VULCAN

The Halo Pilot: Towards A digital Aristotle

Noah S. Friedland and Paul G. Allen

Abstract

Vulcan Inc. has launched a multi-staged effort towards the creation of a Digital Aristotle, an application capable of answering arbitrary questions in an ever-growing number of domains and of providing cogent explanations for its answers. The Pilot phase was a six-month effort intended to investigate the state-of-the-art in question answering, with an emphasis on deep reasoning. To this end we released a selective call for proposals targeted at leaders in the field. Three organizations were ultimately funded: Team SRI, which included significant contributions from Boeing and the University of Texas, Cycorp and Ontoprise.

The program was structured around a challenge problem involving 50 pages of a chemistry syllabus. Over a four-month period, the teams developed chemistry question-answering systems based upon their knowledge representation and reasoning (KRR) technologies. These systems were then sequestered while the teams spent an additional two weeks encoding 100 mostly novel (i.e. previously unseen) chemistry questions into their respective formal languages. These questions were then processed in batches on the sequestered systems. A panel of chemistry professors then evaluated the challenge outcomes. Vulcan's challenge methodology emphasized the following: (i) coverage: the system's ability to answer novel questions across the entire specified syllabus; (ii) explanation generation: the ability to provide concise, coherent, user and domain appropriate explanations for the answers produced, and (iii) question encoding: the ability to create high-fidelity translations of the questions into each system's formal language. Additionally, in order to help characterize and prioritize the central challenges facing the field, a “brittleness” taxonomy of failure modes was developed and failures in each of the Halo systems were classified into cells of that taxonomy.

Despite their differences in approach, all three systems performed very well on the challenge. This outcome leaves Vulcan Inc. highly optimistic about our ability to help identify and overcome many of the key technical challenges facing this effort in years to come. This paper discusses the Halo vision and methodology and provides an overview of Pilot phase and its challenge. It also outlines the next phase of our program.

VULCAN.COM

1. Introduction

Aristotle (384-322 B.C.E) was remarkable for the depth and scope of his knowledge, which included mastery of wide-ranging topics from medicine and philosophy to physics and biology. Aristotle not only had command over a significant portion of his world's knowledge, but he was also able to explain this knowledge to others, most famously Alexander the Great.

Today, the knowledge available to humankind is so extensive (and growing at an exponential rate) that it is not possible for a single human being to replicate Aristotle's mastery. Yet, at the height of the digital age, the role of Aristotle the tutor is more central than ever. We need an intelligent guide to help us make sense of the explosion of information: printed and digital books and documents, web pages, databases, imagery, film and video, etc.

"Making sense" might involve anything from the simple retrieval of facts, to answering a complex set of interdependent questions and providing appropriate justifications for those answers. While the simple fact retrieval might be partially achieved by search and indexing applications like Google®, or other more sophisticated information retrieval (IR) applications, answering complex questions typically requires a significant amount of subject matter knowledge. IR systems emphasize the efficient retrieval of preexisting information. They are designed to work quickly over a large corpus. Their weakness is that they would be incapable of answering questions for which no direct answer is to be found in the corpus. And such systems are completely incapable of generating useful explanations for the answers they find--unless, again, those justifications are explicit in the texts. By contrast, knowledge-based question–answering systems are generally more computationally intense, work over much smaller corpora, but would be capable of generating an answer and its justification for even novel, i.e. previously unseen, questions for which answers are not to be found in texts.

The United States government has invested considerable resources in related fields in the last decade, in particular, on knowledge-based systems and information retrieval/extraction.

In the area of knowledge-based systems, DARPA, AFOSR, NRI and NSF jointly funded the Knowledge Sharing Effort[1] in 1991. This was a three-year collaborative program to develop "knowledge sharing" technologies to facilitate the exchange and reuse of inference-capable knowledge bases among different groups. The aim was to help reduce cost and promote development of knowledge-based applications. This was followed by DARPA's High Performance Knowledge Base (HPKB) program (1996-2000), designed to push knowledge-based technology further and demonstrate that very large (100k+ axiom) systems could be built quickly and usefully applied to question-answering tasks[2]. Although the resulting systems were impressive, one bottleneck was the requirement that knowledge engineers, not domain experts, construct the knowledge bases. This led to DARPA's Rapid Knowledge Formation (RKF) program (2000-2004), whose goal was to develop new technologies for

[1] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. *Enabling technology for knowledge sharing.* AI Magazine, 12(3):16-36, 1991.

[2] Cohen, P.R., Schrag, R., Jones, E., Pease, A., Lin, A., Starr, B., Easter, D., Gunning, D., and Burke, M. *The DARPA High Performance Knowledge Bases Project.* In Artificial Intelligence Magazine 19, 4, pp. 25-49, 1998.

acquiring formal domain knowledge from domain experts directly. The RKF program is now in its last year.

