Conceptual Graphs and Annotated Semantic Web Pages

Pavlin Dobrev[1] and Albena Strupchanska[2]

[1] ProSyst Labs EOOD, 48 Vladaiska Str., Sofia, Bulgaria
pavlin@prosyst.com
[2] Bulgarian Academy of Sciences, Institute for Parallel Information Processing, 25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
albena@lml.bas.bg

Abstract. Semantic Web aims at turning Internet into a machine understandable resource, which requires the existence of ontologies, methods for ontology mapping and pages annotated with semantic markup. It is clear that the manual annotation of pages is not feasible and the fully automatic one is impossible so the current trend is creation of tools for semi-automatic page annotation. Platforms like KIM and SemTag automatically annotate pages with named entities and predefined relations but deeper annotation requires human intervention. In this paper we describe a prototype that visualizes annotations of web pages and assists the user in their verification and enrichment. The ordinary people don't know terms like ontology and OWL/RDF but they can easily understand visualization of ontology terms as a hierarchy and the respective assertions as conceptual graphs. Visualization of knowledge is always an important issue and tools that make this process more intuitive contribute to better semantic annotation and domain understanding.

1. Introduction

Semantic web aims at making web resources more meaningful to computers by adding a semantic layer to the existing presentation layer of web pages. This presupposes the existence of ontologies and pages annotated with respect to them. The issues concerning ontology availability, development and evolution as well as ontology mapping are not addressed in this paper. Rather, we emphasize on the annotation of web pages and the available tools.

The annotation is considered as one of the most effort consuming tasks and needs to be performed by especially developed tools. The volume of pages available on the web and the vagueness of natural language are the main reasons that make the manual annotation of pages not feasible even with the existence of many user-friendly tools. The current trend is semi-automatic annotation of web pages where named entities (NEs) and some simple relations between them are automatically annotated but more complex entities and relations are left to human annotators.

A prototype, which is presented in this paper, gives users the opportunity to interactively exploit the results of automatic page annotations and to add more

complex annotations. Since the production of pages annotated in similar manner requires standardization of different annotation practices, which are not present globally, our prototype addresses to particular users who will annotate pages for their specific purposes and applications. We envisage that our prototype can be successfully applied in the e-learning domain.

The paper is structured as follows. Section 2 sketches related work in the Semantic Web area and some developed tools. It also mentions attempts for the usage of conceptual graphs [16] in the new web generation. Section 3 makes overview of the prototype, its functionality and features. Section 4 and Section 5 describe in more details the prototype and give some examples for its possible applications. Our future plans and the conclusion are stated in Section 6.

2. Related Work

Many tools for ontology creation, visualization and annotation appear in the Semantic Web community. We will shortly overview some of them emphasizing on their features, which are similar or related to our prototype.

Protégé[14] is a tool for ontology creation and visualization. It offers an editing environment with several third party plug-ins. With Protégé users can construct domain ontology, customize data entry forms and enter new data. The ontology editor consist of two panels: the left one shows classes of the ontology and the right one – their respective slots (attributes and relationships). The slots have a predefined structure such as name, type, cardinality etc. and the user has to fill in forms in order to add or change something. A graphical visualization is provided by the OntoViz tab plug-in where the hierarchy is again on the left-side window and the right-side window visualizes selected class and its properties as an RDF graph.

CREAM[11] is a framework for markup, knowledge acquisition, annotation and authoring. CREAM's reference implementation is OntoMat, whose working window consists of two main panels: ontology guidance and fact browser (on the left) and document/editor viewer (on the right). The user can add instances to existing ontological classes, attributes and relationships by: *(i)* typing into the generated templates the needed information; *(ii)* markup i.e. dragging a markup piece of text to the particular class or to the corresponding entry in attribute's table or to the relation of pre-selected instance. OntoMat also provides authoring with content generation. Dragging an instance of a class/attribute/relationship from the ontology guidance and fact browser and dropping it into the document editor/viewer cause an addition of simple sentences into the document. Even though OntoMat is a user-friendly tool, the manual annotation is time-consuming and not scalable.

Manual selection of text fragments and their assignment to concepts in ontology strictly depend on the individual so the results are ambiguous. The factors mentioned above lead to the creation of tools for automatic metadata extraction.

