# Logical analysis of natural language semantics to solve the problem of computer understanding

Yuriy Ostapov

*Institute of Cybernetics of NAS of Ukraine, pr. Acad. Glushkova, 40, Kiev, 03680, Ukraine. E-mail: yugo.ost@gmail.com*

**Abstract**

An object–oriented approach to create a natural language understanding system is considered. The understanding program is a formal system built on the base of predicative calculus. Horn's clauses are used as well–formed formulas. An inference is based on the principle of resolution. Sentences of natural language are represented in the view of typical predicate set. These predicates describe physical objects and processes, abstract objects, categories and semantic relations between objects. Predicates for concrete assertions are saved in a database. To describe the semantics of classes for physical objects, abstract concepts and processes, a knowledge base is applied. The proposed representation of natural language sentences is a semantic net. Nodes of such net are typical predicates. This approach is perspective as, firstly, such typification of nodes facilitates essentially forming of processing algorithms and object descriptions, secondly, the effectiveness of algorithms is increased ( particularly for the great number of nodes), thirdly, to describe the semantics of words, encyclopedic knowledge is used, and this permits essentially to extend the class of solved problems.

## 1. Introduction

The given paper is devoted to logical aspects of object–oriented approach to create a natural language understanding system. This approach was proposed in our works [11, 12, 13].

Consider the next tasks connected with computer understanding:

- what the computer understanding means;

- how to realize the computer understanding;

- how to check the accuracy of computer understanding.

An understanding of complete text is a thorough insight into the sense of assertions which are included in this text. As regards the computer understanding, it is realized with algorithms to represent the semantics of sentences and to check such representation.

Consider at first what the assertion *the sense of sentence is ...* means :

1. It should be distinguished the sense for an author of information from the sense for a reader of this information as this depends on the volume of knowledge. Ontological and epistemological representations are as well of great importance[1].

2. The sense of sentence is contained, firstly, in the meaning of words, secondly, in the functional role of these words in the sentence, thirdly, in properties of objects which are known from previous sentences [2].

3. For abstract concepts and classes of physical objects, the sense of words is concluded in the standard understanding (in explanatory dictionaries). For individual objects, the sense consists in properties of these objects. Some words have not the standard meaning (*good, beauty, love*), then it should be took into account an emotional relation of author to their using.

4. When we speak about the functional role of words in a sentence, we imply such categories as a subject, objects of influence, an action or internal change of subject, other facts of the case (place, time, cause, purpose, tool and method of action). It should be differed an object and its properties.

5. A sentence can not be considered separately, but only in the conjunction with other sentences and knowledge of object domain. The sense of sentences has as well an implicit character which is contained in actions and events following from actual information of sentences.

---

[1]For example, consider the sentence *Napoleon lost the battle of Waterloo*. The name *Napoleon* can consider as the name of historical person or the name of horse. At last, it can assume the migration of souls. Such distinction in semantic representation for the name *Napoleon* implies the distinction in the sense of sentence.

[2] For instance, a sentence describes actions of *Peter*. It is known from previous sentences that *Peter* is unmarried. This can explain his further actions.

By this means, the sense of sentences have objective and subjective aspects. It can point to three aspects [3]:

- *cognitive (propositional)* - the description of facts, as well as general assertions following from facts;

- *communicative* - the organization of text so that the accent is done on certain factors of content;

- *pragmatical (illocutive)* - the explanation of sentence pronouncing as the action that is executed with certain purpose.

We limit ourselves only to cognitive (propositional) aspect. It is necessary to extract an objective component in the sense of declarative sentences. This component presents the general content for a software engineer and an expert which checks the accuracy of understanding program work. To extract this objective component, we assume that, firstly, both are based on certain ontology and epistemology, secondly, meanings of words are understood equally (according to an explanatory dictionary), thirdly, both follow standard rules of grammar (for English see, for example, [21]).

To answer the question *How to realize the computer understanding?*, we must (as noted above) be based on certain ontology and epistemology. As such basis, we use *modern empiricism* [23]. Empiricism is founded on modern physical science concerning the structure of world and explains the reliability of scientific conclusions. We take the main thesis of empiricism: *all synthetic knowledge is based on experience.*

The world consists of *physical objects*, which correspond to certain domains of *physical place* in a given point of time. Physical objects can act or change their state. These actions of objects and processes of internal state change (events) are named *facts*. By this means, *the real world consists of facts*. As B.Russell points, *physical facts, for most part, are independent not only of our volitions, but even of our existence* [23, part 2, ch.11]. We describe both real (or possible) facts and general assertions with the help of declarative sentences. The truth of concrete sentences (for real actions and events) consists in the conformity with available facts. For our approach, the sentence of natural language is presented with a set of predicates describing *actions, events, states, processes, persons, organizations, things, machines*, and other classes of essences and connections.

To answer the question *How to check the accuracy of computer understanding?*, it should be took into account the above determination of sense. To establish the conformity of sentence sense and its semantic representation, it will be sufficient to use simple and logical questions. Simple questions permit to check the availability of all categories (subject, objects, etc) in representations of facts. Logical questions examine the ability of computer system to do inferences on the basis of present facts and knowledge.

## 2. Levels of computer understanding

Preparatory proceeding to the computer understanding, it should be specified basic principles of computer work:

1.A computer has a timepiece. This denotes that the computer *measures a physical time.*

2. A computer permits to save numbers and symbols. Another way, the computer *has a memory*. On–line storage consists of cells. Each cell can save a symbol and has a certain address. On–line storage is short–term memory. Long–term memory is realized using peripheral.

3. A computer maps points on a display. Consequently, the computer *forms points of discrete space with two coordinates.*

4. A computer executes arithmetic. Then, it can say that the computer *understands what arithmetic mean.*

5. A computer moves a symbol from one address to other. This permits *to form abstract words as sequences of symbols.*

6. A computer executes a program, which consists of a sequence of commands. Each command contains the code of operation and addresses. Commands execute arithmetic, transfer of symbols, and conditional transfer. Conditional transfer changes the sequence of command execution.

7. A computer can "see", "hear" by means of peripheral. This means that the computer *has perceptions.*

8. A computer (robot) can go in physical space and implement various physical actions using special mechanisms.

9. As an operation system realizes different actions, controls programs, remembers all actions and events, it can speak about *computer consciousness.*

The construction of computer permitted A.Turing to express the hypothesis about the ability of computer to answer questions so that an answer

of a person and the computer can not be distinguished. Another way, the computer can understand questions and adequately answer.

It can distinguish four levels of computer understanding.

At the first level we take into account only such grammatical categories as subject, objects, place and time action (event), as well as other adverbial modifiers in a sentence. This approach is realized with syntax analysis. For this level a computer system can answer simple questions. Such approach is of frequent use to translate from one language to other by means of computer.

