A Question-Answering System for AP Chemistry: Assessing KR&R Technologies†

Ken Barker[1], Vinay K. Chaudhri[2], Shaw Yi Chaw[1], Peter E. Clark[3], James Fan[1], David Israel[2], Sunil Mishra[2], Bruce Porter[1], Pedro Romero[4], Dan Tecuci[1], Peter Yeh[1]

[1]University of Texas at Austin, [2]SRI International, [3]Boeing Phantom Works, [4]Indiana University-Purdue University Indianapolis
[1]{kbarker, jchaw, jfan, porter, tecuci, pzyeh}@cs.utexas.edu
[2]{chaudhri, israel, smishra}@ai.sri.com
[3]peter.e.clark@boeing.com
[4]promero@iupui.edu

Abstract

Basic research in knowledge representation and reasoning (KR&R) has steadily advanced over the years, but it has been difficult to assess the capability of fielded systems derived from this research. In this paper, we present a knowledge-based question-answering system that we developed as part of a broader effort by Vulcan Inc. to assess KR&R technologies, and the result of its assessment. The challenge problem presented significant new challenges for knowledge representation, compared with earlier such assessments, due to the wide variability of question types that the system was expected to answer. Our solution integrated several modern KR&R technologies, in particular semantically well-defined frame systems, automatic classification methods, reusable ontologies, a methodology for knowledge base construction, and a novel extension of methods for explanation generation. The resulting system exhibited high performance, achieving scores for both accuracy and explanation which were comparable to human performance on similar tests. While there are qualifications to this result, it is a significant achievement and an informative data point about the state of the art in KR&R, and reflects significant progress by the field.

Introduction

Basic research in knowledge representation and reasoning (KR&R) has steadily advanced over the years, but it has been difficult to assess the capability of fielded systems derived from this research. Few systems are built, and even fewer are systematically evaluated in domains for which the criteria for success are clear-cut. To obtain a better idea of the state of the art of one type of KR&R technology, Vulcan Inc. recently conducted the Halo Pilot Project, the first phase of a multiphase effort to create a "Digital Aristotle", an expert tutor for a wide variety of subjects. The Halo Pilot was a 6-month effort intended to assess technology for question-answering systems, with an emphasis on deep reasoning, structured around the challenge of answering variants of Advanced Placement[1] (AP) chemistry questions for a portion of the syllabus.

In this paper, we present the question-answering system that we developed as part of this project, and the results of its evaluation. The system we developed used a combination of several modern KR&R technologies, in particular semantically well-defined frame systems, automatic classification methods, reusable ontologies, and a methodology for knowledge base construction. In addition, we extended existing explanation generation methods; allowing the system to produce high-quality, English explanations of its reasoning.

The challenge problem itself tested new limits of KR&R technology, making the assessment unlike any previous one that we are aware of. First, because of the huge variety in AP question types, the domain requires a truly "multifunctional" solution, that is, the ability of the system to combine information together in novel, unanticipated ways. This requirement is in contrast to early Expert Systems, which were often built to answer just one or two types of questions. Second, there was substantial emphasis placed on the explanation of the system's reasoning, demanding that the system have significant explanatory as well as inferential capabilities. Third, the evaluation was rigorous and conducted against human standards, lending credibility to the results, while raising some interesting evaluation issues.

In the evaluation, the final system we developed exhibited impressive performance in the domain, in terms of both correctness and explanation, as judged by domain experts. While there are qualifications to this result, it is significant because it demonstrates that a novel combination of modern KR&R technologies can rapidly

† Full support for this research was provided by Vulcan Inc. as part of Project Halo

[1] In the U.S., high school seniors may opt to take Advanced Placement exams in various subjects to earn credit for entry-level college courses.

produce a high-performance system for this new type of challenge, reflecting progress by the field as well as successful exploitation and extension of that technology. In addition, the analysis of what did not work is of interest and a significant contribution of this paper in its own right, as it provides some useful insights into directions for future research.

