Knowledge-Intensive Query Processing

Barbara Starr
SAIC
San Diego
Barbara.H.Starr@cpmx.saic.com

Vinay K. Chaudhri
Artificial Intelligence Center
SRI International
vinay@ai.sri.com

Adam Farquhar
Knowledge Systems Laboratory
Stanford University
axf@hpp.stanford.edu

Richard Waldinger
Artificial Intelligence Center
SRI International
waldinger@ai.sri.com

1 Introduction

Innovative query interfaces to knowledge and database systems must go beyond simply returning the requested information. They must be capable of producing intentional answers when a description improves the understanding of an answer [Mot94], producing conditional answers when no one answer matches the conditions of a query, and using ontological information in processing a query. They should be able to call upon stand-alone reasoning modules that are most suitable for a given query. When answering a question involves reasoning beyond a simple lookup, the system must be able to explain the answer to the user.

We are building a question answering system with these objectives. The heart of the system is a knowledge base (KB) and a collection of reasoning methods. The KB is being constructed by a combination of manual and semiautomatic methods. The reasoning methods include conventional database query processing, frame-based reasoning, and full first-order theorem proving. The performance of this system will be tested on the Crisis Management Benchmark (CMB), which defines a collection of queries of interest to a crisis analyst.

We begin the paper by a description of the CMB. We describe the architecture of our system and then sketch some design ideas for two of its components. We conclude the paper by listing a few of the research challenges faced in building such a system. The work described in the paper is preliminary and is aimed at suggesting directions for future work rather than at describing in-depth technical results.

Proceedings of the 5th KRDB Workshop
Seattle, WA, 31-May-1998

(A. Borgida, V. Chaudhri, M. Staudt, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-10/

2 Crisis Management Benchmark

The Crisis Management Benchmark (CMB) defines a collection of approximately 100 queries, their expected answers, and the knowledge sources that can be used to answer those queries [IACR97]. The CMB has been motivated by the needs of a crisis analyst who is monitoring a part of the world with the objective of predicting a crisis. The CMB has been defined in the context of a scenario involving a conflict among Persian Gulf nations. Many of the queries in the CMB are, however, of general interest. Two features distinguishing the CMB from other benchmarks for measuring database performance [Gra93] are as follows: (1) it measures the knowledge content of the system and not just time to process a query, and (2) it is designed for queries that require processing beyond a simple lookup or join of relations.

2.1 Knowledge Benchmark

For a system to be useful to a crisis analyst, it must have access to sufficiently broad geo-political knowledge. The knowledge benchmark represents a user's view of the system in the sense that it defines the domains in which a user is interested. The CMB does not require the system to necessarily collect all the knowledge in one place. We list here the domain areas in which the knowledge must be encoded. For each domain area we give an example question from the CMB that is representative of that type of knowledge.

World trade and economic information. How much oil does Japan purchase from the Persian Gulf states?
Geography and demographics. Which states border the Persian Gulf?
History of international behavior. Has Japan ever refused to trade with some other country?
Country policies. What is the US policy on illegal immigrants?
Country capabilities. Is Iraq capable of refusing inspection by UN officials?
International organizations. What is the International Monetary Fund and who are its members?

2.2 Processing Benchmark

The processing benchmark represents an implementor's view of the system as it defines processing capabilities that are necessary for answering the CMB questions. The CMB does not mandate any specific reasoning method that must be used for answering a given question. The CMB queries can classified into the following categories.
Retrieval queries. At what fraction of its current sustainable capacity is Iran producing oil?
What-if queries. Assuming constant production by Iran and others, would a 5% increase in production by Saudi Arabia have a positive or negative effect on the economy of Iran? Would a 5% increase by Kuwait have as large an effect?
Analysis queries. Is Iraq capable of refusing inspection by UN officials?

