

AFRL-IF-RS-TR-2004-314
Final Technical Report
November 2004



KRAKEN-KNOWLEDGE RICH ACQUISITION OF KNOWLEDGE FROM EXPERTS WHO ARE NON- LOGICIANS

Cycorp, Inc.

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. K444

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2004-314 has been reviewed and is approved for publication

APPROVED:

/s/
PETER J. ROCCI
Project Engineer

FOR THE DIRECTOR:

/s/
JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

| | | | | |
|--|---|--|---|--|
| 1. AGENCY USE ONLY (Leave blank) | | 2. REPORT DATE November 2004 | 3. REPORT TYPE AND DATES COVERED FINAL Jun 00 – Feb 04 | |
| 4. TITLE AND SUBTITLE KRAKEN-KNOWLEDGE RICH ACQUISITION OF KNOWLEDGE FROM EXPERTS WHO ARE NON-LOGICIANS | | | 5. FUNDING NUMBERS C - F30602-00-C-0160 PE - 62301E PR - RKFM TA - 00 WU - 01 | |
| 6. AUTHOR(S) Gavin Matthews | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Cycorp, Inc. 3721 Executive Center Drive, Suite 100 Austin TX 78731-1615 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER N/A | |
| 9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency 3701 North Fairfax Drive Arlington VA 22203-1714 | | | 10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2004-314 | |
| 11. SUPPLEMENTARY NOTES AFRL Project Engineer: Peter J. Rocci/IFED/(315) 330-4654 Peter.Rocci@rl.af.mil | | | | |
| 12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. | | | 12b. DISTRIBUTION CODE | |
| 13. ABSTRACT (Maximum 200 Words) Knowledge-Rich Acquisition of Knowledge from Experts Who Are Non-logicians (KRAKEN) was performed under DARPA's Rapid Knowledge Formation (RKF) Program. The KRAKEN system allows Subject Matter Experts (SMEs) to more easily, efficiently and correctly enter their knowledge into an artificial intelligence knowledge-based system. KRAKEN's usefulness has been demonstrated in three challenge problems related to molecular biology, authoring of course-of-action (CA) critiquing rules and terrain analysis. | | | | |
| 14. SUBJECT TERMS Knowledge Acquisition, Knowledge-Based Systems, Natural Language Processing, Rule Authoring | | | 15. NUMBER OF PAGES 104 | |
| | | | 16. PRICE CODE | |
| 17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED | 18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED | 20. LIMITATION OF ABSTRACT UL | |

ABSTRACT

This report contains a description of KRAKEN, a Knowledge Entry system developed as part of the Rapid Knowledge Formation Project, funded by DARPA. In addition to describing the KRAKEN system as it exists today, this report also discusses the development of the system, its performance in three annual evaluations, the lessons learnt that are of general interest to the community of knowledge entry systems developers, and specific insights for Cycorp's future research.

Cycorp has been supported on this contract by sub-contractors at AIAI, ISI, NWU, SAIC and Teknowledge. The principal investigator (PI) at Cycorp, Inc., is Dr. Douglas Lenat. The final project manager for RKF at Cycorp is Mr. Gavin Matthews. Previous project managers were Mr. Stephen Reed, Mr. Robert C. Kahlert and Dr. Michael Witbrock. The DARPA program manager is Mr. Chuck Taylor.

The primary author of this report was Mr. Gavin Matthews, with contributions from Dr. Jon Curtis, Dr. Kerry Hines, Mr. Robert C. Kahlert, Dr. Pierluigi Miraglia, Dr. Amanda Vizedom, and Dr. Michael Witbrock.

Section 1 of this report sets the project's work in the context of the original proposal, gives an overview of the evolving architecture and collaborators involvement, and describes some synergies with other Cycorp projects. Section 2 covers the goals, developments and insights of each year in more detail, concentrating on the new interface modalities produced. Section 3 describes the project evaluations, with a strong emphasis on the Year Three results. Section 4 discusses how the RKF Project has affected Cycorp, the general lessons learned, and the related ongoing and future work. Section 5 gives details of publications arising from this project.

Appendix A gives more detail on the Year Three evaluation, and explains how close the SME-entered knowledge was to the final system. Appendix B explains recent ongoing developments in parsing technology. Appendix C is an informal SME assessment of working with KRAKEN. Appendix D is an outline of some aspects of military theory used in RKF Year Three. Appendix E is a glossary of terms and abbreviations.

TABLE OF CONTENTS

| | | |
|----------|--|-----------|
| 1 | OVERVIEW OF THE KRAKEN SYSTEM | 1 |
| 1.1 | GOALS FOR THE KRAKEN SYSTEM..... | 1 |
| 1.1.1 | <i>Rich Tools</i> | 2 |
| 1.1.2 | <i>Multidimensional Context Tools</i> | 2 |
| 1.1.3 | <i>Deeply Understand Text</i> | 3 |
| 1.1.4 | <i>Clarification and Discourse</i> | 3 |
| 1.1.5 | <i>Planning and Problem Solving</i> | 4 |
| 1.1.6 | <i>Explicit Reasoning about KE Methodology Itself</i> | 4 |
| 1.1.7 | <i>KBs to fix KBs</i> | 4 |
| 1.1.8 | <i>Pegs</i> | 5 |
| 1.1.9 | <i>Mix and Match</i> | 5 |
| 1.1.10 | <i>Metaphors and Analogies</i> | 5 |
| 1.1.11 | <i>Active Collaboration Aid and Control</i> | 6 |
| 1.1.12 | <i>Automated Tracking of Metrics</i> | 6 |
| 1.1.13 | <i>Ultimate Impact</i> | 7 |
| 1.1.14 | <i>What was left out: OEs, KEs and SMEs</i> | 7 |
| 1.2 | THE CYC SYSTEM | 8 |
| 1.3 | THE KRAKEN ARCHITECTURE | 12 |
| 1.3.1 | <i>RKF Year One</i> | 12 |
| 1.3.2 | <i>RKF Year Two</i> | 12 |
| 1.3.3 | <i>RKF Year Three</i> | 13 |
| 1.4 | RELATED WORK AT CYCORP | 14 |
| 1.5 | EVALUATIONS OF KRAKEN..... | 17 |
| 1.6 | SUMMARY OF COLLABORATOR INVOLVEMENT..... | 17 |
| 2 | COMPONENTS OF KRAKEN TECHNOLOGY | 18 |
| 2.1 | YEAR ONE..... | 18 |
| 2.1.1 | <i>Context</i> | 18 |
| 2.1.2 | <i>Tool Set</i> | 19 |
| 2.1.3 | <i>Lessons Learned</i> | 22 |
| 2.2 | YEAR TWO..... | 23 |
| 2.2.1 | <i>Context</i> | 23 |
| 2.2.2 | <i>Lessons Learned</i> | 27 |
| 2.3 | YEAR THREE..... | 29 |
| 2.3.1 | <i>Context</i> | 29 |
| 2.3.2 | <i>Tool Set</i> | 30 |
| 2.3.3 | <i>Lessons Learned</i> | 44 |
| 3 | EVALUATIONS | 46 |
| 3.1 | YEAR 1: MOLECULAR BIOLOGY..... | 46 |
| 3.2 | YEAR 2: SKETCHING-ASSISTED AUTHORIZING OF COA-CRITIQUING RULES | 46 |
| 3.3 | YEAR 3: GIS-BACKED ADVANCED AUTHORIZING OF TERRAIN ANALYSIS PROCESS..... | 47 |
| 3.3.1 | <i>Evaluation Methodology</i> | 47 |
| 4 | DISCUSSION | 55 |
| 4.1 | HOW THE RKF PROJECT HAS AFFECTED CYCORP | 55 |
| 4.1.1 | <i>Precision Suggestion</i> | 55 |
| 4.1.2 | <i>WFF-Repair</i> | 55 |
| 4.1.3 | <i>Concept Refinement Interview</i> | 55 |
| 4.1.4 | <i>Lexical Matching</i> | 55 |
| 4.1.5 | <i>KB Graphing</i> | 56 |

| | | |
|---|---|-----------|
| 4.1.6 | <i>Dictionary Assistant</i> | 56 |
| 4.2 | LESSONS LEARNT FROM THE RKF PROJECT | 57 |
| 4.2.1 | <i>Listen to the SMEs (But Not Too Much)</i> | 57 |
| 4.2.2 | <i>Avoid Subtle Distinctions</i> | 58 |
| 4.2.3 | <i>Avoid Unnecessary Delay</i> | 58 |
| 4.2.4 | <i>Track What the User Says</i> | 59 |
| 4.2.5 | <i>Focus on Interface Sweet-spots</i> | 59 |
| 4.2.6 | <i>Overall Insights</i> | 59 |
| 4.3 | ONGOING AND FUTURE WORK | 61 |
| 4.3.1 | <i>Natural-Language Processing</i> | 61 |
| 4.3.2 | <i>Interfaces</i> | 61 |
| 4.3.3 | <i>Open Issues</i> | 61 |
| 4.3.4 | <i>Potential Applications</i> | 62 |
| 5 | PUBLICATIONS | 63 |
| APPENDIX A: ONTOLOGY REQUIRED FOR EVALUATION RESULTS | | 65 |
| | HAND-MERGING | 65 |
| | WELL-FORMEDNESS (WFF) PROBLEMS: TYPE/INSTANCE CONFUSION | 65 |
| | VARIABLE MAPPINGS | 69 |
| | ADDITIONAL NODE..... | 70 |
| | VARIABLE MAPPING MISSING FROM ARC | 70 |
| | NO ASSOCIATED QUERY | 70 |
| | WORKING AROUND ABSENCE OF PLANNED VALUE TABLES FEATURE | 71 |
| | QUERIES NOT WORKING AS WRITTEN..... | 71 |
| | EXECUTION FORWARD RULE WRITING..... | 73 |
| | GIS WORKAROUND..... | 74 |
| APPENDIX B: DETAILS OF KB-BASED PARSE REPRESENTATION | | 75 |
| | DEFERRED DISAMBIGUATION AND UNLIMITED UNDO..... | 75 |
| | DIALOGUE MODEL | 76 |
| | USE OF PARSING | 77 |
| APPENDIX C: SME QUESTIONNAIRE | | 78 |
| APPENDIX D: TERRAIN-BASED RISK ASSESSMENT KNOWLEDGE BASE | | 85 |
| | DEFENSE..... | 87 |
| | OFFENSE..... | 87 |
| APPENDIX E: GLOSSARY OF TERMS AND ABBREVIATIONS | | 93 |

List of Figures

| | |
|--|----|
| Figure 1. Dynamically generated glossary | 3 |
| Figure 2. Overview of Cyc's ontology and microtheory hierarchy | 8 |
| Figure 3. KRAKEN architecture for RKF Year One | 12 |
| Figure 4. KRAKEN architecture for RKF Year Two | 13 |
| Figure 5. KRAKEN architecture for RKF Year Three | 13 |
| Figure 6. Kraken-based Factivore application being used to represent a new Terrorist as part of a different project to form a Terrorism Knowledge Base | 14 |
| Figure 7. Cyc uses Kraken-developed NL generation in justifying answer to a chemistry question in a commercially funded project | 15 |
| Figure 8. Three screenshots from the CBRNE threat system | 16 |
| Figure 9. Original architecture design for KRAKEN | 19 |
| Figure 10. D. Lenat's original UI proposal for KRAKEN from the New Orleans PI Meeting | 21 |
| Figure 11. ISI's WhyNot tool suggests a plausible fact that could be added to Make a failing query succeed | 22 |
| Figure 12. KRAKEN's interface at the end of RKF Year Two | 24 |
| Figure 13. Northwestern University's nuSketch BattleSpace | 25 |
| Figure 14. Concept Refinement Interview | 26 |
| Figure 15. Guided Knowledge Entry (GKE) | 27 |
| Figure 16. Three rules created by SMEs in the Year Two evaluation | 28 |
| Figure 17. ESRI ArcView, the GIS of choice | 30 |
| Figure 18. Cyc Factivore Template | 31 |
| Figure 19. Simplified example of results merging | 37 |
| Figure 20. An example of a prototype value table | 38 |
| Figure 21. Extract from a flowchart prepared by an RKF Year Three SME | 40 |
| Figure 22. The Analysis Diagram Tool | 41 |
| Figure 23. Detailed view of terrain analysis process with respect to observation As viewed in the Analysis Diagram Tool | 42 |
| Figure 24. Variable mapping dialog from the Query Library | 43 |
| Figure 25. An example of a rule authored by a SME during the RKF Year Two Evaluation | 47 |
| Figure 26. Query Library showing queries that ask Cyc about a scenario | 52 |
| Figure 27. Query Library shows a selection of queries at the left | 53 |
| Figure 28. Query Library gives full explanations for its results | 53 |
| Figure 29. Map sheet (from GIS) showing Stampede Mountain | 54 |
| Figure 30. Close up of Stampede Mountain map sheet | 54 |
| Figure 31. WFF Repair suggests a new sentence that, if asserted, would Allow the original assertion | 55 |
| Figure 32. The Dictionary Assistant | 56 |
| Figure 33. Dynamically generated graph showing a fragment of the Cyc ontology | 56 |
| Figure 34. Concept Creator | 58 |
| Figure 35. Precision Suggestor | 59 |
| Figure 36. Mock-up of the Interlocutor interface to KRAKEN | 76 |
| Figure 37. Initialization data | 87 |

LIST OF TABLES

| | |
|---|----|
| Table 1. Area of interest (AI determination) | 87 |
| Table 2. Defense tactical terrain evaluation criteria | 88 |
| Table 3. Offense tactical terrain evaluation criteria | 89 |
| Table 4. Battlefield tactical terrain use | 90 |
| Table 5. Functions, interactions, and handoffs | 91 |

1 Overview of the KRAKEN System

In Norse mythology, the Kraken is a giant sea-monster, immortalized by Tennyson in the lines:

*Below the thunders of the upper deep;
Far far beneath in the abysmal sea,
His ancient, dreamless, uninvaded sleep
The Kraken sleepeth: faintest sunlights flee
About his shadowy sides; above him swell
Huge sponges of millennial growth and height;
And far away into the sickly light,
From many a wondrous grot and secret cell
Unnumber'd and enormous polypi
Winnow with giant arms the slumbering green.
There hath he lain for ages, and will lie
Battening upon huge seaworms in his sleep,
Until the latter fire shall heat the deep;
Then once by man and angels to be seen,
In roaring he shall rise and on the surface die.*

KRAKEN¹, on the other hand, is the Knowledge-Rich Acquisition of Knowledge from Experts who are Non-logicians – Cycorp’s Cyc-based contribution to DARPA’s Rapid Knowledge Formation project. The KRAKEN system was intended to break through some of the existing barriers that prevented Subject Matter Experts (SMEs) from entering their knowledge into an artificial intelligence system efficiently and correctly.

1.1 Goals for the KRAKEN System

In the original technical proposal for KRAKEN,² Doug Lenat identified thirteen key attributes for an ideal version of such a system, which functioned as the initial set of goals for the KRAKEN system:

- **Rich Tools:** The knowledge entry tools must themselves be knowledge-based;
- **Multidimensional Context Tools:** The system must model the user’s context in order to display appropriate information, offer suitable choices of action, and interpret the user’s actions;
- **Deeply Understand Text:** The system must be able to gain a deep understanding of text;
- **Clarification and Discourse:** The system must be capable of interactive clarification dialog;
- **Planning and Problem Solving:** The system must be able not only to plan its own dialog, but also to derive action plans for solving the user’s problems;
- **Explicit Reasoning about KE Methodology itself:** The system must be able to reason about the process of knowledge acquisition as such;
- **KBs to fix KBs:** The knowledge base must not only be structured, but the structural principles must be explicitly represented;
- **Pegs³:** The NL understanding must be able to cope with anaphora and cataphora;

¹ In addition to the definitions provided in the body of this report, a glossary of acronyms and selected terms is provided for convenience as Appendix E.

² Douglas Lenat et al, “KRAKEN: Knowledge-Rich Acquisition of Knowledge from Experts who are Non-Logicians”, Volume 1: Technical Proposal, 2000.

- **Mix and Match:** The system must be able to work within a heterogeneous array of different systems;
- **Metaphors and Analogies:** In order to understand normal human communication, the system must be able to interpret metaphors, analogies, and similes;
- **Active Collaboration Aid and Control:** The system must support a collaborative mode of working whereby SMEs can share work, and correct or review each other's work product;
- **Automated Tracking of Metrics:** The system must be capable of automatic bookkeeping to measure which KE techniques are performing best; and
- **Ultimate Impact:** In addition to the benefit to the DoD, the system must provide a significant potential benefit to industry and institutions.

This section presents an end-of-project assessment of the project's achievements with respect to each of these criteria. Many of KRAKEN's components and capabilities are introduced here, with references to further discussions in the sections that follow.

1.1.1 Rich Tools

Cycorp's general philosophy for development is: Design big, implement small. In terms of Knowledge Entry tools, this means that the first prototype of a tool may be limited and somewhat hard-wired in its behavior, but the infrastructure is designed to permit arbitrary extension of its intelligence in the future. A good example of this is the Factivore, which started with hand-written templates (albeit represented in the knowledge base), progressed to template authoring tools that suggested questions, until finally reaching the current state, in which it presents questions that have been induced entirely autonomously. The Factivore is discussed in detail in Section 2.3.2 below.

Specifically, the components developed for KRAKEN have been based on a judicious blend of custom code and use of Cyc's large knowledge base and inference engine. To the greatest extent possible, all aspects of system behavior depend on the results of inference queries over the knowledge base. In this respect, the KRAKEN tools are indeed knowledge-rich.

The same is true for Cyc's natural language capabilities, which are based on a large set of lexical mappings, grammatical rules and templates for parsing and generating natural language, all stored in the knowledge base. Further, the knowledge base contains heuristics for resolving problems of under-specification and vagueness found in natural language text.

1.1.2 Multidimensional Context Tools

KRAKEN exhibits context-sensitive behavior in a number of ways:

- Every User Interaction Agenda (UIA) session has a specific topic. Topics allow conversational focus, adjust the salience of specific NL interpretations, and make different sets of tools available; this also provides a sandbox for the user's contributions and protects users of the system from each other.

³ The term "discourse peg" was used by Susan Luperfoy in her 1992 paper "The Representation of Multimodal User Interface Dialogues Using Discourse Pegs".

- A Year Two innovation was the introduction of topic-specific glossaries (see Figure 1). These glossaries are dynamic — their content reflects the current state of the knowledge base, and so they are updated as the user adds knowledge;
- The knowledge the user enters and the user’s other actions all contribute to the working context, and are available for use in inference and natural language generation, without however impacting the other users, should the user-authored knowledge be semantically incorrect or the lexical information misleading; and
- In Year Three, the major improvement to rule authoring was to allow the component queries to be interpreted in the context of their place in the analysis process, which allowed them to be written as if their predecessor rules had already fired; see also section 2.3.2.1. This greatly simplified the rule-authoring task for Year Three SMEs, and made the resulting knowledge representations better.

1.1.3 Deeply Understand Text

Substantial progress has been made on deep Natural Language Understanding (NLU) over the course of the project, but the state-of-the-art still falls short of being able to attach deep semantics to the vast majority of normal human textual communications. Within the project, Cycorp has developed some promising directions for future research, moving more of the parsing steps into the knowledge base where they can be the focus of arbitrary reasoning as well as re-evaluation of said things in light of additional information (see discussion of the Interlocutor, in Section 2.3.2 and Appendix B for more detail).



Figure 1: Dynamically generated glossary.

The current system has broad lexical coverage, does well at interpreting noun and verb phrases, and does fairly well with simple sentences. Several different types of parsing technology are able to work together. When the system cannot parse a sentence, it is able to let the user know exactly where it became stuck, or what the potential interpretation choices at this point are. This allows the user to intervene, using interactive clarification dialog, or to choose to provide the missing information either by creating new concepts or adding lexical information to existing ones.

1.1.4 Clarification and Discourse

The UIA component of KRAKEN has been extremely effective in developing the concept of mixed-initiative dialog, wherein the user is engaged in a conversation with the system, and either can offer guidance on the direction it should take.

The KRAKEN user has a range of initial input choices, from free-form text, through concept or rule creation, to adaptation of example sentences. Thereafter, every user utterance is parsed for its possible meanings and, when there is ambiguity, clarified by back-and-forth dialogue. In addition, the system is capable of devising its own questions for the user, to flesh out focal concepts; these questions are derived (both deductively and inductively) from Cyc's existing knowledge.

As a simplified example of the way clarification dialog works, if the user asked "Do Americans eat dogs?" then the system might respond with "Do you mean dogs the mammal or hot dogs the foodstuff?" If the user indicates the former, a further clarification might be "Did you mean do *all* Americans eat dogs, or just some of them?".⁴ See Section 4.2.6 and the IJCAI paper for more details.

1.1.5 Planning and Problem Solving

From the initial concept, a key principle of KRAKEN has been to keep the user informed of the tasks in progress (the 'to-do list') and the options available. This is evidenced in the UIA's Agenda, which optionally appears as a panel to the side of the main frame. Every current task is listed with its status, indicating whether the system is ready for user input (see Figures 7 and 12). Via the Agenda, the user can switch between multiple threads of knowledge entry.

The Analysis Diagram Tool, discussed in detail in section 2.3.2.1 below, permits the analyst to set the agenda, sketching out some parts of the process without full semantics, and then filling in the deeper knowledge later. In this mode, the system does not interrupt the user to pin down every detail immediately, but merely indicates the status. Developments like the Interlocutor (see Appendix B) will allow the system to be even more flexible in this respect.

1.1.6 Explicit Reasoning about KE Methodology Itself

As the sophistication of the underlying knowledge representation increases, so too does the ability of Cyc to reason about its own behavior. There are many ways in which Cyc uses such meta-reasoning. For example, rules in the knowledge base are used to determine which Cyc concepts are relevant to present to the user in a given context. Reformulation rules (also in the knowledge base) allow Cyc to reason about how to translate parsed input representations into effective and efficient CycL. The Salient Descriptor, the Factivore and other KRAKEN components that query the user for absent information relevant to the type of concepts being described, based on explicit models of what knowledge is typically known about which concepts.

The Analysis Diagram Tool captures knowledge about how related pieces of knowledge should be used together, and what knowledge is most relevant when. This in itself constitutes a significant kind of meta-knowledge. Moreover, it provides an additional foundation for meta-reasoning concerning the presence or absence of significant knowledge (see Section 4.3.4, on potential applications, for more discussion of this possibility).

1.1.7 KBs to fix KBs

An important aspect of structuring knowledge is the ability to use the existing structure to communicate more effectively with the user. The Cyc ontology was structured by humans, but is not necessarily intuitive for all SMEs, because the distinctions drawn by ontologists are not

⁴ This is a synthetic example. In actual use, the system output includes more involved screen layout, more verbose phrasing and clarifying tool tips.

reflected in normal usage or conceptualization. A key attribute of KRAKEN components, therefore, is that they guide the user to enter knowledge in a way that conforms to the existing KB structure, while remaining comprehensible to the user. For example, the Concept Refinement Interview, harnessing the Salient Descriptor and the Precision Suggestor, makes it easy for users to classify entities and state facts about them at just the right level of generality.

Cycorp was also able to use the METT-T and OCOKA principles⁵ for COA evaluation to great effect in RKF Year Two, and continued to do so through Year Three. The METT-T and OCOKA mnemonics are examples of the incorporation of domain-specific structures that make sense to the SMEs on top of the general domain-independent structure of the knowledge base.

1.1.8 Pegs

KRAKEN has the ability to create terms that it could not parse with reasonable defaults, provided it understood enough of the other elements of the utterance. The current implementation falls short in often pressing for the SME's early agreeing to such auto-creation, though by RKF Year Three this had become tool specific – unlike the UIA, the Factivore will quietly create the PEG and move on. For the more principled approach see the discussion of the Interlocutor in Appendix B.

Cyc has a solid ability to identify and parse discourse references. First and second person references can be resolved in the context of the conversation between the user and Cyc. Anaphoric references within a sentence can be resolved in many cases. Cyc cannot, as yet, resolve anaphora between sentences, but the advanced parse representation developed in RKF Year Three holds great promise for the deduction of links from references to their referents, closing the gap between current capabilities and understanding passages of text (see Appendix B for more detail). The ability to resolve inter-sentential anaphora will also permit SMEs to use shorter and simpler input sentences, thus boosting Cyc's ability to parse them reliably. Ongoing work in other projects is likely to take advantage of this.

Cyc also has the ability to insert anaphora into its generated output. This is especially effective in providing paraphrases for complex assertions such as rules.

1.1.9 Mix and Match

In the course of the RKF project, KRAKEN has brought together a variety of systems from different collaborators that worked together to enable SMEs to convey their knowledge. These collaborations are described in more detail below (see Sections 1.3, 1.6, and 2). Most recently, integration with ERSI's ArcMap GIS provided support for the representation of terrain evaluation knowledge.

1.1.10 Metaphors and Analogies

The comprehension of metaphor is among the most difficult of natural language tasks, and no attempt was made to address it in the RKF project in a principled fashion. Notwithstanding, KRAKEN does have the ability to develop analogies explicitly as a powerful knowledge entry

⁵ Military acronyms for evaluation of courses of action (COAs) and terrain. METT-T stands for Mission, Enemy, Terrain and weather, Troops and support, and Time. OCOKA concentrates on the Terrain aspects, and stands for Observation and fields of fire, Cover and concealment, Obstacles, Key terrain, Avenues of Approach.

technique. This can either be done by asserting a simple sentence like “X is like Y” or by choosing a menu item. In either case, KRAKEN will reason about the possible nature of the analogy, and will ask the user questions to explore the similarities and differences.

1.1.11 Active Collaboration Aid and Control

At the start of the project, although Cyc could test SME-originated knowledge for semantic well-formedness, contradiction and redundancy, it was expedient to have trained ontologists review and correctly place within the microtheory hierarchy all entered knowledge. This manual process limited real-time collaboration between SMEs. As the project progressed, systems to automatically validate and lift SME knowledge to the appropriate microtheory positions were developed to handle the many cases where actual intervention by ontologists was not needed. In the final system, SME-entered knowledge is in the majority of cases suitable for direct addition to the Cyc knowledge base. See Section 3.3 and Appendix A for further discussion.

Throughout RKF Year Three, the KRAKEN system was being used by up to four military SMEs (each from a different service background, and each working in a different city) to develop terrain-evaluation knowledge. The tools employed not only permitted SMEs to add knowledge, but also allowed them to review and correct knowledge entered previously. This collaborative ability was a significant contributor to the quality of the resulting knowledge.

Infrastructure for generating notifications whenever parts of knowledge changed had been developed in RKF Year Two and has just now become reflected into the user interfaces of tools such as the Factivore; however, the RKF Year Three mission and evaluation provided no context for developing or fielding explicit inter-SME notification tools.

One thing not envisioned during RKF was extending SME collaboration to a knowledge acquisition workflow system, where ontologists, knowledge engineers and SMEs can develop a knowledge-vertical together. While such a task was actually undertaken with the Expert Knowledge Base Challenge Problem (see Section 2.1.1 below, IET’s IJCAI 2003 report *Evaluating SME-Elicited Knowledge*), there was little tool support for the KE and OE aspects of this effort; see also Section 1.1.14 below.

1.1.12 Automated Tracking of Metrics

Cyc records a great deal of information about knowledge entry activities. For example, the author, date and time of all assertions are recorded automatically, along with the project for which the work was done, if applicable. When metrics are analyzed for evaluations, they are generally based on Cyc’s default logs, and do not require additional metering.

During RKF Year Two, additional infrastructure was developed to track the authorship of facts back to specific tools more precisely. However, given the collaborative architecture of the KRAKEN system, such individual attributions can be misleading.

Any problems encountered by a SME in using the UIA result in detailed reports that are distributed automatically for use in debugging. Some of these reports are entirely automatic responses to SME actions (such as rejection of a parse interpretation). Others may require a SME to choose a “Report Problem” option, type in a comment, and hit “Send,” but the additional information required for analysis and/or debugging is gathered automatically by the system and appended to the problem report.

1.1.13 Ultimate Impact

KRAKEN is still in the operational prototype stage, but has yielded a wide range of technology components that can be employed commercially. The support for natural language processing is gradually migrating into OpenCyc, and will shortly appear in ResearchCyc, for the benefit of academic researchers. Many of the components and capabilities developed for the RKF project will be relied upon in Cycorp's ongoing commercial contracts. Kraken-developed natural language generation technology, for example, is used extensively in the Cyc-based computer security product currently being commercialized by an outside company.

As is reflected in the Year Three SME comments (see appendix), there is significant potential for the use of KRAKEN technology for education and training. In addition, the fact that Cyc's underlying knowledge representation is natural language independent, coupled with the compositional generation system, suggests that it would work well to support multi-lingual collaboration. See the discussion of potential applications, in Section 4.3.4 below, for more on this subject.

1.1.14 What was left out: OEs, KEs and SMEs

Knowledge acquisition tasks can be distinguished, among other approaches, by the level of competency in logic that they require. Traditionally the key distinction has been between ontologists (a.k.a. OEs), who are experts in logic, and knowledge engineers (a.k.a. KEs), who have received solid logic training. The community suspected, however, that many of the tasks that knowledge engineers were performing could be done by Subject Matter Experts, i.e. did not *inherently* require logic training, provided they were adequately equipped with the appropriate tools and the logical representation was properly hidden from the Subject Matter Experts.

As a program, RKF focused on identifying that sweet-spot in the knowledge representation curve and developing the infrastructure that empowers SMEs to take on KE level tasks. However, its task was not – and could not have been – to make SMEs capable of performing OE level tasks. Neither did it strive to develop OE level tools, though it laid the foundation for and helped identify some of the tools that could be developed in the future.

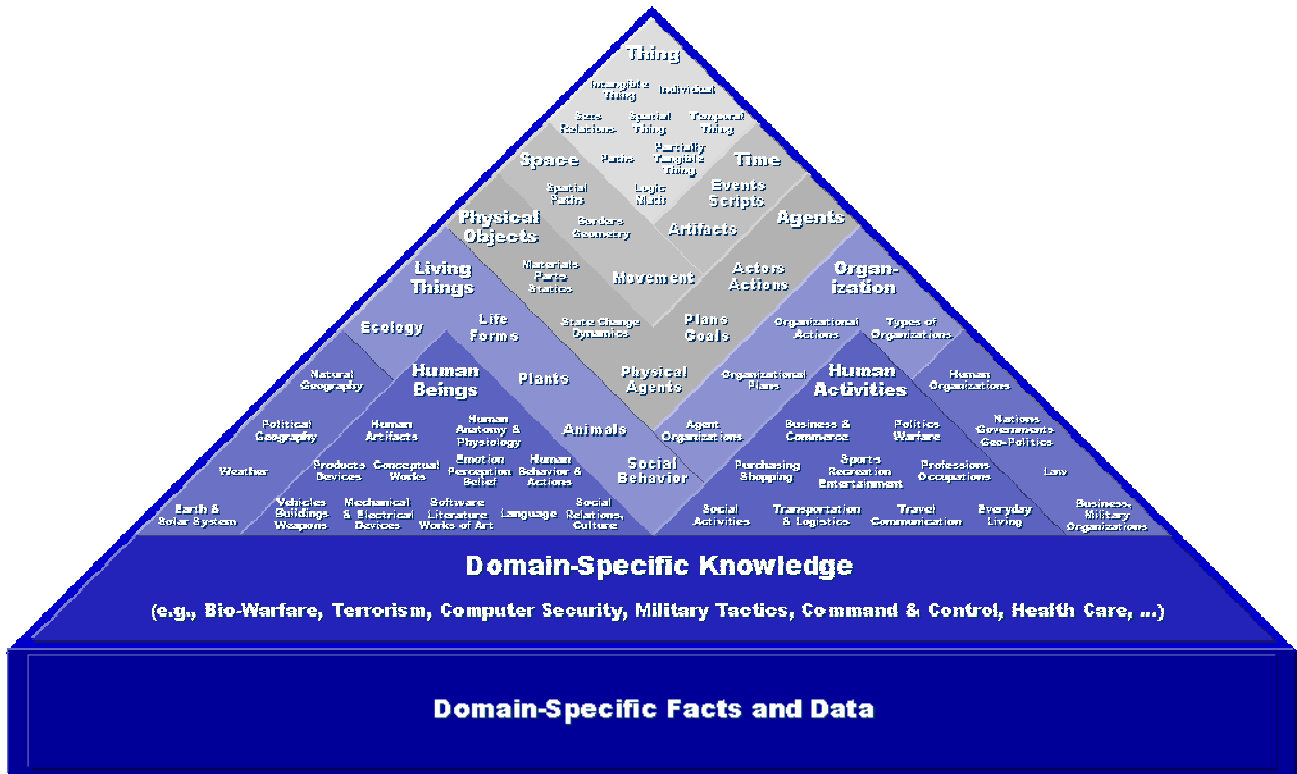


Figure 2: Overview of Cyc's ontology and microtheory hierarchy.

