Modeling of communication processes in information systems

G P Vinogradov[1] and A A Prokhorov[2]

[1] 170023, Tver State Technical University, Marshal Konev St., 12, Tver, Russia, E-mail: wgp272ng@mail.ru
[2] 170024, Research Institute "Centerprogramsystem", 50 let Oktyabrya Ave., 5, Tver, Russia, E-mail: info@cps.tver.ru

Abstract: The limitations imposed by the general logical and mathematical decision-making abstraction of on decision-making procedures reduce the science of decisions to a set of mechanical decision-making methods. In practice, system researchers consider people as objects that mechanically react to input stimuli. This eliminates the possibility of choice management. In this regard, the problem of the development of the theory of choice by ensuring the compatibility of the theological approach and the approach based on causation is currently relevant. In respect to the behavior theory, this approach requires the inclusion of the theory of choice of psychological aspects in the subject area. Primarily, these are the problems of perception, awareness, understanding the properties of the situation of choice and evaluation of the results of choice, communication, conflict, and a number of others.

Key words: subjectively rational choice, communication processes, software agents, information system

1. Introduction

In practical programming, an agent is a computer-generated system that is located in some environment and is designed for autonomous actions in this environment in order to achieve specified goals. There are some known limitations preventing the widespread use of MAC technologies that use the BDI architecture of an agent. The desire to bring the concept of building agents in accordance with modern requirements is currently associated with using biological principles and searching for analogs in wildlife and human society. Thus, the papers [1–4] identify and describe components and observed parameters of purposeful behavior using the formalism of fuzzy systems. In [5-10], the authors consider the problem of a subjective rational choice related to satisfaction of needs, execution of obligations and the system of values and norms. Such form of choice makes individual purposeful behavior possible. In addition, it makes it possible to describe behavior as a process.

Observation over it allows identifying individual steps – subprocesses; their combination of interactions forms a system. An important subprocess in such system is formation of vision of the situation of choice. They should be considered as models. Their form of description can be different: textual, mathematical, verbal, etc. [8]. They are the basis for the forecast of possible states from the selected modes of action, as well as for estimating the probability of their obtaining and the degree of desirability [10]. Formation of vision is based on the processes of obtaining information from the environment. The main channels are: communication with other subjects, internal dialogue, observation. Three main processes – *perception, awareness* and *experience* – together give the concept of a situation, generate new concepts, change and supplement the list of beliefs (knowledge base). This is the basis for forming a model (vision) of the situation of choice. The driving factor of behavior is affective evaluation of the realization of needs and capabilities. Assessments of the current situation cause a state of satisfaction/dissatisfaction in a subject, who characterizes them by feelings and attitudes. The agent selects from the known set of methods and opportunities those that allow him to reach the state of satisfaction rationally.

The model of agent's subjectively rational choice of ways to meet the needs and fulfill obligations is based on the methodology of gradual strengthening of agents' capabilities. The latter assumes that in the process of searching for ways to meet the needs, the agent gradually builds up a variety of ways do this [11]. Here he uses his experience and experience of other agents he interacts with, as well as the available knowledge.

In most models of agents whose behavior is described by sociological and psychological concepts, communication is defined in terms of protocols that do not have a direct connection with the first. It creates problems in developing and modeling communication processes between agents, as well as operators with agents. This is due to the fact that the description of agents' autonomous behavior contains high-level abstractness formalism, communication is defined in terms close to implementation. The difference in description levels does not allow simulating communication between agents at the same level as their autonomous behavior is described.

2. The statement of the research task

In existing information systems (which include software agents), communication is based on the so-called command interface. It is implemented through dialog boxes, menu systems, etc. When artificial entities communicate, the main problem is creation of message formats and ways of organizing information channels. Complication of solved problems leads to the necessity of organizing an interface by artificial entities between themselves and an operator based on some "professional language" that is relevant to some subject area. This involves considering an interface as a sign system.