Question-answering via information retrieval and extraction from texts has also been an active area of research, with a progression of annual competitions and conferences, especially the 12 Text Retrieval Conferences (TREC) from 1992-2003, sponsored by NIST, IAD, DARPA, and ARDA. Initially aimed at , retrieving relevant texts from large collections and then at extracting relevant passages from texts, the earlier systems had little need for inference-capable knowledge bases and reasoning capabilities. However, in recent years the question-answering tasks have become more challenging, e.g., requiring a direct answer to a question rather than a passage containing the answer. This has led to a revival of interest in the use of domain knowledge (e.g., see \cite[3]). ARDA's current AQUAINT program (Advanced Question and Answering for Intelligence), started in 2001, is also pushing text-based question-answering technology further, seeking to address a typical intelligence gathering scenario in which multiple, inter-related questions are used to fulfill an overall information need, rather than answering just single, isolated, fact-based questions.

Project Halo

Project Halo is a Vulcan Inc. initiative to build a Digital Aristotle, an application or platform capable of answering and providing cogent, user-appropriate explanations to novel questions in an ever-growing number of domains and disciplines. The Halo Project will support a phased, incremental approach designed to identify and overcome the significant technological roadblocks and to champion a rigorous, open, measurable and repeatable methodology as the best way forward.

Halo Pilot, the first phase of project Halo, is now complete and its results will be discussed in detail in section 2 of this paper. Section 3 contains our vision for the next steps in Project Halo.

2. Project Halo's first step–The Halo Pilot

2.1 Overview

The Halo Pilot was a six-month effort to investigate the state-of-the-art in "deep reasoning" knowledge-based question answering. This technology was selected for of its ability to answer complex, novel questions, questions of types that have not been previously encountered--a key differentiator from corpus-based (statistical) approaches. Vulcan Inc. wanted to address the following basic questions with regard to knowledge-based systems:

1. Could they be built to provide demonstrably robust coverage of a given domain; i.e., could we obtain performance metrics on these systems' ability to answer complex novel questions for the specified domain?
2. Could they be built to provide user- and domain-appropriate answer justifications?

[3] Chu-Carroll, J., Ferrucci, D., Prager, J., Welty, C. *Hybridization in QA Systems.* In Proc. AAAI Symposium on New Directions in Question-Answering, CA:AAAI. 2003.

3. Could the major causes of brittleness be identified and strategies formulated on improving system robustness?

The pilot was designed to be of limited scope and duration--six months--and, for each team, between 2-4 person-years. It was intended primarily as an exercise in knowledge formulation and question-answering.

Given the narrow focus of the effort, Vulcan Inc. did not require a natural language interface for posing queries, instead allowing each team to use its own formal language to encode the questions, as well as the domain knowledge. Answers and answer-justifications were required to be in English so that domain experts would be able to evaluate the challenge outcome.

Chemistry was selected as a domain, and the chosen syllabus was 50 pages from Brown and LeMay's "Chemistry: The Central Science."[4] Topics covered included: stoichiometry calculations with chemical formulas; aqueous reactions and solution stoichiometry; and chemical equilibrium. This syllabus was selected because it met all of the following criteria:

1. The scope was small enough to be done in a reasonable amount of time, yet large enough to be used to generate many novel questions.
2. It provided complex "deep" combinations of rules.
3. The material was not overly reliant upon diagrams and graphics.
4. A nationwide standardized test was available. In this case the AP chemistry exam. It was important for us to find a challenge topic area for which the results would be easily understandable beyond the AI research community.
5. Chemistry is an exact and "hard" science and hence avoided addressing challenging representational issues, such as those raised in attempts to capture the complexity of psychological states, which are more widespread in other domains.
6. Domain experts were readily available to assist in syllabus preparation, as well as authoring and evaluating the challenge exam.

The project allocated the Pilot participants four months to encode the syllabus and deliver their systems to Vulcan Inc. for sequestration. The sequestered systems were hosted on identical servers and isolated by firewall from the Internet. Once the systems were sequestered, the challenge, consisting of 100 chemistry questions was released. Each team had two weeks to encode these questions in their respective formal languages: CycL for Cycorp; KM for SRI and F-Logic for Ontoprise. All question encodings were subject to review by a peer-based encoding committee (see section 2.4 for more details). Once the committee comments were collected, Vulcan Inc. submitted the 100 questions to each of the sequestered systems in batch mode. The faithfulness of the test was ensured by comparing the output on the sequestered system with the output on the system maintained by the teams at their own site.

Each system produced an English document with the answers and answer justifications. These were rendered into hard copy and delivered to a panel of chemistry professors for evaluation. A final analysis step involved a post-mortem effort

[4] ISBN number 0-13-066997-0

to understand and classify the causes of failures in terms of a brittleness taxonomy developed by the Halo teams. Extensive project documentation, including all the challenge questions, the encodings, the graded exams and a detailed brittleness analysis are available on our project website[5]. To facilitate better understanding of the material, Vulcan Inc. has developed an interactive results browser application that allows the user to examine the results of the chemistry evaluation on a question-by-question basis.

