3

# Providing Semantic Content for the Next Generation Web

Irina Efimenko[1], Serge Minor[1], Anatoli Starostin[1],
Grigory Drobyazko[2] and Vladimir Khoroshevsky[3]
[1]*Avicomp Services,*
[2]*Ontos AG,*
[3]*Dorodnicyn Computing Centre, Russian Academy of Sciences, Russia*

## 1. Introduction

Semantic Technologies and the Semantic Web (SW) as the embodiment of know-how for practical usage of these technologies are widely discussed, and it is already clear that semantic content available within knowledge portals shall lead us to a new generation of the Internet and knowledge intensive applications.

Some methods of knowledge extraction and processing for the Semantic Web have already been developed, and first applications are in use. But many pitfalls are still awaiting developers of such systems and consumers of solutions, since, in general, Tim Berners-Lee's idea that "The Semantic Web will globalize KR, just as the WWW globalized hypertext" at the technical level is still at an early stage. W3C recommendations exist for machine-readable semantics, appropriate markup and description languages, and sharable knowledge representation techniques, but implementation of the upper layers of the so-called Semantic Web tower (Berners-Lee et al., 2001) is still at R&D stage. So, we can say that the SW-era, in contrast to the Internet-age, is only just approaching, and there are many problems that still need to be solved on this path (Benjamins et al., 2002).

On the other hand, according to the Gartner Report (Cearley et al., 2007), during the next 10 years, Web-based technologies will improve the ability to embed semantic structures in documents, and create structured vocabularies and ontologies to define terms, concepts and relations. This will lead to extraordinary advances in the visibility and exploitation of information – especially in the ability of systems to interpret documents and infer meaning without human intervention. Moreover, by 2012, 80% of public Web sites will use some level of semantic hypertext to create Semantic Web documents and, by 2012, 15% of public Web sites will use more extensive Semantic Web-based ontologies to create semantic databases.

Today solutions aimed at overcoming information exposure are shifting from data gathering and processing to knowledge acquisition and usage. The aim of this chapter is to present one particular approach to this task – the Ontos Solution for the Semantic Web (OSSW). The chapter organization is as follows: in the next part we outline the main problems and tasks,

and in part 3 present an overview of the state-of-art in this domain. The main part of this chapter is devoted to the description of the OSSW and to the discussion of some knowledge insensitive applications based on this solution. In particular, in part 4 we present the Ontos instrumental platform, ontology engineering based on this platform, information extraction with systems of the OntosMiner family and the Ontos knowledge store. In part 5 we describe several intelligent Web applications focusing on ontology-driven information extraction, knowledge-based analytics, and integration of extracted knowledge with semantic Wiki. In conclusion we briefly discuss the results presented in this chapter and possible lines of future research and development.

## 2. Core Issues

As it was mentioned above, one of the main goals of the Semantic Web is semantization of the content which already exists within the classic WWW, and of the new content created each day. Significantly, the semantic representation of processed content should be suitable for usage by program agents oriented at solving customers' tasks. This means that we should have the possibility to create semantic annotations of document collections and support appropriate knowledge bases.

Research and development in this domain started almost 50 years ago and has its roots in research on artificial intelligence (Khoroshevsky, 1998; Benjamins et al., 1999; Decker et al., 1999). The results achieved in these and many other projects set down the theoretical foundations of knowledge representation and manipulation on the Internet, and brought into practice several prototypes of instrumental tools for semantic annotation of documents. Later on, the focus of R&D projects in this domain shifted towards the Internet community and the Semantic Web (Maedche & Staab, 2001). Today the main efforts of the scientific community are aimed at the development of methods and tools for automatic and/or semiautomatic ontology-driven content annotation, both on the Internet and the Deep Web.

The main issues that will be discussed in this chapter are the following:

- Emergence of Semantic Content as a new kind of "raw material" for effective use in the framework of the Semantic Web.
- Need for a new kind of intelligent systems supporting customers' activities within the Semantic Web. Such systems can be characterized as common processing platforms with, at least, three key components:
  - Knowledge Extractor based on powerful information extraction methods and tools (multilingual, effective and easily scalable).
  - Knowledge Warehouse based on the RDF, OWL, SPARQL standards (effective and scalable).
  - Set of customer oriented Semantic Services (Semantic Navigation, Semantic Digesting and Summarization, Knowledge-Based Analytics, etc.).
- Presentation of the results achieved within the first stage of our project, which is oriented at the development and implementation of the OSSW.

## 3. State-of-Art in the NLP Domain

Several approaches to solving the content semantization problem have been proposed. Within one of them, actively promoted by W3C, it is proposed to use RDF(S) (Bray, 1998)

and/or OWL (Heflin, 2004) for semantic annotation. According to Alex Iskold (Iskold, 2007; Iskold, 2008) this approach is powerful and promising but complex for understanding and usage by the bulk of specialists engaged in document annotation. Furthermore, this approach presupposes the availability of powerful and effective instrumental tools for converting existing HTML-content into RDF/OWL metadata. Another known approach,– Microformats (Çelik, 2008) ,–allows to add predefined semantic tags into existing HTML-pages with the use of simple instrumental tools. At the moment many popular sites (for example, Facebook, Yahoo! Local) use this approach to annotate events presented at their Web-pages. Significantly, appropriate instrumental tools within both of these approaches are based on Natural Language (NL) understanding.

Automatic natural language processing (NLP) has always been a major topic in the field of computational linguistics and artificial intelligence. But during the last 5-10 years a new surge of interest has occurred in this domain, not only among research teams but also within the IT industry. These R&D projects have special significance for the SW because NLP is the "bottle neck" within systems of semantic annotation.

A backward glance at R&D in the NLP domain shows (Khoroshevsky, 2002) that there have been several distinct periods of activity:

- *1960s - mid 1970s.* Development of formal models and methods, initial experience in NLP-systems prototyping.
- *Mid 1970s – 1980s.* Development of NLP methods and tools, first industry systems for NL-based communication with Data Bases.
- *Mid 1980s – mid 1990s.* Development of cognitive Natural Language Understanding (NLU) models, implementation of NLP-system prototypes driven by domain models.
- *Mid 1990s – 2000s.* Transition from linguistic analysis of individual sentence to the analysis of entire texts, development of methods and tools for NL-texts processing. First commercial systems for NL-text processing.

The main achievements of R&D during these periods include the following: the functionality of different classes of NLP-systems and their main components was specified (for the most part, these results have retained their importance to this day); theoretically and practically significant morphological models of analysis/synthesis of word forms were proposed; basic models of NL syntactic parsers were developed; a range of practical methods was proposed for the implementation of basic NL syntactic parsers; basic techniques were outlined for heuristic implementation of partial models of NL-statement interpretation; partial models of NL-text conceptual synthesis were developed; some models and methods of linguistic synthesis were suggested and tested within prototype implementations; multilevel (both, linguistic and cognitive) models of text understanding were developed; prototypes of intelligent NL-systems were implemented; several commercial implementations of NL-systems appeared that, for the most part, only mimic full-scale natural language understanding (NLU).

Key features of the modern (V) stage of R&D in this domain include the following: typically, automatic processing is aimed at real-life texts and Web-content, as opposed to artificially constructed (model) texts; multilingual document collections are processed instead of isolate (singular) texts; misprinting and\or misspelling, grammatical errors and other mistakes are present in the texts which undergo processing. Furthermore, today the goal of document processing is not simply representation of the text's meaning, but representation of this

meaning in formats suitable for effective storage, acquisition, and further usage of knowledge.

Unfortunately, computational linguistics, artificial intelligence and information technologies haven't up to now come up with powerful and effective NLP-models, and no practically significant solution to the task of full automatic processing of arbitrary NL-texts (even monolingual) from arbitrary domains has yet been proposed. This is why R&D projects in this domain are primarily focused on Information Extraction (IE) systems, Text Mining, and on Semantic Clustering/Classification systems (TREC, 2003). One of the hot topics in this regard is the development and implementation of IE-systems that are oriented at processing multilingual document collections (Poibeau et al., 2003; LREC, 2004) obtained from Internet-pages, RSS feeds and blogs, as well as corporate data bases.

Retrospective literature overview and monitoring of the relevant Internet-resources shows that the leadership in this domain belongs to US, Germany, and Great Britain, followed by Italy, France, Spain, Portugal and Japan. Interesting teams and research centers also exist in Scandinavia and other countries. It should be noted that teams from different countries (and even within one country) differ significantly in the number of members, the level of their proficiency, and quality of results. For example, in the US there are several very large research centers and corporations working in the NLP domain (for instance, Thomas J. Watson Research Center of IBM Research (TALENT, 2009), Intelligent Systems Laboratory and its Natural Language Theory and Technology group from Palo Alto Research Center (PARC, 2009) or Teragram, a division of SAS (TERAGRAM, 2009), and at the same time there exist comparatively small research teams and companies which nevertheless manage to develop very interesting solutions (for example, The Natural Language Processing Group at Stanford University (SNLP, 2009) or company ClearForest (CLEARFOREST, 2009)).