Let an interface language phrase describes some situation *S*, which has a meaning and a name. This phrase might have the mode of action *A* from a set of possible ones. Its implementation puts the environment into a new state. It is clear that a "correct" interpretation of environment perception results will lead to a "correct" choice of a mode of action. This means that between *S* and *A* there should be an information model of a domain that describes and allows determining relationships between stimuli and system reactions. Consequently, autonomous behavior of an agent as an information system involves development and integration of conceptual models of subject areas into its information systems. The main purpose of conceptual modeling is the formalization of accumulated knowledge about a certain subject domain in the form closest to understanding by participants in the communication process. In practical terms, the most promising direction of conceptual modeling is ontological modeling. Ontologies are explicit formal specifications of domain terms and relationships between them. Models of ontologies used in the theory of agents are almost equivalent to the concept of language semantics in a certain subject domain common in programming.

The rules of ontology specify statements that relate concepts and relationships between them. However, these models have a number of limitations. For example, the relationship between concepts may be concepts themselves. Overcoming this shortcoming is associated with using conceptual modeling of a subject domain.

The approach supposes that association is also a concept rather than a named type of connection, which sets a role or relation in a model. In addition, the language of the conceptual model does not contain means for describing relations and roles, since the conceptual structure containes all the rules necessary for derivation.

This allows increasing the level of model abstraction and developing a small number of general algorithms with a small computational complexity that do not depend on the subject domain, since they are formulated in the most general operations on concepts [12, 13].

3. Solving the problem of communication based on ontologies of concepts

There are four types of concepts: single, simple, specific and abstract. Single concepts are the result of a mental selection of unique entities in a subject domain and assigning names to them. This allows replacing the entity with a sign that is identical to it in some sense. Simple concepts are formed by combining entities similar in some sense. They are assigned a unique name, considered as a single concept, and acceptable manifestations (values) range is defined, considered as a set of single concepts. Specific concepts are based on a union of entities with the same characteristics, which allows defining a set of entities that correspond to a notion. The features to be distinguished are considered as simple concepts, and the names of concepts are considered as single concepts. A particular concept is represented by a set of entities that form its extensional or volume. The name of a concept is a sign expression of a concept with an attributed meaning. The scheme or structure of the concept is set by a set of characteristics that characterize the concept. The intensional or content of the concept is considered as a set of values of interrelated characteristics that allow recognizing the entities belonging to the concept. At the same time, entities should be considered as some fragments of a subject domain represented by signs, symbols, images, etc. In order to manipulate entities in an information system, they must be called and treated as single concepts.

Formation (determination) of abstract concepts includes using complex forms of abstraction based on the establishment of relations of independence, differentiation and integration of features between the concepts. To form abstract concepts, there four abstractions: generalization-specialization, typing-concretization, aggregation-decomposition, and association-individualization.

The process of studying a subject domain supposes formation of new concepts or identification of existing ones. In this case, from the perspective of solving a certain problem, there are entities that have or are assigned with certain names. Further, the set of identified entities is analyzed in order to define their similarities and differences. Similar entities are grouped and form concepts or fill existing concepts with problematic content.

Well-known formalisms define a lot of relationships of different nature on concepts. Unlike them, here there is another formalism – a conceptual structure that is defined by a set of concepts with four kinds of mappings, the only purpose of which is to show ways of concept formation, ways of abstracting [12, 13].

Definition 1. A conceptual framework $S = (N, T, G, C, A)$ is a finite set of concepts N with four finite sets of mappings in the form $N \rightarrow N$: type checking T, generalization G, aggregation C and association A.

A conceptual structure is designed to reflect the results of conceptual analysis of the subject domain in a formalized form, and expresses displaying some concepts into others. The abstractions used here are considered as mental operations that are necessary and sufficient for the mental isolation and transformation into separate concepts of representations that are accumulated relative to a formalized domain.

The definition of the concept assumes that each concept has a schema, which is a set of features or simple concepts to define this concept. However, when defining a conceptual structure, only displaying some concepts into others is to be defined. The problem arises of defining schemata of concepts.

A scheme can be found for any concept from the conceptual structure S according to the following recurrent procedure:

- a scheme of a single concept is empty;
- a scheme of a simple concept consists of the name of this concept;
- a scheme of a differentiation concept is equal to the marked intersection of schemes of differentiated concepts;
- a scheme of an integration concept is equal to the marked integration of schemes of integrated concepts;
- a scheme of the concept obtained after integration and differentiation is equal to the marked unification of schemes of integrated concepts belonging to the marked intersection of schemes of differentiable concepts.

The ability to define schemes of abstract concepts allows displaying them as certain concepts as a list of entities that are defined through primary notions of the subject domain.

