

CGWorld – from Conceptual Graph Theory to the Implementation

Pavlin Dobrev¹, Albena Strupchanska² and Kristina Toutanova³

¹ProSyst Bulgaria Ltd., Sofia, Bulgaria

pavlin@prosyst.com

²Linguistic Modeling Lab, CLPP, Bulgarian Academy of Sciences, Sofia, Bulgaria

albena@lml.bas.bg

³Stanford University, Department of Computer Science, Stanford, CA, USA

kristina@cs.stanford.edu

Abstract. This article summarizes the authors' experience in implementing CGWorld - a web-based workbench for distributed development of a knowledge base of conceptual graphs, stored on a central server. The conceptual graph theory is discussed and the implementation of it in CGWorld is presented.

1 Introduction

CGWorld was first introduced at ICCS 2000 [5]. Future development was presented at ICCS 2001 [3, 4] and will be presented in ICCS 2002 [1]. All these papers argue about the architecture and the software design of the application.

CGWorld was used to develop a Knowledge Base (KB) from the financial domain [14]. This KB is an excerpt from the KB of the LARFLAST (LeARning Foreign LAnguage Scientific Terminology¹) Project. Conceptual Graphs (CGs) are used as a knowledge representation core in the complex language-learning environment defined in LARFLAST [6]. In [14] you can find the type hierarchy and Display, CGIF and CGPro forms of the CGs in this KB.

The main goal of this paper is to discuss some of the features of CGWorld that are specific to Conceptual Graph and to show how they are linked to the Conceptual Graph Standard [7].

2 Conceptual Graphs - Representation and Features in CGWorld

As defined in [7] a conceptual graph (CG or graph) is an abstract representation of logic with nodes called concepts and conceptual relations, linked together by arcs.

¹ INCO Copernicus'98 Joint Research Project #977074

They express meaning in the form that is logically precise, humanly readable, and computationally tractable. All these features make them a natural choice for many applications [8,9,10,11,12,13,16] that use them as a knowledge representation format. In CGWorld conceptual graph is any collection of concepts and relations linked by their appropriate arrows or co-referent links.

The graphical CG Editor has been implemented as a Java applet and provides easy to use Drag & Drop interface for definition and manipulation of a conceptual graphs knowledge base. After it is started, the Editor can load and save conceptual objects from the server from which it has been loaded.

