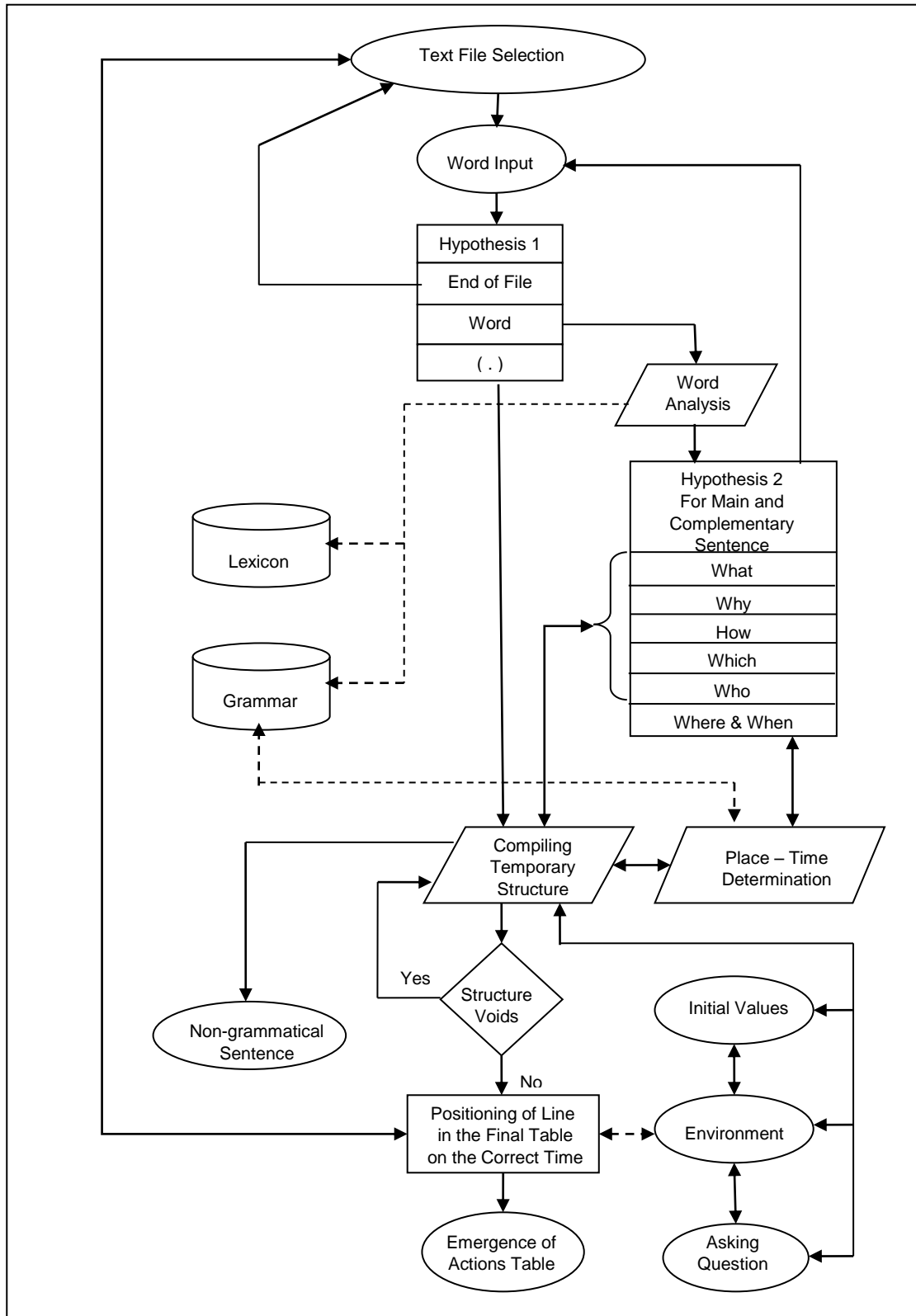

Fig. 1. The basic algorithm.

A. *Text File Selection*

The software package includes text-files containing sentences in SostiMatiko. This procedure is responsible for the opening of selected files and data mining. It receives a report from procedure *Hypothesis 1* and interacts with procedure *Positioning of Line* (*Positioning of Line in the Final Table on the Correct Time*). When *Hypothesis 1* reports that there is *End-of-File* then a request to *Positioning of Line* is sent. When this procedure completes all its work, as well, it returns a confirmation to allow the closure of the current file and a new one opens, if desired.