To verify a conceptual structure, it is required to check the computability of the schemes of all its concepts. The requirement of computability of concept schemes is the extension of a set regularity property to concepts. In this case, the computability of concept schemes guarantees the absence of concept definitions through themselves, which is inadmissible in any formal or substantial theory that claims to be adequate.

Thus, a conceptual analysis of the subject domain provides its ontological description given by a conceptual structure. The principal differences of the conceptual structure from other conceptual schemes are the following:

- terms are not divided into meanings, signs, concepts, connections and roles; instead, there is one term – a concept with particular manifestations such are meanings, signs, concepts, connections and roles;
- the possibility of representing associations as independent concepts, which allows, for example, expressing the generalization of associative links;
- the definition of concepts that can be both generalization and the association of other concepts;
- semantic description invariance, which does not require subject knowledge for its interpretation.

A conceptual structure of a subject area assumes assigning concepts and their abstraction methods. A concept scheme can be also computed. It is a set of simple concepts (attribute concepts) that allows considering abstract concepts as specific ones. There is a concept intension to determine specific sets of simple concept meanings that define a particular essence of specific and abstract concepts. It sets the rules for correlating the sets of simple concept meanings with a particular entity from the extensional of the concept described.

Definition 2. A conceptual model M of the subject domain is its conceptual structure S that is supplemented by a description of intensionals D of all concepts in it, $M = (S, D)$.

From the above reasoning, it follows that the universal form of assigning an intensional is enumeration of attribute sets of each entity belonging to the concept. In practice, it causes difficulties when the scope of the concept is sufficiently large. Another form of describing intensionals is also possible – the use of procedures (formulas, functions) that resolve the extensionality of the defined concept.

This means that a low-level representation of the conceptual model of the subject domain can be solved by means built into the database management system. To set simple concepts (attribute concepts), we might use integrity constraints that narrow the ranges of simple data values, or special tables containing valid values of a simple concept. Specific concepts are represented by tables with columns corresponding to simple concepts from its scheme.

Representation of an abstract type concept is also complicated by using a query to a database that processes records from several tables with the same set of fields simultaneously. The implementation of an abstract generalization concept requires a separate table with intensions of entities that are absent in the extensions of generalized concepts. Therefore, that the abstract generalization is represented by a query to the database, in which all records from the generalization concept table and unique entries from tables of generalized concepts with a set of fields belonging to the generalization concept scheme are a subject to processing.

Aggregation of concepts is represented by a query that executes the product of two or more tables. However, the implementation of the abstract association concept cannot be reduced to a query to a database and requires allocating a separate table that connects records from two or more associated concept tables.

Therefore, when using database management systems, the main tool for describing concept intensionals is the enumeration of sets of interrelated characteristics in the form of a physical or virtual table. A physical table is stored in memory. A virtual table is generated dynamically due to executing a query to a database. In this case, a table row (record) corresponds to the described entity from a concept extension, and the column (field) corresponds to a concept sign.

The use of a conceptual structure that specified in the form of a tree, as well as mapping of entities from concept extensionals in the form of lists are sufficient in the context of functioning of an information system designed to automate decisions of a specified class of applied problems.

If we abstract from a specific content of actions and procedures in the algorithms for solving applied problems, we can conclude that all such actions might be reduced to three abstract operations on concepts: concept creation, change and removal. Thus, a description that consists of three elementary operations [12, 13] is a semantically invariant form of describing a solution of applied problems in a modeled subject domain.

Concept creation. The operation of concept creation arises when a conceptual model of a subject domain is complicated. It includes determining a name of a new concept and setting a method for its abstraction. Calculation of a new concept schema follows automatically after its creation. Depending on the concept type, either a table (for simple and specific concepts, as well as for abstract generalization concepts and association concepts) or a representation/query (for abstract typing concepts and aggregation concepts) that can enumerate entities belonging to a created concept is created in a database.

Removing concepts. The operation of removing a concept arises in the case of changing visions of a subject domain. It consists in changing a description of all concepts in definitions of which include it.

Concept change. The operation of concept change is used when it necessary to fill concepts with a specific subject content. Change operations affect specific and simple concepts. In this case, three actions are possible: editing an existing entity (record), deleting an existing entity (record), and adding a new entity (record). The same actions are also performed on abstract generalization concepts and association concepts that have their own tables.

