Project Halo Update—Progress Toward Digital Aristotle

David Gunning,[1] Vinay K. Chaudhri,[2] Peter Clark,[3] Ken Barker,[4] Shaw-Yi Chaw,[4] Mark Greaves,[1] Benjamin Grosof,[1] Alice Leung,[5] David McDonald,[5] Sunil Mishra,[2] John Pacheco,[2] Bruce Porter,[4] Aaron Spaulding,[2] Dan Tecuci,[4] and Jing Tien[2]

1. Vulcan Inc.
2. Artificial Intelligence Center, SRI International, Menlo Park, CA, 94025
3. The Boeing Company
4. University of Texas at Austin, Austin, TX
5. Raytheon BBN Technologies Corporation

Abstract

In the winter, 2004 issue of AI magazine, we reported Vulcan Inc.'s first step toward creating a question-answering system called "Digital Aristotle." The goal of that first step was to assess the state of the art in applied knowledge representation and reasoning (KR&R) by asking AI experts to represent 70 pages from the advanced placement (AP) chemistry syllabus and to deliver knowledge-based systems capable of answering questions from that syllabus. This paper reports the next step toward realizing a Digital Aristotle: we present the design and evaluation results for a system called AURA, which enables domain experts in physics, chemistry, and biology to author a knowledgebase and that then allows a different set of users to ask novel questions against that knowledgebase. These results represent a substantial advance over what we reported in 2004, both in the breadth of covered subjects and in the provision of sophisticated technologies in knowledge representation and reasoning, natural language processing, and question answering to domain experts and novice users.

Introduction

Project Halo is a long-range research effort sponsored by Vulcan Inc., pursuing the vision of the "Digital Aristotle"—an application containing large volumes of scientific knowledge and capable of applying sophisticated problem-solving methods to answer novel questions. As this capability develops, the project focuses on two primary applications: a tutor capable of instructing and assessing students and a research assistant with the broad, interdisciplinary skills needed to help scientists in their work. Clearly, this goal is an ambitious, long-term vision, with the Digital Aristotle serving as a distant target for steering the project's near-term research and development.

Making the full range of scientific knowledge accessible and intelligible to a user might involve anything from the simple retrieval of facts to answering a complex set of interdependent questions and providing user-appropriate justifications for those answers. Retrieval of simple facts might be achieved by information-extraction systems searching and extracting information from a large corpus of text. But, to go beyond this, to systems that are capable of generating answers and explanations that are not explicitly written in the texts, requires the computer to acquire, represent, and reason with knowledge of the domain (i.e., to have genuine, internal "understanding" of the domain).

Reaching this ambitious goal requires research breakthroughs in knowledge representation and reasoning, knowledge acquisition, natural language understanding, question answering, and explanation generation. Vulcan decided to approach this ambitious effort by first developing a system capable of representing and reasoning about introductory, college-level science textbooks, specifically, a system to answer questions on advanced placement (AP) exams. (For details on the AP exam, see http://www.collegeboard.com/student/testing/ap/about.html)

Question answering has long challenged the AI field, and several researchers have proposed question answering against college-level textbooks as a grand challenge for AI (Feigenbaum, 2003; Reddy, 2003). Project Halo, described in this article, provides an essential component to meet that challenge—a tool for representing and using textbook knowledge for answering questions by reasoning.

As an initial, exploratory step toward this vision, Vulcan initiated the Halo Pilot in 2002—a six-month effort to investigate the feasibility of creating a scientific knowledgebase capable of answering novel questions from an AP (first-year, college-level) chemistry test. Three teams—SRI International, Cycorp, and Ontoprise—developed knowledgebases for a limited section of an AP chemistry syllabus. The knowledgebases could correctly answer between 30 and 50 percent of the associated questions from the AP test (Friedland et al., 2004).

While encouraging, these results had limitations. Only a small subset of knowledge, from one domain, was tested—leaving the question of how well the techniques would generalize to other material and other domains. Knowledge-representation experts, rather than domain experts, had encoded the knowledgebases, making largescale implementation impractical. Also, all test questions were translated manually from natural language into formal logic (also by knowledge-representation experts), not addressing the problem of question formulation by typical users.

In 2004, Vulcan initiated Halo Phase II with the goal of developing tools to enable Subject Matter Experts (SMEs), (e.g., chemists, biologists, and physicists), to formulate the knowledge and tools to enable less-experienced domain users, such as undergraduates in these disciplines, to formulate questions to query that knowledge. Again, multiple teams were awarded contracts to design and prototype knowledge-formulation and question-formulation tools suited for domain experts. The system that emerged as the best of these attempts, and the one described in the rest of this article, is the Automated User-Centered Reasoning and Acquisition System (AURA), which was developed by SRI International, the University of Texas at Austin, and the Boeing Company, with Prof. Bonnie John from Carnegie Mellon University serving as consultant.

In Halo Phase II, the goal was developing a software system that enabled domain experts to construct declarative knowledgebases in three domains (physics, chemistry, and biology) that could answer AP-like questions posed in natural language. The AURA team analyzed the knowledge-representation and question-answering requirements; crafted a user-centered design; implemented an initial system prototype; conducted an intermediate evaluation in 2006; developed a refined version of the AURA system; and conducted a final evaluation of the system in 2008 and 2009. This paper summarizes that system and its evaluation.

AURA System Development

The concept of operation for AURA is as follows: a knowledge-formulation (KF) subject matter expert (KFE), with at least a graduate degree in the discipline of interest, undergoes 20 hours of training to enter knowledge into AURA; a different person, a question-formulation (QF) subject matter expert (QFE), with at least a high-school-level education, undergoes four hours of training and asks questions of the system. Knowledge entry is inherently a skill-intensive task and, therefore, requires more advanced training in the subject as well as in using the system. A QFE is a potential user of the system, and we required less training for this position because we wanted as low a barrier as possible to system use.

We chose the domains of college-level physics, chemistry, and biology because they are fundamental hard sciences, and because they also stress different kinds of representations. The AP test was established as the evaluation criterion to assess progress. Textbooks were selected that covered the AP syllabus for physics (Giancoli, 2004), chemistry (Brown et al., 2003), and biology (Campbell & Reece, 2001). A subset of each AP syllabus was selected that covered roughly 60 pages of text and 15–20% of the AP topics for each domain. The AURA team was challenged to design and develop a system that could fulfill the above concept of operations for the selected AP material.

Overall Design and Requirements Analyses

The initial design requirements were determined by conducting a series of three analyses (Chaudhri et al., 2007; Chaudhri et al., 2010): 1) a domain analysis of textbooks and AP exams in the three domains; 2) a user-needs analysis of the domain expert's requirements for formulating knowledge; and 3) an analysis of a user's question-formulation requirements.

The domain analysis identified the four most-frequent types of knowledge representation needed in these three domains. These four types of knowledge contribute to answering approximately 50% of the AP questions (in order of importance):

- *conceptual knowledge:* representing classes, subclasses, slots, slot constraints, and general rules about class instances
- *equations:* a majority of questions in physics and some questions in chemistry involve mathematical equations
- *diagrams:* all three domains make extensive use of diagrams
- *tables:* often used to show relationships not repeated elsewhere in text

A knowledge-formulation system was designed to accommodate these four knowledge types, but the module for diagram knowledge has not yet been implemented. Subsequent analyses were conducted to catalog the additional KR&R challenges in each domain that will be discussed later.

The user-needs analyses showed three main areas of concern for knowledge formulation by domain experts who are not trained in KR&R: 1) knowing where to begin is often challenging for domain experts (the *blank slate* problem); 2) knowledge formulation consists of a complete lifecycle that includes initial formulation, testing, revision, further testing, and question answering; and 3) the system should place a high value on usability to minimize required training.

The users asking questions are different from the users who enter knowledge, and the training requirements must be kept minimal because we cannot assume that the QFE will have an intimate familiarity with the KB or the knowledge-formulation tools. Because the QFE must specify a wide variety of questions, including problem-setup scenarios in some questions, we could not use a rigid interface; instead, we adopted an approach based on natural language input.