The Challenge Task

The domain chosen for the Pilot was a subset of AP chemistry, namely stoichiometry and equilibrium reactions, spanning about 70 pages of a college-level chemistry textbook (Brown et al. 2003). This domain was chosen to assess several important features of KR&R technology without taking on the entire problem of AI. In particular, this domain requires complex, deep reasoning, but avoids some areas of KR&R, such as reasoning with uncertainty and spatial reasoning. It presented significant new challenges for knowledge representation due to the variability of question types that occur in AP examinations. This variability made it infeasible for us to anticipate all possible question types at the time we built the system. Instead, our system needed to be able to combine its knowledge in new and potentially unanticipated ways in order to answer many of the questions. To give a sense of the wide variety of questions, here are some drawn from the exam used to evaluate our system: "Which of the following compounds will produce a gas when hydrochloric acid is added to the solid compound?" (Choices given); "The spectator ions in the reaction of barium nitrate with sodium sulfate are what?" (Choices given); "In a 100.0 milliliter sample of a different buffer solution, the propanoic acid concentration is 0.35-molar and the sodium propanoate concentration is 0.50-molar. To this buffer solution, solid NaOH is added. Would the pH increase or decrease?"; "Explain why a solution of $HClO_4$ and $NaClO_4$ cannot act as a buffer solution"; "Determine the equilibrium constant for the reaction $H_3C_6H_5O_7 + 3\ H_2O \leftrightarrows 3\ H_3O^+ + C_6H_5O_7^{3-}$."; and so on. This variability in question type continues throughout all example question sets we studied, and also in the set used in the final evaluation. It was not clear at the outset that a system could be constructed to perform competently and robustly at tasks of this sort.

Technical Approach

Overview

To address this challenge, we developed a system based on Shaken, a knowledge capture, representation, and reasoning system (Clark et al. 2001). Our final system used a combination of KR&R technologies, principally automatic classification, backward inference, explanation technology and a semantically clean frame-based system. In addition, the underlying knowledge base was built on top of a generic (reusable) ontology, rather than built from scratch. A representative example of the kind of answer and explanation the final system produced in the final exam is shown in Figure 1. Questions are posed in the

QUESTION: Sodium azide is used in air bags to rapidly produce gas to inflate the bag. The products of the decomposition reaction are:

a. Na and water

b. Ammonia and sodium metal

c. N_2 and O_2

d. Sodium and nitrogen gas

e. Sodium oxide and nitrogen gas

ANSWER (system generated):

- What are the products of the given decomposition reaction? (1)
  - By definition, oxidation-reduction reactions occur when electrons are transferred from the atom that is oxidized to the atom that is reduced. We need to look for changes in the oxidation states of the elements in the reaction. (2)
    - In the reactants, the oxidation state(s) of the element Na is/are (1). In the product, the oxidation state(s) is/are (0) (3)
    - Therefore, the reaction causes a change in oxidation state. (4)
  - Therefore, this is an oxidation reduction reaction. (5)
  - By definition, a Binary Ionic-Compound Decomposition Reaction occurs when a binary ionic compound is heated. (6)
  - Therefore, this reaction is a Binary-Ionic Compound Decomposition reaction. (7)
  - In general, a Binary Ionic-Compound Decomposition Reaction converts a binary ionic-compound into basic elements (8)
  - In this reaction, NaN_3 reacts to produce Na and N_2. (9)
- The products of the decomposition reaction are: (d) Sodium and nitrogen gas. (10)

Figure 1: A question from the Halo Pilot Exam and the answer/explanation produced by our system

system's underlying KR language, not English (NLP was outside the scope of the Pilot); however, the system produced answers and explanations in English as illustrated.

Inference

The knowledge base was encoded in the language KM (the Knowledge Machine), a frame-based knowledge representation language with clear, first-order logic semantics (Clark and Porter 1999), and somewhat similar in spirit to KRL (Bobrow and Winograd 1977). In KM, a frame denotes either a class (type) or an instance (individual), and classes are organized into a semi-lattice. Frames have slots (binary predicates) whose fillers are axioms about the slot's value (the predicate's second argument, given the first). These axioms have both declarative and procedural semantics, allowing inference to proceed.

Our system answers questions by interleaving two fundamental types of reasoning:

- *Automatic classification* introduces new concepts into the scenario. This is a form of terminological reasoning commonly used in description logic (Brachman and Schmolze 1985).
- *Backward chaining* applies problem solving methods to compute bindings for the query variables given in the original question or the variables introduced by previous steps of backward chaining.

Both types of reasoning contribute to the answer and explanation. Explanation steps (2-5) and (6-7) in Figure 1 correspond to steps of automatic classification that augment the scenario description with new terms. Steps (2-5) reflect the classification of the given reaction as an oxidation-reduction reaction, and steps (6-7) reflect two new concepts: the air bag's sodium azide is a binary ionic compound, and the resulting reaction is therefore a binary ionic compound decomposition reaction. This interleaving of backward chaining and classification turned out to be critical, as classification allowed the system to realize when new knowledge could be brought to bear on a problem. For example, only as a result of classifying the reaction as a Binary-Ionic Compound Decomposition reaction (steps (6-7)) is the system then able to apply knowledge of how to compute the products of the reaction (steps (8-9)).