SemTag [4] is an application that aims at bootstrapping the Semantic Web with performing automated semantic tagging of large corpora. The creators of the tool introduce a new algorithm called TBD (Taxonomy- Based Disambiguation), which is used for choosing the correct ontological label of a piece of text that is ambiguous.

For disambiguation they use a 10-word window to both sides of the label. The experiments show that although the 10-word window is typically sufficient to understand the sense of the label the human judgment regarding the placement of the label into the taxonomy remains highly ambiguous.

Knowledge and Information Management (KIM) platform [15] aims at automatic annotation of web resources and ontology population. It uses Information Extraction (IE) techniques provided in GATE to automatically annotate NEs in pages and populate the ontology. The acquired instances are stored in RDF/OWL repository (Sesame) and the annotated documents - in document store (Lucence). The KIM plug-in colours the annotated parts of the text with respect to a given ontology and it also provides browsing of annotations. Another component based on the IE system Amilcare [2], for semi-automatic creation of metadata has been realized in OntoMat too, but it is not available in its download version.

Our prototype presupposes the existence of ontologies and annotated pages. It offers visualization and editing of annotations. Its editing capabilities can be viewed as a supplement to the existing ones in the mentioned above tools. The visualization is based on page annotations and it provides some semantic services to the users. An approach that is realized in the tool Magpie [9, 10] is similar to ours in the sense that it utilizes the annotations as a base for semantic services.

Magpie provides semantic-based assistance in user's web page navigation. It is a framework supporting the interpretation of web pages that is integrated in the standard web browser. Instances on the web page are automatically annotated with the chosen ontology using the ontology-based lexicon approach. User interface components visualize entities found in the page and enable users to interact with the semantic services through contextual menus. Magpie services act as an auxiliary knowledge resource, which is at users' disposal.

The prototype presented in the paper uses Conceptual Graphs (CGs) for visualization of created annotations and for some reasoning. Several attempts for using CGs in the semantic web exist. Corese [3] is one of them and it is integrated in several applications.

Corese is an ontology-based search engine, which retrieves web resources annotated in RDF(S). A query is translated into RDF then into CG. The RDF annotations of web pages are also translated into CGs. In addition, the developers of Corese proposed a RDF Rule extension to RDF and some rules are applied before the query processing. In query processing the projection operation is used for projecting a query graph into annotation graphs. The projection operation is modified so that not only concepts, which are specialization of each others, are subsumable but also concepts that are close enough. Corese provides a user with approximate answers to queries. The semantic distance and approximation operators control that process. During the query Corese enables users to define which concepts can be approximated and which ones must be found exactly. For instance, the approximation of properties is by means of the Rdfs: seeAlso property.

The authors of [18] successfully apply CGs for mining transformation rules for semantic matching. They represent knowledge as CGs and propose an algorithm for discovering transformation rules between CGs describing semantically close assertions. The developed semantic matcher plays crucial role in the transformation search. It uses taxonomic knowledge from the ontology as a baseline for measuring

the distance. The authors also report results concerning the augmentation of the matcher with transformation rules, with human authored transformations and with their combination.

3. System Overview

Our prototype relies on the experience gained from previously developed tools ViSem [5] and CGWorld [6, 7, 8]. The main features that have been developed are in the area of semantic web annotation, visualization and editing and concern also the integration of conceptual graph formalism in the semantic web.

During the years we have implemented lots of functionality in CGWorld concerning visualization and editing of CGs, their representation in different notations and realization of CG operations. The current prototype benefits from all. In addition, we have successfully explored some Natural Language Processing (NLP) scenarios for automatic extraction of CGs from either sentences in restricted natural language [1] or sentences concerning a specific domain [17]. We apply these techniques in our prototype and the extracted CGs are added to the web pages' annotations for further use in some inference.

The prototype utilizes and relies on some resources and techniques developed in the semantic web community. For example the ontology processing is built on top of Jena's API. Jena [19] is a Java framework for building Semantic Web applications, which provides a programmatic environment for RDF, RDFS and OWL. It is an open source and grown out of work with the HP Labs Semantic Web Programme. Since the mapping between RDF and CGs is easy we use the Jena API for creation and manipulation of RDF graphs and modified it to ensure consistency of RDF graphs with the CGs formalism. Jena contains other implementations of its graph interface e.g one which stores its data in a Berkley DB database, and another which uses a relational database (in our case MySQL). So it is very suitable for maintenance of large databases and in particular for a database of CGs.