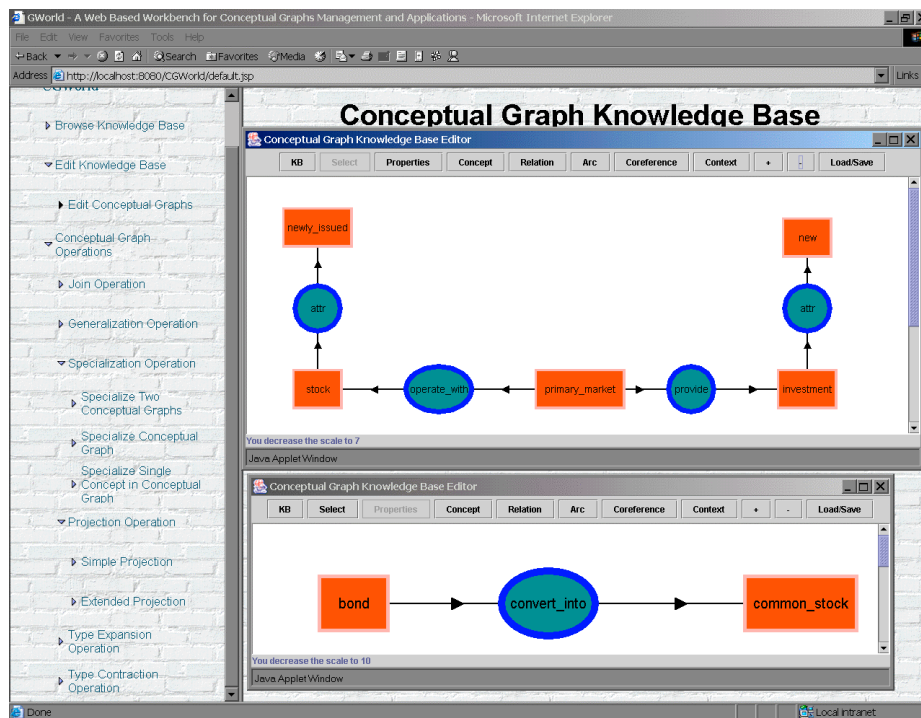


Fig. 1. CGWorld main interface

Fig. 1 gives an example of a visual interface of CGWorld. It shows the main browser window of CGWorld and two graphs opened for editing. The first graph represents the sentence “Primary market operates with newly issued stocks and provides new investments.”. The second one represents “A bond is converted into common stock.”. Both graphs are from the financial KB in [14].

The following paragraphs give a description of the main features in CGWorld that the authors consider useful for CGs application.

Portable across all platforms. The visual interface of the system is a set of HTML pages and applets. It has been developed and has been tested with the most popular browsers - Netscape Navigator and Internet Explorer. The server part of CGWorld is a set of enterprise components that follow the J2EE standard. The portability of Java language “write once, run everywhere” transfers to the application. It can be used on different hardware platforms and operating systems.

Database persistence. Using a relational database as a persistent storage allows large number of CGs to be handled by the system.