The situation in Europe is a little bit different. As a rule, R&D is represented here by university teams, and the results that these groups achieve are "transported" to the industry by appropriate startup companies. Well-known examples of this approach is the German Research Center for Artificial Intelligence (DFKI) and its Competence Center for Speech & language Technology (LT-CC, 2009), and the company ontoprise GmbH (ONTOPRISE, 2009) founded in 1999 as a spin-off from Karlsruhe University. Another example from Great Britain – The Natural Language Processing Research Group within the Department of Computer Science at the University of Sheffield (NLP-RG, 2009).

The situation in Russia differs both from the US and from Europe in the sense that the spectrum of teams and organizations working in the NLP domain is considerably different both qualitatively and quantitatively. First of all, in Russia most of R&D in the NLP domain is performed by numerous small and very small research teams (there are about 100 projects/teams/organizations with 3-5, rarely 10 members) and within a very restricted number of commercial organizations. Secondly, R&D teams in Russia are mostly theoretically oriented. There are very few examples of research teams implementing their ideas in (prototypes of) working systems. Active R&D teams in Russia are concentrated in such institutions of the Russian Academy of Sciences as the Computing Centre (Khoroshevsky, 2008), ISA (Osipov, 2006), IITP (Boguslavsky et al., 2000), PSI (Kormalev et al., 2002), Institute of Automation and Control Processes with the Computation Center of the Far Eastern Scientific Center (Kleschev & Shalfeeva, 2005), as well as in the Moscow State University (Bolshakova, 2001; Boldasov et al., 2002; Bolshakov & Bolshakova 2006), Kazan State University (Suleymanov, 1997) and several others. Beside this, interesting R&D

projects in the NLP domain have recently been initiated in several commercial organizations such as Yandex (Maslov et a., 2006) and RCO (Ermakov, 2007). The leader among these is the Russian IT-company Avicomp Services (Khoroshevsky, 2003; Efimenko et al., 2004; Khoroshevsky, 2005; Hladky & Khoroshevsky, 2007; Efimenko, 2007; Dudchuk & Minor, 2009; Efimenko et al. 2008; Malkovskij & Starostin 2009).
The scope of this chapter does not allow us to present a complete overview of the progress made by different researchers in the domain of NLP. Nevertheless, summarizing these achievements we can state that important results in the domain of information extraction from texts in different languages in restricted domains already exist today. At the same time, there are very few works concerned with NLP of texts from arbitrary domains, there are no recognizable results in the processing of multilingual document collections, and there are practically no systems that support the full technological cycle of generating semantic content from NL-texts and using it within intelligent applied systems for the Semantic Web.
In the next parts of this chapter we present and discuss in depth the OSSW, which addresses many of the problems mentioned above. This approach is the result of multiyear R&D carried out by the Russian IT-company Avicomp Services in collaboration with the Swiss IT-company Ontos AG, and the Computing Center, RAS.

## 4. Our Approach to Semantic Content Generation

### 4.1 Related work

Our activity in the domain of Semantic Technologies and the Semantic Web in context of semantic content generation is related to a number of recent research and development projects outlined below.

#### 4.1.1 Multilingual Information Extraction

Literature on information extraction methods, techniques and systems is well known (Manning & Schütze, 1999; Engels & Bremdal, 2000; Xu et al., 2002; Ciravegna, 2003; LREC, 2004). However, most of R&D projects in this domain focus either on statistical approaches to natural language processing (Manning & Schütze, 1999) or on classic information extraction approaches with the following core limitations: monolingual text processing; a moderate number of named entity (NE) types and very few types of semantic relations which are extracted; processing without control from the domain model (TREC, 2000; Poibeau et al., 2003). In contrast to that, our approach (Khoroshevsky, 2003; Efimenko et al., 2004; Khoroshevsky, 2005) is oriented at ontology-driven multilingual information extraction of a sizeable number of NE types and semantic relations, and at the representation of results in RDF(S)/XML/OWL/N3 formats. At the instrumental level our NLP engine is partially grounded in the GATE software platform (Cunningham et al., 2002), extended within the Ontos project by a powerful knowledge representation language and other linguistic modules and technological components (Karasev et al., 2003). Another important part of our NLP instrumental platform is the OntosMiner Domains Description database combined with a user interface for managing complex ontological data (see section 4.4 below).

### 4.1.2 Knowledge Management

One of the key topics in the domain of knowledge management is development and implementation of effective and scalable knowledge warehouses. A host of materials related to various aspects of such storages has appeared as a result of work performed by European and international workgroups (Beckett et al., 2001; Berners-Lee, 2000), within the proceedings of different conferences (WWW, 2003), and in open source communities (Broekstra et al., 2002; Wood et al., 2005). Some related materials are distributed by turnkey DBMS vendors, such as Oracle, etc. (ORACLE, 2007a). Furthermore, a number of semantic storages has already been developed and implemented, for example the KIM platform (Popov et al., 2004).

Generally speaking, within the Ontos project we follow the mainstream tendencies in the development and implementation of knowledge warehouses. At the same time, our prime focus is on the development of methods for aggregating knowledge extracted from documents and/or document collections, and on effective implementation of an integrated semantic RDF-store based on open source software platforms and packages, and within a commercial RDBS (ORACLE, 2007b). The basic requirements for our knowledge storage are the following: full support of all the main operations related to aggregation of semantically meaningful entities and relations, efficient pattern-matching search, and exchange with external applications in XML and/or OWL format.

### 4.1.3 Semantic Services

The idea of "Semantic Services" is widely discussed, and is sometimes seen as the next logical step for the Service Oriented Architecture (Hinchcliffe, 2005). There are many ideas and approaches in this field (Alesso, 2004; Akkiraju et al., 2005) but the mainstream seems to be the development and implementation of useful sets of knowledge intensive applications based on widgets technology (Garrett, 2005). Our approach to Semantic Services is oriented at the development and implementation of several knowledge intensive applications, such as Semantic Navigation, Semantic Digesting and Summarization, Business Intelligence, etc. (Hladky et al., 2007; Efimenko et al., 2007; Hladky, 2009).

## 4.2 Ontos Solution: An Overview

Semantic Content within the Semantic Web framework can be viewed as a new kind of "raw material", which serves as input for Semantic Services that process it and present the results to customers. According to such an understanding, the Ontos solution is oriented at the following aspects of Semantic Technologies:

- Information-intensity, and particularly:
  - Semantic content as a product.
  - Semantic services based on semantic content.
  - Interfaces for third-party development of services based on semantic content.
- Semantic Infrastructure for A2Ai (Application-to-Application integration) and B2Bi (Business-to-Business integration) solutions, including:
  - Services pertaining to the integration of existing applications and data based on semantics.
  - Semantic content as an additional information resource.
  - Semantic data warehouses with services.

The Ontos Service Oriented Architecture (Ontos SOA) and an appropriate software platform were developed to support these aspects of Semantic Technologies within the Ontos solution. The general workflow within the Ontos Solution is illustrated below.

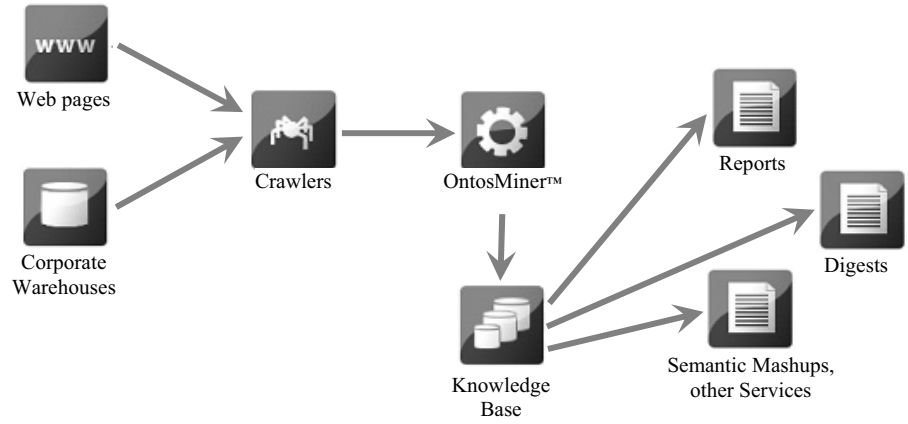


Fig 1. Workflow within the OSSW

This diagram consists of five basic components: input documents (from the WWW or corporate warehouses), crawlers, linguistic processors of the OntosMiner family, and semantic applications (reports, digests etc.).
The crawler component gathers web-pages from a pre-defined list of WWW-resources or documents from corporate warehouses, and transforms them into plain-text documents. These are then fed as input to the OntosMiner linguistic processors, which are discussed in detail in sections 4.3 and 4.4. The output of these processors is a semantic graph (in RDF/XML, OWL, Turtle or N3 format) which represents named entities and relations recognized in the input text.
This graph is then stored in the knowledge base, where incoming knowledge is integrated with existing data. This process of integration is based on algorithms of object identification which make use of the Identification Knowledge Base (IKB). Properties of the Knowledge Base and the IKB employed in our system are discussed in more detail in section 4.5.
The data in the knowledge base is accessed by various web-oriented semantic applications, which were designed to provide end users with interesting and powerful services based on semantic metadata (see section 5 of the present chapter for a detailed discussion of some of these applications).