B. *Word Input*

Each word is accompanied by a space character or a space & dot. This procedure reads the next word in the file or the dot or the end of file. The result is sent in Hypothesis 1 (a case-structure).

C. *Hypothesis 1*

The incoming string passes through a three-option filter:

- If it is a word, then it is sent for analysis (*Word Analysis*).

- If it is a dot, then the procedure *Compiling Temporary Structure* is informed that the sentence was completed.

- If it is the end of file, then the control returns to procedure *Text File Selection* to close the file and open a new one, if desired. To terminate the current file, a report must be also received from procedure *Positioning of Line in the Final Table on the Correct Time*.

D. *Word Analysis*

Here the word tagging is conducted and the pointers of various word-categories are assigned. For example, be it a verb, a pronoun that simultaneously states the subject, passive voice, plural, etc. Tagging is achieved through a constant communication with the *Grammar* and *Lexicon* external databases that are included in the system. The former contains all the necessary grammatical knowledge of SostiMatiko, while the latter contains the whole dictionary of SostiMatiko. The result of this procedure is a one-dimensional table with the processed word and a series of false or true values, which represent the corresponding categories that any word can belong to. This table is sent to procedure *Hypothesis 2*.

E. *Hypothesis 2*

The incoming table passes a six-option filter that classifies the processed word according to the relevant respond to the suitable question (as described in subsection: *The specifications of processing*). Please note that if we may have an explanation or an indication of a supplementary sentence, then this information enables *Compiling Temporary Structure* to receive this new sentence word by word, through the same case (*Hypothesis 2*). *Hypothesis 2* also informs procedure *Word Input* to proceed to a new reading. There is also a two-way communication with procedure *Place-Time Determination* in matters related with the corresponding filter (*Where & When*).

F. *Place-Time Determination*

This procedure exchanges information with *Hypothesis 2* and communicates with the *Grammar* database to form the temporal and local context of the conducted event. If the result is not desired, then the final formulation will be achieved by exchanging information with procedure *Compiling Temporary Structure*, which handles the issues of environment initial values and questions externalization. The procedure also takes care to mark a string for time-level. This sequence corresponds to a range from a minimum to a maximum value of time, regarding the event. Thus, every event will bear an attribute denoting its correct time position in the output table, among a series of other events.

G. *Compiling Temporary Structure*

This is the most important procedure of the code as it is the most pivotal point for many other procedures. Besides its functions that have been already presented (*Hypothesis 1 & 2, Place-Time Determination*), additional details of the procedure include the following:

- If a sentence does not meet the features of grammar rules (*Non-grammatical Sentence*) according to the syntax of the language, then a relevant error message is forwarded to the outside world.

- To complement any deficiencies in the structure of the sentence (such as the absence of an explicit subject), a two-way communication is conducted with the auxiliary procedures *Initial Values*, *Environment* or *Asking Question*. These three procedures can be called either directly or one through another, from the current procedure (see subsection: *Auxiliary Procedures*).

- The externalization of *Structure Voids* may cause the syntax process to terminate or continue. Here it is checked whether the temporary structure of the output data-line is complete or has deficiencies. If there are voids (it answers *Yes*), then the procedure for searching missing data is activated. This action is repeated until the answer is negative (lack of data) in order to allow the assignment of the data-line in an output table. Care must be taken to avoid endless loops.

Finally, this procedure creates a temporary one-dimensional table, which registers the data of a sentence to provide a data line that determines: who will do what (action), why (maybe), how (intensively too), where (from-to), when (and for how long). It is called a data line because the result is not a new sentence but output data that will dictate to the robot what is requested by the sentence.

H. Auxiliary Procedures

Procedure *Initial values* returns default or zero values for gaps in the sentences that are implicitly meant though by the grammar. Besides filling gaps, this procedure also interacts with the procedure *Environment* to forward a question that was not answered or to accept an answer in return to a question that was made.

Procedure *Environment* obtains information about the wider environment of the conducted events in the current text. It aims at managing queries and existing knowledge to fill gaps that exist in data lines.

Procedure *Asking Question* externalizes questions to the outside world. These questions arise from gaps that cannot be filled by the grammar or the context. The response returns according to the manner of request and exactly where it came from.