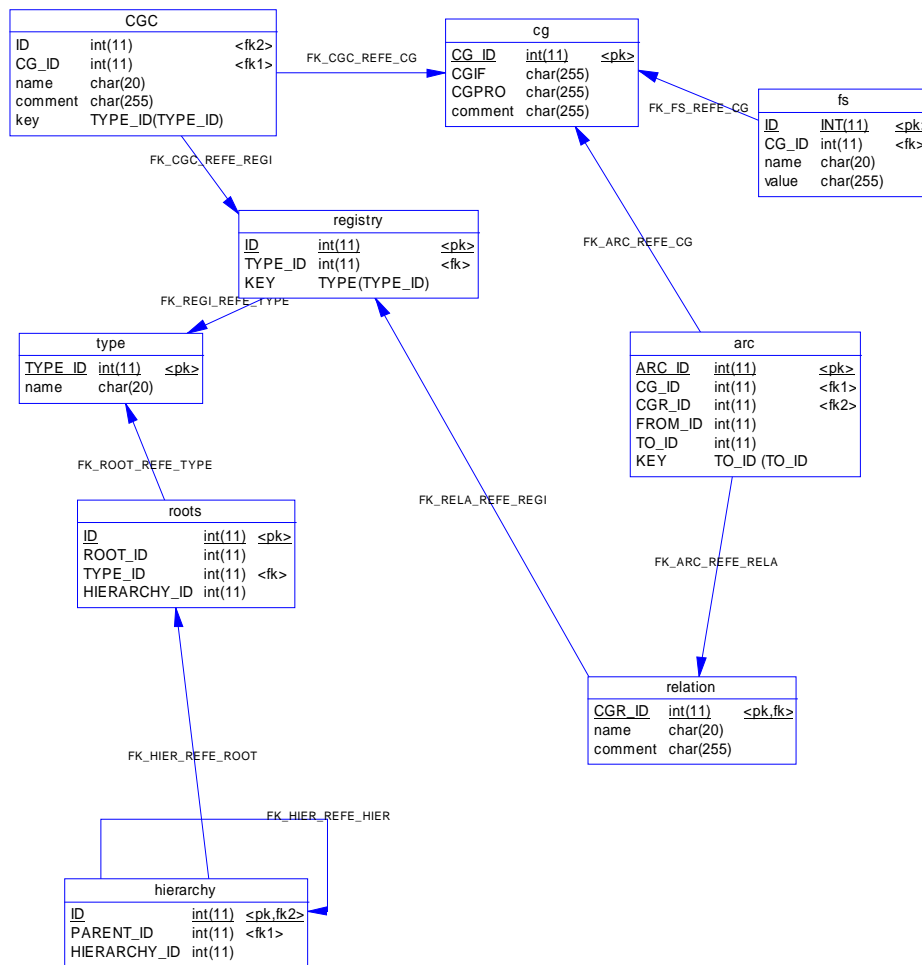


Fig. 2. Data model

Standard way to access the data. A set of Entity Enterprise JavaBeans represents persistent objects that are used to store concepts, relations, contexts, referents, arcs and information about the KB. This allows the maintenance of large amounts of data.

The control of the data integrity is performed by the built-in mechanisms for transaction maintenance.

Distributing of application. The use of Enterprise Java Beans enables the manipulation of larger amounts of data and increased numbers of concurrent users. The stated above features allows distributed acquisition and editing of CGs knowledge bases. Applications developed on top of J2EE can be distributed on several computers because most J2EE compliant application servers provide this feature.

Any number of graph windows may be opened for editing. An example of this is given at Fig. 1. In this way the user has the opportunity either to compare different CGs or to visualize them easily.

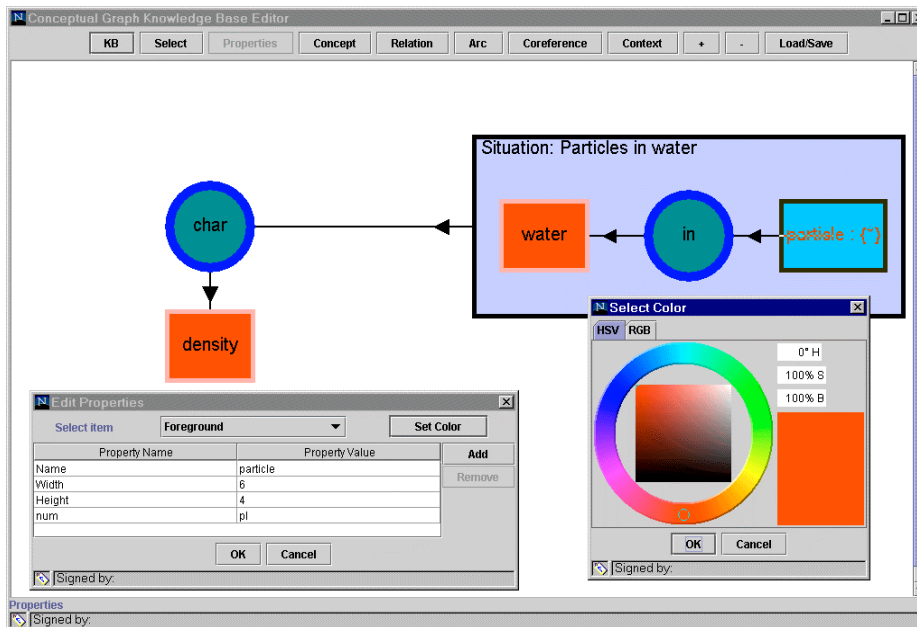


Fig. 3. Conceptual Graph Editor Overview

Drag & Drop interface. Concepts, relations, arcs, co-referent links and contexts are supported for editing via a simple Drag & Drop interface. Fig. 3 shows the select mode of the editor. This is a mode for editing the position and the names of the conceptual objects. After selecting an object and pressing the mouse once, the object is highlighted. As long as the mouse button is down, the object remains selected and it can be moved along the working pane. The object's movement follows the mouse's movement. After the mouse button is released, the new object's position is remembered and the new positions of the incoming and outgoing arcs are automatically recomputed. Double clicking invokes an edit filed in which the object's name can be edited. Pressing the 'Enter' button causes acceptance of the changes made

so far. If an object's name is erased and then this change is confirmed (with 'Enter'), then object is deleted from the working pane.

Ability to customize the color, the position and the size of conceptual objects. Fig. 3 and Fig. 4 show that every conceptual object can be chosen directly by the mouse and also its color, position and size can be changed.