The main features of the prototype are implemented in Java using Swing graphical library. Conceptual graph operations and transformation of sentences into CGs are realized using SICStus Prolog.

4. Visualization of annotations

Concerning the visualization in Semantic Web, Kimani et al. [12] classifies existing approaches, where our approach would be sorted as generated rendering with direct manipulation interaction style. Our experimental tool provides visualization and navigation support.

When choosing a concept from the hierarchy semantic indices of the uploaded web page are calculated and text parts that are semantically anchored to the concept are highlighted. One option for visualization at this point is just to highlight the parts and the concept in both windows with the same colour. However, a natural way for connecting a web page to the ontology is to draw a line between phrases/word(s) in

the text and its corresponding ontological concepts from the graphical ontology view (see Fig.1). This way of visualization is similar to the one proposed by Ted Nelson in his Cosmic Book (http://xanadu.com/cosmicbook/). We consider this option as more intuitive in cases when the parts of the texts, which are semantically anchored to the concept from the hierarchy, are dispersed in the web page. Our prototype supports both options for visualization.

We believe that showing simultaneously a concept in the ontology hierarchy and its instances on the page could be very useful in learning scenarios. Since this way of visualization shows both the language context of a concept usage as well as its ontological environment it could be applied for supporting users'comprehension while reading web pages.

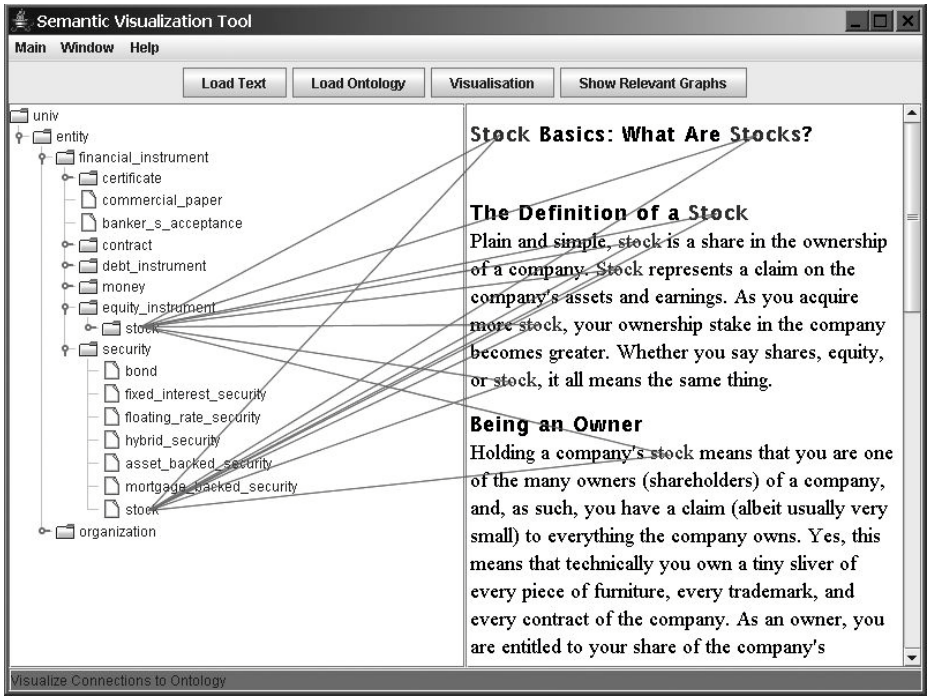

Fig. 1. Visualization of links between ontology and annotated page

Knowledge engineers would like to see visualization of concepts' properties and their relationships too. Many semantic web tools (for instance, see above the comments about Protégé) show such kind of information as RDF graph. We propose visualization as CG. On one hand the mapping between RDF/N3 and CG is relatively easy (see http://www.w3.org/DesignIssues/CG.html). Ordinary users can easily understand CGs. On the other hand CGs have querying and inference capabilities that can be further exploited. As shown in Fig. 2 a user can view assertions relevant to the chosen concept by clicking on the tab "Show Relevant Graphs". These graphs are extracted from the knowledge base of conceptual graphs, which has been developed for a previous project and extended by the usage of the current prototype.