1.2 The Cyc System

At the heart of the KRAKEN system is the Cyc knowledge base and inference engine.

The Cyc knowledge base contains an ontology of over 190,000 concepts (nearly 4,000 created for RKF), including relations (functions, predicates, etc.), collections and individuals. These are related by more than 2,200,000 assertions (over 133,000 of which were made for the RKF project) organized into a hierarchy of microtheories. These assertions are expressed in CycL, a declarative language based on first-order predicate calculus with some higher-order features. CycL strikes a balance between expressivity and efficiency that permits it to be used for representing real-world knowledge and drawing conclusions from that knowledge.

Each concept in Cyc is represented in CycL as either a collection or an individual. Each concept, so represented, is given a definition, which consists of one or more “isa” assertions ascribing membership to some collection. In the case of collections, the definition also includes one or more “genls” assertions that identify those collections for which the defined collection is a specialization. For example:

```
(isa Spur-TopographicalFeature MilitaryMinorTerrainType)
```

means that “Spur” is classified as a minor terrain-type

```
(genls Spur-TopographicalFeature TerrainHighGround)
```

means that every spur is high ground

Ground Atomic Formulae (or GAFs) can be used to express relations that hold among specific concepts. So, for example, where `Spur001` and `BlueUnit075` are defined thus:

```
(isa Spur001 Spur-TopographicalFeature)
(isa BlueUnit075 GroundForce)
```

the predicate `on-Physical` can be used to form a GAF that states that this particular blue unit is on this particular spur:

```
(on-Physical BlueUnit075 Spur001)
```

General knowledge about types can be captured in rules, which can then be used to draw conclusions about instances of those types. For example,

```
(implies
  (and
    (isa ?TERRAIN Spur-TopographicalFeature)
    (on-Physical ?FORCE ?TERRAIN)
    (isa ?FORCE GroundForce))
  (isaForAgent ?TERRAIN ConcealmentLimitingTerrain ?FORCE))
```

is a rule that captures the general knowledge that spurs are concealment limiting terrain for ground forces on them. This rule can be used to conclude:

```
(isaForAgent Spur001 ConcealmentLimitingTerrain BlueUnit075)
```

i.e., that spur 001 is concealment limiting terrain for Blue 075, which is on it.

To capture such rules at a level that will allow for special HL (heuristic level) reasoning support (see the discussion of the Cyc inference engine, below), CycL allows for the arbitrary introduction of rule "macro" vocabulary. For example, the predicate `tacticalTerrainTypeFromPositionForUnitType` can be used to form a GAF that expresses the above rule:

```
(tacticalTerrainTypeFromPositionForUnitType
  Spur-TopographicalFeature ConcealmentLimitingTerrain
  on-Physical GroundForce)
```

The Cyc inference engine uses general deductive reasoning in combination with over 500 specialized modules that supply sweet-spot optimizations or interface with external knowledge sources.

Cyc's knowledge base is also a repository for the information used to support its natural language processing. Cyc has over 28,000 root words represented, with semantic translations for >5,000 of those single-word forms. There are more than 80,000 mappings from proper names to entities in Cyc, and there is syntactic and semantic info for 38,000+ multi-word combinations like "machine gun".

Cyc has a range of parsing technologies available, such as the Template Parser, which is good at recognizing the overall structure of sentences, and the Phrase Structure Parser, which is good at parsing noun and verb phrases. Together, they provide a versatile and powerful parsing ability.

In addition to parsing, Cyc also has NL generation technology so that the contents of the knowledge base can be presented in human-readable form.

Like common sense, humans find it easy to communicate in NL. As a result, human expectations for NLP systems inevitably outstrip system performance. However, and due in large part to the RKF effort, Cyc's parsing and generation technologies represent partial solutions to many of the well-known difficulties surrounding Natural Language Processing.

For example, human conversation is context-rich and context-sensitive, but parsing and generation technologies are often relatively context free. Context is implemented in the knowledge base with Microtheories. This enables NLP technologies to be applied intelligently, using methods as applicable given facts about the given conversational context. For example, certain parts of the lexicon (e.g., slang speech and rude speech) are unavailable in conversational contexts that revolve around COA authoring or battle space analysis.

Secondly, the logical features of concepts differ, sometimes dramatically, from the grammatical features of the language in which they are described. For example, in English, both "cat" and "catalyst" are nouns, but the concepts to which they refer are represented in CycL quite differently, the former being represented with a collection (#\$Cat), the latter with a relation

```
(catalyst EVENT CATALYST)
```

Similarly, verbs fail to map uniformly into a single type of CycL term: "sleeps" maps to the CycL collection #\$Sleeping, whereas "likes" maps to a relation

```
(likes AGENT THING)
```

This poses a problem insofar as SMEs have no preconceptions about what sort of CycL would serve as an appropriate "target" for their English expressions, and so might browse the KB for a 'catalyst' collection to use.

This problem is mitigated in Cyc, where parsing proceeds through a layer of indirection: Since any collection can be defined compositionally, a collection-construction procedure is used to produce ephemeral collections that can stand in for what might otherwise appear to be gaps in the collection hierarchy. In the context of parsing a sentence or question, this intermediate, or "I-CycL" representation, is canonicalized into to a more inferentially productive form, through the use of "reformulator" rules.

Thirdly, NL is highly ambiguous, whereas a logical language is precise. To get a sense of the ambiguity of the verb "to be," consider these pairing of sentences of the form "X is Y" and their CycL translations,

| | |
|------------------------------------|--|
| "Water is H2O" | (completeAtomicComposition-List Water (TheList Hydrogen Oxygen) (TheList 2 1)) |
| "Chile is a country" | (isa Chile Country) |
| "Santiago is the capital of Chile" | (capitalCity Chile CityOfSantiagoChile) |
| "Pure water is potable" | (genls (PureFn Water) Drink) |

| | |
|-------------------------|--|
| “Pure water is clear” | (relationAllInstance transparencyOfObject (PureFn Water) Transparent) |
| “Batman is Bruce Wayne” | (alterEgoOfHero Batman BruceWayne- MillionaireSocialite) |

This sort of widespread ambiguity represents two major problems: First, an English-to-CycL mapping for each possible meaning of "is" simply isn't feasible – too much time would be spent writing specialized parsing templates to cover each case. Second, relying on the user to clarify the ambiguity in each case would require the user to understand CycL sufficiently well to make parsing uninteresting. So, while the “intermediate CycL” approach has not resolved all problems, it has made significant progress toward handling such cases in a fashion that involves much of the KB’s knowledge and little assistance from the SME.

Ambiguity is a problem for NL generation, as well. What counts as the “right” English generation is dependent on context. These two CycL sentences,

```
(relationAllExists mother American FemaleAnimal)
(relationExistsAll myPresident American President-HeadOfState)
```

might equally generate as "every American has a mother" and "every American has a president," respectively. However, the quantifier scope is different in each case, and so might confuse an end user if the generation was too “straight-forward”. Or consider:

```
(likesRoleInEventType
  JohnDoe (PlayingFn Basketball-TheGame) doneBy)
(likesRoleInEventType
  JohnDoe (WatchingFn Basketball-TheGame) doneBy)
```

Either of these would generate naturally as “You enjoy basketball.” However, because these must be presented to the user as competing interpretations, the generation system knows to generate English more precisely than it might were the single sentence to be presented on its own. Though the resulting generation may be awkward and even sound stilted, the system purposefully generates more verbosely to decrease chances of the SME assuming KRAKEN meant something it did not.

1.3 The KRAKEN Architecture

KRAKEN is built around the Cyc system, and provides a range of interfaces to the Subject Matter Expert, including a simple HTML one, via a web browser. KRAKEN and the Cyc system also rely on various external data sources. Because the KRAKEN architecture changed significantly over the three years of the RKF project, each year's architecture is discussed separately below.

1.3.1 RKF Year One

In the first year of the project, illustrated by Figure 3, there was a single HTML interface; this was intended as a temporary interface, until a suite of Java applications using a common blackboard substrate could be developed. Supporting software included an early version of Northwestern's Analogy Server, ISI's WhyNot? Tool, and AIAI's Process Description system.

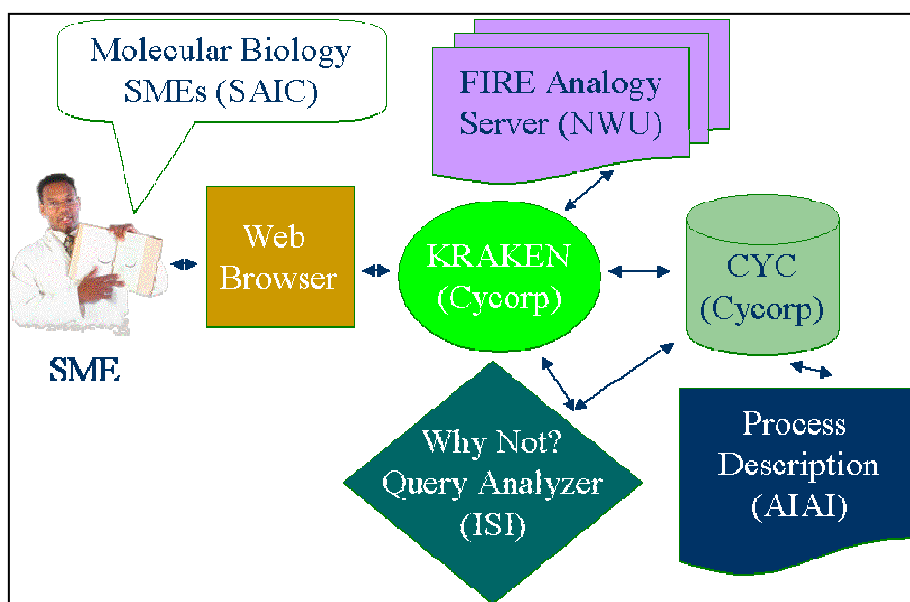


Figure 3: KRAKEN architecture for RKF Year One.

1.3.2 RKF Year Two

In the second year, illustrated by Figure 4, Northwestern deployed another tool, nuSketch BattleSpace, which provided a graphical interface for input and output of information and played a major role in the evaluation. AIAI extended their Process and Plan Representation support to include a Java interface. ISI continued to develop their WhyNot? Proof failure diagnosis Technology. Teknowledge continued work on their SME collaboration system SCOOP, which was refocused from a CVS-like knowledge repository to a cross-SME consistency verification. Cycorp made major improvements to the KRAKEN web interface, utilizing dynamic HTML (DHTML) and some Java applets, including the Guided Knowledge Entry tool (GKE) which allows for sentence editing and ontology browsing. Efforts to extend the technology were somewhat slowed by the degree of domain specific adaptation required due to the change in challenge problem.

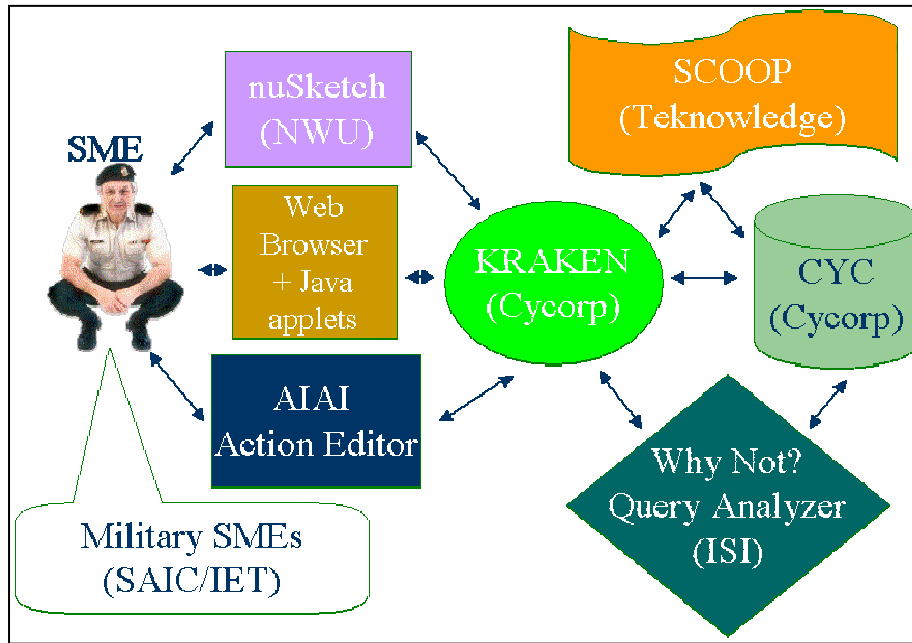


Figure 4: KRAKEN architecture for RKF Year Two.

1.3.3 RKF Year Three

In the third and final year of the project, illustrated by Figure 5, Cycorp deployed a larger range of Java interfaces (both applet and standalone applications, such as the Factivore, Query Library, and Analysis Diagram Tool) that SMEs used for Knowledge Entry (KE). The major external data source in this phase was ESRI's ArcMap GIS database, which was accessed via a lightweight XML-based interface provided by SAIC. ISI turned their attention to rule induction. Although not used directly in this year, Northwestern continued to improve their technology.

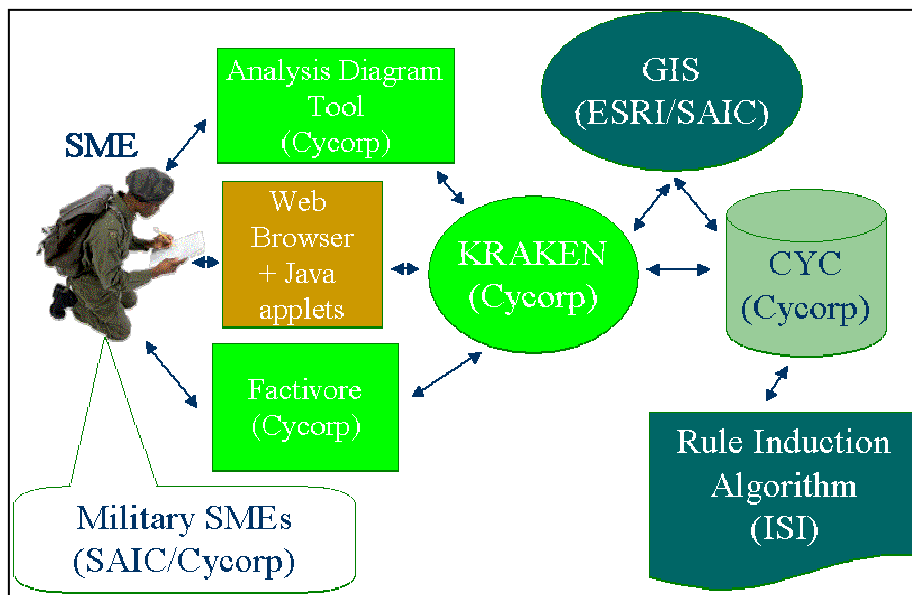


Figure 5: KRAKEN Architecture for RKF Year Three.

| Description | Fact |
|-------------------------|----------------|
| Year of birth: | 1955 |
| Month of birth: | January |
| Day of birth: | 3 |
| Year of death: | |
| Month of death: | |
| Day of death: | |
| Ethnicity: | Arab |
| Nationality: | Palestinian |
| Citizen of: | Palestine |
| Social Security Number: | |
| Resided in: | Italy |
| At this time: | 1980s |
| Was located in: | Iraq |
| At this time: | April 14, 2003 |

Figure 6: Kraken-based Factivore application being used to represent a new terrorist as part of a different project to form a Terrorism Knowledge Base.

1.4 Related Work at Cycorp

Over the course of the RKF project, there has been considerable synergy with other Cycorp projects. Three of them are worth further description here:

- A project in which Cycorp is working with SAIC to build a comprehensive Terrorism Knowledge Base (TKB) – This project pioneered the development of the Factivore, which extends the idea of KRAKEN’s Concept Refinement Interview to the use of prepared question templates. This tool, illustrated in Figure 6, relies heavily on Kraken NL technology to parse phrases, and also on the automated repair of SME assertions. RKF SMEs have been using this tool extensively to flesh out military domain concepts.
- A project to provide question answering abilities in the domain of Chemical Biological Radiological Nuclear and Explosive (CBRNE) threats – This project developed the Query Library, which uses the Guided Knowledge Entry (GKE) component developed in RKF Year Two in addition to NL generation. The Analysis Diagram Tool, used in the Year Three evaluation, incorporates the Query Library. The Query Library is also be used to evaluate the results of that evaluation.

- A project to provide question answering with human-readable justifications in the domain of high-school (AP) chemistry (see Figure 7) – This was a short-term evaluation project that largely used technology inherited from RKF to provide the explanations required for each answer. This starting point allowed for a principled use of compositional domain-independent natural-language generation; this contrasted sharply with the more bespoke systems used by other teams on the same project. This project made significant contributions to Cyc’s ability to lay explanations out, and to filter irrelevant or uninteresting proof steps; the lessons learnt from this project were drivers for the work on the CBRNE threat system.

In addition to the components that are directly part of the KRAKEN system, the technology developed for use by SMEs has also been adapted for use by Cycorp’s ontologists. See below under Discussion for more details.

Cycorp also received funding through the RKF program vehicle for work on DISA-Secure and Scenario Generation. Final reports for those projects appear as accompanying documents.

| |
|--|
| <p>Given (from the question): The acid-dissociation constant for benzoic acid is 6.3E-5. Benzoic acid and <i>BASE</i> form a conjugate acid-base pair.</p> <p>Applicable Rule: If</p> <ul style="list-style-type: none"> • <i>ACID</i> and <i>BASE</i> form a conjugate acid-base pair • and the acid-dissociation constant for <i>ACID</i> is <i>KA</i>, <p>then the base-dissociation constant for <i>BASE</i> is the ratio of K_w to <i>KA</i>. -- from Section 16.8 of <i>Chemistry: The Central Science</i></p> <p>Rule Application: The ratio of K_w to 6.3E-5 = 1.5873E-10.</p> <p>Conclusion: The base-dissociation constant for <i>BASE</i> is 1.5873E-10. Trivially: 1.5873E-10 ? 1.59E-10.</p> |
|--|

Figure 7: Cyc uses Kraken-developed NL generation in justifying its answer to a chemistry question in a commercially funded project.

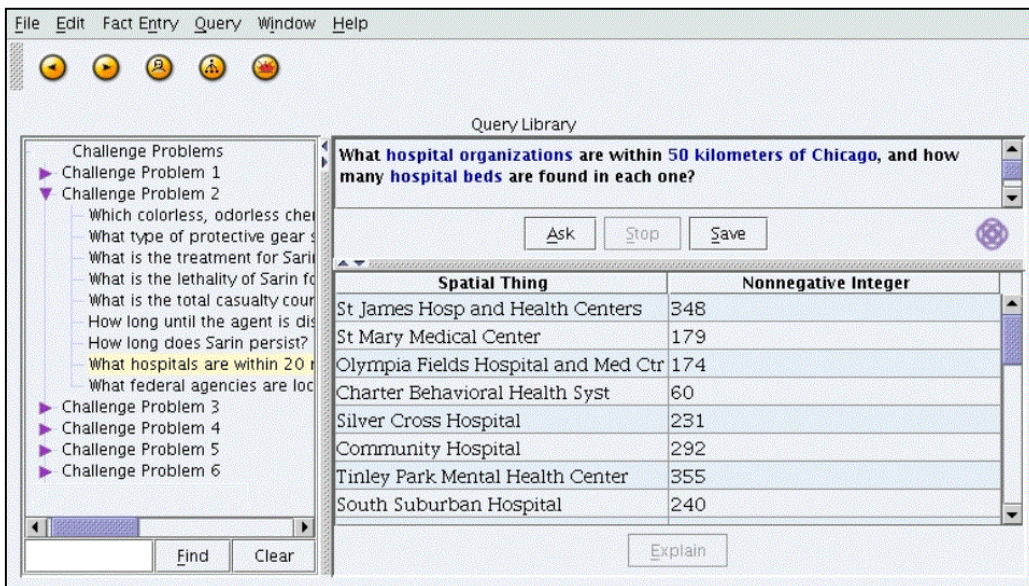
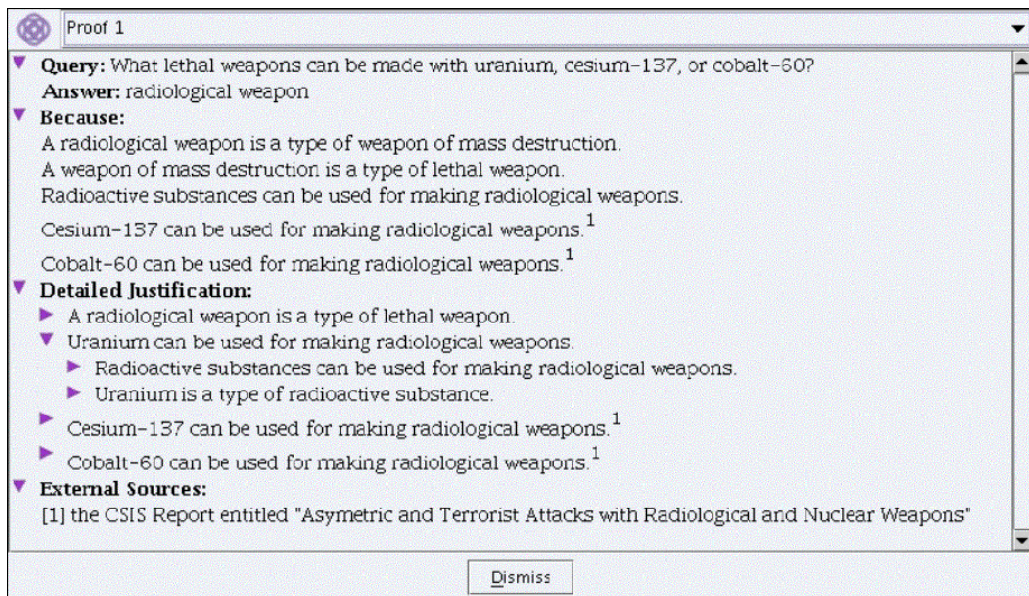
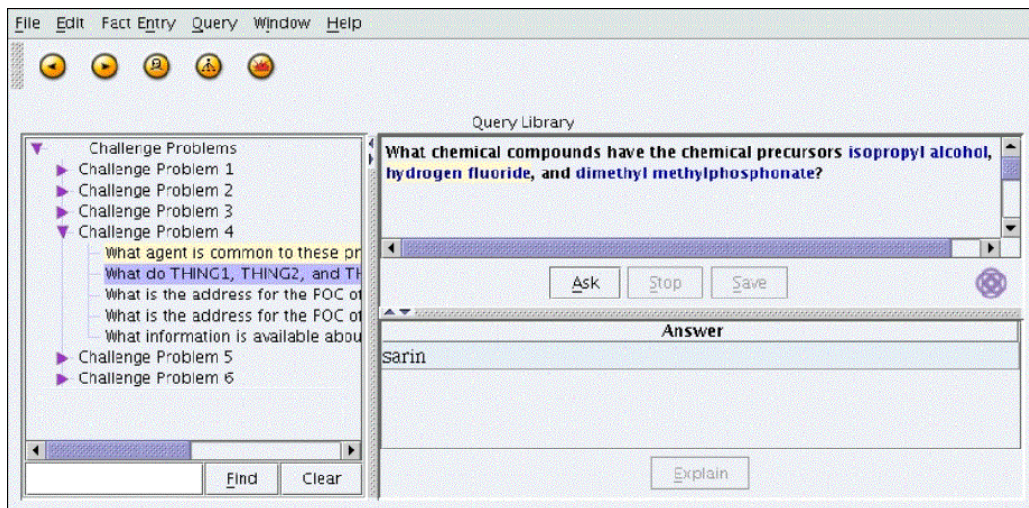


Figure 8: Three screenshots from the CBRNE threat system. At the top, Cyc's knowledge of chemistry is used to determine precursors. In the middle, Cyc is able to justify its answer with respect to source material that has been represented in the knowledge base. At the bottom, Cyc uses its Semantic Knowledge Source Integration (SKSI) to draw information from databases to answer the question.

1.5 Evaluations of KRAKEN

In each year of the RKF project, an evaluation was performed of the KRAKEN system. In Year One, the evaluation required a team of molecular biology SMEs to enter textbook information over the course of weeks. In the Year Two evaluation, two military SMEs spent a week entering Courses Of Action (COAs) and authoring rules to permit those COAs to be critiqued. In Year Three, the theme of COA evaluation was continued, and three military SMEs entered complex information about the terrain analysis process. The first two evaluations were performed by IET, and the third was internal. See Section 3 below for more details.

1.6 Summary of Collaborator Involvement

The following table summarizes the major contributions to the project by all collaborators over the three project years.

| | <i>Year One</i> | <i>Year Two</i> | <i>Year Three</i> |
|-----------------------------|--|--|---|
| <i>Project Focus</i> | Molecular biology | COA representation and evaluation | Terrain analysis process |
| <i>AIAI</i> | Provided a Process Description tool for representation of biological processes | Developed alpha versions of Plan Editor and Action Editor | |
| <i>ISI</i> | Provided WhyNot? system for inducing missing knowledge | Improved WhyNot? system | Provided rule induction system to suggest new rules from large bodies of data |
| <i>NWU</i> | Provided analogy reasoning system. | Improved analogy server, and provided nuSketch BattleSpace, a graphical sketching tool | Improved existing tools |
| <i>SAIC</i> | Provided SMEs and paraphrases of textbook | Provided SMEs | Provided SMEs and GIS integration |
| <i>Teknowledge</i> | Developed SCOOP tool for co-operative knowledge authoring | | |

2 Components of KRAKEN Technology

The following sections describe the development of KRAKEN throughout the project. Each section describes the specific context of the year's work, including the evaluation, and the toolset that was developed in response. While the overall direction of the project has been in line with the original goal, the specific challenge problem of each year had a significant effect on the actual tool set developed in response. For each year there is a discussion of the insights resulting. Taken together, the contexts and lessons learned link the three years together, pointing backwards to HPKB and forwards to future work.

2.1 Year One

2.1.1 Context

HPKB established that large, broad-coverage common-sense knowledge bases were useful in answering intelligence analysts' questions, in some cases providing answers to questions that analysts had previously thought unanswerable. The value of such systems thus established, DARPA turned its attention to issues surrounding the practicality of deploying such systems. A central obstacle to such deployment was the absence of any sort of automated or semi-automated knowledge-transfer process. The transfer of knowledge from the SME to the system required the expensive and time-consuming intervention of a trained ontologist. The ontologists had to first transfer the knowledge into their own head, and then manually translate that knowledge into the formal language usable by the system. The goal of RKF was thus to test whether knowledge entry tools could be developed that would enable analysts (and SMEs, generally) to write to a knowledge base more or less directly, thereby reducing the criticality of the AI expert's direct involvement.

The plan was for Year 1 to revolve around a Textbook Knowledge Challenge Problem (TKCP) in which SMEs would enter the content of select passages from a source text. In Year 2, IET set an Expert Knowledge Challenge Problem (EKCP) that was to involve adding knowledge from the Year 1 domain, without constraining the SMEs to the knowledge as captured by a particular set of textual passages.

Originally, the domain was intended to be biological and chemical warfare; however, with no official source text available, and the prospective construction one being both expensive and of questionable value, it was decided that the TKCP would focus on the sub-domain of molecular biology. An introductory-level undergraduate textbook — *Essential Cell Biology*, by Alberts, *et al.*, — was chosen as the source text. Accordingly, molecular biology graduate students, certain to be familiar with the material covered in the book, were used as SMEs. The domain chosen had several advantages: the domain was thoroughly type-level (see Appendix E), and so genuinely represented an area of knowledge (versus mere data); the domain was a highly specific sub-domain of many interesting fields (*e.g.*, biology, chemistry, chemical engineering, bio-chemical warfare), and so afforded the opportunity to provide knowledge-based systems with knowledge needed for deep reasoning across those fields; and the domain required the representation of complex processes, and so would involve the creation of background knowledge and the introduction of tools that were highly reusable (like DNA replication, a military course of action, for example, is another type of complex process).

The domain also provided, it was initially thought, a level of objectivity and clarity characteristic of hard science — there was nothing 'fuzzy' about the facts of molecular biology, and this was

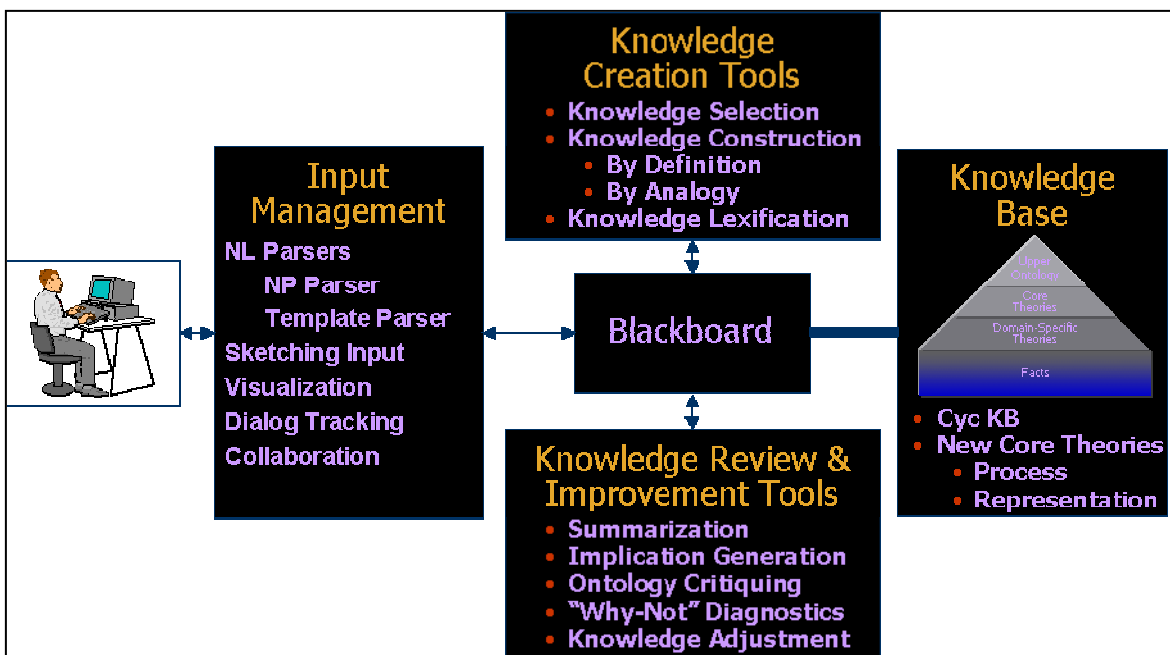


Figure 9: Original architecture design for KRAKEN. The final KRAKEN architecture is very similar, but with a discourse context within the Cyc KB replacing the blackboard.

thought to make the domain a good one on which to have their systems "cut their teeth." However, through the course of Year 1, it was discovered that the text made widespread use of metaphor, which introduced an unanticipated element of difficulty. For example, the behavior of individual molecules was more frequently than not described using agentive language; CycL, however, being a logically perspicuous language, is constrained in such a way so as to prevent its terms for describing agentive behavior from being used to describe the interactions among inanimate things.

