Project Halo Update — Progress Toward Digital Aristotle

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

■ *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 (KRR) 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 article 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 knowledge base and that then allows a different set of users to ask novel questions against that knowledge base. 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.*

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 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 (that is, to have genuine, internal "understanding" of the domain).

 ISSN 0738-4602

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.[1]

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 knowledge base capable of answering novel questions from an AP (first-year, college-level) chemistry test. Three teams — SRI International, Cycorp, and Ontoprise — developed knowledge bases for a limited section of an AP chemistry syllabus. The knowledge bases could correctly answer between 30 and 50 percent of the associated questions from the AP test (Friedland et al. 2004a, 2004b).

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 knowledge bases, making large-scale 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) (such as 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 Professor 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 knowledge bases 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 article summarizes that system and its evaluation.

AURA System Development

The concept of operation for AURA is as follows: a knowledge-formulation (KF) SME, 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) SME, with at least a high-school-level education, undergoes 4 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 training in using the system. The questioner 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 and Reece 2001). A subset of each AP syllabus was selected that covered roughly 60 pages of text and 15–20 percent of the AP topics for each domain. The AURA team was challenged to design and develop a system that could fulfill the 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 percent of the AP questions (in order of importance): conceptual knowledge, equations, diagrams, and tables. (1) Conceptual knowledge represents classes, subclasses, slots, slot constraints, and general rules about class instances. (2) A majority of questions in physics and some questions in chemistry involve mathematical equations. (3) All three domains make extensive use of diagrams. (4) Tables are 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 KRR 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 KRR: (1) knowing where to begin is often challenging for domain experts (the blank slate problem); (2) knowledge formulation consists of a complete life cycle 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 questioner will have an intimate familiarity with the knowledge base or the knowledge-formulation tools. Because the questioner 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 percent of questions used direct anaphora, 50 percent used indirect anaphora, and 60 percent 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 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 et al. 2008).

AURA System Architecture

The AURA system has three broad classes of functionality: knowledge formulation; question formulation; and question answering. 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. Figure 2 illustrates a domain expert working with AURA.

Knowledge Representation and Reasoning

AURA uses the Knowledge Machine (KM) as its core knowledge representation and reasoning engine, a powerful, mature, frame-based knowledge representation system.[2] 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 represents the properties of all members of a concept using a notional example of that concept. The syntax of a prototype is a graph data structure, depicting the properties of that notional example as a set of interconnected nodes and relations (see later figures for examples). The use of a graph-based representation is highly significant as it means that the internal form and its presentation to the user are the same, allowing the user to view and modify the repesentation directly through graph manipulation, rather than editing logical axioms that would encode the same knowledge. The semantics of a prototype have a formal axiomatic specification, asserting that all individuals of that concept have the properties of the notional example..

Syntactically during reasoning, to infer properties of an individual, KM merges, or "unifies," all the prototype graphs of the concepts that the individual belongs to with that individual, thus constructing a graph-based representation of an individual with all the properties of its concepts' prototypes. Semantically, this operation of unifying two individuals, called *UMAP,* is simply to equate them plus recursively conditionally unifying the value(s) of their properties. Two property values are unified if either deductively they must be the same (for example, due to cardinality constraints), or heuristically they appear to be the same (for example, are of the same type). The lat-

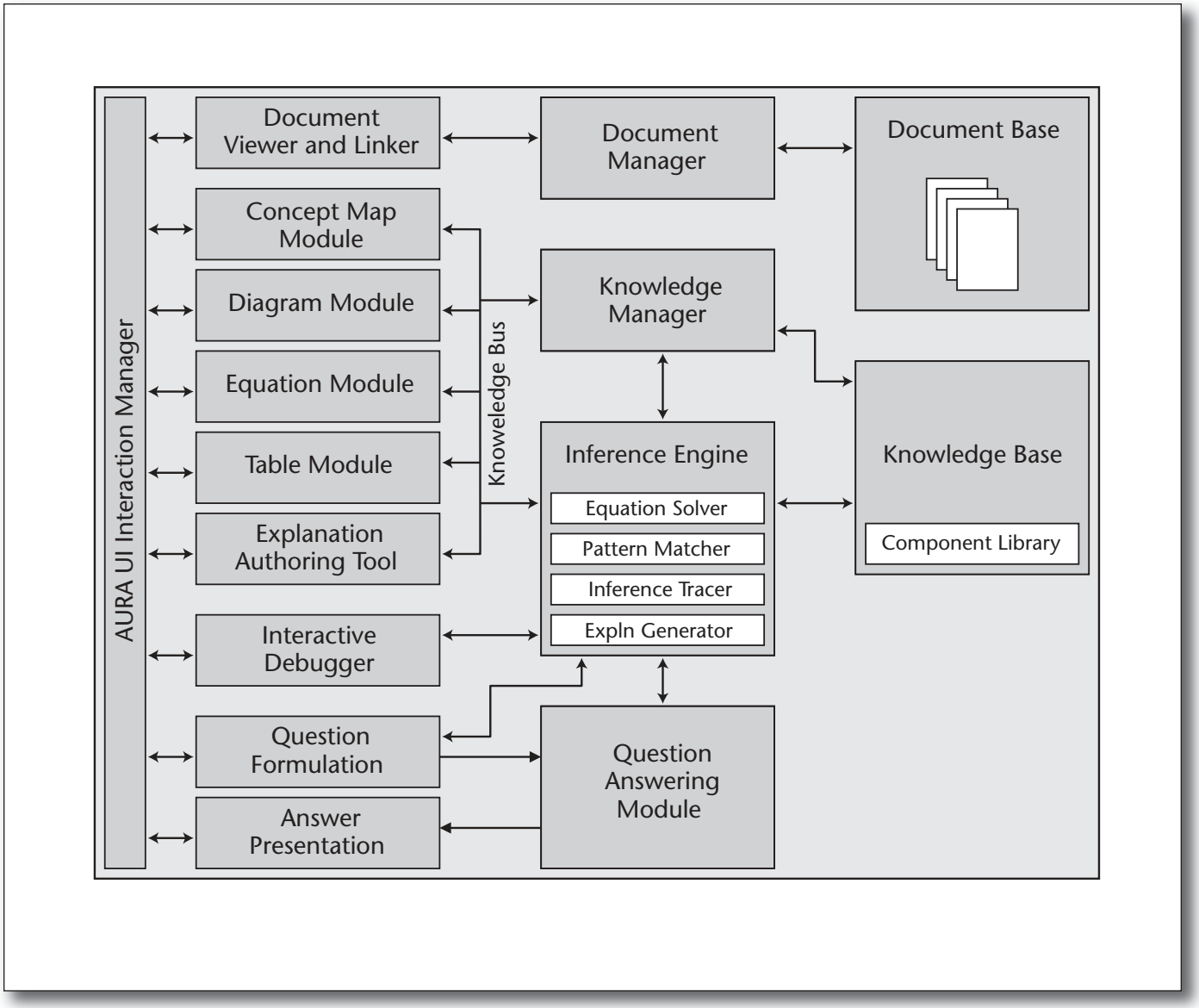

Figure 1. AURA System Architecture.

ter use of equality heuristics distinguishes UMap from equality, and allows KM to draw plausible inferences in an underspecified knowledge base, filling in details that an SME might leave out. Although in principle UMAP can make mistakes (as it is unsound), in practice this is rare and significantly outweighed by its advantages in replicating the kind of equalities that a person would naturally assume. 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 the U.S. Defense Advanced Research Project Agency'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 includes three salient features: (1) the use of a document as a starting point and context for all knowledge entry; (2) a prebuilt 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

Figure 2. A Domain Expert Working with AURA.

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 (Felbaum 1998) mappings from words in the document to concepts in the knowledge base serves as the basis of making suggestions for concepts relevant for encoding that word.

The SMEs build their knowledge bases by reusing representations in a domain-independent knowledge base called the *Component Library* or CLIB (Barker, Porter, and 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 SMEs in the KF process. A selection of top-level classes in CLIB is shown in figure 3.

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 expected that this capability would enable users to encode knowledge sufficient to answer approximately 50 percent 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 gray nodes represent nonroot 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. Examples of specific logical forms generated are included in a later section of the article.

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.

As an example of a process in biology, in figure 5, we show a (simplified) concept map for mitosis. This concept map shows the different steps in mitosis (prophase, metaphase, and so on), their relative ordering, and that its object is a diploid cell

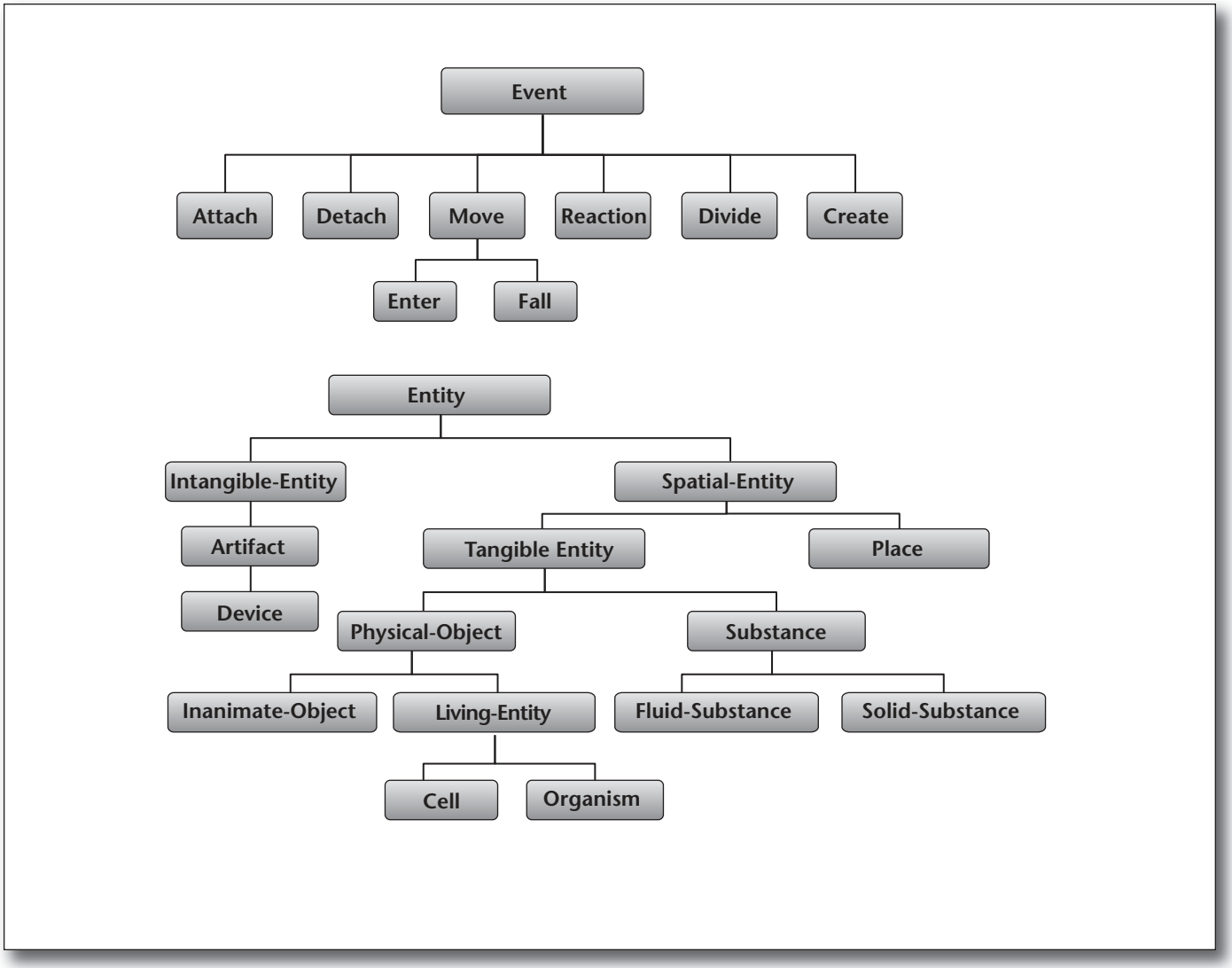

Figure 3. The Top-Level Event and Entity Classes in CLIB.