I. Positioning of Line in the Final Table on the Correct Time

Whenever a new data-line arrives here is placed at the bottom of the two-dimensional output table. The arrangement of lines in accordance with the timing of events follows immediately. The data generated by the procedure serve as the basis of information for the system's environment in the current text. When all the sentences have been completed and by interacting with procedure *Text File Selection*, this procedure provides the data to the outside world and at the same time gives the permission of opening a new file.

J. Emergence of Actions Table

This is the final output of the system. Specifically, it is a two-dimensional table with the events listed in the input text. The table appears at the outside world, signaling the actions required by our robotic system.

VII. An application

In this section, it will be briefly demonstrated how the incoming words in a sentence are analyzed. The recursion process with the final configuration will be presented and the resulted action plan for the robot. The input sentence is a command in SistiMatiko:

```
"ksepa  podas  saxino semo  edno duo ameros"
```

which means

```
"return to the red point (by) walking (after) three days".
```

The entire sentence will go through partitioning and so a word table will occur. The system detects the number "edno duo", which becomes the word "edno_duo". The number is inserted to the knowledge base, meaning "3". Each word is sequentially analyzed. The analysis algorithm [20] deconstructs every word in its individual parts:

a) It looks up the root of the first word ("ksepa") in the dictionary. A correct word must have correct both the root and the accompanying affixes (if any) of all possible combinations that may occur (Fig. 2).

```
214 kse-pa: the root "pa" is found in position 214;

216 ksep-a: the root "a" is found in position 216;

217 ks-e-pa: the root "pa" is found in position 217;
```

Fig. 2. Possible combination roots and the accompanying affixes

Fig. 2 shows that we have the correct combination of the prefix "kse" and the root "pa" in position {214}. So based on the grammar we have the imperative of the verb "return".

b) In a similar manner for the next word ("podas"), we have the root "pod" (='foot'), the suffix "a" (denotes manner) and the ending "s" (plural number), meaning "on foot/(by) walking".
c) The word "saxino" is analyzed as "sax (root: 'blood') – in (suffix: color indicator) – o (ending: noun or adjective indicator)", meaning "red".
d) The word "semo" is analyzed as "sem (root: 'point') – o (ending: noun or adjective indicator)", meaning "point".
e) The last word "ameros" means "days".
f) It proceeds in recursion. It doesn't find a subject, but since the verb is in imperative (a) the subject is "you". There is no object either, but it is not required since the verb is not a transitive one in SostiMatiko. The robot cannot find a starting point, therefore its current position is taken as the initial. The destination is the "red point". The command will be executed after three days. The question "Why" is not answered, because the system is not authorized to ask!
g) The output list is created (Fig. 3).

```
NumOfSentence = 0;
Who = tewo = "you";
What = ksepa = "return";
How = podas = "(by) walking";
HowMuch = = null;
Why = = null;
Where from = BASE = current position;
Where to = saxino semo = "red point";
When = edno_duo ameros = "(after) 3 days";
Tense = Present;
TimeCategory = 2;
TimeSubCategory = 2;
AbsoluteTime = 2200048;
```

Fig. 3. Output list an application

The process is completed and the result is externalized by using only the necessary data from the above list. Any further voids on the right of the output list will trigger a question by the robot.

VIII. Commentary & Conclusions

The issue of interaction between a human and a robot is practically a problem of communication, which is determined by the level of NLU on behalf of the robot. The present study attempted to explore the conditions of an actual communication and knowledge acquisition by an intelligent robot. The primary realization of such a communication has been designed according to OMAS-III formalism, which allows the organizing of textual information in terms of slots that semantically correspond to the fundamental questions of humans. Any void in these slots can trigger a simple human-robotic conversation that will eventually fill an empty slot.

During processing, some general problems appeared, such as engaging in endless processes in the recursion of syntactically incorrect sentences, as well as issues of monitoring the whole system. The former have been corrected under various conditions, while the latter could not be entirely overcome. Processing problems can be encountered in the following three distinct main categories:

- Morphology problems regarding the ability of the system to analyze very composite words in simpler meanings. The number of possible affixes that can be found in a word is rather limited compared to the possibilities of SostiMatiko.