2.3 Operation of the Benchmark

The initial evaluation criteria for the CMB are qualitative, and the answers produced by the system will be judged by a team of experts. Each answer will be tested on the following criteria. Is the answer correct and accurate? Does the answer include any correct, nontrivial analysis which was not obvious? Are the assumptions behind the knowledge appropriate? Does it constitute a realistic model for the question's purpose? Are simplifications appropriate? Is the level of generality or detail appropriate? Is an explanation offered? Is it intelligible to a nonexpert? The benchmark is being refined to include quantitative measures to evaluate these aspects of the answers.

The system will be tested over a period of three years. Each year will consist of a development period of 11 months and a testing period of one month. Each year will begin with a specific scenario and a list of questions that are relevant for it. During the 11-month development period, a system will be developed to satisfy this scenario. During the final month, the system will be tested in three phases. In the first phase, the KB and the scenario will be kept fixed and new questions will be asked. In the second phase, the KB will be fixed and the scenario will be changed. In the third phase, new knowledge will be added to the KB to address the new questions and the changed scenario.

3 System Architecture

The components in our system can be classified into three categories: user interface, knowledge services, and question answering. Many of the components have been developed by different research groups working independently.

An overview of our system architecture is shown in Figure 1. The components are held together by a common application programming interface: Open Knowledge Base Connectivity (OKBC) [CFF+98]. OKBC interfaces for some of the components are already functional, whereas others are being constructed.

Two kinds of user interface are supported by our system. HIKE is a form-based interface that allows a user to construct queries by using pull-down menus. Forms are provided to construct any of the queries in the CMB. START (Syntactic Analysis using Reversible Transformations) is a natural language interface that accepts queries in English [Kat97]. START generates a formal representation of a query and transmits it for evaluation to one of the question answering systems. Automatic generation of a formal representation for an English query is supported only for a subset of queries. Even though the goal of START is to accept arbitrary questions expressed in English, during the course of the current project, the use of START will be restricted to the questions defined in the CMB.

Knowledge services are provided by three components: GKB-Editor, WebKB, and Ontolingua. GKB-Editor is a graphical tool for browsing and editing large knowledge bases [KCP98]. It is primarily used for manual knowledge acquisition. WebKB is a semiautomatic tool for extracting information from the World-wide Web (WWW) [CDF+98]. Given an ontology, and a few examples of the information to be extracted, WebKB can extract objects, relations and probabilistic rules from the text sources on the Internet. The extraction of knowledge is done in a semiautomatic fashion. Ontolingua is the knowledge server and stores all the knowledge in the system [FFR97]. Since it has a focal role in our architecture, we discuss it in detail.

Ontolingua is a tool for creating, evaluating, accessing, using, and maintaining reusable ontologies. It contains a collection of tools and services to support not only the development of ontologies by individuals, but also the process of achieving consensus on common ontologies by distributed groups. These tools include a