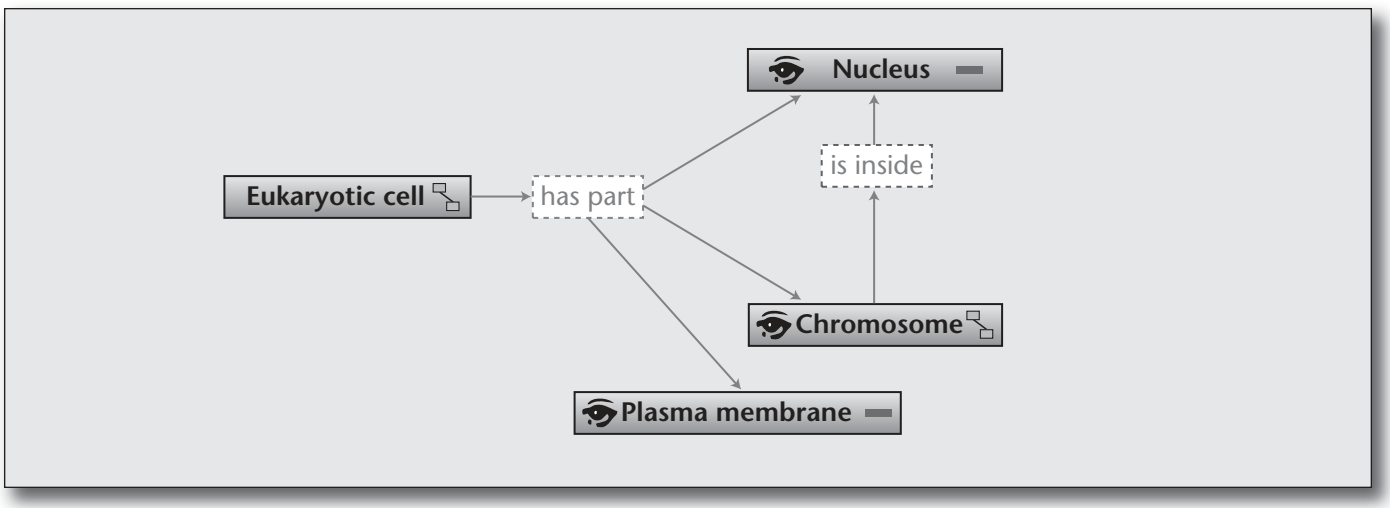

Figure 4. A Biology Concept for the Eukaryotic Cell.

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.

The SMEs 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 predefined 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 H_2O (or water) from chemistry. The top part of this graph encodes that every individual instance of H_2O has-part an OH^- ion and H^+ ion, and further an H^+ ion has-atom H. The lower part of the graph shows another H_2O individual that is added to this graph. If the user equates the two H_2O 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 SME would need to encode cardinality constraints on has-part and has-atom relations. UMAP can draw equality inferences even when the knowledge base 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 underspecification in the knowledge base 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 7, 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 "what you see is what you get" editor for entering equations, and the equations can be related to properties that are represented in the knowledge base.

We have designed a training course for SMEs that prepares them to enter knowledge into AURA. The current KF training is approximately 20 hours. The training introduces the SMEs to the mechanics of using the system and to basic knowledge engineering principles. In the knowledge engineering section of the training, the SMEs learn about different classes and relations in CLIB, and how to use them. The training program includes several hands-on exercises in which SMEs encode knowledge and are given feedback on their specific

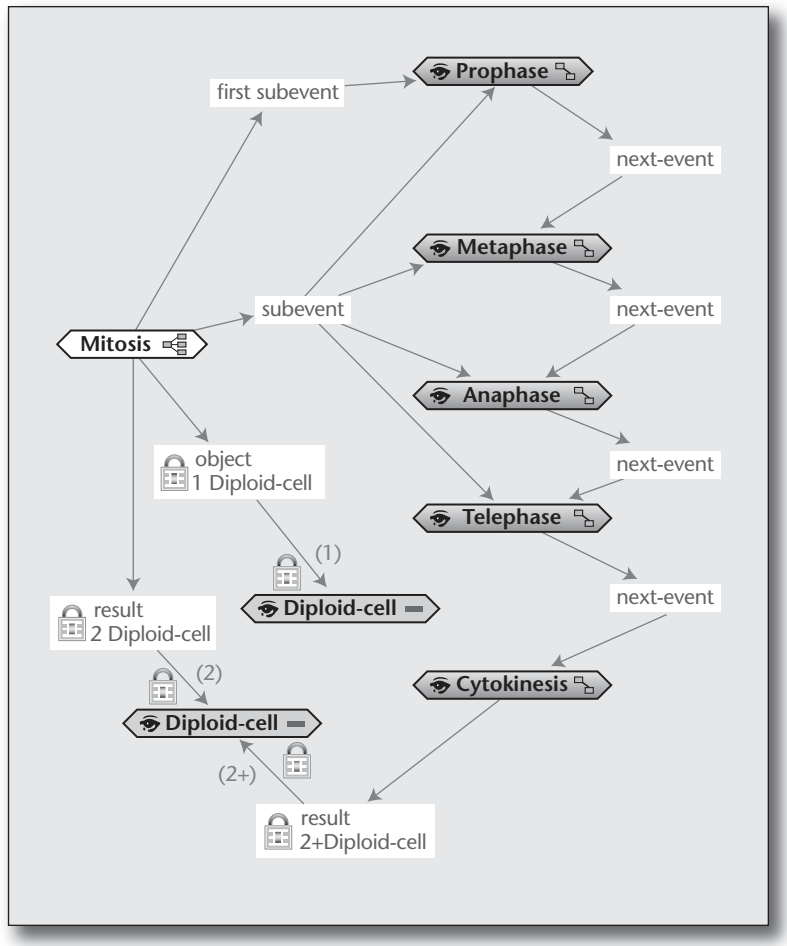

Figure 5. A Biology Concept Representing Mitosis.

choices. The core of the training program is common across all three domains. There are, however, several domain-specific modules. For example, physics SMEs must learn to properly use vector math, which does not arise in the other two domains. For chemistry, the SMEs must learn about entering chemical compounds and reactions, and about chemistry-specific, system-available knowledge. For biology SMEs, 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 be kept low. Further, we cannot assume that the questioner will have an intimate familiarity with the knowledge base 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 (for example, Voorhees and Buckland 2008), our context is somewhat different, namely posing

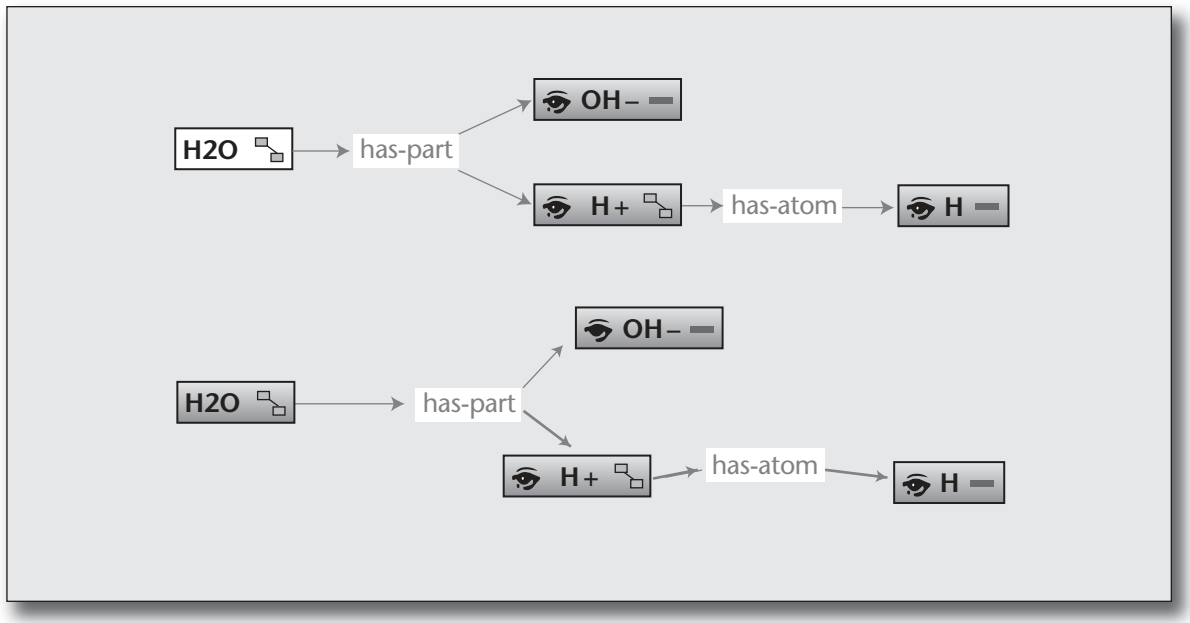

Figure 6. The Use of UMAP on Two Entities Recursively Equates All Its Parts.