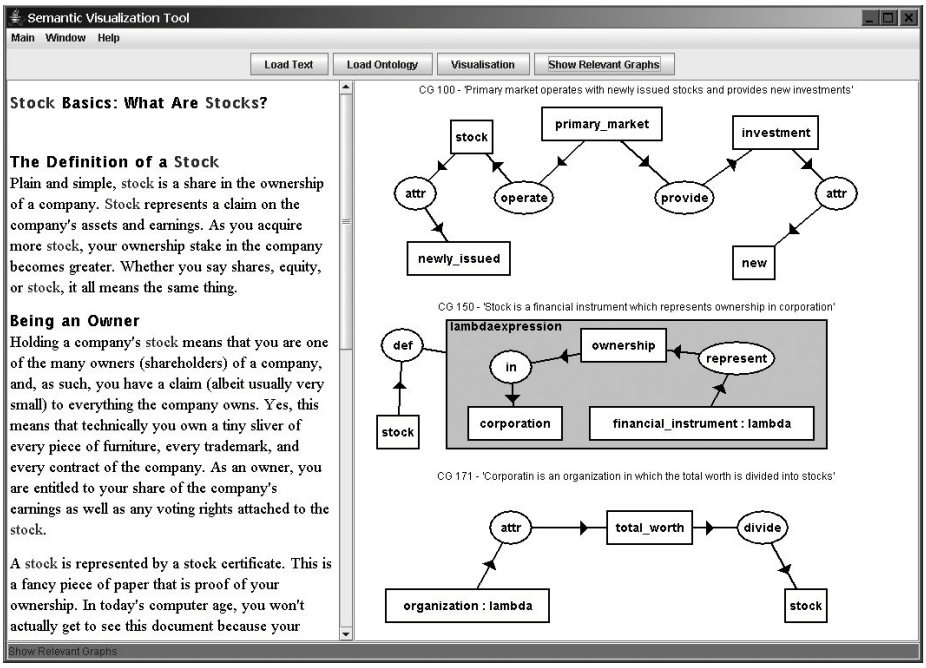

Fig. 2. Show relevant graphs

5. Annotations Editor

Users have the opportunity to edit annotations in all the tools for manual annotation of pages that are developed in the Semantic Web. They can freely choose the ontology and the assignment of words/phrases found in pages to the concepts of this ontology. The concept/individual properties are predefined and they can be included in annotation by filling in forms. The tools that use automatic processing to extract annotations of named entities allow visualization only but not editing capabilities. Our prototype presupposes the usage of an output of such tools and further provides more functional capabilities to users.

5.1. Automatic extraction of annotations

We propose the user to highlight sentence in a web page and to extract automatically a CG from it. A technique described in more details in [17] has been integrated in our prototype. We found this technique very suitable for application in a web-based content. First, it uses GATE [20] for preprocessing which has a relatively big lexicon. Second, it applies a partial parsing to produce logical forms (further converted into

CGs). For the highly varied text structure and style in the web only partial parsing can be successfully applied

A right click on the highlighted sentence triggers the appearance of a context menu with several choices. When pressing "CG Extract" a graph is extracted and it is loaded in the right window (see Fig. 3). Two possibilities are further proposed to the user.