2.2 The Pilot Teams

To truly capture the state-of-the-art in deep reasoning, it was important to identify and engage the leading teams and technologies in this space. To achieve this goal Vulcan Inc. studied the HPKB, RKF and AQUAINT programs and worked to establish cooperative relationships with their respective program managers. These relationships and the Vulcan team's own due diligence process produced a short list of candidate teams. The main selection criteria were:

1. A significant investment in working technology. As an examination of the state-of-the-art, pilot candidates were limited to teams that had mature, operational KRR systems with a considerable number of person-years invested.
2. A track record or either government or commercially funded projects.
3. A world-class team that could allocate significant human resources to the task.
4. Significant interest and activity in a broad range of topics in this general area, e.g. the Semantic Web.
5. A proposal that appropriately addressed the domain specific issues called out in the CFP.
6. The ability to work within the IP, budget and contractual framework established by Vulcan Inc.

Three teams were funded: Ontoprise, based in Karlsruhe Germany; Cycorp, based in Austin Texas; and SRI, based in Menlo Park, with significant contributions from the University of Texas and Boeing. These teams represented significant different approaches to KRR.

Cycorp has over 600 person-years invested in the development of the world's largest knowledge base, featuring over a million entities and relationships and tens of thousands of axioms organized into several thousand microtheories. These constructs are tied together by an "upper ontology." Cycorp tries to leverage its knowledge base extensively when constructing new knowledge. Its reasoning engine has over 500 computational modules and claims to be fundamentally non-monotonic in its reasoning approach. It employs a truth maintenance system (TMS) to verify that new knowledge does not corrupt existing knowledge. Its formal language, CycL is an extremely expressive logic-based representation.

SRI's approach to knowledge formulation relies on a component library consisting of several hundred concepts, which can be combined into more complex knowledge

[5] See www.projecthalo.com

constructs. Their approach relies on the assumption that these fundamental building blocks can be easily extended and specialized for each new domain. SHAKEN, their KRR environment, features the KM formal language, an expressive frame-based representation. The engine supports monotonic reasoning only, with heuristics for handling identity, and does not, to date, employ a TMS. It does feature an automated entity classification capability.

Ontoprise's formal language is F-Logic. F-logic ("F" stands for "Frames") combines the advantages of frame-based languages and the expressiveness, compact syntax, and well-defined semantics of logic programming languages such as Prolog. The original features of F-logic[6] include signatures, object identity, complex objects, methods, classes, inheritance and rules. Their inference engine OntoBroker® provides means for efficient reasoning in F-Logic through a mixture of forward and backward chaining, based on a dynamic filtering algorithm[7] to compute (the smallest possible) subset of the model for answering the query. The semantics for a set of F-Logic statements is then defined by a transformation process of F-Logic into normal logic (Horn logic with negation) and the well-founded semantics[8] for the resulting set of facts and rules and axioms in normal logic. Their engine does not currently provide any framework for knowledge reuse, rather each knowledge base is constructed from scratch and customized for the defined requirements of the given project or problem at hand.

2.3 Knowledge Formulation & Answer Justification

The Halo teams were given four months to formally encode the specified chemistry syllabus with an emphasis on robust coverage, i.e. the ability to answer novel questions and question types. The evaluation objective was to prove that the underlying reasoning engine could address complex question types without prior knowledge of what the structures of these questions might be.

The pilot also put a strong emphasis on answer justification. The intent was to address two issues: (i) to exhibit the system's reasoning with the chemistry laws, and (ii) to create justifications that would allow the chemistry subject matter experts (SMEs) to evaluate the problem solutions in a way that is most appropriate for the provided evaluation metric: an AP-style exam.

SRI's Halo chemistry system was built on top of their SHAKEN system and features approximately 500 domain-specific knowledge elements (entities, relations, rules). SRI's was the only team to employ professional chemists to assist in knowledge formulation. These chemists reviewed the syllabus and produced a specification of the knowledge they extracted from the text; most important, they produced a detailed specification of the chemical laws and rules of calculation These were subsequently shipped to the knowledge engineering team at the University of Texas, who rendered

[6] M. Kifer, G. Lausen, and J.Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42:741–843, 1995.
[7] M. Kifer and E. Lozinskii. A framework for an efficient implementation of deductive databases. In *Proceedings of the 6th Advanced Database Symposium*, pages 109–116, Tokyo, August 1986.
[8] A. Van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3):620–650, July 1991.

them into KM constructs. The laws were used in backward chaining to invoke classification and elaboration to answer the test questions. The SRI Team's approach to Explanation Generation involved designing an explanation design plan that allowed significant deviation from the order of steps generated by the question answering inference procedure. The explanation design plan was a knowledge structure that could contain an inference chain or a set of analogical inferences or a qualitative model. The explanation generator took the explanation design plan, and organized it into packets that became sentences and paragraphs in the resulting explanation. For the Halo project, the explanation generation was based on presenting a trace of the proof tree, with substantial guidance from the knowledge engineer on steps of the inference that were important. English output was generated by substituting inferred and computed values for variables in human-authored templates associated with each chemical law. This approach provided a great deal of control over the generated explanation, at the cost of being tedious to develop, hard-coded for the narrow domain at hand, and highly susceptible to human error.