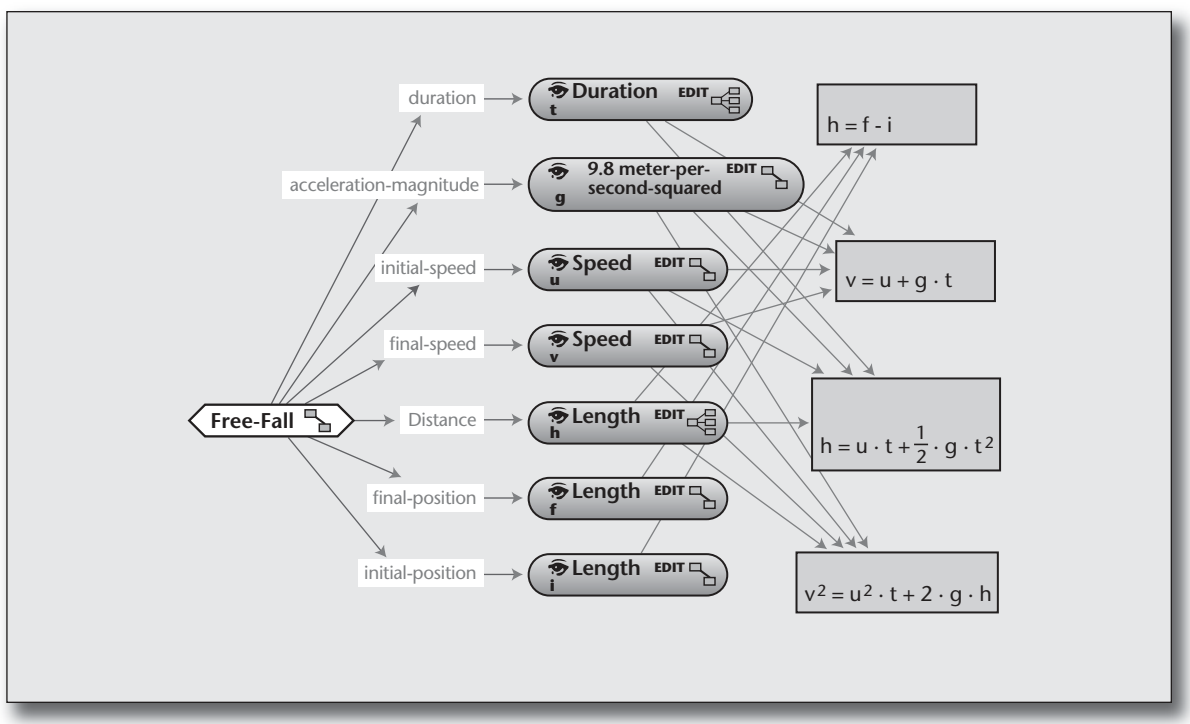

Figure 7. A Physics Concept of Free Fall.

questions to a formal knowledge base, 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 computer-processable language (a simplified version of English) called CPL for posing questions, with feedback mechanisms to help in the

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.

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 run-time 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 (for example, keep sentences short, use just one clause per sentence), while others are firm constraints on vocabulary and grammar (for example, words of uncertainty [for example, "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 reexpressed 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

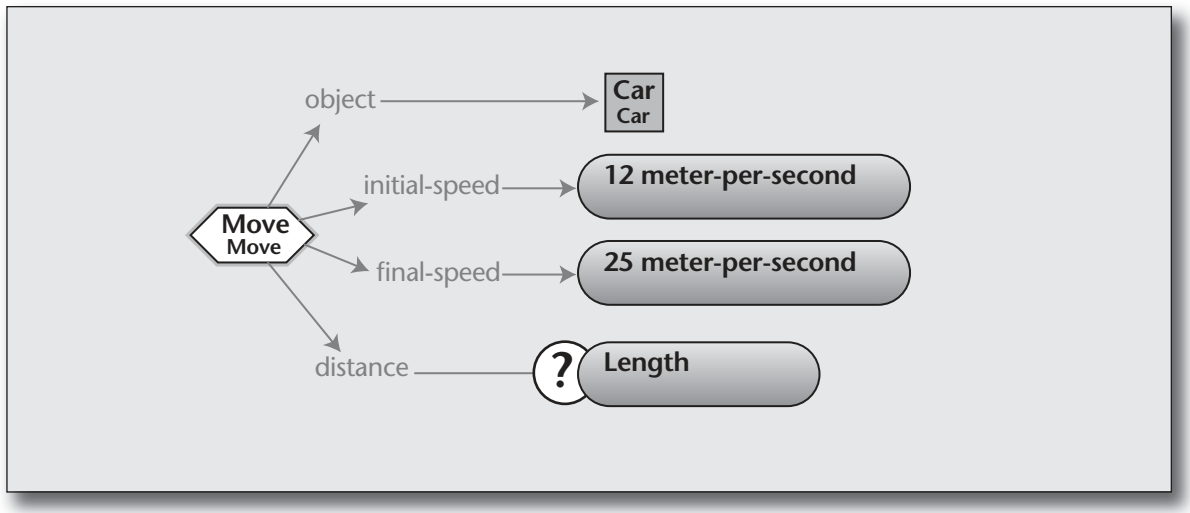

Figure 9. Graphical Feedback during Question Formulation.

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.

Note that using a controlled language involves a trade-off between machine understandability and fidelity, that is, the process of making the question machine understandable may involve simplifying or expanding the original question's semantics. For many questions (for example, "Does a eukaryotic cell have a nucleus?") there is no loss of fidelity, but for more complex questions a more significant rewording may be needed. Example 2 in figure 8 illustrates this, where "What two molecules must always be present in the products...?" is reexpressed as "What are the products...?" In such cases there is some cognitive burden on the user to use his or her linguistic and general knowledge to simplify "wordy" English and know what simplifications are reasonable, combined with the user's knowledge of the kind of statements AURA understands, acquired from training and experience using the system. A controlled language represents a pragmatic middle ground, trying to balance the machine-understandability/fidelity trade-off. We evaluate the effectiveness of this later in this article.

Let us now consider how this design meets the requirements of the questioner. The CPL formulations expected of questioners are in terms of English words and, thus, do not require intimate knowledge of the knowledge base's vocabulary. To read the interpretation graph, the questioners must understand the meaning of the concepts and relations. Through AURA, the questioners 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 knowledge base by inspection is significantly easier than using those terms for creating new concepts as the SMEs are required to do. CPL also allows questioners 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 knowledge base, identifies the question type, and invokes the necessary reasoning. In some cases, relating an individual to a class in a knowledge base is straightforward, especially as AURA allows SMEs 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 as its core reasoning engine. AURA has a special-purpose reasoning module for solving algebraic equations that

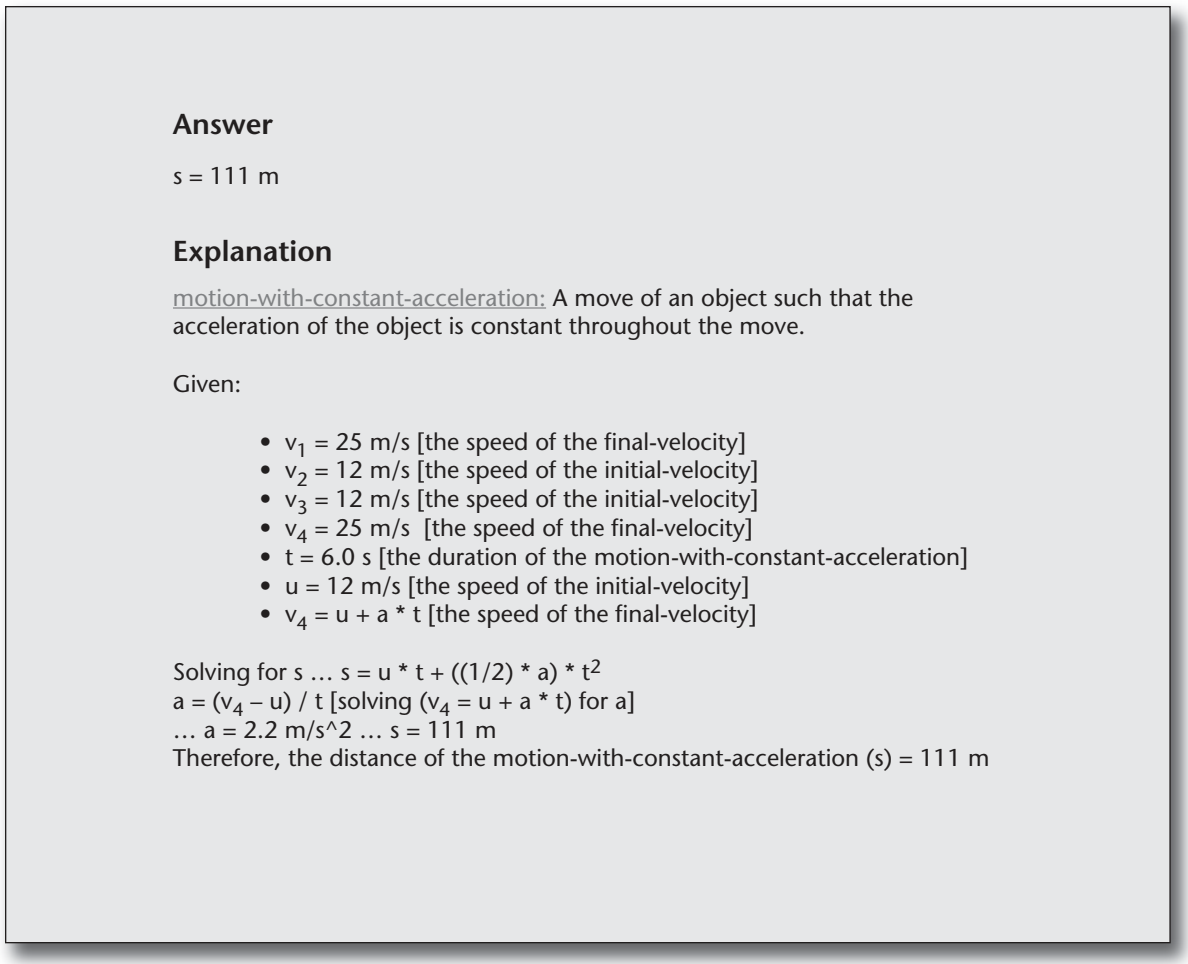

Figure 10. Answer to an Example Physics Question.

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 (figure 12), 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.

AURA in Action

In this section, we give a tour of AURA by starting from a sample paragraph of text and showing how it is encoded by a KF SME, followed by how questions are posed, and finally showing a sample answer produced by the system.