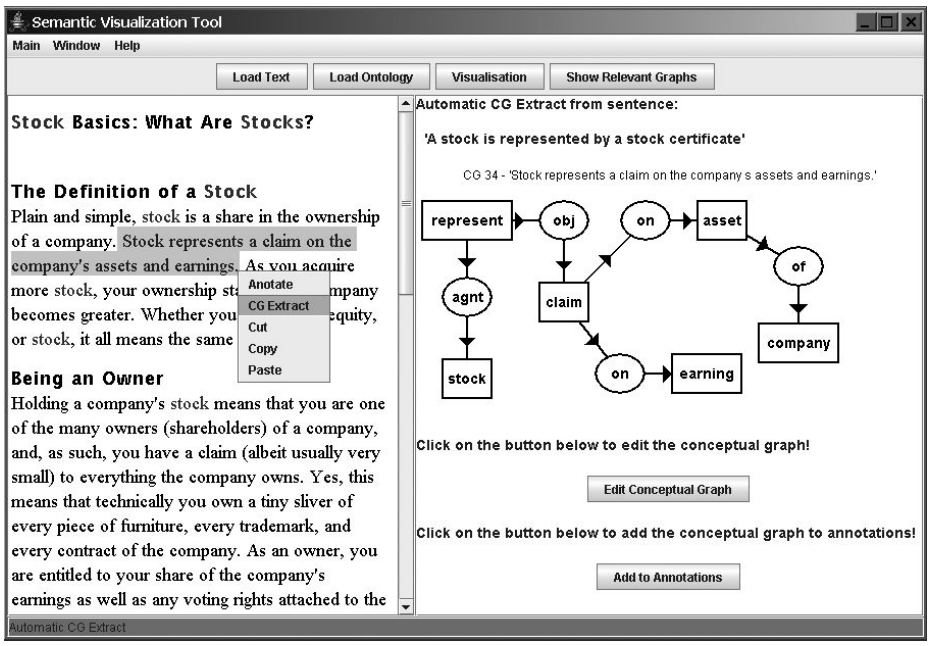

Fig. 3. Extract Conceptual Graph from a given sentence

5.2. Personalization of annotations

If a user is familiar with CG presentation formalism s/he can choose to edit the derived CG. The edit option gives him/her a possibility to add some missing parts to the CG, which are not returned as a result from the partial parsing but the user considers them as important.

Conceptual Graph Editing mostly reuses the functionality developed for the CGWorld. Concepts, relations, arcs, co-referent links and contexts are supported for editing via a simple Drag & Drop interface. Fig. 4 shows the main window of the editor. It is very easy via the simple click to add new concept, relation or to draw directly an arc between conceptual objects. An important functionality is the ability to assign any number of additional properties to the conceptual objects. These properties are related mainly to conceptual objects that represent concepts in CGs. Some possible properties are number, individual name or marker, qualifier, designator, comment etc. There is no limitation about which properties could be assigned to a

conceptual object. Such additional properties can be added and used during the annotation of web page in order to allow more knowledge information to be included and automatically processed afterwards. The editor has an excellent zooming capability. After zooming new positions of the objects are visualized i.e all dimensions are re-computed. This feature is very useful for editing large conceptual objects.

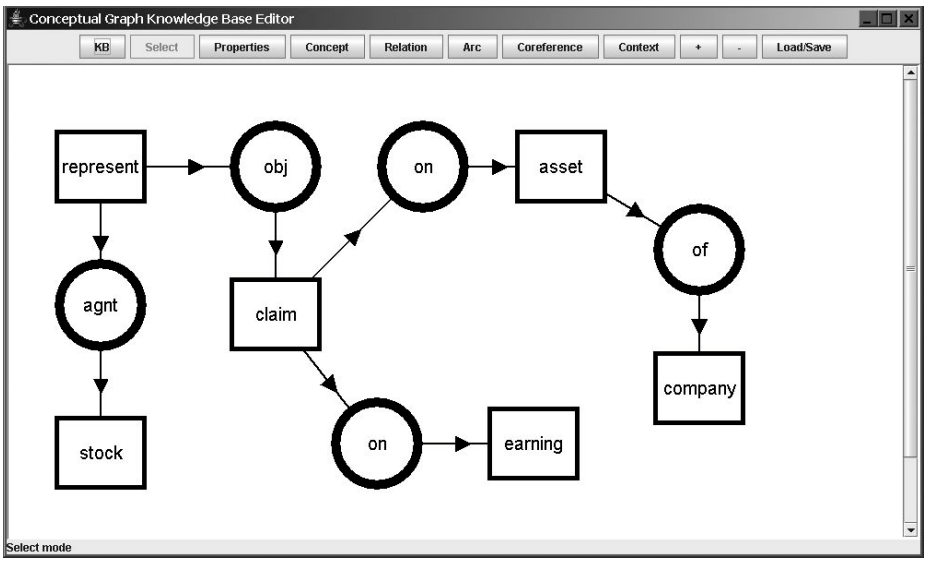

Fig. 4. Conceptual Graph editor

The application scenario we have in mind for extracting and editing of CGs in a prototype is again in the e-learning domain where teachers can choose important sentences and enrich web pages annotations.