2.1.2 Tool Set

RKF Year One took an atomistic approach: each of the tools was supposed to handle one and only one task and was conceived in a stand-alone fashion. Part of this had to do with the research involved; some of the tools were more clearly conceived than the others and aiming for a plug & play approach seemed most likely to lead to success.

The original KRAKEN design (as described by the PI, Dr Douglas Lenat, in the proposal) envisioned a blackboard architecture for integrating the individual tools and the contributions of the collaborators; unfortunately, the blackboard architecture implementation conceived by its initial implementers proved too elaborate to come together, and in the end a simpler, queue-based task approach was used and deployed.⁶

The primary focus was on identifying the sets of tools that were needed to make any knowledge additions at all. Thus, for each of the KB entity types — instances and collections, predicates, facts and rules, queries — there had to be creation tools, editing tools, and removal tools.

⁶ This experience, in part, drove much of the progress in later years in representing the state of the system and even of discourse processing in the KB itself.

Another group of tools was supposed to help ensure and improve the quality of the entities thus created — redundancy and contradiction detection as well as precision support was needed. Cycorp's collaborator ISI contributed the tool *WhyNot?*, which assisted in improving knowledge coverage and queries by identifying missing sub-proofs in failed queries.

Yet another group attempted to support the leveraging of existing knowledge — one of the advantages that a Cyc-based system brought to the table — for the creation of new knowledge; the Salient Descriptor tool (now called the Concept Refinement Interviewer) would ask the SME salient questions about any new terms introduced. The KRAKEN team also provided a rudimentary analogy developer as a placeholder, since the NWU team was planning to bring their analogy-reasoning engine to the table as their contribution.

The set of tools was then harnessed into a common user interface, which dictated the need for communication tools. In particular, the original user interface for KRAKEN (as presented at the New Orleans kick-off meeting) envisioned a primarily NL-driven interface that allowed cut and paste from source text. So there was a need for parsing support, generation support and ways in which the SMEs could provide lexical information for newly created knowledge, ranging from individuals and collections to sentences constructed from predicates that the SMEs had defined. Cycorp collaborator SAIC supported this part of the effort by providing SME paraphrases for parts of the biology text that simulated how real users might have formulated the knowledge content.

Another category of tools supported the SME's ability to find out what the KRAKEN system already knew. A term finder and an assertion finder, as well as a mechanism for listing all the facts known about a term in natural language were devised. Example sentences and example queries could be listed, both as building blocks for new facts and questions, and as a means to make it obvious what information might be missing. Late in RKF Year One, an Ontology Browser was added to the system, to show the knowledge hierarchies underlying the instances and collections. History mechanisms to keep track of useful terms and facts encountered in one's searching through the knowledge base were added as well.

Furthermore, there was a set of tools dictated by the mechanics of the evaluation. A test suite tool allowed grouping the evaluation questions, once authored, so that progress could be constantly measured. In order to make swapping out the programs less disruptive, support for saving intermediate stages of work already done, and even the interesting terms accumulated, was added. (The SCOOP tool for cooperative knowledge authoring, which could have fulfilled some of the same roles, was not ready for deployment by the start of the SME evaluation and did therefore not participate during RKF Year One.)

Finally, the domain of biology, as a knowledge engineering problem, is particularly heavy on type-level information and reasoning. This is particularly true in the domain of biological processes, and the focus of the evaluation on RNA and DNA transcription made this even more so. As a consequence, a special tool was developed that would support the description of processes and their sub-processes, roles played and action taken, etc. The underlying

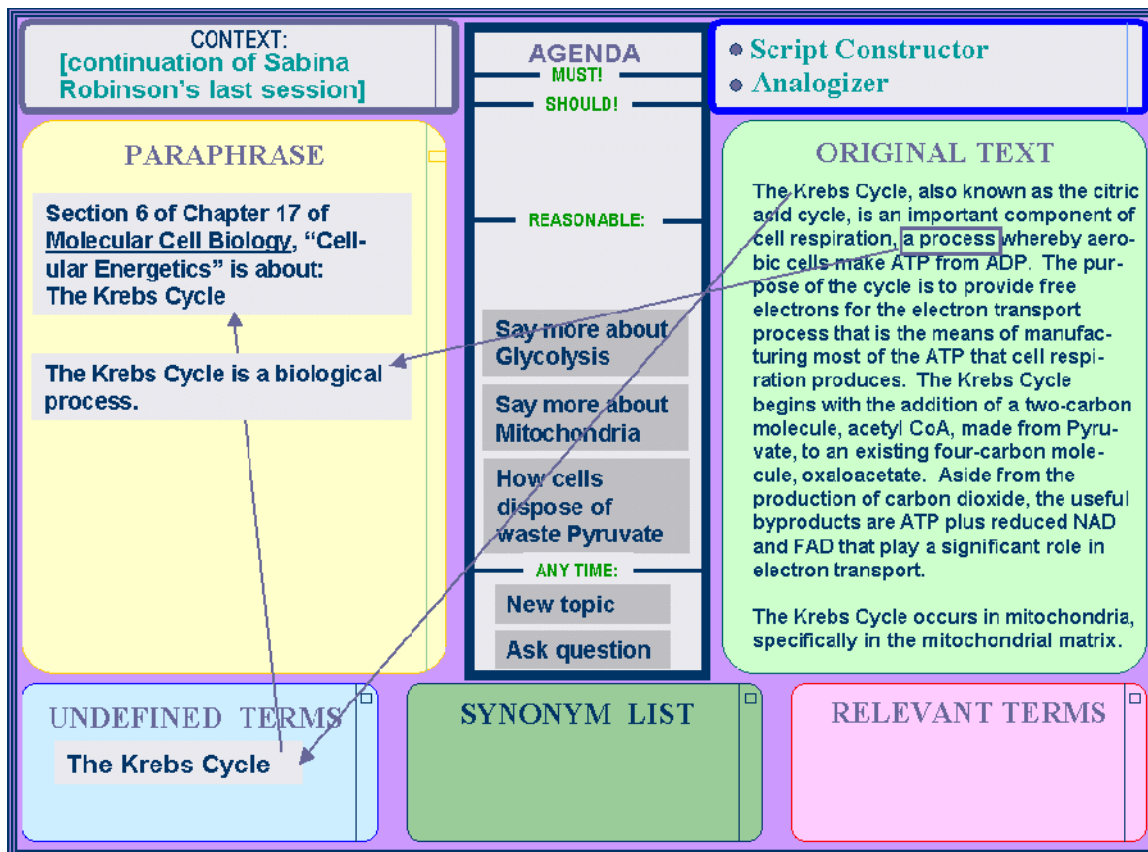


Figure 10: Doug Lenat's original UI proposal for KRAKEN from the New Orleans PI Meeting.

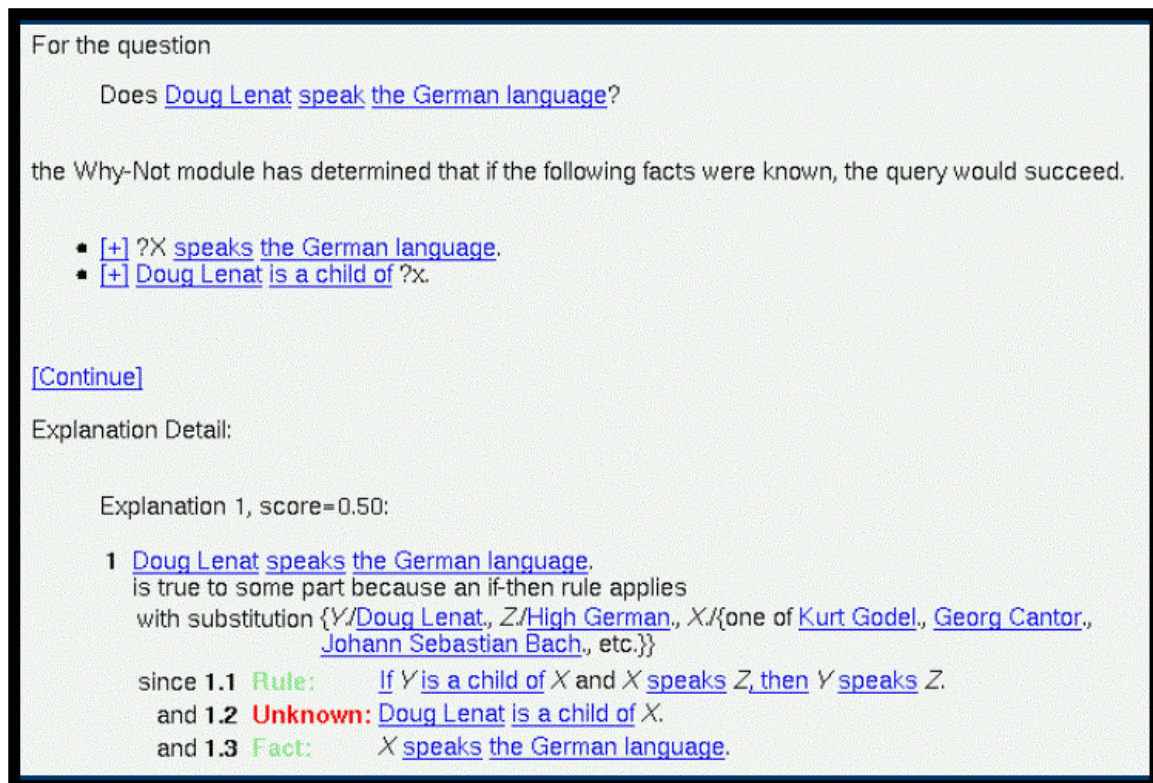
representations for these processes were heavily supported by ontology contributions and research provided by subcontractor AIAI.

2.1.3 Lessons Learned

Considering the size of the task undertaken, the KRAKEN tool set functioned reasonably well. Most of the individual tools worked according to their design. The Salient Descriptor especially was very helpful and produced good knowledge at decent speed when left to drive the course of the conversation. As for the NL problem, which has been recognized as a research problem in its own right for many years, Cycorp did make significant inroads in understanding simple sentences and using background knowledge to resolve common cases of ambiguity.

At the same time, many of the tools were too closely tied to the underlying knowledge engineering concepts that were not necessarily the way SMEs conceptualized the domain. While the NL intermediate representation managed to cushion some of this impact, there must have been many cases of the SMEs simply not knowing what to try next.

In some cases, the KRAKEN team had not succeeded in finding suitable user interface metaphors for all the knowledge engineering concepts. Specifically, the story-telling based approach to question construction failed to take into account the fact that the biology domain, as mentioned above, was a type-level domain; story-telling works best when one can introduce individuals. Rule construction and predicate creation, which were based on the same metaphor, fared little better. The process description tool proved very challenging, due to the complexity of the processes involved, and due to its lack of an adequate visualization metaphor for the process steps.



For the question

Does [Doug Lenat speak the German language?](#)

the Why-Not module has determined that if the following facts were known, the query would succeed.

- [\[+\] ?X speaks the German language.](#)
- [\[+\] Doug Lenat is a child of ?x.](#)

[\[Continue\]](#)

Explanation Detail:

Explanation 1, score=0.50:

1 [Doug Lenat speaks the German language.](#)
is true to some part because an if-then rule applies
with substitution {Y/[Doug Lenat.](#), Z/[High German.](#), X/{one of [Kurt Godel.](#), [Georg Cantor.](#),
[Johann Sebastian Bach.](#), etc.}}

since 1.1 **Rule:** [If Y is a child of X and X speaks Z, then Y speaks Z.](#)
and 1.2 **Unknown:** [Doug Lenat is a child of X.](#)
and 1.3 **Fact:** [X speaks the German language.](#)

Figure 11: ISI's WhyNot tool suggests a plausible fact that could be added to make a failing query succeed.

The approach of using an HTML user interface for the rapid development of the individual tools became a burden, primarily because of the lack of dynamic updating. The lack of dynamic update capability also increased user wait time, because all of the information that the user might possibly need had to be pre-computed up front.

2.2 Year Two

2.2.1 Context

In the second year of the program effort, and under the impact of the events of September 11th, 2001, the focus of the program shifted from molecular biology to military course-of-action description. This was a non-trivial shift in the nature of the knowledge engineering required, quite apart from the change in domain. Knowledge in both domains can obviously be organized in highly systematic ways, but the way in which a hard science like biology is systematized is somewhat different from that of a more holistic discipline like military doctrine. Indeed, while there is general agreement among the knowledge experts about broad evaluative criteria in military doctrine, a much greater leeway is allowed with respect to the details. In this sense, military decision making is as much an art as it is a science, in that expert practitioners often find it difficult to articulate their own decision making process in full detail.

In response to the change in focus, AIAI converted their Process Description system into a rule-authoring tool called the Action Editor. It was determined that Teknowledge's SCOOP system would not be very useful to an individual authoring a military course of action, and SME collaboration was deemphasized for year II.

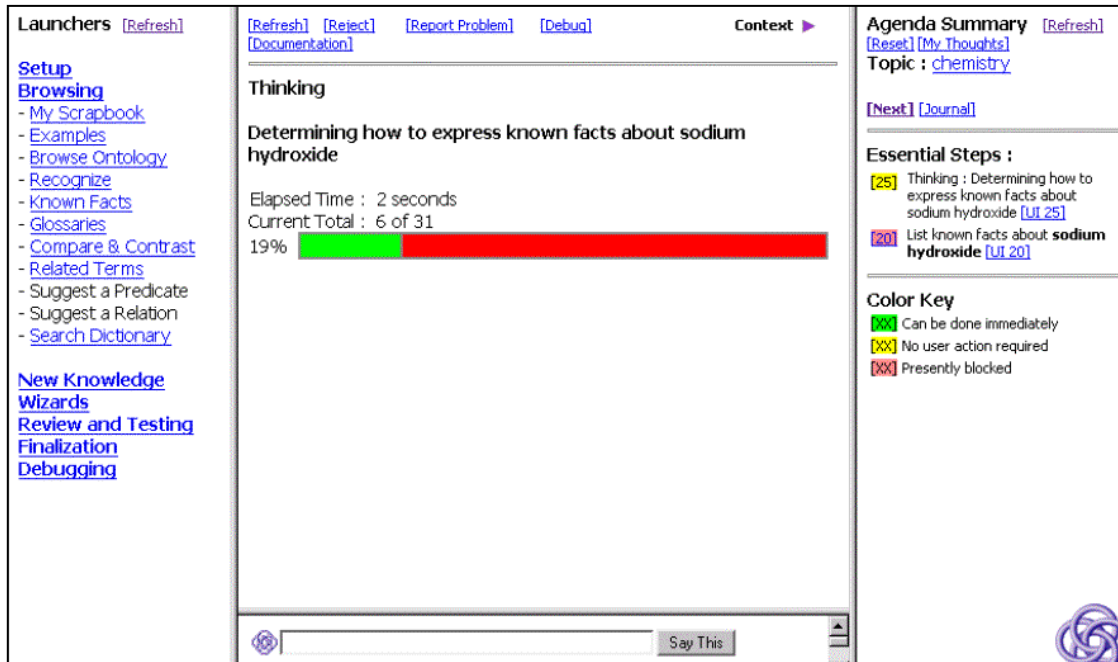


Figure 12: KRAKEN's interface at the end of RKF Year Two. The system is generating a page of facts, and is using DHTML to keep the user apprised of progress.

2.2.1.1 Tool Set

KRAKEN always had been conceived as a tool for generic knowledge entry. Internally, the KRAKEN team was testing the use of the tool to describe elements of popular culture, such as politicians, music bands and film stars, just to ensure this goal. But even during RKF Year One, the SMEs had always been inclined – not without reason – to press for KRAKEN, and especially KRAKEN's NL generator and tool interfaces, to be specialized into something optimized for the authoring of the year's domain knowledge. For example, a tool was needed to perform OCOKA⁷ evaluation of the state of the battlefield, using the background knowledge, the situation description and the military analysis rules specified by the SMEs.

⁷ Military classification of factors affecting use of terrain: Observation and fields of fire, Cover and concealment, Obstacles, Key terrain, and Avenues of approach.

The emphasis on military courses of action brought a new tool to the fore, NWU's nuSketch Battlespace battle sketching tool. This gave an additional enhancement to the user interface experience, but brought additional problems as well, as the Cycorp team now had to figure out how to include the information gathered by nuSketch into the reasoning process, how to match up the subtly different ontologies, and how to make the integration between the two systems, Cyc and nuSketch (which don't even run on the same operating system) work in real time.

Beyond that, though, the KRAKEN team had to avoid making tools overly specific to the military domain, which entailed an inability to honor some suggestions by SAIC SMEs, in order not to violate the general spirit of the RKF program. For itself, the team wanted to improve the overall system, especially the user interface experience, and focus on the tasks of query and rule construction, as the exemplary setup of the military units might provide the fodder for doing so. In addition, the team wanted to build on the strengths of the tools that had proven themselves useful in the RKF Year One evaluation.

The Salient Descriptor was improved and extended to use the existing rules that the system already had as a basis for coming up with new questions to ask the user. The basic assumption was that if the system were attempting to get the rules it already had learned to fire, then the SMEs would be led toward providing the information needed to make the OCOKA evaluation work. In addition, the Salient Descriptor gave more control to the user, trying harder to stay on topic and allowing the choice of what questions to ask next and the like.

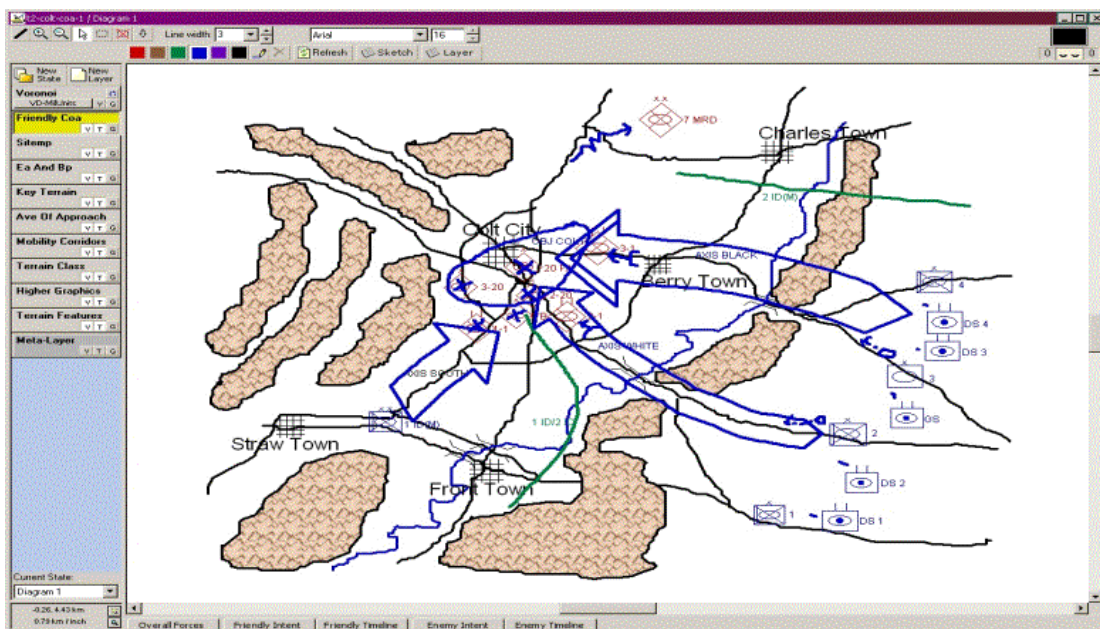


Figure 13: Northwestern University's nuSketch BattleSpace.



Figure 14: In the Concept Refinement Interview (using the Salient Descriptor), the system induces questions it believes would lead to plausible and interesting statements.

In terms of knowledge presentation, the ontology browser was revamped and extended to include salient example sentences about the terms. A new user interface metaphor, the notion of a Glossary of terms, was added, and infrastructure developed so that KRAKEN could generate these Glossaries itself.

The user interface problems were addressed by turning to the use of dynamic HTML, which provided less screen clutter and more presentational possibilities, and a first round of adding Java applets as user interface components because of the higher degree of interactive capabilities. The Sentence Editor and the Question Editor were chosen to be replaced with the Guided Knowledge Entry tool (GKE), which allowed editing all of the parts of a stated sentence, substituting natural language components for parts, and even browsing the environs of the terms involved using a compact hierarchy browser.

2.2.2 Lessons Learned

During the first part of the final SME evaluation, things went rather well. The Glossaries were so convincing that SMEs at first could not believe that they were computer-generated. The combination of SMEs sketching out the position of the individual military units on the battle field using nuSketch, followed by Cyc importing the information from nuSketch and passing it as background information to the Salient Descriptor provided for a decent user experience and made for rapid knowledge formation. The main drawback of that setup was the strange experience of working with two separate, although coupled, user interfaces.

The second part, OCOKA based COA evaluation, was less clearly successful. The SMEs still had problems formulating the rules, and the tools for the construction of queries and rules were still insufficiently flexible. Part of the problem continued to be the limitations of the HTML interface, which did not give the system as much chance to support the SME as the KRAKEN team would have liked to. Nevertheless, the SMEs managed to author several high quality rules that combined well to explain a specific aspect of the course of action.

Dr. Kerry Hines, who was an IET evaluation SME in Year Two, and a Cycorp consulting SME in Year Three, identified three key issues encountered in the Year Two KE:

- “First, we had to develop meaningful (measurable) evaluation criteria, when COA ‘standards’ (like the principles of war) are generally expressed in platitudes (e.g. achieve surprise, establish overwhelming force superiority at the decisive point, allocate minimum force to secondary efforts).

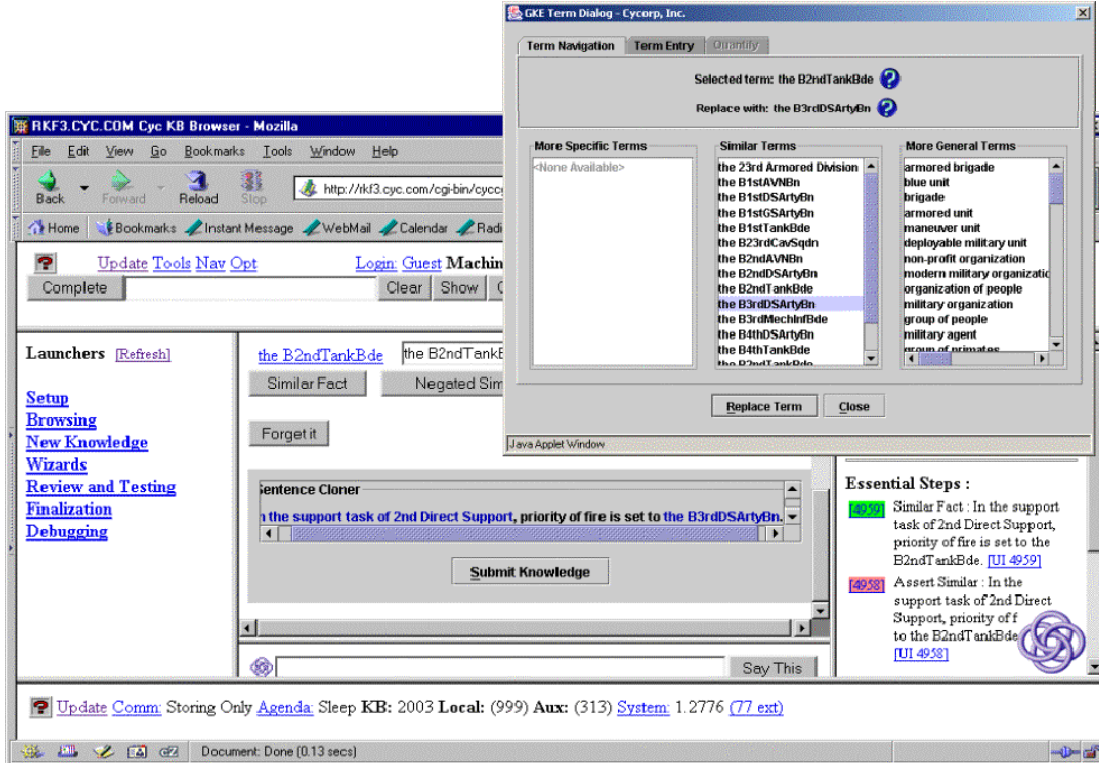


Figure 15: Guided Knowledge Entry (GKE) permits individual terms in a sentence to be replaced using an ontology hierarchy browser.

- “Second, we had to develop an understanding of how to enter the rules; that is, how to break down a rule into manageable entry pieces (we had to communicate with KRAKEN on KRAKEN’s conditions).
- “Third, the process of actual entry required that we deal with the limited and sometimes erroneous military knowledge in KRAKEN, which we generally gained awareness of through trial and error.”

With the end of RKF Year Two, the RKF team was split up, as some of its team members were needed on other projects that Cycorp had since the beginning of RKF begun to participate in. Those team members took their experience from RKF with them and began to reevaluate the tool sets for their new projects. This is worth mentioning in this context, because the work done in these projects not only built on top of the RKF tools that had been developed, but also brought synergies which the remaining RKF team was able to exploit in Year Three.

Despite the success of the evaluation, the limiting factors of the knowledge acquisition process

```

If two brigades in one battalion are being attacked, then the brigade is being attacked.
(implies
  (and
    (isa ?THE-1ST-TASK AssignedTaskType)
    (targetInAttackTask ?THE-1ST-TASK ?THE-1ST-BATALLION)
    (isa ?THE-1ST-BATALLION Battalion-MilitaryEchelon)
    (subOrganizations ?THE-BRIGADE ?THE-1ST-BATALLION)
    (isa ?THE-BRIGADE Brigade-MilitaryEchelon)
    (different ?THE-1ST-TASK ?THE-2ND-TASK)
    (isa ?THE-1ST-TASK AssignedTaskType)
    (different ?THE-1ST-BATALLION ?THE-2ND-BATALLION)
    (isa ?THE-2ND-BATALLION Battalion-MilitaryEchelon)
    (subOrganizations ?THE-BRIGADE ?THE-2ND-BATALLION)
    (targetInAttackTask ?THE-2ND-TASK ?THE-2ND-BATALLION))
  (targetInAttackTask ?THE-1ST-TASK ?THE-BRIGADE))

If a unit is assigned to attack another unit, then it is responsible for that unit.
(implies
  (and
    (isa ?UNITA ModernMilitaryUnit-Deployable)
    (different ?UNITA ?UNITB)
    (isa ?UNITB ModernMilitaryUnit-Deployable)
    (targetInAttackTask ?THE-ATTACKING-TASK ?UNITA)
    (unitAssignedToAction ?THE-ATTACKING-TASK ?UNITB)
    (isa ?THE-ATTACKING-TASK AssignedTaskType))
  (unitAssignedToUnit ?UNITB ?UNITA))

If no blue unit is assigned to attack some red unit, then the COA can be criticized with respect to Engagement of Enemy Combat Power.
(implies
  (and
    (isa ?COA CourseOfAction)
    (coaRedUnit ?COA ?RED)
    (unknownSentence
      (thereExists ?BLUE
        (and
          (isa ?BLUE BlueUnit)
          (unitAssignedToUnit ?BLUE ?RED))))))
  (evaluatesDimensionOfCOA ?COA engagementOfEnemyCombatPowerInCOA
    IneffectiveForPurpose))

```

Figure 16: Three rules created by SMEs in the Year Two evaluation. Note that these three rules can be chained together to permit the evaluation of a Course of Action’s effectiveness with respect to engagement of enemy combat power.

were performance and interface. Part of the problem stemmed from the fact that even work-horse tools such as the Salient Descriptor were only asking the user one question at a time, in arbitrary order; some of the military SMEs found this especially annoying, as it forced them to sit through many questions that the system thought might be interesting, but which were irrelevant to the military problem at hand, and sometimes completely inappropriate. In addition, the Salient Descriptor would recompute the question set for each new term of the same type, ensuring greater currency, but failing to learn aggressively from the success of past experiences. This whole process, moreover, was somewhat slow, due to the amount of computation involved.

Thus, the notion of formula templates was conceived, which provided a notation for questions to ask the SME and how to ask them as a group; thus, it provided the organizational infrastructure to construct a questionnaire-like form interface (called the Factivore) exclusively from knowledge descriptions inside Cyc. In this interface the rendering is done by a separate process; therefore, the user interface remains responsive while the Cyc server is engaged in computing new information or storing the SME's results. This eliminates the waiting that the SMEs had faced during the previous two years, and also lets the user in which order facts should be entered.

In the same vein, the system for storing RKF SME test questions in the KB was extended and provided with an organizational infrastructure, dubbed query folders. For both of these infrastructure components, these other projects developed Java-based standalone applications, the Factivore and the Query Library. In its initial deployment, the Factivore templates were laboriously handcrafted, in some ways a step backward from the Concept Refinement Interview, but this was outweighed by the benefits of the interface.

2.3 Year Three

2.3.1 Context

For RKF Year Three, the domain remained the military evaluation of battlefields and battle operations, but the approach was changed. The emphasis was shifted to consolidating the gains and using the authoring tools to capture subject matter expert level domain knowledge in sufficient quantity that a "legacy" system would be available to other DARPA projects to document the success of RKF.

In addition, the KRAKEN team was reduced: AIAI's and Teknowledge's subcontracts had expired; NWU's nuSketch would not be relevant in this context, especially given the consistent feedback by the SMEs that nuSketch lacked a needed GIS integration, and the application for the analogy-reasoning engine in this domain was equally unclear; ISI was still contributing, and it was decided that, in an effort to start alleviating the difficulties in rule construction in year II, they should focus their efforts on developing an approach to rule induction based on exemplars.

Thus, the evaluation-directed section of the KRAKEN team was primarily Cycorp and SAIC, who both provided military SMEs. One conclusion from Year Two was that a solid GIS integration would be of enormous value to a terrain evaluation system, so the SAIC team was also engaged to use its GIS capabilities, in order to ground the analysis of the battle situation in real world data. Thus, Cycorp's SMEs developed a battle scenario using the Fort Hood GIS data set as the basis for military analysis.

2.3.2 Tool Set

Often, tools fail not on technical grounds, but because their developers did not adequately anticipate the appropriate metaphors that make sense to the intended audience. But sometimes the developers become lucky and their audience can describe to them how their current approach is falling short. The biggest tool contribution of the third year of RKF resulted from just such an insight of Cypcorp's lead SME, Dr. Kerry Hines.

Dr. Kerry Hines brought to the table his experience as an evaluation SME for RKF Year Two, where he had been a co-author of the few high quality COA evaluation rules that the RKF Year Two evaluation had produced. Dr. Hines' insight from that experience was that the rules, if stated in isolation, were simply too complicated; the preconditions would be too many to enumerate and impossible for humans to verify. The way that human subject matter experts got around this problem in the real world was by developing evaluation strategies that allowed the SME to focus on one problem at a time, in the full understanding that all of the previous concerns were still 'active' as preconditions to the current situation. At the RKF Year Three PI meeting in San Diego, Dr. Hines sketched such an evaluation process for the lead developers.