Knowledge Formulation

Consider the following paragraph from the biology textbook:

> All cells have several basic features in common: They are all bounded by a membrane, called a plasma membrane. Within the membrane is a semifluid substance, cytosol, in which organelles are

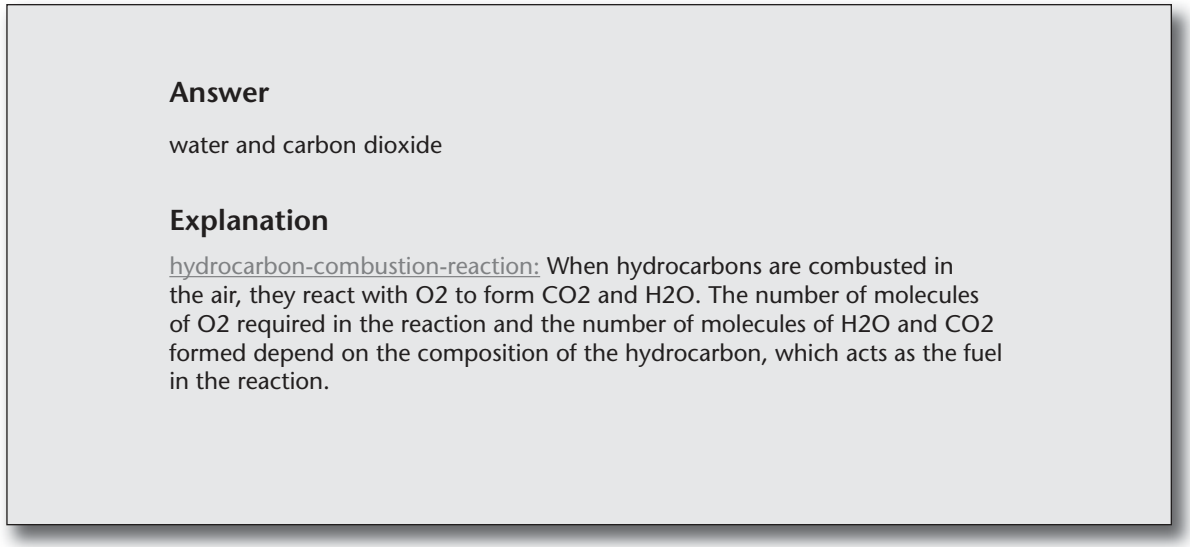

Figure 11. Answer to an Example Chemistry Question.

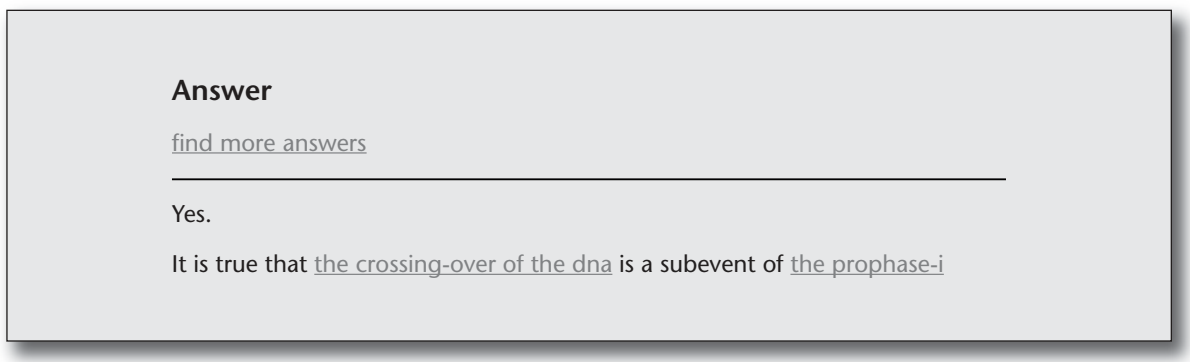

Figure 12. Example Answer to Biology Questions.

found. All cells contain chromosomes, carrying genes in the form of DNA. And all cells have ribosomes, tiny organelles that make proteins according to instructions from the genes.

A KF SME begins by highlighting the paragraph of text and then underlining the words that are central to capturing the knowledge in the paragraph. Based on this input, AURA performs a semantic search against its knowledge base and suggests a few starting points to begin the knowledge encoding progress. The semantic search is one of the solutions in AURA to address the blank slate problem, that is, giving a KF SME a place to begin instead of expecting a start from scratch.

As an example, in figure 13, we show the results returned by semantic search for the underlined words in the previous paragraph. The word *cell* matches against the concept Cell already existing in the knowledge base. The concept Cell is part of the pump-primed knowledge in the knowledge base. The pump-primed knowledge in a knowledge base is the domain-specific knowledge that is prebuilt into the knowledge base before the KF SME begins the knowledge entry process. We incorporate some basic biology knowledge in the system to bootstrap the knowledge entry for the KFEs. For the word *contain*, a direct match is the concept Container, and several semantic matches such as Add, Event, Restrain, and so on (shown in the pull-down menu). As part of the search result, one of the options is to create a new concept if none of the matches is satisfactory. The semantic matches are retrieved based on the links between the concepts in the knowledge base and the words in Wordnet.

Even though the concept Cell exists in the knowledge base, there are no properties or rules defined for it. To define the properties of a cell as introduced in the paragraph, the KF SME will also need to create a concept to represent "plasma membrane," "cytosol," "organelle," "chromosome," and so on. For the purpose of this tour, we assume that a KF SME will take a diversion from the knowledge entry of this paragraph, and create all the concepts necessary to represent this paragraph, and then resume the knowledge entry for Cell that we describe next.

The KF SME enters the knowledge in the Cell, by connecting the concepts that have already been created and using a set of predefined relations. As a specific example, in figure 14, we show that the KF SME is connecting the concept of Cell to the concept of Ribosome.

While making this connection, the system presents to the KF SME a collection of legal relations that are selected based on the domain and range constraints of those relations. The set of relations is designed by knowledge engineers and a KF SME is not allowed to change them. If the need for a new relation arises, a KF SME can make a request to the KEs, who can carefully design and choose a new relation to be added to the component library.

With the successive use of basic graph editing operations, a KF SME can create a complete concept map for the knowledge about a Cell in the above paragraph as shown in figure 15.

AURA translates the graph shown in figure 15 into the internal logical representation of KM called *prototypes* that is logically equivalent to the rule shown in figure 16. This rule should be read as an if-then rule, that is, the formula immediately following the implication symbol => is the antecedent followed by the consequent:

The graph shown in figure 15 captures only those properties of a Cell that are necessary properties and this is reflected in the corresponding logical axioms as shown in figure 16. The graphical interface in AURA also enables the capture of sufficient properties. To illustrate the capture of sufficient properties, we consider the following sentence that appears in the next paragraph:

> In contrast, the eukaryotic cell (Greek eu, true, and karyon) has a true nucleus, bounded by a membranous nuclear envelope.

To encode this sentence, the KFE will follow a process similar to what has been already illustrated and create a graph shown in figure 17.

Since a Eukaryotic-Cell is a subclass of Cell, it inherits Ribosome and Chromosome as its parts. (Other inherited information has been hidden for brevity.) The KFE will add a Nucleus as an additional part (over and above what was inherited), and indicate that if a Cell has a Nucleus as its part, it is sufficient for it to be a Eukaryotic Cell. This is highlighted in red, and it also appears at the top of the graph. The highlighted part of the graph is translated into the rule depicted in figure 18.

In addition to the examples considered so far, AURA is capable of capturing knowledge about mathematical equations and tables. These knowledge capture interfaces have been discussed in previously published papers (Clark et. al. 2001, Chaudhri et. al. 2004).

Question Formulation

Let us now consider how the user poses questions for the knowledge that has already been formulated. We will illustrate this by taking a specific AP-style question.

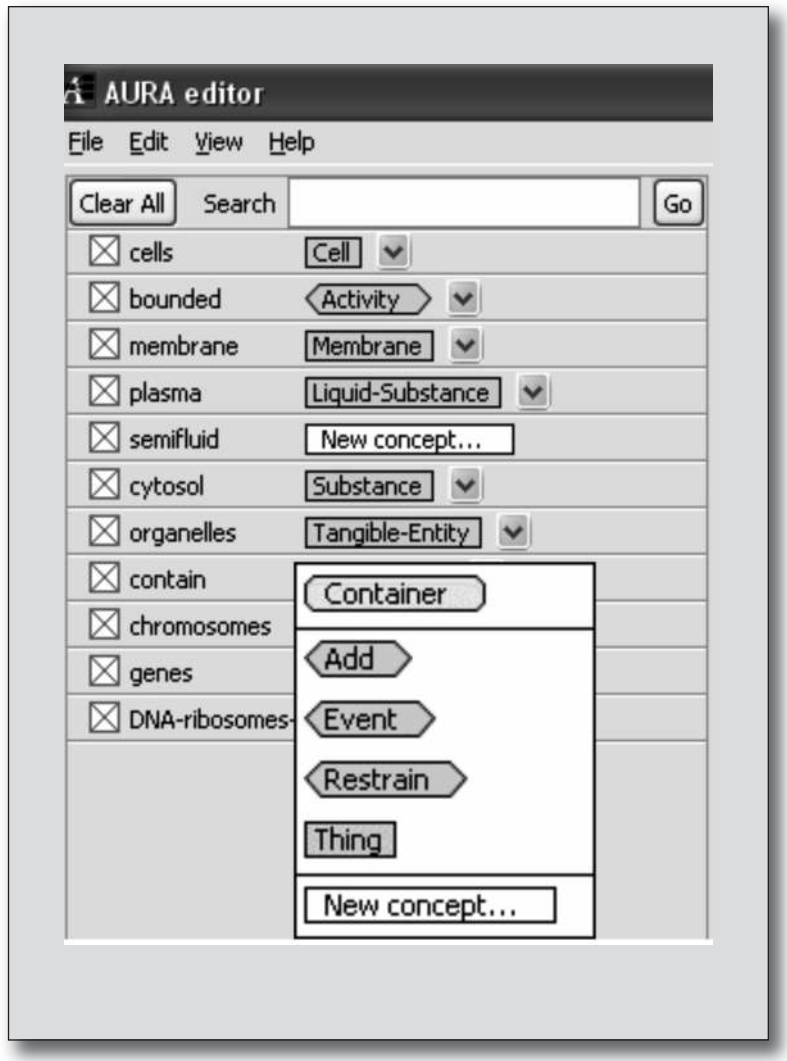

Figure 13. Semantic Search Facility Suggests Starting Points for Knowledge Formulation.