At the second level paradigmatic and syntagmatic relations between words are established. Sentences are presented with semantic net where nodes have a simple structure. Investigations in [1, 4, 6, 9, 22, 24, 25, 26, 27] relate to this level. At the second level an understanding program can answer many questions using an inference. The understanding of semantics (for actions, events and physical objects) is not sufficient and is contained usually in inference rules.

At the third level sentences are represented with a set of predicates (the object–oriented approach)[11, 12, 13]. Properties and changes of physical and abstract object are described with typical predicates[3]. Physical, physiological, and psychological characteristics are estimated using special scales. If measuring is executed with a physical device, then the semantics of property is described by physical regularities. For psychological characteristics, it is necessary to use algorithms of recognition. To determine economical activities, special algorithms are demanded.

At third level there is the certain advance to describe actions and processes. In the work [24] the classification of action was proposed. We use it in the modified view:

*PROCESS* — long purposeful occupation (*job, sport, studies*);
*PROPEL* — applying a force to an object;
*MOVE* — movement of body part;
*INGEST* — ingesting something inside;
*EXPEL* — expelling something from a subject;
*GRASP* — grasping an object;

---

[3]Our approach is ontological. Based on functions, properties and structure of real objects, we advert to the description of words. If we shall form a special structure (frame) for every word, then we shall be at a deadlock as the variety of words is enormous. To solve this problem, we use encyclopedic knowledge.

*GO* — displacement of subject;

*TRANSFER* — change of general relation for a subject (*to buy, to sell, to come into fortune*);

*ATTEND* — perceptions of subject (*to see, to hear, to touch*);

*MESSAGE* — transmission of information between a subject and object;

*BE* — identity of subject and object, existence of subject or connection between a subject and certain class of objects;

*CHANGE* — transition of subject to another internal state (event);

*CREATE* — thinking (*decision-making, problem-solving, prediction*);

*HAVE* — possession of object.

Clearly this classification is not sufficient as it does not take account of semantics for many verbs. To describe the semantics of simple movements of hand and leg, it can use cinematic equations. The complex action (for example, *to lead an orchestra, to play tennis, to send a message*) is a sequence of simple movements and described with *verbal definition* and *operation* (see further). T.Winograd was a pioneer in this direction. In the work [30] virtual movements of robot are realized by certain programs.

To represent concrete and general sentences (for a given object domain), we apply a semantic net. Nodes of such net are typical predicates. This approach is perspective ( particularly for the great number of nodes). Firstly, such typification of nodes simplifies greatly forming of processing algorithms and object descriptions, secondly, the effectiveness of algorithms is increased, thirdly, we apply encyclopedic knowledge to describe the semantics of words, and this extends essentially the class of solved problems.

At the fourth level the object–oriented approach is connected with functions of perception and physical movement. To do this, it is necessary to create a robot which has organs of sense, executes physical measurements, as well as realizes physical movements. The availability of recognition algorithms permits to teach such robot the cognizance of physical objects and determination of their properties. Then it can use *visual determinations* to teach basic words [23, part II-ch.2]. Other words are determined *verbally*. Besides, such robot can implement certain physical actions. This speaks about more deep understanding of these actions.

Searle's point is that *the mere carrying out of a successful algorithm does not in itself imply that any understanding has taken place* [16, ch.1]. However, at the fourth level a robot and a person overcome the same path to the understanding of that how to represent physical objects and actions (events)

associated with words (using the realization of physical actions and recognition algorithms of objects). Besides, the robot recognizes words of certain language and discovers their grammatical role in a sentence, takes account of sense relations between words, checks the accuracy of word combinations. Consequently, at the fourth level there is much common in computer and human understanding although physical principles of work are different. No all actions of robot can be described formally, that is, it can not be spoken that the robot works only according to strict algorithms.

Thus, the fourth level permits essentially to approximate to the human understanding. However, there are open problems connected with the understanding of such abstract concepts as *good, beauty, love, truth*, etc. At the moment it is not clear how to do this. Some questions can be solved now. For example, the program of understanding can be taught to create new classes using algorithms of classification. We can form new properties by means of analogous algorithms. Induction and analogy can be used to solve creative tasks [17].

## 3. Typical predicates to represent sentences

To represent concrete and general sentences, it is necessary to use the next predicates for:

- physical objects (persons, things, machines, animals, natural phenomena);

- actions and processes connected with physical objects;

- events and states for physical objects;

- categories connected with objects, actions and processes (time, place, property);

- abstract objects;

- sense relations between actions, processes, objects, events and states.

Physical objects are described with predicates of type $\psi(u_1, u_2, ..., u_n)$ where $u_1, u_2, ..., u_n$ are variables providing, in the aggregate, the identification of object. For example, it is necessary to use as such variables for persons: *first* and *second name, sex, age, place of birth*, etc. A concrete

7

object is presented by means of substitution of individual constants instead variables. Each real object is characterized by a compact connected domain of *physical space* at a given point of time. Points of this domain have coordinates $(x_1, x_2, x_3)$, as well as physical parameters (mass, charge, temperature, etc)[2, App.A]. The predicate $\psi$ is true during a certain interval of time if a real object exists in this interval.

Categories connected with objects, actions and processes are presented with predicates of type $\delta(y_1, y_2, ..., y_n)$ where $y_1, y_2, ..., y_n$ are variables providing, in the aggregate, the identification of given category. As categories, we consider a time, place, and property of object. Concrete category is formed by means of substitution of individual constants instead variables. Every real time is characterized by an interval of *physical time*, the place corresponds to a domain of *physical space* with coordinates $(x_1, x_2, x_3)$. The property of physical object is a characteristic conditioned by the physical and psychological regularities.

Sense relations between objects are described with predicates of type $\mu(q_1, q_2, ..., q_n)$ where $q_1, q_2, ..., q_n$ are variables. Abstract objects correspond to predicates of type $\rho(r_1, r_2, ..., r_n)$ where $r_1, r_2, ..., r_n$ are variables. A concrete sense relation (abstract object) is formed using the substitution of individual constants instead variables.

Actions (processes) connected with physical objects are presented with predicates of type $\varphi(v_1, v_2, ..., v_n)$ where $v_1, v_2, ..., v_n$ are variables providing, in the aggregate, the identification of given action or process. As such variables, we use a subject, objects of influence, time and place of action, as well as other facts of the case. As these factors correspond to other predicates, to point to them, we use unique codes appointed these predicates. A concrete action (process) is described using the substitution of individual constants instead variables. A real action (process) is characterized by a set of compact connected domains in *a physical space–time*[4]. Each such domain consists of points with coordinates $(x_1, x_2, x_3, t)$ and physical parameters. The predicate $\varphi$ is true in a certain interval of time if a set of domains corresponds to a real (i.e., checked) fact. A place of action (process) embraces domains of subject, objects of influence in physical space with coordinates $(x_1, x_2, x_3)$ during a given interval of time.

