THE DEVELOPMENT SUPPORT SYSTEM "ONTOINTEGRATOR" FOR LINGUISTIC APPLICATIONS

Olga Nevzorova, Vladimir Nevzorov

Abstract: The article is described architecture and functionality of development support system "OntoIntegrator" for NLP.

Keywords: Natural language processing, ontological models, program tools for NLP

ACM Classification Keywords: H.3.1.Information storage and retrieval: linguistic processing

Introduction

The intelligent development support system "OntoIntegrator" is a ontolinguistic research integrated environment for NLP using complicated structured ontological models.

The system "OntoIntegrator" is a specialized systems for the decision of next main tasks: design of ontological models with free structure (supporting of large-scale of data visualization); modelling of applied tasks used natural language texts; NLP based on ontological and linguistic models.

The system "OntoIntegrator" is focused on the application development used NLP and realizes the onto-linguistic approach integrating the next main processes: new domain adaption; design of task decision based on ontological models: deep linguistic analysis; knowledge representation in the ontological models and specialized knowledge bases; modularity and extensibility of toolkits.

Applications spectrum of developed technologies is wide: text information structuring and visualization; information extraction from texts for computer-aided database updating; systems for automatic document annotation; information retrieval systems; document classification systems; knowledge extraction systems.

Architecture and functionality of development support system "OntoIntegrator"

Next important methods of software development were used for design of intelligent development support system "OntoIntegrator" which determine the efficiency of developed technological solution:

- using of Clarion development technology which represents the projected system as the system of hierarchy levels. The levels differ the degree of abstractness (database drivers, database dictionaries, application conceptual framework, basic templates of object-oriented programming, process-oriented programming).
- method of toolkits vertical integration which provides links and compatibility under control between different software tools.
- method of software tools balance which provides "necessity and sufficiency" for each tool (table and graphic ontology editors, import/export data processors, search engine, logical inference processor and others).
- method of horizontal integration of data processing models and inference models which provide data specifications compatibility.

Main functional subsystems of system "OntoIntegrator" is shown in Fig. 1.

The system includes next main functional subsystems: the subsystem "Integrator"; the ontology development subsystem "OntoEditor"; the subsystem "Text Analyzer"; the subsystem for supporting external linguistic recourses; the subsystem of ontological models.

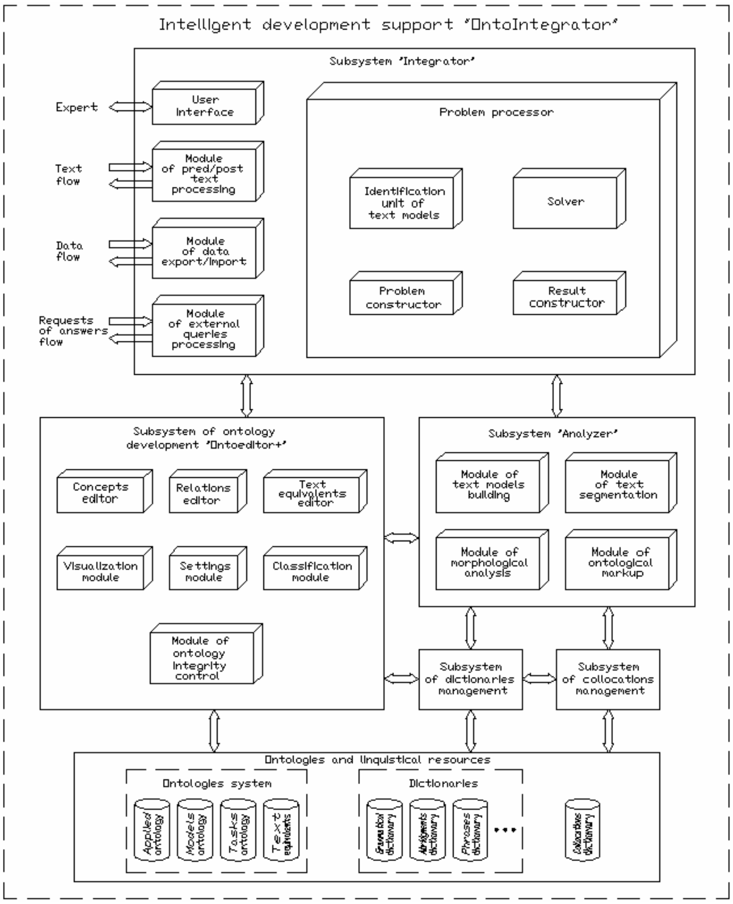

Figure 1. Architecture of system "OntoIntegrator"

A brief survey of main functional characteristics of listed systems is given below.