> Studying a picture of a cell with an electron microscope, you find that the cell has a cell wall, a nucleus, and a large central vacuole. You conclude that the cell is probably a(n)
>
> (a) Eukaryotic cell
> (b) Plant cell
> (c) Prokaryotic cell
> (d) Bacterial cell

For answering this question, the important information is that the cell has a cell wall, a nucleus, and a large central vacuole. The fact that these parts were observed through an electron microscope is irrelevant. During the four hours of training, we teach the question formulation engineers to factor out such extraneous detail from their queries. We also do not explicitly address the multiple-choice aspect of a question and instead

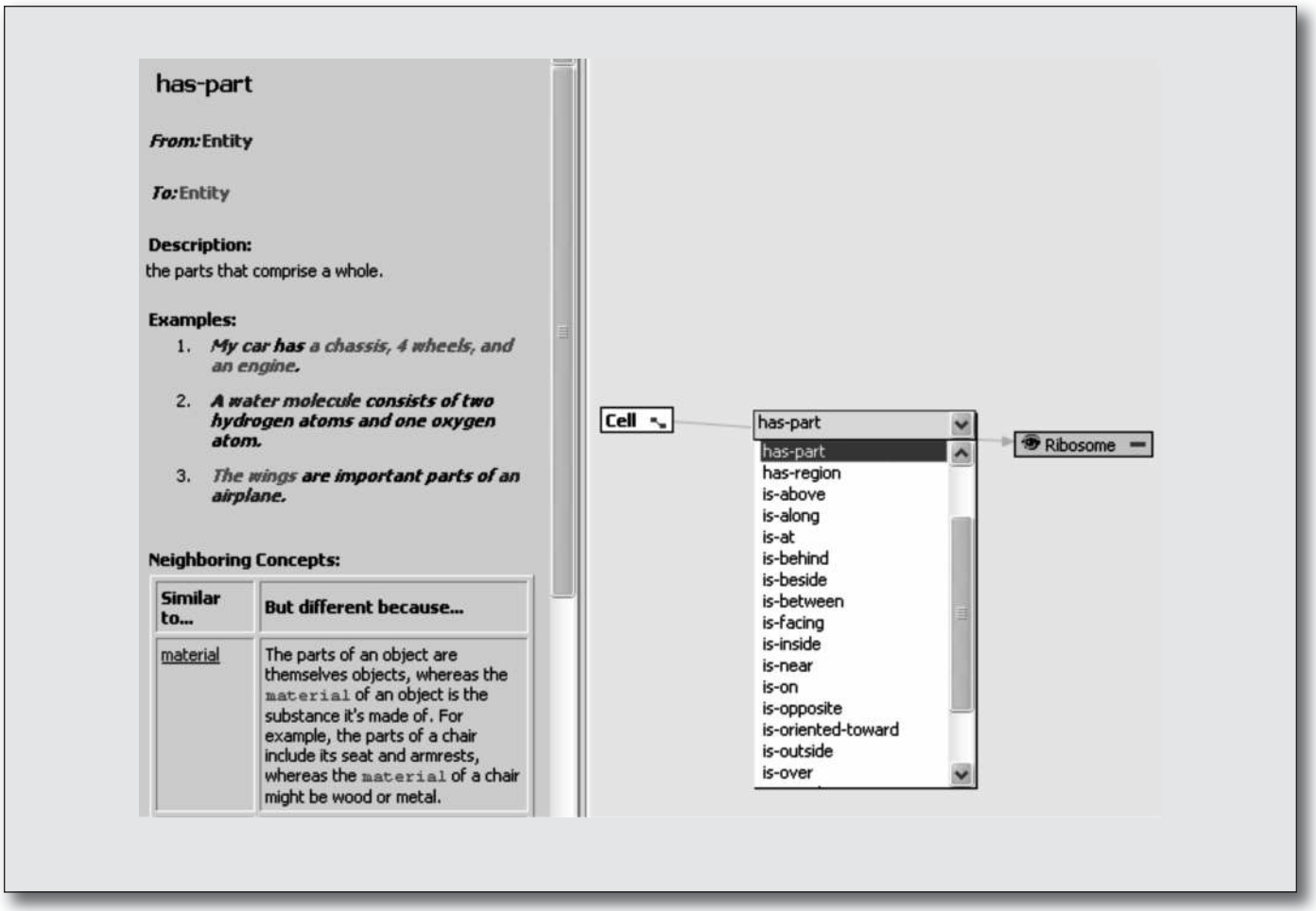

Figure 14. KFEs Connect Nodes on a Graph Using a Set of Predefined Relations.

expect the user to formulate this question as four separate questions. Using the controlled English facility of the system, this question will be formulated as four separate questions as follows:

A cell has a cell wall, a nucleus, and a large central vacuole.

Is it true that the cell is a Eukaryotic Cell?
Is it true that the cell is a Plant Cell?
Is it true that the cell is a Prokaryotic Cell?
Is it true that the cell is a Bacterial Cell?

The first sentence stating the facts about the cell is common to all four formulations. Using these formulations, the question-understanding module produces a logical interpretation of the query. For example, the logical presentation for the first part is as follows:

Query premise:

(_Cell9426 instance-of Cell)
(_Cell-Wall9431 instance-of Cell-Wall)
(_Nucleus9429 instance-of Nucleus)
(_Vacuole9430 instance-of Vacuole)
(_Cell9426 has-part _Cell-Wall9431)
(_Cell9426 has-part _Nucleus9429)
(_Cell9426 has-part _Vacuole9430)

Query pattern:

(_Cell9427 instance-of Eukaryotic-Cell)

The query premise asserts a hypothetical individual that is introduced in the query and the facts known about it. The query pattern states a logical formula that is to be proven.

The English statement of the query was phrased as "a Cell has a Cell wall, Nucleus…," but it is mapped to the has-part relation in the logical representation of the query. This computation is done by the semantic role-labeling module of the system that has access to a large database of paraphrases that it uses to map English phrases to the logical constructs known to the knowledge base. The English statement of the query also contained the word *central* but it appears nowhere in the logical formulation of the query. The query-understanding module has rules about producing an underspecified version of the query that drops certain modifiers for which the knowledge base has no knowledge.

A query formulated in logic in the vocabulary of the knowledge base is sent to the inference engine for query answering.

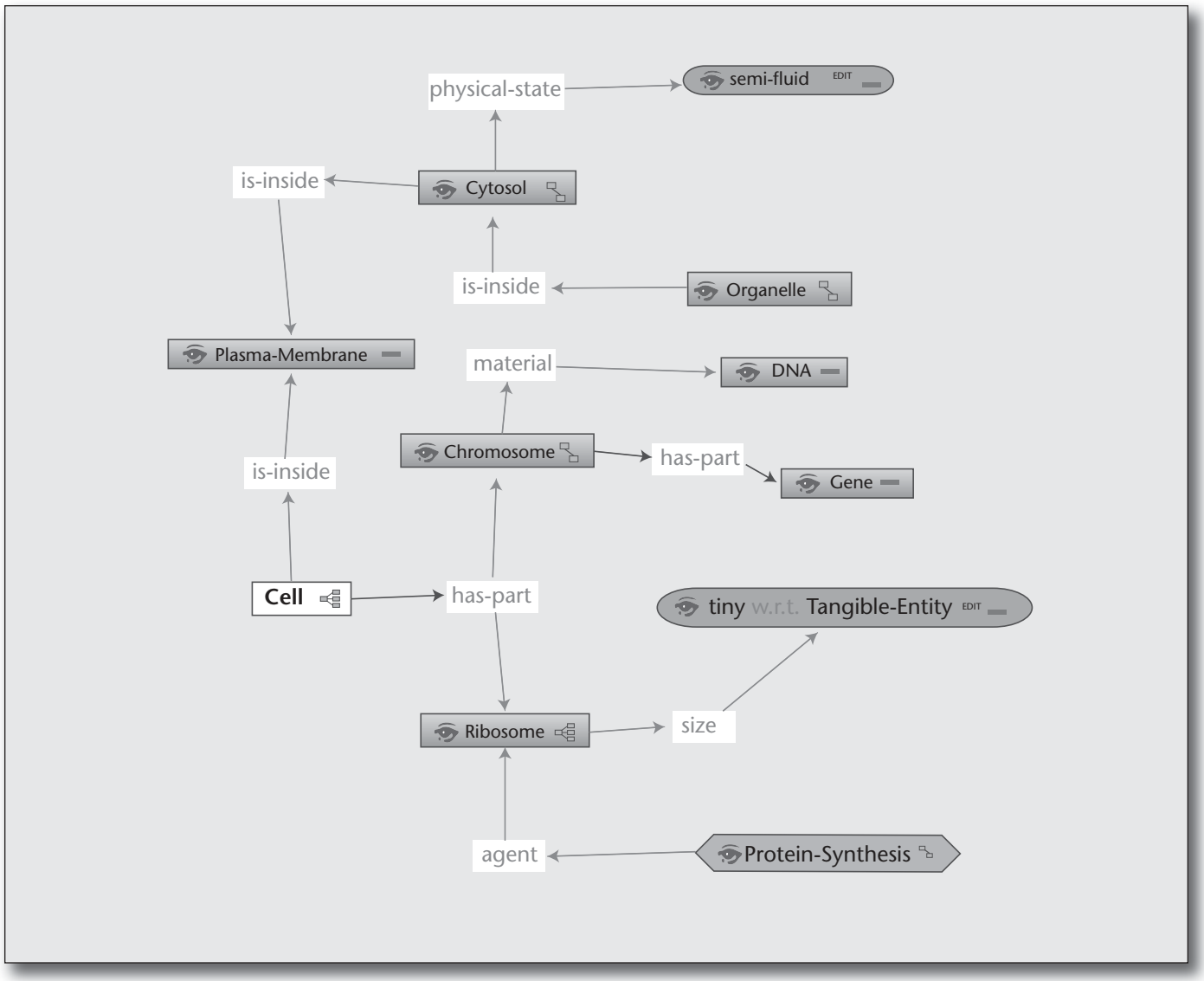

Figure 15. A Representation of Cell Based on a Paragraph of Text.

Question Answering

The reasoning control takes the logical specification of the query from the question-understanding module and identifies the correct question-answering method to apply, which for this question is a question of the form "Is it true that…." AURA will attempt to answer the query by first classifying the individual _Cell9427 in the knowledge base using a description logic inference (Baader et al. 2003). As we saw in the knowledge formulation section, a KFE had authored a rule asserting that if a Cell has a Nucleus, it must be classified as a Eukaryotic Cell, and thus, this question will be answered as Yes.

```
(forall ?c
  (=> (instance-of ?c Cell)
    (exists ?r ?ch ?g ?d ?p ?pm ?cy ?o
    (and
      (instance-of ?r Ribosome)
      (instance-of ?ch Chromosome)
      (instance-of ?g Gene)
      (instance-of ?d DNA)
      (instance-of ?p Protein-Synthesis)
      (instance-of ?pm Plasma-Membrane)
      (instance-of ?cy Cytosol)
      (instance-of ?o Organelle)
      (has-part ?c ?ch) (has-part ?ch ?g)
      (has-part ?c ?r) (material?ch ?d)
      (size ?r (scalar-value tinyTangible-Entity))
      (agent ?p ?r)
      (is-inside ?c ?pm) (is-inside ?cy ?pm)
      (is-inside ?o ?cy) (physical-state ?cy semi-fluid))
```

Figure 16. If-Then Rule.