Structure of the Knowledge Base

The two types of reasoning correspond to the two principal knowledge structures in the system's knowledge base: chemistry terms and laws. Chemistry terms are concept definitions, such as the term "binary ionic compound" used to answer the question above. Chemistry laws are problem solving methods and knowledge about how and when to use them. Laws are represented in four parts, using four main predicates (slots), each playing a role in backward chaining inference. The `context` slot is a structured representation consisting of instances, functions, variables, and relations. Its role is to describe the conditions under which the law applies. The `input` is a subset of the variables of the `context` – just those variables that must be bound for the law to apply. The `output` is that subset of variables of the `context` that will be bound by the use of the law. Finally, the `method` is one or more axioms that declaratively specify the relationship between the `input` and `output` variables. Procedurally it can be used to compute values for the `output` variables given values for the `input` variables. An informal description of the content of the law for computing the concentration of a solute is shown in Figure 2. Actual laws were encoded in KM.

```
Concentration of Solute Law
  Context:
    a mixture M such that:
      volume(M) = V liters
      has-part(M) =
        Chemical C such that:
          quantity(C) = Q moles
          concentration(C) = Conc molar
  Input:  V, Q
  Output: Conc
  Method: Conc ← Q/V
```

Figure 2: An informal (yet faithful) representation of the content of a law in our system

Reusing Knowledge Content

Another important characteristic of our solution was that we were able to leverage pre-existing, general knowledge content for this task, rather than starting from scratch. For several years we have been building a library of representations of generic entities, events, and roles (Clark and Porter 1997, Barker, Clark and Porter 2001, Clark et al. 2001, Thoméré et al. 2002) and we were able to reuse parts of this for the Halo Pilot. We estimate that 10-20% of the final chemistry knowledge base was reused content and that much of the more general knowledge encoded for the Halo Pilot will in turn be reusable in future domains.

In addition to providing the types of information commonly found in ontologies (class-subclass relations and instance-level predicates), our representations include sets of axioms for reasoning about instances of these classes. The portion of the ontology dealing with properties and values was especially useful for the Halo Pilot. Our Properties and Units of Measure ontology includes representations for numerous dimensions (e.g., capacity, density, duration, frequency, quantity) and values of three types: scalars, cardinals, and categoricals. The ontology also includes methods for converting among units of measurement (Novak 1995), which our system used to align the representation of questions with representations of terms and laws, even if they are expressed with different units of measurement.

Explanation Generation

In the evaluation of the system, the quality of the English explanations was weighted as heavily in grading as the correctness of answers. (Students taking the AP exam do not have this requirement). Taking the lesson from early Expert System applications that proof trees or inference traces are not comprehensible to users, we built into our system mechanisms for controlling which parts of the proof tree are exposed to the user, and text generation techniques to summarize those parts in a concise and readable form.

During reasoning, KM records the rules (KM expressions) used to derive every ground fact. This produces a database of "proof tree fragments" (the explanation database), the key raw material for explanations. Simply reciting the rules used in a derivation still produces explanations containing too much detail of uninteresting inferencing (e.g., unification, breaking up conjunctive expressions, handling variables, performing unit conversion).

To produce more appropriate explanations, KM allows the knowledge engineer to specify which proof tree fragments should be used and to author text templates to produce coherent English paraphrases for those fragments. These features allow the engineer to specify supporting facts that merit further explanation and allow complete control over the resulting English text. The supporting elements for the explanation of a fact need not even be the same as the elements used in the derivation of the fact.

This new approach proved effective for the Halo project, but also has some disadvantages. First, it is more labor intensive for the knowledge engineer, as it requires her to write many rules twice: once as formal KM, and once as an informal paraphrase together with a supporting fact list. Second, there is currently no mechanism in KM to ensure that the informal rendition is faithful to the formal one, requiring that the knowledge engineer ensure that the paraphrase and supporting facts genuinely reflect what the formal KM is computing. Nonetheless, even under the strict time constraints of the Halo Pilot, we were able to encode both the formal KM and the explanation structures to produce many explanations that were correct and appropriate.

For each of the chemistry laws and terms, we identified the facts that would require explanation. For each of these facts, the engineer tagged the fact and provided an explanation frame containing three pieces of information: an "entry text" template, an "exit text" template, and a list of the more specific facts on which the current explanation is dependent. The explanation of dependent facts is recursive and appears nested between the current fact's entry and exit text. Figure 3 shows an example law with text generation template.

The explanation of the concentration of ions for strong electrolytes can be requested explicitly by a user, or may be triggered automatically if *I* (the ion-concentration in Figure 3) appears as a dependent fact in some other explanation. Figure 4 shows the explanation generated for the output of Compute-Concentration-of-Ions when the input is NaOH.

```
Compute Concentration of Ions Law
  Context:
    a Chemical C such that:
      electrolyte-status(C) = E
      ion-concentration(C) = I molar
  Input:   E
  Output:  I
  Method:  if C is a strong electrolyte
              I ← max          [I-expl-1]
           else ...            [I-expl-2]

Explanation Frame [I-expl-1]
  Entry Text:
   "If a solute is a strong electrolyte,
   the concentration of ions is maximal"
  Exit Text:
   "The concentration of ions in" C
   "is" I
  Dependent Facts:
   E
```

Figure 3: An example law with explanation tags and one of the corresponding explanation frames

- If a solute is a strong electrolyte, the concentration of ions is maximal
  - Checking the electrolyte status of NaOH.
    - Strong acids and bases are strong electrolytes.
    - NaOH is a strong base and is therefore a strong electrolyte.
  - NaOH is thus a strong electrolyte.
- The concentration of ions in NaOH is 1.00 molar.