The automatically extracted CGs might look rather complicated and not so easily understandable by a user who is non-familiar with CG formalism. As a result of parsing, all verbs are presented as concepts and all their valency roles (also called thematic roles) as relations. This kind of representation makes the reading of a CGs difficult due to the following reasons:

- a user has to know the semantics of valency roles;
- the size of CGs grows rapidly with the number of verbs and prepositions therefore capturing the meaning of a CG becomes quite effort consuming.

We have decided to use the type contraction operation in order to propose more readable CGs to the users. The previous realization of this operation takes as input a graph ID and a concept for which a type definition exists. In our case concepts are not known in advance moreover more than one concept can be replaced by respective definitions in one CG. The algorithm for type contraction has been modified to find suitable concepts and to apply contraction operation on several concepts in a CG.

Correctly identifying the semantic roles in a sentence is crucial for its interpretation. Moreover in order to develop more general natural understanding systems, broad coverage semantic resources have been created in several projects e.g. FrameNet (http://www.icsi.berkeley.edu/~framenet/) and PropBank [13]. Such resources describe a word, especially a verb by the number of arguments and their syntactic categories that can be filled by other words which are connected with the described word. Some of them contain also the semantic categories of those constituents. For instance, looking at the verb "sell" in a PropBank the following information can be found:

SELL - [*Arg0*: ***seller***]
[*Arg1*: ***thing sold***]
[*Arg2*: ***buyer***]
[*Arg3*: ***price paid***]
[*Arg4*: ***benefactive***]

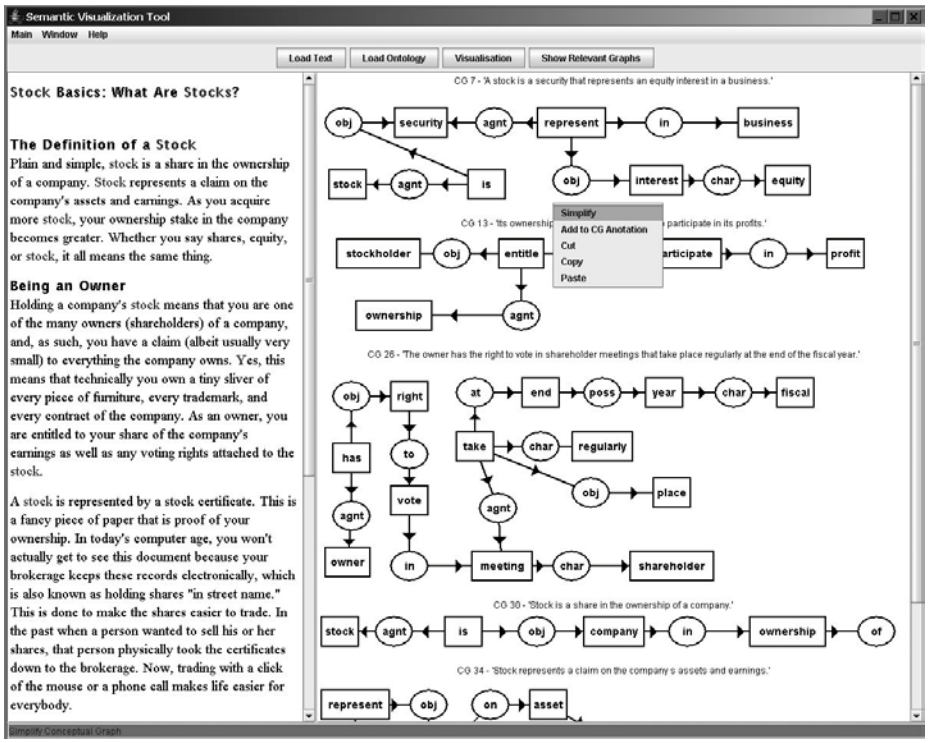

Fig. 5. Simplify Conceptual Graph

Some information that relates these roles with their syntactic categories (e.g. "*Arg0* is an *agnt")* can be found too. The example mentioned above has the purpose to show that such information about words exists. Having this information, the type definitions of words i.e their conceptual structures can be extracted. Our verbs' definitions and their roles are based on PropBank and a valency dictionary.

In [16] Sowa proposed a classification of thematic roles and types of their participants. Such classification makes reasoning more effective as specialization and generalization can be applied not only to concepts but to relations too. We plan to use this classification in our prototype when making reasoning on the CGs in the annotated pages.