AURA Evaluation

We conducted a full user evaluation to find out how well AURA enables graduate students in the

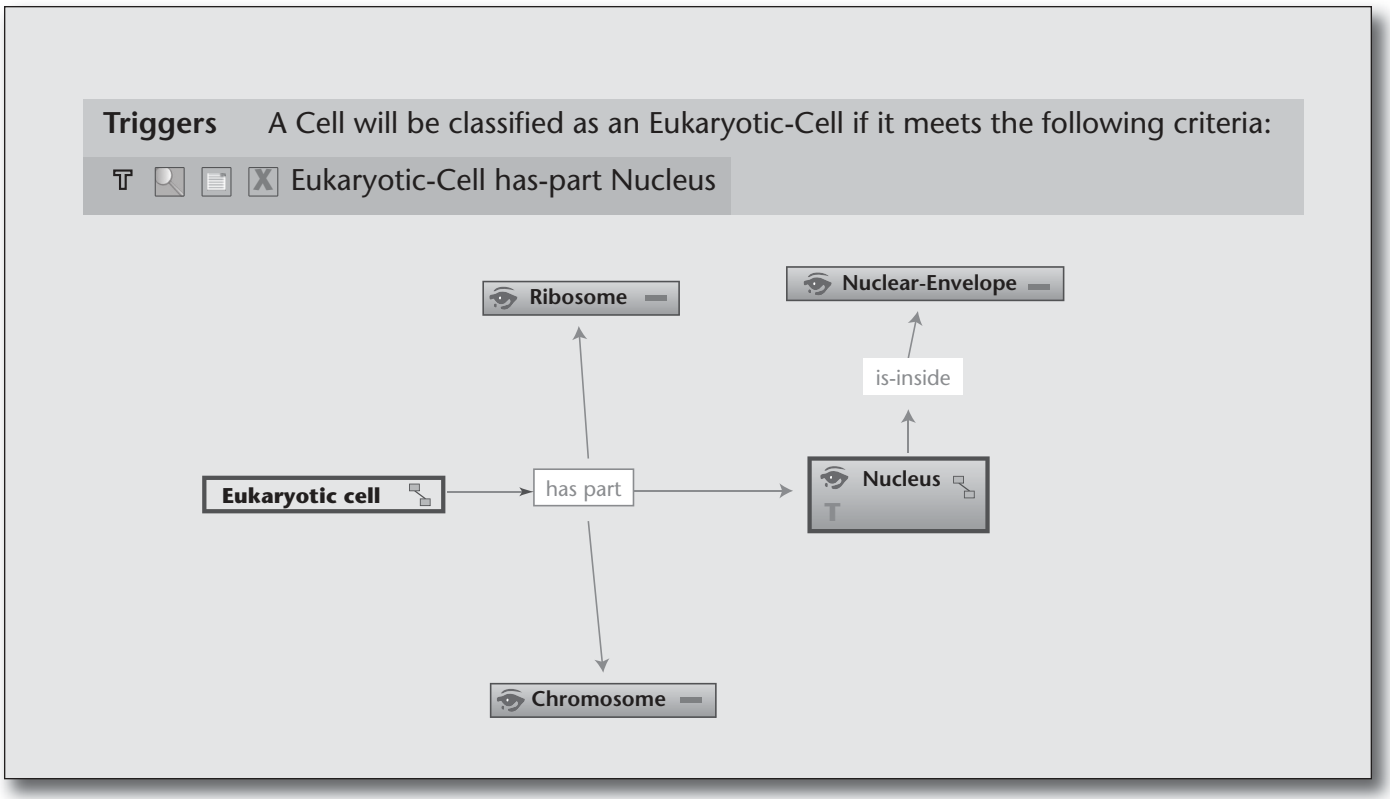

Figure 17. Capturing Sufficient Properties.

```
(<=>
  (and
    (instance-of ?c Cell)
    (exists ?n
      (and (instance-of ?n Nucleus)
           (has-part ?n ?c))))
  (instance-of ?c Eukaryotic-Cell))
```

Figure 18. Rule for Figure 17.

three domains (physics, chemistry, and biology) to construct knowledge bases 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 three main questions: (1) How well does AURA support knowledge formulation by domain experts? (2) How well does AURA support question formulation by domain experts? (3) How good are AURA's question-answering and explanation generation?

Experimental Design

To address the experimental questions, three sets of experimental conditions were evaluated: expert versus nonexpert KF experience, expert versus nonexpert QF experience, and question familiarity and difficulty.

For expert versus nonexpert KF experience, the *expert condition* was represented by domain experts with significant training and previous experience using AURA, working in close collaboration with the members of the AURA team, over many months. The *nonexpert 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.

For expert versus nonexpert QF experience, the *expert condition* was represented by the same SMEs as in the expert KF condition, and the nonexpert condition was represented by individuals qualified in their domain at an undergraduate level, with limited training (4 hours) and no previous experience using AURA.

For 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.

	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

Table 1. AURA Syllabus.

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 knowledge bases. Finally, 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

There were 10 main steps in the test procedure. First, in step 1, the AURA team selected the textbook sections and AP syllabus for each domain. Second, in step 2, expert SMEs of the AURA team authored knowledge bases for the selected textbook sections and AP syllabus, testing the knowledge against the reference questions. These SMEs worked closely with the development team. In step 3, experienced AP teachers recruited by BBN generated the set of novel questions in each domain to cover the topics in the selected syllabus. In step 4, expert SMEs at SRI formulated and asked the set of novel questions of their expert knowledge bases. For step 5, BBN and SRI chose 50 selected novel questions in each domain that best matched AURA's implemented reasoning capabilities. In step 6, SRI trained the nonexpert SMEs recruited by University of Colorado Denver in the use of AURA for knowledge formulation in a 20-hour training course. In step 7 the nonexpert SMEs at University of Colorado Denver authored knowledge over a four-week period (using approximately 120 hours of KF time). In step 8, SRI trained the nonexpert questioners in the use of AURA for question formulation in a 4-hour training course. In step 9, for each expert-formulated and nonexpert-formulated knowledge base, one or more questioners from the same domain asked selected novel questions. Finally, in step 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

Three textbooks were used. For biology, we used the sixth edition of *Biology* (Campbell and Reece 2001). For chemistry, we used the ninth edition of *Chemistry: The Central Science* (Brown et al. 2002). For physics, we used the sixth edition of *Physics: Principles with Applications* (Giancoli 2004). 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 counts are shown in table 1.

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 percent of the full syllabus, in chemistry, 67 pages covered 11 percent of the full syllabus, and in physics, 78 pages covered 15 percent 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 nonexpert SMEs consisted of nine students, three in each domain, recruited from the Denver area, through the University of Colorado at Denver, where the nonexpert 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 nonexpert questioners consisted of 19 (6 in biology and 5 each in chemistry and physics) undergraduates or very recent graduates, who were recruited in the Boston area, through BBN, where

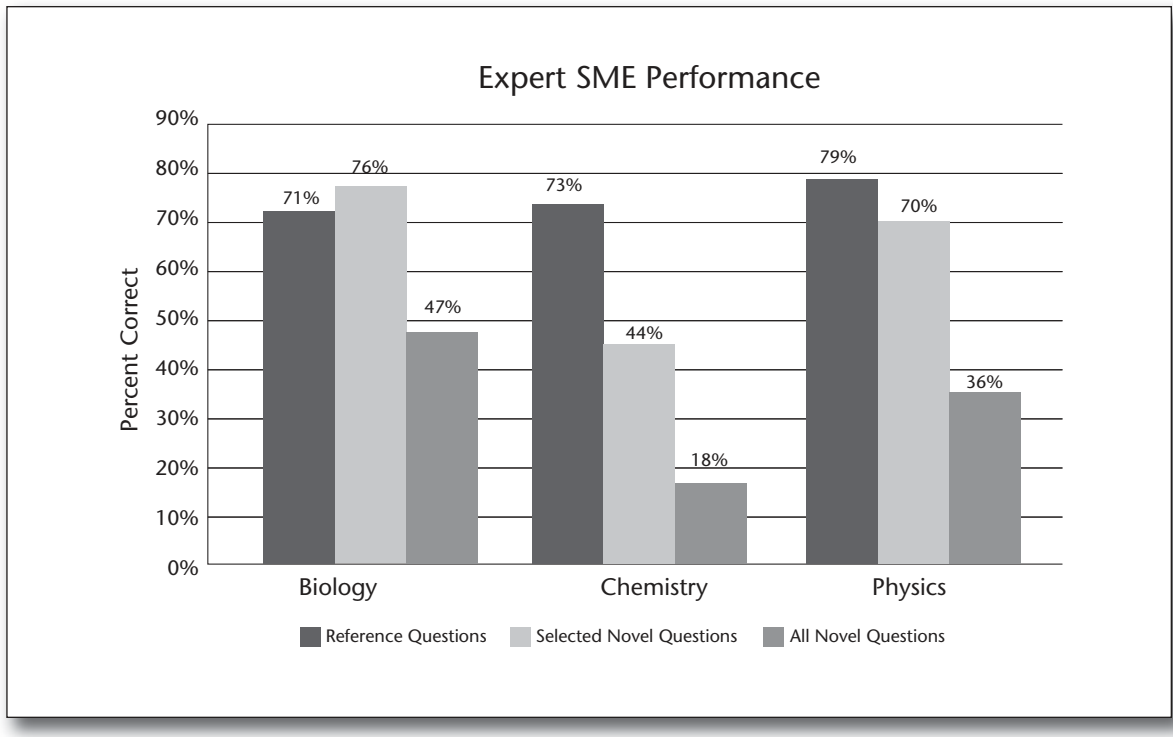

Figure 19. Expert SME Performance.

the nonexpert 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 questioners 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 knowledge bases authored by the expert SMEs (see figure 19). In biology and physics, the expert knowledge bases correctly answered more than 70 percent of the reference and selected questions and more than 40 percent of all novel questions. The expert chemistry knowledge base did not perform as well, especially for novel questions with a score of 18 percent for all novel questions and 44 percent 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 19 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 SME overtuned the knowledge base 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.

Second, we look at how the nonexpert SMEs did in comparison to the experts. The experimental design produced a 2x2 comparison of expert versus nonexpert performance for both KF and QF. To understand the 2x2 aspect of the experiment design, we can interpret the four points shown in figure 20 as follows: the upper-left point represents the question-answering correctness score when the knowledge was formulated by an expert SME, but the questions were asked by a nonexpert questioner; the lower-left point represents the situation when the knowledge was formulated by a nonexpert SME, and the questions were also asked by a nonexpert questioner. 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 20), we can see the effect of knowledge-formulation expertise by observing that the knowledge bases authored by