### 4.3 Information Extraction with Systems of the OntosMiner Family

#### 4.3.1 Processors of the OntosMiner Family: Architecture and Basic Modules

Generally speaking, each IE-system of the OntosMiner family takes as input a plain text written in a natural language and returns a set of annotations, which are themselves sets of feature-value correspondences. Each annotation must have at least four features which define the type of annotation, its unique numerical identifier, and its start and end offsets (e.g. its placement in the input text). These output annotations represent the objects and relations which the processor was able to extract from the text.

Each OntosMiner linguistic processor (shortly, OntosMiner) consists of a set of specialized modules called 'resources' which are organized into 'resource chains' (Fig. 2). In this chain the resources are launched one after another and each subsequent resource has access to the output of previously launched resources. E.g. each resource modifies a common annotation set which is then fed as input to the next resource in the resource chain. Configurations of resource chains are created, edited and stored within the OntosMiner Domains Description database which is described below.

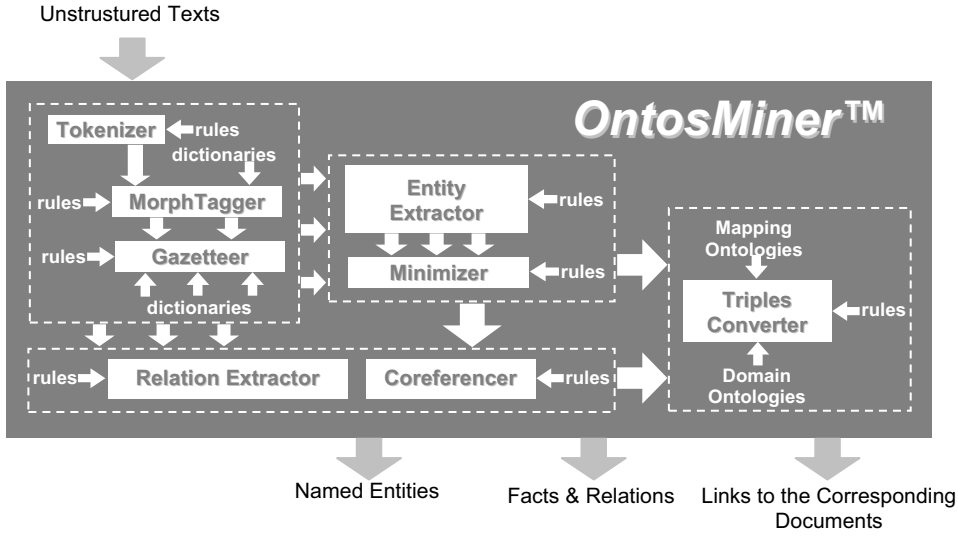


Fig. 2. Basic OntosMiner functionality and resource chain

We shall now give a short overview of the main types of resources employed in OntosMiner processors.

The first step is to determine word boundaries based on formal signs (spaces, paragraphs etc.) This is done by a resource called Tokenizer which generates a set of annotations of the type 'Token', corresponding to individual words in the input text. These annotations also contain some formal information about the corresponding words, e.g. length, distribution of uppercase and lowercase letters etc.

Next the set of Token annotations is fed to the Morphological Analyzer. This resource generates a set of annotations of the type 'Morph' which correspond to possible variants of morphological analysis for each word. Morphs include information about the word's base form and its morphological attributes (e.g. case, number, tense etc.), as well as formal features inherited from the corresponding Token annotations. One word (e.g. Token) can have several variants of morphological analysis; this is why the Morphological Analyzer often generates several Morphs with identical offsets. In certain processors we make use of statistical POS-tagging modules to reduce ambiguity in the morphological analysis.

The work of the Morphological Analyzer is based on dictionaries which store morphological information about individual words of a given language. These dictionaries can be accessed and updated via a specially developed user interface.

The set of Tokens and Morphs serves as input for the next resource - the Gazetteer. This resource annotates key words and key phrases which are later used for the recognition of named entities and relations (see below for more details). The key words and phrases are stored in special dictionaries in their base form. The developer has a possibility to semi-automatically determine the syntactic structure of a key phrase in the dictionary to reduce the risk of false recognition. This is useful when working with languages with rich inflectional morphology, such as Russian or German. For instance, if a key phrase consists of an adjective modifier X and a head noun Y the developer can indicate that the adjective must agree with the noun in certain morphological features (for instance gender and number). This means that only those sequences of adjective X and noun Y which fall under the restriction on agreement shall be annotated as a key phrase.
The annotations of key words and key phrases contain features which determine the role that these keys play in the recognition of named entities and relations, as well as the formal and morphological features inherited from the corresponding Morph annotations. They can also contain some additional information, such as the 'strength' of a specific key (see below).
These three modules - Tokenizer, Morphological Analyzer and Gazetteer - prepare the input for the two main modules of the system - Entity Extractor and Relation Extractor.

### 4.3.2 Named Entity Recognition

Named entity recognition is performed by the resource Entity Extractor. In this domain we have adopted the rule-based approach to NLP which means that named entities are identified according to rules defined by developers (Engels & Bremdal, 2000). Thus, the Entity Extractor consists of a set of rules divided into subsets called 'phases' which are applied sequentially to the annotation set. Rules from each subsequent phase have access to the output of rules in previous phases. Each rule consists of a pattern on the annotation set and a sequence of commands which define the action that has to be performed when the pattern is encountered. The pattern is written in the Jape+ language, which is an extended version of the Jape language developed by the Natural Language Processing Group at the University of Sheffield (Cunningham et al., 2002). The action parts of rules are mostly written in Java.
The main idea underlying the approach that we adopt is that named entities in natural language texts can be recognized based on two types of keys: internal and external keys (McDonald, 1996). Internal keys are key words and phrases which themselves form part of the named entity to be recognized. For instance, the key word *University* is an internal key for names of educational organizations such as *University of Michigan* or *Cornell University*. External keys, on the other hand, are not included into the named entity, but constitute the context for its recognition. For instance, job titles can be used as external keys for the recognition of persons' names, as in *Microsoft **CEO** Steve Ballmer*.
The list of possible keys for named entity recognition includes the key words and phrases annotated by the Gazetteer module, as well as annotations generated by previous phases of the Entity Extractor, and even specific features of annotations. For instance, the fact that a word begins with an upper case letter (this feature is supplied by the Tokenizer) can play a significant role in the recognition of proper names in languages like English and French.
Typically, the system of rules for the recognition of a certain type of named entity comprises several dozens of interlinked rules which 'build' the target annotations through a number of intermediate steps.

The table below contains two simple examples of patterns used for the identification of names of universities written in Jape+, and text fragments which correspond to these patterns.

| Pattern | Text fragment |
|---|---|
| {Location} *// an annotation of the type 'Location'*<br>{Lookup.majorType == "org_base",<br>Lookup.minorType == "org_edu"<br>Lookup.orth == "upperInitial"} *// a key word or phrase for educational organizations which starts with an uppercase letter*<br>{Token.string == "of"} *// the word 'of'*<br>{Token.orth == "upperInitial"} *// a word which starts with an uppercase letter* | Massachusetts Institute of Technology |
| {Lookup.majorType == "org_base",<br>Lookup.minorType == "org_edu"<br>Lookup.orth == "upperInitial"} *// a key word or phrase for educational organizations which starts with an uppercase letter*<br>{Token.string == "of"} *// the word 'of'*<br>{Location} *// an annotation of the type 'Location'*<br>(<br>{Token.string == "at"}<br>{Location}<br>)? *// an optional sequence of the word 'at' and an annotation of the type 'Location'* | University of New York at Stony Brook |

Table 1. Examples of patterns for named entity extraction

One of the main difficulties with the rule based approach that we adopt is the emergence of conflicts between different rules. For instance, one set of rules within the Entity Extractor can identify a certain text fragment as part of a person's name, while a different set of rules identifies it as part of a company name. We discovered that when the number of rules involved grows beyond one hundred, it becomes increasingly difficult and inconvenient to try to control for such conflicts within the rule system itself. This is why in OntosMiner processors we allow the rules for named entity extraction to apply freely, but complement the Entity Extractor with a special module called Minimizer which defines the policies for conflict resolution. The idea is that different rules have a varying measure of reliability (i.e. varying potential for overgeneration), and that the developer can evaluate this measure for each rule and state it as a feature of the annotation created by this rule using a common scale for all annotation types.