---

[4]Hereafter we are dealing with classical (Galilean) space–time[16, ch.5]. For the object domain connected with social system, it is sufficient.

Events (states) connected with physical objects are presented with predicates of type $\gamma(w_1, w_2, ..., w_n)$ where $w_1, w_2, ..., w_n$ are variables providing, in the aggregate, the identification of this event and state. As such variables, we use codes of subject, objects of influence, time and place, etc. A concrete event (state) is described using the substitution of individual constants instead variables. Each real event (state) is characterized by a set of compact connected domains in *a physical space–time*. Each domain contains points with coordinates $(x_1, x_2, x_3, t)$ and physical parameters. The predicate $\gamma$ is true in a certain interval of time if a set of domains corresponds to a real fact. A place of event (state) embraces domains of subject, objects of influence in physical space with coordinates $(x_1, x_2, x_3)$ during a given interval of time.

Go on now to the detailed description of above types of predicates. Only after that we can consider the formal representation for concrete and general sentences. Values of variables, as a rule, correspond to names of variables, otherwise these values are explained. If we speak about the type of something, this means that values of variables are divided into several classes. As an illustration, we can point to some names of classes. Codes of objects, action, events and states included in descriptions of predicates pose references to these objects, actions, events and states. These references present semantic relations between objects, actions, events and states.

*3.1. Typical predicates of physical objects*

Predicates of this type are divided into the next sorts: *person, organization, thing, machine, animal, nature.* The variable *number* in these predicates denotes the grammatical number of object in a sentence: plural or singular.

*3.1.1. Description of persons*

A corresponding predicate takes the form:

*person (cod_pers, sex, age, number, first_name, second_name, cod_pl_birth, cod_dt_birth, nath, lang, face, nose, eyes, hair, stature, prof)*

where *cod_pers* is a code of person, *cod_pl_birth* is a code of birth place, *cod_dt_birth* is a code of birth date, *nath* is a nationality, *lang* is a mother tongue, *prof* is a profession.

*3.1.2. Description of organizations*

A predicate of organization is of the form:

*organization (cod_org, name_org, typ_org , number, cod_loc_org, director)*

where *cod_org* is a code of organization, *name_org* is a name of organization, *typ_org* is a type of organization, *cod_loc_org* is a code of organization location.

*3.1.3. Description of things*
    A corresponding predicate is of the form:

*thing (cod_th, name, maj_class, number, weight, color, length, height, thickness, cod_owner)*

where *cod_th* is a code of thing, *maj_class* is a name of major class.

*3.1.4. Description of machines*
    A machine differs from a thing by the availability of energy source and transformation of one energy to other. A predicate of machine takes the form:

*machine (cod_mach, name, function, number, typ_eng, color, trademark, name_prod, cod_owner)*

where *cod_mach* is a code of machine, *typ_eng* is a type of engine, *name_prod* is a name of producer.

*3.1.5. Description of animals*
    A predicate of animal takes the form:

*animal (cod_an, typ_an, number, maj_class, weight, name, color, cod_owner)*

where *cod_an* is a code of animal, *typ_an* is a type of animal, *maj_class* is a name of major class.

*3.1.6. Description of natural phenomena*
    A corresponding predicate is of the form:

*nature(cod_nat, typ_nat, number, name, charact)*

where *cod_nat* is a code of phenomenon, *typ_nat* is a type of phenomenon, *charact* is a characteristic of phenomenon.

### 3.2. Typical predicates of categories

Predicates of this type are divided into the next sorts: *place, time, property.*

### 3.2.1. Description of place

A predicate of place takes the form:

*place (cod_pl, country, typ_reg, name_reg, ter_entity, name_ter, locat, name_loc, constr, name_constr, add_inf_constr, fin_locat, name_room)*

where *cod_pl* is a code of place, *typ_reg* is a type of region, *name_reg* is a name of region, *ter_entity* denotes a territorial entity (*town, village*), *name_ter* is a name of territorial entity, *locat* is a location (*street, square, park, line*), *name_loc* is a name of location, *constr* is a construction (*house, theatre, station, industrial object*), *name_constr* is a name of construction, *add_inf_constr* is additional information for construction (*stair, roof, garret, floor*), *fin_locat* is the designation of final location (*apartment, hall, library, office, restaurant, cafe*), *name_room* is the designation of room (*living room, kitchen, bathroom, bedroom*).

A number of *house, floor* and *apartment* is realized by the predicate *number*.

### 3.2.2. Description of time

A predicate of time takes the form:

*time (cod_tm, year, season, month, numb_mon, day_week, holyday, part_day, hours, minutes)*

where *cod_tm* is a code of time, *numb_mon* is a date.

### 3.2.3. Description of property

A predicate of property is of the form:

*property (cod_prop, name, scale, state, cod_obj)*

where *cod_prop* is a code of property, *scale* is a name of scale, *state* is a state according to this scale, *cod_obj* is a code of object to which this property belongs.

The estimation of properties is realized by means of scales[24, sec.3-5]. Scales can measure:

- geometric parameters (*length, volume, area*);

- physical values (*weight, temperature, charge, velocity, pressure*);

- physical states (*hard, liquid, green, blue, cold, hot*);

- psychological states (*joy, distress, anger, fear*);

- physiological characteristics (*roof pressure, pulse*);

- economical activities (*income, expenses, profit*);

*3.3. Typical predicates of sense relations*

Predicates of this type are divided into the next sorts: *cause, relation, link.*

*3.3.1. Description of cause*

We consider causal dependence as physical or physiological regularity, but no the result of simple induction [23, part 6, ch.5]. Besides, there is an inference as the kind of cause-and-effect relations.

A predicate of cause takes the form:

*cause (cod_cs, typ_cs, typ_sit, cod_cause, cod_res)*

where *cod_cs* is a code of cause, *typ_cs* is a type of cause (*motive, objective causation, inference*), *typ_sit* is a type of situation (*event, action, thought, message*), *cod_cause* is a code of cause, *cod_res* is a code of result.

*3.3.2. Description of sense relations*

The main role of predicate for sense relations is to map paradigmatic relations [3]. This predicate is of the form:

*relation (cod_rel, typ_rel, emot_estim, first_obj, sec_obj, cod_first, cod_sec)*

where *cod_rel* is a code of relation, *typ_rel* is a type of relation (*sexual, familiar, official, possession of something*), *emot_estim* is an emotional estimation (*passion, love, animosity*), *first_obj* denotes the first object of relation (*father, mother, brother, husband, chief*), *sec_obj* points to the second object of

relation (*soon, daughter, sister, wife, subordinate*), *cod_first* is a code of the first object, *cod_sec* is a code of the second object.