We analyzed the English text of AP questions in all three domains (Clark et al., 2007). The language of science questions involves a variety of linguistic phenomena. We identified 29 phenomena and their frequency of occurrence (Clark et al., 2007). For example, approximately 40% of questions used direct anaphora, 50% used indirect anaphora, and 60% used prepositional phrases. This data served as the basis for the question-formulation language design of AURA.

For the current phase of development, we consciously chose to not leverage any methods for automatic reading of the textbook for the following reasons: First, we expected the system challenges to be significant without introducing a language-understanding component. Second, for the detailed knowledge representation and reasoning (KR&R) needed to answer AP questions in all three domains, we did not expect any automatic technique to approach the needed representation fidelity. Finally, for knowledge that involves computations and diagrams as in physics and chemistry, we did not expect fully automatic methods to be very effective. The AURA architecture does include provisions to import information from external sources, such as semantic web sources or well-developed ontologies, that might have been created automatically (Chaudhri, Greaves, Hansch, Jameson, & Pfisterer, 2008).

AURA System Architecture

The AURA system has three broad classes of functionality: knowledge formulation (KF); question formulation (QF); and question answering (QA). In addition, there is a training program for both KF and QF, which was developed over several years of experience training domain experts for both roles. In Figure 1, we show the overall system architecture.

Knowledge Representation and Reasoning

AURA uses the Knowledge Machine (KM) as its core knowledge representation and reasoning engine (Clark & Porter, 1999), a powerful, mature, frame-based knowledge-representation system. Though KM is comparable to many state-of-the-art representation and reasoning systems, there are two features that are distinctive and have played a special role in AURA: prototypes and unification mapping (or UMAP).

A *prototype* is a set of axioms in the form of a description of each of a set of interconnected individuals. Prototypes are in a form that mirrors the structure of a concept map (i.e., a graph of descriptions of interconnected individuals). In AURA, a prototype eases the task of synchronizing the user-interface representation of concept maps as they are created and modified by a user with the knowledgebase representation. It also provides a way to acquire a group of axioms together instead of acquiring one rule at a time. We will give a concrete example of a prototype in a later section.

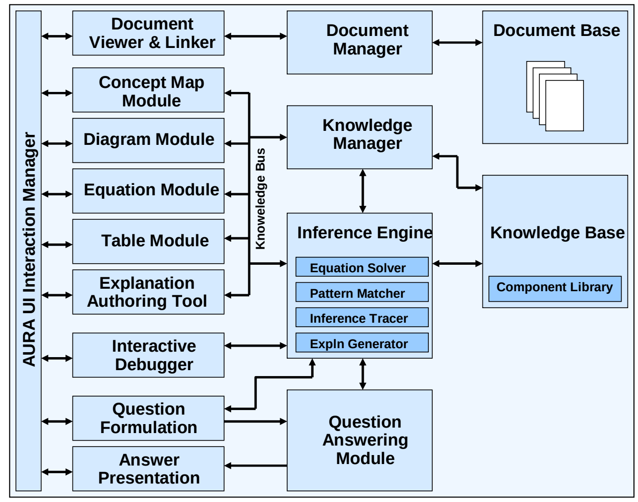

Figure 1. AURA System Architecture

Unification Mapping or UMAP is a method for equating two objects that may heuristically infer additional equalities as a side effect. Because of UMAP, a KFE need not be as complete in specifying a knowledgebase as a highly trained logician might be. The heuristics in UMAP are designed to draw plausible inferences in an under-specified knowledgebase, filling in details that a KFE might leave out. We give an example of the use of UMAP in the next section.

Both prototypes and UMAP were first used in the context of a system called SHAKEN, which was developed as part of DARPA's Rapid Knowledge Formation program (Clark et al., 2001). The positive result from this prior work was the basis for including them as a central design feature in the AURA system.

Knowledge Formulation

Our approach to knowledge formulation (KF) includes three salient features inform: 1) the use of a document as a starting point and context for all knowledge entry; 2) a pre-built library of components that provides the starting point for any KF process; and 3) the choice of user-interface abstractions that are driven by a usability analysis and the native representations of knowledge within a textbook. We discuss each of these aspects of KF in greater detail.

We embed an electronic copy of each of the three textbooks into the user interface of AURA to serve two purposes: First, it helps specify the context and the scope of the knowledge to be entered. Second, a semantic search facility based on WordNet mappings from words in the document to concepts in the KB serves as the basis of making suggestions for concepts relevant for encoding that word.

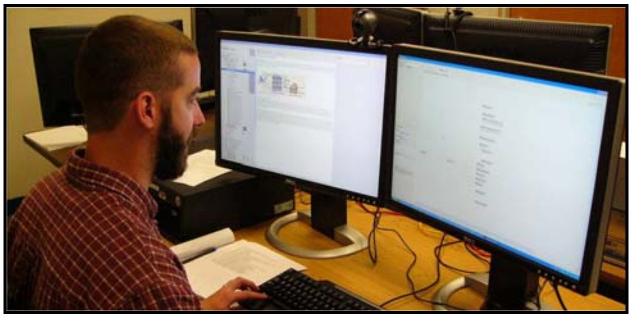

Figure 2. A domain expert working with AURA

The KFEs build their KBs by reusing representations in a domain-independent KB called the Component Library or CLIB (Barker, Porter, & Clark, 2001). The Component Library is built by knowledge engineers (KEs) and contains domain-independent classes such as *Attach*, *Penetrate*, *Physical Object*; predefined sets of relations such as *agent*, *object*, *location*; and property values to help represent units and scales such as *size* or *color*. These classes and relations and their associated axioms provide a starting point to the KFEs in the KF process. A selection of top-level classes in CLIB is shown in Figure 3.

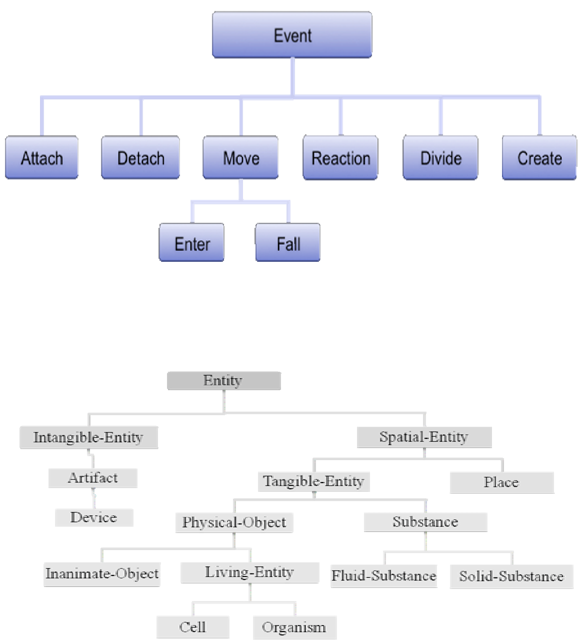

Figure 3. The Top-Level Event and Entity Classes in CLIB

To capture the most frequently occurring knowledge types identified earlier, we settled on the following user-interface elements: directed graphs for structured objects (concept maps) and logical rules and equations for mathematical expressions. To enhance the usability of the system, we implemented interfaces for chemical reactions and tabular data. We expect that this capability will enable users to encode knowledge sufficient to answer approximately 50% of the AP questions in all three domains. A detailed account of these choices and the underlying theory is available elsewhere (Chaudhri et al., 2007).

As an example, in Figure 4, we show a (simplified) representation of the concept of a Eukaryotic Cell. The node labeled as *Eukaryotic-Cell* is the root of the graph and is a prototypical individual of that class. The grey nodes represent non-root individuals in the graph; the unboxed words such as *has-part* are relations between individuals and are shown as the labels on the edges. Logically, the graph denotes a collection of rules that assert that for every instance of *Eukaryotic-Cell*, there exist instances of each node type shown in this graph, and that they are related to each other using the relations in the graph. Within AURA, this graph is represented as a prototype and is equivalent to the following rule:

```
(forall ?c
  (=> (instance-of ?c Eukaryotic-Cell)
    (exists ?x ?y ?z
      (and
        (instance-of ?x Nucleus)
        (instance-of ?y Chromosome)
        (instance-of ?z Plasma-Membrane)
        (has-part ?c ?x) (has-part ?c ?y)
        (has-part ?c ?z) (is-inside ?y ?x)))))
```

From a logical point of view this rule could be broken into multiple rules, for example, each rule stating the existence of a part, and another rule stating their relationships. The prototypes combine multiple rules into a single rule to provide a coarser granularity of knowledge acquisition. Abstraction offered by prototypes, and the fact that a prototype mirrors the structure of a concept map as seen by a user, contributed to enabling the domain experts to author knowledge.