Thus, annotations generated by the Entity Extractor come with a feature called 'Weight' which has an integer value ranging from 0 to 100. This feature reflects the probability (as estimated by the developer) that this annotation is correct. One of the things that determine the weight of a rule is the measure of potential ambiguity of the keys employed in this rule (e.g. keys can be 'strong' or 'weak'). For key words and phrases generated by the Gazetteer this information can be tied to the key already in the dictionary.

The Minimizer resource contains a set of rules which describe different types of conflict and define which annotations should survive and which should be deleted, based on the types of annotations involved in the conflict and their weights. The resulting 'minimized' annotation set is passed on to the Relation Extractor.

### 4.3.3 Recognition of Semantic Relations

Semantic relations are certain facts or situations mentioned in the input text which relate one named entity to another, such as information about a person's employment in a company,

about a meeting between two persons, or about a contract deal between two companies. The module which is responsible for the recognition of semantic relations in OntosMiner processors is the Relation Extractor. Just like the Entity Extractor, the Relation Extractor contains a set of rules written in Jape+ and Java, grouped into a sequence of phases.

Recognition of semantic relations differs from the recognition of named entities in that named entities are expressed by compact word groups, while the keys for semantic relations can be situated quite far apart from each other within one sentence or within the whole text. This is why in developing rules for relation recognition we exploit a different strategy: we reduce the set of annotations which is fed as input to the rules, so that it includes only key words and phrases needed to identify a particular relation, and conversely, 'stop-words' and 'stop-phrases' which should never interfere between the keys. All other annotations are not included into the input and are not 'visible' to the rules. For instance, there is a pattern for the recognition of the relation 'PersonalMetting' (a relation which connects two persons which are claimed to have met on some occasion) which includes an annotation of the type 'Person', an annotation for the key verb 'meet' and another 'Person' annotation, in that order. This pattern covers simple cases like *Barack Obama met Ehud Olmert in Jerusalem*. But it is obvious that different kinds of syntactic material can interfere between the elements of this pattern (for instance, *Barack Obama met* ***the Israeli Prime Minister*** *Ehud Olmert in Jerusalem*, or *Barack Obama* ***after landing in Jerusalem*** *met* ***the Israeli Prime Minister*** *Ehud Olmert*), and it is impossible to enumerate all such potential 'interveners'. This problem is solved if only the relevant annotation types are fed as input to the discussed rule, i.e. the type 'Person' and the type of annotation for the key verb 'meet'. In this case all irrelevant syntactic material becomes 'invisible' for the rule, and the pattern works in a desired fashion. If we leave it at that, our pattern would probably lead to overgeneration, that's why certain stop annotation types are also included into the input. If in some text an annotation of this type occurs between elements of the pattern, the relation will not be generated because the interfering annotation would be 'visible' to the rule.

A semantic relation extracted by the Relation Extractor can be codified in two different ways. It can either correspond to a separate annotation with two obligatory features ('to' and 'from') which contain unique identifiers of the named entities that are connected by this relation. Such relations can have features which carry certain additional information about the fact or situation in question. For instance, the relation 'BeEmployeeOf' which relates persons to organizations where they work has an additional feature 'JobTitle' which carries information about the person's position in the organization. The offsets of such annotations are not as important as the offsets of named entities and are usually taken to be equal either to the offsets of some key word or phrase, or to the offsets of one of the related entities, or alternatively, to the offsets of the sentence where the relation was recognized.

Another way to codify a semantic relation is by means of a feature in the annotation of a named entity. Such a feature would have as its name the type of relation and as its value a unique identifier of the related named entity. Thus, instead of creating a separate annotation for the relation 'BeEmployeeOf' we could create a feature with the name 'BeEmployeeOf' in the annotation of a person, and put the unique identifier of the corresponding organization as its value. The drawback is that in this case we cannot add any additional features to the relation, but the advantage is that such a way of codifying relations is more compact and requires less storage space.

### 4.3.4 Co-reference

A distinguished type of relation is the relation 'TheSame' (also called the 'identification relation') which is established between two co-referring occurrences of a named entity within a text. The resource which establishes relations of this type is called OntosCoreferencer. This resource builds a matrix of all the relevant annotations and compares them two by two to establish whether the annotations in each pair can count as two co-referring occurrences or not. Its work is based on a set of identification rules which determine what kinds of correspondences between features of two annotations are sufficient to establish an identification relation. For instance, consider the following text fragment:

***Daimler AG*** *(DAI) Tuesday posted a worse-than-expected net loss in the first quarter as global demand for trucks and luxury cars collapsed, confirming that full-year revenue and vehicle sales will come in significantly lower than in 2008.*
"***Daimler*** *anticipates a gradual improvement in operating profitability as the year progresses. Earnings in the second quarter are expected to be significantly negative once again, however," the German automaker said in a statement, adding that it targets EUR 4 billion in cost savings this year.*

This text mentions the same company 'Daimler AG' twice, but in slightly different form. The first occurrence contains the key word 'AG' which is often found at the end of company names, while the second occurrence does not contain this ending. To ensure that an identification relation is established in such cases two conditions must be met. First, the Entity Extractor which identifies the string 'Daimler AG' as a company name should add an additional feature to this annotation which is equal to the company name without the ending. We call this feature 'MatchName'. Thus, the annotation of the type 'Company' corresponding to 'Daimler AG' should contain both the full name 'Daimler AG' and the shorter MatchName 'Daimler'.
Second, the set of rules for the OntosCoreferencer should include a rule which establishes an identification relation between two annotations of the type 'Company' if the full name of one of these annotations matches the MatchName feature of the second annotation. In our example the full name of the second occurrence is equal the MatchName feature of the first occurrence, and so the necessary relation shall be established.
Similar rules are used on the Knowledge Base level for discovering multiple occurrences of the same named entity in different texts (see below). We employ a separate set of rules within OntosMiner processors primarily for two reasons. First, identification rules within a single text can be less restrictive than similar rules operating between different texts. For instance, if we discover two occurrences 'John Smith' and 'Mr. Smith' within a single text it is very likely that they refer to the same person. Thus, in the OntosCoreferencer resource we can state a rule that matches two annotations of the type 'Person' if they have identical family name features and non-conflicting first name features. On the other hand, if we formulate such an identification rule between different texts we face a significant risk of uniting objects which refer to completely different people.
The second reason is that on the Knowledge Base level we are much more restricted by productivity issues, because rules on that level generally apply to a much larger body of data. This means that within the OntosCoreferencer resource we can formulate complex conditional rules which on the Knowledge Base level would lead to an unacceptable slump of the system's productivity.

#### 4.3.5 The Output

The final resource in the resource chain of every OntosMiner processor is the Triples Converter. This module takes as input the set of annotations created by previous modules and generates an output in the form of an RDF/XML, OWL, Turtle or N3 document. During its work the Triples Converter accesses the OntosMiner Domains Description database (see below) and replaces all the names of annotations generated by the OntosMiner processor with the names of corresponding concepts and relations of the Domain Ontology, using the mapping rules defined in the Mapping Ontology. All the OntosMiner annotations for which mapping rules have not been defined, are excluded from the output.

As was already mentioned above, the work of OntosMiner processors is defined by and based upon the information stored in the OntosMiner Domains Description database. We give an overview of its functions in section 4.4.

#### 4.3.6 Language (In)Dependence of OntosMiner Processors

Within the OSSW we have developed several OntosMiner linguistic processors which work with English, German, Russian and French languages. All of these processors can make use of common domain ontologies which allows us to unify the results of processing multilingual text collections. On the other hand, the processors have to use language specific morphological modules and Gazetteers. The linguistic rules for entity and relation extraction have much in common, but there are also important differences which are due to differences in the syntactic structure of these languages, and even to orthographic peculiarities. For instance, as it was mentioned above, the fact that a certain word starts with an uppercase letter is an important key for the recognition of proper names in English, French and Russian, but this heuristic does not work for German because common noun are also written with an uppercase letter in German texts.

### 4.4 Ontological Engineering in the Ontos Solution

During the development of OntosMiner processors it is often necessary to work with new domains, to change or to make more exact existing domains' specifications or to tune these specifications to suit the needs of a certain customer. In all these cases efficiency plays a decisive role. Similar procedures have to be applied not only to ontological domain descriptions, but also to the sets of linguistic rules and resources, and to dictionaries and thesauri. This led us to develop an instrument which would support all the mentioned activities in a convenient way. This instrument is called OntosMiner Manager.