*3.3.3. Description of connections between assertions*
 A corresponding predicate is of the form:

*link (cod_link, base_str, sub_str, cod_base, cod_sub , conj, sem_char)*

where *cod_link* is a code of connection, *base_str* is a type of basic structure, *sub_str* is a type of subordinate structure, *cod_base* is a code of basic structure, *cod_sub* is a code of subordinate structure, *conj* denotes a conjunction, *sem_char* is a semantic characteristic of subordinate structure (*place, time, cause, purpose, condition, method*).

*3.4. Typical predicates of abstract objects*
 Predicates of this type are divided into the next sorts: *abstr, number*.

*3.4.1. Description of abstract objects of general kind*
 A corresponding predicate takes the form:

*abstr (cod_ab, concept, domain, ad_prop, cod_owner)*

where *cod_ab* is a code of abstract object, *concept* denotes an appropriate concept, *domain* points to area of expertise ( *physics, mathematics, economics, sociology, control*), *ad_prop* is additional information, *cod_owner* is a code of owner (for *stock, deposits*) or author (for *articles, books, patents*).

*3.4.2. Description of numbers*
 A predicate of number takes the form:

*number (cod_numb, descr_word, numb, cod_obj)*

where *cod_numb* is a code of number, *descr_word* is a phrase describing this number, *numb* presents a decimal number, *cod_obj* is a code of object connected with the given number.

*3.5. Typical predicates of actions and processes*
 Predicates of this type are divided into the next sorts: *action, process, thought, message.*

*3.5.1. Description of actions*

Actions of persons, organizations, machines, animals, natural phenomena are described using the predicate *action*. Such actions are usually localized in a time and space. This predicate takes the form:

*action (cod_act, sem_typ_act, sort_act, neg_act, tense, char_act, adverb, word, cod_sub, cod_obj, cod_from_obj, cod_to_obj, scale, res_state, cod_time, cod_loc, cod_way, cod_purp, cod_cause)*

where *cod_act* is a code of action, *sem_typ_act* is a semantic type of action (PROPEL, MOVE, GO, etc), *sort_act* is a kind of action (*real, possible, necessary*), *neg_act* denotes the negation of action, *tense* is a grammatical tense, *char_act* is a character of action (*complete, incomplete*), *adverb* is an adverb which is used to estimate this action, *word* is a verb describing this action, *cod_sub* is a code of action subject, *cod_obj* is a code of influence object, *cod_from_obj* is a code of object from which the action is transferred, *cod_to_obj* is a code of object to which the action is directed, *scale* denotes a scale (see above), *res_state* is a state of subject or object (pointed by a value of scale) as a result of action, *cod_time* is a code of action time, *cod_loc* is a code of action place, *cod_way* is a code of action way, *cod_purp* is a code of action purpose, *cod_cause* is a code of action cause.

*3.5.2. Description of processes*

Long procedures and other occupations of persons and organizations are described by means of the predicate *process* . Such processes contain usually many actions. This predicate is of the form:

*process (cod_pr, sort_pr, typ_pr, neg_pr, tense, char_pr, adverb, word, cod_sub, cod_obj, cod_start_pr, cod_end_pr, cod_start_loc, cod_end_loc, cod_way, cod_purp, cod_res)*

where *cod_pr* is a code of process, *sort_pr* is a kind of process (*real, possible, necessary*), *typ_pr* is a type of process (*job, teaching, sport, hobby*), *neg_pr* denotes the negation of process, *tense* is a grammatical tense, *char_pr* is a character of process (*complete, incomplete*), *adverb* is an adverb which is used to estimate the process, *word* is a verb describing this process, *cod_sub* is a code of process subject, *cod_obj* is a code of influence object, *cod_start_pr* is a code of start time, *cod_end_pr* is a code of final time, *cod_start_loc* is a code of

start place, *cod_end_loc* is a code of final place, *cod_way* is a code of process way, *cod_purp* is a code of process purpose, *cod_res* is a code of process result.

### *3.5.3. Description of thoughts*

Perceptions and thoughts of persons shaped by assertions are described using the predicate *thought*. This predicate takes the form:

*thought (cod_th, sort_th, neg_th, tense, char_th, adverb, word, cod_sub, cod_obj, cod_time, cod_loc, cod_purp)*

where *cod_th* is a code of thought, *sort_th* is a kind of thought (*real, possible, necessary*), *neg_th* denotes the negation of thought, *tense* is a grammatical tense, *char_act* is a character of thought (*complete, incomplete*), *adverb* is an adverb which is used to estimate the thought, *word* is a verb describing this thought, *cod_sub* is a code of subject, *cod_obj* are codes of thought sorts (*action, object, event, state*), *cod_time* is a code of time (when this thought arose), *cod_loc* is a code of place (connected with this thought), *cod_purp* is a code of solved problem.

### *3.5.4. Description of message*

A predicate of message is of the form:

*message (cod_ms, sort_ms, neg_ms, tense, char_ms, adverb, word, cod_sub, cod_adr, theme, cod_time, cod_loc, cod_purp, cod_way, cod_cause)*

where *cod_ms* is a code of message, *sort_ms* is a kind of message (*real, possible, necessary*), *neg_ms* denotes the negation of message, *tense* is a grammatical tense, *char_ms* is a character of message (*complete, incomplete*), *adverb* is an adverb which is used to estimate the message, *word* is a verb describing this message, *cod_sub* is a code of subject, *cod_adr* is a code of addressee, *theme* are codes of message sorts (*action, object, event, state* ), *cod_time* is a code of message time, *cod_loc* is a code of message place, *cod_purp* is a code of message purpose, *cod_way* is a code of transfer way (*post, e-mail, verbally*), *cod_cause* is a code of message cause.

### *3.6. Typical predicates of states and events*

A change of *internal state* (for a physical object) is referred to as *an event*. States and events of persons, organization, machines, things, animals, natural phenomena are described using the predicate *event*. This predicate

takes the form:

*event (cod_evt, sort_evt, neg_evt, tense, char_evt, adverb, word, cod_sub, cod_obj, scale, beg_state, res_state, cod_time, cod_loc, cod_cause)*

where *cod_evt* is a code of event (state), *sort_evt* is a kind of event (state) (*real, possible, necessary*), *neg_evt* denotes the negation of event (state), *tense* is a grammatical tense, *char_evt* is a character of event (state) (*complete, incomplete*), *adverb* is an adverb which is used to estimate this event (state), *word* is a verb describing this event (state), *cod_sub* is a code of subject for this event (state), *cod_obj* is a code of influence object, *scale* denotes a scale (see above), *beg_state* is a start state (pointed by a value of scale), *res_state* is a final state, *cod_time* is a code of time (for this event or state), *cod_loc* is a code of place (for this event or state), *cod_cause* is a code of cause (for this event or state).