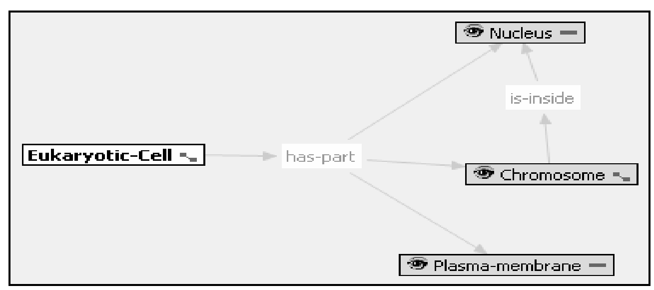

Figure 4. A Biology Concept for the Eukaryotic Cell

As an example of a process in biology, in Figure 6, we show a (simplified) concept map for Mitosis. This concept map shows the different steps in Mitosis (Prophase, Metaphase, etc.), their relative ordering, and that its *object* is a Diploid Cell and its *result* is two Diploid Cells. The numbers shown next to a lock symbol in the relations, such as *result*, represent the cardinality constraints. For example, the *result* of Mitosis is exactly two Diploid Cells. The current AURA system supports such declarative descriptions and reasoning about processes, but does not currently support running process simulations.

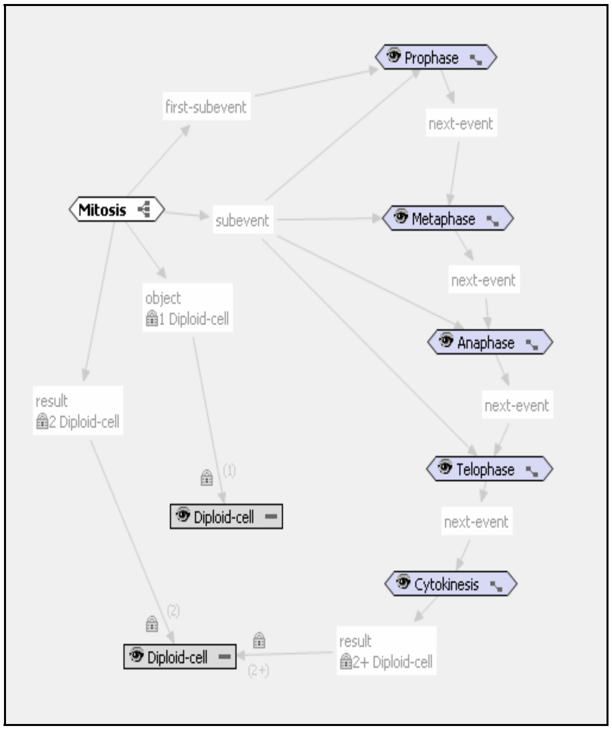

Figure 5. A Biology Concept Representing Mitosis

The KFEs create the concept maps using four primary graph-manipulation operations: 1) adding a new individual to a graph; 2) specializing an individual to be an instance of a more specific class; 3) connecting two individuals using a set of pre-defined relations; and 4) equating two individuals. Equating two individuals uses the UMAP. As an illustration of UMAP, in Figure 6, we show the concept of *H2O* (or water) from chemistry. The top part of this graph encodes that every instance of *H2O has-part* an *OH-* ion and *H+* ion, and further an *H+* ion *has-atom H*. The lower part of the graph shows another *H2O* individual that is added to this graph. If the user equates the two *H2O* individuals in this graph, the *UMAP* operation will recursively equate the *H+*, *OH*-that are related by *has-part* and *H* that is related by the *has-atom* relation. This inference is heuristic and plausible. For this inference to follow deductively, the KFE would need to encode cardinality constraints on *has-part* and *has-atom* relations. UMAP can draw equality inferences even when the KB is underspecified in that the cardinality constraints are not specified. In some cases, all the cardinality constraints are not known; in other cases, adding cardinality constraints may be incorrect. The ability of UMAP to work with such under-specification in the knowledgebase substantially contributed to the usability of the concept map-editing interface of AURA.

As a final example of a concept formulated using AURA, in Figure 8, we show a concept map for *Free Fall*. The concept map encodes different properties of *Free Fall*, and the mathematical equations that relate them. The property values are shown in green ovals, and the mathematical equations are shown in green squares. AURA supports a WYSIWIG editor for entering equations, and the equations can be related to properties that are represented in the knowledgebase.

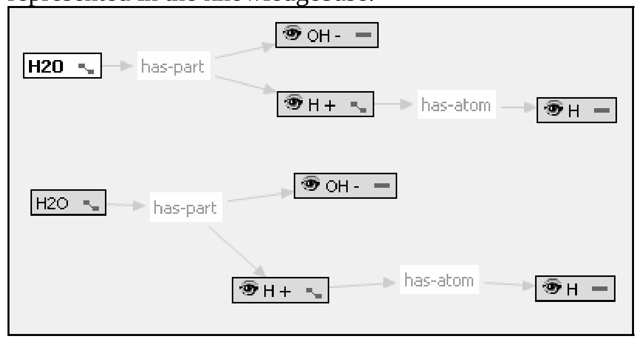

Figure 6. The Use of UMAP on Two Entities Recursively Equates All Its Parts

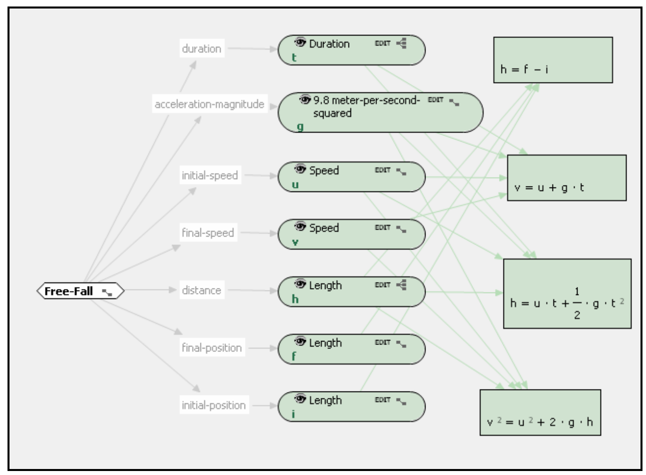

Figure 7. A Physics Concept of Free Fall.

We have designed a training course for KFEs that prepares them to enter knowledge into AURA. The current KF training is approximately 20 hours. The training introduces the KFEs to the mechanics of using the system and to basic knowledge-engineering principles. In the knowledge-engineering section of the training, the KFEs learn about different classes and relations in CLIB, and how to use them. The training program includes several hands-on exercises in which KFEs encode knowledge and are given feedback on their specific choices. The core of the training program is common across all three domains. There are, however, several domain-specific modules. For example, physics KFEs must learn to properly use vector math, which does not arise in the other two domains. For chemistry, the KFEs must learn about entering chemical compounds and reactions, and about chemistry-specific, system-available knowledge. For biology KFEs, there is an added emphasis on learning about describing processes.

Question Formulation

Recall that the users asking questions are different from the users who enter knowledge, and that the training requirements must kept low. Further, we cannot assume that the QFE will have an intimate familiarity with the KB or the knowledge-formulation tools. Our question-formulation design aims to account for these requirements.