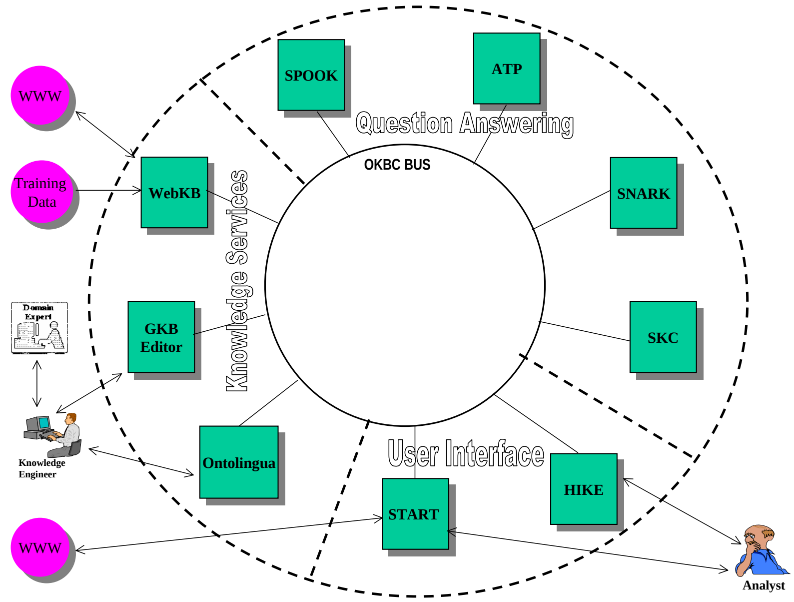

Figure 1: System Architecture: ATP – Abstract Theorem Prover, GKB-Editor – Generic KB Editor, HIKE – form-based GUI, OKBC – Open Knowledge Base Connectivity, Ontolingua – Knowledge Server, SKC – Scalable Knowledge Composition, SNARK – SRI's New Automated Reasoning Kit, SPOOK – System for Probabilistic Object-Oriented Knowledge

semiformal representation language that supports the description of terms in a representation language that is an extension of the Knowledge Interchange Format (KIF) [GF92], browsing and retrieval of ontologies, and facilities for translating ontologies into multiple representation languages.

Question answering is supported by multiple reasoning methods. SNARK is a first order theorem prover [SWL+94]. SPOOK is a reasoner based on Bayes nets [KP97]. When answers are returned by multiple reasoning methods or by using alternative knowledge sources, the answers should be ranked before being presented to the user. Such a ranking will be done by SKC [WG95]. SKC is not yet a part of our system and will be integrated in the later phases of development. Finally, some of the question answering is done by START that uses information retrieval methods to process a query expressed in English on a text-based source. We will illustrate a sample answer produced by one of these components later in the paper.

Let us consider an example of how a question is answered by our system. A user poses her query by using either HIKE or START. In either case, the query is mapped to a formal representation expressed in a language which is an extension of KIF. The query is shipped to one or more of the question answering components. The components may query the knowledge server (Ontolingua) during query evaluation. The resulting answer is returned to HIKE and/or START.

4 Knowledge Services

Two knowledge bases have been playing an active role in our work so far — the HPKB upper-level ontology (HPKB-UL) and the World Fact Book Knowledge Base (WFBKB).

4.1 Upper Ontology

The HPKB-UL is the upper-level ontology of the Cyc KB [LG89] augmented by some links to the Sensus ontology [KL94]. It is being used in DARPA's High Performance Knowledge Bases (HPKB) program. HPKB-UL provides a taxonomy of about 3000 terms and relations for general terms such as tangible-object, action, and transportation. It also defines some relations between them, such as the starting time of an event, the relationship between an object and its parts, and the borders of a geographic region. The X3T2 working group of ANSI has adopted HPKB-UL as the current draft for a standard upper ontology.

There are at least two advantages in using HPKB-UL. First, for formalizing the CMB questions, we need vocabulary. HPKB-UL provides a significant subset of the vocabulary necessary for this purpose. Second, since our knowledge base is being developed by multiple research groups, we need a terminology that will help us in combining KBs developed by different groups. Using a common ontology makes the task of sharing knowledge easier.

For formalizing the CMB queries, we used the following approach: map nouns to concepts, verbs to events and actions, and adjectives to quality attributes, identify individuals, and finally; specify temporal and spatial information. If a necessary term is not found in HPKB-UL, we extend HPKB-UL appropriately. As an example, consider the question:

```
Has post-Shah Iran launched ballistic
missiles in wartime?
```

Our formalization of this query is as follows.

```
(and
   (attack ?act)
   (performed-by ?act Iran)
   (device-used ?act ballistic-missile)
   (later-than (start-of ?act)
               (start-of post-shah-iran)))
```

In this formalization, `performed-by`, `device-used`, `later-than`, and `start-of` are predicates defined in HPKB-UL. For example, `device-used` relates an action to the device that was used in performing it. The predicate `attack` represents a collection of actions in which an agent attacks another agent, and `post-shah-iran` is a constant representing the time interval after Shah of Iran. This formalization does not explicitly represent `wartime` as it assumes that an `attack` is performed only during wartime.