## 4. A database

A database contains a set of typical predicates describing facts (after the substitution of individual constants instead variables). The database includes such predicates: *person, organization, thing, machine, animal, abstr, nature, action, process, event, message, thought, property, number, place, time, cause, relation, link.* Besides, the database contains dictionaries of paradigms, verbs and nouns[5].

### 4.1. A dictionary of paradigms

To represent a dictionary of paradigms, we use the predicate:

*paradigm (cod_par, osn_form, paradigm, gram_cat, synt_char)*

where *cod_par* is a code of paradigm, *osn_form* is a basic form of word, *gram_cat* is a grammatical category, *synt_char* is a syntax characteristic.

A basic form is a basic word in dictionaries. It can be several descriptions with the same basic form to present all paradigms for all parts of speech. A grammatical category points to belonging to a certain part of speech (*noun,*

---

[5]Described further project solutions are oriented to English. For other languages, these solutions must be modified.

*verb, adverb*, etc). A syntax characteristic describes a syntax function of given word in a sentence. For example, an adverb can be the next sorts: *place, time, degree, manner.*

*4.2. A dictionary of nouns*

To represent a dictionary of nouns[6], we use the predicate:

*noun (cod_int, osn_form, gram_cat, maj_class, sem_cod, verb, scale, state, combin )*

where *cod_int* is an unique code, *osn_form* is a basic form of word, *gram_cat* is a grammatical category (noun or adjective), *maj_class* is a generic class, *sem_cod* is a semantic code, *verb* is a corresponding verb for verbal noun, *scale* denotes a scale (see above), *state* is a state pointed by a value of scale, *combin* describes the compatibility of nouns.

The variable *sem_cod* has the next values: *person, prof* (profession), *sibl* (sibling connection), *org* (organization), *anim*(animal), *plant, place, time, thing, weapon , mach* (machine), *occup* (occupation), *event, state, prop* (property), *body , natur* (natural phenomenon), *mat* (matter), *env* (environment), *psych* (psychic process or action), *mes* (message), *act* (physical action of person), *abstr* (abstract concept), *scale, cloth, food, quant* (attribute of quantity).

The description of compatibility for nouns connected by means of prepositions permits to point to admissible combinations of preposition and semantic code (for example, for the word *book*: *on_thing, on_abstr, of_person, of_org*).

*4.3. A dictionary of verbs*

The predicate of this dictionary is of the form:

*verb (cod_verb, inf, after_verb, sem_typ, scale, beg_state, end_state, subj, contr_at, contr_from, contr_to, contr_with)*

where *cod_verb* is a code of verb, *inf* is a basic form (Infinitive), *after_verb* is an adverbial particle, *sem_typ* is a semantic type of verb (PROPEL, MOVE, GO, etc), *beg_state* is a start state (pointed by a value of scale) as a result of action, *end_state* is a final state, *subj* is semantic codes of subject (see the

---

[6]Besides nouns, this dictionary contains as well adjectives.

description of the predicate *noun*), *contr_at* is the description of control in the construction AT, *contr_to* is the description of control in the construction TO, *contr_from* is the description of control in the construction FROM, *contr_with* is the description of control in the construction WITH.

The description of control in the construction AT is used to point to semantic codes of noun realizing a direct object (see the description of the predicate *noun*). The control in the construction FROM characterizes an indirect object from which an action is transferred. To do this, it is necessary to point to admissible combinations of preposition and semantic code (for example, for the verb *get: from_person, from_org*). The control in the construction TO presents an indirect object to which an action is directed. To do this, it is necessary as well to point to admissible combinations of preposition and semantic code (for example, for the verb *put: on_mach, on_body, to_thing*). The control in the construction WITH describes possible tools and methods to realize the given action.

For example, for the verb *shoot* (in the meaning *kill*):

| | |
|---|---|
| *inf* | shoot |
| *sem_type* | PROPEL |
| *scale* | HEALTH |
| *beg_state* | +50 |
| *end_state* | -100 |
| *subj* | person |
| *contr_at* | person |
| *contr_with* | weapon |

Depending on the sense, a verb can have several descriptions in the dictionary *verb*.

The dictionary maps the admissible compatibility of subject and objects with the character of action [7]. This permits to eliminate the semantic ambiguity for verbs(nouns) and to determine nonsensical sentences (see further).

---

[7] The analogous approach was proposed in [9] using the concept of *semantic valence*. The dependence of subject and objects on the type of verb is considered as well in works of Katz-Fodor [8].

## 5. Semantic representations of sentences using typical predicates

To represent every sentence, one must execute semantic descriptions:

- groups of nouns;

- simple sentences;

- compound and complex sentences.

The semantic representation is based on syntax analysis of sentence. The description of grammar is executed by means of *Backus-Naur form*[11] .

### 5.1. Semantic representations of noun groups

Subjects, objects, adverbial modifiers are described using *a group of noun.* Consider *a simple group of noun* including a *basic noun* and *attributes.* At the beginning of group a preposition can be. If the basic noun describes a physical or abstract object, then this noun is transformed to the predicate of type: *person, animal, organization, thing, machine, nature, abstr.* To determine the type of predicate, the semantic code of this noun (pointed in *the dictionary of nouns*) is used. A verbal noun is considered in association with an action or event described in a given sentence(see further).

If the basic noun poses a time or place, it is necessary to take account of a preposition. In this case, the noun is transformed to the predicate of type: *time, place.*

By means of article the individualizing character of basic noun is determined. If the attribute of basic noun is a property, then the predicate *property* is formed for this attribute. If the attribute is a participle, then the predicate of action or event is built. If nouns are used as attribute, then it can consider such group as an extended group of noun (see further).

When the group of noun is analysed, the *identification of objects* is executed. The identification permits to establish the identity of objects. Algorithms of identification are considered in [12]. If an identical object already exists for the given basic noun, then the new predicate is not formed, and only the available predicate is modified.

To analyse *an extended group of noun* (connecting simple groups of nouns using prepositions), rules of compatibility for nouns are used (see *the dictionary of nouns*). If a noun denoting a property is connected with other noun using the preposition *of*, then the predicate *property* is formed for the first

and pointed to this property in the second. If the preposition *of* is used to present an owner of thing or machine, then this owner is pointed in the predicate *thing* or *machine*. In other cases, rules of compatibility are used. The connection between simple groups of nouns is realized with the predicate *relation*.