It is well known that ontological engineering is one of the core processes in the life cycle of semantic-oriented applications. Today there exist several methodologies, technologies and tools supporting this activity (Gómez-Pérez et al., 2004; Nicola et al., 2009). An overview of the most popular tools for ontological engineering is presented, for example, in (Simperl & Tempich, 2006). An overwhelming majority of them is oriented at creating and maintaining domain ontologies and doesn't have anything in common with editing linguistic dictionaries or developing natural language processors.

However, on the conceptual level, configuring a linguistic processor or a system of linguistic dictionaries may also be viewed upon as a new domain, the complexity of which is comparable with, for instance, political or business domains. This new domain in its turn may be modeled by an ontology. For example, while describing the workflow of a linguistic

processor one can use such concepts as 'TextProcessingResource' and 'TextProcessingResourceChain'. Resources which are configured in a certain way will become instances of these concepts (e.g. Tokeniser configured to analyze the German language, Entity Extractor configured to extract organizations from French texts etc.). The list of upper concepts from dictionary ontologies includes, for instance, 'Dictionary' and 'DictionaryEntry' concepts. These ontologies also contain more specific concepts. For instance, concepts which describe the morphological categories of a given language.

Thus, a significant part of the information which determines the way an OntosMiner processor works may be represented using ontologies. All this information as a whole is called OntosMiner Domains Description (OMDD). On the physical level OMDD is the data, which is uploaded to an RDF-based triplestore (OMDD database). Ontological data in the OMDD is stored in a format which is completely compatible with OWL.

Generally speaking, OMDD is a system of ontologies which can be divided into 6 classes:

- Domain ontologies

These ontologies contain concepts and relations which are relevant for a certain domain (e.g. Politics, Business, Medicine etc.). Domain ontologies are interconnected by relations of inheritance. In OWL terms, one ontology can import concepts and relations from another ontology.

- Internal ontologies

These ontologies represent the sets of annotation types, features and possible feature values used in specific OntosMiner processors. Each annotation type corresponds to a concept in the internal ontology.

- Dictionary ontologies

These ontologies are used to store morphological dictionaries and dictionaries of key words and phrases. Each dictionary entry is linked to a concept from a certain thesaurus, which is also stored in the OMDD. Ontological dictionaries are used by the Gazetteer to recognize entities from thesauri in texts. Besides, dictionaries of a similar structure are used within a special component called OntoDix which allows end-users to add their personal instances and concepts to the set of objects recognized by OntosMiner.

- Resource ontologies

These ontologies represent sequences of text processing resources which are used by OntosMiner processors. Each resource type corresponds to a concept in a resource ontology. For instance the resource type 'JPlusTransducer' includes all the resources which are able to execute Jape+ rules.

- Mapping ontologies

These ontologies are accessed by OntosMiner processors in the course of generating the output document. Mappings ensure that concepts from the internal ontology are correctly replaced with concepts from the domain ontology.

- Other (auxiliary) ontologies

Each OntosMiner linguistic processor is defined by a group of ontologies which necessarily includes a domain ontology, an internal ontology, a resource ontology and a mapping ontology. Apart from that, it can include a dictionary ontology and one or more auxiliary ontologies (for instance, the ontology representing German morphological categories is included into the group of ontologies which defines the OntosMiner processor for the German language).

The current OMDD contains about 120 ontologies (around 2,5 million triples). Obviously, such level of complexity calls for an effective and convenient ontology management subsystem.
The core of OntosMiner Manager is an ontology editor which has the following characteristic features:

- effective work with complex multi-ontology systems
- capacity for automated ontology refactoring
- effective management of large sets of instances
- a flexible Graphical User Interface (GUI), which allows for easy automation of routine procedures
- convenient visual data representation specifically designed to work with complex graphs

OntosMiner Manager also includes the following extensions:

- a component for viewing and editing morphological dictionaries and dictionaries of key words and phrases
- a component for bulk dictionary extension

### 4.5 Ontos Semantic Knowledge Base

The Ontos Semantic Knowledge Base is one of the core components within the Ontos solution. Its main function is to provide effective storage of the RDF-graph which accumulates all the information extracted from large collections of texts by OntoMiner processors. Data in the Knowledge Base can be accessed via queries in the SPARQL query language. This task is facilitated by a special module called SPARQL Console which provides a GUI to query the knowledge base. It allows developers and knowledge managers to create and delete graphs, construct views based on SPARQL queries, export and import RDF data sets to/from files in standard RDF representation formats.
At the moment, we have two implementation of the Knowledge Base – one based on RDMS Oracle 11g and another one based on Open Source libraries and platforms for the implementation of RDF-stores.
A crucial problem in this regard is the presence of duplicate objects (i.e. objects that represent the same real world entities) within the accumulated RDF graph. For instance, if the linguistic processor identifies an object *Barack Obama* with the feature 'Status=president' in one text, and an object *Obama* with the feature 'Status=president' in another text, we need to have a mechanism that would enable us to merge these two objects, so that all the knowledge about one real world person Barack Obama would be related to a single object in our Knowledge Base. In our system this task is performed by algorithms of object identification which make use of Identification Knowledge Bases.

#### 4.5.1 Object Identification and Identification Knowledge Bases

The task of object identification is performed in several steps. First, each object which is extracted from an input text receives a set of identifiers, which are calculated based on the values of the object's own features and on the values of features of other objects that are connected with this object by semantic relations. For instance, objects of the type 'Person' may receive identifiers based on the combination of values of the following features: FirstName + FamilyName, FamilyName + Status, FamilyName + name of the organization

which is connected to the person via the 'BeEmployeeOf' relation etc. If an object has all the relevant features and relations it will receive several identifiers. Returning to an earlier example, the object *Barack Obama* with the feature 'Status=president' will receive two identifiers: one based on the combination 'Barack' + 'Obama', the other based on the combination 'Obama' + 'president'. On the other hand, the object *Obama* with the feature 'Status=president' will receive only one identifier, based on the combination 'Obama' + 'president'. Importantly, identical combinations of values give rise to identical identifiers. Thus these two objects shall have one identifier in common - the one generated from the values of 'FamilyName' and 'Status' features (i.e. 'Obama' + 'president').
Next, the identifiers of each object are compared with the identifiers of objects in the so called Identification Knowledge Bases (IKBs), and if a matching object (i.e. an object with an intersecting set of identifiers) is found, the objects are merged.
IKBs are databases which ideally contain validated objects with no duplicates. IKBs perform a dual role of filtering and merging the content generated by OntosMiner linguistic processors. There are several modes of initially building up IKBs. One possibility is to compose them manually, possibly taking as starting point all the objects extracted from a large collection of documents and then filtering them by hand, merging duplicates and removing errors. This approach guarantees high precision of the results, but it is labor-intensive and does not guarantee satisfactory recall, especially when we are dealing with a constant inflow of new content, for instance in case of processing news content. The problem is that if the set of objects in IKBs is fixed from the start based on some initial collection of texts, the system will never be able to identify objects which were not mentioned in that collection but became prominent in later texts. This is why we adopt a semi-automatic approach to composing IKBs. This means, that the initial set of objects is validated manually, but new objects which are not merged with objects from this initial set are not discarded, but placed in a secondary IKB database. Once the number of recognized occurrences of an object from the secondary IKB in different texts passes a certain threshold, it is transferred to the primary IKB. Functionally, the difference between primary and secondary IKBs is that only objects present in the primary IKB are accessible to end users via semantic applications (see below).

## 5. Intelligent Applications for the Next Generation Web

### 5.1 Own vs. Third-party Applications

The presented Ontos solution presumes two modes of access for external users: the accumulated semantic content can either be accessed via our own implemented semantic applications, or semantic content can be provided for use by third-party applications via an API.
There are three modes of access to Ontos semantic content for external systems:

- access to the output of OntosMiner which contains the results of processing separate documents;
- access to personalized content which is enriched with user-defined concepts by means of OntosDix (see above);
- access to identified content which appears as a result of filtering the content through the IKBs.

Conformity with W3C standards, flexibility and a wide range of output formats makes Ontos semantic content easy to use in external applications.
Our own solutions based on semantic content include, but are not limited to, packages for media-analysis, law enforcement, publishers, science & technology.

### 5.2 Semantic Portals for Innovative Fields

The main goals of "semantizing" NL-content are related to integrating pieces of information, identifying implicit connections, and providing the possibility to receive an object's profile, to find out trends, etc. All these issues are particularly important for innovative fields.
The OSSW underlies a number of portals for corporate customers and communities working in the fields of science and technology, including innovative fields, such as Nanotechnology, Nuclear energy, Power industry, Pharmacology, etc. The structure and functionality of these portals are similar in many respects, since users from different fields generally have common basic requirements.