Figure 4: The explanation generated for an application of the law in Figure 3

Note that the text templates and list of dependent facts in an explanation frame may contain arbitrarily complex KM expressions. In the Halo Pilot, we exploited the full power of this mechanism to produce fairly sophisticated explanations.

Knowledge Base Construction Methodology

In 4 months, we built the knowledge base through the coordinated efforts of four groups of experts. First, ontological engineers (working about 4 person-months) designed representations to support chemistry content, including the basic structure for terms and laws, chemical equations, reactions, and solutions. Second, chemists (working about 6 person-months) consolidated the domain knowledge into a 35-page compendium of terms and laws summarizing the relevant material from 70 pages of the textbook. Third, knowledge engineers (working about 15

person-months) encoded that knowledge in KM, creating representations of about 150 chemistry laws and 65 terms. The laws and terms translated to many more knowledge base rules (sometimes as many as 40 rules per law) and roughly 500 concepts and relations. While building this knowledge base, the knowledge engineers compiled a large suite of test cases for individual concepts and rules as well as for combinations of them. This test suite was run daily. Finally, the "explanation engineer" (working about 3 person-months) augmented the representation of terms and laws to generate English explanations.

An End-to-End Example

In this section we present an example of the knowledge engineering effort for the knowledge that was used in answering a single question on the Halo Pilot exam. The actual knowledge base was not built by analyzing one question at a time.

Figure 5 shows three of the chemistry laws as defined by our chemists (guided by the chemistry textbook and an informal analysis of previous exams).

L.6) Solubility of ionic compounds: Given an ionic compound, its solubility (soluble/insoluble) can be determined using the guidelines in Table 4.1 in the book. (By definition, all ions are soluble).

L.7) Precipitate formation: A precipitate forms when at least one of the products of a reaction is insoluble.

L.9) Chemical reaction of two ionic compounds (metathesis reaction): The formulas for the products of the reaction of two ionic reactants are determined as follows: The cation of one reactant (if any) is combined with the anion of the other reactant (if any) to form one product, and vice-versa for the second product.

Figure 5: Some chemistry laws defined by our chemists

```
Metathesis Reaction
  definition:
    any Reaction R such that:
      raw-material(R) = Ionic-Compound I1,
                        Ionic-Compound I2
  result:
    Ionic-Compound I3 such that:
      has-part(I3) = the cation part of I1,
                     the anion part of I2
    Ionic-Compound I4 such that:
      has-part(I3) = the cation part of I2,
                     the anion part of I1

Precipitate
  definition:
    any Chemical C such that:
      C = result-of(some Reaction)
      solubility(C) = insoluble
```

Figure 6: An informal rendering of some of the knowledge encoded to solve the sample question

In the next step the knowledge engineers encoded the chemistry terms and laws by adding KM concepts, rules and procedure-like methods to the knowledge base. The encodings also made use of the description-logic-style automatic classification rules (called "definitions" in KM) described earlier. Some of the knowledge corresponding to the chemistry laws from Figure 5 appears in an informal, simplified (but faithful) representation in Figure 6. The KM encodings for all chemistry knowledge, explanation templates and exam questions are available on the Halo Pilot Project web site (http://www.projecthalo.com).

The final step involved attaching explanation generation templates to knowledge base rules. Some of these explanation templates are shown in Figure 7.

The knowledge was then used to answer questions on the final Halo Pilot Project exam. Multiple-choice question 9 (MC9) from the exam describes a reaction of nickel nitrate and sodium hydroxide and asks for the result of the

```
Explanation Frame [Metathesis Reaction definition]
 Entry Text: "By definition, a reaction involving ionic reactants is a metathesis reaction"
 Exit Text:  "Therefore, this reaction is a metathesis reaction"
 Dependent Facts: none

Explanation Frame [Metathesis Reaction result]
 Entry Text: "In a metathesis reaction, the cation of each reactant is combined with
              the anion of the other reactant"
 Exit Text:  "The products of a metathesis reaction of" raw-material(R) "are thus"
             result(R)
 Dependent Facts: none

Explanation Frame [Precipitate definition]
 Entry Text: "By definition, the result of a reaction is a precipitate if it is
              insoluble in water"
 Exit Text:  "Therefore," C "is a precipitate"
 Dependent Facts: solubility(C)
```