Cycorp’s OpenHalo chemistry knowledge base was built on an enhanced version of OpenCYC, and features over 15,000 knowledge elements. The platform used general theorem-proving techniques of unification and resolution, supported by a set of general modules that solved common classes of special-case problems. CycL, the formal representation language used, is extremely expressive and quite verbose. Although it is possible to express inference-guiding heuristics in CycL, it was not necessary to do so as part of question encoding. While some additional support modules were written for Project Halo, these were claimed to be largely independent of the chemistry domain. OpenHalo employed a meta-reasoning capability to choose whether to answer multiple-choice questions by positive selection or negative elimination. Also, instead of making the closed world assumption, meta-knowledge about completeness was represented directly in the knowledge base at a fine-grained level. The answer justification system used for OpenHalo was developed from existing systems in Cyc. Justifications were generated from appropriately filtered and ordered proof trees. OpenHalo employed compositional natural language generation to produce the English explanation texts, which tended to be highly verbose and often were difficult to parse and understand.

Ontoprise built the OntoNova system on top of OntoBroker™. It features no more than 500 knowledge elements, mostly rules. Answer justification in OntoNova was handled by a meta-inferencing process. For all successful queries, i.e. for all instances where the inference engine successfully generated a result, it produces a log file of the proof tree. This log file contains all the instantiated rules that were successfully applied to derive an answer. This file is then used as input for a second run of the inference engine, in which template-based natural language explanations for the answer of the original query are generated. This two-step process allowed OntoNova to apply the full power of inferencing to generate answer justifications, though in reality the OntoNova implementation took very little advantage of these capabilities. In future an entire knowledge base could be developed for this purpose. This would allow: (i) additional knowledge to be integrated, (ii) redundancies in the justification text to be reduced, (iii) different explanation abstractions to be created, and (iv)personalized explanations to be provided.

2.4 Question Encoding

Once the systems were sequestered, 100 challenge questions were released to the teams, who had two weeks to produce formal encodings in their respective logical languages. The SME who designed the evaluation was instructed to produce an AP-style chemistry exam, but to minimize the question type similarity between it and the set of 50 sample questions provided to the teams during the initial phase of the pilot. Once the formal encodings were produced, an encoding committee led by Vulcan (with the participation of representatives from all three teams), assessed the question encodings for fidelity to the original English text. The criterion used to assess question fidelity was as follows:

Suppose a student was fluent both in English and in the formal language in which the question encoding was formed. If that student could infer additional relevant knowledge about the chemistry problem beyond what was stated in the English text from the formal encoding, then a fidelity violation has occurred.

These violations typically fell into two categories: violations of omission or commission. In the former case, information stated in the English text was missing from the formal encoding, e.g. the encoding may have omitted the color or smell of a compound. In the latter case, new information was added to the formal encoding that did not exist in the original English. An example of this might be providing the ionic decomposition of a substance that was implied but not specified in the original question.

The formal encodings were all required to be inputted to the Halo systems as a single “encoding” file. The systems were required to process these files in batch mode and generate a “results” file. Both these files were subject to strict formatting requirements, which were largely complied with by the Halo teams. The challenge questions and detailed encoding files, with the encoding committee comments, are available on the project Web site.

2.5 The Challenge

The Halo challenge was designed to determine if KRR systems could answer and provide domain-appropriate answer justifications for previously unseen question types within the scope of the specified syllabus. Our experiment did not require an NLP front end, because (i) we had a very limited scope and timeframe, (ii) we didn’t want to compound and confuse reasoning and representation failures with NLP related errors, and (iii) because in the pilot we were not evaluating SME-oriented knowledge base formulation tools. Rather, we wanted to see how well the knowledge experts could do under the best possible conditions.

Since the evaluation systems were sequestered, the teams were not able to make adjustments to the knowledge bases and inferencing engines after receiving the challenge questions. This requirement also meant that the sequestered systems needed to be user friendly and stable enough to allow a third party (Vulcan) to run the challenge. This maturity of technology resulted in the high quality downloads that have been made available to the general public through our project Web site.