Since all the question answering components subscribe to the same ontology, the above question can be sent for evaluation to any of them. A component may decide to transform this representation into an alternative representation which is more efficient for evaluation by it.

4.2 World Fact Book KB

The World Fact Book KB, being developed by the Knowledge Systems Laboratory at Stanford is a substantial knowledge base covering basic geographic, economic, political, and demographic knowledge about the world's nations. The goal of the project is to provide a useful knowledge resource, to explore ways for structuring large knowledge bases, and to develop a knowledge base large enough to stress existing knowledge representation systems.

The primary source for the World Fact Book knowledge base is the CIA World Fact Book, which collects a broad range of information about the countries and territories of the world.[1] The fact book includes geographical, economic, demographic, and some historical information.

We have been augmenting HPKB-UL and linking it to WFBKB. HPKB-UL stops well above many of the concrete terms that appear in the fact book, such as bauxite mining, the food-and-beverage industry, sandy beaches, ethnic minorities, and spoken languages. We are working to construct a richer ontology that spans the substantial gaps between the HPKB-UL concepts and the terms that are introduced by the World Fact Book. We expect the WFBKB to provide knowledge necessary to answer many of the CMB questions.

5 Question Answering

For the analysis queries defined in the CMB, a simple yes or no answer is not appropriate. Instead, the system must return a descriptive answer and provide some justification for that answer. Let us consider an example of a conditional answer for an analysis question. This answer was produced using SNARK, which is one of the question answering components.

Consider the following question: What will be the likely position of Iraq on allowing inspection by UN officials? For an analysis question like this, usually we want to know more than what the question literally requires. For this question, on one hand, we mean that if Iraq is likely to refuse inspection, we would like to know the reasons. On the other hand, if Iraq is likely to go along with such an inspection, we need confirmation of it; simply failing to find a proof that Iraq can refuse UN inspection does not necessarily imply that it will—there are many true things we cannot prove. In either case, we would like some indication of the reasoning by which the conclusion was reached.

Suppose we have a domain knowledge rule in the KB which states that Iraq is likely to refuse the inspection if it has political support from Russia: if (supports Russia Iraq) then (refuses Iraq UN-Inspection) else (delays Iraq UN-Inspection)

In finding a proof for this question, SNARK performs a case analysis, depending on whether Iraq has support from Russia. If Iraq has support from Russia, it returns an answer (refuses Iraq UN-Inspection). If Iraq does not have support from Russia, it returns an answer (delays Iraq UN-Inspection). If it cannot determine whether Iraq has support from Russia, it returns a *conditional answer* stating that if Iraq has support from Russia, it can refuse UN inspection; otherwise, it will try to delay it. The capability to produce conditional answers is not supported in any of the existing

[1] For more information on the World Fact Book, see `http://www.odci.gov/cia/publications/nsolo/wfb-all.htm`.

query processing systems.

SNARK also prints an explanation for the answer produced by it. The explanation shows all the axioms used and the intermediate inference steps. Each inference step shows the axioms, inference method, and rewrites used in that step, the conclusion derived, and the current answer term. If an English description of an axiom is available, it is shown. The output is produced in the HTML format making it easier for a user to navigate amongst several inference steps.

6 Summary and Conclusions

We have presented initial design ideas of an innovative query interface for a knowledge and database system. The system is capable of producing conditional answers when no one answer matches the conditions of a query, and uses ontological information in processing a query. The heart of the system is a knowledge base (KB) and a collection of reasoning methods. The KB is being constructed by a combination of manual and semiautomatic methods. The reasoning methods include conventional database query processing, frame-based reasoning, and full first-order theorem proving. The performance of this system will be tested on the Crisis Management Benchmark (CMB) that defines a collection of queries which are of interest to a crisis analyst.

We believe that innovative query interfaces such as the one described here represent a major advance in the query processing capabilities of current knowledge and database systems. They open up several challenging research problems that must be addressed. We believe that the following problems are fundamental to enabling the construction of such interfaces.