The ontology development subsystem "OntoEditor" [Nevzorova et. al., 2004] supports main table functions for ontology processing (editing the records; automatic record correction; supporting the using of several ontologies including mixed ontologies (for instance, the ontologies which have the common lists of relation types, classes, synonymic rows and others); import the ontologies in different data formats from external databases, gathering automatically statistics of ontology objects; ontology filtering; search the chains of concept relations with fixed properties and others).

Fig. 2 shows the graphical form with concepts hierarchy of applied ontology. Different form components show information about statistical characteristics of hierarchy including the list of ontology concepts unclassified in hierarchy; set of filtering buttons for hierarchy display; set of buttons for editing concepts hierarchy.

The subsystem "Text Analyzer" includes linguistic tools which are meant for solution different linguistic tasks: morphological analysis, ontological markup, disambiguation , segmentation, building text models for applied purposes. The solution of these tasks is based on the technologies of ontological models and external linguistic recourses interactions [Nevzorova, 2007]. In [Nevzorova et. al., 2007] the technologies of homonymy disambiguation used in the system are described.

The process of building (extraction) of text models (set of T– components) is based on the results of segmentation. The different types of segments are interpreted as the different T– components (T– components interpreted as the properties models of domain; T– components interpreted as examples models of domain; T–components interpreted as tasks models of domain).

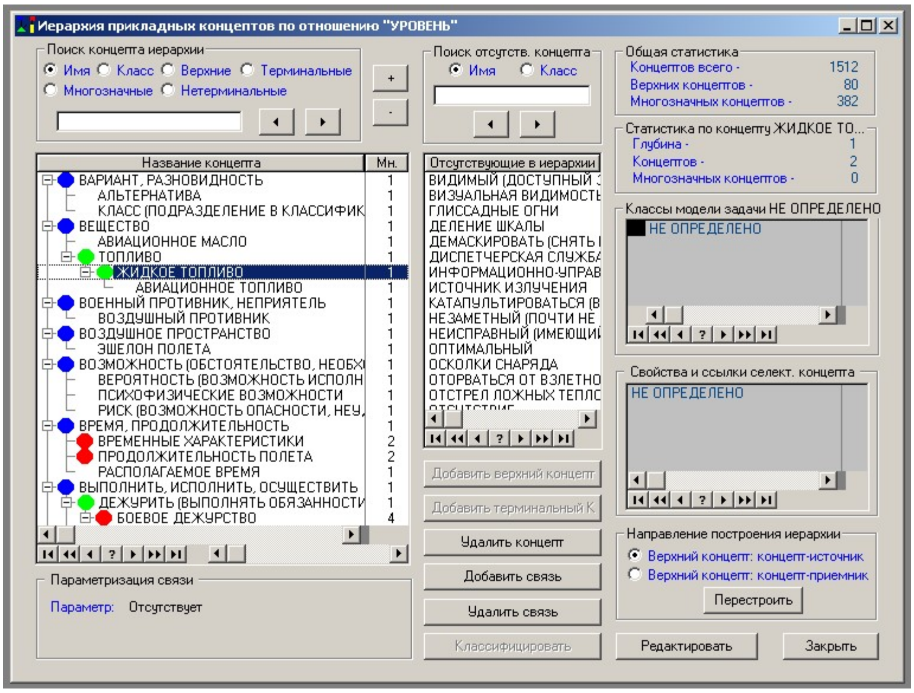

Figure 2. Graphical form of the subsystem "OntoEditor"

Fig. 3 shows the results of text analysis represented in database table structure. Information about lexical items, grammatical characteristics of word forms, references on concepts of ontological markup, results of homonymy disambiguation based on homonymy index, results of segmentation are included in table columns. The set of buttons are meant for editing table records, showing the results of segmentation and information about text statistics.