Out of this sketch grew the Analysis Diagram Tool, the only major addition to the RKF Tool suite in Year Three. In Year Two, rule complexity was a significant issue, and so this tool supports a method of rule construction where the process represented is broken down into a series of questions (and conclusions), each of which is contextualized by the preceding queries. This allows SMEs to stay closer to their intuitive granularity and format for questions and rules, by making explicit their underlying practice. The Analysis Diagram Tool makes use of the Query Library, which was extended to be a query constructor as well, so that the SMEs could build their

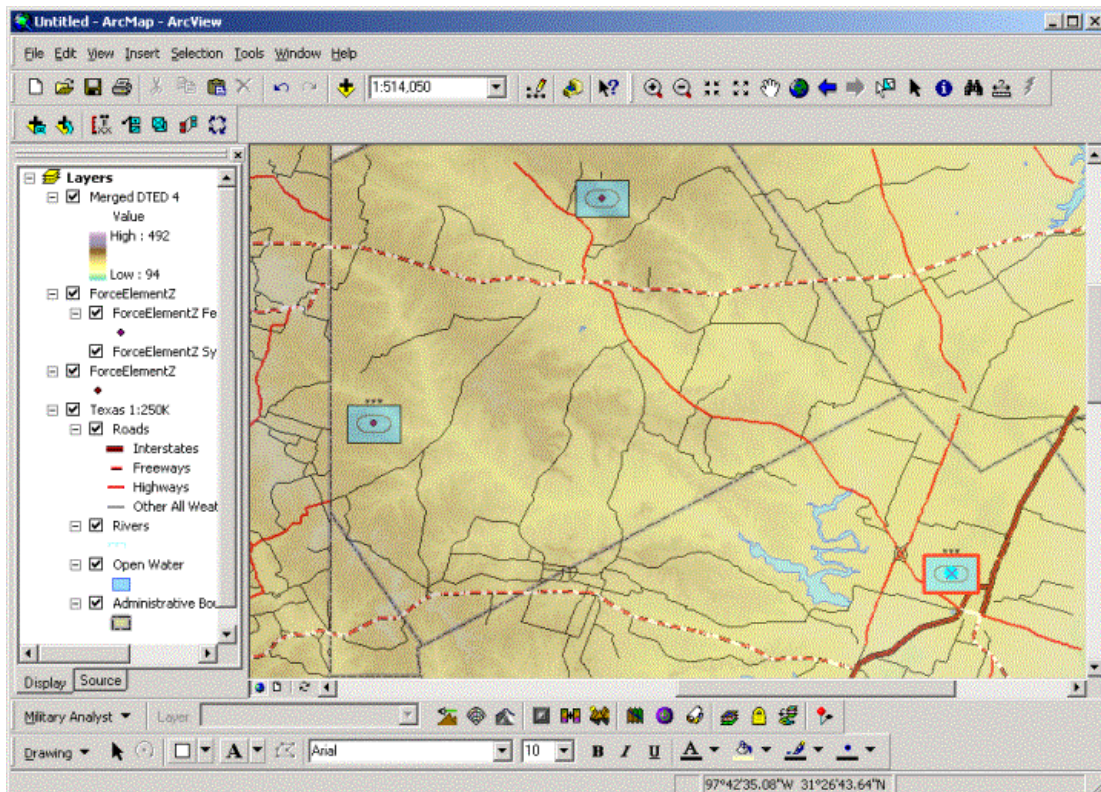


Figure 17: ESRI ArcView, the GIS of choice.

questions and place them in their diagrams using drag-and-drop. For this purpose, the example questions were migrated into the Query Library so they could be easily reused and shared.

In order to facilitate the description of the background of the battle scenario at Fort Hood, the Factivore was integrated as yet another Java applet into the existing KRAKEN interface. The new order of tool use was now to first create a new term, choose the template that would allow for its initial description, and then to have the Concept Refinement Interviewer (based on the Salient Descriptor) look into those things that were not described in the templates.

SAIC provided the KRAKEN team with a minimal GIS system that is implemented as an XML wrapper over a set of class libraries running on ESRI's ArcGIS system, the GIS tool of choice for both industry and the military. Access to the GIS system and semantic interpretation of its query results was integrated with the Cyc system, allowing at least some of the queries that the SMEs developed to be backed by real-world GIS data. Cyc has a general capability to integrate with external sources of semantic knowledge, which can be used to tie together an array of legacy systems and massive bodies of data into an intelligent system.

| Description | Fact | | |
|--|------------------|--|--|
| Unit size: | battalion | | |
| Unit specialty: | armored unit | | |
| Specialty activity: | | | |
| Equipment type: | M1A1 Abrams tank | | |
| <i>Number:</i> | 44 | | |
| <i>Allegiance:</i> | the Blue Force | | |
| Specialized equipment: | tank | | |
| Max unit road speed (ideal conditions): | | | |
| Max unit off-road speed (ideal conditions): | | | |
| Terrain type: | mountain | | |
| <i>Trafficability:</i> | no-go terrain | | |
| Typical type of obstacle for unit type: | | | |
| <i>Obstacle role:</i> | | | |
| Higher headquarters: | | | |
| <i>Number:</i> | | | |
| <i>Allegiance:</i> | | | |

Types of Units

Figure 18. Cyc Factivore Template

The Salient Descriptor, previously part of the Concept Refinement Interviewer, was extended to support the automatic generation of Factivore templates. This brings together the advantage of dynamically induced queries with the superior Java interface.

In the attempt to improve the quality of the natural language interaction, development towards a new sentence-based tool called the Interlocutor was undertaken. Specifically, design work was done for tracking the individual interpretations of the user's utterances in the Cyc knowledge base, so that Cyc's truth maintenance system could be used to undo interpretation choices if the continuing dialog showed that these became untenable.

The Interlocutor also attempted to address a problem that was first seen in RKF Year One with the cloning of example sentences, and which appeared again with the Factivore: given a blank slot in a user interface, SMEs will attempt to complete the implied English sentence, which may or may not fit syntactically or semantically into the underlying logical representation. However, given the Interlocutor's knowledge base of stored parse representations and a representation of the logical form that the Factivore is attempting to get filled in, the Interlocutor will be able to substitute the answer into the syntax tree of the formula and perform a reparse into logical form, thereby handling the resulting syntactic and semantic transformations gracefully. See the Appendix for more details.

2.3.2.1 The Analysis Diagram Tool

The Analysis Diagram Tool (ADT) is the largest innovation in KRAKEN's interface modality that was achieved in RKF Year Three. It therefore deserves a more in-depth treatment of its vision, design and status.

2.3.2.1.1 ADT Vision

As was seen in Year Two, the rules entered by RKF SMEs using Cyc are all somewhat complex, and hence simultaneously demonstrate not only the success achieved in getting SMEs to enter rules, but also the failure of such systems to move the interface discourse into the SMEs' domain.

An important feature of expert intelligence analysis knowledge is that it is much more natural for SMEs to speak of it procedurally than in terms of static situations and universal rules. Declarative knowledge-modelers and (with some overlap) those aiming for reusable knowledge representation rather than narrow-context procedures have been aware of this for some time, but have seen it primarily as a communication barrier.

A major opportunity has therefore been missed, because it is the procedure, or more accurately the reasoning history, that sets much of the context for the experts' reasoning, and makes their decisions manageable. That is, the expert does not need to consider an enormous number of antecedents when reaching any conclusion because the expert has already used the general situation to narrow down what decision procedures to use and what factors to consider, and more importantly, *knows what has already been considered and decided*. This history is the part of the context that gets lost during shift changes, the part that can get lost when presenting decisions as static analyses and the part that is not being harnessed in the modeling of the knowledge and reasoning.

In May of 2003, Kerry Hines drew something he called a ‘decision tree’ to depict some of the interactions between factors in evaluating battlefield terrain.⁸ In discussing this document, it was realized that there was no reason that SMEs could not enter knowledge in such a format directly. This format is both more intuitive and more natural for the task than direct rule authoring. It was also observed that, at any point in the analysis diagram, the questions were relatively straightforward, and the SMEs could articulate relatively simple rules for answering them, because a position in the analysis diagram itself represents the reasoning context — what the situation is, what has already been asked, and what is yet to come before a final conclusion is reached. It was therefore determined that this form of representation was amenable to both human discourse and machine reasoning.

2.3.2.1.2 ADT Requirements

The following high-level requirements were identified for KE tools in general, and the ADT in particular:

- **Intuitive:** The system must support representation in a form close to those used in the SMEs’ problem domain, rather than pulling SMEs towards formal inference-friendly representations.
- **Revisable:** SMEs must be able to go back and edit, remove, or rearrange portions of knowledge previously entered, and this editing facility must be backed by robust truth maintenance.
- **Modular:** Teams of SMEs must be able to divide subtasks among themselves, and must be able to request input on independent portions, or save portions of work for development by someone with more relevant experience, without being held up from other tasks. They also must be able to revise sub-portions of analysis knowledge without having to go back and revise the rest.
- **Verifiable:** SMEs must be able to independently test that the knowledge they have entered enables the desired reasoning and does not go awry, without any assistance from knowledge engineers. Similarly, they must be able to determine the causes of test failures unaided.
- **Powerful:** SMEs must be able to represent their analysis processes with an appropriate degree of complexity. They must not be constrained to implement only gross simplifications of their regular procedures.
- **Flexible:** The analysis representation must not be tied to a specific problem domain, but must have the power to represent a wide range of fields of expertise.
- **Incremental:** It must be possible for the SME to represent simplified working models of their process and then refine them in a series of evolutionary steps towards the final product.
- **Collaborative:** The system must enable multiple SMEs to work together in different locations or at different times, acting as a medium of information exchange.

2.3.2.1.3 ADT Design

It was determined that effective use of an Analysis Diagram Tool to perform terrain analysis would have to incorporate the following elements (with some overlap):

⁸ In fact, it is rarely a tree, in the graph-theoretic sense of having a maximum in-degree of one, and it does not necessarily reach decisions in all cases. For these reason, such representations are referred to herein as Analysis Diagrams.

- Query construction: A means for SMEs to represent the queries underlying each choice node.
- Conclusion construction: Similarly, a means to represent the output conclusions of the process.
- Diagram construction: A means for SMEs to represent the nodes and edges of their diagram, and associate them with the relevant semantics.
- Result merging: A means to collapse multiple, potentially complex query results onto a single qualitative value.
- Script construction: A means to represent the graphical process as a script.
- Script execution: A means for Cyc to execute the process described, and report on the results.
- GIS integration: A means for Cyc to obtain relevant terrain data and graphically-represented information about the terrain.
- Collaboration support: A means for SMEs to share and merge their work.

2.3.2.1.4 ADT Status

By and large, the Analysis Diagram Tool and supporting systems were completed to the extent that they demonstrate a proof of concept. Many issues remain outstanding and hence required manual intervention for the evaluation, but in most cases an automated solution is clearly possible given more resources.

2.3.2.1.4.1 Query Construction

SMEs were able to construct queries using the Query Library component, originally developed for a CBRNE threat system. This component was integrated into the Analysis Diagram Tool, and augmented in a number of ways:

- Query merging: It is often necessary to construct a new query from two simpler existing queries. In doing so, an equality mapping must be drawn between the variables used in each query. Many conceivable mappings are excluded by the argument constraints on the relationships used, but in many cases it is still necessary for the SME to indicate the intended combination. This was implemented as a variable mapping table, with the variables from the two queries appearing in the rows and columns, with interior cells either excluded or available for joining.
- Query saving: The ability for the SME to save new queries was developed.

Query construction at the end of RKF Year Three is therefore significantly better than that used for RKF Year Two. The following open issues remain:

- More automated variable mapping: In many cases, the system should be able to determine likely semantic intent (e.g. by knowledge of typical argument types), and eliminate irrelevant options (e.g. arbitrary choice between the arguments of a symmetric predicate). Although the variable mapping table is a tremendous improvement over previous interfaces, it should also be possible to represent the potential mappings, and the differences between them, in more intuitive ways.
- In general, use of the Query Library depends on having the general domain (in this case military terrain analysis) prepared in advance with example queries and builder query fragments. It is still not easy for a SME to construct a query using relationships in a way that has not been anticipated. To improve this, it will be necessary to provide better search facilities to give access to the vast range of relationships in the Cyc KB, and a way to induce query fragments from those relationships.
- The Query Library is currently limited to combining queries by simple conjunction, and the SME therefore cannot introduce disjunction, negation, implication, or quantification.

It is also difficult to introduce functional terms. In practice, these restrictions did not prevent the SMEs from representing most of the queries they wanted to enter. The primary barrier to adding these facilities is one of high-level interface design — enabling the user to comprehend their meaning without requiring them to be skilled logicians; hence, their benefit must not outweigh their cost in terms of ease-of-use.

- In the Year Three evaluation, the Query Library permitted the SMEs to construct some queries incorrectly or even invalidly without adequately alerting them to their error. To mitigate this in the future it will be necessary not only to improve the presentation of queries to the SMEs, but also to augment Cyc's ability to correct certain common classes of error (e.g. type/instance confusion) automatically.

2.3.2.1.4.2 Conclusion Construction

In general, conclusions are just like queries, only without unknowns. Supporting them was a simple matter of providing a menu option for a distinguished node type, and then using the same Query Library facilities. Example conclusions were readily determined from the desired output of the analysis process. Conclusions have the same outstanding issues as for query construction, but with less impact because of their simpler nature.

2.3.2.1.4.3 Diagram Construction

Graph visualization technology had already been developed in RKF Year One and Year Two. Graph editing facilities were developed for a separate link discovery project within Cycorp. This technology was adapted to the representation of the nodes and edges (also known as boxes and arrows) representing an intelligence analysis process.

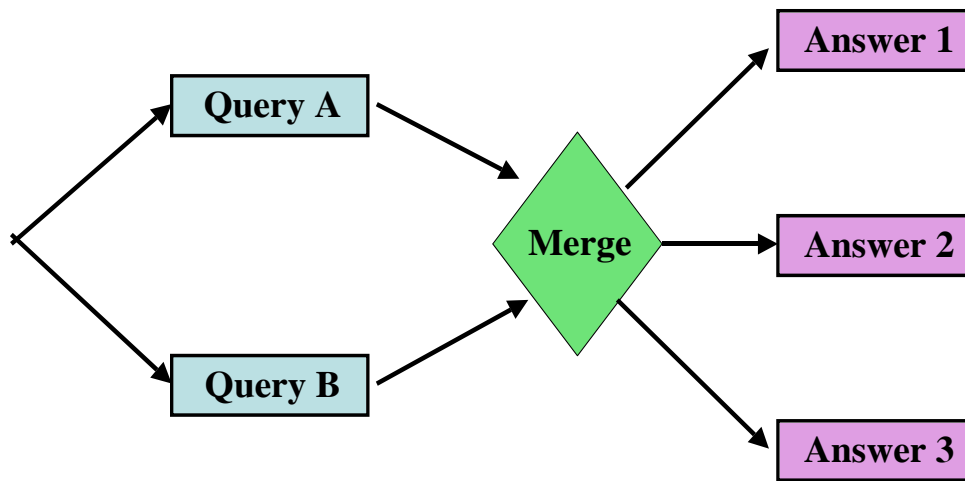
The association of nodes with their semantics was done in two parts: an English summary of the meaning is directly added by the SME; and the underlying query is prepared in the Query Library, and then simply dragged-and-dropped onto the relevant node. An icon on the node indicates whether its semantics have been fully specified. See Figures 22 and 23.

Three basic types of edge are supported: Yes, No, and Next. The “Yes” and “No” edges represent whether the query at the source node was successful in the sense of finding bindings for its open variables, or being proven true. The “Next” edge type indicates that the destination node should be examined unconditionally whenever the source node is, and without using any information from it. Strictly speaking, the “Next” edge type is redundant, as the relevant edges can simply be drawn as duplicates of the in-edges of the source node; for this reason, this edge type was not provided initially. In practice, SMEs turned out to have an irresistible urge to use edges of this type in their process diagrams, and so, according to the principle of intuitiveness, support was added in the tool.

A final aspect of diagram semantics lies in the way two nodes are related along an edge. Because the diagram serves to contextualize the queries, the query at one node has to be able to rely on the variable bindings determined by the queries for previous nodes. For example, a query “Is the high ground accessible by road” might depend in an obvious way on the results of a previous query like “Is there any high ground within 3km of a battle position?” To support this inheritance of variable bindings between queries, the same variable mapping interface used in query combination was employed by SMEs to assign edge semantics.

One minor irritation in the system used by the SMEs was that manual graph layout was not preserved between sessions. Instead, the system did its own automatic layout. This did not have a significant impact on the ability of SMEs to enter knowledge correctly, but did affect efficiency;

as SMEs found that they needed to rearrange the graph each time they opened it, to return it to a layout they found intuitive.



| Query A | Query B | Answer |
|---------|---------|--------|
| No | No | 1 |
| No | Yes | 2 |
| Yes | No | 2 |
| Yes | Yes | 3 |

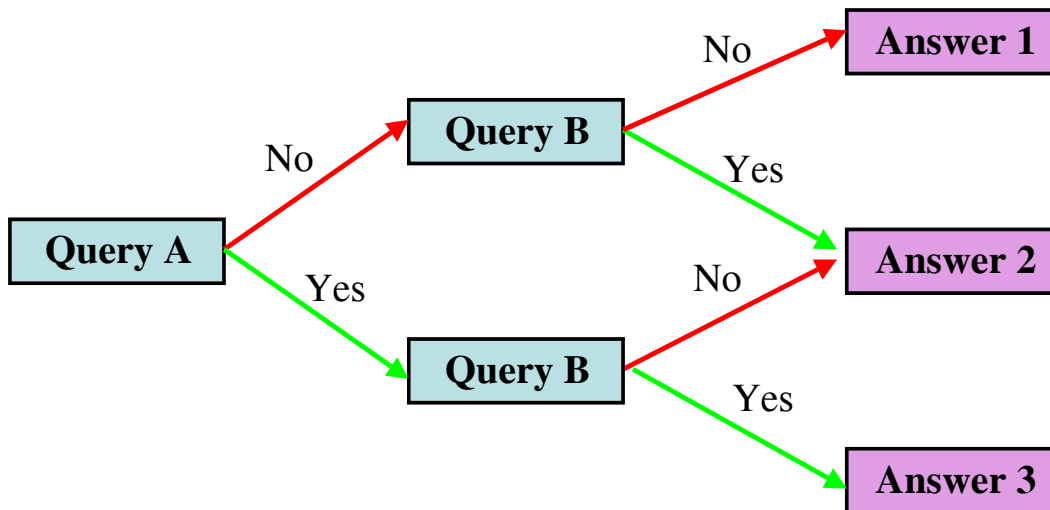


Figure 19: Simplified example of results merging. The top diagram is how the SME might have drawn it by hand initially, with the results of two queries merged into three possible conclusions in an unspecified way. The table in the middle indicates how the SME might have described (for example, using the Value Table Tool described below) how the inputs of the merge map onto its outputs. The diagram at the bottom shows the complexity that results if the same merge is represented the hard way. Note that Query B is repeated and consider how the complexity increases as the number of inputs goes up.

2.3.2.1.4.4 Result Merging

In the diagrams that SMEs drew, there were several cases where two or more query results were merged into a single conclusion. For a real example of this, see the large diamond in Figure 21, where the outputs of three yes/no queries are merged to assess the advantages to attacker and defender. A simplified example is given in Figure 19, which shows how such a decision logic could be represented using the same sorts of box and line, but at the cost of making the resulting graph complex and repetitive.

A solution to this issue was designed and met with SME approval, but resources did not permit its deployment within RKF Year Three. Such result merging can clearly be modeled using more decision nodes, but at the expense of making the diagram more complex, and of moving further from the SMEs’ intuitive representation.

| | Vegetation conceals (type/size) defending force from direct LOS from front? | Vegetation conceals (type/size) defending force from direct LOS from flanks? | Vegetation conceals (type/size) force shifting laterally within defensive area? | score: | Assessment: | SME chart | |
|----------------------|---|--|---|--------|-------------------------|-----------|---|
| True (1) / False (0) | 0 | 0 | 0 | 0.00 | POOR | POOR | Entered by user Determined by system |
| True (1) / False (0) | 0 | 0 | 1 | 0.33 | MARGINAL | MARGINAL | |
| True (1) / False (0) | 0 | 1 | 0 | 0.33 | MARGINAL | MARGINAL | |
| True (1) / False (0) | 0 | 1 | 1 | 0.67 | ADEQUATE | ADEQUATE | |
| True (1) / False (0) | 1 | 0 | 0 | 0.33 | MARGINAL | MARGINAL | |
| True (1) / False (0) | 1 | 0 | 1 | 0.67 | ADEQUATE | ADEQUATE | |
| True (1) / False (0) | 1 | 1 | 0 | 0.67 | ADEQUATE | ADEQUATE | |
| True (1) / False (0) | 1 | 1 | 1 | 1.00 | EXCELLENT | EXCELLENT | |
| Relative Importance: | 1 | 1 | 1 | 3 | Total importance weight | | |

| Assessment scale | | |
|------------------|----------------|----------|
| Output scale: | assessment is: | if over: |
| | EXCELLENT | 0.9 |
| | ADEQUATE | 0.6 |
| | MARGINAL | 0.15 |
| | POOR | 0 |

System seeds outputs table above with all possible combinations, applies the user-entered relative importance numbers and does the math to get the score for each combo, then applies the user-entered scale and threshold numbers to get the assessment.
User can directly edit table, overriding the assessment accompanying any individual combo.

Figure 20: An example of a prototype value table. In this case, the inputs are true/false, and the output is four-valued, from Poor to Excellent. The yellow boxes indicate the three weights and four threshold values that the SME would have to enter to achieve the initial assignment of output value for each possible set of inputs.

The long-term solution is to support value tables, a generalization of truth tables. A truth table represents how multiple Boolean inputs are combined to give a Boolean output (or sometimes several outputs). In a value table, the inputs and outputs are not (necessarily) Boolean, but are restricted to being discretely and finitely valued (by imposition of boundary values on continuous input values if required). Thus they can be seen as a function from qualitative inputs to a qualitative output. The SME must assign an output value to each possible set of input values.

As the number of inputs and the number of possible input values increases, the size of a complete value table explodes exponentially, increasing the time required to fill in all outputs. To combat this effect, it was proposed to allow the SME to construct a simple numerical model: this might take the form of a facility to assign real numbers to the possible values for each input (e.g. 0 for false, and 1 for true), and weights to the various inputs, and then assign output values using

thresholds on their weighted sum or product in the obvious way. This was not intended to supplant SME intuition but rather to populate the output column initially. SME reaction to this part of the proposal was mixed, and this is therefore an issue to be resolved empirically.

2.3.2.1.4.5 Script Construction

Cyc needs to be able to consider the analysis process not merely as the graph inherent in an Analysis Diagram, but also as the series of steps that some agent might execute to perform the analysis. The latter is known as a script in the Cyc ontology.

The script representation of the analysis process was generated entirely automatically from the graph representation of the Analysis Diagram. This was done using forward rules, which are executed eagerly.

2.3.2.1.4.6 Script Execution

As designed, the execution of the script was to be performed by another layer of forward rules generated automatically by specialized code. Although this code was designed and implemented, resources were insufficient to permit its testing and deployment, and these additional rules were therefore written by hand for the evaluation (in a way that was, of necessity, specific to the evaluation use-case).

2.3.2.1.4.7 GIS Integration

SAIC provided an XML-based socket interface to ESRI's ArcMap GIS product. SMEs used ArcView to examine and annotate the Fort Hood terrain related to the scenario (see Section 3.3). The resources available for RKF Year Three did not permit a very extensive GIS integration, and there were outstanding issues of coverage, performance and reliability. The evaluation results obtained were based on information obtained from GIS, with a Cyc-side cache, and a certain amount of synthetic data. Notwithstanding these setbacks, the integration was able to demonstrate that Cyc could use the quantitative data available in a GIS system as part of a qualitative analysis of terrain, thereby bringing closer together the art and the science of military theory.

2.3.2.1.4.8 Collaboration Support

An ideal version of this system would permit SMEs to split and merge process diagrams, and to share constructed queries in real time. While this was not achieved in the scope of this project, the knowledge base representation of graphs and queries readily supports this sort of extension. In practice, SMEs could see each other's work from the previous day (see Section 3.3.1.2), and the final merge was performed by an ontologist (see Appendix A).

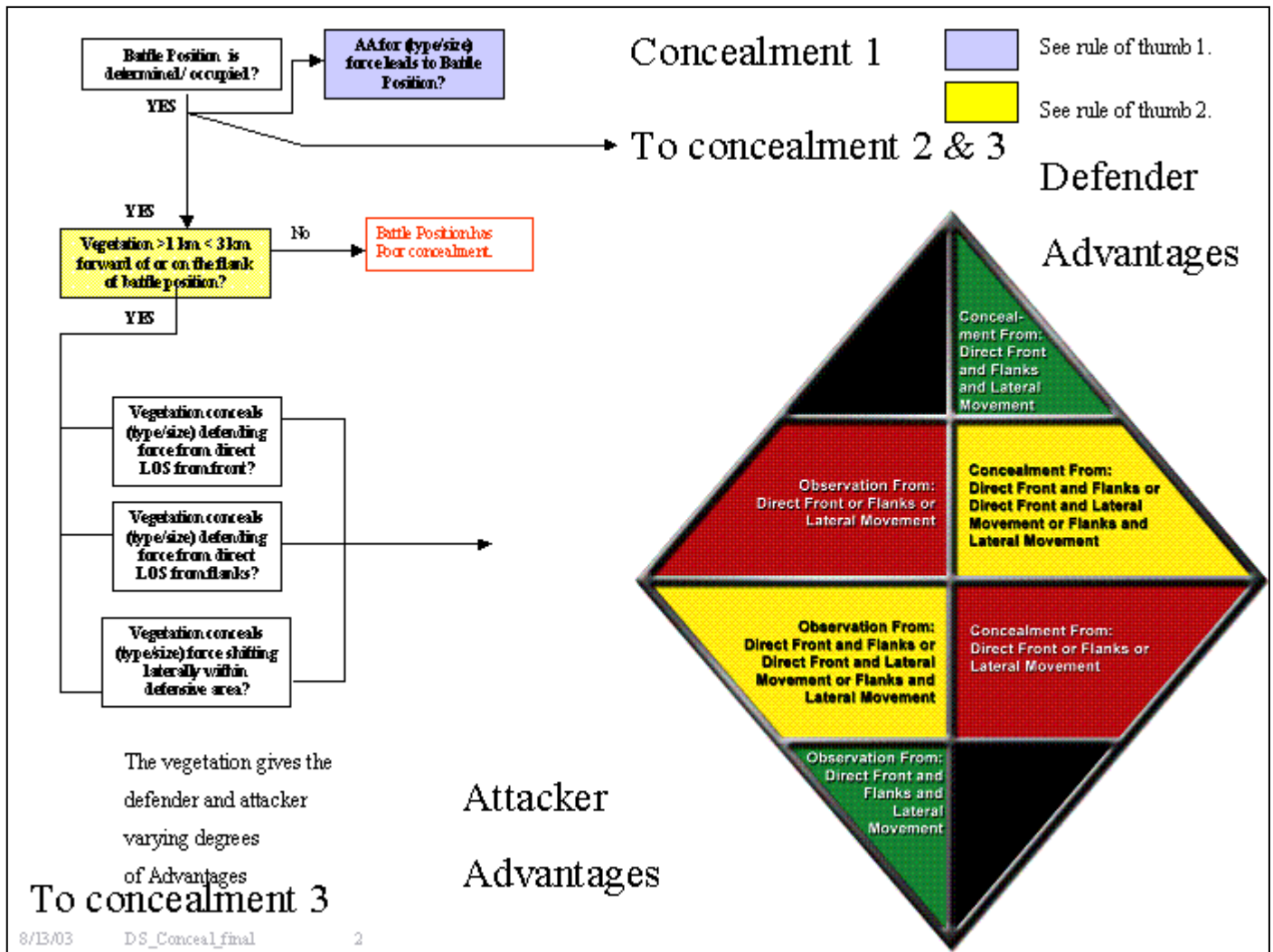


Figure 21: Extract from a flowchart prepared by an RKF Year Three SME. Originally intended for inter-human communication, this format inspired a new interface: the Analysis Diagram Tool. Each box contains either a query or a conclusion and that boxes are linked with arrows labeled “Yes” and “No”. Boxes make implicit reference to the answers to previous queries, e.g. “vegetation”, “defensive area”. The yellow and blue rectangles indicate hierarchical decision-making, as they refer to sub-processes. The three questions at the left are to be answered in parallel to reach a pair of qualitative conclusions, namely the advantages conferred on the attack and defender. The possible results are indicated in the diamond at the bottom right, although the SME has not indicated how the eight possible answers map onto the four rows of the diamond; this is the gap filled by Value Tables (see Section 2.3.2.1.4.4)

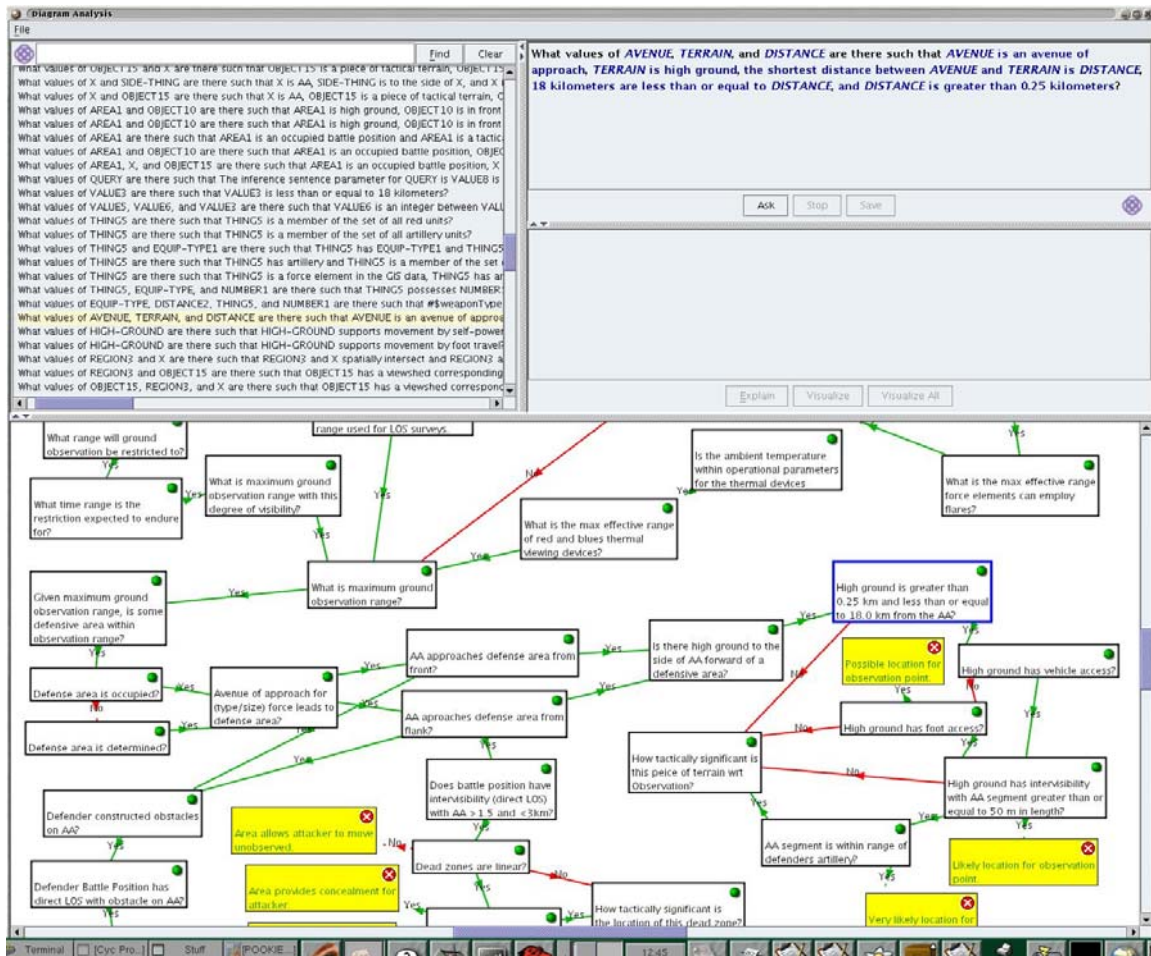


Figure 22: The Analysis Diagram Tool. An analysis process is being graphically edited in the bottom panel. The top three panels are from the Query Library, with the set of queries on the left, the focal (editable) query at the top right, and the answers (currently blank) on the middle right. The user can construct queries, and then drag them onto nodes in the graph.