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, comment etc. There is no limitation about which properties could be assigned to the conceptual object. Some of the properties have a special meaning and change the text that is displayed on the concept. Currently CGWorld interpret the following properties:

- Number. The name of the property is num. When the value is 'pl' (from plural) the text that is displayed is "name : {*}".
- Designator. The name of the property is type. When the value is 'def' the text that is displayed is "name : #" and it represents the definite concept. When the value is 'quest' it means that the concept is a subject of quest in some query.
- Individual marker or name. The name of the property is refID. The text that is displayed is "name : value".
- Quantifier. The name of the property is quant. When the value is 'every' the text that is displayed is "name : every". When the value is 'lambda' the text that is displayed is "name : lambda" showing that the concept is the genus concept in some conceptual graph representing type definition

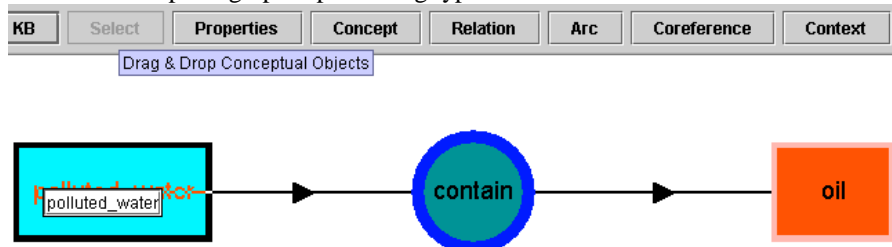


Fig. 4. Drag and drop interface.

Zooming capability. When the zoom is changed new positions of the objects are visualized (all dimensions are recomputed). This feature is very useful for editing large conceptual objects. In Fig 1 the first graph is displayed with the scale 7 and the second one is displayed with the scale 11.

Storing and retrieving of conceptual graphs to/from the application server. Graphs could be either created by editing or acquired automatically by CGExtract [4]. In both cases a natural language representation of a graph is kept as a comment. Also some graphs could be received as a result of a conceptual graph operation, the comment in this case is the identifiers of graphs and the applied operation on them.

3 Reasoning with Conceptual Graphs

As a formal knowledge representation model CGs are provided with reasoning operations, which are sound and complete with respect to deduction in first order logics. All operations are based on canonical formation rules described in [7].

Operations on GCs have been integrated in CGWorld and have been used for generation and acquisition of conceptual graphs using KB in the financial domain. Supported operations are join, projection, specialization, generalization, type contraction and type expansion and they are realized for all kinds of graphs without cycles and actors. A detailed description of each operation including important aspects of its implementation and its user interface is given in [3]. In this paper we will present some extensions of the operations and argue about their usefulness in practical systems. All these operations have been realized in Prolog by now.

3.1 Extensions of the operations

All implemented operations have been used by now mainly from GC users to get the idea of their essence and their usage. Given two CGs identifiers one can perform whichever operations he/she wants. It presumes that the user will browse the KB in order to choose graphs for an operation. So the more experienced and the more acquainted with the definition of operations the user is, the better and faster results he/she will receive. Browsing the KB for suitable graphs is not very useful for non-experienced users as they may receive a lot of results as “Impossible to perform the operation” before they see some result as CG/CGs in both graphical and CGPro representations. To facilitate the user interactions with the system the projection and the join operations have been extended during the last year such that a given CG is projected respectively joined to the whole KB and the result is retrieved to the user. Another benefit from this extension is that the system becomes more flexible in retrieving information from the KB as given a CG (fact) one can retrieve other relevant CGs (facts) from this KB.

Using projection operation (fig. 5) one can extract true facts from the KB either as a part of CGs or whole CGs. Also the projection operation can be used for processing queries to the KB. We have developed a system [2] that process queries in controlled English with negation to the KB using projection operation and we plan to integrate this system in CGWorld in the future. Both simple and extended projection operations have been realized.