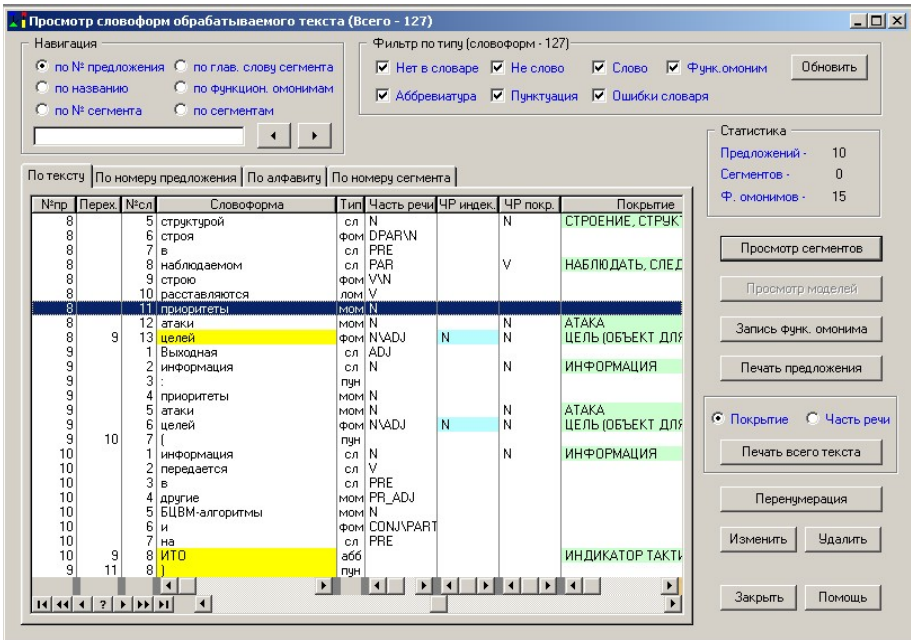

Figure 3. Graphical form of the subsystem "Analyzer"

The subsystem for supporting external linguistic recourses is meant for supporting of main linguistic recourses including marked grammatical dictionary and different specialized databases. Grammatical dictionary involves various markups included semantic markup that used in the algorithms of text analysis. The example of specialized databases is the database of collocation used for homonymy disambiguation in Russian. The collocation context allows to disambiguate homonyms included in given collocation. The method of homonymy disambiguation based on collocations models uses the search algorithms of collocation in text and assigns pre-defined grammatical characteristics to homonyms included in collocation model.

The subsystem of ontological models includes different ontologies types (applied ontology, the models ontology, the tasks ontology). The core of ontological system is the models ontology.

The concepts of models ontology are being created and edited by user in table mode or graphical mode. The concepts have the next mandatory attributes:

- name (any string). For link to T-components is used a set of text equivalents of name.
- functional class (F-class). The next F-classes are used:
  - basic class (the models of given class are built automatically in any new ontology);
  - derived (custom) class included the models of special purpose.
- color (for graphical mode);
- contents (the description of concept-model).
- semantic class. The next semantic classes are used:
  - relation;
  - property;
  - constructed by user;
  - nonsemantic;
  - referential.

All types have specific visual markings.

The concepts of tasks ontology are being created and edited by user in table mode or graphical mode. The concepts have the next mandatory attributes:

- name (any string). For link to T-components is used a set of text equivalents of name.
- functional class (F– class). The next F– classes are used:
  - operation (operation is being imported to new tasks ontology);
  - input data(input data determine information source for operations);
  - result (result determines the output result of operation);
  - task (the concept determining the structure of task decision).
- contents (the description of concept-task).

Any concept-task may be added as the subtask to another concept-task. The link between tasks is realized as the relation "operations sequence". The relation "operations sequence" is implemented with the attribute "sequence number", the operation metric is used for parameters transmission between operations.

The concepts of applied ontology are being created and edited by user in table mode or graphical mode. The concepts have the next mandatory attributes:

- name (any string). For link to T-components is used a set of text equivalents of name.
- abstractness (the attribute is used for research purposes).

Applied concepts may be classified, i.e. applied concept is being linked to some class (some concept-model). At that additional links (properties, references) arise in relation to description model.

Current version of software system "OntoIntegrator" was used to the developing of applied ontology (avia-ontology). Avia-ontology was developed for the domain which describes behavior of both an operator (air crew) and board equipment in various flight situations. Avia-ontology is a hierarchical concepts network (1600 concepts, 4700 terms) [Dobrov et. al., 2003].

The subsystem "Integrator" contains software tools for assignment of system interface settings including set of interface functions for settings of graphical mode of ontology (selection of object icons, object color; selection of functions being linked with mouse buttons and keyboard combinations). Another functional abilities support the functions of initial text processing (text corpora loading into databases, automatic statistics gathering for all linguistic objects; sentence segmentation; abbreviation recognition; building of linguistic shell of ontology and others), functions of data export/import, functions for processing of external text queries. The core of given system is problems processor that includes problems constructor, identification unit of text models, problem solver and result constructor. Functioning of problems processor is based on ontological models system.