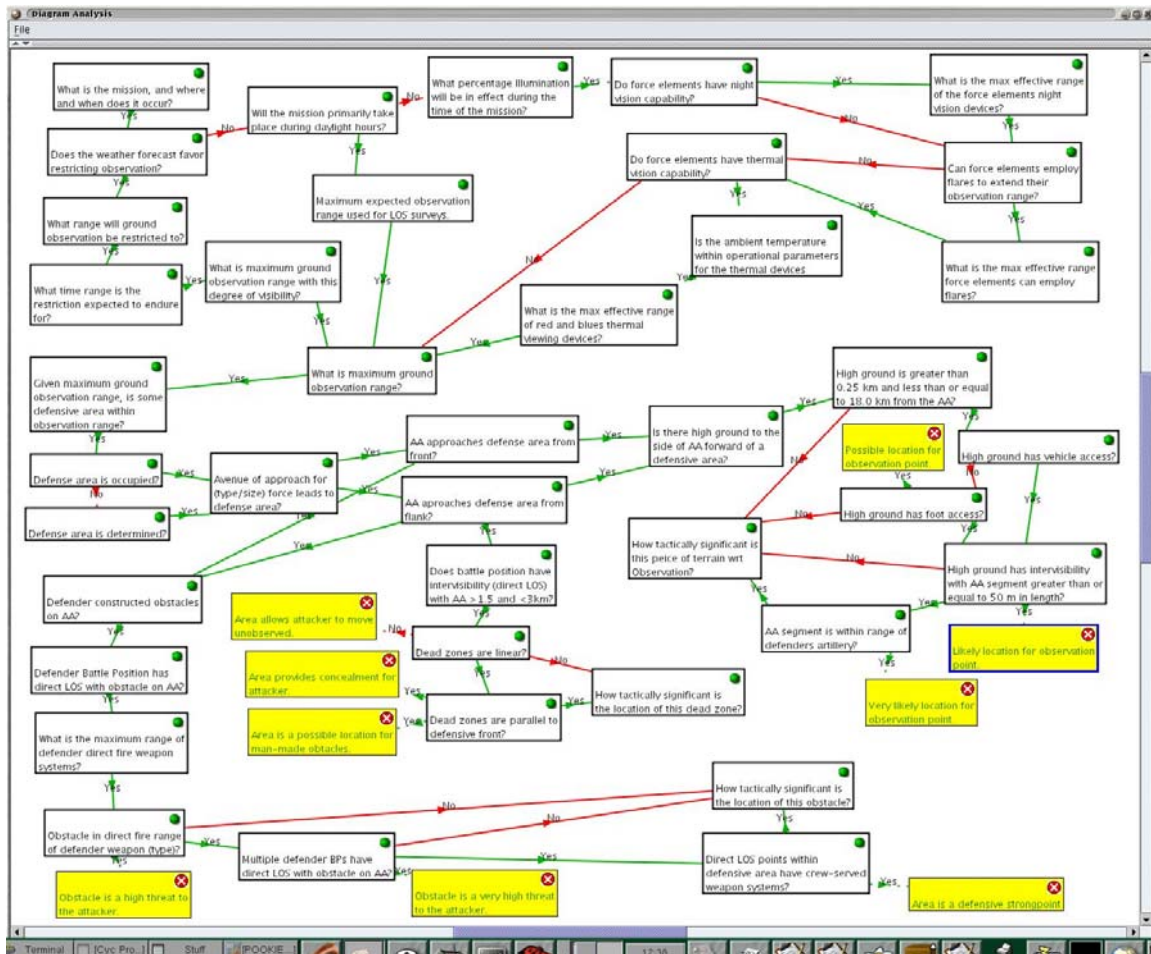


Figure 23: Detailed view of terrain analysis process with respect to observation as viewed in the Analysis Diagram Tool. White nodes represent queries, and yellow nodes are conclusions. Green and red edges represent the "Yes" or "No" links in the flowchart.

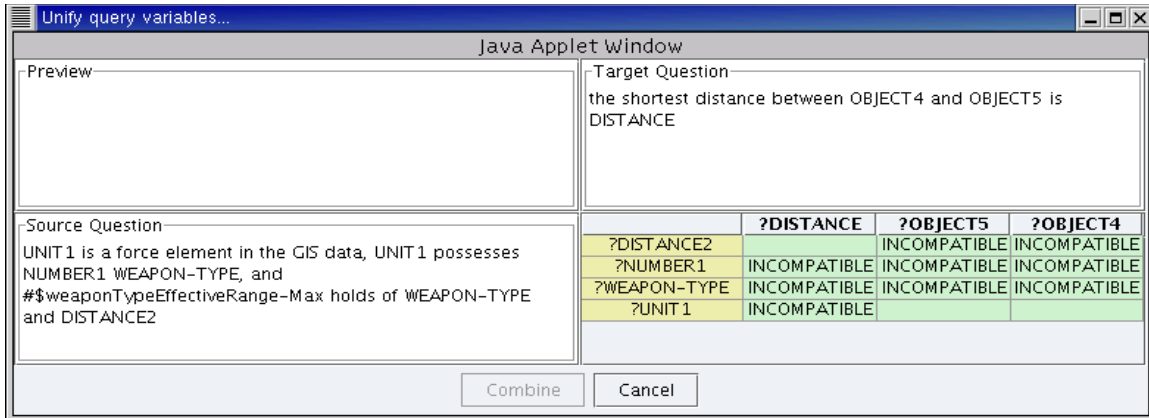


Figure 24: Variable mapping dialog from the Query Library (or the Analysis Diagram Tool). The user is building up a complex query from multiple fragments. The current query, formed from three fragments is shown on the bottom-left. The fragment to be added is shown at the top-right. The panel at the bottom-right shows a matrix with the variables from the two queries as rows and columns respectively. The user can now select one or more variables to pair between the queries. Cyc’s semantic analysis has already eliminated 9 of the 12 possible mappings. If the user chose to map ?UNIT1 to ?OBJECT4, then the cell representing the potential mapping between ?UNIT1 and ?OBJECT 5 becomes unavailable. The panel at the top-left is provided for a preview of the combined query.

This dialog may seem complex, but it is actually quite an effective way to present the possible variable mappings to the user. Before this interface modality was invented, it was necessary to present the user with a long list of subtly different possible combined queries, which was almost impossible to use correctly.

This dialog does not appear at all in cases where Cyc can deduce the only appropriate mapping between queries for itself, and this is likely to happen in more and more cases. The limiting factor is how tightly the possible arguments to the predicates involved are constrained; in general, the possibility of exceptional circumstances prevents tight argument constraints, but a notion of typical argument constraints will assist in eliminating unlikely possibilities. For example, the object of “eat” is usually a foodstuff, but the sentence “My child eats dirt.” is still meaningful.

2.3.3 Lessons Learned

In many ways, the system at the end of RKF Year Three is a critical success toward the goal of rapid knowledge formation. The rate of knowledge acquisition using the Factivore was the highest for a non-task-specific tool yet recorded in RKF. The authoring of rules for evaluations using analysis diagrams was very much faster and a lot more intuitive than the Rule Constructor of Year Two could ever have been. Because of limited funding in RKF Year Three, the GIS integration was very basic, but significant steps toward using GIS facilities in symbolic reasoning were made nevertheless.

The fact that the best work was achieved by using the newly developed Java components suggests that responsive user components written as dedicated applets is the approach to take for future interfaces. The only problem remaining is that the infrastructure for these interfaces is still too specific, *i.e.* not enough of the user interface description comes from the Cyc knowledge base, which would make it amenable to being reasoned about. Thus, any future knowledge formation effort will want to focus on making the user interface description even richer, so that the task-specific focusing can be achieved on the fly.

The analysis diagram technology is only at its beginning, but many of the rules that SMEs can be expected to author have the same structure as situation evaluation rules. Cycorp has already received a SBIR grant to continue work in this area after RKF, thereby pushing the rule-writing envelope even further.

There were points at which the arrangement of nodes created by the SME were quite different from what an ontologist might create, either in the number of distinct query nodes or in their ordering and dependencies. There were also differences in the way individual SMEs chose to arrange and break up queries, even when working from the same reference document. These differences gave rise to several additional benefits, some points of possible difficulty, and a number of areas in which system reasoning could be called upon to improve diagram structure, and perhaps even analysis procedures themselves.

It is an advantage of this approach that many such arrangements are possible and will work. A SME might layout

 “Does the weather forecast favor restricting observation?”

 ⇒ “What time range is the restriction expected to endure for?”

 ⇒ “What range will ground observation be restricted to?”

while an ontologist might be more likely to condense this step into a single node, asking: "The weather forecast favors restriction of observation to VISIBILITY for TIME-PERIOD."

Similarly, a SME might be more likely to go through the complete processes for assessing the impact of night vision equipment, thermal devices, and flares sequentially, while an ontologist might be more likely to have each of these branches in parallel from a point at which the shared required background has been established and any could be relevant. In each case, though, either approach will work, as will many others. This allows room for idiosyncrasy of expression, without compromise to the logical implications that result. The ability of the SME to specify the analysis procedures is not particularly brittle to the individualities of that person's thinking.

On the other hand, the flexibility here could lead to two possible downsides. First, it may be hard for a SME to tell whether a procedure represented by another SME is wrong or just harmlessly divergent from their own. Second, analysis efficiency may be sometimes lessened by the organization of a particular SME's representation.

Both considerations suggest that it would be beneficial for the system to reason about the logical equivalence of divergent diagrams, and about changes that could be made to any given diagrammed procedure to result in a logically equivalent but more efficient process, and to suggest such changes to the user.

Additional benefits of Analysis Diagrams and the flexibility of the representation:

- It would be quite simple to represent the analysis methods of a variety of SMEs (current or historical) in a single KB, thereby allowing a user to get virtual feedback regarding how other experts might analyze the same piece of terrain.
- It would also be feasible to add sub-processes for some queries such that if they fail to yield answers, the system checks what information is available for answering that query, and if that information falls below some criteria for completeness and freshness, assert the presence of an intelligence gap. Many options unfold from here:
 - The intelligence gap can be added to a list of intelligence needs for possible deployment of reconnaissance assets.
 - This noted gap could be annotated with some representation of significance, automatically generated by reasoning about the graph paths, and eventual conclusions that would be possible if this piece of information were known.
 - Default assumptions can be specified by a SME, representing what the SME would do, reasoning-wise about an intelligence gap of this type in this context.
 - The query can be treated as though it answered yes (e.g., if you don't know, assume there is an enemy presence on the high ground) or treated as though it answered no (e.g., if you don't know about any aerial surveillance equipment in enemy possession, assume they don't have any).
 - The intelligence gap can be treated as a show-stopper (e.g., if we don't know about any elevation data, stop trying to do the automated analysis).
 - An "unknown" arc type could be added, and distinct reasoning paths could be diagrammed for cases where a query is unanswerable given the known information and, in the SME's opinion, it would be unwise to make any assumptions.

3 Evaluations

3.1 Year 1: Molecular Biology

The Year One evaluation has already been reported on extensively by both Cycorp and IET, the external evaluator, and its results are not reproduced here.

3.2 Year 2: Sketching-assisted Authoring of COA-Critiquing Rules

Although the Year Two evaluation has also been reported on in the past, it is instructive to reproduce some of the results here.

Two military SMEs were given the task of entering as much as possible of two Courses of Action (COAs) for two different background scenarios within a week. The evaluation COAs had many more tasks than any of the COAs that IET (the external evaluator) had provided previously for testing. Consequently, the evaluation gave the KRAKEN system a stronger stress test than it had ever received during the development process.

During the evaluation, the military SME team produced 594 facts for the first COA, and 745 facts for the second. Some of these facts were produced by forward inference; that is, KRAKEN was able to deduce additional information automatically from the knowledge entered by the SMEs. Discounting this deduced knowledge gives 251 and 238 facts for the two COAs. Dividing by the amount of time spent on each COA, this gives assertion rates of about 10.87 facts/hour. The target assertion rate for RKF Year Two was 10 facts/hour.

For RKF Year Two, the KRAKEN team had selected rule authoring as a focus of Knowledge Entry. Rule authoring is considered a very hard problem, often beyond the capabilities of lower-level knowledge engineers. It was a very encouraging result that the SMEs were able to write rules at all.

During RKF Year One, four SMEs had authored only a total of three rules of mediocre quality over the course of several weeks. In Year Two, in addition to describing the COAs, the pair of SMEs was able to write 10 rules in the five-day period. These rules were more complicated than those of Year One. They also chained together in an interesting fashion, suggesting that the rules had been abstracted and factored correctly. Partially, this success was made possible by the stronger context for rule authoring: both the COA evaluation matrix and the COA Completeness queries that IET has designed required the SMEs to write rules. See Figures 16 and 25 for examples.

3.3 Year 3: GIS-backed Advanced Authoring of Terrain Analysis Process

Cycorp has been using four military SMEs (Subject Matter Experts) in the course of RKF Year 3:

- Dr. Kerry Hines, Lieutenant Colonel, U. S. Army (Retired), Air Defense Artillery
- Dr. Paul Girard, Commander, USN (Retired), Surface Warfare Officer, Engineering Duty Officer (R&D)
- George “Dutch” Sley, Lieutenant Colonel, USMC (Retired), Armor Officer
- William J. Mathews, former Sergeant U.S. Army, Ranger

These SMEs provided a breadth of military experience and were able, as a team, to distill the essence of the terrain analysis process.

3.3.1 Evaluation Methodology

3.3.1.1 Preparation

In advance of the Year Three Evaluation, the four military SMEs spent considerable time documenting the OCOKA analysis processes they wanted Cyc to understand. Drafts of the Observation, Fields of Fire, Cover, Concealment, Obstacles, and Key Terrain charts were created by the SMEs in PowerPoint, and then discussed, both among SMEs and with Cycorp ontologists. This process had several goals. The most straightforward goal was to describe the analysis process in a way that the SMEs could agree on, thereby glossing over many idiosyncrasies of approach. A larger goal, however, was to elicit and expose the many layers of knowledge

Meaning: **[+] For every THE-COA, if THE-COA is a COA, some red unit is a candidate for engagement in THE-COA, and the red unit is a reserve unit in the COA THE-COA, then THE-COA is advantageous with respect to the enemy reserve force engagement.**

Cycl: **(implies (and (isa ?THE-COA CourseOfAction) (engagedUnitInCOA ?THE-COA ?THE-DEPLOYABLE-MILITARY-UNIT) (reserveUnitInCOA ?THE-COA ?THE-DEPLOYABLE-MILITARY-UNIT) (isa ?THE-DEPLOYABLE-MILITARY-UNIT ModernMilitaryUnit-Deployable) (isa ?THE-DEPLOYABLE-MILITARY-UNIT RedUnit)) (evaluatesDimensionOfCOA ?THE-COA engagementOfEnemyReserveInCOA EffectiveForPurpose))**

Figure 25: An example of a rule authored by a SME during the RKF Year Two evaluation.

implicit in the diagram. That is, it was necessary to understand the nature of the knowledge that allows a SME to comprehend such a diagram and execute the process described. This enabled the Cycorp team to get a remarkably good picture of the knowledge encoded by the diagram, in addition to determining what information the SMEs had been unable to add (such as dependencies and assumptions), and the background knowledge on which the diagrams depended.

From analysis of this picture, two things emerged: A design for an Analysis Diagram Tool (ADT) that would allow the SMEs to enter their tactical terrain analysis knowledge, and a list of concepts and relationships that Cyc would need to understand and have available in order for the diagram-level entry to be possible. While the SMEs used the Factivore and UIA to enter the latter knowledge, including many tactical and natural terrain concepts, with assistance from ontologists, the RKF team built a prototype of the ADT.

The SME Team had developed an evaluation scenario at the start of Year Three, entitled “The Attack on Royalty Ridge”. This scenario is set near Ft. Hood, and this provided the terrain data and subject of analysis. Red (defending) and Blue (attacking) forces, weather, date and time, and general missions were also specified as part of that scenario. This scenario data was available for the evaluation in a number of forms. The terrain data was made available digitally by the DARPA Library, and was loaded into ESRI ArcView. One of the SMEs, Bill Mathews, entered the tactical terrain objects, such as force element locations and objectives, into an accompanying ArcView data set. Additionally, because of the limited range of information and GIS functionality that Cyc was able to access via the SAIC-provided XML interface, other GIS objects were created, such as significant natural and artificial terrain features, and avenues of approach, that one might expect to be obtainable by regular GIS queries and computations. Other scenario data such as the forces, their organization and equipment, basic mission descriptions, and weather, were entered into Cyc by SMEs via the Factivore or, if SME time was not available, by Cycorp ontologists.

3.3.1.2 The Evaluation

The goal of the Year Three SME evaluation was to have three of the SMEs enter into Cyc one OCOKA analysis diagram, using the ADT, and to have this chart enable actual analysis of the corresponding tactical aspects of our scenario terrain. The Observation chart was selected for the evaluation because it is the first in the normal order of analysis, and thus least likely to have hidden dependencies on other charts. This chart was not used as a use-case during tool development.⁹

During the Evaluation, each SME logged into a Cyc image, started up the Analysis Diagram Tool as a freestanding application, and established a connection between the two. The SME then worked directly at creating a diagram that represented some portion of terrain analysis knowledge. By creating nodes and associating queries with those nodes, the SMEs were able to tell the system: given what you know up to this point (what has brought you to this node), this piece of knowledge matters. By associating the node with another, for example by drawing a “yes” arc between them, the SMEs could then tell the system: if you find any cases that meet these criteria, or if the answer to this yes/no questions is yes, then this next piece of knowledge matters. By linking some parts (strictly, variables) of one query to pieces of a previous query, the

⁹ All development use-cases and internal testing examples were drawn from the Concealment chart. This intentional avoidance of use-cases and test examples from the evaluation chart was intended to help avoid over-fitting the design to the specifics of any particular kind of analysis.

SMEs could tell the system: what matters here is the answer for this query when applied to these objects.

For example, a SME creates a node, and labels it in English as:

“High ground is greater than 0.25 and less than or equal to 18.0 km from the AA?”

The SME would then create a more precise (that is, with underlying CycL) version of the query to be associated with this node. In most cases, the queries were created by combining and/or modifying basic queries that were already available in the Query Library. This example is an actual one from the SME-authored Observation diagram, but the SME was free to create the query and node in any order, at any time. Relevant queries in the library at the beginning of the evaluation included

“OBJECT1 is what distance from OBJECT2?”

“Is NUMBER1 greater than or equal to NUMBER2?”

“What vegetation features are known to this GIS?”

In each case, variables can be merged, and specific concepts such as the type “vegetation features” can be replaced with others such as “terrain high ground” or “avenues of approach.” The Query that resulted, in this case, is presented to the user as:

“What values of AVENUE, TERRAIN, and DISTANCE are there such that AVENUE is an avenue of approach, TERRAIN is high ground, the shortest distance between AVENUE and TERRAIN is DISTANCE, 18 kilometers are less than or equal to DISTANCE, and DISTANCE is greater than 0.25 kilometers?”

The resulting precise forms of the queries, when paraphrased by Cyc, come across somewhat clumsily, but were intelligible by the SMEs.

The SME then creates another node, gives it the English label “High ground has vehicle access?” and creates and associates a query of the form “GROUND is trafficable by TransportationDevice-Vehicle?”. The SME then draws an edge linking the first node to this one, and selects “Yes” from the menu of possible edge semantics. This tells this system that the second query only matters if the first query is answered in the affirmative. The SME then selects that link and is presented with a menu of actions to perform on it. The SME selects “Edit variable mapping.” which causes a table to pop up, offering semantically possible ways to connect pieces of the two queries. The SME finds the box joining TERRAIN in the first query and GROUND in the second, and chooses “connect.” This tells the system that the second query should be interpreted more narrowly than the formula itself might suggest, in that it should be applied only to whatever pieces of high ground are found as the result of the first query. In other words, the second node and link tell the system: “If there is high ground, within a certain distance of an avenue of approach, it’s important to know whether it has vehicle access.” This process of mapping variables between queries whose nodes are linked by an edge is very similar to the process of combining simple queries into a more complex one.

The SMEs worked on such node, query, and link creation for an average of 4-5 hours at a time. Two SMEs used the system remotely; one was working on-site. There were 6 SME sessions held over the two week Evaluation period, with time taken in between for system improvements, and a

few hours of SME time taken at the beginning for training in using the ADT¹⁰. Not all SMEs participated in all 6 sessions. At the end of the Evaluation, most of the Observation chart had been completed. A small number of nodes had to be added by a Cycorp ontologist, particularly in order to link the output of different SMEs, or to join results from multiple queries into a single conclusion. A tabular interface in mind for this last function had been designed, and discussed with the SMEs, but there was insufficient time to implement it before evaluation.

3.3.1.3 Collaboration

Several relatively independent chunks of the Observation analysis diagram were identified, and the SMEs were each asked to work on a distinct chunk from the outset. The version of the Observation diagram that the SMEs had developed earlier in PowerPoint was redistributed, and the SMEs were instructed to take this as their starting point.

In earlier RKF evaluations, all SME entered knowledge was reviewed carefully and potentially hand-edited before being loaded into Cyc's knowledge base. For this evaluation, however, the quality of output of the tools used meant that the SMEs' work product was suitable to go directly into the knowledge base. As they drew their diagrams, creating queries, linking queries together, corresponding assertions were made in the KB. Each SME used a separate workspace diagram, and the assertions from it went into a distinct workspace microtheory. Because of the evaluation conditions, some of the SMEs worked on separate Cyc images, and hence could not see each other's work while it was in progress. At the end of each session, however, their assertions were transmitted to the main master transcript and were included in the next Cyc build. For the next session, therefore, each could open and view the charts created last time by the other SMEs. At least one of the SMEs routinely did this, to see how much alike or unlike his work was from the others'. At one point he noticed that another SME had begun working on an overlapping area to his own. It would have been possible to enable more real time collaboration, but no strong need was identified.

Additionally, queries created by SMEs went into their personal Query Library folders, but those queries that they determined particularly useful and solid could be moved into a shared folder visible to all of them. Lack of time prevented the development of a mechanism for doing this automatically, so SMEs instructed Cycorp ontologists when such queries should be moved. Similarly, as SMEs completed a segment of the chart to their satisfaction, an ontologist migrated it into the main Observation chart and microtheory, and then hooked it up with work from the other SMEs. The SME could then move on to another section of the chart. In practice, this merging process only took place in the latter part of the two-week evaluation period, so the SMEs were not able to supply their responses to the merged work.

3.3.1.4 Metrics

Three of the SMEs took part in the evaluation. During the evaluation, they used the Analysis Diagram tool to represent graphically the terrain analysis process over a two-week period. The results of this process were analyzed to produce the table below.

| | Kerry Hines | Dutch Sley | Bill Matthews |
|------------------------|--------------------|-------------------|----------------------|
| Assertion count | 522 | 760 | 969 |
| Assertion rate | 52/hour | 40/hour | 51/hour |

¹⁰ For comparison, SME tool training in RKF Year Two was performed in several long hands-on sessions prior to the evaluation period.

| | | | |
|-------------------|-----|-----|-----|
| Node count | 5 | 26 | 23 |
| Edge count | 9 | 43 | 20 |
| Term count | 156 | 208 | 332 |

Assertion counts are for extant assertions, and therefore exclude any work that was deleted or modified. This yields a more reliable measure of KE productivity. These counts do not include the many assertions that Cyc is able to derive from the SME assertions.

Each SME worked different hours during the evaluation, and spent some of the time examining source material or conferring. To provide an estimate of the assertion rate, the assertion count was divided by the number of calendar-aligned hour-long periods during which at least one assertion was made. This measure appears to correspond well with estimates of SME working time.

Nodes and edges are the queries and links between them that appear in the Analysis Diagram or flowchart. Terms are the CycL constants, whether individuals, collections, or relations.

In the period of RKF Year Three before the evaluation, SMEs used other KRAKEN tools, mainly the UIA and the Factivore to flesh out basic military domain information and the details of the evaluation scenario. Using the same measure, the SMEs had a knowledge entry rate of 30 – 33 assertions/hour.

For comparison purposes, the same analysis was applied to three team ontologists working over a similar period with a range of interface modalities from the SME tools down to raw CycL. They achieved knowledge entry rates of 13 – 37 assertions/hour.

Thus, although the ontologists' output may have been more complex and subtle in nature, it can clearly be seen that the SME-oriented KE tools are achieving impressive productivity rates that compare extremely favorably with traditional manual KE.

3.3.1.5 Legacy Terrain Knowledge Base

The driver and use-case for RKF Year Three was the development of a terrain knowledge base of sufficient quality that it could be used by future projects. The terrain knowledge was to be SME-entered and/or vetted, and standard with respect to military and geographical usage. The knowledge was to be also sufficiently deep to be used in military terrain analysis.

Much of this goal was accomplished, though the knowledge is not complete. Cyc's knowledge base now contains SME-entered, or SME-vetted (particularly in the case of high-level infrastructural concepts not yet amenable to RKF tool entry) knowledge in the following areas:

- Natural terrain features: Military Major, Minor, and Supplemental.
- Artifactual terrain features (e.g., bridges, roads, buildings, power lines).
- Definitional information on all characteristics such as shape, dimensionality, and slope types, as relevant (e.g., a saddle can be viewed tactically as four-sided, with the ground sloping upward away from it on two sides and downward away from it on two sides, and with these slope types alternating).
- Tactical effects of terrain types, for units of certain types on or near the terrain feature (e.g., a hill generally facilitates observation for a ground unit on it, and generally obscures observation for a ground unit near it).

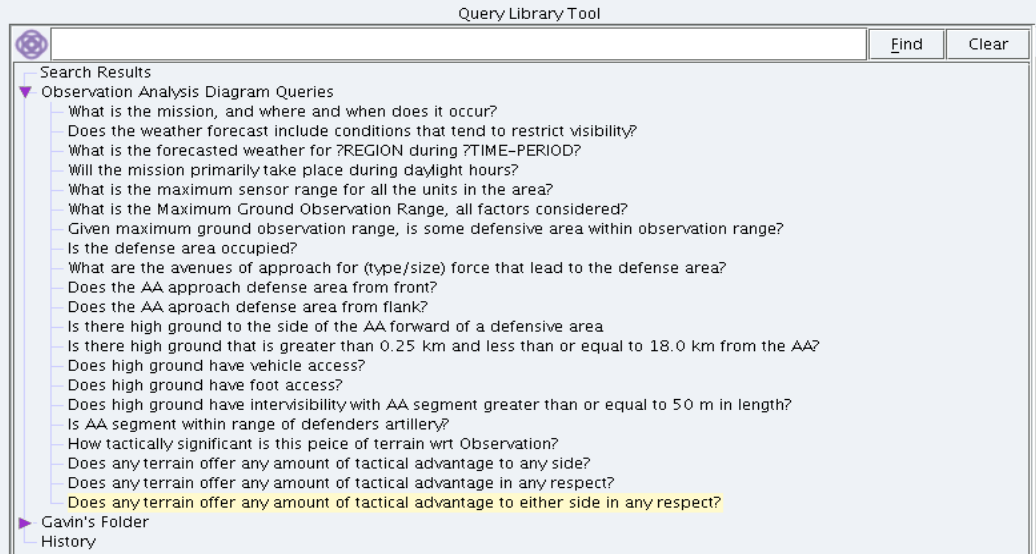


Figure 26: Query Library showing queries that ask Cyc about a scenario. The highlighted query asks Cyc to evaluate the terrain.

- Knowledge of the general relevance, if any, of the terrain type to observation. Infrastructure is there for entering similar knowledge for other OCOKA dimensions, but they have not been covered.
- Faceting of terrain by feature type (relief, vegetation, construction).
- Faceting of terrain by tactical type (cover providing, concealment providing, etc.).
- Weather: representation of forecast data and infrastructure for specifying tactical effects with data entered for a few cases. Reasoning to determine effect on observation.
- Interaction with equipment: infrastructure with facts for some specific cases
- Prototype representation of analysis procedure for evaluating the tactical characteristics of a piece of terrain with respect to observation.

Further discussion of the nature and purpose of the Legacy Terrain KB can be found in Appendix D.

3.3.1.6 Conclusions and Insights

The key insight of RKF Year Three was that it was possible for the SMEs to communicate an intelligence analysis process to KRAKEN in a form that better corresponded to their usual mode of work. Historically, Knowledge Entry projects have started by attempting to teach the SMEs to use some knowledge-engineer-oriented logical representation language, and to do this it has always been necessary to devote a lot of time to ‘fixing’ the SMEs’ irrational thought processes.

A better approach, and that attempted by Cycorp, is to find ways to move the domain of discourse away from the formal AI representation, and towards something more natural for humans. It was observed that the military SMEs were exchanging their thoughts on the intelligence analysis process in the form of flowcharts (some would call them ‘decision trees’). The Analysis Diagram Tool allows such diagrams to be entered directly into Cyc.

The Analysis Diagram Tool is not complete, but it has already shown impressive results.

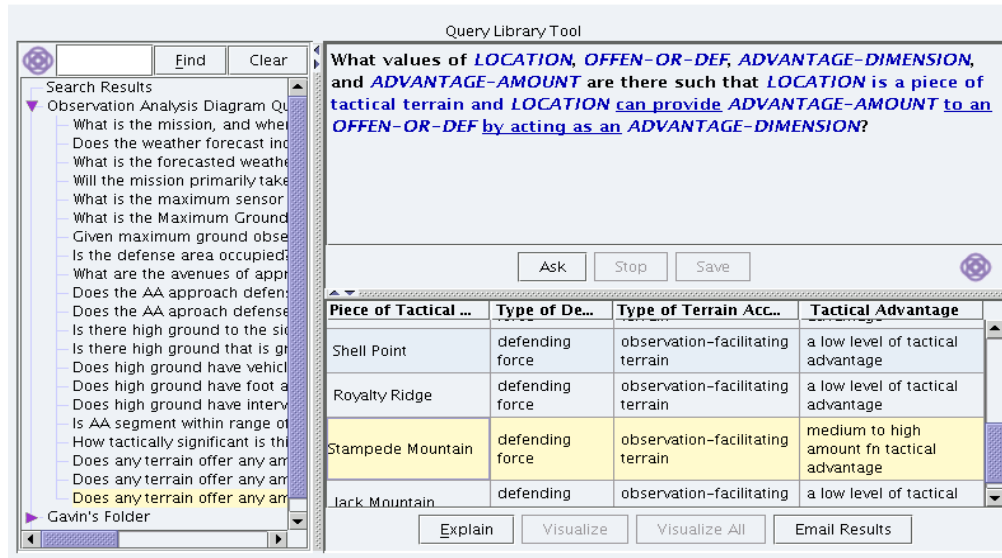


Figure 27: The Query Library shows a selection of queries at the left. The highlighted query (about the tactical advantage offered by terrain) is also displayed top-right, in a far more explicit way. Answers are given bottom-right. Note that most hills offer a defender only a low advantage for observation unless they meet certain criteria, as Stampede Mountain does.