#### 5.2.1 Information Sources and Domain Models

In order to carry out a full-scale analysis of different aspects of any innovative field, one should integrate information from a variety of sources of different structure and content. This arduous work is among the daily tasks of researchers and analysts working on scientific papers, project appraisal, investment and patent analysis, etc. Thus, relevant information sources can include (but are not limited to) the following ones:

- Patent collections;
- Databases with descriptions of international projects and programmes;
- Conference materials and scientific papers;
- Blogs and forums in a specific domain;
- Regulatory documents;
- Opinion letters of analytic companies;
- Internet portals, news in technology, RSS feeds.

It is also worth mentioning that the most interesting data can be extracted from multilingual document collections, allowing users, above all, to get a picture of a certain field on an international scale.
This makes it evident that the technology underlying the knowledge extraction process should be as flexible as possible in order to be valuable for the users.
The OSSW possesses the needed level of flexibility due to the key features of its architecture, such as fine-tunable crawlers, powerful ontology-driven NLP engines and easy-to-combine components, as well as ontology editing tools supporting sophisticated techniques of inheritance and mapping.
The ontological system used for knowledge extraction is based on a combination of ontologies corresponding to specific domains and information sources. This means that each particular ontology is determined by concepts and relations relevant for the domain and typical for the considered source (e.g. "Inventors" and "Assignees" for Patent analysis).
An example of a system of domain models which underlie portals for innovative fields is presented below in Table 2. Points 2-7 correspond to ontologies, which inherit the so-called 'common' ontology, which in its turn inherits a domain-independent upper-ontology.

Partially, sub-ontologies can intersect, which is supported in OntosMiner Manager by a number of corresponding methods and tools.

| № | Ontology | Description, Concepts, Relations |
|---|---|---|
| 1 | "Common" | "Basic" concepts and relations relevant for most of the ontologies in the considered domain. It can be viewed as an upper ontology specific for the domain of interest |
| 2 | Patents | Inventors, Inventions, Assignees, Agents, Key terms, Fields, etc. |
| 3 | Conferences | Events, Participants, Papers, Authors, Co-authors, etc. |
| 4 | News (specific for the field) | Mostly coinciding with the "Common" ontology; Sentiment |
| 5 | Projects | Projects, Investment, Programmes, Programme Types, etc. |
| 6 | Finance | Revenue, Shareholders, Producers, Customers, Stock information, Officers, etc. |
| 7 | Analytical research | Technology maturity, Producers, Customers, Competence, etc. |

Table 2. Example of a system of domain ontologies for innovative fields

All the domain ontologies are language independent. This means that the NLP modules for any language relevant for the project are driven by the same ontologies. Language specificity is taken into consideration at the level of linguistic rules and linguistic (dictionary) ontologies.

### 5.2.2 Semantic Portals' Functionality

In this section we discuss a particular example of Web portal created on the basis of OSSW. This portal is oriented at users working in the field of innovative technologies. It includes the following sections: News/Monitoring, Experts, Companies and Institutions, Shadow groups, Analytics, "My Objects" analysis, Geographic Information System (GIS), Graph Navigation.

**News/Monitoring**. This page is meant for on-line monitoring of media-sources which are considered relevant by the customer/community. Objects and relations relevant for the field are extracted which makes it possible to form ratings, illustrate trends, and determine the semantic focus of processed documents. A multilingual thesaurus is integrated into the page. Filtering by categories, sources, object types, etc. is provided.

**Experts. Companies and Institutions**. **Shadow groups**. For the most part, the content for these sections is related to patents, scientific papers, PhD theses, and conference materials. OntosMiner extracts information about inventors, authors and co-authors, assignees, affiliations, etc. This allows users to find experts and leaders in the domain of their interest, and to look for shadow groups of people and institutions working in the domain, based on thesauri and objects of interest. Let us consider a typical task of finding a reviewer for a paper or for a project, which is relevant, say, for investment analysts, or finding scholars of authority, which is important for young researchers. In order to solve this task, one can select terms, objects of interest, nodes of the thesaurus, etc. and a collection of processed documents, e.g. conference materials. Based on a given input, the system helps the user to find personalities with proper reputation in the domain of interest. These are selected based on the number and status of publications or inventions, on the citation index, on a ranking of semantic relevance etc.

**Analytics**. **"My Objects" analysis.** These sections provide Business Intelligence (BI) tools for presenting a variety of views on the data stored in the Knowledge Base. Pie-charts, column diagrams, matrices help users to discover trends, areas of concentration of financial, intellectual and other resources, discover lacunae, etc. Ontos tools, as well as third-party

instruments, can be used for presenting information in this section. Several standard formats such as RDF/XML, Turtle, etc. are supported for the output of the OSSW which facilitates integration with a variety of tools, and gives an opportunity to build knowledge-based analytics into the portals. "My Objects" functionality allows users to form personalized collections of objects, which are stored in user profiles, so that one can monitor their occurrence in the media and their public image (sentiment analysis is performed by OntosMiner), compare their ratings, discover the most interesting statements about them, etc.

**GIS**. **Graph Navigation.** The GIS section is designed for representing objects and facts from the Knowledge Base on geographic maps. Graph Navigation gives access to all objects and relations in the Knowledge Base, allowing users to discover connections between objects starting from an object of interest, with the possibility to filter relations by type, relevance, etc. (Fig. 3).

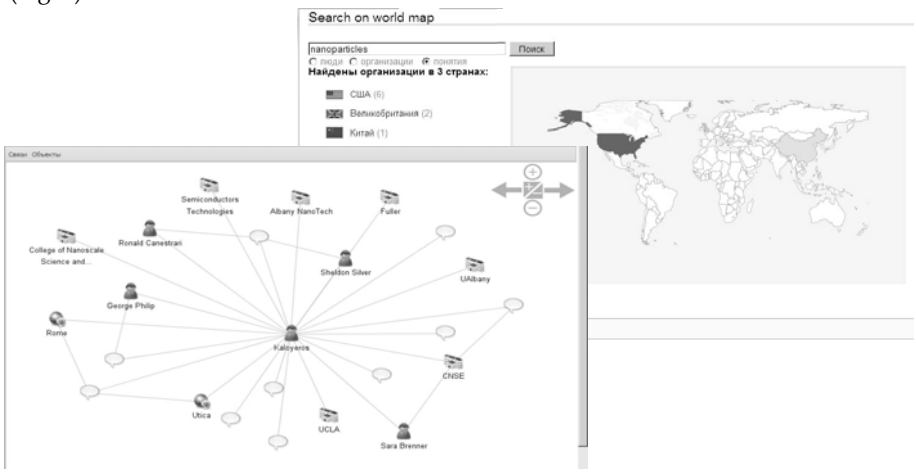


Fig. 3. Widgets within the Ontos Semantic Applications

### 5.2.3 Semantic Wiki, Bookmarking and Navigation

Ontos Portals for innovative fields are based on a wiki-engine, since one of their purposes is to create an environment for a community of experts. This functionality is in harmony with the Semantic Wiki approach. Initially the content of wiki-pages is generated in automated mode based on the accumulated semantic metadata. Later, these data can be supplemented manually by users in the standard wiki fashion. Semantic bookmarking tools are also integrated into these wiki-pages (Dudchuk & Minor, 2009). So, an object's wiki-page includes a semantic summary based on the facts present in the Knowledge Base, tags, relevant documents, graphics, etc., all generated automatically (Fig. 4). Experts with proper access rights can add their own texts and comments (which can then be processed by OntosMiner), as well as edit the semantic metadata.

Another option for the user is to install a specialized Semantic Navigation plug-in into the browser. This plug-in is able to superimpose semantic metadata upon the original content of processed web pages. This allows the user to get access to data stored in the Ontos Knowledge Base without leaving the original web page. Superficially, this looks similar to

standard hypertext, but the functionality is different. Once the user clicks on a highlighted object a navigation card appears, which delivers accumulated information on the object's features and relations, and provides the possibility to navigate through the semantic graph starting from this object. We believe that this can be viewed as an implementation of the Semantic Web, since in this case navigation takes place via a web of data, not just a web of documents (http://www.w3.org/2001/sw/).

