Ontological Approach to the Development of Decisions of Applied Problems in the "Ontointegrator" System

O. A. Nevzorova
Research institute "Applied Semiotics"
Academy of sciences RT
Research institute of mathematics and mechanics Kazan state university
Kazan, Russia
e-mail: olga.nevzorova@ksu.ru

V. N. Nevzorov
Kazan state technical university (KAI)
Kazan, Russia
e-mail: nevzorov@mi.ru

Abstract[1]

The paper presents the new method of decision constructing for applied problems in the "OntoIntegrator" system. The method is based on idea of decomposing of the problem solution into structural components of the ontological models. We discuss the structure of ontological system and the decisions of some applied problems.

1. Introduction

The "OntoIntegrator" system is an ontolinguistic research tool environment for NLP problems. An ontolinguistic approach [1] is realized in the system and integrates the following basic processes which are:

- development of applied problems solution based on ontological models;
- an in-depth linguistic analysis;
- a knowledge representation in form of ontology models and specialized databases;
- a modularity and expandability of toolkits;
- an adaptation of linguistic technologies to a new domain.

The "OntoIntegrator" system is directed to solution of complicated problems in the text processing using semantic (ontological) models. One of most attractive idea in the field of solving complicated problems is the hierarchical decomposition. A complex of hierarchical structures are designed based on hierarchy of classes. The main preference of hierarchical structure is the fact that the computational problem on each level of hierarchy comes to short list of tasks from lower level. Therefore the computational cost of the current task at each level could be estimated well enough.

The methods of tasks and processes decomposing are discussed in several papers [4–5]. The HTN (Hierarchical Task Network) method is one of most common methods of planning hierarchical net of tasks. In the HTN planning the primary high level of problem structure is specified by using operations of decomposition. The process is finished on the level of primitive actions, which are, as a rule, realized automatically. Using HTN planning for linguistic problems solution is almost not analyzed.

The distinctive feature of linguistic problems is polysemy, which resolution in the current context influences on choosing the following operation in the sequence of operations. The operation planning should be combined with ordering the partial alternatives and possibility to decline the chosen alternative in case of conflict on lower level.

This paper suggests a new approach to designing the solution of linguistic problems based on the idea of decomposing the problem solution into structural components of the ontological models, which is realized on the "OntoIntegrator" system.

2. The general decomposing scheme of the problem solution in "OntoIntegrator" system

The "OntoIntegrator" system consists of the following functional subsystems:

- "Integrator" subsystem;
- "OntoEditor" subsystem of ontology development;
- "Text Analyzer" subsystem;
- Subsystem of external linguistic resources managing;
- Subsystem of the ontological models.

The "OntoEditor" subsystem of the ontology

Proceedings of the 12th international workshop on computer science and information technologies CSIT'2010, Moscow – Saint-Petersburg, Russia, 2010

development [3] provides the basic table functions which is necessary in working with ontology (adding to, changing, records deleting, automatically records correction, several ontologies managing, including compound ontologies with common list of relation types, text equivalents; importing the ontologies with different data format, ontology refining, statistic control under objects of ontology, searching for chains of relations.

The visualization module supports different graphic modes of the system, including the mode of ontology constructing.

"Text Analyzer" subsystem includes linguistic tools of the system, which might be used for solution of problems in morphological analysis, ontological marking, ambiguity resolving, segmenting and text models constructing in applied purposes.

The subsystem of external linguistic resources managing supports keeping basic linguistic resources consisted of marked grammar vocabulary and set of specialized data bases.

The core of "Integrator" subsystem is Problem Processor, which includes Problem Constructor, Text model identification unit, Problem Solver and Result Constructor.

The text processing task solution in the "OntoIntegrator" system realized by subsystem of the ontological models managing, which includes different types of ontologies: applied ontology, ontology of models and ontology of tasks. The core of the ontology system is the ontology of models.

The Ontology of tasks formally describes as a structure $\Psi = \{P, \Theta, Z\}$ where P – set of concepts-tasks, Θ – set of relations, Z – set of interpreting functions.

The concept-task $p(A_1, A_2, A_3) \in P$ is defined by a number of attributes where an attribute A_1 – the name of task, an attribute A_2 – the functional type of task, an attribute A_3 – the task descriptor.