Figure 7: An informal rendering of some of the explanation templates for the knowledge in Figure 6

```
MC9
  context:
    a Reaction R such that:
      raw-material(R) = nickel nitrate, sodium hydroxide
      result(R) = ?RR

  output:
    if ?RR does not include a Precipitate
       then "a. A precipitate will not form"
    if ?RR includes sodium nitrate and sodium nitrate is a Precipitate
       then "b. A precipitate of sodium nitrate will be produced"
    ...

Explanation Frame [MC9 output]
  Entry Text: "A solution of nickel nitrate and sodium hydroxide are mixed together.
               Which of the given statements is true?"
  Exit Text:  "The following statement is true:" output(MC9)
  Dependent Facts: definition(R), result(R), definition(result(R))
```

Figure 8: An informal rendering of multiple-choice question 9

reaction. A simplified rendering of the KM encoding for this question appears in Figure 8. The knowledge base handles the translation between chemical names and chemical formulae.

All questions on the exam were posed in the same way: query the `output` slot of the question, then generate the explanation text for the `output` slot of the question. When the explanation is requested for the `output` of MC9, the entry text is printed, then any explanation text associated with the three dependent facts is generated, then the exit text is printed. The system is able to classify the MC9 reaction automatically as a metathesis reaction, since the raw-materials can be proven to be ionic compounds. The first dependent facts then trigger the

QUESTION: A solution of nickel nitrate and sodium hydroxide are mixed together. Which of the following statements is true?

a. A precipitate will not form

b. A precipitate of sodium nitrate will be produced

c. Nickel hydroxide and sodium nitrate will be produced

d. Nickel hydroxide will precipitate

e. Hydrogen gas is produced from the sodium hydroxide

ANSWER (system generated):

- d. Nickel hydroxide will precipitate.

- A solution of nickel nitrate and sodium hydroxide are mixed together. Which of the given statements is true?
  - By definition, a reaction involving ionic reactants is a metathesis reaction.
  - Therefore, this reaction is a metathesis reaction.
  - In a metathesis reaction, the cation of each reactant is combined with the anion of the other reactant.
  - The products of a metathesis reaction of $Ni(NO3)_2$ and NaOH are thus $Ni(OH)_2$ and $NaNO_3$.
  - By definition, the result of a reaction is a precipitate if it is insoluble in water.
    - $Ni(OH)_2$ contains Ni^{2+} and OH^-.
      - According to Table 4.1 of Brown, Lemay and Bursten (2003), an Ionic Compound containing Ni^{2+} and OH^- is insoluble in water.
    - Therefore, $Ni(OH)_2$ is insoluble in water.
  - Therefore, $Ni(OH)_2$ is a precipitate.
- The following statement is true: d. Nickel hydroxide will precipitate.

Figure 9: The output of multiple-choice question 9 from the Halo Pilot exam

explanation of [Metathesis Reaction definition] and [Metathesis Reaction result] as defined in Figure 7. The third dependent fact from [MC9 output] will only trigger the [Precipitate definition] explanation for reaction results that get automatically classified as Precipitate. Finally, the explanation template for the dependent fact for [Precipitate definition] is also triggered, resulting in a nested explanation of the solubility of the Precipitate result of the reaction. The full output for question MC9 appears in Figure 9.

Evaluation

Methodology

To assess the system (and two other systems submitted by other teams), Vulcan set an exam consisting of a wide variety of new, unseen, AP-like questions. The systems' answers and explanations for the exam questions were judged by independent domain experts. The evaluation methodology proceeded as follows. First, after 4 months of development effort, we delivered our system (software and knowledge base) to Vulcan, where it was sequestered behind Vulcan's firewall. We could not make any changes to the knowledge base or the inference engine after this time. At this sequestration point Vulcan released the Halo Pilot Project Exam and we had 2 weeks to encode the questions as KM expressions. We submitted the question encodings to Vulcan, whose experts vetted them with a panel of KR&R experts for fidelity to the original English statements of the questions. Finally, Vulcan posed the encoded questions to the sequestered system.

The Halo Pilot Exam consisted of 50 multiple-choice questions (MC1-MC50), 25 detailed-answer questions (DA1-DA25), and 25 free-form questions (FF1-FF25). Many of the detailed-answer questions and free-form questions had multiple parts, each of which counted for as many marks as a single multiple-choice question. In terms of contribution to the final grade, there were 80 detailed-answer questions (excluding DA4e as out of scope) and 38 free-form questions. The "normalized" total number of questions was therefore 168. Three graders, working independently, graded the answers, and assigned each one a score of 0, 0.5, or 1 for correctness and a score of 0, 0.5, or 1 for the English explanation of the answer.