The Encoding file was required to contain a sequential listing of the challenge questions. For each question or question subpart, the following information was required:

1. The question and question subpart designation (when applicable)
2. The question or question subpart original English text
3. The formal encoding of the original question or question subpart English text
4. An English explanation of the formal encoding
5. The fidelity comments of the fidelity committee for the provided encoding

The Results file was required to contain a sequential listing of the challenge results. This was the document that was presented to the panel of chemistry professors for evaluation.

For each question or question subpart, the following information was required:

1. The question and question subpart designation (when applicable)
2. The question or question subpart original English text
3. The question or question subpart answer. This was required to be a letter answer for multiple-choice questions or a concise sentence for essay questions
4. The question or question subpart answer justification. This was required to be a concise, domain appropriate answer justification, not to exceed a page for each question or question subpart (when applicable)

The challenge questions were organized into three sections: (i) 50 multiple-choice questions, (ii) 25 detailed answer questions, and (iii) 25 free-form questions. The latter two sections contained mostly multipart questions, requiring “English” answers, while the multiple-choice section required choosing a letter answer from among (typically) 5 alternatives. The free-form section was more comprehension oriented and less computational in nature. Answer justifications were required for all three sections, even for multiple-choice questions. In all, the 100 questions broke down into 168 question parts, each awarded a score for correctness and answer justification. These scores ranged from 0, 0.5 to 1 point per question part; thus the maximum grade for a single exam could be 168 for correctness and 168 for answer justifications, for a combined maximum score of 336 points. Since each exam was graded by three separate SMEs, the maximum score a team could obtain on the challenge was 1008 points.

2. When lithium metal is reacted with nitrogen gas, under proper conditions, the product is:

(a) no reaction occurs
(b) LiN
(c) Li2N
(d) Li3N
(e) LiN3

Figure 1.A: Challenge Question, MC-2

Figure 1.A depicts question MC-2, which is the second question in the multiple-choice section. A correct response to this question must include the correct identification of the letter answer in conjunction with a correct answer justification. Figure 1.B depicts question DA-6, the sixth question in the detailed answer section, which contains four question subparts. Here, each subpart requires a correct answer "phrase" and a detailed answer justification.

6. For the following, indicate the oxidation number of each element, which species is reduced, and which is oxidized.

(a) $Ni + Cl_2 \quad NiCl_2$

(b) $3Fe(NO_3)_2 + 2Al \quad 3Fe + 2Al(NO_3)_3$

(c) $Cl_2 + 2NaI \quad I_2 + 2NaCl$

(d) $PbS + 4H_2O_2 \quad PbSO_4 + 4H_2O$

Figure 1.B: Challenge Question, DA-6

Vulcan was able to run all of the applications independently during the challenge run, despite minor problems associated with each of the three systems. After the challenge evaluation was complete, the teams put in a considerable effort to make improved versions of their application for use by the general public. These improved versions address many of the problems encountered on the sequestered versions. Vulcan Inc. has made both the sequestered and improved versions available for download on the project Web site.

Figures 2 and 3 provide the challenge results, as percentages of the 168-point maximum scores, for answer correctness and quality of justification respectively. Scores are broken down by the respective SMEs to provide some insight into the variability exhibited in the scoring.

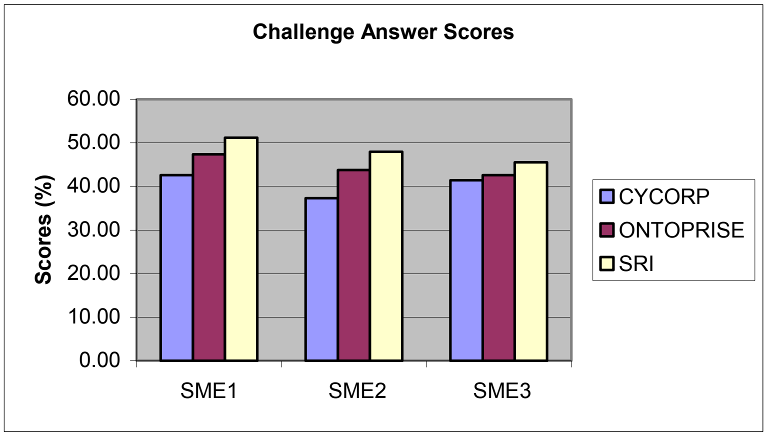

Figure 2: Challenge Answer Scores

Figure 2 shows a similar trend for the three SMEs, with SRI slightly outperforming Ontoprise and Ontoprise slightly outperforming Cycorp. By contrast, the justification scores depicted in Figure 3 display a significant amount of variability. Team SRI was rated first by the three SMEs and its scores exhibit the most stability overall, whereas the other two teams were rated significantly differently by the three SMEs.