- How can one take a KB such as WFBKB or HPKB-UL developed for one purpose and use it in a different context?
- How can we reformulate a KB into a form that allows efficient evaluation of a query with a new reasoner?
- When, and in what forms are conditional answers useful?
- How effective can be a generic API such as OKBC in integrating diverse technology components into one system?
- Given a query, on what basis should it be dispatched to a component subsystem and how to combine the results of sub-queries returned by different systems?
- How can we quantitatively measure the performance of answering analysis questions?
- What are the principles and techniques for designing a large knowledge base that would enable knowledge reuse?
- What is a good interface for allowing a user to construct queries by using ontological information?
- What techniques are useful for explaining the answers of a system that uses multiple reasoning methods?

Acknowledgments

At Science Applications International Corporation (SAIC), this work was supported by a DARPA/SSC contract entitled *Knowledge-Base Technology Development and Integration* (Contract Number N66007-97-C-8546). At SRI International, it was supported by a DARPA contract entitled *Ontology Construction Toolkit* (Contract Number N66001-97-C-8550). At Stanford University, it was supported by a DARPA contract entitled *Large-Scale Repositories of Highly Expressive Reusable Knowledge* (Contract Number N66001-97-C-8554).

References

[CDF+98] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proceedings of the National Conference on Artificial Intelligence*, July 1998. To appear.

[CFF+98] Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, and James P. Rice. OKBC: A Foundation for Knowledge Base Interoperability. In *Proceedings of the National Conference on Artificial Intelligence*, July 1998. To appear.

[FFR97] Adam Farquhar, Richard Fikes, and James P. Rice. A Collaborative Tool for Ontology Construction. *International Journal of Human Computer Studies*, 46:707–727, 1997.

[GF92] Michael R. Genesereth and Richard E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, Stanford, CA, 1992.

[Gra93] James N. Gray. *The Benchmark Handbook for Database and Transaction Processing*

Systems. Morgan Kaufmann Publishers, 1993.

[IACR97] IET, Alphatech, Paul Cohen, and Pacific Sierra Research. HPKB year 1 end-to-end challenge problem specification, version 1.1. Technical report, Information Extraction and Transport (IET) Inc., Rosslyn, Virginia, December 1997. See `http://www.iet.com/Projects/HPKB/`.

[Kat97] Boris Katz. From Sentence Processing to Information Access on the World Wide Web. In *AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, Stanford, CA, 1997.

[KCP98] Peter D. Karp, Vinay K. Chaudhri, and Suzanne M. Paley. A Collaborative Environment for Authoring Large Knowledge Bases. *Journal of Intelligent Information Systems*, 1998. To appear.

[KL94] K. Knight and S. Luk. Building a Large-Scale Knowledge Base for Machine Translation. In *Proceedings of the National Conference on Artificial Intelligence*, Seattle, WA, August 1994.

[KP97] D. Koller and A. Pfeffer. Object-Oriented Bayesian Networks. In *Proceedings of the 13th Annual Conference on Uncertainity in AI (UAI)*, Providence, RI, August 1997.

[LG89] Douglas B. Lenat and R.V. Guha. *Building Large Knowledge-based Systems: Representation and Inference in the Cyc Project*. Reading, MA, Addison-Wesley Publishing Co., 1989.

[Mot94] Amihai Motro. Intensional Answers to Database Queries. *IEEE Transactions on Knowledge and Data Engineering*, 6(3):444–454, 1994.

[SWL+94] M. Stickel, R. Waldinger, M. Lowry, T. Pressburger, and I. Underwood. Deductive Composition of Astronomical Software from Subroutine Libraries. In *Proceedings of the Twelfth International Conference on Automated Deduction (CADE-12)*, pages 341–355, June 1994.

[WG95] Gio Widerhold and Michael Genesereth. The Conceptual Basis for Mediation Services. In *Proceedings of the International Conference on Cooperative Information Systems*, Vienna, Austria, May 1995.