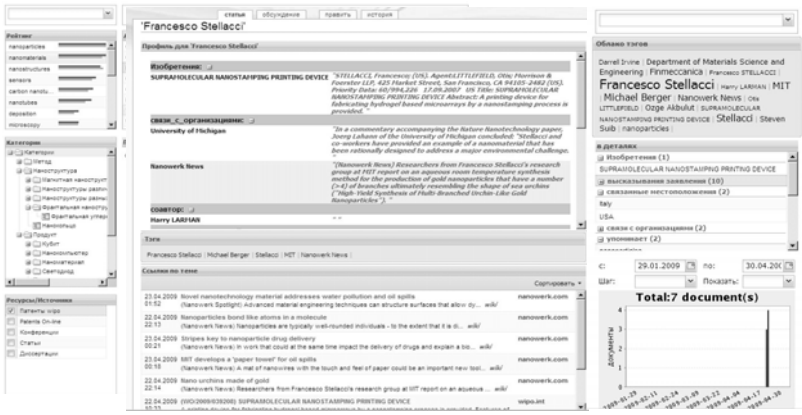


Fig. 4. Semantic Wiki and Semantic Bookmarking within the Ontos solution

## 6. Conclusion and Future R&D Trends

In this chapter we have presented the Ontos solution for the Semantic Web. This solution is based on automatic processing of large multilingual natural language text collections, gathered from Internet resources and corporate databases. This process is controlled by ontological representations of the domains of interest. The output of this analysis is represented in a standard format and stored in an RDF Knowledge Base, where data is merged and accumulated. Finally, we described a number of working Semantic Web Applications which are based on this accumulated semantic content.

Future steps in our view involve development and implementation of OntosMiner processors for new domains, and of new semantic services for the Internet community and corporative customers.

## 7. Acknowledgments

We would like to say many thanks to Alexander Alexeev, Igor Belopuhov, Maria Brykina, Philip Dudchuk, Oleg Ena, Daria Fadeeva, Polina Kananykina, Victor Klintsov, Daria Kondrashova, Alexander Pototsky, Alexander Ren, Vyacheslav Seledkin, Vitaly Volk, Vyacheslav Vorobyev, Natalia Zevakhina, Petr Zhalybin, and other specialists at Avicomp Services and Ontos AG. It is impossible to overestimate their contribution to the development and implementation of the presented Ontos solution.

## 8. References


Akkiraju, R.; Farrell, J.; Miller, J.A.; Nagarajan, M.; Schmidt, M-T.; Sheth, A. & Verma, K. (2005). Web Service Semantics - WSDL-S, Technical Note, Version 1.0, April 2005, *http://lsdis.cs.uga.edu/Projects/METEOR-S/WSDL-S*

Alesso, H.P. (2004). Developing the Next Generation Web Services - Semantic Web Services. *http://www.webservicessummit.com/Excerpts/BuildingSemanticWS.htm. A. K. Peters, Ltd.*

Beckett, D. (2001). Semantic Web Advanced Development for Europe (SWAD-Europe). *http://www.w3.org/2001/sw/Europe/reports/rdf_scalable_storage_report*

Benjamins R.; Decker S.; Fensel D. & Gomez-Perez A. (1999). (KA)2: Building Ontologies for the Internet: A Mid Term Report. *International Journal of Human Computer Studies (IJHCS).* 51(3). September 1999.

Benjamins, V. R.; Contreras, J.; Corcho, O. & Gomez-Perez, A. (2002). Six Challenges for the Semantic Web, http://www.cs.man.ac.uk/~ocorcho/documents/KRR2002WS_BenjaminsEtAl.pdf

Berners-Lee, T. (2000). XML and the Web, *XML World 2000*, Boston, *http://www.w3.org/2000/Talks/0906-xmlweb-tbl/*

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The Semantic Web. Scientific American Magazine - May

Boguslavsky, I.; Frid, N.; Iomdin, L.; Kreidlin, L.; Sagalova, I. & Sizov, V. (2000). Creating a Universal Networking Language Module within an Advanced NLP System. *In Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pp. 83-89

Boldasov, M.; Sokolova, E.; Malkovsky, M. (2002). User Query Understanding by the InBase System as a Source for a Multilingual NL Generation Module. *In Text, Speech and Dialogue*, Springer, Berlin, pp. 1-7

Bolshakova, E. (2001). Recognition of Author's Scientific and Technical Terms. *In Computational Linguistics and Intelligent Text Processing,* Springer, Berlin, pp. 281-290

Bolshakov, I.; Bolshakova, E. (2006). Dictionary-Free Morphological Classifier of Russian Nouns. *In Advances in Natural Language Processing,* Springer, Berlin, pp. 237-244

Bray, T. (1998). RDF and Metadata, June 09, 1998, *http://www.w3.org/RDF*

Broekstra, J. ; Kampman, A. ; van Harmelen, F. (2002). Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *In Proceedings of the First Internation Semantic Web Conference (ISWC'02)*. Sardinia, Italy

Cearley, D. W.; Andrews, W. & Gall, N. (2007). Finding and Exploiting Value in Semantic Technologies on the Web. 9 May 2007, ID №: G00148725. Gartner, Inc.

Çelik, T. (2008). microformats. 2008. *http://microformats.org/wiki/microformats*

Ciravegna, F. (2003). Designing adaptive information extraction for the Semantic Web in Amilcare. *In S. Handschuh and S. Staab, editors, Annotation for the Semantic Web, Frontiers in Artificial Intelligence and Applications*. IOS Press

CLEARFOREST. (2009). *http://www.clearforest.com/index.asp*

Cunningham, H.; Maynard, D.; Bontcheva, K. & Tablan, V. (2002). GATE: an Architecture for Development of Robust HLT Applications. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, July 2002

Cunningham, H.; Maynard, D.; Bontcheva, K. & Tablan, V. (2002). GATE: an Architecture for Development of Robust HLT Applications. *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, July 2002

De Nicola, A.; Missikoff, M. & Navigli, R. (2009). A Software Engineering Approach to Ontology Building. *Information Systems, 34(2)*, Elsevier, 2009, pp. 258-275.

Decker, S.; Erdmann, M.; Fensel, D.; Studer, R. (1999). Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. *In R. Meersman et al. (eds.): Semantic Issues in Multimedia Systems. Proceedings of DS-8*. Kluwer Academic Publisher, Boston

Dudchuk, P. & Minor, S. (2009). In Search of Tags Lost: Combining Social Bookmarking and SemWeb Technologies, http://www.semanticuniverse.com/articles-search-tags-lost-combining-social-bookmarking-and-semweb-technologies.html

Efimenko I. ; Hladky D. ; Khoroshevsky V. & Klintsov V. (2008). Semantic Technologies and Information Integration: Semantic Wine in Media Wine-skin, *In Proceedings of the 2nd European Semantic Technology Conference (ESTC2008)*, Vienna

Efimenko, I.; Drobyazko, G.; Kananykina, P.; Khoroshevsky, V.; et. al.: Ontos Solutions for Semantic Web: Text Mining, Navigation and Analytics. *In Proceedings of the Second International Workshop "Autonomous Intelligent Systems: Agents and Data Mining" (AIS-ADM-07)*. St. Petersburg, Russia, June 3-5, 2007

Efimenko, I.V. (2007). Semantics of Time: Identification Models, Methods & Algorithms in Natural Language Processing Systems. *Vestnik of Moscow State Regional University. Vol. «Linguistics». – № 2, 2007*. Moscow State Regional University Publ., Moscow, 2007, p.p. 179-185 (in Russian)

Efimenko, I.V.; Khoroshevsky, V.F.; Klintsov, V.P. (2004). OntosMiner Family: Multilingual IE Systems. *In the Proceedings of International Conference SPECOM-2004,* St.-Petersburg, Russia

Engels R.; Bremdal B. (2000). Information Extraction: State-of-the-Art Report, *CognIT a.s., Asker*, Norway

Ermakov, A.E. (2007). Automatical Extraction of Facts from Texts of Personal Files: Experience In Anaphora Resolution. *In Proceedings of International Conference Computational Linguistics and Intellectual Technologies (Dialogue 2007)*. Bekasovo, 30 May - 3 June, 2007. p.p. 172-178 (in Russian)

Garrett, J.J. (2005). Ajax: A New Approach to Web Applications. February 18, 2005. *http://www.adaptivepath.com/publications/essays/archives/000385.php*

Gómez-Pérez, A.; Mariano Fernández-López, and Oscar Corcho Ontological engineering: with examples from the areas of knowledge management, e-commerce and the Semantic Web, Springer-Verlag New York, LLC, 2004, 403 p. ISBN-13: 9781852335519

Heflin, J. (ed.) (2004). OWL Web Ontology Language Use Cases and Requirements, W3C Recommendation, 10 February 2004, *http://www.w3.org/TR/2004/REC-webont-req-20040210/*

Hinchcliffe, D. (2005). Is Web 2.0 The Global SOA?, *SOA Web Services Journal, 28 October 2005*

Hladky, D. & Khoroshevsky V. (2007). Semantic Technologies for Real Commercial Applications: Experiences and Lessons based on Digesting and Summarization of Multilingual-Text Collections, *In Proceedings of International Conference "Semantic Technologies 2007" (SemTech-2007)*, San Jose, USA

Hladky, D. (2009) Sustainable Advantage for the Investor Relations Team through Semantic Content, *Chapter in this Book*