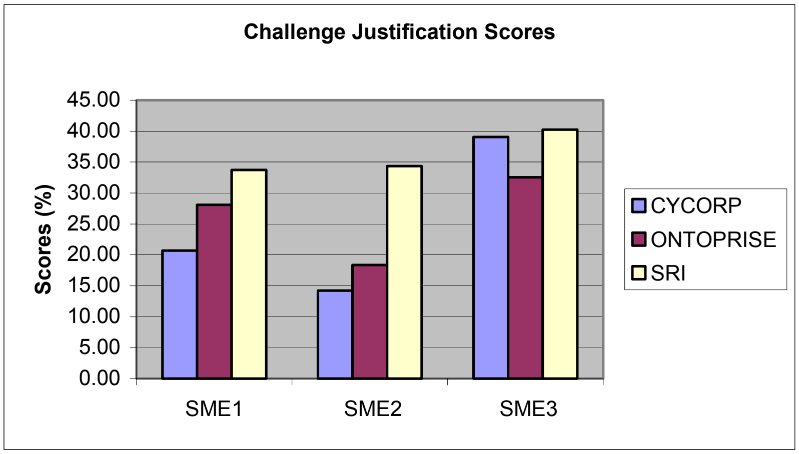

Figure 3: Challenge Justification Scores

The SMEs provided high-level comments, mostly focused on the organization and conciseness of the justifications. In some instances, justifications were quite long. For example, Cycorp had some justifications in excess of 16 pages in length. The SMEs also complained that many of the arguments were used repetitively and that proofs took a long time to "get to the point." In some multiple-choice questions, proofs involved invalidating all wrong answers, rather than proving the right one. All the teams appeared to rely on instance-based solutions to prove generalized comprehension-oriented questions, indicating a lack of meta-reasoning capability. Gaps in knowledge coverage were also evident. For example, many of the teams had significant gaps in their knowledge of net ionic equations. Detailed question-by-question scores are available on the project Web site.

2.6 Failure Analysis

The Halo Pilot put considerable emphasis on understanding brittleness and other causes of system failure. A "Brittleness Taxonomy" was collaboratively developed by the Halo participants to provide a common framework within which to quantitatively categorize and analyze challenge point losses. A brittle system is one that experiences a precipitous drop in performance when it moves outside of its original scope of application. The taxonomy contained 10 high-level "influences:"

1. **(MOD) Knowledge Modeling:** the ability of the knowledge engineer to model information/write axioms
2. **(IMP) Knowledge Implementation/Modeling Language:** the ability of the representation language to accurately represent axioms
3. **(INF) Inference and Reasoning:** the ability of the inference engine to "find the needle in the haystack"
4. **(KFL) Knowledge Formation and Learning:** the ability of the system (KB + inference engine) to acquire and merge knowledge through automated and semi-automated techniques
5. **(SCL) Scalability:** the ability of the KB to scale
6. **(MGT) Knowledge Management:** the ability of the system to maintain, track changes, test, organize, document; the ability of the knowledge engineer to search for knowledge

7. **(QMN) Query Management:** the ability of the system to robustly answer queries
8. **(ANJ) Answer Justification:** the ability of the system to provide justifications for answers in the correct context and resolution
9. **(QMT) Quality Metrics:** the ability of the developers to determine how "good" the knowledge base is at any given point in its evolution
10. **(MTA) Meta Capabilities:** the system's ability to utilize meta-reasoning or meta-knowledge

Based on these influences, 24 brittleness types were defined, each containing a designation, a name, a description, an example of an instance of this type of brittleness, ways to mitigate this brittleness type using current technology and practices, and ways to address this brittleness through future research.

After the challenge was graded by the three SMEs, the teams were instructed to review the points lost on each of the questions and assign them to one or more of the brittleness types. A "non-brittleness related failure," or OTHER type was defined as a catchall for failures that were not considered to be brittleness related.