To implement the operations of creating, deleting and changing concepts in a database management system, there might be special stored procedures that ensure integrity and consistency of a conceptual model of a subject domain.

Inferences. We might defined any reasoning as a transition from one or more judgments that are a premise of an inference to a statement that is the consequence of an inference. Rules for constructing inferences are based on the inference rules generally valid in a subject domain, i.e. generating true statements under all possible assumptions. The rules for constructing inferences in the case under consideration are based on inference rules that are formalized in the conceptual structure of the subject domain. The conceptual structure itself is considered as a formal theory that preserves the truth of all consequences in it.

The following logical statements specify inference rules on knowledge [13]:

$$\bigvee_{(\forall N_i > N} N_i(E) \rightarrow N(E) \quad \bigvee_{(\forall N_i \triangleright N} N_i(E) \leftrightarrow N(E) \tag{1}$$

$$N(E) \rightarrow \bigwedge_{(\forall N_i \succ N} N_i(E) \quad N(E) \leftrightarrow \bigwedge_{(\forall N_i |\!\succ N} N_i(E) \tag{2}$$

where $\vee$ ($\wedge$) is a logic connective OR (AND); $>$ ($\succ$, $\triangleright$, $|\!\succ$) is a mapping generalization sign (association, typing, aggregation) of concepts; $\forall$ is a generality quantifier; $\rightarrow$ is a logic connective of following.

The inverse proposition of the first inference rule from (1) is true only for the abstraction of specification typing (second rule), since the specialization generalization abstraction describes a larger number of entities than in the union of extensions of generalized concepts.

In turn, the inverse proposition of the inference first rule from (2) is true only for the abstraction of decomposition aggregation (second rule), since the abstraction of the individualization association describes a smaller number of entities than in the Cartesian product of associated concepts extensions.

Queries. To turn an information system into a complete knowledge base, it is necessary to implement queries for extracting facts (assertions) and outputting meaningful statements about the modeled domain.

Facts (assertions) and statements are propositions with logical connectives AND, OR, NOT with two types of predicates [13]:

- one-place membership predicates of the entity *E* to the concept *N* of the type *N(E)*;
- relations $P[E] \circ V$, where *P*[*E*] is a functor that returns the value of the attribute concept *P* of the entity *E*, $\circ$ is a relation sign (=, $\neq, >, \geq, <, \leq$ □, > etc.), *V* is some value concept.

After setting restrictions on one or more characteristic values, the information system searches for entities that satisfy the specified conditions and then displays them as child nodes of the search node.

The search for concept essences by the key is performed in a similar way after entering the key or its part in the name of the search node. If the key has several signs, then they are separated by a special sign.

A more complex search requires using queries to the knowledge base, so there is a corresponding query language and a knowledge output machine supporting it, which implements inference rules (1) and (2). It should be noted that for many application problems it is sufficient to use a client application subject line to find concepts in their conceptual structure by their names with the subsequent search for the required entities in the extensions of the found concepts.

4. Knowledge representation in a communication process

In addition to knowledge presentation, extraction and actualization, another important task must be solved – knowledge representation. It consists in changing the form of knowledge presentation. It is based on construction of conceptual submodels with their subsequent visualization by special programs.

Definition 3. A submodel M' of the concept model *M = (S, D)* is a conceptual model $M' = (S', D')$ with the following relations: $S' \subseteq S$, $D' \subseteq D$, where S' is some substructure (fragment) of the concept structure *S*, D' is a description of concept intensionals in the conceptual structure S' [13].

A conceptual submodel is constructed according to the following procedure. First, a number of basic concepts is identified, which must be included in the submodel according to the conditions of the problem being solved. Then the conceptual substructure is iteratively constructed. It includes all concepts having connections with the initial and then with the current set of them. Iterations finish when a current set of concepts ceases to replenish itself. At the end of the procedure, a description of intensionals of the concepts from the conceptual substructure is created.

The construction of conceptual submodels is necessary to create data that is required for visualizing subject domain fragments by third-party programs. To display such submodel, the forms implemented by the corresponding application programs might be used: Gantt charts, resource lists, resource scheduling, figures, slides, videos, etc. For this purpose, an information system includes a module which performs their visualization based on their conceptual models.

Submodels might be used to automatically create various kinds of documents (files). In this case, the submodel has expression rules (representation) of concepts in a document body. The expressive means for such representation will depend on the required display form (text, graphics, sound, animation, etc.).