*5.2. Semantic representations of simple sentences*

An analysed text is divided into fragments. Each fragment is characterized by unity of place and time. All actions or events of fragment are executed during a certain interval of time. The place and time are described usually in the first sentences. According to the *principle of coherence* [8, sec.9-2], these place and time, as a rule, are not pointed in following sentences. Then to form the semantic representation for these sentences, the earlier–described place and time are used. Personal, demonstrative and possessive pronouns pose references to earlier–described actions, processes, events, states and objects. Indefinite pronouns point to the existential quantifier (*some, any*) and the universal quantifier (*all, every, each*) [23, part II,ch.10].

Consider the semantic representation of simple declarative sentence. After the semantics of a subject, objects, place, time, other adverbial modifiers is described, to represent the semantics of sentence, it is necessary to form the semantic description of actions, processes, events, states. To do this, the analysis of predicate (in grammatical sense) is executed. The following predicates are formed as a result of semantic analysis: *action* — for physical actions, *message* — for the transmission of information, *thought* — for feelings and thoughts, *process* — for long goal–seeking occupations, *event* — for events and states. The choice of predicate is based on the indication of semantic type in *the dictionary of verbs*.

Forming of semantic predicates describing actions and events is reduced to the determination of factors which are typical for these actions and events:

- subject of action (event);

- objects which take part in the transmission of information or action;

- location and time of action (event);

- purpose (result) and method (tool) of action (event).

Semantic roles of indirect objects and adverbial modifiers are determined using prepositions for noun groups.

To overcome the problem of ambiguous expressions, we select a semantic variant corresponding to one of the descriptions of the given verb (in *the dictionary of verbs*). For example, consider the sentence *A man works with magnetic field*. As the verb *work* is combined with the abstract object *field* using the preposition *with*, it can eliminate the meaning describing the place of action.

Nonsensical sentences are found as well using *the dictionary of verbs*. For instance, consider the sentence *A man plays physic*. The word *physic* is inadmissible as the verb *play* is not combined with the object denoting *science*.

Paradigmatic relations between words of sentences ( see [3]) are described in a knowledge base and dictionaries of database. For a concrete sentence, these relations are presented by means of the predicate *relation*. Syntagmatic relations are pointed using connections between actions (events) and subjects, objects, adverbial modifiers. Such connections are formed by means of references in predicates *action, message, thought, process, event*(codes of subject, objects, time, place, etc).

*5.3. Semantic representations of compound and complex sentences*

A complex declarative sentence includes clauses, participle and infinitive constructions. For each clause and construction, the same algorithm is used as for a simple sentence. The connection between the main sentence and subordinate constructions is realized with the predicate *link*.

A compound sentence consists of several simple and complex sentences. The connection between independent sentences is described with the predicate *link*.

## 6. A knowledge base

To form answers for logical questions, it is necessary to have the dictionary containing explanations of words. This dictionary is realized as the *knowledge base*[11, 13]. The knowledge base consists of the predicates *tperson, taction, torganization,...* which have precisely the same structure as appropriate predicates of database (*person, action, organization,...*). Building the knowledge base is founded on algorithms of syntax and semantic analysis described in [11].

There is the set of words whose sense is implied in the system and realized by appropriate algorithms. All other words are explained by means of knowledge base. These explanations pose practically semantic(verbal) definitions of words.

The knowledge base includes the next descriptions:

- general properties of physical and abstract objects;

- general properties of actions, processes, and events;

- operations and scripts;

- schemas and plans.

There is a special article (fragment) in the knowledge base for each description of concept. We limit ourselves to the description of general properties (for physical and abstract objects, actions, processes, events and operations). The description of scripts, schemas, and plans is considered in [13]. Previously, we discuss the problem of primitives — the use of minimum dictionary to explain words [23, part II][8].

The minimum dictionary contains such classes of words to present:

- types of objects (*person, organization, place, people, animal, plant,...*);

- semantic role of words in sentence (*subject, object, way, tool, place, time, property, cause, purpose, ...*);

- basic actions, processes and events (*be, go, move, message, transfer, create, attend, thought, consist, ...*);

- the direction of moving (*top, down, left, right*);

- scales (*color, length, weight,...*);

- abstract concepts (*part, whole, knowledge, decision, analysis,...*);

---

[8]The problem of primitives is considered in [29]. Ch.K.Ogden had proposed the minimum dictionary of English including 850 words. If the control problem for complex social systems is considered as the object domain, then the correction of this dictionary is demanded.

- physical concepts (*force, velocity,...*);

- mathematical concepts (*number, set, point, line, plane,...*).

It should be emphasized that primitives are words having monosemantic, intuitively obvious meaning. Such parts of speech as conjunction, preposition, pronoun, particle are included as well in the number of primitives.

To find primitives for a set of words S (describing an object domain), each definition will be presented as *Subject+Predicate* where *Subject* is a determined word, *Predicate* contains a generic concept and difference. If $A$ and $B$ are nonintersecting sets of words, then $A(B)$ means that the set of words $A$ is determined by means of the set of words $B$.

.

**Definition 1.** *Let $S$ is a set of words, $S_0, S_1, S_2, ..., S_n$ are subsets of $S$ so that $S_i \cap S_j = \oslash, i, j = 0, ..., n, i \neq j, S = S_0 \cup S_1 \cup S_2 \cup ... \cup S_n$. Assume that $S_1(S_0), S_2(S_0 \cup S_1), S_3(S_0 \cup S_1 \cup S_2), ..., S_n(S_0 \cup S_1 \cup S_2... \cup S_{n-1})$. Then $S_0$ is the set of primitives for $S$.*

It is easy to verify that if $S' \supset S$, then for $S'$ the set of primitives $S'_0$ exists so that $S_0 \subseteq S'_0$. Another way, each word in the object domain can be determined using primitives.

The general purpose of definitions is to explain the sense of words. This permits to solve logical questions, which demand the analysis of connections between concepts[13]. It does not mean that we shall go the chain $S_{j1}, S_{j2}, S_{j3}, ..., S_0$ in the reverse order ($j1 > j2 > j3 > ... > 0$ ) completely. For most calls to the knowledge base, it is sufficient to establish the conformity with available facts [9].

*6.1. An article of noun*

An article of noun describes general properties of physical or abstract object. To form the article of noun (for example, *doctor*), at first the phrase of this kind is entered: *frame is a doctor.*

Then functions of this noun are indicated:

---

[9] For instance, it is necessary to check the availability of weapon for a person. It is known that this person has a pistol. The definition of pistol says that a pistol is a weapon which is kept in hand. It is sufficient for the check.