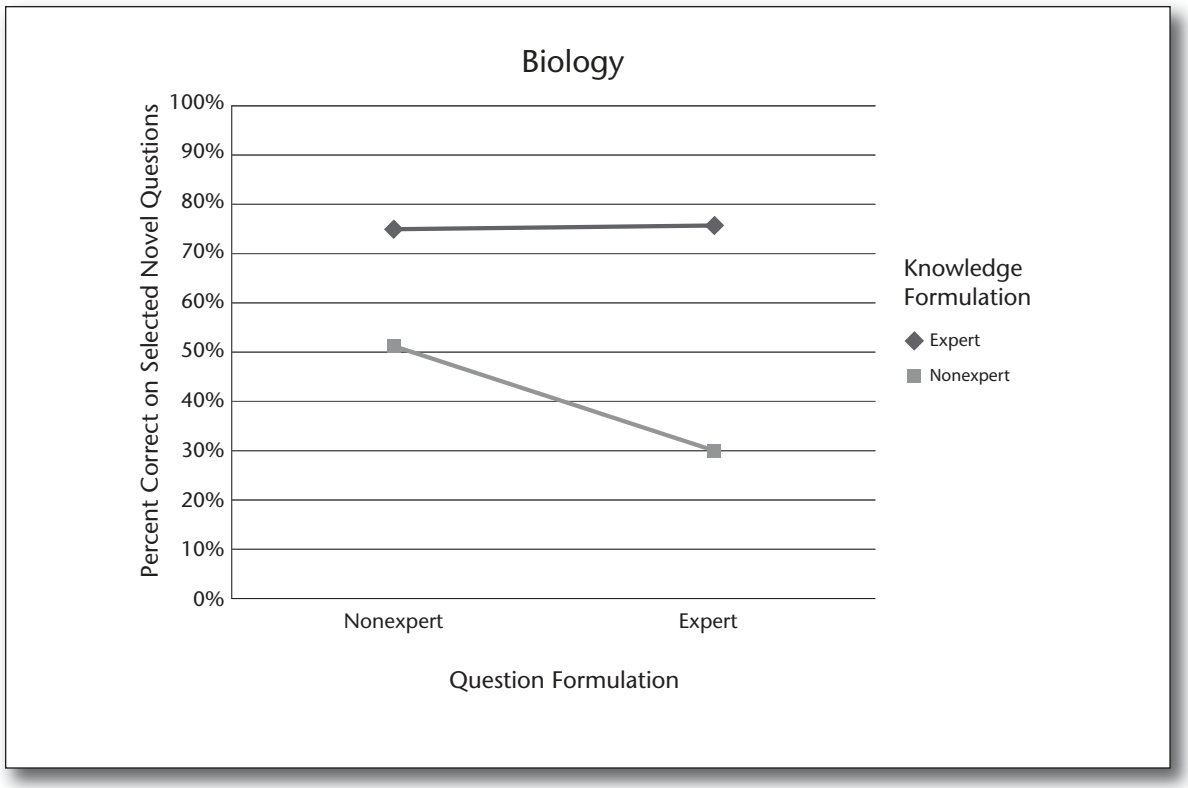

Figure 20. Experts Versus Nonexperts in Biology.

expert SMEs always had better scores than the knowledge bases authored by nonexpert SMEs. 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 knowledge bases that were authored by expert SMEs. But for knowledge bases authored by nonexpert SMEs, the nonexpert questioners outperformed the expert questioners. This is an anomaly, where it appeared that the nonexpert questioners outperformed the expert SMEs by 20 percent. Further analysis revealed that much of this difference resulted from the nonexpert SMEs being less rigorous in how they formulated questions. Some SMEs were taking short cuts in question formulation that would avoid the complexities of the full question. These simplified questions would produce a correct answer, but without requiring the system to compute all of the inferences implied by the question. We discount these differences as poor experimental control.

In chemistry (figure 21), there were no significant differences among the four conditions. Expert versus nonexpert KF was equivalent as was expert versus nonexpert QF.

In physics (figure 22), experts outperformed nonexperts in both KF and QF. Physics is the only domain where the experts outperformed nonexperts 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 nonexpert questioners to formulate. An obvious next question is to explain the reason for the differences between expert and nonexpert 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 19 that the expert-authored knowledge bases scored only 18 percent 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 21. The newly trained SMEs were able to perform as well as the expert SMEs, because the score of the expert SMEs was too low to start with.

The results for physics were easier to explain because there are known limitations of the system that make it harder for the SMEs 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 SMEs and nonexpert SMEs. The expert SMEs for biology had worked on

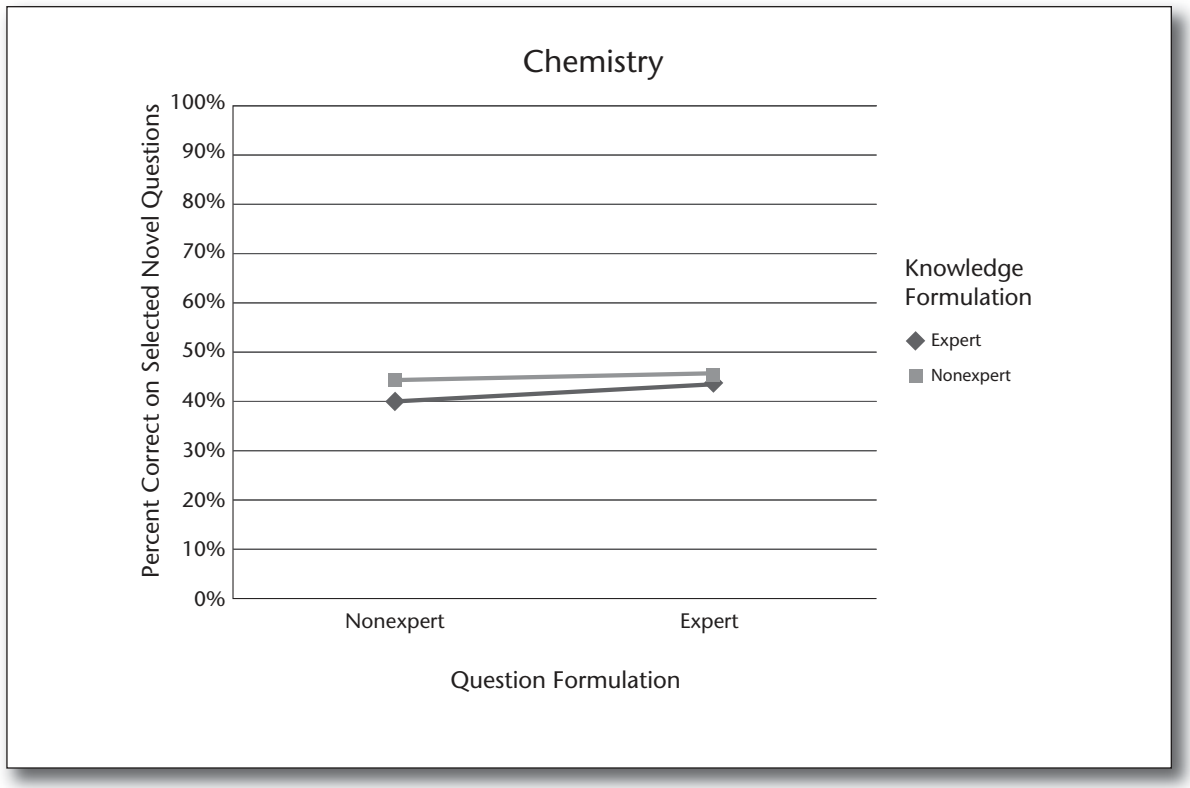

Figure 21. Experts Versus Nonexperts in Chemistry.

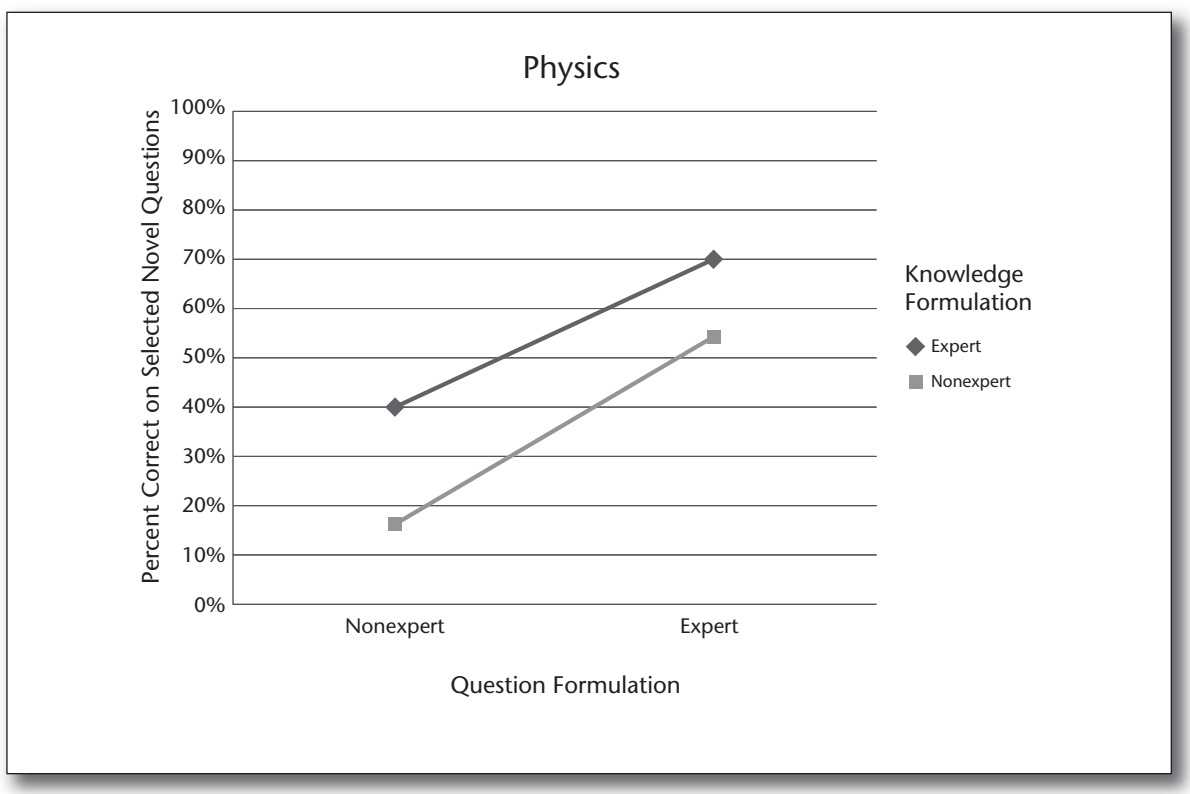

Figure 22. Experts Versus Nonexperts in Physics.