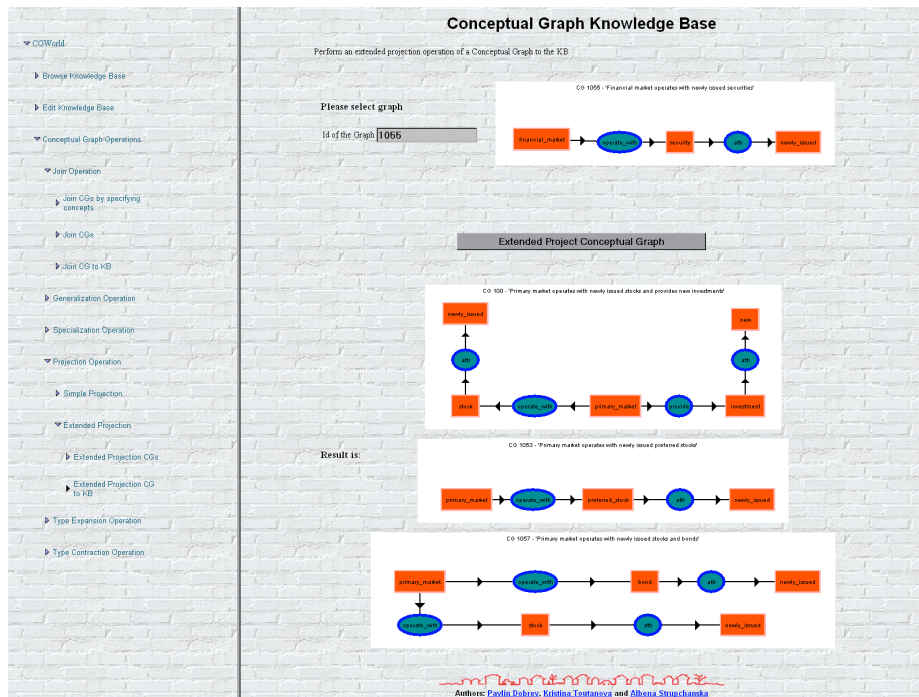


Fig 5. Extended projection operation of a graph to the knowledge base

The join operation is the only operation used by now for generating knowledge (extending assertions in KB) from the existent KB. In most of the cases the new assertions are not necessary true assertions but this is a convenient way for automatically enlarging the KB. Performing join of a CG to the KB is actually maximal join since it joins graphs on maximal common overlap. In this operation we take into account almost all kinds of referents well as correferece links. Processing graphs with correferece links gives some delay in the execution time as we check for compatibility the correferece concepts.

In both operations a user choose an identifier of either existent graph or a new graph created in CG Editor and receives the answer as one or more CGs from the KB.

The generalization operation is also very useful to be implemented in this way since it preserves truth so we can use it to automatically generate true assertions about specific domain. The only disadvantage of this operation is that it does not preserve selection constraints about types and individuals. So in future we will implement generalization of a CG to the KB too.

4 Further work

Currently CGWorld is implemented as a Java Application. It is not straightforward to integrate it in applications written in a language other than Java. Supporting different formats for knowledge representation allows importing and exporting of data to other applications. Using XML [15] as a format for knowledge representation allows data to be exchanged between different applications. The next step that we intend to undertake is extension of GGWorld components to WEB services. WEB services, as the name implies, are services offered via the Web. In a typical Web services

scenario, a business application sends a request to a service at a given URL using the SOAP protocol over HTTP. The service receives the request, processes it, and returns a response. The idea is to offer a set of services for processing conceptual data and exporting it in different formats. This will allow integration with applications written in languages other than Java. For example languages supported by the Microsoft .Net platform such as C#, C, J#, Visual Basic etc.

There are also some legacy parts implemented in Prolog, especially conceptual graph operations. We start to implement some of the operations in MySQL with respect to J2EE architecture and we plan to implement all of them soon.