To display a submodel in text form, representation rules might be made in the form of a document template. Template creation involves using a special markup language that allows specifying the forms of concept expression in the text.

The extraction operator allows retrieving and inserting into its location a formatted value retrieved from the conceptual model according to the specified path. The calculation operator is used to represent a formatted value of some calculated expression in the text form. The syntax and semantics of expressions are like those of high-level languages. In the implemented information system, interpretive languages such as VBScript and JScript might be used as languages for specifying expressions.

The setup operator serves to change values in a conceptual model and additionally might be used to create temporary simple concepts (variables). The selection operator is necessary for implementation of text branching in the model representation process. The iteration operator represents composite concepts. Operators can be interleaved, since all their parts are a plain text.

Submodels of other stable fragments of subject domains and their corresponding visualization forms have a similar way of creation. For example, charts and diagrams; infographic (graphical representation of charts, maps, figures, formulas, etc.); technical graphics (graphical representation of schemes, drawings, axonometry); dynamic business process models in various notations (graphical representation of processes and their current states).

5. Knowledge base

Any information system for processing knowledge is based on a formal apparatus for knowledge representation and manipulation in order to simulate human reasoning to solve applied problems. In turn, a knowledge base is a database that contains facts about a certain subject domain, as well as inference rules that allow making conclusions automatically and obtaining new statements about available or newly introduced facts [13].

This allows us to consider an information system with a conceptual model of the subject domain $M = (S, D)$ as a knowledge base. In this case, the concept structure S (concepts and abstraction methods) sets the inference rules on knowledge, and the concept intensionals D sets facts (assertions) about the subject domain.

Assertions. The facts (assertions) are statements about the belonging of subject domain entities to concept extensionals. According to the formula (1), the entity E belongs to the concept extensional N if and only if the set of characteristic values of the entity E, which is ordered according to the shm N concept scheme, belongs to the concept intensional int N:

$$\left(\coprod_{\forall P_i \in shmN} P_i[N] \right) \in \text{int } N \leftrightarrow N(E), \tag{3}$$

where $\coprod$ is a marked (ordered) union performed with the repetition of elements; $Pi[E]$ is a functor that returns the attribute concept value Pi oa the entity E; $\leftrightarrow$ is a bipolar logic connective; $N(E)$ is a one-place membership predicate of the entity E to the concept extensional N, $N(E) \leftrightarrow E \in \text{ext } N$ [13].

According to the formula (3), the feasibility of the one-place predicate $N(E)$ is determined by the current state of the intensionalals D, i.e. the information system implements the open world model.

6. Implementation

A multi-layered structure of an information system with a conceptual model of a subject domain contains a number of subsystems.

The subsystem providing consumer operation is implemented as a client application that functions according to the browser principle. It means that the application receives data from the knowledge subsystem to display the conceptual structure in the form of a tree, and concept intensionals and extensions in the form of tabs and a list.

The subsystem for supporting work with the knowledge base serves to form data in formats used by a client application for displaying and manipulating the concept model. Data exchange is carried out via two interfaces. One interface is to display a tree node and provides its name, icon, description of the context menu and tabs, as well as interpreted program texts for changing a node name, processing interface events, displaying child nodes. The second interface is to display the list and provides a description of the list header (columns), a number of rows consisting of cells by the number of columns, as well as interpreted program texts for changing the contents of cells and processing interface events.

In order to increase the effectiveness of the information system and to shorten the time for the implementation of new procedures, classes of specific concepts are used in a uniform canvas, for example:

- a class of concepts represented by standard questionnaires;
- a class of concepts that express all kinds of reports for analyzing the current state of the modeled domain;

- other classes of concepts with a typical representation and the same type of processing.

The point to note is that an information system itself also has some conceptual model. Working with this model is also through the subsystem of working with a consumer. The model can include such concepts as:

- a module loaded in the process of working client application; it serves to implement a specific function of mapping a conceptual model or solving a specific domain problem;
- an event registered in the information system; it allows specifying a processor for operations of concept creation, removal, or change;
- a form for implementation of various scenarios for data inputting and processing by a user;
- other concepts necessary to implement the requirements for a particular domain model.