The functional type of a problem defines its functional class from the open set of classes: Operations, Data Sources, Outcomes, Realizations. P-realization class defines the structure of problem solution in the form of chain of the basic operations and P-realizations. The instance of the P-realization class constitutes an executable module of the specific applied problem solution.

The operation class contains extended set of basic operations for which the specific semantics is defined.

The Data Source and the Outcome classes define different types of data and their formats. The instance of the P-realization class is formed according to specific mechanism of assigning basic chains of operations based on inclusion relation from various of relations. The inclusion relations is attributed by "number of sequence", which helps to transmit parameters between operations.

At the fig. 1 were presented the process of assignment a set of operations and realizations used during the construction of a new P-realization and also the logical structure of the produced P-realization.

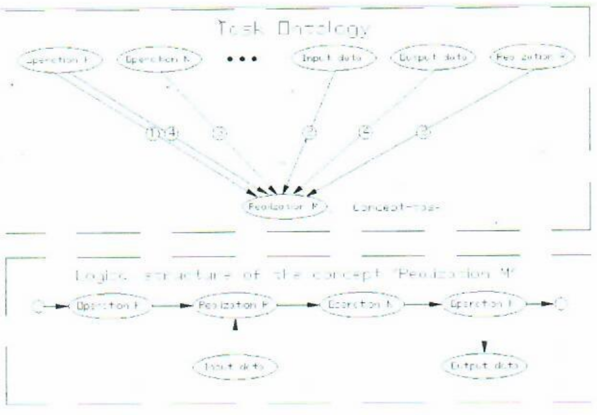


Fig. 1. Concept-task constructing

A new concept-problem contraction based on the chosen operations, realizations and data sources. A logical structure formation of Realization N concept structure based on mechanism of prescribing numbers of sequence to the chosen operations and realizations in general consecution.

Thus, in our example the "Operation K" gets number 1 and 4, the "Operation N" gets number 3, the "Realization K" and Data Source take number 2, the Outcome is taken number 4. The logical structure planning of realization occurs in the window of "Task Constructor" module with the following automatically code generation of the new realization.

The Ontology of models formally describes as a structure $M = \{M_1, M_2, M_3\}$, where: M_1 is a set of classes of the models, M_2 is a set of relations, M_3 is a set of functions for interpreting.

A set M_1 contains the following classes: quality (unary predicate), relation (n-argument predicate), reference (reference type) and m-realization (library module).

The proper hierarchy might be constructed for each class from ontology (for example, a hierarchy of properties, a hierarchy of relations, a hierarchy of realizations). Development and usage those hierarchies depends on type of linguistic problem, because it is a dynamic process. Actually, the ontology of model represents the aggregate of different hierarchies, which are the hierarchies of concepts (qualities, relations) and computing hierarchies (class of realization). Some elements from hierarchies of concepts might be compared with methods of computing hierarchies.

The structural elements of linguistic problem solution represented as a concept from ontology of tasks which go into set of the ontology of models structure. Substantially, the models are used to solve a concept-problem and allow us to assign (to interpret) the components of the solution as a assignment task or as a task of determining relations or as a task with known computational procedure. The set models are opened and dynamically enlarged.

A specified concept-model S (a_1, a_2) is attributed in the following way: attribute a_1 is a name of the class and attribute a_2 is a functional type.

The process of a concept-model construction takes two steps: first, defining type of model with help of the certain mechanism of specification and then constructing a concrete model according to adjusted type. The M-realization concept-model type might be constructed owing to the mechanism of concept – examples aggregation, examples of concept-models unite according to aggregation relations from set of others.

3. The applied linguistic problem solution in the "OntoIntegrator" system

The "OntoIntegrator" system is oriented to modeling a solution of the automatically text processing problems, including both the solution of standard linguistic tasks (morphological analysis, polysemy resolving, syntactic analysis) and applied problems (knowledge extraction from texts, abstracting, classification) based on the method of task decomposing into structure components of the ontological models system [2]. The construction of task solution is an automatic process which consists of the set of steps (fig. 2). On the first step the Task Constructor makes a decomposing of the applied problem into the ontology of tasks structure (a set of P-components). A new problem demands the decomposing by way of interaction forming the consecution of subtasks (P-components). Constructed solutions save in the base of P-realizations and might be built in other solutions.

On the second step the components of task ontology map onto set of structure from the ontologies of models (a set of S-components). The problem components representation on the set of S-components occurs in interactive mode, in which choosing a model of realization takes place (for a computational process) or interpretation of problem components in the form of the structure elements of ontologies (relations, properties).