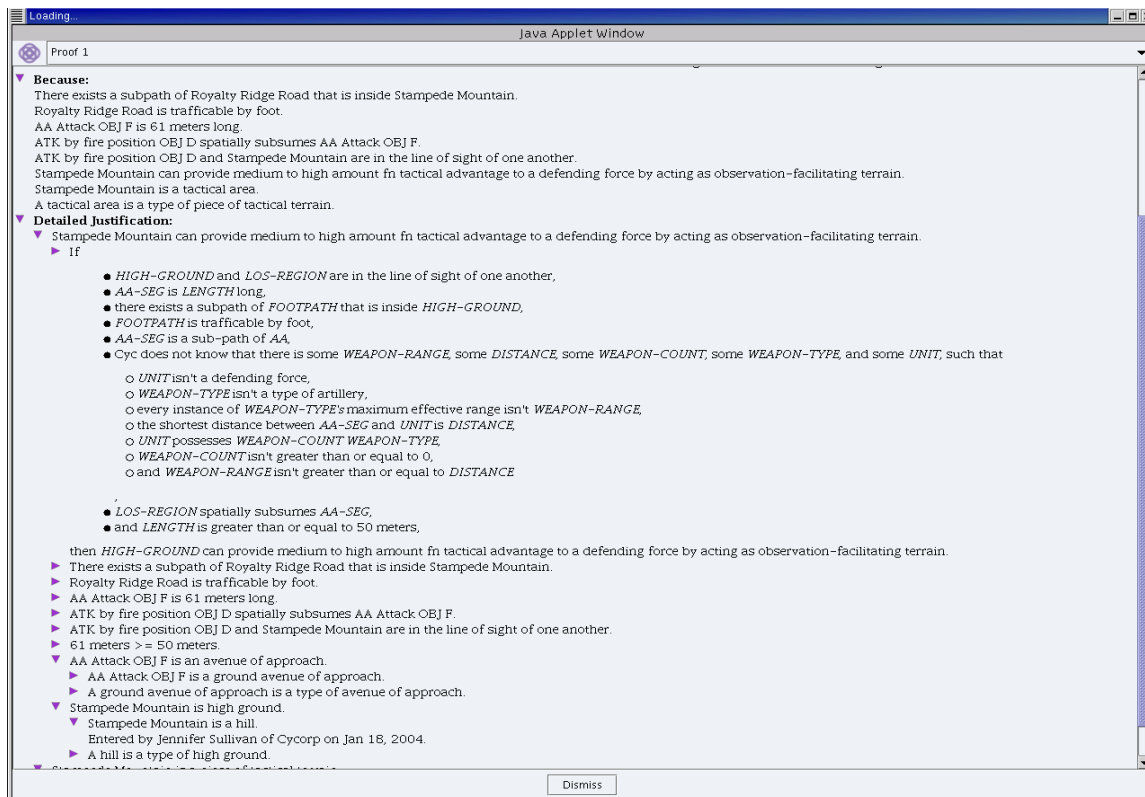


Figure 28: The Query Library gives full explanations for its results, an extract from which is shown here. This explains why Stampede Mountain offers better than normal tactical advantage with respect to observation.

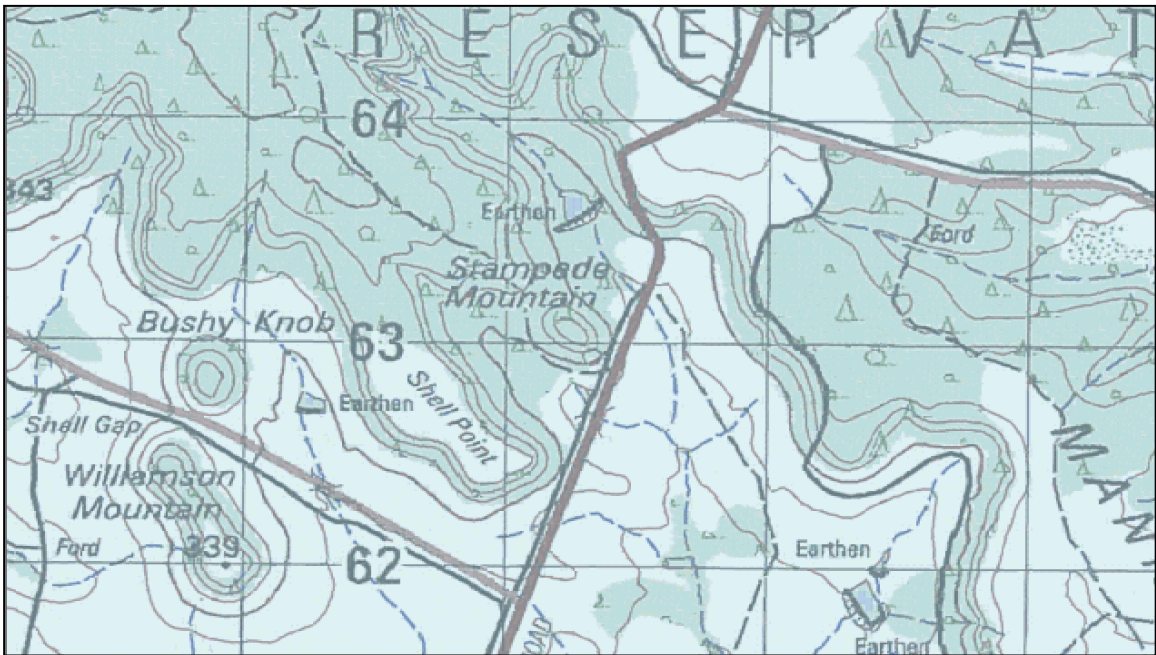


Figure 29: Map sheet (from GIS) showing Stampede Mountain. As Cyc notes, it is foot-accessible high ground within easy visibility of a north-south road and tank trail identified as an avenue of approach.



Figure 30: Close up of Stampede Mountain map sheet with aerial photograph overlay.

4 Discussion

4.1 How the RKF Project has affected Cycorp

Participation in the RKF project has had a profound effect on Cycorp. It has been instrumental in revolutionizing the interfaces used for knowledge entry and knowledge retrieval. In addition to the specific project synergies outlined above, and the specific tools developed by and in concert with the RKF team, RKF technology has always been in demand within the company, and there are a number of ways in which it has been migrated into the regular Cyc interface. Seven examples stand out: Precision Suggestion, WFF-Repair, Concept Refinement Interview, Lexical Matching, NL Generation, KB Graphing, and the Dictionary Assistant.

4.1.1 Precision Suggestion

In the UIA, every time the user asserts a new fact, the system offers suggestions for stronger versions of the statement that might also be true. This has been made available within the regular Cyc browser, and can be used by ontologists to ensure that facts are expressed as productively as possible.

4.1.2 WFF-Repair

Whenever knowledge is added to Cyc, each sentence is checked to see if it is a semantically Well-Formed Formula (WFF) and will be rejected if not. The UIA, to enhance the user experience, attempts to repair any rejected formulas, adding supporting facts as necessary. This capability has been added to the regular Cyc browser, and enables ontologists to enter facts the first time, without having to go back and manually add the prerequisites.

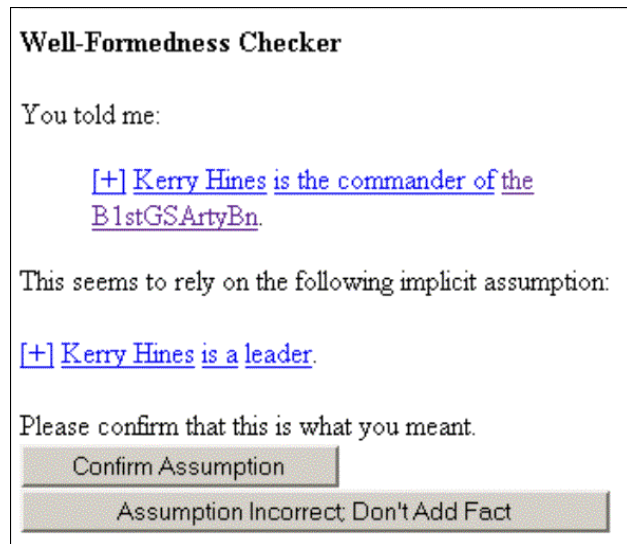


Figure 31: WFF Repair suggests a new sentence that, if asserted, would allow the original assertion

4.1.3 Concept Refinement Interview

A long-standing feature of KRAKEN's UIA has been the ability of the system to induce its own questions about focal concepts. This is now available in the Cyc Browser, in concert with the Factive, which was also made available as an applet (as opposed to the original free-standing application) to support RKF.

4.1.4 Lexical Matching

A widespread problem encountered in extending large knowledge bases is the fact that it is hard to find an existing concept, and this impedes reuse. The Cyc Browser now permits concepts to be found using not only their CycL constant names, but also by matching their lexical

representations. This enables concepts to be found using synonym expressions, and should enable Cyc use in languages other than English, although this has not been extensively tested as yet.

4.1.5 KB Graphing

One of the earliest tools developed for RKF during year II and deployed in year III was the Blue Grapher (see Figure 33), which displays nodes and edges graphically. In particular, the nodes are concepts (Cyc terms), and the edges are the relationships between them (Cyc assertions, labeled with the predicate). This KB visualizer is now available in the Cyc Browser, and is used by ontologists to visualize parts of Cyc's ontology in a highly configurable way.

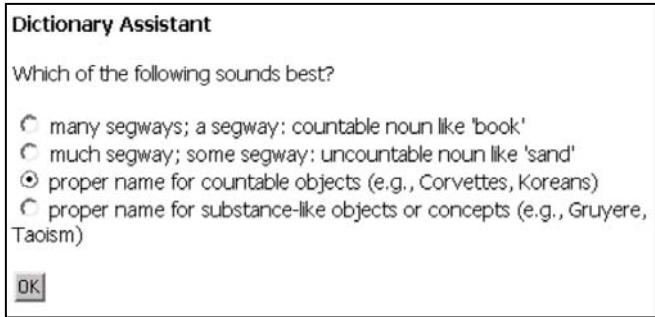


Figure 32: The Dictionary Assistant guides the user to make lexical mappings.

4.1.6 Dictionary Assistant

RKF's NL work was initially somewhat slow because it was difficult for anyone except a skilled linguist to represent the lexical mappings required for natural-language parsing and generation. To solve this problem, substantial effort was put into tools that allowed SMEs to extend Cyc's lexical mappings for new terms created (see Figure 32). Early version of these tools were first developed for use by ontologists, and the improved versions are now available to ontologists

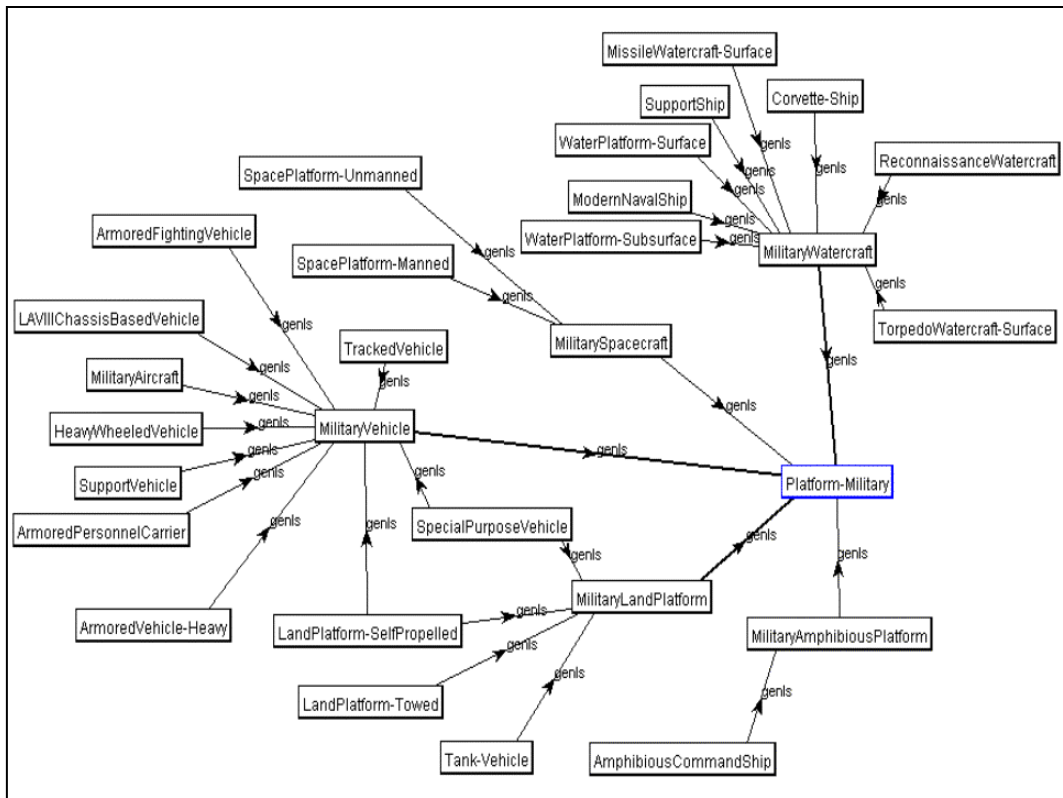


Figure 33: Dynamically generated graph showing a fragment of the Cyc ontology.

using the Cyc Browser who can therefore create lexical mappings at the same time as new ontology is created.

4.2 Lessons Learnt from the RKF Project

The Knowledge Entry work done under the RKF project was really the first time that regular people (untrained in logical representation or ontology) had participated. Their observations about what was intuitive and unintuitive about the system helped immensely with interface design. All of these points seem obvious with hindsight, but were hard-won through practical experience, and development of the infrastructure required to support them.

4.2.1 Listen to the SMEs (But Not Too Much)

It has proven extremely important to solicit from SMEs information about how they would normally conceptualize their domain, and then attempt so far as possible to place the domain of discourse into their worldview rather than Cyc's. This arose repeatedly in RKF, reaching its zenith in the creation of the Analysis Diagram Tool. At the same time, this must be done in a way that integrates tightly with Cyc's underlying structure, and is as domain-independent as possible. In the case of the ADT, it is not tied to terrain analysis or military applications, and is only slightly specific to the representation of the intelligence analysis process. It allows for the representation of arbitrary decision-making processes, without constraining the domain or the information sources.

In general, there are many ways in which the user's domain of discourse or format of communication may differ from that which is more natural for ontologists and knowledge engineers. Some examples are:

- Difference in use of terminology (see in the next sub-section);
- Declarative descriptions versus procedural instructions; and
- Linear text versus graphics.

These vary depending on the situation on ease of implementation, and the benefit derived.

4.2.2 Avoid Subtle Distinctions

A recurrent problem throughout the RKF project was the fact that the relationship between concepts as expressed in English, and as expressed in CycL is a complex one (see Section 1.2). In particular, English tends to have more vagueness and ambiguity, and this under-specification means that the process of translation into CycL requires the SME to appreciate a variety of subtle distinctions that can only be imperfectly expressed in English.

For example, SMEs stumbled continually over the distinction between types and instances of those types; this distinction is very important, but is often elided in English (e.g. “Moses and Princess are dogs.” versus “Poodles and dachshunds are dogs.”). In CycL, this difference is represented in the two predicates `isa` and `genls`. When describing decision-making rules, it is more common to deal with types (e.g. “military units”), whereas the description of a specific scenario will deal with instances (e.g. “Red 3rd MRD”). On one occasion, RKF SMEs attempted to replace “is high ground” with “is Unit #7”; the former designates a type of place, the latter a specific place, but both made sense in English.

Problems of this type were mitigated by two means: clarifying the distinction with helpful examples; and automatically detecting and repairing cases where the intent of the SME is clear. While significant improvements have been made, this remains an area of active research.

4.2.3 Avoid Unnecessary Delay

Initial versions of KRAKEN components accepted user input and did not display a result until Cyc had processed it as far as it was able; worse, they would then provide all information that the user might require. This made the interface slow and clumsy to use. Introduction of Dynamic HTML helped a little in tidying up the screen, but did not resolve the issue of latency, and actually encouraged the provision of additional information.

Feedback from SMEs in years one and two strongly indicated that these delays were unacceptably disruptive to the knowledge entry task. Work in Year Three therefore concentrated on alleviating this problem. This was done partly by making the underlying inference faster, but mainly by producing Java interfaces that permitted proper asynchronous interface management. No longer should interface responsiveness be tied directly to the performance of Cyc. Information is presented to the user as it becomes available, and the user can be kept continuously apprised of status.

Concept Creator

Please describe the new concept:

The best way to refer to it is

The best way to refer to something is both terse and generally understood.
For example, 'George W. Bush' is a better preferred name than 'Dubya', or 'W'.

Please fill in at least one of the following about it:

| | |
|---|---|
| It is similar to <input type="text" value="scooter"/> | e.g. George W. Bush is similar to Bill Clinton. Cyc will ask you to confirm similarities between them. |
| It is an instance of <input type="text"/> | e.g. Lassie is an instance of dog. |
| It is a more specific kind of <input type="text"/> | e.g. Siamese is a type of cat. |

For many concepts, it is inappropriate to use both the 'instance of' and 'kind of' boxes.

Figure 34: Here the Concept Creator tries hard to explain the instance/type distinction.

Precision Suggestor

In the sentence

[+] A Segway HT is a type of self-powered device.

could you replace the phrase "self-powered device" with any of the following?

[No, the original sentence is fine]

- [+] vehicle
- [+] internal combustion powered device
- [+] battery powered device
- [+] explosive device
- [+] rocket
- [+] internal combustion powered device
- [+] self-powered projectile
- [+] radio-controlled self-powered device

Figure 35: The Precision Suggestor suggests some possible replacements for one term in the sentence. Note that it is possible to choose multiple replacement sentences, each of which can be strengthened in turn.

4.2.4 Track What the User Says

Extension of KRAKEN's parsing capabilities, particularly to accommodate idiom or domain-specific expressions must rely not just on general domain-independent lexical information, but also on tracking how SMEs actually express themselves. In many cases, a SME must try several ways of phrasing an input sentence before KRAKEN will accept it.¹¹ Cyc should be capable of tracking the fact that all these attempts mean the same thing, and extending its parsing support accordingly on either a domain-specific or user-specific basis. Some initial work has been done on this, but much remains to be done.

4.2.5 Focus on Interface Sweet-spots

Although the goal of the KRAKEN system is to be a domain-independent knowledge entry system, there are many specific types of interface modality that are suggested by a specific domain, but which can be generalized. The Factivore is an example of this.

4.2.6 Overall Insights

Throughout the RKF project, a great many KE tools have been developed of various different types; each was directed at some sweet spot in terms of interface efficiency and productivity. It is useful to draw out some possible classifications and common themes.

¹¹ Ironically, SMEs sometimes attempt to anticipate parsing problems and use deliberately awkward English to "help" Cyc. Some science fiction movies encourage this practice.

The first obvious dimension along which KE tools can be classified is the extent to which they constrain the user. Clearly, the more the user's input is constrained, the more reliable the processing of that input can be made to be. At one end of the scale lies the "Say This" box (using the Sentence Reader), where the user can input any sentence (statement or question); while this is a basic ability for any science fiction AI¹², it has proven to be a genuinely hard problem, and it will be a long time before AI systems can chat as freely as humans. At the other end of the scale lie menus and Yes/No questions, and somewhere in the middle are forms that require user input (typically of noun phrases).

A related dimension is that of initiative: KRAKEN is a mixed-initiative dialog system, which means that either the user controls the focus of a KE session, or the system can suggest things for the user to do. A simple example is the Yes/No or multiple-choice questions that appear when clarifying ambiguities in parsing. A more complex example is the Concept Refinement Interview (based on the Salient Descriptor) where Cyc uses its large common sense knowledge base to induce new questions for the user; these questions may be either Yes/No or fill-in-the-blanks. Another good example of a system-initiative tool is the Precision Suggestor, which suggests stronger versions of a fact (e.g. for "Rover is a dog." it might suggest "Rover is a [specific breed].").

When the system is devising its own questions, it must ensure that they are both plausible and interesting (in terms of the statements they are suggesting); each of these criteria has at least two levels of satisfaction. At a basic level, plausibility implies that the suggested fact should not contradict existing knowledge, and should also be semantically well formed. At a higher level, the Salient Descriptor uses information from sibling instances to induce what would be plausible attributes for some concept. Basic interestingness is equivalent to novelty; it is uninteresting to learn a fact that is already known¹³. Advanced interestingness is subtler, and is as yet under-explored; this provides a significant opportunity for future research. For example, a statement is likely to be interesting if it will make likely future queries answerable (or is otherwise relevant to common kinds of reasoning in the user's domain) but this does not make it plausible unless it is also known that those queries should be answerable. It should be noted that KRAKEN does not perform any rigorous probabilistic analysis, yet does succeed in asking good questions. More mundanely, Factivore templates permit ontologists to seed the knowledge base with suggestions for plausible and interesting facts¹⁴; this process is increasingly subject to automation.

Although these system-initiative tools were developed for use by untrained SMEs, they have proven very popular with trained ontologists, and have therefore been migrated into the regular Cyc browser. The same principle has also been used to develop new system-initiative tools for use only by ontologists; for example, the 'Disjointness Tool' suggests that collections may have no instances in common, and allows the user to strengthen that claim up the collection hierarchy.

¹² Being apparently easier than, say, handling contradictory input.

¹³ Admittedly, this breaks down for attributes that may have several values, like "is friends with" or "speaks language".

¹⁴ At present, this arguably comes at the expense of novelty, because the Factivore will present the same questions for a concept, regardless of whether they are already known. In practice, this is an advantage because it permits the user to examine and correct existing knowledge, and the parallel nature of the interface means that the user's time is not wasted. In future, as Factivore templates become more dynamic, it may cease to be an issue.

4.3 Ongoing and Future Work

RKF has contributed significantly to Cycorp's core technologies. This will be carried forward in a number of different ways. In particular, many current projects — both research and commercial — will rely on this technology.

4.3.1 Natural-Language Processing

Cycorp will continue to expand and improve Cyc's lexicon, concentrating not only on the requirements of individual projects, but also on broad coverage of commonly used terms, as indicated by analysis of available corpora.

More advanced work will take place to improve Cyc's parsing ability via example-based machine translation (EMBT). This entails building up a corpus of mappings between NL expressions and their CycL meanings, and then using that to induce general templates.

The Interlocutor's model of representing parse results and discourse in the knowledge base — supporting deferred resolution, anaphora, and undo — may not appear with exactly the interface described here, but will be integrated into the Cyc system over the course of time.

4.3.2 Interfaces

The Factivore and Query Library are rapidly being assimilated into almost every project, as they provide immense benefits to both SMEs and ontologists. The Analysis Diagram Tool will be central to an ongoing SBIR project, and, as that work matures, will be migrated to serve the needs of other current Cycorp projects focused on intelligence analysis.

The current UIA continues to provide benefit for SME knowledge entry, but because of its interface limitations, it is likely that more and more of its benefits will be ported into other interfaces until its entire functionality is subsumed elsewhere. In particular, the Salient Descriptor — formerly part of the Concept Refinement Interview — is now integrated with the Factivore within the regular Cyc Browser used by ontologists, and will continue to be developed and used. The Precision Suggestor is similarly integrated with the Cyc Browser, and there are a host of ideas on how to develop this concept of strengthening further that will not only assist SMEs in knowledge entry, but will also assist ontologists in repairing and improving Cyc's ontology. See Section 4.1 for details.

4.3.3 Open Issues

One area of future research for knowledge entry is to do better at hiding some of the subtler aspects of the ontology from the user, such as the type/instance distinction. Some progress has been made on this, but it continued to be a significant problem, up and including the Year Three evaluation.

Another open issue is how to improve the questions that Cyc induces through the Salient Descriptor. It is hoped that the synergy between the Salient Descriptor technology and the handcrafted Factivore templates will lead to the development of insights that can be used to better direct Cyc's queries.

Much work remains to be done on solidifying the integration with collaborator technologies, such as NWU's Analogy Developer and ISI's Rule Inductor, or similar. It is anticipated that rule induction technology will play a significant role in BUTLER, Cycorp's ongoing project under IPTO's machine learning pilot program.

4.3.4 Potential Applications

The range of potential applications of RKF/KRAKEN technology is virtually unlimited. Areas worth exploring include:

- Intelligent personal digital assistants
- Training tools for the classroom
- Staff officer in a box, to support junior military officers in making real-time decisions on the basis of high-bandwidth highly-dynamic data
- Supporting any type of intelligence analysis project

5 Publications

The following peer-reviewed papers describe some of the results of the RKF project at Cycorp:

A Process Ontology (EKAW 2002)

Stuart Aitken, Jon Curtis

This paper describes an ontology for process representation. The ontology provides a vocabulary of classes and relations at a level above the primitive event-instance, object-instance and timepoint description. The design of this ontology balances two main concerns: to provide a concise set of useful abstractions of process, and to provide an adequate formal semantics for these abstractions. The aim of conciseness is to support knowledge authoring - ideally a domain expert should be able to author knowledge in the ontology - providing a sufficiently advanced toolset and interface has been implemented to support this task.

Knowledge Acquisition incorporating Interactive NL Understanding (ACL 2002)

David Schneider, Kathy Panton, David Baxter, Jon Curtis, Pierluigi Miraglia, Nancy Salay

A central issue in building knowledge-based systems is making them available to and understandable by naive users. This demonstration shows how a user and a knowledge-based system can collaborate to both create new knowledge and to extract existing knowledge from a knowledge base via natural language. KRAKEN is an interface designed for users who have expert knowledge of some field, but no special training in knowledge representation, logic, or the like. Our basic assumption is that the interfacing medium between the user and the knowledge base should be as close as possible to natural language (though see (Clark et al. 2001) for another possible approach).

The demonstration will show how we parse from English into CycL (the logical representation used by the Cyc ontology and inference engine) using a variety of different parsing methods and post-processing steps, and will also demonstrate how this versatile representation language can be rendered back into English that is suitable for lightly trained users. These NLP tools are integrated with a large number of knowledge-engineering tools, since we (both Cycorp and the NLP community at large) have not succeeded in building a natural language system that can do everything necessary to construct knowledge via simple back-and-forth NL dialogue with a user.

Cyc's natural language understanding abilities consist of a lexicon with syntactic and semantic information, a hybrid top-down/bottom-up parsing system, a reformulation module, and a generation system.

Knowledge Formation and Dialogue Using the KRAKEN Toolset (IAAI 2002)

Kathy Panton, Pierluigi Miraglia, Nancy Salay, Robert C. Kahlert, David Baxter, Roland Reagan

The KRAKEN toolset is a comprehensive interface for knowledge formation and acquisition based on the Cyc knowledge base, currently in the operational prototype stage. In particular, the KRAKEN system is designed to allow subject-matter experts to make meaningful additions to an existing knowledge base, without the benefit of training in the areas of artificial intelligence, ontology development, or logical representation. Users interact with KRAKEN via a natural-language interface, which translates back and forth between English and the KB's logical representation language. A variety of specialized tools are available to guide users through the process of creating new concepts, stating facts about those concepts, and querying the knowledge base. KRAKEN has undergone an independent performance evaluation. In this paper we describe the general structure and several of the features of KRAKEN, focusing on key aspects of its functionality in light of the specific knowledge-formation and acquisition challenges they are intended to address.

Experimental Evaluation of Subject Matter Expert-oriented Knowledge Base Authoring Tools (PerMIS 2002)

Robert Schrag, Mike Pool, Vinay Chaudhri, Robert C. Kahlert, Joshua Powers, Paul Cohen, Julie Fitzgerald, and Sunil Mishra

We describe a large-scale experiment in which non-artificial intelligence subject matter experts (SMEs)—with neither artificial intelligence background nor extensive training in the task—author knowledge bases (KBs) following a challenge problem specification with a strong question-answering component. As a reference for comparison, professional knowledge engineers (KEs) author KBs following the same specification. This paper concentrates on the design of the experiment and its results—the evaluation of SME- and KE-authored KBs and SME-oriented authoring tools.

Evaluation is in terms of quantitative subjective (functional performance) metrics and objective (knowledge reuse) metrics that we define and apply, as well as in terms of subjective qualitative assessment using several sources. While all evaluation styles are useful individually and exhibit collective power, we find that subjective qualitative evaluation affords us insights of greatest leverage for future system/process design. One practical conclusion is that large-scale KB development may best be supported by “mixed-skills” teams of SMEs and KEs collaborating synergistically, rather than by SMEs forced to work alone.

An Interactive Dialogue System for Knowledge Acquisition in Cyc (IJCAI 2003)

Michael Witbrock, David Baxter, Jon Curtis, Dave Schneider, Robert Kahlert, Pierluigi Miraglia, Peter Wagner, Kathy Panton, Gavin Matthews, Amanda Vizedom

Cycorp has developed a knowledge acquisition system, based on Cyc, that can engage a user in a natural-language mixed-initiative dialogue. In order to achieve an intelligent dialogue with the user, it employs explicit topic- and user-modeling, a system of prioritized interactions, and a transparent agenda to which either the user or the system can add interactions at any time. Interactions initiated by the Cyc system are directed at closing system-perceived knowledge gaps, at optimizing the inferential utility of existing and added knowledge, and at generally facilitating the user’s knowledge-entry task. This is done both deductively, in response to explicit knowledge elicitation rules, and inductively, from the existing content of the Cyc Knowledge Base.

Evaluating SME-Elicited Knowledge (IJCAI 2003)

Julie Fitzgerald, Mike Pool, Bob Schrag

Evaluation of mixed-initiative systems is an evolving field of study. Information Extraction and Transport (IET) has been conducting large-scale knowledge base evaluations for the past six years. We describe an approach to mixed-initiative systems evaluation that was devised for DARPA’s Rapid Knowledge Formation (RKF) program and how this was implemented in the course of two very distinct domain challenge problems. We assess the methods and results used and make recommendations for future mixed-initiative evaluations.

Appendix A: Ontology Required for Evaluation Results

The work completed by the SMEs during the evaluation period was extended and edited in several ways by ontologists in order get the analysis procedure inferences working.

Hand-merging

The assertions made in the individual graph microtheories¹⁵ were copied to a KE¹⁶ file. Working in the KE file by hand, an ontologist replaced the references to specific SME working graphs and associated microtheories and information structures with references to the unified Observation graph and its associated microtheories and infrastructures. These assertions were then loaded into Cyc's Knowledge Base.

The result was a raw duplication in a single diagram of the all of the sub-diagrams created by the SMEs in their own workspace graphs. Because the merging was accomplished via duplication into another microtheory, the original SME-created graphs remained untouched, and could be referred to later for clarification or comparison.

This merged diagram now contained several sub-diagrams that were not connected. Additionally, there had been some duplication of effort between SMEs; some of the queries had more than one node. These duplicated nodes were merged by hand. To connect sub-diagrams, an ontologist drew the link according how each sub-diagram fitted into the overall analysis structure, as shown in the SME-drawn chart that was serving as reference. The result of linking and resolving of duplications was a single chart containing all of the SMEs' work.