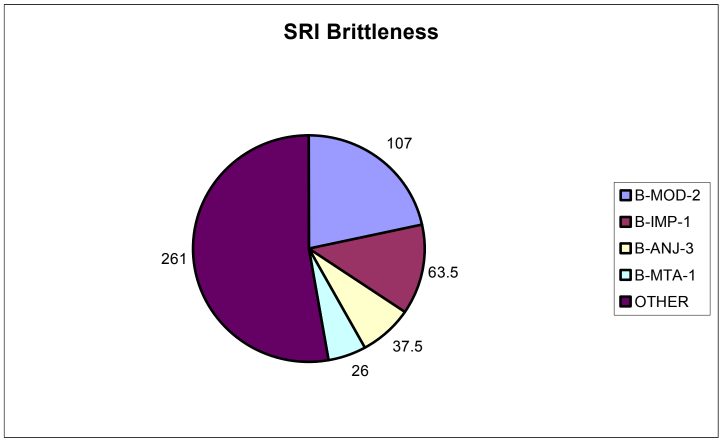

Figure 4: SRI Brittleness Analysis

Team SRI indicated OTHER as the primary failure contributor. The lion's share of points lost to this category were due to gaps in the knowledge encoding, a result of lack of time on a four-month encoding effort. Other major sources of brittleness were the system's lack of meta-reasoning and meta-knowledge (B-MTA-1). This manifested itself in the system's inability to fully answer some of the example-oriented comprehension questions. Under-expressive language brittleness (B-IMP-1) addressed cases were KM was not expressive enough to correctly represent the question or knowledge. Modeling assumption brittleness (B-MOD-2) covers cases where designers implicitly make "context" assumptions. Context justification brittleness (B-ANJ-3) addresses instances where appropriate justifications cannot be produced by the system.

.

Ontoprise identified modeling error brittleness (B-MOD-1) and under-expressive language brittleness (B-IMP-1) as its two major causes of failure. Cycorp also attributed most of its point losses to modeling error brittleness, followed by exposition brittleness (B-ANJ-1), which deals with the limitations of producing coherent justifications on the basis of proof trees. Practical incompleteness (B-INF-3), where inferencing fails to converge within the allotted timeframe, was another major Cycorp brittleness, followed closely by modeling assumption brittleness.

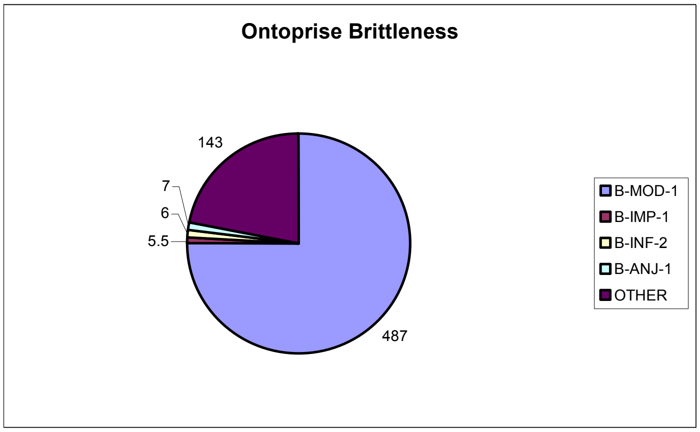

Figure 5: Ontoprise Brittleness

Additional work is required to tighten definitions in the taxonomy so as to allow a better cross-platform comparison. An example of this is how the teams opted to handle gaps in their constructed knowledge bases. While Cycorp and Ontoprise attributed this to modeling error brittleness (B-MOD-1), SRI opted to assign this to the OTHER category. The detailed question-by-question failure analysis and the complete brittleness taxonomy are available on the project Web site.

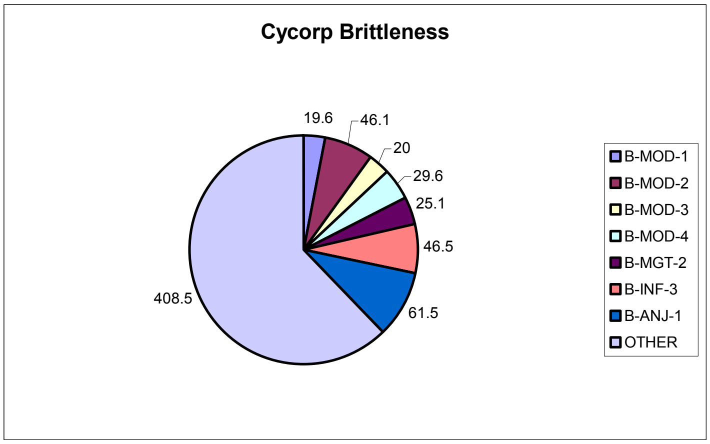

Figure 6: Cycorp Brittleness

2.7 A Word On Performance

Although run-time performance was never factored into the evaluation, the teams did exhibit distinct performance profiles. The end-to-end run-time numbers for the challenge runs are tabulated in Table 1. Note that the SRI and Ontoprise systems exhibit a considerable speedup in their processing times, while the Cycorp system considerably increased its processing time. This very long run was curtailed in the sequestered Cycorp system during the challenge due to a memory leak. It will be an interesting experiment to determine whether the faster implementations maintain their improved speed performance numbers when encountering a new set of novel questions.

Table 1: Team End-To-End Challenge Run Times

Team	Sequestered	Improved
Cycorp	> 12 hours	> 27 hours
Ontoprise	2 hours	9 minutes
SRI	5 hours	38 minutes

2.8 Halo Pilot Summary

All the Halo systems did surprisingly well on a very difficult challenge. Extrapolating the results against the limited 50-page syllabus to the entire AP syllabus yielded the equivalent of an AP-3 score for answer correctness. The Halo teams believe that with additional, limited effort they would be able to improve the scores to the AP-4 level and beyond. Vulcan Inc. has retained two additional challenge question sets to possibly validate these claims at a future date.