While there has been considerable recent progress in question answering against a text corpus (e.g., Voorhees and Buckland, 2008), our context is somewhat different, namely posing questions to a formal KB, where a complete, logical representation of the question is needed for the reasoner to compute an answer. In this context, the designer is typically caught between using "fill-in-the-blank" question templates (Clark et al., 2003), which severely restricts the scope of questions that can be posed, or attempting full natural language processing on questions, which is outside the reach of the current technology. In AURA, we have aimed for a "sweet spot" between these two extremes by using a controlled language (a simplified version of English) called CPL (Computer-Processable Language) for posing questions, with feedback mechanisms to help in the question-formulation process. Our hypothesis is that a controlled language such as CPL is both easily usable by people and reliably understandable by machines and that, with a small amount of training and good runtime feedback mechanisms, users can express their questions easily and effectively in that form.

A basic CPL sentence has the form

subject + verb + complements + adjuncts

where complements are obligatory elements required to complete the sentence, and adjuncts are optional modifiers. Users follow a set of guidelines while writing CPL. Some guidelines are stylistic recommendations to reduce ambiguity (e.g., keep sentences short, use just one clause per sentence), while others are firm constraints on vocabulary and grammar (e.g., words of uncertainty (e.g., "probably," "mostly," are not allowed, not because they cannot be parsed but because their representation is outside the scope of the final logical language)). Examples of typical AP questions from the three domains, and a typical reformulation of them within CPL, are shown in Figure 8. As shown, questions (especially in physics) may be multiple sentences divided into a "setup" describing a scenario and a "query" against that scenario. Multiple-choice questions are re-expressed in CPL as separate, full-sentence questions.

To pose a question, the user first enters a CPL form of it in the interface. If a CPL guideline is violated, AURA responds with a notification of the problem, and advice about how to rephrase the question. If this happens, then the user rephrases the question, aided by a searchable database of example questions and their CPL equivalents, and a list of the vocabulary that CPL understands, and the process repeats. Alternatively, if the question is valid CPL, then AURA displays its interpretation in graphical form for the user to validate. An example of this graphical form is shown in Figure 9, depicting how AURA interpreted the first example in Figure 8 in terms of individuals, relationships, and the focus of query (denoted by a question mark). If the interpretation appears incorrect then the user would again rephrase the CPL to correct the problem. The graphical interface also allows a user to perform a limited number of edits, for example, changing the relation or asserting that the two nodes are equal. Otherwise, the user instructs AURA to answer the question invoking the query answering described in the next section.

Example 1 (Physics) : Original Question:
A car accelerates from 12 m/s to 25 m/s in 6.0 s. How far did it travel in this time?
Reformulation in CPL:
A car is driving.
The initial speed of the car is 12 m/s.
The final speed of the car is 25 m/s.
The duration of the drive is 6.0 s.
What is the distance of the drive?

Example 2 (Chemistry) : Original Question:
What two molecules must always be present in the products of a combustion reaction of a hydrocarbon compound?
Reformulation in CPL:
What are the products of a hydrocarbon combustion reaction?

Example 3 (Biology) : Original Question:
Crossing over occurs during which of the following phases in meiosis? a. prophase I; b. ...[etc]…?

Reformulation in CPL:
Does crossing over occur during prophase I?

Figure 8. Example Questions Reformulated in CPL

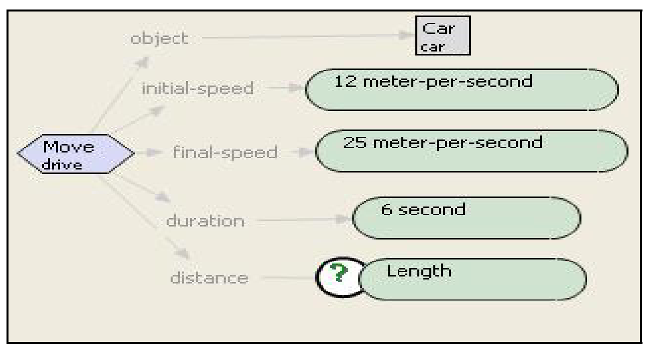

Figure 9. Graphical Feedback during Question Formulation

Let us now consider how this design meets the requirements of the QFEs. The CPL formulations expected of QFEs are in terms of English words and, thus, do not require intimate knowledge of the KB's vocabulary. To read the interpretation graph, the QFEs must understand the meaning of the concepts and relations. Through AURA, the QFEs can access the documentation of the classes and relations, and a vocabulary list of all classes and relations known to the system. The task of understanding the terms of the KB by inspection is significantly easier than using those terms for creating new concepts as the KFEs are required to do. CPL also allows QFEs to construct problem scenarios with respect to which a question is asked.

Question Answering

Once a question has been formulated to a user's satisfaction, AURA attempts to answer it. Conceptually, the question-answering module of AURA has four functional components: reasoning control, a reasoning engine, specialized reasoning modules, and explanation generation.

The reasoning control relates the individuals in the question interpretation to the concepts in the KB, identifies the question type, and invokes the necessary reasoning. In some cases, relating an individual to a class in a KB is straightforward, especially as AURA allows KFEs to associate words with the concepts that they create. In other cases, AURA must resort to specialized reasoning based on search and semantic matching (Clark et al., 2007; Chaw et al., 2009).

A question type denotes a style of formulation and reasoning used for answering a question. Currently supported question types are: computing a slot value, checking if an assertion is true or false, identifying superclasses, comparing individuals, describing a class, computing the relationship between two individuals, and giving an example of a class.

AURA uses the Knowledge Machine (KM) as its core reasoning engine. AURA has a special purpose reasoning module for solving algebraic equations that is used extensively both in physics and chemistry. It has a graph-search utility to support the question type that computes relationships between two individuals. There is a chemistry-specific module aimed at recognizing chemical compounds and reactions, and a physics-specific module to support vector arithmetic.

Finally, AURA supports an incremental explanation system that produces explanations in (rudimentary) English. Some of the terms in the explanation are hyperlinked, and the user can drill down to obtain more information. As an example, in Figure 10 , we show the answer to the question shown as Example 1 in Figure 10.

AURA first presents an answer to the question (*s=111 m*) followed by the explanation. In the explanation, AURA shows the equation and specific variables used to solve the equation. In more complex questions that use more than one equation, the explanation includes the specific order in which the equations are applied.

In Figure 11, we show an example answer to a chemistry question that was shown earlier as Example 2. The answer shows that the reactants of a combustion reaction include a chemical and oxygen gas. As a final example, we show the answer to the Example 3 considered earlier. The answer for this question shows that, indeed, crossing over happens during Prophase-I. The phrases such as "the crossing-over of the DNA'' are generated using the rudimentary English generation facility in the system.

Answer

s = 111 m

Explanation

motion-with-constant-acceleration : A move of an object such that the acceleration of the object is constant throughout the move.

Given:

- v_1 = 25 m/s [the speed of the final-velocity]
- v_2 = 12 m/s [the speed of the initial-velocity]
- v_3 = 12 m/s [the speed of the initial-velocity]
- v_4 = 25 m/s [the speed of the final-velocity]
- t = 6.0 s [the duration of the motion-with-constant-acceleartion]
- u = 12 m/s [the speed of the initial-velocity]
- v_4 = u + a * t [the speed of the final-velocity]

Solving for s ... $s = u * t + ((1 / 2) * a) * t^2$
$a = (v_4 - u) / t$ [solving ($v_4 = u + a * t$) for a]
.:. a = 2.2 m/s^2 .:. s = 111 m
Therefore, the distance of the motion-with-constant-acceleartion (s) = 111 m

Figure 10. Answer to an Example Physics Question

Answer

water and carbon dioxide

Explanation

hydrocarbon-combustion-reaction : When hydrocarbons are combusted in air, they react with O2 to form CO2 and H2O. The number of molecules of O2 required in the reaction and the number of molecules of H2O and CO2 formed depend on the composition of the hydrocarbon, which acts as the fuel in the reaction.

Figure 11. Answer to an Example Chemistry Question

Answer

find more answers

Yes.

It is true that the crossing-over of the dna is a subevent of the prophase-i

Figure 12. Example Answer to Biology Questions

AURA Evaluation

We conducted a full user evaluation to find out how well AURA enables graduate students in the three domains (physics, chemistry, and biology) to construct knowledgebases that are able to answer AP-like questions posed by undergraduates.