Well-formedness (wff) problems: Type/Instance confusion

A recurrent problem was found with respect to the SME-created queries. The type/instance confusion that had been observed with earlier tools raised its head again here. Several queries had type level terms where instance level terms needed to be. In each case, the SME had clearly wanted a typed variable — that is, an unknown instance of the type. Ambiguity in the generated paraphrases led them to believe that this is what they had chosen, and the Cyc Term Chooser did not object to, or prevent, their substitution of a type-level concept in a place that syntactically required an instance. This resulted in a number of queries that (a) were not well formed, and (b) did not contain variables, and so could not be linked to other queries via variable mapping. The SMEs were therefore not able to use the ADT to specify when results should be inherited from one step to another, when either of the steps contained this type of error.

¹⁵ Cyc's knowledge is divided into a hierarchy of microtheories, each of which represents a specific knowledge context, and which can be used to separate knowledge.

¹⁶ The KE file format is used by ontologists for large-scale knowledge entry. It is very similar to CycL (and is therefore readily and automatically translated into it), but has some syntactic sugar to save on typing. With the development of powerful knowledge entry tools such as those produced for RKF, the need for KE files is diminishing.

The SMEs confirmed the source of these errors. Most occurred when a SME was modifying a Query Library query template, and went to substitute in for an individual-denoting variable. Clicking on the variable, the SME would get the Cyc Term Chooser. The SME could go to the "Entry" tab, type in something like "high ground," select "English" as the input form, hit "Expand" for the Chooser to search for matching CycL concepts, get "high ground" as the gloss for #`$TerrainHighGround` appearing in the term box, and replace. The SME would then believe that he had put an instance of high ground in this place, when he had actually put in the collection of all high ground. The gloss "high ground" is natural, but not informative enough in this case, and perhaps should not be used at all — perhaps only the more wooden "the collection of all high ground" and "an instance of high ground" should be used in this kind of clarification/term choice dialogue. Independently, it should be possible to prevent the Term Chooser from allowing such ill-formed substitutions.

In one SME's graph, these problems were found in almost all queries. Another SME's graph contained no such errors. It is noteworthy that this SME had experience with CycL from the Year Two evaluation, and was aware of the danger of type/instance confusions. In total, out of forty-six nodes with SME-written queries, type/instance-related well-formedness problems occurred in nine of them.

Queries in which Type/instance confusions were corrected by an ontologist:

1. "What values of UNIT and PATH3 are there such that defending forces place an obstacle to UNIT on PATH3?"
(`obstacleOnToBy ?PATH3 ?UNIT DefendingForce`)

This query was not well-formed because the third argument of `obstacleOnToBy` should be an instance of `IntelligentAgent`, while `DefendingForce` is a collection.

Diagnosis: SME intended a variable constrained to be an instance of `DefendingForce`.

Changes: Query Formula edited to:

```
(and
  (obstacleOnToBy ?PATH3 ?UNIT ?DEF)
  (isa ?DEF DefendingForce))
```

2. "What values of WIDTH are there such that shadow is WIDTH wide?"
(`widthOfObject Shadow ?WIDTH`)

This query was not well-formed because the first argument of `widthOfObject` should be an instance of `SpatialThing`, while `Shadow` is a collection.

Diagnosis: SME intended a variable, constrained to be some instance of "shadow," in the observation sense, i.e., a dead zone in observation. Given the position of the query in the diagram, the dead zones – the possible values of `?SHADOW` -- would have been found by the previous query.

Changes: Query formula edited to:

```
(widthOfObject ?SHADOW ?WIDTH)
[The first argument is now a variable rather than collection.]
```

A variable mapping was added so that the input values of ?SHADOW were inherited from the previous query.

3. "What values of HIGH-GROUND are there such that HIGH-GROUND supports movement by foot travel?"
(pathTrafficableForTransportVia TerrainHighGround byFoot)

This query was not well formed because the first argument of pathTrafficableForTransportVia needs to be an instance of Path, while TerrainHighGround is the collection of high ground; also, the second argument must be a type of transporter, while byFoot is a relation.

Diagnosis: Discussed with SME. SME intended a variable constrained to be an instance of TerrainHighGround. The collection/instance confusion was corrected by replacement with variable and addition of first conjunct. SME did not, however, realize that this predicate only discussed path trafficability, rather than any piece of terrain. This seemed to be a problem with the English gloss on the builder query, which was not sufficiently clear. This is a more complex fix, requiring some indirection.

Changes: Query formula edited to:

```
(and
  (isa ?HIGH-GROUND TerrainHighGround)
  (pathTrafficableForTransportVia ?PATH
    TransportationDevice-Vehicle)
  (pathIntersectsRegion ?PATH ?HIGH-GROUND))
```

4. "What values of HIGH-GROUND are there such that HIGH-GROUND supports movement by self-powered vehicles?"
(pathTrafficableForTransportVia TerrainHighGround
 TransportationDevice-Vehicle)

This case was quite similar to #3. The query was not well-formed because the first argument of pathTrafficableForTransportVia must be an instance of path, while TerrainHighGround is a collection.

Diagnosis: Discussed with SME. SME intended a variable constrained to be an instance of TerrainHighGround. The collection/instance confusion was corrected by replacement with variable and addition of first conjunct. SME did not, however, realize that this predicate only discussed path trafficability, rather than any piece of terrain. This seemed to be a problem with the English gloss on the builder query, which was not sufficiently clear. The SME had created this query by modifying the same builder query used for #3, hence the same problem. This is a more complex fix, requiring some indirection.

Changes: Query formula edited to:

```
(and
  (isa ?HIGH-GROUND TerrainHighGround)
  (pathTrafficableForTransportVia ?PATH
    TransportationDevice-Vehicle)
  (pathIntersectsRegion ?PATH ?HIGH-GROUND))
```

5. "Is it true that avenues of approach and machine guns are in the line of sight of one another?"
(lineOfSight-ObjectToObject AvenueOfApproach MachineGun)

This query was not well-formed because both the first and second arguments of `lineOfSight-ObjectToObject` need to be instances of `PartiallyTangible`, while `AvenueOfApproach` and `MachineGun` are both collections.

Diagnosis: Discussed with SME. SME intended variables restricted to be instances of these two collections.

Changes: Query formula edited to:

```
(and
  (isa ?AA AvenueOfApproach)
  (isa ?MG MachineGun)
  (lineOfSight-ObjectToObject ?AA ?MG))
```

6. "Is it true that obstacles and observation posts are in the line of sight of one another?"
(lineOfSight-ObjectToObject Obstacle ObservationPost)

Similar to above.

Changes: Query formula edit to:

```
(and
  (isa ?OBST Obstacle)
  (isa ?OP ObservationPost)
  (lineOfSight-ObjectToObject ?OBST ?OP))
```

7. "Is it true that defensive positions and obstacles are in the line of sight of one another?"
(lineOfSight-ObjectToObject Obstacle DefensivePosition)

Similar to above.

Changes: Query formula edited to:

```
(and
  (isa ?DP DefensivePosition)
  (isa ?OBST Obstacle)
  (lineOfSight-ObjectToObject ?DP ?OBST))
```

8. "Is it true that the shortest distance between artillery of Defender and avenue of approach is firing range?"

```
(and
  (isa ?FA
    (SubcollectionOfWithRelationFromFn FieldArtillery
      possessiveRelation DefendingForce))
  (lessThanOrEqualTo ?DIST ?RANGE)
  (weaponEffectiveRange ?FA ?RANGE)
  (distanceBetween AvenueOfApproach ?FA ?DIST))
```

This query had several problems, though type/instance problems were the most significant. `AvenueOfApproach` and `DefendingForce` both appear here where individual instances are

needed. Other problems: the Subcollection expression should have been automatically reformulated; this was done by hand and the example handed off to the reformulator team.

Changes: Query formula edited to:

```
(and
  (isa ?UNIT DefendingForce)
  (genls ?WEAPON-TYPE FieldArtillery)
  (greaterThanOrEqualTo ?NUMBER1 0)
  (equipmentOfUnit-TypeCount ?UNIT ?WEAPON-TYPE ?NUMBER1)
  (relationAllInstance weaponEffectiveRange-Max ?WEAPON-TYPE
?RANGE)
  (isa ?AA AvenueOfApproach)
  (lessThanOrEqualTo ?DIST ?RANGE)
  (distanceBetween ?AA ?UNIT ?DIST))
```

9. "What values of WEATHER-FORECAST are there such that WEATHER-FORECAST limits observation of red units from blue units and WEATHER-FORECAST is the weather forecast for Fort Hood for March 18, 2003?"

```
(and
  (limitsObservationOfFrom ?WEATHER-FORECAST RedUnit BlueUnit)
  (weatherForecastForRegion ?WEATHER-FORECAST
    ArmyBase-FortHood-Grounds-Texas
    (DayFn 18
      (MonthFn March
        (YearFn 2003))))))
```

This query was not well formed because both of the second and third arguments to `limitsObservationOfFrom` must be instances of `SpatialThing`, but `RedUnit` and `BlueUnit` are collections.

Diagnosis: SME intended variables constrained to be instances of these types.

Changes: Query formula edited to:

```
(and
  (isa ?REDUNIT RedUnit)
  (isa ?BLUEUNIT BlueUnit)
  (limitsObservationOfFrom ?WEATHER-FORECAST ?REDUNIT ?BLUEUNIT)
  (weatherForecastForRegion ?WEATHER-FORECAST
    ArmyBase-FortHood-Grounds-Texas
    (DayFn 18
      (MonthFn March
        (YearFn 2003))))))
```

Variable Mappings

In the course of constructing analysis diagrams, there are two contexts where SMEs must form a mapping between two sets of variables: when combining two queries together in the Query Library; and when assigning the semantics to the edges in the diagram to show the inheritance of bindings from previous queries for the context of a node. The same interface dialog is used in both cases.

Variable mappings were added by ontologists for those arcs where well-formedness problems in the queries had prevented entry by SMEs. Had there been time after the well-formedness

problems were corrected, it would of course have been far better to have the SMEs do the variable mapping themselves. The mappings SMEs did create showed that the variable mapping interface worked well and was sufficiently intuitive for SMEs to make the assertions they intended. However, there was no such time, and these mappings were completed by ontologists. The reference diagram, SME comments, and memory of previous discussion were used to determine the intended mapping.

Additional Node

An initialization node was added by an ontologist:

“What is the mission, and where and when does it occur?”

Really, this is not part of the analysis proper, but a precursor in which Cyc determines what data to run the analysis over. The intended mode for this was to have the analysis represented, then have an interface for executing it, in which the scenario to be analyzed could be specified with some small number of selections that would instruct Cyc which bodies of data to connect to. This interface, however, was not feasible within the available resources.

Variable mapping missing from arc

For two SME-created arcs, the SMEs either did not get to, or were not able to specify, the variable mapping. In both cases, however, the intended mapping was quite clear, and was added by an ontologist.

1. Link from “Do force elements have thermal vision capability?” to “Is the ambient temperature within operational parameters for the thermal devices?”

Added variable mapping to pass the thermal equipment types found in the source query as the thermal device types to be considered in the target query.

2. Link from “Is the ambient temperature within operational parameters for the thermal devices?” to “What is the max effective range of the force elements thermal viewing devices?”

Added variable mapping to pass the useable thermal equipment types found in the source query to the device types to be considered in the target query.

No associated query

Four nodes were created and given an English gloss by the SMEs, but the SMEs did not create queries for them. In the case of the fourth query, the SME assigned to it explicitly decided that because it required input from four other queries, it was too complex for the SME tools and should be done by an ontologist. In fact, that query should have been handled by a Value Table, had we had the Value Table code ready in time.

1. “What time range is the restriction expected to endure for?”
2. “What range will ground observation be restricted to?”
3. “Given maximum ground observation range, is some defensive area within observation range?”
4. “What is maximum ground observation range?”

Working around absence of planned Value Tables feature

Two nodes required the determination of an answer by considering together the outputs of several different previous queries.

1. Max Observation Range: “What is maximum ground observation range?”

This query had to be answered by consideration of the results of several other queries: the max range of any night vision devices, thermal devices, and flares, as well as the forecasted visibility and time of day or level of illumination. It would therefore have been best handled by a value table. As a workaround, an ontologist wrote a corresponding query to perform the needed comparisons and calculations. Functionally this worked fine, however, it is not something that could be expected of a SME user, and should be replaced by Value Table functionality in the future (see Section 2.3.2.1.4.4 above).

2. Final Assessment of tactical significance: “How tactically significant is this piece of terrain wrt Observation?”

This query had to be answered by consideration of the results of several other queries:

- “High ground has intervisibility with AA segment greater than or equal to 50 m in length?”
- “High ground has foot access?”
- “High ground has vehicle access?”
- “AA segment is within range of defenders artillery?”

An ontologist hand wrote rules logically equivalent to the rules set that would have followed had the value table been filled in, using SME comments as a guide.

Queries not working as written

Several queries were well formed but did not answer as expected. These required debugging and repair by ontologists.

1. “Defense area is occupied?”

changed from

```
(and
  (forceElement-FromGIS ?X)
  (isa ?AREAL DefensivePosition)
  (locatedAtPoint-SurfGeog-FromGIS ?X ?AREAL))
```

to

```
(and
  (forceElement-FromGIS ?X)
  (isa ?AREAL DefensivePosition)
  (objectFoundInLocation ?X ?AREAL))
```

This was required because the constraints on `locatedAtPoint-SurfGeog-FromGIS` are narrow in a way that is not obvious given its English generation. This could be avoided with more precise generation.

2. “Is there high ground to the side of AA forward of a defensive area?”

changed from

```
(and
  (tacticalLine-FromGIS ?X)
  (tacticalArea-FromGIS ?AREA1)
  (isa ?AREA1 BattlePositionOccupied)
  (isa ?OBJECT15 TacticalTerrainObject)
  (isa ?OBJECT15 TerrainHighGround)
  (isa ?X
    (SubcollectionOfWithRelationFromTypeFn Path-Spatial
      avenueOfApproachInCOA CourseOfAction-Offensive))
  (inFrontOf-Generally ?OBJECT15 ?AREA1)
  (toTheSideOf ?OBJECT15 ?X))
```

to

```
(and
  (tacticalLine-FromGIS ?X)
  (tacticalArea-FromGIS ?AREA1)
  (isa ?AREA1 BattlePositionOccupied)
  (isa ?OBJECT15 TacticalTerrainObject)
  (isa ?OBJECT15 TerrainHighGround)
  (isa ?X AvenueOfApproach)
  (inFrontOf-Generally ?OBJECT15 ?AREA1)
  (toTheSideOf ?OBJECT15 ?X))
```

This normally would have been taken care of by reformulation.

3. “High ground is greater than 0.25 km and less than or equal to 18.0 km from the AA?”

changed from

```
(and
  (isa ?AVENUE AvenueOfApproach)
  (isa ?TERRAIN TerrainHighGround)
  (lessThanOrEqualTo (Kilometer 18) ?DISTANCE)
  (greaterThan ?DISTANCE (Kilometer 0.25))
  (distanceBetween ?AVENUE ?TERRAIN ?DISTANCE))
```

to

```
(and
  (isa ?AVENUE AvenueOfApproach)
  (isa ?TERRAIN TerrainHighGround)
  (lessThanOrEqualTo ?DISTANCE (Kilometer 18))
  (greaterThan ?DISTANCE (Kilometer 0.25))
  (distanceBetween ?AVENUE ?TERRAIN ?DISTANCE))
```

This was necessary because the argument order was reversed in the 3rd conjunct.

4. “High ground has intervisibility with AA segment greater than or equal to 50 m in length?”

changed from

```
(and
  (greaterThanOrEqualTo ?LENGTH (Meter 50))
  (spatiallyIntersects ?REGION3 ?X)
  (lengthOfObject ?AREA4 ?LENGTH)
  (different ?REGION3 ?X)
  (spatialIntersectionOf ?AREA4 ?GROUP)
  (elementOf ?REGION3 ?GROUP)
  (elementOf ?X ?GROUP)
  (lineOfSight-ObjectToRegion ?OBJECT15 ?REGION3))
```

to

```
(and
  (greaterThanOrEqualTo ?LENGTH (Meter 50))
  (isa ?X AvenueOfApproach-Ground)
  (lengthOfObject ?AREA4 ?LENGTH)
  (subPaths ?X ?AREA4)
  (lineOfSight-ObjectToObject ?OBJECT15 ?REGION3)
  (spatiallySubsumes ?REGION3 ?AREA4))
```

This was necessary because the spatial intersection vocabulary in the above case is not really suited for this type of case, and so is more difficult to work with, and requires more indirection, than the spatial subsumption vocabulary. This is a case of not having quite the right builder query available, and a SME trying to make the best of what was there.

Execution Forward rule writing

Once an intelligence analysis process has been represented as a diagram, it is necessary for Cyc to execute that process in order to perform the evaluation. The graph representation is associated with an underlying script, with each node represented by a scene within that script. As designed, this was to be done with the assistance of special code that generates the necessary rules. These rules are known as “forward rules” because they are executed eagerly, and their answers cached for maximal performance and reliability. While substantial progress was made on this automation, resources did not permit completion in time to be used for the evaluation. See Section 2.3.2.1.4.6 above.

Ontologists wrote the forward rules required for execution of the resulting analysis procedure. This emulated the forward rules that would have been written on-the-fly by the supporting code, had it been finished and integrated in time.

Using the automatically inferred analysis script representation, the ontologists created a forward rule for each “yes” arc, stating that if the query for the source scene has any full binding sets, the query formula of the target scene applies. This triggers the passage from one scene to the next.

For “no” arcs, ontologists created forward rules stating that if the query for the source scene has no known bindings, the query for the corresponding target scene applies. Ontologists also implemented the execution of variable mapping by constraining mapped variables in target query to allow as possible substitutions only output bindings for mapped variables in the source query.

While these hand-written rules are specific to the analysis process in question, the process of creating them is entirely determined, and thus amenable to automation.

GIS workaround

Ontologists hand-entered GAFs¹⁷ representing data available from GIS lookup or basic GIS calculations (such as Line Of Sight) that had been expected. This reflected facts that were obtainable from GIS, but for which the integration could not be relied upon.

¹⁷ Ground Atomic Formula – the simplest sort of fact assertible in Cyc.

Appendix B: Details of KB-based Parse Representation

The Interlocutor is a new interface design for visualizing natural-language based knowledge-formation in the Cyc knowledge representation system. It is a direct descendant of the UIA, which was the interface designed for and used in the first two years of the RKF project. The implementation of that interface is work in progress.

The design of the Interlocutor is the result of experience with the UIA, but perhaps even more of reflections on the users' experiences with the UIA and the Cyc system in the course of the Year 1 and Year 2 challenge problems. By Cycorp's findings, these were the factors that seemed to interfere negatively with user experience:

- Users felt they had little control in general over the direction of the “conversation”: in fact, they did not see their interactions with Cyc as a conversational *exchange* at all, but rather as a linear sequence of knowledge-entering prompts.
- The system did not react as expected in conversation: in particular the questions posed by Cyc as knowledge entering prompts (for example in Concept Refinement Interview) were not necessarily the ones an expert user would have chosen to pursue at a determinate juncture.
- There was much repetition of steps, and the system's attempt to interpret each user action unambiguously broke the natural flow of the information exchange. The inherent brittleness of semantic parsing constantly gets in the way.

The Interlocutor's design addresses these difficulties. The guiding goal is that of enabling a conversational exchange between the expert user and the knowledge representation system in which the system plays the role of a “good student” to a knowledgeable teacher.

Deferred Disambiguation and Unlimited Undo

SMEs interacted with the UIA typically by asserting a phrase, asking a query, or naming a concept whose properties they wanted to describe. In a significant number of such interactions, Cyc's interpretation of the phrase only partially succeeded: certain components or segments are correctly understood, but others aren't.

A robust knowledge acquisition model, therefore, should allow the user-system session to proceed on the basis of what has been so far well understood, and to postpone the resolution of those parts of a phrase, query, etc. that need further work. This is what the Interlocutor does. The interface displays graphically (through color coding) the elements of the discourse (user discourse, that is) that Cyc has so far been able to interpret, with some positive level of confidence. The user can then choose to facilitate the interpretation of the unrecognized elements, or simply forge ahead on the basis of what has been correctly recognized — for instance, if the unrecognized parts are not essential to current topic.

Of course, to do this the Interlocutor must store at any given moment an internal representation of the state of the discourse. This is achieved entirely through the representational resources of the Cyc KB. In other words, the Interlocutor's agenda of operations is at all times fully spelled out in CycL. It supports inference and is updated via the truth maintenance system (TMS). Unlike the

UIA, therefore, the Interlocutor offers the user the possibility of backtracking along the interaction history, and of modifying the interpretation of previous elements in the discourse.

Dialogue model

The UIA toolset was designed around the notion that the basic "unit of information exchange" in an expert-knowledge formation environment is the same as the basic unit of knowledge in Cyc, the knowledge representation system: that is to say, a single sentence (an assertion). In the Interlocutor the basic unit is a multi-sentence discourse — the target model is that of a conversation.

In ordinary discourse interpretation is dynamic: the understanding of a later assertion depends on

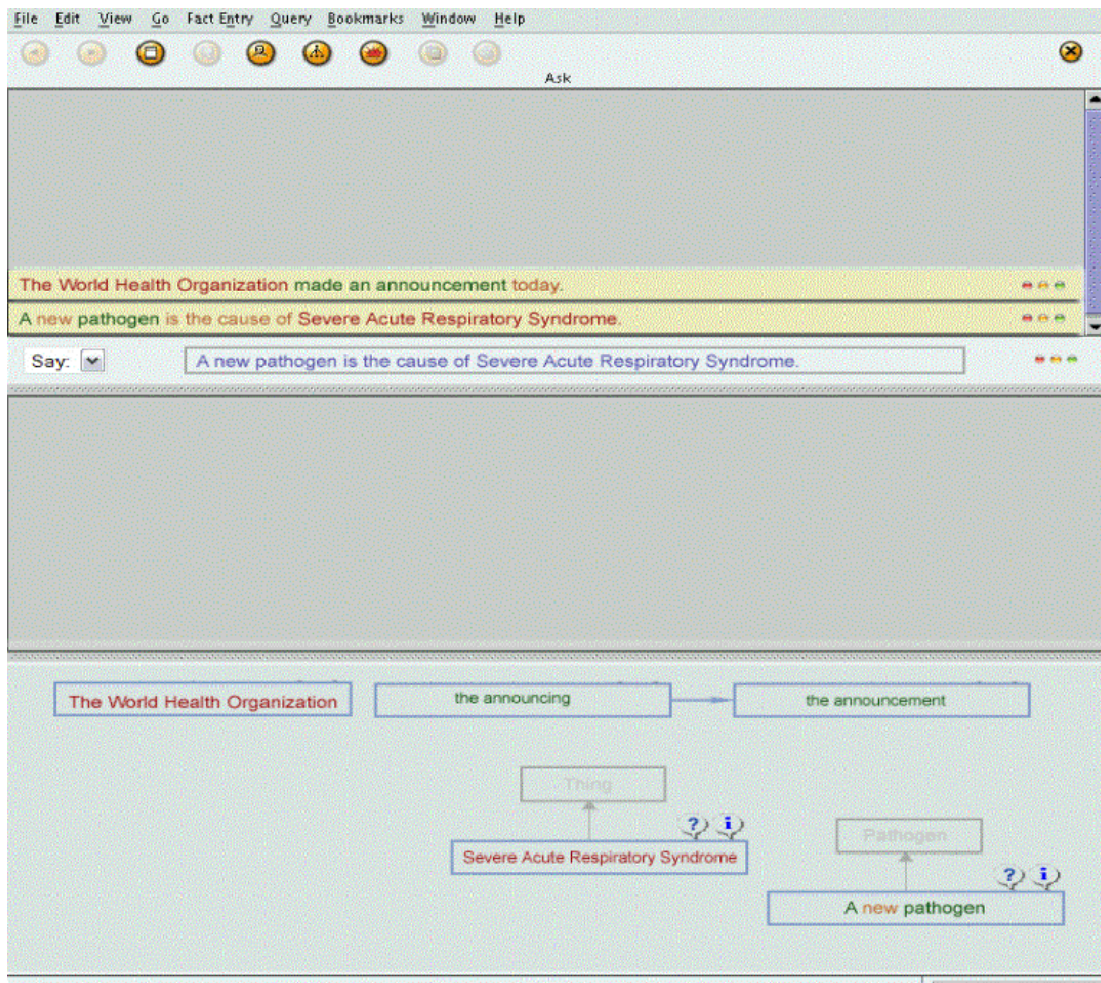


Figure 36: Mock-up of the Interlocutor interface to KRAKEN. The user types input (say from a news article) into the "Say" box. Past user utterances are listed at the top, color-coded by the level of system comprehension: red is unparsed, green is fully parsed, and orange is parsed with outstanding ambiguities. The utterances transition from red to green as the either Cyc works harder on them, or the user gives clarifying information. The entities referred to by the user are displayed graphically at the bottom. This graph can be manipulated as another interface modality. The blank panel in the middle is for system-initiative questions, which can be triggered by selecting the question mark on a term in the graph.

the interpretation of an earlier one. The Interlocutor supports the dynamic interpretation of successive assertions and/or queries from the user. It allows the user to employ ordinary means of referring to discourse entities that may be presupposed given the previous stages in the discourse, thus affording a much more transparent style of interaction. Additionally, it keeps track of the topical evolution of a discourse: for instance, when faced with multiple interpretations of an expression, it promotes interpretations that have been already used in the same discourse or are more coherent with those that have already been used, and demotes the alternatives.

Use of parsing

As noted above, central to the implementation of the Interlocutor is the representability of all aspects of the interpretive process — elements of the discourse model, fully or incompletely disambiguated interpretations, etc. — in the knowledge base itself. This extends to parsing the assertions and queries entered by the user.

Parsing structures are represented as (locally) persistent knowledge structures in the knowledge base, since this is necessary to support deferred disambiguation and unlimited undo. This representation is fully general and independent of the particulars of the current Cyc parsers: it could easily be adapted to different parsers. In addition, it turns out that much of the discourse model construction can be driven simply by logical rules once this layer of principled syntactic representation is available.

Appendix C: SME Questionnaire

The three SMEs who participated in the RKF Year Three evaluation were asked to answer some questions about their experience of working with KRAKEN. The full results are given in the table below. A few points stand out:

- The new Java tools (the Factivore and Analysis Diagram Tool) greatly enhance usability and productivity. Much more information could have been entered had these tools been available at the beginning of Year Three, rather than being developed throughout.
- The Analysis Diagram Tool greatly simplifies the entry of knowledge of certain complex types, and could even potentially be used in the field. Its use to represent intelligence analysis processes could assist the battlefield commander in managing his high-bandwidth high-dynamism information stream. Similarly, it could be used in training.
- Representing specializations in a collection hierarchy would benefit from some notion of inheritance and default values. In general, displaying default values is something that the Factivore ought to be able to do.
- As applied to terrain analysis, the system would have benefited greatly from a better integration with GIS than was achieved.

| <i>Question</i> | <i>Kerry Hines</i> | <i>Dutch Sley</i> | <i>Bill Mathews</i> |
|--|---|--|---------------------|
| Pick one or two User Interface technologies that were introduced in the course of RKF Year Three, and identify some type of knowledge that they enabled you to enter. | <p>Factivore templates: These were very beneficial in simplifying the process of entering basic parameters on weapons and equipment—in terms of both ease of entry and consistency of data fields. I think an improvement here would be clear class association – e.g., tanks, APC/IFVs, light trucks, heavy trucks, SP artillery, etc – with generalized, default performance parameters (speed, slope, vertical obstacle, etc.). I am not convinced that we got all of the associations correct because of the inherited hierarchy, which had some errors. The point of the default values would provide a rudimentary analytic approximation of battlefield performance in the event that the actual performance parameters were unknown or unavailable at the time of knowledge entry. Of course, there would need to be flags on the generic values so that users would be reminded to replace those values when possible.</p> <p>Diagram Editor: This is one of the primary products of our year’s efforts and a significant improvement for KE. The complexity of the knowledge that we proposed to enter required an ordered process to help the user capture the conditions (pre-conditions) and relationships without attempting to enter multi-tiered IF-THEN statements. While everyone is unlikely to consciously think in a flow-process logic, I think that the diagram editor is a positive step toward simplifying the knowledge entry process. From the limited opportunity I had to test the tool, I found that the current user interface left something to be desired in terms of a friendly display – the query list became cluttered (one had to remember the general sequence in which queries were created in order to find the one desired) and the graphic required considerable scrolling at times to view the desired node and had to be rebuilt each time it was re-opened -- but there is a sound foundation upon which to</p> | <p>I never had any luck with the first because of Firewall problems. The second interface was much easier. The net meeting training helped considerably. With some polishing, I was able to enter some knowledge related to observation.</p> | |

| | | |
|---|---|---|
| | <p>evolve the interface, and it has considerably broader applications than the military OCOKA problem, as a tool to help a user structure, organize, and evaluate the completeness of an analytic process on a problem/issue.</p> | |
| <p>Identify some type of knowledge that you wanted to be able to enter, but couldn't.</p> | <p>My knowledge entry was limited by time more than by technology. I generally found the process of knowledge entry to be very time consuming if there was any complexity at all to the concept being attempted. That said, I think that I generally constrained my entry to knowledge/concepts that I felt I could successfully enter.</p> | <p>I was never satisfied with the box "dead zones are parallel to defensive front?"</p> |
| <p>Pick one or two User Interface technologies that were introduced, and identify cases in which you could express knowledge in a comfortable intuitive form, or continue thinking in a natural way.</p> | <p>This is a deceptive question, because I think that we tend to adapt our methods for expressing knowledge to the conditions under which it must be expressed – e.g., there is a difference in communicating with a first grader or a graduate student, a native language speaker or a foreigner, or an experienced or inexperienced person on an issue. I think that my basic thoughts in 1, above, apply here as well. Because of the effort that we expended on elaborating the OCOKA trees and previous experience with a similar tool for developing technology roadmaps, I found that the diagram editor was generally a comfortable environment for expressing that knowledge. I did find it practical to manually make notes as I went along as a way of keeping a sort of scorecard on the knowledge entry process.</p> | <p>The nodes and edges fit my thought process</p> |
| <p>Identify some ways in which you were forced out of your natural way of thinking in the attempt to express your knowledge.</p> | <p>In general, the knowledge entry process requires considerably more deliberate thought and preparation than, say, the composition of a text document or the performance of a comparable analytic process manually. An analogous process is the research and development of (lesson) plans and materials prior to attempting to present a class or some type of presentation. In other words, one must know what one wants to do before starting. It is not an environment that favors on-the-fly creative thought.</p> | <p>Developing the questions using the query was much more difficult because not all military concepts were represented like dead zone, defensive area, crew-served weapons.</p> <p>Early in the project during the creation of the scenario I realized while looking at the map, Kerry and I used our experience to rapidly discard certain areas and focus on others. It felt odd at first that we would need to think about why and how we made our decisions so that we could teach it to cyc. If we had been teaching</p> |

Identify those Cyc interfaces and subsets of Cyc's capabilities, which would be especially useful in some operational context.