- Syntax problems because there are many ways to express a sentence syntactically but only one to be interpreted correctly. A set of rules, depending on the circumstances, should run to avoid the confusion regarding parts of speech. SostiMatiko has a dictionary where each word can express many parts of speech (namely, both a noun and a verb), a linguistic phenomenon that is found in many natural languages. This creates confusion to the system where many indicators must be used to avoid it. The problem can be overcome by following a strict sequence of syntax, like the Subject-Verb-Object one, combined with indicators for the definition of space and time.

- Semantic problems because the dictionary of SostiMatiko is minimal (222 words). Most words have to be expressed in a composite manner. For example, the word "wheel", which is not explicitly registered, could be expressed as "roundfoot" to resolve the semantic problem, since the words "round" and "foot" are explicit dictionary entries. Ultimately the problem is solved by entering into a knowledge base the new compound words, which will provide an interpretation of natural language.

In conclusion, the system analyzes all the data and then takes actions (such as time determination) and questioning. Data guide the activation of the corresponding indicators, so that their combination makes sense and gives the feeling of a conversation. The transmitted information to the robot can be used either as knowledge or as commands to be executed. Despite the encountered problems, the results surpassed our initial expectations, because it was demonstrated that OMAS-III as semantic formalism can form the backbone of an intelligent robotic system.

References

[1] A. Akmajian, R.A. Demers, A.K. Farmer, and R.M. Harnish, "Linguistics," 6th ed., The MIT Press, 2010.

[2] A. Cruse, "Meaning and Language: An introduction to Semantics and Pragmatics," Chapter 1, Oxford Textbooks in Linguistics, 2004.

[3] E.C. Papakitsos and I. Giachos, "The Study of Machine Translation Aspects Through Constructed Languages," International Journal of Electronic Engineering and Computer Science, Vol. 1, No. 1, pp. 28-34, 2016.

[4] http://roila.org

[5] http://tokipona.org/

[6] www.unilang.org/blog/SostiMatiko/

[7] I. Giachos, "Implementation of OMAS-III as a Grammar Formalism for Robotic Applications," Postgraduate Dissertation, Interdisciplinary Postgraduate Course "Technoglossia" (MSc). National & Kapodistrian University of Athens, Dept. of Philology and National Technical University of Athens, School of Electrical & Computer Engineering, 2015 (in Greek).

[8] E.C. Papakitsos, "Linguistic Software Engineering: I. Preparation." Athens: National Library of Greece, 2013, pp. 92-97 (in Greek).

[9] R.C. Berwick, "Natural Language Processing," MIT, Lectures 6.863J, 2003.

[10] A. Lavie and B. Frederking, "Algorithms for NLP," Carnegie-Mellon University, Dept. LTI, Course 11-711, 2010.

[11] F. Karlsson, "Constraint Grammar as a Framework for Parsing Unrestricted Text," in Proceedings of the 13th International Conference of Computational Linguistics, Vol. 3, H. Karlgren, Ed. Helsinki, 1990, pp. 168-173.

[12] F. Karlsson, V. Atro, H. Juha, and A. Arto (Eds.), "Constraint Grammar: A Language-Independent System for Parsing Running Text," Natural Language Processing, No 4. Berlin and New York: Mouton de Gruyter, 1995.

[13] E. Papakitsos, "The Systemic Modeling via Military Practice at the Service of any Operational Planning," International Journal of Academic Research in Business and Social Science, vol. 3(9), pp.176-190, 2013.

[14] D.T. Ross, "Structured Analysis: A Language for Communicating Ideas," IEEE Trans. Software Engineering, pp. 16-34, January 1977.

[15] V. Grover and W.J. Kettinger, "Process Think: Winning Perspectives for Business Change in the Information Age," IGI Global, 2000.

[16] M. Sanders, "Communication Technology Today and Tomorrow." Glencoe/McGraw-Hill, 1991.

[17] D.H. Lasswell, "The structure and function of communication in society," in The message of media, The explosion of massive communication, K. Livieratou and T. Frangouli, Eds. Athens: Alexandria, 1991, pp. 65-83 (translated into Greek).

[18] A. Mantoglou, "Models, basic principles and skills of communication," in Counselling Horizons for School Guidance - SOS Orientation, C. Kapoli, Ed. Athens: Panteion University of Social and Political Sciences, 2007, pp. 508-530 (in Greek).

[19] https://eclipse.org/

[20] E.C. Papakitsos, "Linguistic Software Engineering: II. Realization," Athens: National Library of Greece, 2014, pp. 36-37 (in Greek).