*doctor examines a person*
*doctor determines a disease*
*doctor prescribes a medicine*

The structure of object (for example, for the noun *car*) is formed by means of the phrase:

*A car consists of chassis, engine,...*

*6.2. An article of verb*

An article of verb describes general properties of actions, processes, and events. To form the article of verb (for example, *to go on*), the phrase of type *frame is to go on* is entered. To describe different values of verb, the group of noun can be used in addition:

*frame is to shoot from a gun*
*frame is to shoot a person*

Then semantic definitions and descriptions of concrete actions are entered. For example, after the phrase *frame is to learn*, there will be the descriptions:

*to do exercises*
*to answer a teacher*
*to visit a lesson in a class*
*to study a textbook*

After semantic definitions, motives and causes of action can be described. For example, after the phrase *frame is to shoot a person*, there will be the expressions:

*to kill the person by gun*
*to get money from the person as a subject is criminal*
*to pay off the person as this person outrages a subject*
*to annihilate the person as this person is the enemy of a subject*

*6.3. An article of operation*

An operation is a sequence of actions to execute a certain purpose. The algorithm of operation is based on principles of production systems [7]. All

conscious human activity includes the set of different operations planed beforehand and modified in the course of realization.

The description of operation can involve several alternatives. Each alternative is entered with a separate article. For example, for the purpose *how to rob an organization* one can indicate the following description of the first alternative:

*frame is how to rob an organization*
*alternative 1 ; to go to the organization*
*alternative 1 ; to come in*
*alternative 1 ; to neutralize personal*
*alternative 1 ; to open safes using tools*
*alternative 1 ; to take moneys and things*
*alternative 1 ; to come out*

The above sequence of actions consists of stages. Each stage can contain an action, object of influence, motive and cause, condition and way (tool) of action.

## 7. The comparison with other approaches to analysis of language

Let us compare the proposed approach with other approaches to semantic analysis of language. We limit ourselves to the most advanced and widely known researches.

The most deep investigations of semantics (it is our opinion) are based on the representation of objects, events, processes and actions in the view of *frames* [10, 19, 20, 27]. Firstly, this approach is adequate ours in the sense that the representation of semantics (for objects, states, events, actions, and processes) is equivalent ours. Secondly, for long-term saving of the great volume of information, it is necessary to use modern databases, i.e., we arrive again at predicates. Thirdly, the description of frame structures is a laborious process taking into account the great variety of objects, states, events, and actions. The essential distinction of our approach implies that we use typical predicates and verbal definitions from encyclopedias.

The *conceptual approach* has gained wide acceptance [4, 22, 24, 25, 26]. The main imperfection of this approach is that objects, states, events, actions and processes have not adequate and detailed semantic description. There is only the typification of action. This simplifies realization, but essentially

constricts an area of solved tasks.

The Russian school of semantics has gained essential successes[1, 6, 9, 28]. The main feature of these works consists in the limitation of language aspects — paradigmatic and syntagmatic relations between words [3, ch.1]. This does not permit to describe the functional purpose and structure of physical and abstract object, as well as the functional role and complex character of action and processes. Of particular interest is the work [18]. To describe facts and knowledge, the simple semantic net is applied — each node is a word. This does not permit to discover in full measure the semantics of words and sentences. To present processes, in this work scripts were proposed.

## 8. A formal logic system

The computer system for natural language understanding can be considered as a formal logic system $\mathcal{S}$ which is presented by the tuple:

$$\mathcal{S} = < \mathfrak{A}, \mathfrak{B}, \Phi, \Sigma, \Omega >, \tag{1}$$

where $\mathfrak{A}$ is an alphabet, $\mathfrak{B}$ is a set of atomic propositions, $\Phi$ is a set of functions, $\Sigma$ are rules of inference, $\Omega$ is a set of Horn's formulas (clauses).

The alphabet consists of letters, ciphers, special signs. Symbols are formed from letters. Variables, individual constants, names of function and predicates are pointed using symbols. A term is an individual constant, variable or expression $f(t_1, t_2, ..., t_n)$ where $t_i$ is a term, $f$ is a functional symbol.

*The atomic proposition* $P(t_1, t_2, ..., t_n)$ is formed from the predicate symbol $P$ and a list of terms. If terms do not contain variables, the atomic proposition is called *an atom*.

As well–formed formulas, only *Horn's clauses* are used[5]:

$$P_j \leftarrow Q_1^j, Q_2^j, ..., Q_n^j, j = 1, ..., m, \tag{2}$$

where $P_j$ and $Q_i^j$ are atomic propositions.

Atoms and Horn's clauses execute the function of axioms. Rules of inference use the *principle of resolution* and the *algorithm of unification* [5]. It can prove that every algorithm can be realized using this formal system.

The semantic interpretation of the formal system $\mathcal{S}$ is the tuple:

$$\mathfrak{I} = < \mathfrak{D}, I_f, I_P >, \tag{3}$$

where $\mathfrak{D}$ is the *domain of interpretation* including ranges of variables for predicates and functions. $\mathfrak{D}$ consists of sets of integer and real numbers, as well as sets of abstract words in the alphabet $\mathfrak{A}$. Each individual constant is connected with a certain value of variable. $I_f$ is the map of sort:

$$I_f(f) : M^n \to M \tag{4}$$

for each function $f$ where $M$ is a set from $\mathfrak{D}$. $I_P$ is the map of sort:

$$I_P(P) : M_1 * M_2 * ... * M_n \to \{true, false\} \tag{5}$$

for each predicate $P$ where $M_i$ is a range of a variable with the index $i$ for this predicate. This range is included as well in $\mathfrak{D}$.

Each concrete real object is characterized by a compact connected domain of *physical space* at a given point of time and described by the above predicate $\psi$ (after the substitution of individual constants instead variables). The *intension* of $\psi$ is the aggregate of properties described by means of variables of this predicate, the *extension* of $\psi$ is the set of real objects corresponding to this predicate [2, ch.1]. The above predicates $\varphi$ and $\gamma$ (after the substitution of individual constants instead variables) correspond to a set of compact connected domains in *a physical space–time* (for real actions and events). The predicates $\psi$, $\varphi$ and $\gamma$ can use to describe possible objects, actions and events. For example, it is necessary for future time.

**Definition 2.** *The aggregate $\mathfrak{P} = \{M_\psi, M_\varphi, M_\gamma, M_\delta, M_\mu, M_\rho\}$ is called the semantic representation of a declarative sentence if $M_\psi$ is the set of predicates $\psi$ corresponding to physical objects from this sentence, $M_\varphi$ is the set of predicates $\varphi$ presenting actions and processes of the given sentence, $M_\gamma$ is the set of predicates $\gamma$ describing events or states for objects of this sentence, $M_\mu$ is the set of predicates $\mu$ corresponding to sense relations between objects from this sentence, $M_\delta$ is the set of predicates $\delta$ corresponding to categories connected with objects, actions and processes of the given sentence, $M_\rho$ is the set of predicates $\rho$ pointing to abstract objects from this sentence.*

For a concrete declarative sentence describing a real (i.e., checked) or possible (i.e., unchecked) fact, individual constants must be substituted instead variables in predicates $M_\psi, M_\varphi, M_\gamma, M_\delta, M_\mu, M_\rho$. The *extension* of $\mathfrak{P}$ is the value of truth for the given sentence [2, ch.1]. If the sentence describes a real fact, then it is true. The *intension* of $\mathfrak{P}$ (in this case) is the real fact corresponding to a set of compact connected domains in *a physical space–time*.