To ensure that the assessment was independent and unbiased, Vulcan contracted BBN Technologies to design and run the evaluation. BBN teamed up with Larry Hunter at the Medical School of the University of Colorado at Denver. The evaluation was designed to answer four main questions:

- How well does AURA support Knowledge Formulation (KF) by domain experts?
- How well does AURA support Question Formulation (QF) by domain experts?
- How good are AURA's Question Answering (QA) and Explanation Generation?

Experimental Design

To address the experimental questions, three sets of experimental conditions were evaluated:

- Expert versus non-expert KF experience:

  The expert condition was represented by individuals highly knowledgeable in their domain, with significant training and previous experience using AURA, working in close collaboration with the members of the AURA team, over many months.

  The non-expert condition was represented by individuals qualified in their domain at a graduate school level, with limited training (20 hours) and no previous experience using AURA, working independently for a limited amount of time (approximately 120 hours) over a four week period.
- Expert versus non-expert QF experience:

  The expert condition was represented by the same SMEs as in the expert KF condition.

  The non-expert condition was represented by individuals qualified in their domain at an undergraduate level, with limited training (4 hours) and no previous experience using AURA.
- Question familiarity and difficulty:

  A set of *Reference Questions* was developed in each domain by SRI. These questions were known to AURA development team and available at KF time. These questions were used by SMEs to test their knowledge as it was entered.

  A set of *Novel Questions* was developed by BBN specifically for the evaluation. These were not known to the AURA development team and were not available at KF time. They were used only during the QF evaluations of the newly developed KBs.

  A subset of *Selected Novel Questions* was chosen from the set of all Novel Questions as an experimental control variable. The choice was made in a way that AURA was able to answer a large fraction of these questions but not all of them. This was done to avoid *floor* and *ceiling* effects while comparing results.

Experimental procedure

The main steps in the test procedure were:

1. The AURA team selected the textbook sections and AP syllabus for each domain.
2. Expert SMEs of the AURA team authored knowledgebases for the selected textbook sections and AP syllabus, testing the knowledge against the Reference Questions. These SMEs worked closely with the development team.
3. Experienced AP teachers recruited by BBN generated the set of Novel Questions in each domain to cover the topics in the selected syllabus.
4. Expert SMEs at SRI formulated and asked the set of Novel Questions of their expert KBs.
5. BBN and SRI chose 50 Selected Novel Questions in each domain that best matched AURA's implemented reasoning capabilities.
6. SRI trained the non-expert KFEs recruited by UC Denver in the use of AURA for knowledge formulation in a 20-hour training course.
7. The non-expert KFEs at UC Denver authored knowledge over a four-week period (using approximately 120 hours of KF time).
8. SRI trained the non-expert QFEs in the use of AURA for question formulation in a 4-hour training course.
9. For each expert-formulated and non-expert-formulated KB, one or more QFEs from the same domain asked selected novel questions.
10. BBN scored the results by submitting the question formulation and answering transcripts to two independent AP teachers for grading. The graders were different from the AP teachers who were used in step 3 to design the questions.

Science Textbooks and Syllabus

The following textbooks were used for the three domains:

- *Biology*. Campbell, Neil, 6th Edition
- *Chemistry: The Central Science*. Brown, Theodore, 9th Edition
- *Physics: Principles with Applications*. Giancoli, Douglas, 6th Edition.

The AURA syllabus was selected to represent a set of key concepts within the AP curriculum in each domain. The syllabus was necessarily limited so that it would present a manageable amount of knowledge to be encoded yet included enough material to support a significant number and variety of questions. The main topics and approximate page count are shown below in Table 1.

Table 1. AURA Syllabus

	Main topics	Pages
Biology	Cell structure, function, and division; DNA replication; protein synthesis	44
Chemistry	Stoichiometry; chemical equilibria; aqueous reactions; acids and bases	67
Physics	Kinematics; Newtonian dynamics	78

There were significant differences in the information content of the selected pages and how well they covered the full AP syllabus in each domain. In biology, the selected 44 pages covered 23% of the full syllabus, in chemistry, 67 pages covered 11% of the full syllabus, and in physics, 78 pages covered 15% of the full syllabus.

Test Subjects

The expert SMEs consisted of three domain experts, one in each domain, each with at least a graduate degree in the respective discipline. These SMEs had worked with the AURA team throughout the development process and, though still primarily domain experts, had become very familiar with AURA and its knowledge-engineering process.

The non-expert KFEs consisted of nine students, three in each domain, recruited from the Denver area, through the University of Colorado at Denver, where the non-expert KF experiment was conducted. Subjects were recruited and screened with an abbreviated AP-level exam to ensure domain knowledge. The participants were mostly graduate students or graduates, with one advanced undergraduate. They were all computer literate, with a range of previous computer experience, but none had studied artificial intelligence, knowledge representation, or used AURA before.

The non-expert QFEs consisted of 16 (six in biology and five each in chemistry and physics) undergraduates or very recent graduates, who were recruited in the Boston area, through BBN, where the non-expert QF experiment was conducted. Participants were considered qualified in their domain if they 1) had passed a first-year university course that covered the AP curriculum with an A or B grade or 2) had passed the AP exam with a score of 4 or 5 during the previous three years. None had prior experience with AURA.

It should be noted that the QFEs were aware of the correct answers to the questions, and thus could recognize when the system had produced the correct answer, a somewhat unnatural situation compared with use "in the wild." The results thus represent an upper bound on the performance that one might expect with a more natural class of users, who are less knowledgeable about the domain and the questions.

Data Results and Analysis

First, we look at the question-answering performance of the knowledgebases authored by the expert SMEs (see Figure 13). In biology and physics, the expert KBs correctly answered more than 70% of the Reference and Selected Questions and more than 40% of all Novel Questions. The expert chemistry KB did not perform as well, especially for Novel Questions with a score of 18% for all Novel Questions and 44% for Selected Novel Questions. Because the selected set was artificially constructed for experimental control, the score on the selected questions should not be interpreted as an indication of the overall performance of the system. The score on the selected questions is shown in Figure 14 as this number is used in later graphs for comparative analysis across different experimental situations. There were two reasons for the low scores in chemistry: The expert KFE over-tuned the KB to the set of reference questions and did not provide good coverage of the syllabus for novel questions. Plus, the current version of AURA does not support a facility to author procedural knowledge, which was required for some questions.

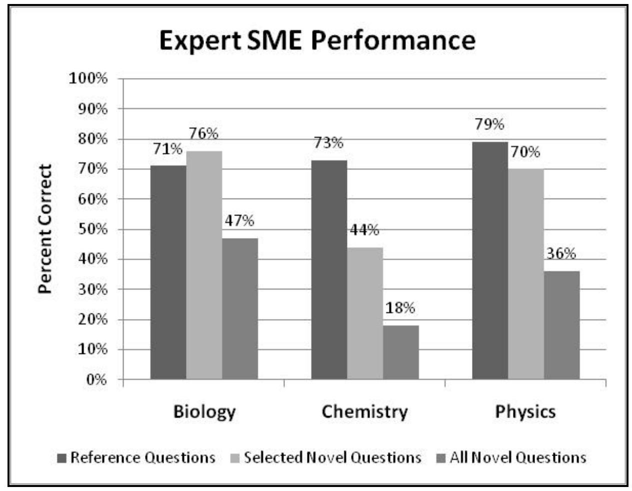

Figure 13. Expert SME Performance

Second, we look at how the non-expert SMEs did in comparison to the experts. The experimental design produced a 2x2 comparison of expert vs. non-expert performance for both KF and QF. To understand the 2x2 aspect of the experiment design, we can interpret the four points shown in Figure 14 as follows: the upper-left point represents the question-answering correctness score when the knowledge was formulated by an expert KFE, but the questions were asked by a non-expert QFE; the lower-left point represents the situation when the knowledge was formulated by a non-expert KFE, and the questions were also asked by a non-expert QFE. The other two points can be analogously interpreted. To see the effect of question-formulation expertise, the graph should be read left to right; to see the effect of knowledge formulation expertise, the graph should be read top to bottom.