On the third step the ontology of models components map onto set of applied ontology structure (a set of E-components). On that step a specification of chosen relation or property (as a model) and/or specification of a computational model according to the applied problem is carried out.

The final step is an automatic mapping of set of models (a set of T-components) extracted from the text on the set of structures from the ontology of tasks, the applied ontology and the ontology of models. The mechanism of mapping based on the models identification process, which establishes a correspondence between formal (examples of classes) and actual (text segments) parameters. The segments representation contains the specific set of parameters according to which the process of comparison takes place.

Using an outcomes of constructed mapping the "Problem Solver" subsystem forms the solution process as a consecution of P-components

As an illustration we take the decomposing process of applied problem solution "Functional homonymy disambiguation". The functional homonymy is resolved by the context rules method. The condition-action rules are described on certain formal language. The general form of a rule is as $IF\ B\ THEN\ X = C$, where B is a context condition, the following equality X = C means that the homonym X has a grammatical class C. Formal record of the context condition represents a disjunctive normal form where variables are substituted by names of grammatical categories. Each term of disjunction (disjunct) defines the local context of a homonym and represents the conjunction of variables (classes).

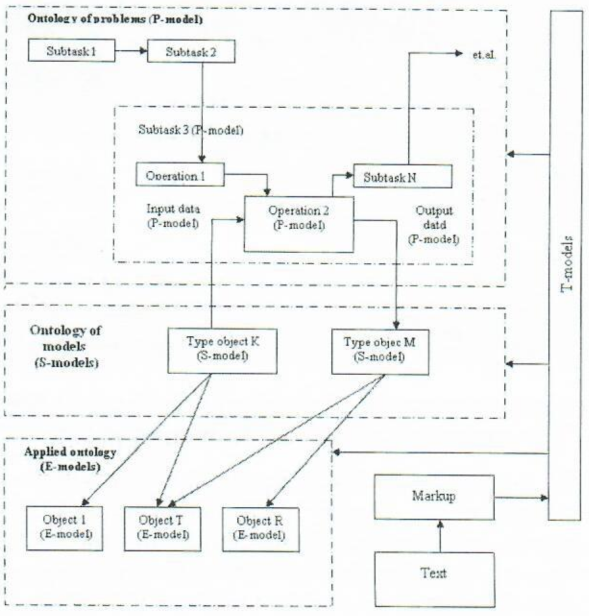


Fig. 2. A problem decomposing in the ontological system

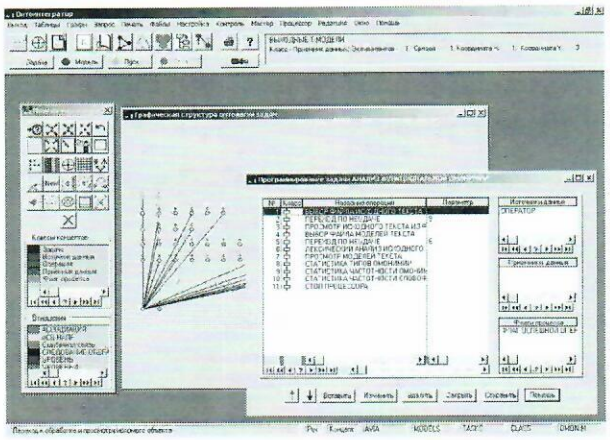
For example, there is a rule

$$\text{If } \left[X \frac{(Z) + \overline{P} + N^{*}_{p2}}{\leq 2/\otimes} \right] \Big/ \left[\frac{N^{*}_{p2}(Z)}{\leq 2/\otimes} X \right] \text{ then } X = N^{*},$$

Where X, P, $N_{p2}{}^{*}$, Z – names of the classes and at the denominator the length of context is indicated (for example, no more than 2 words on the left of X, the context might be shorten if it has special separator. The interpretation of the first disjunct is the following: if there is no a preposition and there is a noun in genitive case on the right of homonym, then the homonym belongs to the N^* class of nouns or pronouns.

Decomposing the problem of functional homonymy disambiguation connected with automatical constructing of computational procedure where an analysis of the rule disjuncts is managed by the models of their descriptions (S-components). The appropriate model sets up the properties and relations between objects of the rule and interpretation of the models which is realized according to real context of input text (on a set of T-components).