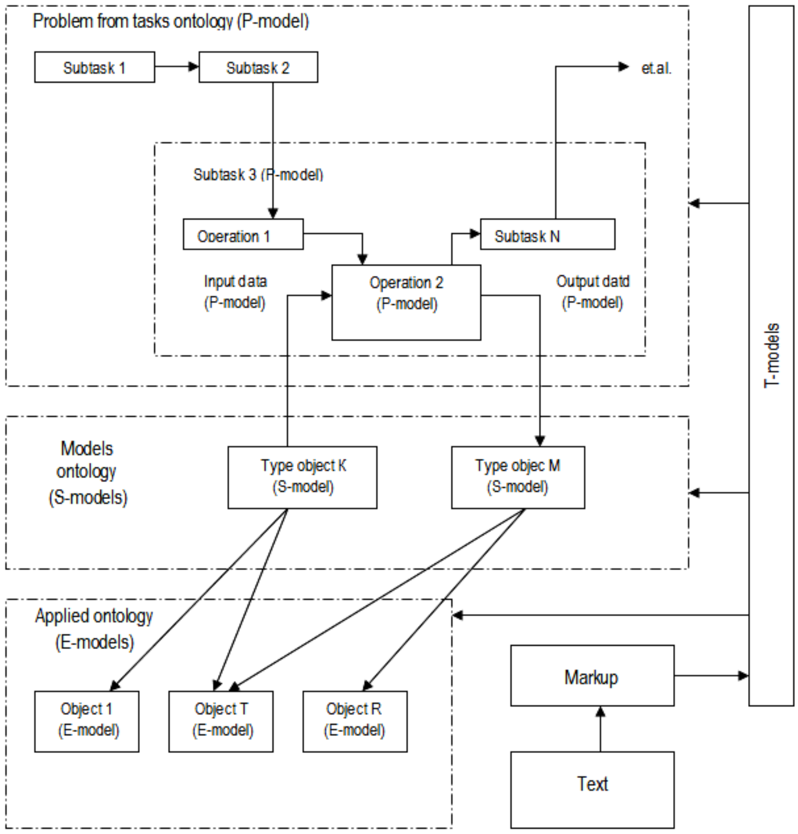

Figure 4. Task decomposition

The problems constructor performs decomposition of applied task into the structures of tasks ontology (set of P–components). The components of tasks ontology go into some set of structures of models ontology (set of S–components). The components of models ontology go again into the set of structures of applied ontology (set of E–components). Identification unit of text models establishes a one-to-one correspondence between the set of text models (set of T– components) and sets of structures of tasks ontology, models ontology and applied ontology. The problem solver forms a problem-solving process as the sequence of P– components based on built mapping.

Fig. 4 shows the method of problem decomposition into structures of ontological system. Input task (P– model) is being represented in the form of solvable subtasks sequence that are linked by consequence relation. Each subtask ((P– model) goes into the set of S– models (task operations). Each operation goes into the set of E–models that realize the operation.

Selection of P,S,E–models is based on the T-models that correlate with P,S,E–models. Recognition of T–models is implemented by the methods of linguistic analysis including text segmentation method and method of ontological markup [Nevzorova, 2007].

Reference method is used for different tasks of linguistic analysis including disambiguation, segmentation, ontological markup and others. The software tools of system "Integrator" support realization of decomposition method (development of concepts of task ontology, models ontology and applied ontology; building the mapping of various ontological levels and others).

Solution of applied linguistic problem "Analysis of functional homonymy"

The system is oriented to process linguistic tasks including as standard linguistic problems (morphology analysis, disambiguation, syntactical analysis) as well as applied problems (knowledge acquisition. annotation, classification). The solution of these problems is based on the method of problem decomposition into structures of ontological system. We will describe the using of method for applied linguistic problem "Analysis of functional homonymy" (Fig. 5).