Thus, for biology (Figure 14), we can see the effect of knowledge-formulation expertise by observing that the KBs authored by expert KFEs always had better scores than the KBs authored by non-expert KFEs. We can see the effect of the question-formulation expertise by reading the graph left to right and noticing that question-formulation expertise had no effect for KBs that were authored by expert KFEs. But for KBs authored by non-expert KFEs, the non-expert QFEs outperformed the expert QFEs. This is an anomaly, where it appeared that the non-expert QFEs outperformed the expert KFEs by 20%. Further analysis revealed that much of this difference resulted from the non-expert SMEs being less rigorous in how they formulated questions, and so we discount this difference as poor experimental control.

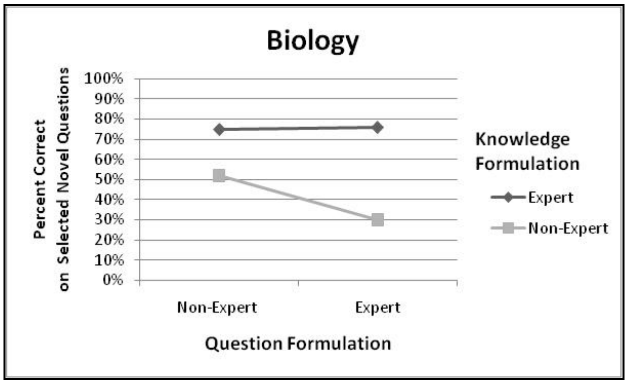

Figure 14. Expert vs. Non-Experts in Biology

In chemistry (Figure 15), there were no significant differences among the four conditions. Expert vs. non-expert KF was equivalent as was expert vs. non-expert QF.

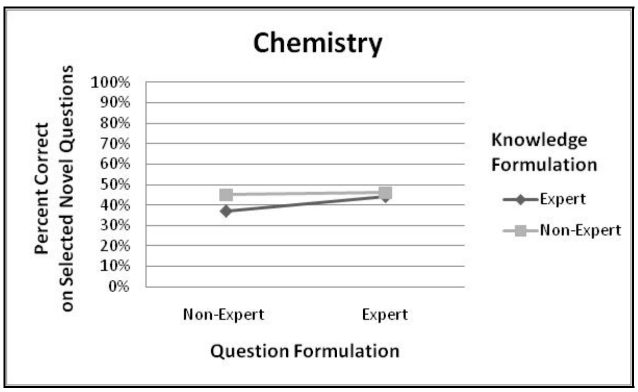

Figure 15. Experts vs. Non-Experts in Chemistry

In physics (Figure 16), experts outperformed non-experts in both KF and QF. Physics is the only domain where the experts outperformed non-experts at QF. Physics questions were generally more complex to formulate because the formulations included several statements to describe the problem setup as well as language simplifications. The questions that involved specifying vector quantities were especially challenging for the non-expert QFEs to formulate. An obvious next question is to explain the reason for the differences between expert and non-expert conditions for each of the three domains.

For chemistry, our analysis of the results suggested that the results were confounded by a *floor* effect. Recall from Figure 13 that the expert-authored KBs scored only 18% on the novel questions. This significantly limited the kinds of questions that could be put in the selected set of questions considered in the experiment reported in Figure 15. The newly trained KFEs were able to perform as well as the expert KFEs, because the score of the expert KFEs was too low to start with.

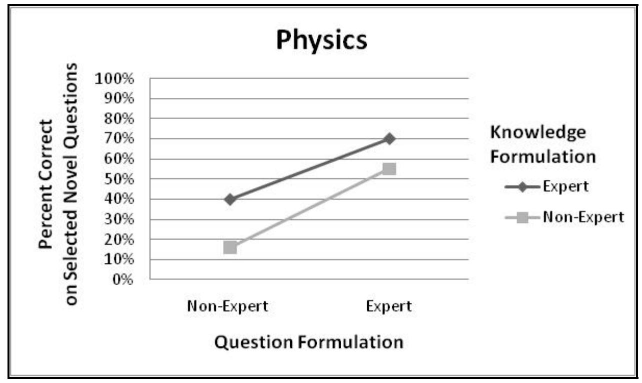

Figure 16. Experts vs. Non-Experts in Physics

The results for physics were easier to explain because there are known limitations of the system that make it harder for the KFEs to formulate knowledge about forces, and limitations in the inference technique to answer questions that may lead to a very large search space.

For biology, the situation was the most complex. Our initial hypothesis for this difference was that it was due to difference in the knowledge-entry time given to the expert KFEs and non-expert KFEs. The expert KFEs for biology had worked on their KB for about 600 hours whereas the evaluation experiment had allowed them to work for only 120 hours. Based on the progress the KFEs had made and the review of their KBs, we believe that the knowledge-entry time was not an issue for physics and might have had minor impact for chemistry. Based on this analysis, we designed a follow-up experiment only for biology to assess the effect of the knowledge-entry time on the question-answering performance.

In the follow-up experiment, one expert KFE was asked to create the same biology KB, but was limited to 120 hours for knowledge-entry time. One of the better performing non-expert KFEs was given an additional 180 hours, thus giving them a total of 300 hours, to continue authoring and refining their KB. We show the result in Figure 17. When expert was limited to 120 hours of KF time and the non-expert was allowed 300 hours, the two KBs exhibited similar performance with 60% correct answers. The additional 180 hours of KF time improved the non-experts score form 21% to 60%. The subject reported that the extra time gave her a much better understanding of AURA, the knowledge-entry process, and her KB.

This result shows a steep improvement in the performance of a KB authored by a newly trained KFE as the knowledge-entry time increased from 120 hours to 300 hours. The corresponding rate of improvement for an expert KFE as they are given more knowledge-entry time is much smaller. This is quite likely because the expert KFE has already reached a high level of performance, and the marginal value of additional knowledge-entry time toward question-answering performance diminishes. The most important conclusion that followed from this follow-up study was that given additional experience with the system, a KB authored by a newly trained KFE significantly improves in question-answering performance, and starts to approach the performance of an expert KFE. This was an excellent result in support of AURA's ability to enable a newly trained KFE to author competent knowledgebases.

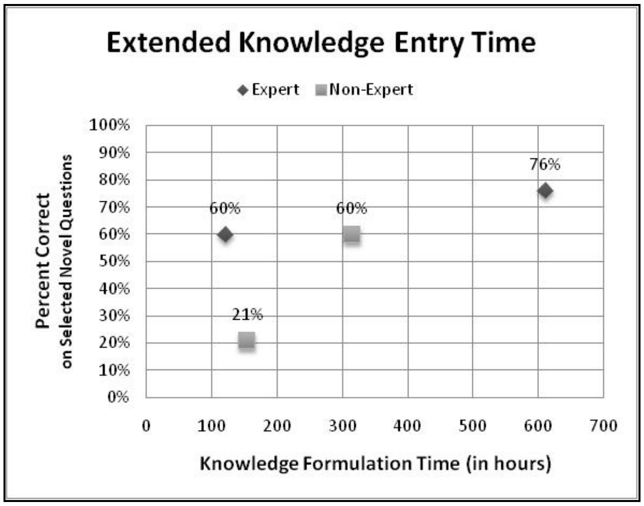

Figure 17. Results of Extended KF in Biology

Let us now return to the questions that this evaluation set out to answer. First, we consider the question: "how well does AURA support KF by domain experts?" The evaluation results show that for biology, a newly trained KFE can construct knowledgebases that, given sufficient knowledge-entry time, approach in performance to the performance of the knowledgebases constructed by expert KFEs. For physics, the KBs constructed by expert KFEs outperform the KBs constructed by newly trained KFEs. For chemistry, while the results show that the performance of the KBs authored by newly trained KFEs was very close to the KBs authored by expert KFEs, we believe this result to be confounded by the floor effects in the experimental data.