Fig. 5 shows a user interface to the type contraction operation. It is very simple and intuitive. When a user clicks on a graph, a context menu appears. Choosing the option "simplify" s/he will receive a simplified version of the graph. The result is shown on Fig. 6.

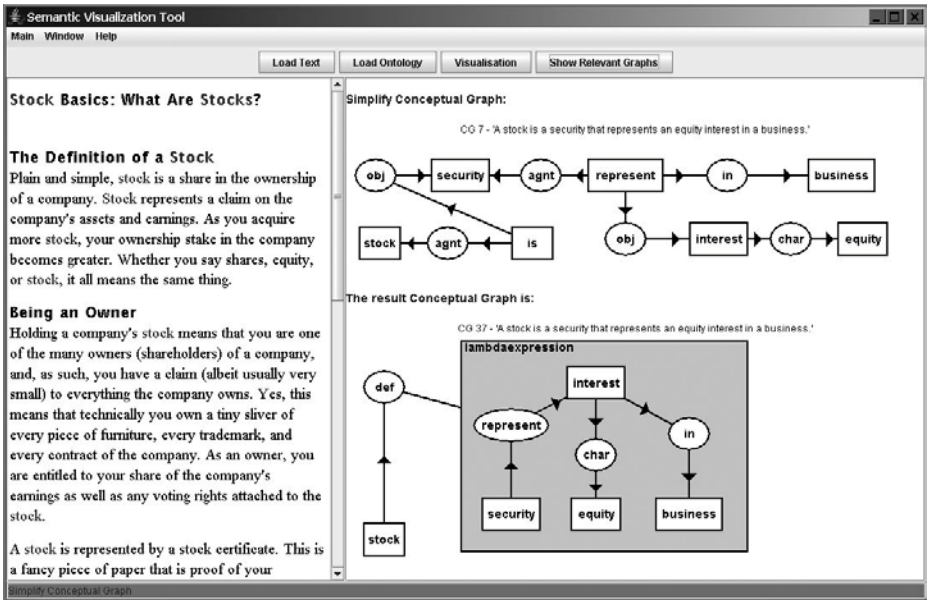

Fig. 6. Result from simplify Conceptual Graph

At the moment we have implemented type contraction on very simple definitions. Most of them have the following structure:

typedef *"verb"* is

[AgentType] <- (agnt) <- [verb] -> (obj) -> [ObjectType].

The example on Fig. 6 concerns the type contraction of the verb "be" used for definition of something. This verb and its synonyms form a special group to which additional rules are applied after performing the type contraction operation. An additional step is required because we have chosen to represent a definition of a concept as:

[Concept] - > (def) -> [Concept: CG].

The second concept is a complex one and shows particular restrictions on a genus of the defined concept.

6. Conclusion and Further Work

Manually annotating large amounts of pages is an arduous work. To master this process, techniques for semi automatic annotations are currently under development. The Semantic Web community realizes that NLP approaches are crucial for the success of the new web generation and such approaches are integrated in recently developed tools. IE techniques have been successfully applied so far for annotation of individuals and ontology population. Our prototype shows a possible subsequent step in the acquisition of knowledge structures from textual content of web pages.

The proposed way for visualization i.e. the visual connections between ontology and text parts, can be extended to cover the conceptual graph visualization. The other possible extension in this area is not only to show the connections between entities but also to allow visualizing of other relevant information like concept properties, annotation information, etc. Our research in the area of better visualization of semantic web knowledge will continue in this direction.

The integration of CGs in the web pages annotation enables better visualization and easy annotations' editing and enrichment. The inference capabilities of CG formalism together with the semantic web resources and technology can be applied to extend the current semantic search.

In contrast to the existing annotation tools, our prototype:

- supports automatic extraction of elaborated knowledge structures;
- allows annotators to edit the extracted formal structures;
- visualizes assertions about concepts in a way, which is intuitive for understanding.

The main application of the prototype is to assist annotators in the annotation process making it easier and allowing deeper annotation. We also believe that it can serve as a base for a practical application that benefits from deep semantic annotations and allows exploitation of different scenarios.

Acknowledgement

We would like to thank to our advisor Galia Angelova for her consistent support, guidance and encouragement during the years.