5 Acknowledgements

We would like to thank all researchers involved in the LARFLAST (LeARning Foreign Language Scientific Terminology²) project and especially Galia Angelova, Svetla Boytcheva, Ani Nenkova and Toma Nikolov that helped us with the programming and knowledge base development.

6 References

1. P. Dobrev and K. Toutanova. CGWorld – Architecture and Features, Proceedings of ICCS'2002, 10th International Conference on Conceptual Structures, Borovets, Bulgaria.
2. S. Boytcheva, A. Strupchanska and G. Angelova. Processing Negation in NL Interfaces to Knowledge Bases, Proceedings of ICCS'2002, 10th International Conference on Conceptual Structures, Borovets, Bulgaria.
3. P. Dobrev, A. Strupchanska and K. Toutanova. CGWorld-2001 - New Features and New Directions, CGTools Workshop at ICCS 2001
4. S. Boytcheva, P. Dobrev and G. Angelova. CGExtract: Towards Extraction of Conceptual Graphs from Controlled English. In: G. W. Mineau (Ed.), Conceptual Structures: Extracting and Representing Semantics, Contributions to ICCS 2001, pp. 89-101.
5. P. Dobrev and K. Toutanova. CGWorld - A Web Based Workbench for Conceptual Graphs Management and Applications. In: G. Stumme (Ed.), Working with Conceptual Structures, Contributions to ICCS 2000, Shaker Verlag, Germany, pp. 243-256.
6. G. Angelova, A. Nenkova, S. Boycheva and T. Nikolov. Conceptual Graph as a Knowledge Representation Core in a Complex Language Learning Environment. In: G. Stumme (Ed.), Working with Conceptual Structures, Contributions to ICCS 2000, Shaker Verlag, Germany, pp. 45-58.

² INCO Copernicus'98 Joint Research Project #977074

7. Conceptual Graph Standard Information Technology (IT) - Conceptual Graphs draft proposed American National Standard (dpANS) NCITS.T2/98-003 (<http://www.bestweb.net/~sowa/cg/cgdpansw.htm>).
8. G. Angelova, K. Toutanova and S. Damianova. Knowledge Base of Conceptual Graphs in DBR-MAT. University of Hamburg, Computer Science Faculty, Project DBR-MAT (funded by the Volkswagen Foundation). Technical Report BG-3-98, July 1998.
9. G. Angelova, S. Damianova, K. Toutanova, K. Bontcheva: Menu-Based Interfaces to Conceptual Graphs: The CGLex Approach. In Proc. ICCS 1997, LNAI 1257, Springer, 1997, pp. 603-606.
10. G. Angelova, K. Bontcheva: DB-MAT: Knowledge Acquisition, Processing and NL Generation Using Conceptual Graphs. In Proc. ICCS 1996, LNAI 1115, Springer 1996, pp. 115-129.
11. S. Pollitt, A. Burrow, P. Eklund. WebKB-GE - A Visual Editor for Canonical Conceptual Graphs. ICCS 1998: pp. 111-118
12. H. Delugach, CharGer – A Conceptual Graph Editor written by Harry Delugach (<http://www.cs.uah.edu/~delugach/CharGer/>).
13. H. Delugach, CharGer: Some Lessons Learned and New Directions. Working with Conceptual Structures, Contributions to ICCS 2000: pp. 306-309.
14. A. Strupchanska, P. Dobrev, S. Boytcheva, T. Nikolov, K. Toutanova, Sample Knowledge Base in Finance, Contribution to CGTools Workshop at ICCS 2001 (<http://www.ksl.stanford.edu/iccs2001/CGTools/>)
15. M. Altheim, XML Conceptual Graphs (XCG) 1.0, Sun Microsystems Technical Report 23 August 2001.
16. Ph. Martin, P. Eklund. Large-scale cooperatively-built heterogeneous KBs. Proceedings of ICCS'2001, 9th International Conference on Conceptual Structures (Springer Verlag, LNAI 2120, pp. 231-244), Stanford University, California, US.