CPE Design Considering Interoperability

Heather D. Pfeiffer, Nemecio R. Chavez, Jr., Joseph J. Pfeiffer, Jr.

New Mexico State University
emails: {hdp, nchavez, pfeiffer}@cs.nmsu.edu

Abstract The Conceptual Programming system (CP) was upgraded to the Conceptual Programming Environment (CPE) back in 2004. At that time, the building and storage of Conceptual Graphs were implemented in a modular fashion in order to allow not only interoperability with CPE, but also to share modules with other applications. Since then a module has been added to the environment for doing foundational reasoning operations, *projection* and *maximal join*. CPE's interoperability is also being enhanced by allowing not only C and C++ applications to access this module, but developing this knowledge system so that a story understanding system, written in Java, can interface with it. Future work is beginning on developing a module that will allow interoperability with CPE and the Common Logic Standard.

1 Introduction

The Conceptual Programming system (CP) was originally developed as a single, standalone application [1,2] that handles temporal, spatial and constraint information [3,4] using a knowledge base of Conceptual Graphs (CGs) [5]. More recently, the New Mexico State University group has begun to investigate alternative data structures and models to allow other applications and systems to communicate with the CP application [6,7,8]. By 2004 the Conceptual Programming Environment (CPE) had been introduced with its new modular, multi-component design to increase the flexibility of the environment and to allow modules to be used outside of the environment by other systems [9]. At that time, the main form of interoperability was by using the CGIF[1] interchange format. CPE included simple wrapper modules to allow other languages, besides C and C++, to use the CGIF module. The knowledge system has now expanded not only into the storage and retrieval of graphs, but also into improving the performance of foundational reasoning operations [10]. When redesigning the algorithms and data structures for the reasoning operations, it was taken into account that CGs can be represented by a *triples* data structure many times seen in databases. Other researchers have also exploited this data structure [11,12] in representing CGs. NMSU is unique in using the data structures directly by the reasoning algorithms, not just in the knowledge representation used for storage

[1] The current archived copy of CGIF from the ICCS2001 workshop is located at: http://www.cs.nmsu.edu/~hdp/CGTools/cgstand/cgstandnmsu.html#Header_44

and retrieval. To show the usefulness of this research, CPE has been used as the knowledge base of objects with query-answer facilities in connection with a new story understanding system.

2 Story Understanding System

This story understanding system is in the process of being built at New Mexico State University using a multi-agent design [13]. The primary goal is for the story understanding system to perform in-depth story understanding by using both world and prototype knowledge through a knowledge base. A short story that has been tested by the new understanding system is the following text:

> Down at the very bottom of the pitcher there was a little water and the thirsty crow tried every way to reach it with her beak. But the pitcher was much too tall. The crow got thirstier and thirstier. At last she thought of a clever plan. One by one she dropped pebbles into the pitcher. Every pebble made the water rise a little higher. When the water reached the brim, the thirsty crow was able to drink with ease [14].

3 Knowledge Base

The knowledge base is divided into two categories: 1) world knowledge and 2) prototype knowledge. The world (or object) knowledge comes from a common sense knowledge system that can perform some form of non-monotonic reasoning, has the ability to change its mind and makes inferences when there is hardly enough information available to make any inference; while the prototype knowledge comes from the new story understanding system. The world knowledge describes objects and their relationship with other objects. This part acts as a basic query-answer facility being able to find elements within the story and how they relate to other objects in the story. The prototype knowledge is a higher-order or abstract knowledge. Unlike the world knowledge, this knowledge is very dependent on the domain context of the text being understood.

Given the text defined above, an English version of a knowledge base would need to stored the following propositions:

Water is a liquid.
Liquid is a thing.
All pitchers have a brim, bottom, and top.
Containers are things that hold things.
Pebbles are things.
Crows are birds.
Birds are animals that have beaks.
Animals are live things.
Live things are things that need air, food, water, and shelter.

3.1 Basic Knowledge About Objects

This part of the knowledge base has open world knowledge and is not specific to any particular domain. This is where CPE and its modules are very applicable. The English knowledge given above can be translated into the following CGIF notation.

```
;The first relations sets up the type hierarchy for the concepts
found in the knowledge base;

(GT [TypeLabel: ‘Entity’] [TypeLabel: ‘Thing’])
(GT [TypeLabel: ‘Thing’] [TypeLabel: ‘Liquid’])
(GT [TypeLabel: ‘Thing’] [TypeLabel: ‘Container’])
(GT [TypeLabel: ‘Thing’] [TypeLabel: ‘Pebble’])
(GT [TypeLabel: ‘Thing’] [TypeLabel: ‘LiveThing’])
(GT [TypeLabel: ‘Liquid’] [TypeLabel: ’Water’])
(GT [TypeLabel: ‘Container’] [TypeLabel: ‘Pitcher’])
(GT [TypeLabel: ‘LiveThing’][TypeLabel: ‘Animal’])
(GT [TypeLabel: ‘Animal’][TypeLabel: ‘Bird’])
(GT [TypeLabel: ‘Bird’][TypeLabel: ‘Crow’])

;The next set of Conceptual Graphs are definitional for giving
structure to some of the concepts in the type hierarchy just
defined;

[Pitcher*p1:@every ;all pitchers;]
(ATTR ?p1 [Bottom])
(ATTR ?p1 [Top])
(ATTR ?p1 [Brim])
(HOLD [Container*c1] [Thing:@{}]
(PART [Bird] [Beak])
(NEED [PROPOSITION:
 (BREATHE [LiveThing*lt1] [Air])
 (EAT ?lt1 [Food])
 (DRINK ?lt1 [Water])
 (LIVEIN ?lt1 [Shelter])])
```

Yet this knowledge does not have to be added directly into the story understanding system. The story understanding system can use CPE completely as a query-answer knowledge system, where graphs and parts of graphs are returned to the story system with a Java wrapper set of routines. All the knowledge in the CPE knowledge base is totally independent of the story understanding system and can be used by any application.