I thought that our initial intent for this effort was to produce a terrain-based knowledge base that would have utility in military operations. While we did produce a complete knowledge base, many of the tools are now in place to build the KB. With the increasing frequency and importance of first-encounter decision making by junior officers, there should be a corresponding need for capabilities to provide virtual support and experience to those young decision makers, which will help them predict where and what types of decisions they are likely to have to make, based on potential enemy actions (e.g., potential danger areas)

Not sure

humans the methods we used for the teaching would have been very different I think. It was a new way of thinking about how to teach for me. So in essence I think it was the transition from learned instincts to granular facts about tiny details that was the biggest change in thinking. I wouldn't normally consider slope direction on a hill when I think about a hill.

The headquarters on a modern battlefield is swamped with information coming from several sources at once. The speed and volume of information can be staggering. Soldiers are asked to take all that information and put it together to create a complete as possible vision of the battlefield, so the commander can make the most informed decisions possible. If the terrain analysis tool were fully fleshed out and connected with current

Identify those Cyc interfaces and subsets of Cyc's capabilities, which would be especially useful in some training context.

The legacy knowledge base as originally envisioned would be useful in both operational and training environments. Military units need to train using the same tools with which they will fight. As envisioned, our effort would have developed a capability that would allow junior officers to compare their analysis of a given problem to that accomplished by CYC, and thereby, learn which factors they might have overlooked or identify those aspects of CYC's analysis that required modification (e.g., mutual adaptive experience in preparation for actual operational applications).

Cyc could be used to train the Military Decision Making Process (MDMP)

information systems on today's battlefield it could be a great asset in the process of creating a full spectrum vision of the battlefield and could rapidly update that vision as new information was provided.

The military today uses electronic training with some exercises taking place primarily within computers. The logic and reasoning cyc could bring to that arena in making opposing forces in the computer act in a more realistic manner would provide more realistic training. It could also be used to assess the operation for both sides and determine if they used terrain to their best advantage.

For the previous two questions, identify any area in which Cyc is almost there, and describe what piece is missing.

CYC is almost there in providing the KE tools for elaborating analytic processes like the OCOKA mnemonic. To have practical utility, I would want to complete the entry of the terrain assessment process knowledge, interface to digital terrain data via a GIS so that the knowledge can be tested, and validate the knowledge against a set of operational scenarios/variants to ensure that it is expandible.

Not intuitive because of the GUI. I would prefer military jargon or plain English.

The terrain assessment tool we created in RKF3, if completed and integrated with more information systems in current operational use, could be used as a functioning tool in the commander's hands on

| | | | |
|--|---|--|--|
| | | | today's battlefield. |
| For those of you involved in previous years of RKF, or with comparable systems, compare the complexity or depth of knowledge that you were able to enter. | <p>In my earlier DARPA knowledge-based system development efforts, knowledge development was an iterative process, typically involving knowledge facilitators/elicitors. The SMEs developed rules and constraints for various operational situations, and the systems developers entered the knowledge. Then, the analytic performance was tested and evaluated by the SMEs to determine if the knowledge was being applied as expected and producing the intended results. Adaptability of the knowledge was limited to the capability to change some numerical values, turn selected rules and constraints on and off, or change an approximation constraint. The ultimate military users had to be brought into the process so that would have some confidence in the validity of the knowledge.</p> <p>CYC offers the potential of allowing a user to change basic knowledge “on the fly” and adapt the knowledge to the changing behavior of one or both opponents. Moreover, in comparison with the knowledge entry process for CYC during last year’s Challenge Problem, I would estimate that the revised Year 3 UI and tools provide the means to enter knowledge at least 5 times faster and to enter significantly more complex concepts – sort of like a move from elementary education to graduate school.</p> | N/A | NA |
| Describe any potential directions that you wanted Cycorp to take in this project that were decided against, or which time and resources did not permit. | <p>I think that the principal shortfall in our Year 3 effort is the limited amount of knowledge entry we achieved, which denies the opportunity to produce a proof of concept demonstration. This seemed to result from a number of factors. First, there was the limited ability to integrate with a GIS/terrain data in order to test the terrain queries. Second, we expended considerable effort evolving and documenting the knowledge before we made any serious entry, which can be attributed, at least in part, to the need to achieve a common understanding of what the focus and intent of the effort were and the level of detail required. Third, I believe that we expended too much effort entering force and equipment concepts at the expense of the real “meat” of the effort, the operational concepts related to the OCOKA analysis. The former are transitional data, since organizations and equipment change, but the fundamental terrain assessment concepts have a more enduring nature and were the primary intent for the legacy KDB.</p> | N/A | I would have liked to seen a tighter integration with the GIS systems. The GIS systems have a great deal of information cyc could use to reason with that we just where not able to get to in the time allotted. |
| Please list any positive or negative aspects of the evaluation methodology that you observed. | <p>Not sure exactly what is intended by “evaluation methodology.” I thought that the support, feedback, and general interaction with the development staff was quite good and provided a responsive means for SMEs to critique the UI and KE tools and for developers to clarify SME entry actions and intentions. If the Year 3 effort was considered a general test of the fungibility of the CYC technology, then I believe we accomplished that while adding some valuable KE capabilities.</p> | I enjoyed the whole project from start to finish | I think we managed to capture a method that is very detailed oriented with room to add details in the future. The level of the detail of the terrain |

| | | | |
|---|---|--|---|
| What expectations did you have coming into RKF Year Three? | My initial expectation was that the Year 3 effort would produce a coherent and useful knowledge base that focused on terrain assessment to identify danger areas for small unit actions. Admittedly, I typically over-scope an effort, but our end results fell short of what I thought was possible (for the reasons mentioned above). I also had hoped to be able to test the efficacy of an approach based on a generalized assessment of terrain, that is, by assessing regions and large features rather than individual pixels of terrain. Don't know if this is a pie-in-the-sky idea or if technology does not yet support such assessments. | Interface with the GIS data | evaluation would be hard to match using humans to-do the evaluation. Being new to RKF this year I came in with an open mind and not sure what to expect. I did expect that whatever we did it would, all or in some part, be used in the future for a real application for military use. |
| In what ways were these expectations met? | I think that we evolved the KE tools and elaborated a coherent body of assessment knowledge but ran out of time before we could accomplish sufficient knowledge entry to validate our concepts. | N/A | It would remain to be seen if the work done here will result in a future application. |
| In what ways were they not met? | Same as above. | Didn't see it happen | It would remain to be seen if the work done here will result in a future application. |
| How did your expectations evolve over the course of the project? | Like I said, I always over-scope an effort and have to adjust my expectations as a project evolves. As a former team chief explained to me after he and his team had finished a 6-months research effort: "we did not accomplish all that you proposed, but we achieved a lot more than we thought possible." Over the course of this effort, my expectations evolved from the belief that we would build a working prototype, to the expectation of having a proof-of-concept demonstration capability, to the hope that we would enter the basic structure of the OCOKA assessment methodology. We developed some sound foundations for knowledge entry but did not get far enough long on the actual knowledge base. | Yes. I think we got closer to a usable tool after the break. | At some point I expected to see a fully functioning analysis tool before the end of the project. |
| What aspect of the project did you find most satisfying? | Evolution of the knowledge entry methodology for the OCOKA flow charts, which I believe is a significant improvement in methods of entering complex concepts. Likewise, the close interaction with the development staff, which provided for responsive feedback at each step. | Being able to enter data | I enjoyed seeing the charts being turned into functioning queries. |
| What aspect of the project did you find most frustrating? | The frequent disruptions forced by the discussion of tangential or completely unrelated issues to the basic problem, which prolonged the activities of developing the basic terrain evaluation concepts, and the inability to achieve a clear understanding of what was available as GIS assessment products and involved in CYC-GIS interaction. | Interfaces | The late date in which the tools started being developed. I was hoping we would have gotten further before the end |

Any other comments?

When do we get to finish and test the work?

All team members were very professional and had a positive attitude.

of the funding/time.

I enjoyed the debates about the charts with the other SME's. I also liked the look in people's eyes here at Cycorp. There was a very visible excitement about the work being done and a passion for the advancement of the AI.

Appendix D: Terrain-Based Risk Assessment Knowledge Base

As described above, a Year Three goal was the development of a legacy knowledge base containing terrain knowledge that was SME-entered, using the RKF KRAKEN interfaces, to the greatest extent possible, and SME-vetted at minimum. This goal also required a reasoning challenge in which this terrain knowledge would be put to work, demonstrating that the SME-entered knowledge was sufficiently rich to enable real terrain reasoning. A considerable amount of time was spent at the outset of Year Three, discussing and selecting a reasoning challenge, and working to focus the task, identify required knowledge content and define component subtasks.

Identification of terrain-based risks was selected as the Year Three Challenge Problem. This requires both a certain breadth of knowledge regarding terrain characteristics and a depth of knowledge regarding the effects and interactions of those terrain characteristics with each other and with military forces. This challenge focused the task and enabled the SMEs to begin identifying the knowledge that would be required, and how it would need to be combined. The following document (authored by Dr. Kerry Hines) outlines both of these levels of knowledge, and how the resulting knowledge base should work to implement expert terrain analysis in an automated system, in a way that complements and enhances the tactical understanding of commanders and their staffs.

1. The purpose of the Terrain-Based Risk Assessment Knowledge Base is to assist commanders and staffs in identifying potential danger areas on the battlefield. Danger areas are locations on the battlefield where friendly or enemy forces could be vulnerable or be placed at risk by the opponent's ability to exploit the specific terrain characteristics and weather effects of those locations. Some examples include restrictive and close terrain, choke points, obstacles, terrain that naturally exposes a flank, and areas dominated by key terrain. Use of the Terrain-Based Risk Assessment Knowledge Base (TRAKB) should help tactical commanders and staffs to visualize the danger and devote more effort to the evaluation of measures to reduce the potential risk to their own forces or to exploit the potential opportunities the terrain may offer to place the opponent at risk.

2. In combat, each antagonist seeks to exploit terrain characteristics that put the opponent at a disadvantage. Each strives to identify positions that offer the best opportunity to engage, delay, and/or destroy the opponent. Typically, the positions that offer an advantage to one opponent are danger areas to the other opponent and include:

- Restrictive terrain that may slow an attacker, cause a separation of forces, create difficulties in command and control, or force the attacker to conduct defile drills (for example, narrow valleys, passes, or urban areas).
- Chokepoints or natural obstacles that may cause a loss of momentum, a potential fragmenting of forces, or a vulnerable concentration of an attacker's forces (for example, rivers and canals and urban or complex terrain)
- Terrain that canalizes attacking formations into areas that provide defending forces good fields of fire, observation, and flanking fires
- Terrain deficient in cover and concealment that exposes attacking formations to the defender's long-range observation and direct fields of fire

- Areas dominated by key or defensible terrain that allow the defender to mass fires against the attacking formations or the attacker to mass fires against the defender's positions.

3. Generally one opponent attacks and the other defends, although there are some situations when both opponents advance, as in a meeting engagement or a movement to contact. The terrain rarely favors one type of operation throughout the width and breadth of the battlefield. Within a given area certain sub-sectors will favor various operations to varying degrees. Typically, there are limited locations within an area of operations that provide both good fields of fire and adequate cover for a defender. Similarly, an attacking enemy will have only a limited selection of avenues of approach that provide adequate cover and concealment.

4. Basic data and initializing conditions are needed to establish the operational and battlefield framework for the evaluation of danger areas. Those data/conditions are depicted in Figure 37 and outlined below.

- Military force parameters. Unit identification (ID) -- including type and size -- and location of friendly and enemy forces, the assigned friendly force mission, and the assessed enemy force mission. Unit ID should provide force type and size, which are necessary for assessing suitability of mobility corridors (effect of obstacles and width of corridors) and the effective ranges of available weapons; unit locations determine the battlefield and directionality (orientation of the long axis of the avenues of approach [AAs]); and mission determines which force will attempt to move/maneuver along the AAs. These parameters also include data accessible from forces databases that include basic organization/subordination and weapons information.
- Weather parameters. Forecast period and forecast specifics for visibility, temperature and humidity, wind speed and direction, cloud cover. Forecast period can be related to expected accuracy of the forecast, visibility conditions affect observation.
- Temporal parameters. Expected start time and duration of the operation. Used to determine degree of natural illumination and the effect on visibility (observation). Can be combined with directionality of AAs to assess silhouetting effects.
- Operational parameters. Graphical control measures, boundaries and objectives. Define the specific area of operations (AO) and, along with force locations, help define the area of interest (AI). The AI will include areas occupied by enemy forces who could jeopardize the accomplishment of the mission. The expansion of the AO to the AI is based on the rules of thumb in Table 1. The forces data also includes basic rules-of-thumb for tactical activities, such as nominal widths and depths for deployed units of a given size.

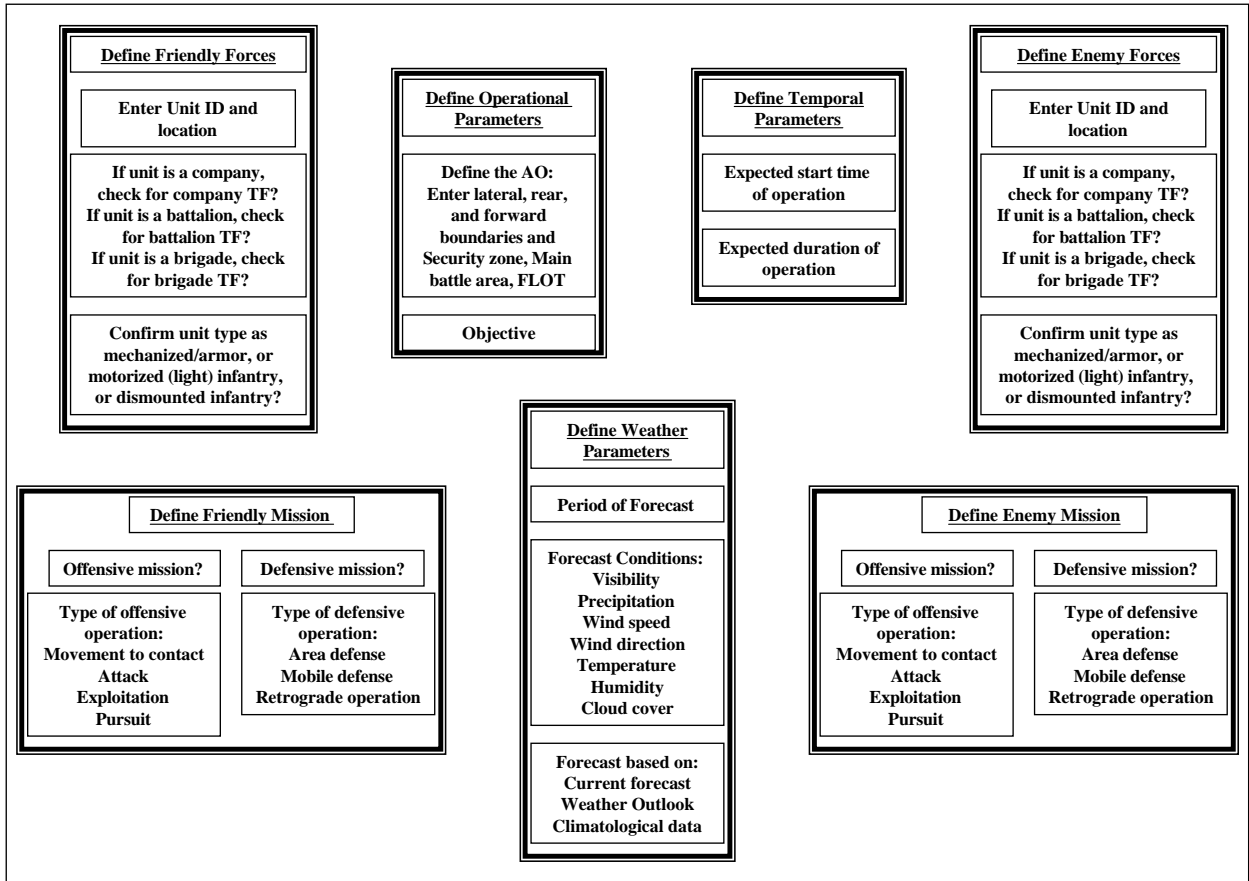


Figure 37. Initialization data

Table 1. Area of interest (AI) determination

| DEFENSE | | |
|----------------|--------------|---|
| ECHELON | TIME | DISTANCE |
| BN | Up to 12 hrs | Fwd: 15 km Flanks: 3-6 km |
| BDE | Up to 24 hrs | Fwd: 30 km Flanks: 6-10 km |
| DIV | Up to 72 hrs | Fwd: out to 100km Flanks: 20-30 km Rear: out to 30 km |
| OFFENSE | | |
| BN | Up to 12 hrs | Fwd: out to subsequent objective Flanks: 3-6 Km of axis or zone |
| BDE | Up to 24 hrs | Fwd: out to subsequent objective Flanks: 6-10 Km of axis or zone |
| DIV | Up to 72 hrs | Fwd: out to subsequent objective |

| | | |
|--|--|--|
| | | Flanks: 25-30 Km of axis or zone Rear: Out to 30 Km |
|--|--|--|

5. The main military aspects of terrain are used to identify and assess threats impacting military forces activities and to evaluate whether the terrain can accommodate unit capabilities and mission demands:¹⁸. If the military aspects of the terrain for a given location satisfy the tactical criteria as outlined in Tables 2 and 3 below, then the location is nominated as a potential danger area.

- *Observation and fields of fire.* Threats associated with observation and fields of fire usually involve when the enemy will be able to engage a friendly unit and when the friendly unit's weapon capabilities allow it to engage the enemy effectively. The considerations for evaluating these aspects of terrain are depicted in Charts 1 and 2.
- *Cover and concealment.* Threats associated with cover and concealment are created either by failure to use cover and concealment or by the enemy's use of cover and concealment to protect his assets from observation and fire. The considerations for evaluating these aspects of terrain are depicted in Charts 3 and 4.
- *Obstacles.* Threats associated with obstacles may be caused by natural conditions (such as rivers or swamps) or man-made conditions (such as minefields or builtup areas [e.g., urban terrain]). The considerations for evaluating this aspect of terrain are depicted in Chart 5.
- *Key terrain.* Threats associated with key terrain result when the enemy controls that terrain or denies its use to the friendly forces. The considerations for evaluating this aspect of terrain are depicted in Chart 6.
- *Avenues of approach.* Threats associated with avenues of approach include conditions in which an avenue of approach impedes deployment of friendly combat power or conditions that support deployment of enemy combat power.

Table 2. Defense tactical terrain evaluation criteria

| DEFENSE TACTICAL CRITERIA | TERRAIN EVALUATION ASPECT |
|--|---|
| Controls likely attacker avenue of approach (AA) | Elevated defensible terrain; proximity to AA |
| Engages attacker where his movement is most canalized; Facilitates maximum effect on attacker with least force | Choke point; RESTRICTED terrain; Natural obstacle; complex terrain |
| Provides long-range direct fires of attacker | Observation and fields of fire from defensible terrain; Lack of cover and concealment along AA |
| Allows massing of direct and indirect fires on the attacker | Overlapping observation and flanking fields of fire of AA segment |
| Minimizes likelihood of unintended decisive engagement; Maintains freedom of maneuver and disengagement of defender | Covered and concealed ingress/egress and lateral routes; Routes leading away from direction of attacker |

¹⁸Field Manual No. 3-100-12, p. II-9

Table 3. Offense tactical terrain evaluation criteria

| OFFENSE TACTICAL CRITERIA | TERRAIN EVALUATION ASPECT |
|--|--|
| Controls likely attacker/counterattacker avenue of approach (AA) | Elevated defensible terrain; proximity to AA |
| Maintains freedom of maneuver and momentum of attacker | Observation and fields of fire from defensible terrain dominating choke point, RESTRICTED terrain, natural obstacle, or complex terrain |
| Engages defender where his deployment is most restricted; Facilitates maximum effect on defender with least force | Lack of obstacles to defender front or flank, Restricted or gap in observation and fields of fire from BP; Covered and concealed approach and lateral routes for attacker; |
| Provides long-range direct fires of defender | Observation and fields of fire from defensible terrain; Lack of cover and concealment within BP |
| Allows massing of direct and indirect fires on the defender | Overlapping observation and flanking fields of fire of defender's BP and counterattack AA |
| Increases likelihood of intended decisive engagement; Limited freedom of maneuver of defender | Lack of cover and concealment on defender's route of withdrawal from BP |

6. Evaluation of the terrain's effects on military operations supports conclusions about the places on the battlefield best suited for use as engagement areas; battle positions; defensible terrain (including the attacker's overwatch positions and support-by-fire or attack-by-fire positions); infiltration lanes; and sites/positions for specific system or assets.

- Battle positions: Identify concealed and covered positions that offer observation and fields of fire into potential engagement areas. If the TRAKB User is defending, the identified locations are potential defensive positions. If the TRAKB User is attacking, the positions provide a start point for determining possible courses of action (COAs) that the opponent might adopt. Also, attacking forces might use identified positions to support the assault of defensive positions, block the defender's counterattacks, or support their advance.
- Engagement areas and ambush sites: Using the results of evaluating concealment and cover for the attacker along likely AAs, identify areas where maneuvering forces are vulnerable to fires from potential battle positions. Duration of vulnerability is a factor of the defender's weapon ranges and the likely speed of maneuvering forces. If the TRAKB User is planning an attack, these are areas where the attacking forces will be vulnerable to the defender's fires. If the TRAKB User is assigned a mission to defend, these are potential engagement areas for his forces.
- Infiltration lanes: Identify routes along which an attacking force can conduct undetected movement (mounted or dismounted) through or into an area occupied by the opponent's forces and secure a position of advantage in the defender's rear while exposing only small elements to the defender's fires. Typically, forces infiltrate in small groups and reassemble to continue their mission.

- Key terrain: Identify areas or terrain features that permit or deny movement or maneuver. The seizure, retention, or control of the area or feature affords a marked advantage to either combatant. These areas will usually be key terrain because of their potential use as a combat position in support of attacker's intended objective or by the defender to deny the attacker his intended objective.

Table 4. Battlefield tactical terrain use

| SECURITY ZONE | | CLOSE BATTLE AREA | |
|---------------------------------------|----------------------|--------------------------|-------------------|
| Attacker | Defender | Attacker | Defender |
| Observation Position | Observation Position | Support-by-Fire Position | Battle Position |
| Overwatch Position | Ambush Site | Attack-by-Fire Position | Strong Point |
| Engagement Area/ Blocking Position | Delay Position | Infiltration Lanes | Blocking Position |
| | Engagement Area | Attack Position | Engagement Area |

7. The Terrain-Based Risk Assessment evaluation process requires the input of the parameters outlined in paragraph 4, above, plus the potential avenues of approach within the AO. The evaluation process draws on the terrain considerations in the KB to create queries for a supporting Geographic Information System concerning the qualitative suitability of a specific location/position/area for a specific combat activity. Interactions among the evaluation routines for each military aspect of terrain and between TRAKB and the GIS, required inputs to each evaluation routine, and the expected output from each routine are outlined in Table 5.

- Each AA is evaluated for observation and fields of fire to identify defensible terrain that will provide potential observation and battle positions and potential engagement areas for the defender to delay, block, or destroy the attacker or that will provide potential overwatch, attack-by-fire, or support-by-fire positions for the attacker. The outputs from this evaluation step are a list of locations within defined spatial relationships to an AA where attacking (counterattacking) maneuver forces are most vulnerable to the defender's observation and fires and a list of locations within defined spatial relationships to an AA and defender's BP where attacking maneuver forces can realize the greatest advantage over the defender's observation and fires.
- The observation and fields of fire nominated defensive terrain is evaluated for cover and concealment to determine if the type and size units on the battlefield will be protected from the opponent's observation and fire while executing contemplated functions. The output is a list of nominated terrain locations that are likely positions for the defender or attacker to occupy in the course of executing their assigned tasks with an annotation for each location as to the activity for which it is best suited and the danger or advantage that would accrue to each opponent if the location were used in the assessed manner.
- The refined defensive terrain nominations are correlated with choke points, restricted and complex terrain, and other obstacle locations where the

attacker's movement and maneuver is constrained or those locations where the movement and maneuver of a counterattack or reserve force is most constrained. The output is a correlated list of potential locations where the attacking and defending forces could be exposed to the greatest danger, or conversely, locations where each antagonist may gain the greatest advantage over the other through the capability to exploit the specific opportunities offered by the terrain characteristics of the nominated locations.

- Each of the potential danger areas on the final list of nominations is evaluated for the relative advantage it affords the attacker and defender. Those areas deemed to afford marked advantage to either combatant are nominated as key terrain, and any area deemed to offer one combatant the opportunity to deny the opponent mission accomplishment is nominated as decisive terrain.

Table 5. Functions, interactions, and handoffs

| CHART | INPUT DATA | GIS QUERY | INTERMEDIATE PRODUCT | OUTPUT & DISPOSITION |
|----------------|---|---|---|---|
| Observation | Forces, locations, and missions; Weather visibility and precipitation forecasts and natural light data: AO boundaries; Objective for attacking force; Likely AAs. | Route Visibility - Identifies areas where a route is most vulnerable to attack. Region Visibility - Identifies locations that can see the maximum portion of a region or areas that are masked from the region. | AI definition; Visibility reduction factors; Intervisibility between likely AAs and potential defensive terrain along AAs and within defender's BPs | Potential observation and/or delay positions in Security Zone; Potential positions to support assault/attack of objective; |
| Fields of Fire | Forces (weapons) and locations,; Weather/light visibility reduction factors: AO & AI boundaries; Likely AAs. | Fields of Fire - Identifies areas that are good, fair, poor, and unsuitable for fields of fire. Good areas have little to no terrain and vegetative effects on line of sight for both enemy and friendly units. Fair, poor, and unsuitable areas are based on high degrees of vegetation density and slope. | | |
| Cover | | Cover product identifies areas with good, fair, and poor protection from enemy fire. Cover is based on vegetation density and percent slope. Good areas of cover have high slope and/or dense vegetation. | | |
| Concealment | | | | |
| Obstacles | | Linear Obstacles product shows linear terrain features that form natural | | |

| | | | | |
|-------------|--|---|--|--|
| | | obstacles within an area of interest. Obstacles include escarpments, embankments, road cuts and fills, depressions, fences, walls, hedgerows, pipelines, bluffs, dragon teeth, and moats. | | |
| Key Terrain | | | | |

The output from the Terrain-Based Risk Assessment process are nominations of locations where the intersection of military aspects of terrain and effects of terrain indicate that one opponent can likely gain a significant advantage over the other by exploiting the opportunities that the terrain offers at those locations. Once the tactical terrain effects are evaluated, assessment of the degree of danger or duration of vulnerability requires consideration of the speed of movement of forces, the protective qualities of their equipment (armor), and the ranges of weapon and sensor systems that would pass through or be located on the potential danger areas.

Appendix E: Glossary of Terms and Abbreviations

ADT

Analysis Diagram Tool, a user interface that can be used to represent an intelligence analysis process graphically.

AI

Artificial Intelligence — Making computers do things that people don't think computers can do; making computers like the ones in movies.

AIAI

The Artificial Intelligence Applications Institute at the University of Edinburgh.

anaphora

A backward reference to a concept referred to earlier. Typically both the reference and the referent can be associated with an explicit word or phrase, but they can be implicit. Common examples of anaphora are pronouns (e.g. "it") and definite references (e.g. "that document").

cataphora

Similar to anaphora, but a forward reference. For example, "There is one thing that must be kept in mind: Always do you your best."

CBRNE

Chemical Biological Radiological Nuclear and (High-) Explosive — a shorthand for certain categories of threat.

COA

Course of Action.

DARPA

Defense Advanced Research Projects Agency.
<http://www.darpa.mil/>

DHTML

Dynamic HTML — an inclusive term referring to various ways to implement user interfaces using HTML and JavaScript.

EKCP

Expert Knowledge Challenge Problem — An evaluation based on the transfer of SME knowledge.

ESRI

Founded as Environmental Systems Research Institute, a world-leader in GIS technology.
<http://www.esri.com/>

Factivore

A KRAKEN component that is a template-based knowledge-entry tool that allows the user to enter knowledge rapidly, and without interruption.

forward rule

Rules in the Cyc knowledge base that are executed eagerly. Whenever a new assertion is made, all existing forward rules are checked for applicability.

GAF

Ground Atomic Formula, the simplest sort of fact asserted in the Cyc knowledge base.

GIS

Geospatial (or sometimes Geographic) Information System — software that performs quantitative analysis of terrain information.

HPKB

High-Performance Knowledge Base — A predecessor project to RKF.

instance/type confusion

The Cyc ontology makes a careful distinction between whether something is a member (instance) of a collection, or a more specific type of collection. For example, a poodle is a specific type of dog, whereas Rover might be an individual dog. The two concepts are often conflated in English.

ISI

Information Sciences Institute at the USC Schools of Engineering.
<http://www.isi.edu/>

KB

Knowledge base – Cyc’s repository of assertion — including definitions, rules, etc. — over which inference operates.

KE

Knowledge Entry — the general practice or theory of entering knowledge into the knowledge base of an artificial intelligence system. Primarily used of SME knowledge entry.

The term is also used to refer to the file format — essentially CycL — that is used by ontologists to prepare knowledge for batch entry.

KRAKEN

Knowledge-Rich Acquisition of Knowledge from Experts who are Non-logicians — the system of components developed by Cycorp and collaborators around Cyc for the RKF project to permit SME knowledge entry.

METT-T

Mission, Enemy, Terrain and weather, Troops and support, and Time — military acronym for COA evaluation. Often seen as METT-TC, with the addition of Civil considerations.

microtheory

A subdivision of Cyc’s knowledge base. The assertions in a microtheory relate to a specific domain or context.

mixed-initiative dialog

Interaction between the user and a software system where each side can take the initiative in suggesting what direction the conversation might take. For example, the user may indicate a

desire talk about a specific concept, and the system might seek clarification or ask relevant questions.

NLP

Natural Language Processing — A general term for technology that either parses a natural language (such as English) into a formal representation (such as CycL), or generates natural language paraphrases from a formal representation.

Some authors use NLP to refer to Natural Language Parsing only.

Northwestern University

<http://www.northwestern.edu/>

OCOKA

Observation and fields of fire, Cover and concealment, Obstacles, Key terrain, Avenues of Approach — military acronym for assessment of terrain.

OE

Ontological Engineering or Ontological Engineer — the former is the work done by an ontologist, the latter is a synonym for an ontologist.

ontologist

Someone skilled at communicating with a knowledge-based artificial intelligence system.

peg

A word or phrase that could potentially be a discourse referent (see *anaphora*). The term was used by Susan Luperfoy in her 1992 paper *The Representation of Multimodal User Interface Dialogues Using Discourse Pegs*.

Query Library

A KRAKEN component that allows the user to construct and ask queries.

SAIC

Science Applications International Corporation.

<http://www.saic.com/>

Salient Descriptor

An inference-based tool that induces questions that may be interesting and relevant for the user to answer with respect to the conversational focus.

SCOOP

Collaboration system developed by Teknowledge.

SME

Subject Matter Expert — someone attempting to perform knowledge entry or retrieval with an artificial intelligence system who is not expert in using such systems, but has expertise in a specific knowledge domain.

Also used by NWU to refer to the Structure Mapping Engine.

TKCP

Textbook Knowledge Challenge Problem — an evaluation based on a system using knowledge from a well defined corpus.

TMS

Truth Maintenance System — A Cyc sub-system that ensures that any deduced facts are tied back to their supporting facts, and can be re-evaluated when the supports are changed.

UIA

User Interaction Agenda — the primary KRAKEN interface in Year One and Year Two, also used in Year Three, and in ongoing projects. Supports mixed-initiative dialog (qv).

WFF

Well Formed Formula — Every assertion entered into Cyc undergoes strict syntactic and semantic checking to eliminate many types of nonsensical expressions. This gatekeeper is somewhat strict.