The decomposing of context rule connected with interpretation of the rule in which the appropriate methods assign a certain grammatical class to each symbol and define the order of logical test, accomplish test for agreement and other actions. Actually, the solution of the problem constructs automatically as a consecution of the procedures. It is important that a solution of the problem forms completely on basis of the ontological approach.



№	Класс	Название операции	Параметр
1		ВЫБОР ФАЙЛА ИСХОДНОГО ТЕКСТА	
2		ПЕРЕХОД ПО НЕУДАЧЕ	9
3		ПРОСМОТР ИСХОДНОГО ТЕКСТА ИЗ	
4		ВЫБОР ФАЙЛА МОДЕЛЕЙ ТЕКСТА	
5		ПЕРЕХОД ПО НЕУДАЧЕ	6
6		ЛЕКСИЧЕСКИЙ АНАЛИЗ ИСХОДНОГО	
7		ПРОСМОТР МОДЕЛЕЙ ТЕКСТА	
8		СТАТИСТИКА ТИПОВ ОМОНИМИИ	
9		СТАТИСТИКА ЧАСТОТНОСТИ ОМОНИМ	
10		СТАТИСТИКА ЧАСТОТНОСТИ СЛОВОФ	
11		СТОП ПРОЦЕССОРА	

Онтоинтегратор
Графическая структура онтологии задач
Программирование задачи АНАЛИЗ ФУНКЦИОНАЛЬНОЙ ОМОНИМИИ
ОПЕРАТОР
Вставить
Изменить
Удалить
Закрыть
Сохранить

Fig. 3. Construction of solution of the linguistic task "The analysis of a functional homonymy in the text"

Fig. 3 presents the screen form on which process of construction of solution of the linguistic task "The analysis of a functional homonymy in the text" is displayed. Construction of solution of the problem can be fulfilled both graphics, and text modes. The tabular mode is the habitual form of programming, thus links (with the following metrics) between concepts of problems ontology are presented in the form of sequence of operations (with references to the input and output data) and flags of process of performance. The graphic mode allows to handle the task as a element of problems ontology. At the left the control panel is shown by a processing graphic mode. There are buttons of operations over concepts, text equivalents and relations, and also colour lists of concepts classes концептов and relations, admissible for usage in problems ontology and defined in connected models ontology. At the present time the development of functional homonymy disambiguation module based on described technology is finishing.

5. Conclusion

In this paper a new approach connected with construction of linguistic problem solution in the "OntoIntegrator" system was overviewed. The method of decomposing the problem solution into structural components of ontological system serves as a basis for our approach. This method allows us to unify the process of constructing the solution which is useful for different classes of problems, including linguistic problems, which permit natural decomposing according to this approach. The development of general functional modules of the system, which support basic planning of the problem solution, is almost finished by the present. Within the bounds of this technology the set of applied linguistic problems solutions were developed, which also includes different classic problems of the text processing.

Acknowledgements

The research was partly supported by Russian Foundation of Basic Research (grant № 08-06-00183).

References

1. Nevzorova O. A. Ontolinguistic systems: Technologies of interaction with applied ontology // Scientific notes of the Kazan state university. Physical and mathematical sciences issue. Vol. 149. Book 2. 2007. P. 105–115. In Russian.

2. Nevzorova O. A., Nevzorov V. N. The method of decomposing the problem into structural components of ontological system // Proceedings of Int. Conference "Artificial Intelligence. Intelligent systems". AI-2009 (28 Sept. – 3 Oct. 2009, Divnomorskoe, Russia). 2009. P. 285–288. In Russian.

3. Nevzorova O. A. The development support system of visual designing of ontology "OntoEditor+" in linguistic applications // The bulletin of the Kazan State Technical university. 2006. No 3. P. 56–60. In Russian.

4. Lekavy M., Navrat P. Expressivity of STRIPS-Like and HTN-Like Planning. Lecture Notes in Artificial Intelligence, Vol. 4496 Agent and multi-agent Systems. Technologies and applications. 1st KES International Symposium, KES-AMSTA 2007, Wroclaw, Poland, May/June 2007. Germany, Springer-Verlag Berlin Heidelberg, 2007. pp. 121–130.

5. Richardson N. E. Towards inducing Hierarchical Task Network domain models for AI Planning from examples // In Proc. of Computing and Engineering Researchers' Conference CEARC06. University of Huddersfieeld, Huddersfieeld. pp. 1–5.