The subsystem of inference and logic formation is responsible for performing operations on concepts. This layer can be implemented through the procedures on the dedicated server (logic layer), as well as through the stored procedures of the database management system (data layer).

To limit operations on concepts, as well as to form individual conceptual models, the information system in the logic layer implements a developed mechanism for determining and inheriting rights.

A database management system represents the data storage and processing subsystem. In the case of deploying large information systems, each layer can be implemented not on one but on a server group, and contain means for dynamic distribution of the load on next layer servers. In this case, the data layer is implemented as a distributed database.

Conclusion

The presented approach to the construction of an information system with a conceptual domain model have been used in development of integrated simulators. They are designed to test interaction and teamwork of operators when managing complex special purpose software and hardware facilities. The main requirement for such facilities is the simplicity and naturalness of communication between operators during operational commitment. Communication is based on a "professional language", which makes it possible to use conceptual models of a subject domain. They exclude the use of concepts and all sorts of relationships between them, which are a part of model semantic load. This fact significantly improves the universality of communication models and ensures invariance regarding the subject domain.

This effect is due to the fact that the links between concepts in conceptual models are concepts themselves, and the model is built on the basis of identifying and describing the abstractions that formed (defined) concepts.

Concept intensionals define subject semantics completely. The abstraction of concepts formalized in a conceptual structure of the subject domain determines only structuring of intensionals. In this case, it is not necessary to specify logical statements (formulas, functions) that characterize concepts and are inference rules. All that is necessary for a knowledge-based inference is in the conceptual structure of the subject domain and concept intensionals.

Refusal of describing associations as links with different semantic markup makes the conceptual structure of the subject domain representable as a tree.

Since the cooperative training of operators is associated with the implementation of known structure management scenarios, the use of conceptual models of the domain allows implementing voice control. It implies the ability to manage the script using a set of commands consisting of specific words. Thus, the operator has the opportunity to enter information using his voice. After pronouncing certain words, the device starts to recognize speech (converting audio signal into digital information). After the entered information is correctly recognized, the program proceeds to the specified algorithm. It performs a function that attached to a particular command. A virtual assistant continuously collects all data about the entered requests and operator's order to build his profile. It tries to adapt to each user as much as possible and allows evaluating the quality of the model of choice of each particular operator.

References

[1] Yunfeng Chen, Hazar Dib 2011 Modelling Information Behavior of Communication System Supported by Building Information Modelling 2011 2nd International Conference on Construction and Project Management/ IPEDR vol.15 © (2011) IACSIT Press, Singapore pp. 120-125.
[2] Vagin V N, Fomina M V 2014 Reasoning and inductive notions formation *Educational resources and technology*.. Vol.2. No. 5 (2) pp. 34-39
[3] Karpova I P 2016 Pseudo-Analog communication in the group of robots *Mechatronics, Automation, Control* 2 (17) pp 94-101
[4] Lapshin V A 2010 Ontology in computer systems (Moscow) p 203
[5] Vinogradov G P, Chabarov A P and Burdo G B 2015 Subjective models of rational choice in automated systems design and management *IJAER* **10** 24 pp 45731–35
[6] Dobrov B V, Ivanov V V, Lukashevich N V, Soloviev I D 2009 Ontologies and thesaurus: models, tools, applications (Moscow) p 173
[7] Vinogradov G P 2017 The subjective model of rational choice in multi-agent systems *Mahtematical Modelling* **3** pp 155–58
[8] Freeman J 2005 Argument Structure. Lawrence Erlbaum Associates p 149
[9] Vinogradov G P 2017 A subjective rational choice *J. of Physics: Conf. Series* **803** pp 012–176
[10] Pavlov A V 2018 Initial generation of concepts in the processing of images on the Fourier-dual operations algebra Artificial Intelligence and Decision Making **2** pp 85-98
[11] Chudova N V 2017, a Conceptual model of the semantic regulation of behavior of intelligent agents *Artificial Intelligence and Decision Making* **4** pp 23-32
[12] Vykhovanets V S 2009 Applied conceptual analysis *Proc. 8th Int. Conf. "Cognitive Analysis and Management of Situation Development"* (Moscow: Institute of Control Sciences) pp 62–65
[13] Vykhovanets V S Information system using conceptual domain model // Large-Scale Systems Control . - 2017, Vol. 66. - C. 25-67. - ISSN 1819-2440, ISSN 1819-2467.