All three logical languages, KM, F-Logic and CycL were expressive enough to represent most knowledge in this domain. F-Logic was by far the most concise and easy to read, with a syntax most resembling an object-oriented language. F-Logic also yielded very high-fidelity representations that appear to be easier and more intuitive to construct. Ontoprise was the only team to conduct a sensitivity study of the impact of different, if correct, question encodings on system performance. In the case of the two questions they examined, their system produced similar answers with slightly different justifications. For the most part, the encoding process and its impact on question-answering stability remain an open research topic. The reader should treat the Halo results as optimized in this regard.

SRI and Ontoprise yielded comparably sized knowledge bases. OntoNova was built from scratch using no pre-defined primitives, while SRI's Halo-SHAKEN leveraged their component library, though not as extensively as they had initially hoped. SRI's use of professional chemists in the knowledge formulation process was a huge advantage and the quality of their outcome is reflected by this fact. The other teams have conceded that, had they the opportunity to revisit the challenge, they would have adopted the use of SMEs in knowledge formation. Cycorp's OpenHalo knowledge base was two orders of magnitude larger than the other teams'. They were unable to demonstrate any measurable advantage in using this additional structure, even in example-based questions, where they exhibited similar meta-reasoning brittleness observed in the other systems. The size of their knowledge base does however explain some of the significant run-time differences. They have also yet to demonstrate successful, effective reintegration of Halo knowledge into the extended Cyc platform. Reuse and integration appear to remain open questions for all three Halo teams.

The most novel aspect of the Halo Pilot was the great emphasis put on answer justifications. This served two primary purposes: (i) to exhibit and thereby verify that deep reasoning was occurring and (ii) to validate that domain and user-specific explanations can be generated. This is an area that is still open to significant improvement. SRI's challenge approach produced the best quality results, but it leaves many questions regarding how well it might be scaled, generalized and reused. Cycorp's generative texts may eventually scale and generalize, but the current results were extremely verbose and mostly unintelligible. Ontoprise's approach of running a second inferencing step appears to be very promising in the near term and we look forward to following the development of this approach.

The brittleness analysis provided some encouraging insights into system frailties. Vulcan Inc. will continue to work on improving the taxonomy and tightening its definitions. We are very interested in determining whether the detailed quantitative analysis produced during the project will result in higher-quality, more robust inferencing engines.

Vulcan Inc. and the pilot participants have invested considerable efforts in promoting the transparency of the Halo pilot. The project Web site provides all the scientifically relevant documentation and tutorials, including an interactive results browser application and fully documented downloads representing both the sequestered systems and improved Halo Pilot chemistry knowledge bases. We eagerly anticipate comment from the AI community and look forward to its use by universities and other researchers.

Finally, the issue of cost must be considered. We estimate that the per-page expense for each of the three Halo teams was on the order of $10,000 per page for the 50-page syllabus. This cost must be significantly reduced before this technology can be considered viable for the Digital Aristotle.

3. Next Steps

Phase II of Project Halo will focus on reducing the cost of knowledge and question formulation by assembling tools to allow domain experts, like chemists, physicists, biologists and historians, to create knowledge representations independently. Successfully constructing such tools and their widespread adoption by communities of domain experts will "democratize" knowledge formulation, significantly reducing its cost. An additional benefit would be the reducing B-MOD-1 type modeling brittleness--the major single source of brittleness in the Halo systems--mostly attributable to knowledge engineers "misunderstanding" the true domain knowledge models due to lack of deep expertise. Vulcan Inc. views this effort a necessary precondition to constructing the Digital Aristotle.

A number of knowledge acquisition tools, like Stanford's Protégé, currently exist and have several thousand registered users. These appear to be more oriented towards knowledge engineers and emphasize ontology or taxonomy formulation rather than full knowledge base formation. Vulcan anticipates a considerable amount of innovation in producing the necessary domain tools.

One approach Vulcan is currently considering is a document-rooted model, in which a document is loaded into the knowledge formulation application and the domain expert performs tasks to extract rules/axioms, entity and relation structure (ontology) and instance information from the text. Vulcan is actively seeking contributions for more ideas and insights in this area.

Efforts in phase II will also focus on making knowledge more accessible to users interested in posing queries to the systems. Tools will also be developed to enable naïve users, like high school seniors and college freshmen, to form queries.

The current state of natural language understanding (NLU) may not facilitate a pure NLU solution to this problem. The Vulcan team is currently exploring hybrid text and graphical solutions and is actively seeking new ideas on this front.

Acknowledgements

The authors would like to acknowledge the Halo Pilot teams for their contributions to the project and feedback on this paper. Additionally we'd like to thank Murray Burke and John Prange for their help and guidance.