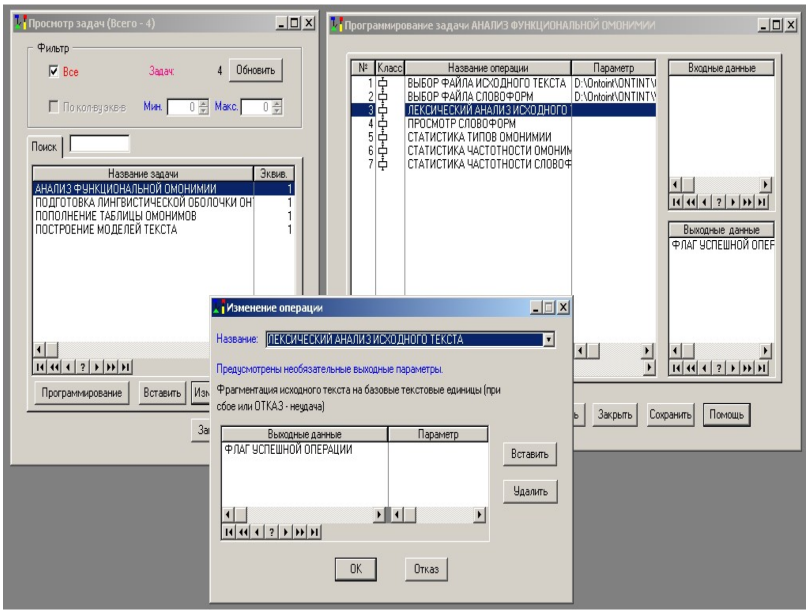

Figure 5. Development of solution for the task "Analysis of functional homonymy"

Given task is one of the basic tasks that the system "Analyzer" performs during text analysis. The objective of this task is to disambiguate functional homonymy using different applied methods. Disambiguation of functional homonymy is being implemented by group of methods [Nevzorova et. al., 2007]:

- method of disambiguation based on linguistic shell of applied ontology;
- method of disambiguation based on homonyms index of database of collocations;
- method of disambiguation based on context rules.

The solution of given problem is described as sequence of definite operations (sequence of P– components). Fig. 5 shows the sequence of given P– model: lexical analysis of text, show the word forms, flag of successful operation and others. All operations are being linked by consequence relation that is belong to S–models. Mapping of T–models into P,S,E–levels is being realized by procedures of morphological analysis. The development of solution is realized in special interactive mode in which the concepts of ontological system are being selected and the links between concepts are being set up. First two methods have the precision estimate 100 %, the third method have the precision estimate about 95% for different types of functional homonymy.

Conclusion

In the article we discussed the methodology of development of NLP–problems solution based on interactions of ontological models system. Given methodology was realized in program tools research environment "OntoIntegrator" oriented for NLP. At the present time discussed methods are successfully used for NLP including these tasks as disambiguation, text segmentation, ontological markup and others. Program tools of system "OntoIntegrator" support the realization of method of decomposition (development of ontology tasks concepts, ontology models concepts, applied ontology concepts; mapping of different ontological levels and others).

Acknowledgements

The work has been completed with partial support of Russian Foundation Basic Research (grant № 08-06-00183).

Bibliography

[Dobrov et. al., 2003] B. Dobrov, N. Loukachevitch, O. Nevzorova An Approach to New Ontologies Development: Main Ideas and Simulation Results // Int. Journal “Information Theories & Applications”. – 2003. – Vol.10. – P.98-105.

[Nevzorova et. al., 2004] Nevzorova O.A., Nevzorov V.N. Analysis of the Structures of the Linked Objects by “OntoEditor” Software Toolkit // Proc. of the 6-th Int. Workshop on Computer Science and Technologies CSIT’2004, Budapest, Hungary, 2004. – P.177-181.

[Nevzorova et. al., 2007] Nevzorova O., Nevzorov V., Zinkina J., Pjatkin N. Integral Technology of Homonymy Disambiguation in the Text Mining System "LoTA"// Proceedings of Fifth Int. Conf. Information Research and Applications, Varna, Bulgaria, 2007. – Vol. 1. – P. 129-133.

[Nevzorova, 2007] Nevzorova O.A. Ontolinguistic system: some technologies of interaction with applied ontology //Interactive systems and technologies. - Collection of scientific papers. – Ulyanovsk: UlSTU, 2007. – P. 44-51.

Authors' Information

Olga Nevzorova – Chebotarev Research Institute of Mathematics and Mechanics, Tatar State University of Humanities and Pedagogics, Kazan, Russia; e-mail: olga.nevzorova@ksu.ru

Vladimir Nevzorov – Kazan State Technical University, Kazan, Russia ; e-mail: nevzorov@mi.ru