Results

In Figure 10, we show the point totals assigned by each grader for each section of the exam. Overall, our system scored 49% on answer correctness and 36% on explanation. The system required just over 5 hours of CPU time (on a 1.4 GHz Windows PC with 2 GB RAM) for the final exam. After the evaluation we were able to reduce processing time to 38 minutes, primarily through debugging.

The grades were quite consistent among graders. Correctness of multiple-choice questions was objective. The detailed-answer questions were more difficult than the multiple-choice questions, often requiring reasoning beyond simple calculation. Grading of detailed-answer correctness was somewhat more subjective. The free-form questions were more difficult still. These results reflect the design of the evaluation: Vulcan wanted to push the systems beyond their capabilities, and the free-form questions were meant to do just that. Explanation scores follow correctness scores for the most part, but may be somewhat artificially low, as graders rarely awarded points to good explanations when the answer was incorrect.

This achievement compares favorably with student performance on the AP chemistry test. This comparison is only approximate because there are several differences between the Halo Pilot Exam and the AP test: The Pilot Exam covered only a portion of the chemistry syllabus, corresponding to about 70 pages of a standard textbook; the questions on the Pilot Exam were similar to questions on an AP test, but not identical to them (due to copyright restrictions, Vulcan could not use AP questions); and there

	Grader1		*Grader2*		*Grader3*		*Average*	
	C	***E***	***C***	***E***	***C***	***E***	***C***	***E***
MC/50	35 (70%)	23.5 (47%)	35 (70%)	26 (52%)	35 (70%)	27 (54%)	35 (70%)	25.5 (51%)
DA/80	37 (46%)	24.5 (31%)	33 (41%)	26 (33%)	30 (38%)	31 (39%)	33.3 (42%)	27.2 (34%)
FF/38	14.5 (38%)	9 (24%)	13 (34%)	6 (16%)	12 (32%)	10 (26%)	13.2 (35%)	8.3 (22%)
TOTAL/168	86.5 (51%)	57 (34%)	81 (48%)	58 (35%)	77 (46%)	68 (40%)	**81.5 (49%)**	**61 (36%)**

Figure 10: Summary of test performance (C = correctness score; E = explanation score)

was no negative scoring. Given these caveats, a score of 49% earned by our system corresponds to an AP score of 3 on their 1-5 point scale. This is high enough to earn course credit at many top universities, such as the University of Illinois at Urbana-Champaign and the University of California at San Diego.

Two other systems built for the Halo Pilot used quite different technology yet performed almost as well, which reflects well on the state of KR&R. The system built by Ontoprise, was based on the OntoBroker system (Angele 1993, Decker et al. 1999) and F-Logic (Kifer, Lausen and Wu 1995), a Prolog-like language. The system built by Cycorp was based on Open Cyc and its elaborate inference engine. The Ontoprise system scored 44% on answer correctness and 32% on explanation, and the Cycorp system scored 37% on correctness and 24% on explanation.

The three systems – including software and knowledge base downloads, documentation, the complete exam, the graders' scores and comments, and a comprehensive analysis of the systems' failures – are available on the Project's Web site: http://www.projecthalo.com.

Evaluation of System Failures

In addition to evaluating overall performance, we examined the cases for which our system scored less than perfect in order to understand the factors that contributed to the failure. In general, the system performed well on questions involving mathematic computation for properties of specific instances, such as chemical substances. The system was less successful with more abstract questions, such as finding patterns in the activity series, or predicting the properties of typical members of a class of substances.

We found several recurring causes of system failure. The first was errors in domain modeling, which were caused by errors introduced by knowledge engineers and gaps in the knowledge base. It is hard to pin down the major cause of gaps but there were several factors that made knowledge engineering slow. First, the domain experts' knowledge was being encoded by knowledge engineers who sometimes misunderstood the knowledge, rather than by the experts directly. For example, the knowledge engineers unwittingly asserted that only cations can be Lewis acids, a statement that chemists know is clearly false. Second, much of the knowledge to be encoded was mathematical or procedural. Our declarative representation language is not particularly well-suited to this kind of knowledge. Finally, testing and debugging were slow due to the lack of good debugging tools and efficient testing procedures.

Another major cause of failures was inappropriate modeling assumptions. This type of mistake might be made by domain experts, but more frequently was made by knowledge engineers. For example, while building the knowledge base we assumed that questions that required computing an ionic equation must pertain to chemicals that are in solution. This assumption was violated by a question that asked about a chemical heated to 300ºC.

These two types of failure – modeling errors and inappropriate modeling assumptions – have a common cause. Because knowledge engineers are largely ignorant of the domain, their work is slow and error prone. Domain experts are the ideal encoders, but they require more effective tools than are currently available. Developing such tools is the focus of the next phase of Project Halo, building on the results of DARPA's Rapid Knowledge Formation project (Clark et al. 2001, Thoméré et al. 2002).