Second, we consider the question: "how well does AURA support QF by domain experts?" The results show that most non-expert QFEs in the domains of biology and chemistry were able to perform question formulation as effectively as experts KFEs after only four hours of training. The non-expert users in physics had some difficulty in posing the questions.

Third, we address the question: "how good is AURA's question answering performance?" The results show that AURA was able to answer significant numbers of AP-level difficulty questions in the domains of biology and physics, reaching or nearly reaching performance needed for a passing score on the AP test. We conclude that, with some caveats, the goal of comfortable use of AURA with minimal training has been met for question formulation, and for knowledge formulation it is well advanced.

Multi-User Knowledge Entry Using a Team of KFEs

A major lesson from the evaluation results reported above was that the capabilities of AURA in enabling knowledge formulation and question formulation for biology were well advanced while some challenges remain in other domains. Based on that assessment, a natural scaling question was to undertake some preliminary work to support the construction of a KB from a full biology textbook.

The experiment results reported earlier involved only one user working in isolation in constructing a KB. Such a constraint was an artifact of a controlled experiment and is no longer practical when a KB is developed by a team of domain experts. So, as a step toward scaling to a knowledgebase for a full biology textbook, we devised a pilot experiment to answer the following questions: "Can we replicate the training and knowledge-entry process by teaching it to professionals external to the AURA development team?"; and "Can a team of experts collaborate to create a shared knowledgebase of a scope similar to what was created in the controlled experiment?"

To address these questions, SRI teamed with an organization based in India to organize a multi-user knowledge-entry experiment (MUKE). Two knowledge-engineering professionals from the MUKE team came to SRI and underwent a "trainers training." The trainers training included the training designed for KFEs as well as in-depth exposure to AURA. These knowledge-engineering professional returned to their parent organizations and delivered the AURA training to a team of three biologists.

The current AURA system has no software support for multi-user knowledge entry. We designed a collaboration process external to AURA that the team of biologists could use for knowledge entry. The process defined specific roles for the members of the team as *contributors* and *integrators.* The contributors developed representations for the portion of a syllabus, and *an integrator* combined the contributions into an integrated whole. The combined knowledge-entry time of the three-member biologist team was comparable to the sum total of the knowledge-entry time of the three biologists who had participated in the controlled experiment reported earlier. The team collaboratively constructed the KB for the same syllabus, and using the same set of test questions. The three-person SME teams were explicitly directed to work together to discuss, partition, and collaborate in performing the knowledge-entry tasks.

The KB produced by the team was tested on the identical set of novel questions that was used in the controlled study. The results are shown in Figure 18.

Let us now discuss how these results answer the questions that we set out to answer. We first address: "Can we replicate the training and knowledge-entry process by teaching it to professionals external to the AURA development team?" Given that the knowledge-engineering professionals of an organization external to AURA development team could learn the AURA training and deliver it to the biologists who constructed KBs that performed very closely to those constructed by SRI's expert KFEs suggests that we could successfully replicate the knowledge-engineering process. Initially, the AURA development team needed to provide constant support to the knowledge engineers from the MUKE team; but, such need significantly dropped during the exercise.

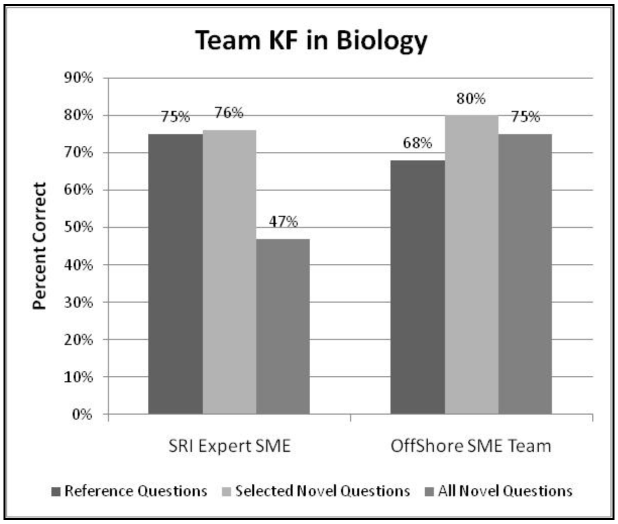

Figure 18. Multi-user KF Team Results

Second, we address the question: "Can a team of experts collaborate to create a shared knowledgebase of scope similar to what was created in the controlled experiment?" Here again, we believe that the MUKE team succeeded as the correctness scores on their knowledgebases were comparable to the scores on the ones authored by the expert KFEs at SRI.

Finally, because the score on the all novel questions on the KB produced by the MUKE team (75%) is much higher than the corresponding score on the KB produced by the expert KFEs (47%), one can naturally ask, "Did MUKE team outperform the expert KFEs at SRI?" We believe that the experiment design does not support such conclusion because the knowledge-entry process, resources, and conditions for the KBs authored by the SRI expert KFEs were significantly different from those used by the MUKE team.

Discussion

The results demonstrate significant progress since the Halo Pilot in 2004. We now have SME-authored KBs achieving question-answering scores of 70% in many conditions. Non-expert SMEs, with light training in AURA, can create KBs that achieve scores of 60% when given a similar amount of knowledge-entry time as the expert SMEs. Even non-expert SMEs with light training and limited entry time achieve scores in the 40–50% range, equivalent to the scores achieved in the Halo Pilot by AI experts. The multi-user knowledge-entry results were very encouraging—demonstrating that a dedicated KF team of domain experts can author a biology KB that achieved a score of 75%, even for novel questions.

However, the results also demonstrate remaining challenges. In general, question-answering performance drops when the KBs are presented with novel questions that the knowledge formulator did not specifically prepare the KB to answer. Sometimes, this drop is dramatic, even for the expert KFEs. The knowledge capture and reasoning capabilities are still incomplete because none of the SMEs, not even the expert SMEs, could create KBs that performed above the 80% level, even for the reference questions that were known in advance.

Moreover, the danger of over-optimizing a system to perform well on a specific test problem always exists—in ways that do not generalize to the real-world problem. Because we rigorously focused the Halo work on this particular AP question-answering task, there is certainly that danger here. AP exams generally test only a special band of conceptual knowledge. They try to avoid simple memorization questions about instance data. They also avoid questions that require overly complex reasoning or calculation that would be difficult both to complete during a time-based test and to grade.

We also simplified many aspects of a standard AP exam to facilitate administering the test to a computer program. Because AURA could not process diagrams, all knowledge found in diagrams, either in the textbook or in test questions, had to be explicitly encoded into the system. Because AURA could not handle full natural language, all test questions were reformulated by the SMEs into simpler statements using AURA's Controlled Processing Language (CPL). This usually required multiple attempts with some amount of question interpretation by the user. AURA could also not process multiple-choice questions as a single chunk and therefore required the user to break the question into separate sub-questions for each multiple-choice option.

Despite these caveats, our overall assessment is that AURA has achieved a well-engineered process for SMEs to encode basic conceptual knowledge, especially if the SMEs have sufficient experience with AURA and work as a member of a dedicated KF team. Based on our initial multi-user experiment, scaling up this process to a large KF team that can encode the conceptual knowledge for a complete, college-level textbook appears possible.

AURA has also achieved a question-formulation capability that enables users to easily and effectively ask questions of the system. CPL works well. Users find it easy to learn. Non-experts are generally as effective as experts at formulating and asking questions of the system. Yet, room for improvement exists here as well. Finding the right reformulation often requires several iterations, and finding the precise terms to match the correct KB concept is sometimes awkward. Nevertheless, the overall question-formulation process worked well enough.