3.2 Prototype Knowledge

The prototype knowledge builds understanding that goes beyond the given text. However, this knowledge is part of the story understanding system and its relevance to the topic of this paper is in how it interoperates with the world knowledge system. The current design for testing different knowledge systems with the story understanding system is to have a fixed application programming interface (API) for all knowledge systems to be used.

Originally it was believed that all basic knowledge had to be stored in all knowledge systems being used by the story understanding system. So some preliminary work was done addressing the translation of data from one knowledge system to another [15]. However, it turns out how the knowledge is stored in the knowledge system is not important, only that the fixed API is necessary for the story understanding system to be able to move between knowledge systems.

4 Future Research

Work has begun on the API between the prototype knowledge and the world knowledge to use the new Common Logic standard, CLIF[2]. This will give a consistent API that will not have to change if another knowledge reasoning system is to be used with the story understanding system. It will also allow CPE to have available a sharable CLIF module for standardized communication with other applications or systems.

Work is also proceeding on interfacing CPE to John Sowa's FMF blackboard application [16]. Modules are being investigated that will take input in either CLCE [17] or ACE [18] format.

References

1. Pfeiffer, H.D., Hartley, R.T.: The Conceptual Programming Environment, CP: Reasoning representation using graph structures and operations. In: Proc. of IEEE Workshop on Visual Languages, Kobe, Japan (1991)
2. Pfeiffer, H.D., Hartley, R.T.: The Conceptual Programming Environment, CP. In Nagle, T.E., Nagle, J.A., Gerholz, L.L., Ekland, P.W., eds.: Conceptual Structures: Current Research and Practice. Ellis Horwood Workshops, Ellis Horwood (1992)
3. Hartley, R.T.: A uniform representation for time and space and their mutual constraints. In Lehmann, F., ed.: Semantics Networks, Oxford, ENGLAND (1992)
4. Pfeiffer, H.D., Hartley, R.T.: Temporal, spatial, and constraint handling in the Conceptual Programming Environment, CP. Journal for Experimental and Theoretical AI **4**(2) (1992) 167–182
5. Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading, MA (1984)
6. Pfeiffer, H.D., Hartley, R.T.: Visual CP representation of knowledge. In Stumme, G., ed.: Working with Conceptual Structures - Contributions to ICCS 2000. (2000) 175–188 Shaker-Verlag.

[2] The current Common Logic standard is located at: http://cl.tamu.edu/

7. Pfeiffer, H.D., Hartley, R.T.: ARCEdit - CG editor. In: CGTools Workshop Proceedings in connection with ICCS 2001, Stanford, CA (2001) [Online Access: July 2001] URL:http://www.cs.nmsu.edu/ hdp/CGTOOLS/proceedings/index.html.
8. Hartley, R.T., Pfeiffer, H.D.: Data models for Conceptual Structures. In: Foundations and Applications of Conceptual Structures, Contributions to ICCS 2002, ICCS2002 (2002)
9. Pfeiffer, H.D.: An exportable CGIF module from the CP environment: A pragmatic approach. In Wolff, K.E., Pfeiffer, H.D., Delugach, H.S., eds.: Conceptual Structures at Work. Volume LNAI of 3127., ICCS2004, Springer (July 2004) 319–332
10. Pfeiffer, H.D.: The Effect of Data Structures Modifications On Algorithms for Reasoning Operations. PhD thesis, New Mexico State University, Las Cruces, NM (2007)
11. Kabbaj, A., Moulin, B.: An algorithmic definition of CG operations based on a bootstrap step. In: Proceedings of ICCS'01. (2001)
12. Skipper, D., Delugach, H.: OpenCG: An open source graph representation. In de Moor, A., Polovina, S., Delugach, H., eds.: First Conceptual Structures Tool Interoperability Workshop, CS-TIW 2006, Aalborg Universitetsforlag (2006) 48–57
13. Chavez, Jr., N.R., Hartley, R.T.: The Role of Object-Oriented Techniques and Multi-Agents in Story Understanding. In: Proceedings of the International Conference on Integration of Knowledge Intensie Multi-Agent Systems, Waltham, Mass, KIMAS 2005 (2005)
14. Aesop: Aesop's fabeles. Doubleday & Company, Garden City Books (1954)
15. Pfeiffer, H.D., Chavez, Jr., N.R., Hartley, R.T.: A generic interface for communication between story understanding systems and knowledge bases. In: Richard Tapia Celebration of Diversity in Computing Conference, Albuquerque, NM (2005)
16. Sowa, J.F.: Architectures for intelligent systems. IBM Systems Journal **41**(3) (2002) 331–349
17. Sowa, J.F.: Common Logic Controlled English. http://www.jfsowa.com/clce/specs.htm (February 2004)
18. Fuchs, N.E., Schwitter, R.: Attempto Controlled English (ACE). In: Proceedings of the First International Workshop on Controlled Language Applications, Belgium, CLAW'96 (March 1996)