Consider a general *synthetic sentence* presented by means of $\mathfrak{P}$. If this sentence summarizes facts using the universal and existential quantifier, then the verification of a such sentence can be a hard problem [23, part 2, ch.10]. In this case, the *intension* is the unification of facts.

The algorithm of forming $\mathfrak{P}$ is realized in the system LEIBNIZ using the language PROLOG[11]. This language includes Horn's clauses and atomic propositions. An inference is executed with the help of resolution principle and unification algorithm. To check the accuracy of semantic representation, simple and logic questions are applied[12, 13].

## 9. Conclusion

The proposed object–oriented approach poses the essential advance to solve the computer understanding problem. The representation of facts and knowledge using typical predicates permits greatly to simplify algorithms of processing and object descriptions. Besides, the effectiveness of processing is increased. This is of great importance for the object domain having the large quantity of words. The main feature of our approach (as compared with other investigations) is in the use of encyclopedic knowledge. This permits essentially to extend the class of solved problems.

Using the proposed technology, new principles of control for complex social systems can be realized: both for enterprise and state structures. For example, consider the problem of enterprise management. At the moment the technology ERP (Enterprise Resource Planning) is applied. A new technology (in addition to ERP) is focused on ensuring of profit increase and normal cash flow[14, 15]. This technology involves the next functions: forming of intellectual interface on a natural language to communicate with a control system; joint planning of production and sales to get the maximal profit; an adaptation of control system to internal and external events.

Other example concerns the domain of criminology. Living computer systems save information about criminal offences by means of databases. However, this information can not be used in full measure for logical processing as algorithms of semantic analysis are not applied in such systems. Thus, many actual problems are solved only by criminalists. Proposed algorithms permit to create more intellectual systems based on methods of criminalistics. Such systems will form answers for natural language questions about criminal offences using the purposeful selection from a database and logical processing of selected data. If at first the great selection from the database is demanded,

then it can gain serious results, which it can not get by other means. Some algorithms of such task solution are considered in the paper[13].

By this means, the presented technology discovers the perspective for the successful solution of problems in the social domains with the help of intellectual computer systems. We shall name this direction of investigations as *computer semantics*. Computer semantics is the part of cybernetics that studies questions connected with the application of semantic representations (for expressions of natural language) to control problems.

## References

[1] Ju.D.Apresjan and oth., Linguistic processor for complex information system, Nauka, Moscow, 1992

[2] R.Carnap, Meaning and Necessity, Ch., 1956.

[3] I.M.Cobozeva, Linguistic semantics, LIBROCOM, Moscow, 2012.

[4] W.G.Lehnert, M.G.Dyer , P.N.Johnson, C.J.Yang , S.Harley, BORIS - An Experiment in-Depth Understanding of Narratives, Artificial Intelligence, vol.20, 1, 1983.

[5] R.Kowalski, Logic for Problem Solving, Elsevier North Holland, 1979.

[6] N.N.Leont'eva, Automatic understanding of texts: systems, models, resources, Academia, Moscow, 2006.

[7] G.F.Luger, Artificial Intelligence. Structures and Strategies for Complex Problem Solving, Fourth edition, Addison Weslay, 2002.

[8] J.Lyons, Linguistic semantics, Camb. University Press, 1995.

[9] I.A.Melchuk, Experience of theory for linguistic models "sense-text", Moscow, 1999.

[10] M.Minsky, A Framework for Representing Knowledge, in P.H.Winston, The Psychology of Computer Vision, N.Y., 1975.

[11] Yu.Ostapov, Object-oriented semantics of English in natural language understanding system, arXiv:1109.5798v1 [cs.CL] 27 Sep 2011.

[12] Yu.Ostapov, Question Answering in a Natural Language Understanding System Based on Object-Oriented Semantics, arXiv:1111.4343v1 [cs.CL] 18 Nov 2011.

[13] Yu.Ostapov, Inference and Plausible Reasoning in a Natural Language Understanding System Based on Object-Oriented Semantics, arXiv:1202.0116v1 [cs.CL] 1 Feb 2012.

[14] Yu.Ostapov, Intellectual Management of Enterprise, arXiv:1207.0313v1 [cs.CE] 2 Jul 2012.

[15] Yu.Ostapov, Adaptive Control of Enterprise, arXiv:1211.5890v1 [cs.CE] 26 Nov 2012.

[16] R.Penrose, The Emperor's New Mind, Oxford University Press, N.-Y., 1989.

[17] G.Polya, Mathematics and Plausible Reasoning, Princeton, University Press, 1954.

[18] E.V.Popov, Contact with Computer on Natural Language, Nauka, Moscow, 1982.

[19] D.A.Pospelov, Situation Control: Theory and Practice, Nauka, Moscow, 1986.

[20] D.A.Pospelov, G.S.Osipov, Applied Semiotics, see http://do.gendocs.ru/docs/index-213185.html

[21] R.Quirk, S.Greenbaum, G.Leech, J.Svartvik, A University Grammar of English, Longman, London, 1973.

[22] C.K.Riesbeck, R.C.Schank, Inside Case–Based Reasoning, Lawrence Erlbaum Associates, New Jersey, 1989.

[23] B.Russel, Human knowledge. Its scope and limits, George Allen and Unwin Ltd, London, 1958.

[24] R.C.Schank, N.M.Goldman, C.J.Rieger, C.K.Riesbeck, Conceptual Information Processing, North–Holland, Publishing Company, 1975.

[25] R.C.Schank, M.Lebowitz, L.Birnbaum, An Integrated Understander, American Journal of Computational Linguistic, vol.6, numb.1, Jan-March, 1980.

[26] R.C.Schank, L.Birnbaum, J.Mey, Integrating Semantics and Pragmatics, Quaderni di Semantica,vol.6,numb.2,1985.

[27] M.Selfridge, Integrated Processing Produces Robust Understanding, Computational Linguistic, vol.12, numb.2, April-June, 1986.

[28] V.A.Tuzov, Computer semantics of Russian language, Saint Petersburg State University, 2003.

[29] A.Wierzbicka, A Semantic Primitives, Frankfurt, 1972.

[30] T.Winograd, Understanding natural Language, Academic Press, New York, 1972.