At the same time, knowledge-representation and reasoning challenges require further research before we can break through the 80% barrier and can represent all knowledge in a full textbook. As mentioned earlier, we have performed analyses of the KR&R requirements of AP exams for our scientific domains and have identified several areas where we need improvement:

- Actions and processes: especially in biology, much of the knowledge involves complex processes. Currently AURA uses a STRIPs-style representation of the events and sub-events, which works well for many AP questions, but we expect will not be rich enough to master more advanced material.
- Computational knowledge: in many situations, such as balancing chemical reactions, the knowledge needed involves computational procedures that do not lend themselves to a declarative representation.
- Qualitative reasoning: all three domains require qualitative reasoning, which we have yet to add to the system.
- Naïve physics and core commonsense reasoning: we currently rely on the user to add commonsense context as he formulates questions, but question-answering performance could be greatly improved, especially in physics, where non-experts had the most difficulty in question formulation.
- Diagram understanding and spatial reasoning: much of the textbook knowledge and many of the test questions, in all three domains, use diagrams to portray implicit spatial knowledge. Knowledge formulation could be streamlined if the system could ingest and understand diagrams with implicit spatial knowledge.
- Abduction, abstraction, analogy, and uncertainty: these well-known KR&R challenges are present here as well. We avoid some of these complexities by focusing on well-established, clearly defined scientific knowledge, but even then, these challenges arise.
- Web-scale collaborative authoring: so far AURA has been developed as an authoring tool for individual authors or small authoring teams but not for web-scale collaborative authoring.

Future Plans

Vulcan, Inc. plans to continue pursuing the vision of the Digital Aristotle by: 1) scaling-up AURA's current capabilities to handle a full textbook in biology, while simultaneously 2) conducting advanced research on the remaining KR&R challenges.

Given the encouraging results for encoding basic conceptual knowledge into AURA, we plan to employ a multi-user collaborative KF team to encode all of the knowledge possible for an introductory biology textbook and to then see how well that KB performs on a full AP biology exam. To this end, we plan to improve AURA's software infrastructure to support a knowledge-formulation team and to redesign the question-formulation and question-answering capability. The result will be a knowledgebase and improved question-answering system for the complete biology textbook.

This will produce the first prototype of what we are calling a *HaloBook*—a new kind of electronic textbook that contains an underlying knowledgebase capable of answering the reader's questions and providing tailored instruction. We have explored the concept with researchers in education and interactive tutoring and feel this may produce a rich set of possibilities for creating a new educational technology.

In parallel, Project Halo will continue to develop semantic extensions to Semantic MediaWiki (SMW)+, which provides a community-based environment for authoring ontologies and creating semantically enhanced wikis (Pfisterer et al., 2009). SMW+ has been widely used and is being applied to project management, enterprise information, the management of large terminology sets, and the semantic enhancement of Wikipedia. Vulcan will continue to explore applications of SMW+, especially in the semantic enhancement of Wikipedia and the creation scientific datasets on the Web.

Also in parallel, Vulcan will continue to explore solutions to the hard KR&R challenges listed above. In 2007, Vulcan began a new effort, Halo Advanced Research (HalAR), to address the difficult knowledge-representation and reasoning (KR) challenges that prevent the realization of Digital Aristotle. This effort has produced a new semantic rule language and reasoning system, Semantic Inferencing on Large Knowledge (SILK), which includes major advances, including for default and higher-order reasoning (Grosof et al., 2009; Wan et al., 2009). In the next year, we will refine the SILK system, exploring richer models of process based on SILK, developing an authoring environment to enable SMEs to use its more powerful KR&R features, and eventually integrating the best features of AURA, SMW+, and SILK into the next generation Halo system.

In summary, Vulcan continues to make a steady progress toward its long-term goal of producing a Digital Aristotle. Central to achieving this goal is Vulcan's plan of development, which revolves around the encoding of well-defined bodies of knowledge such that the success of the encoding can be measured using an objective and easily understood test. Vulcan's development plan is driving the formulation and solution of fundamentally difficult problems in knowledge representation and reasoning; knowledge acquisition; question answering; and web-scale authorship and reasoning. As the technology develops and matures further, Vulcan will explore opportunities for using this technology to solve important problem for education, bio-discovery, and business enterprise.

References

Barker, K., Porter, B., & Clark, P. (2001). A Library of Generic Concepts for Composing Knowledgebases. In *Proc. 1st Int Conf on Knowledge Capture (K-Cap'01)* (pp. 14--21).

Barker, K.; Chaudhri, V. K.; Chaw, S. Y.; Clark, P. E.; Fan, J.; Israel, D.; Mishra, S.; Porter, B.; Romero, P.; Tecuci, D.; and Yeh, P. 2004. *A Question-Answering System for AP Chemistry*: Assessing KR&R Technologies. In Proceedings of the Ninth International Conference of Knowledge Representation and Reasoning. Menlo Park, Calif.: AAAI Press.

Brown, T.L. et al., *Chemistry: The Central Science*. 2003, New Jersey: Prentice Hall.

Campbell, N. A., & Reece, J. (2001). *Biology, Sixth Edition*: Benjamin Cummings.

Chaudhri, V. K., Greaves, M., Hansch, D., Jameson, A., & Pfisterer, F. (2008). *Using a Semantic Wiki as a Knowledge Source for Question Answering.* In the Proceedings of the AAAI Spring Symposium on Symbiosis of Semantic Web and Knowledge Engineering, Stanford, CA.

Chaudhri, V. K., John, B., Mishra, S., Pacheco, J., Porter, B., & Spaulding, A. (2007). *Enabling Experts to Build Knowledgebases from Science Textbooks*. In the Proceedings of the International Conference on Knowledge Capture Systems (KCAP).

Chaudhri, V.K., Bredeweg, B., Fikes, R., McIlraith, S., Wellman, M. (2010). *A Categorization of KR&R Methods for Requirement Analysis of a Query Answering Knowledgebase.* International Conference on Formal Ontologies in Information Systsems, Toronto, ON, Canada.

Clark, P. et al. *Knowledge Entry as Graphical Assembly of Components*. In International Conference on Knowledge Capture. 2001. Victoria, BC.

Clark P., et. al. (2003) *Enabling domain experts to convey questions to a machine: A modified, template-based approach.* 2nd International Conference on Knowledge Capture Systems.

Clark, P., Chaw, J., Barker, K., Chaudhri, V., Harrison, P., John, B. et al. (2007). *Capturing and Answering Questions Posed to A Knowledge-Based System.* In the Proceedings of the International Conference on Knowledge Capture Systems (KCAP), Whistler, Canada.

Clark, P., & Porter, B. (1999). KM -- The Knowledge Machine: Users Manual.

Feigenbaum, E. (2003). *Some Challenges and Grand Challenges for Computational Intelligence.* Journal of the Association of Computational Machinery 50(1): 32-40.

Friedland, N., Allen, P., Whitbrock, M., Matthews, G., Salay, N., Miragalia, P. et al. (2004, June 2-5). *Toward a Quantitative Platform-independent Analysis of Knowledge Systems.* Paper presented at the KRR-2004, Whistler, Canada.

Friedland, N., Allen P. et al. (2004), *Project Halo: Toward a Digital Aristotle*. AI Magazine 25 (4).

Giancoli, D.C., *Physics Principles with Applications*. 2004: Benjamin Cummings.

Grosof, B., Dean, M., and Kifer, M. (2009), *The SILK System: Scalable Higher-Order Defeasible Rules*, International RuleML Symposium on Rule Interchange and Applications (RuleML-2009), Las Vegas, Nevada.

Pfisterer F., Jameson, A., Barbu, C. (2008) *User-Centered Design and Evaluation of Interface Enhancements to the Semantic Media Wiki.* In the Proceedings of the Workshop on Semantic Web User Interaction at CHI 2008, Florence, Italy.

Reddy, R. (2003). *Three Open Problems in AI.* Journal of the Association of Computational Machinery 50(1).

S. Chaw, K. Barker, B. Porter, D. Tecuci, and P. Yeh (2009), *A Scalable Problem-Solver for Large Knowledge-Bases,* In Proceedings of the 21st International Conference on Tools with Artificial Intelligence (ICTAI 2009)

Wan, H., Grosof, B., Kifer, M., Fodor, P., and Liang, S. (2009), *Logic Programming with Defaults and Argumentation Theories*, 25th International Conference on Logic Programming (ICLP 2009), Pasadena, California

Yeh, P., Porter, B., & Barker, K. (2006). *A Unified Knowledge-Based Approach for Sense Disambiguation and Semantic Role Labeling.* Paper presented at the AAAI-2006, Boston.