Another major cause of failures was the system's inability to reason about differences in the knowledge base. For example, one question asked for the difference between the subscript 3 and the coefficient 3 in $3HNO_3$. Subscripts and coefficients are different concepts in the knowledge base, but we did not explicitly encode knowledge of such modeling decisions. In some cases writing quantified query expressions over relations could have allowed us to answer these types of questions, but such expressions would possibly have been in violation of the rule requiring that question encodings be faithful to the original English.

With respect to explanation generation, the system's most common failure was producing an inappropriate level of detail. In some cases, the explanation was too shallow (e.g., when the system recalled the value of an equilibrium constant, stored as a ground fact, rather than explaining how it might be derived). More often the graders found the explanations to be too verbose. For example, if an explanation contained passages repeated multiple times with only small variations, the graders expected a general statement that covered them all. To overcome these problems, our system would need the ability to reason about its own explanations, which is a type of meta-reasoning that we are exploring in the next phase of Project Halo.

A complete discussion of the failure analysis of all three Halo Pilot teams appears in (Friedland et al. 2004).

Related Work

There are a few other systems that have been developed in the past to answer exam-style science questions, although with a significantly narrower scope and evaluation. Isaac (Novak 1977) and Mecho (Bundy et al. 1979), both developed several decades ago, demonstrated that a system could be built to answer certain types of high school level physics questions correctly, including interpreting the original English expression of those questions as stated in the exams. Novak later extended his work to include diagrams as an additional modality for stating questions (Novak and Bulko 1990). More recently, Forbus and Whalley's CyclePad system (Forbus and Whalley 1994) provides an "articulate virtual laboratory" for students to perform thermodynamic experiments and receive comprehensible explanations for the behavior they see.

More generally, despite the community-wide trend towards information retrieval style question-answering, there have been some systems developed to support the kind of knowledge-based question-answering behavior described here, including dealing with questions unanticipated at the time of system construction. These include Cyc (Lenat and Guha 1990), the Botany Knowledge Base system (Porter et al. 1988, Clark, Thompson and Porter 1999), the two systems developed for DARPA's High Performance Knowledge Base (HPKB) project (Cohen et al. 1998), and the two systems for DARPA's Rapid Knowledge Formation (RKF) project (Schrag et al. 2002). The work presented here demonstrates a new level of capability for this style of system, along with a new level of rigor in the evaluation. In addition, the two other systems developed for this Halo Pilot Project also performed well, as described earlier in the Results section of this paper. An integrated description and comparison of the three systems (outside the scope of this paper) is provided in (Friedland et al. 2004).

Our work on explanation builds on the now well-known lesson from expert systems that simply reciting the proof tree is ineffective. Other recent work on generating effective explanations include work on Xplain (Swartout 1983), Expect (Blythe et al. 2001), and InferenceWeb (McGuinness and Pinheiro da Silva 2003). We plan to leverage these ideas further in subsequent phases of the project.

Summary and Conclusions

We have described a large-scale knowledge engineering effort, designed to help assess the state of the art in knowledge-based question-answering. The scope of the effort was to encode the knowledge from 70 pages of a college-level chemistry textbook into a declarative knowledge base, and to answer questions comparable to questions on an Advanced Placement exam. Our solution integrated several modern KR&R technologies, in particular semantically well-defined frame systems, automatic classification methods, reusable ontologies, a methodology for knowledge base construction, and a novel extension of methods for explanation generation. The resulting system exhibited impressive performance by scoring about 50% on overall correctness – which is comparable to a passing grade on the AP exam – and about 35% on the explanation quality. Although many challenges remain to achieve the long-term objective of a "Digital Aristotle" – in particular acquiring domain knowledge more economically, reasoning with meta-knowledge, and making explanations more natural – these results are encouraging.

The significant conclusion from this work is that knowledge systems can be built to perform competently in scientific domains, and they can be built quickly, because KR&R research has developed the important building blocks that are required.

The results also confirm that the quality of the knowledge base is affected by the knowledge engineer's lack of expertise in the domain being modeled. Our current focus is on developing appropriate tools to allow domain experts to encode knowledge directly themselves.

The Halo Pilot Project evaluation presented significant new challenges for knowledge representation due to the large variability of question types that occur – so much so that it was not clear at the outset that a high-performing system could be constructed. The positive result is thus an achievement and an informative data point about the state of the art in KR&R, and reflects significant progress by the field.

References

Angele, J. 1993. *Operationalisierung des Modells der Expertise mit KARL*. DISKI, Infix Verlag.

Barker, K., P. Clark and B. Porter 2001. A Library of Generic Concepts for Composing Knowledge Bases. In Proceedings of the First International Conference on Knowledge Capture, 14-21. Victoria.