Hladky, D.; Ehrlich, C.; Khoroshevsky, V. (2007). Social and Semantic Web Technologies: Semantic Navigation, Digesting and Summarization. *In Proceedings of ESTC-2007*, Vienna, Austria

Iskold, A. (2008a). The Structured Web - A Primer. *http://www.readwriteweb.com/archives/structured_web_primer.php*

Iskold, A. (2008b). How YOU Can Make the Web More Structured. *http://alexiskold.wordpress.com/2008/01/31/how-you-can-make-the-web-more-structured/*

Karasev,V.; Mishchenko, O. & Shafirin, A. (2003). Interactive Debugger of Linguistic Programs in the GATE Environment, *In Proceedings of International Workshop "Information Extraction for Slavonic and Other Central and Eastern European Languages", IESL-2003*, Borovets, Bulgaria

Khoroshevsky, V.F. (2003). Shallow Ontology-Driven Information Extraction from Russian Texts with GATE. *In Proceedings of International Workshop "Information Extraction for Slavonic and Other Central and Eastern European Languages", IESL-2003*, Borovets, Bulgaria

Khoroshevsky, V.F. (2005). Semantic Indexing: Cognitive Maps Based Approach, *In the Proceedings of International Conference RANLP-2005*, Borovets, Bulgary

Khoroshevsky, V.F. (1998). Knowledge vs Data Spaces: How an Applied Semiotics to Work on Web, *In: Proceedings "3rd Workshop on Applied Semiotics", Proceeding of National Conference with International Participation (CAI'98)*, Pushchino, Russia

Khoroshevsky, V.F. (2002). Natural Language Texts Processing : From Models of Natural Language Understanding to Knowledge Extraction Technologies, *AI News, № 6* (in Russian)

Khoroshevsky, V.F. (2008). Knowledge Spaces in Internet and Semantic Web (Part 1), *Artificial Intelligence & Decision Support, N 1 2008*, p.p. 80-97 (In Russian)

Kleschev, A.S. & Shalfeeva, E.A. (2005). Ontologies Features Classification. Ontologies and Their Classifications. *NTI, seria 2, №9, 2005*, p.p. 16-22 (in Russian)

Kormalev, D.A.; Kurshev, E.P.; Syleymanova, E.A. & Trofimov, I.V. (2002). Data Extraction from Texts. Newsmaking Situations Analysis. *In Proceeding of 8-th Natiaonal Conference on Artificial Intelligence (CAI-2002)*. FizmatLit, Moscow, 2002, p.p. 199-206 (in Russian)

LREC. (2004). *Proc. 4th International Conference On Language Resources And Evaluation (LREC 2004)*, Lisbon, Portugal, 26-28 May 2004

LREC. (2004). *Proceedings of 4th International Conference On Language Resources And Evaluation (LREC 2004)*, Lisbon, Portugal, 26-28 May 2004

LT-CC. (2009). *http://www.lt-cc.org/index-e.html*

Maedche, A. & Staab, S. (2001). Learning Ontologies for the Semantic Web, *In: Proceedings Semantic Web WorkShop 2001*, Hongkong , China

Malkovskij M.; Starostin A. (2009). Treeton system: the analysis under penalty function control. *Software and Systems* 1(85). MNIIPU, Tver, p.p. 33-35 (in Russian)

Manning, C. ; Schütze, H. (1999). Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999

Maslov, M.; Golovko, A.; Segalovich, I.; Braslavski, P. (2009). Extracting News-Related Queries from Web Query Log. *In Proceedings of the 15th International World Wide Web Conference (WWW-2006)*

McDonald, D. (1996). Internal and External Evidence in the Identification and Semantic Categorisation of Proper Nouns. *In Corpus-Processing for Lexical Acquistion*, Pustejovsky J. and Boguraev B. (eds.), pp. 21-39, MIT Press

NLP-RG. (2009). *http://nlp.shef.ac.uk/*

ONTOPRISE. (2009). *http://www.ontoprise.com*

ORACLE. (2007a). Semantic Data Integration for the Enterprise. An Oracle White Paper. *http://www.oracle.com/technology/tech/semantic_technologies/pdf/semantic11g_dataint_twp.pdf*

ORACLE. (2007b). Innovate Faster with Oracle Database 11g. An Oracle White Paper. *http://www.oracle.com/technology/products/database/oracle11g/index.html*

Osipov, G.S. (2006). Linguistic Knowledge for Search Relevance Improvement. *Proceedings of Joint conference on knowledge-based software Engineering JCKBSE'06*, IOS Press

PARC. (2009). *http://www2.parc.com/isl/groups/nltt/*

Pautasso, C.; Zimmermann, O. & Leymann, F. (2008). RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision, *In Proceedings of the 17th International World Wide Web Conference (WWW2008),* Beijing, China

Poibeau, T.; Acoulon, A.; Avaux, C.; et. al. (2003). The Multilingual Named Entity Recognition Framework, *In the EACL 2003 Proceedings (European Conference on Computational Linguistics)*, Budapest, 15-17 April 2003

Poibeau, T.; Acoulon, A.; Avaux, C.; Beroff-Bénéat, L.; Cadeau, A.; Calberg, M.; Delale, A.; De Temmerman, L.; Guenet, A.-L.; Huis, D.; Jamalpour, M.; Krul, A.; Marcus, A.; Picoli, F. & Plancq, C. (2003). The Multilingual Named Entity Recognition Framework, *In the EACL 2003 Proceedings (European Conference on Computational Linguistics)*, Budapest, 15-17 April 2003

Popov, B.; Kiryakov, A.; Ognyanoff, D.; Manov, D. & Kirilov, A. (2004). KIM - a semantic platform for information extaction and retrieval, *Journal of Natural Language Engineering, Vol. 10, Issue 3-4, Sep 2004*, pp. 375-392, Cambridge University Press

Simperl, E.P.B. & Tempich, C. (2006). Ontology Engineering: A Reality Check. *OTM Conferences (1) 2006*, p.p. 836-854

SNLP. (2009). *http://nlp.stanford.edu/*

Suleymanov, D.Sh. (1997). The semantic analyzer as a part of the embedding Teacher's model in Intelligent Tutoring Systems. *In Proceedins of the Workshop: Embedding User Models in Intelligent Applications. Sixth International Conference on User Modeling (UM97) (Chia Laguna, Sardinia, Italy, 1-5 June, 1997)*. Chia Laguna, 1997. p.p. 48-53

TALENT. (2009). *http://www.research.ibm.com/talent/index.html*

TERAGRAM. (2009). *http://www.teragram.com/*

TREC. (2000). *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, E.M. Voorhees and D.K. Harman (eds), NIST Special Publication 500-246. *http://trec.nist.gov/pubs.html*

TREC. (2003). *Proceedings of the Twelfth Text Retrieval Conference (TREC 2003)*, http://trec.nist.gov/pubs/trec12/

Wood, D.; Gearon, P. & Adams, T. (2005). Kowari: A Platform for Semantic Web Storage and Analysis. *In Proceedings of XTech 2005*. 24-27 May 2005, Amsterdam, Netherlands

WWW. (2003). *Proceedings of The Twelfth International World Wide Web Conference*. Budapest Congress Centre, 20-24 May 2003, Budapest, HUNGARY

Xu, F.; Kurz, D.; Piskorski, J. & Schmeier, S. (2002). Term Extraction and Mining of Term Relations from Unrestricted Texts in the Financial Domain, *In Proceedings of BIS 2002*, Witold Abramowicz (ed.), Poznan, Poland


**Semantic Web**
Edited by Gang Wu

ISBN 978-953-7619-54-1
Hard cover, 310 pages
**Publisher** InTech
**Published online** 01, January, 2010
**Published in print edition** January, 2010

Having lived with the World Wide Web for twenty years, surfing the Web becomes a way of our life that cannot be separated. From latest news, photo sharing, social activities, to research collaborations and even commercial activities and government affairs, almost all kinds of information are available and processible via the Web. While people are appreciating the great invention, the father of the Web, Sir Tim Berners-Lee, has started the plan for the next generation of the Web, the Semantic Web. Unlike the Web that was originally designed for reading, the Semantic Web aims at a more intelligent Web severing machines as well as people. The idea behind it is simple: machines can automatically process or “understand” the information, if explicit meanings are given to it. In this way, it facilitates sharing and reuse of data across applications, enterprises, and communities. According to the organisation of the book, the intended readers may come from two groups, i.e. those whose interests include Semantic Web and want to catch on the state-of-the-art research progress in this field; and those who urgently need or just intend to seek help from the Semantic Web. In this sense, readers are not limited to the computer science. Everyone is welcome to find their possible intersection of the Semantic Web.



**InTech Europe**
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

**InTech China**
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821