The work presented in this paper is partially supported by the project BIS 21++ funded by the European Commission in FP6 INCO via grant 016639/2005.

Reference

[1] Boytcheva, Sv., Dobrev, P. and G. Angelova. *CGExtract: towards Extraction of Conceptual Graphs from Controlled English*. In: G. Mineau (Ed.), Conceptual Structures: Extracting and Representing Semantics, Contributions to ICCS-2001, the 9th Int. Conference on Conceptual Structures, Stanford, California, August 2001, pp. 89-116.

[2] Ciravegna, F., Dingli, A., Petrelli, D. and Yorick Wilks: *Document Annotation via Adaptive Information Extraction*. Poster at the 25th Annual International ACM SIGIR Conference on

Research and Development in Information Retrieval August 11-15, 2002, in Tampere, Finland.

[3] Corby, O., Dieng-Kuntz, R. and Catherine Faron-Zucker: Querying the Semantic Web with Corese Search Engine. ECAI 2004: 705-709

[4] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran,, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J. and Jason Zien. *SemTag and Seeker: bootstrapping the semantic web via automated semantic annotation*. In Proceedings of the Twelfth International Conference on World Wide Web, 2003

[5] Dobrev, P., Strupchanska, A. and G. Angelova. *Towards a Better Understanding of the Language Content in the Semantic Web*. Lecture Notes in Artificial Intelligence, Volume 3192, Aug 2004, pp. 267-276

[6] Dobrev P.and K. Toutanova, *CGWorld - Architecture and Features*, ICCS 2002, Borovets, Bulgaria, July 2002, Lecture Notes in Computer Science 2393 Springer 2002, ISBN 3-540-43901-3

[7] Dobrev, P., Strupchanska, A. and K. Toutanova. *CGWorld - from Conceptual Graph Theory to the Implementation* , ICCS 2002 Workshop, July 2002, Borovets, Bulgaria, http://www.lml.bas.bg/iccs2002/acs/CGWorld2002.pdf.

[8] Dobrev, P., Strupchanska, A. and K. Toutanova. *CGWorld-2001 - new features and new directions*, ICCS 2001 Workshop, July 2001, Stanford University, USA http://www.cs.nmsu.edu/~hdp/CGTools/proceedings/papers/CGWorld.pdf

[9] Domingue, J.B.Dzbor, M., Motta, E. *Collaborative Semantic Web Browsing with Magpie*, In Proc. of the 1st European Semantic Web Symposium (ESWS), May 2004, Greece

[10] Dzbor, M., Domingue, J. and Motta, E. Magpie - Towards a Semantic Web Browser. 2nd International Semantic Web Conference (ISWC2003) 20-23 October 2003, Sundial Resort, Sanibel Island, Florida, USA http://kmi.open.ac.uk/projects/magpie/main.html

[11] Handschuh, S. and S. Staab. *CREAM: CREAting Metadata for the Semantic Web*. Computer Networks: The International Journal of Computer and Telecommunications Networking, Volume 42 , Issue 5, 2003, pp. 579 – 598

[12] Kimani, St., Catarci, T., and I. Cruz. *Web Rendering Systems: Techniques, Classification Criteria and Challenges*. In Vizualizing the Semantic Web Geroimenko, V.Chen Ch. (Eds.), Berlin: Springer 2002, pp. 63 – 89.

[13] Kingsbury, P. and M. Palmer. *From Treebank to PropBank*. 2002. In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002), Las Palmas, Spain.

[14] Noy, N., Sintek, F., Decker, M., S., M., R. W., Fergerson and M. Musen. *A. Creating Semantic Web Contents with Protege-2000*. IEEE Intelligent Systems 16(2):60-71, 2001.

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

[16] Sowa, J. Conceptual Structures: *Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1984

[17] Strupchanska, A., Yankova, M. and Sv. Boytcheva. *Conceptual Graphs Self-Tutoring System*, In Proc. ICCS 2003, July 2003, LNAI 2746, Dresden, Germany, pp. 323-336

[18] Yeh, P., Porter, B., and Ken Bakker. *Mining Transformation Rules for Semantic Matching*. Proceedings of the Workshop on Mining Graphs, Trees and Sequences (MGTS'04). Pisa, 83-94.

[19] JENA, see http://jena.sourceforge.net

[20] GATE, see http://gate.ac.uk/