Bobrow, D.G. and T. Winograd 1977. An Overview of KRL, A Knowledge Representation Language. *Cognitive Science* **1**(1):3-46.

Blythe, J., J. Kim, S. Ramachandran and Y. Gil 2001. An Integrated Environment for Knowledge Acquisition. In Proceedings of the International Conference on Intelligent User Interfaces, 13-20. Santa Fe.

Brachman, R.J. and J.G. Schmolze 1985. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* **9**(2):171-216.

Brown, T.L., H.E. LeMay, B.E. Bursten and J.R. Burdge 2003. *Chemistry: The Central Science*. New Jersey: Prentice Hall.

Bundy, A., L. Byrd, G. Luger, C. Mellish, R. Milne and M. Palmer 1979. MECHO: a program to solve mechanics problems. Technical Report Working paper 50, Department of Artificial Intelligence, Edinburgh University.

Clark, P. and B. Porter 1997. Building Concept Representations from Reusable Components. In Proceedings of the Fourteenth National Conference on Artificial Intelligence, 369-376. Providence.

Clark, P. and B. Porter 1999. *KM -- The Knowledge Machine: Users Manual*. http://www.cs.utexas.edu/users/mfkb/km.html

Clark, P., J. Thompson and B. Porter 1999. A Knowledge-Based Approach to Question-Answering. In Proceedings of the AAAI Fall Symposium on Question-Answering Systems, 43-51. North Falmouth, MA.

Clark, P., J. Thompson, K. Barker, B. Porter, V. Chaudhri, A. Rodriguez, J. Thoméré, Y. Gil and P. Hayes 2001. Knowledge Entry as the Graphical Assembly of Components. In Proceedings of the First International Conference on Knowledge Capture, 22-29. Victoria.

Cohen, P., R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Easter, D. Gunning and M. Burke 1998. The DARPA High Performance Knowledge Bases Project. *AI Magazine* **19**(4):25-49.

Decker, S., M. Erdmann, D. Fensel and R. Studer, eds. 1999. *Ontobroker: Ontology-based Access to Distributed and Semi-Structured Information*. Database Semantics: Semantic Issues in Multi-media Systems, ed. R. Meersmann. Kluwer Academic Publishers.

Forbus, K. and P.B. Whalley 1994. Using Qualitative Physics To Build Articulate Software For Thermodynamics Education. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 1175-1182. Seattle.

Friedland, Noah S., P.G. Allen, M. Witbrock, G. Matthews, N. Salay, P. Miraglia, J. Angele, S. Staab, D. Israel, V. Chaudhri, B. Porter, K. Barker and P. Clark 2004. Towards a Quantitative, Platform-Independent Analysis of Knowledge Systems. In Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning. Whistler.

Kifer, M., G. Lausen, and J. Wu 1995. Logical Foundations of Object Oriented and Frame Based Languages. *Journal of the ACM* **42**:741-843.

Lenat, D.B. and R.V. Guha 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* MA: Addison-Wesley.

McGuinness, D.L. and P. Pinheiro da Silva 2003. Infrastructure for Web Explanations. In Proceedings of the Second International Semantic Web Conference (ISWC 2003), 113-129. Florida.

Novak, G. 1977. Representations of Knowledge in a Program for Solving Physics Problems. In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, 286-291. Cambridge, Mass.

Novak, G. 1995. Conversion of Units of Measurement. *IEEE Transactions on Software Engineering* **21**(8):651-661.

Novak, G. and W. Bulko 1990. Understanding Natural Language with Diagrams. In Proceedings of the Eighth National Conference on Artificial Intelligence, 465-470. Boston.

Porter, B., J. Lester, K. Murray, K. Pittman, A. Souther, L. Acker, T. Jones 1988. AI Research in the Context of A Multifunctional Knowledge Base: The Botany Knowledge Base Project, Technical Report, AI-88-88. Department of Computer Sciences, University of Texas at Austin.

Schrag, R., M. Pool, V. Chaudhri, R.C. Kahlert, J. Powers, P. Cohen, J. Fitzgerald and S. Mishra 2002. Experimental Evaluation of Subject Matter Expert-oriented Knowledge Base Authoring Tools. In Proceedings of the 2002 PerMIS Workshop: Measuring the Performance and Intelligence of Systems. NIST Special Publication 990, 272-279. Gaithersburg.

Swartout, W. 1983. XPLAIN: A System for Creating and Explaining Expert Consulting Programs. *Artificial Intelligence* **21**(3):285-325.

Thoméré, J., K. Barker, V. Chaudhri, P. Clark, M. Eriksen, S. Mishra, B. Porter and A. Rodriguez 2002. A Web-based Ontology Browsing and Editing System. In Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference, 927-934. Edmonton.