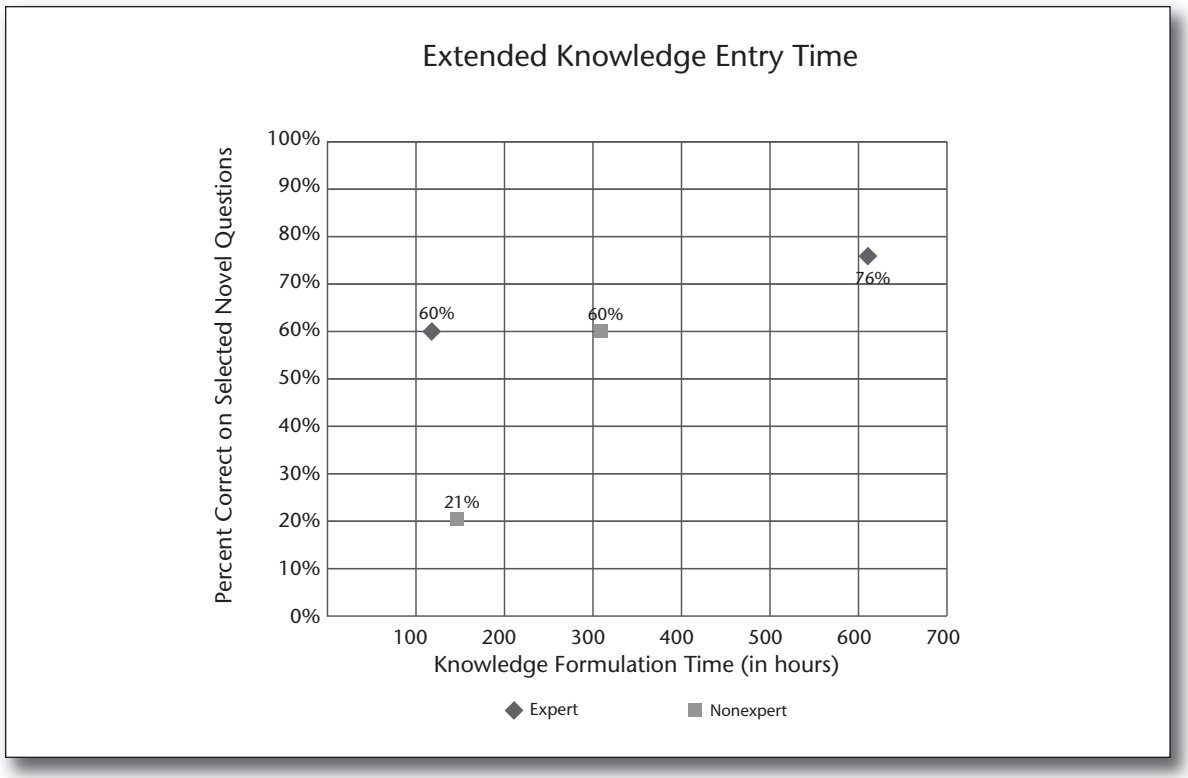

Figure 23. Results of Extended KF in Biology

their knowledge base for about 600 hours whereas nonexpert SMEs in the evaluation were limited to only 120 hours. Based on this limitation, 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 SME was asked to create the same biology knowledge base, but was limited to 120 hours for knowledge entry time. One of the better performing nonexpert SMEs was given an additional 180 hours, thus giving them a total of 300 hours, to continue authoring and refining their knowledge base. We show the result in figure 23. When the expert was limited to 120 hours of KF time and the nonexpert was allowed 300 hours, the two knowledge bases exhibited similar performance with 60 percent correct answers. The additional 180 hours of KF time improved the nonexpert's score form 21 percent to 60 percent. The subject reported that the extra time gave her a much better understanding of AURA, the knowledge entry process, and her knowledge base.

This result shows a steep improvement in the performance of a knowledge base authored by a newly trained SME as the knowledge entry time increased from 120 hours to 300 hours. The corresponding rate of improvement for an expert SME as they are given more knowledge entry time is much smaller. This is quite likely because the expert SME 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 knowledge base authored by a newly trained SME significantly improves in question-answering performance, and starts to approach the performance of an expert SME. This was an excellent result in support of AURA's ability to enable a newly trained SME to author competent knowledge bases.

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 SME can construct knowledge bases that, given sufficient knowledge entry time, approach in performance to the performance of the knowledge bases constructed by expert SMEs. For physics, the knowledge bases constructed by expert SMEs outperform the knowledge bases constructed by newly trained SMEs. For chemistry, while the results show that the performance of the knowledge bases authored by newly trained SMEs was very close to that of the knowledge bases authored by expert SMEs, 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 nonexpert questioners in the domains of biology and chemistry were able to perform question formulation as effectively as experts SMEs after only four hours of training. The nonexpert 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.

Multiuser Knowledge Entry Using a Team of SMEs

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 knowledge base from a full biology textbook.

The experiment results reported earlier involved only one user working in isolation in constructing a knowledge base. Such a constraint was an artifact of a controlled experiment and is no longer practical when a knowledge base is developed by a team of domain experts. So, as a step toward scaling to a knowledge base 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 knowledge base 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 SMEs as well as in-depth exposure to AURA. These knowledge engineering professionals 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 multiuser 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 knowledge base 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 knowledge base 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 24.

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 knowledge bases that performed very closely to those constructed by SRI's expert SMEs 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.

Second, we address the question: "Can a team of experts collaborate to create a shared knowledge base 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 knowledge bases were comparable to the scores on the ones authored by the expert SMEs at SRI.

Finally, because the score on the all novel questions on the knowledge base produced by the MUKE team (75 percent) is much higher than the corresponding score on the knowledge base produced by the expert SMEs (47 percent), one can naturally ask, "Did MUKE team outperform the expert SMEs at SRI?" Overall the answer would have to be a qualified "yes." It is difficult to compare knowledge entry rates of individual SMEs, because of differences in the knowledge entry process, resources, and conditions for the knowledge bases authored by the SRI expert SMEs and those by the MUKE team. Yet, it is clear that the overall performance of the MUKE team was superior.

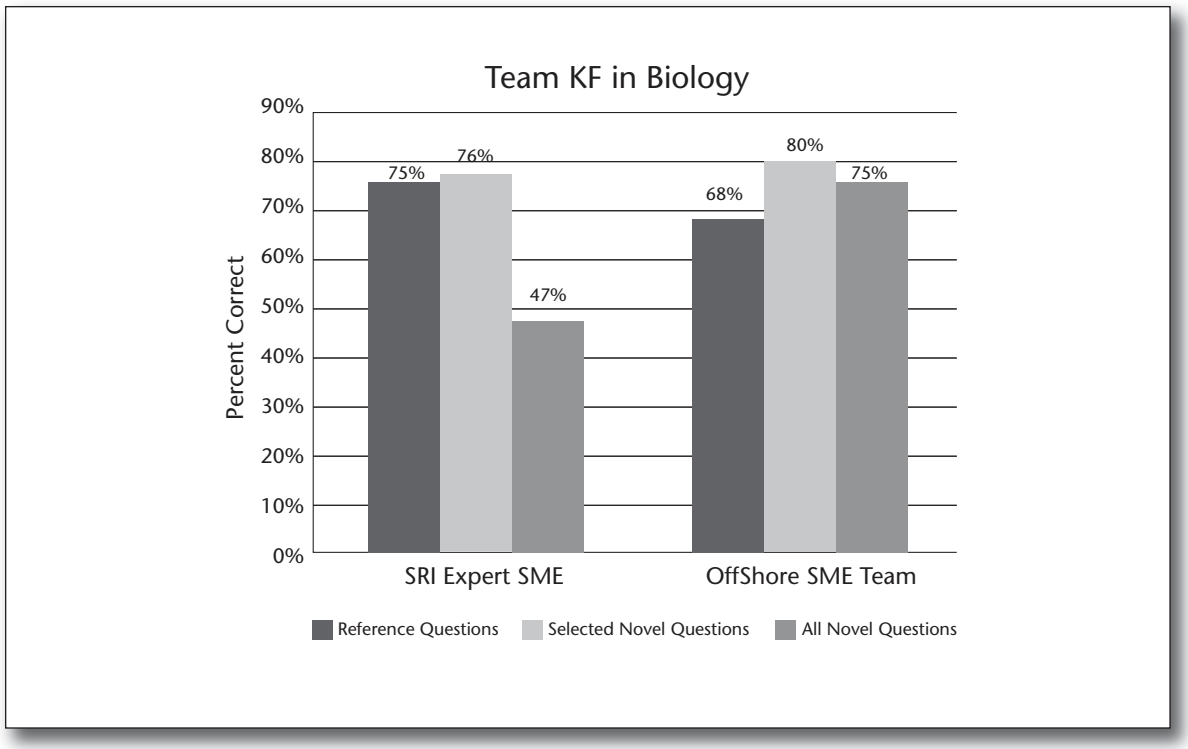

Figure 24. Multiuser KF Team Results.

Discussion

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

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

Moreover, the danger of overoptimizing a system to perform well on a specific test problem always exists — in ways that do not generalize to real-world problems. 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. 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 subquestions 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 multiuser 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. Nonexperts 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 knowledge base 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 percent barrier and can represent all knowledge in a full textbook. As mentioned earlier, we have performed analyses of the KRR 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 subevents, 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 nonexperts 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 KRR 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.

Comparison to Related Work

Several researchers have investigated the problem of representing textbook knowledge and answering questions about that knowledge. The earliest work on this topic was on answering Algebra questions (Bobrow 1964), and a recent work is on answering Physics questions (Klenk and Forbus 2009). A survey of similar systems was recently reported by Mukherjee and Garain (2008).

The knowledge bases built using AURA up to now are substantially smaller than in the Cyc knowledge base (Lenat 1995). It is still, however, useful to compare our approach with Cyc. Cyc's goal has been to capture human common sense while our focus has been focused on knowledge that is explicitly written down in textbooks and that can be empirically tested for completeness and coverage.

We can also draw parallels between various component technologies used in AURA and related work. For knowledge formulation, a closely related system was Visual Language (VL) (Gaines 1991). The primary difference between approach used in AURA and VL is that AURA works from the description of an example instance of a class while VL worked by directly editing the classes and their descriptions. The work most closely related to use of controlled English for question formulation is in natural language interfaces to databases that have more limited schema and do not support queries that might involve complex setup (Popescu, Etzioni, and Kautz 2003). The knowledge representation and reasoning used in AURA is comparable to a system that supports both description logic inference and horn rules and can be represented using an expressive representation language such as SILK (Grosof, Dean, and Kifer 2009).

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 KRR challenges.

Given the encouraging results for encoding basic conceptual knowledge into AURA, we plan to employ a multiuser collaborative KF team to encode all of the knowledge possible for an introductory biology textbook and then to see how well that knowledge base 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 knowledge base 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 knowledge base 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, Jameson, and Barbu 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. We have created an interface between SMW+ and AURA that enables users to import community-authored ontologies from SMW+ into AURA. 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 KRR challenges we listed. 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, Dean, and Kifer 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 KRR features, and eventually integrating the best features of AURA, SMW+, and SILK into the next generation Halo system.

In summary, Vulcan continues to make 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, biodiscovery, and business enterprise.

Note

1. For details on the AP exam, see www.collegeboard.com/student/testing/ap/about.html.

2. See KM — The Knowledge Machine 2.0: Users Manual (userweb.cs.utexas.edu/users/mfkb/km/userman.pdf).

References

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; Patel-Schneider, P. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. New York: Cambridge University Press.

Barker; K.; Porter, B.; and Clark, P. 2001. A Library of Generic Concepts for Composing Knowledge Bases. In *Proceedings of the First International Conference on Knowledge Capture,* 14–21. New York: Association for Computing Machinery.

Bobrow, D. G. 1964. A Question-Answering System for High School Algebra and Word Problems. In *Proceedings of the American Federation of Information Processing Societies Fall Joint Computer Conference*. New York: Association for Computing Machinery.

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

Campbell, N. A., and Reece, J. 2001. *Biology,* Sixth Edition. Menlo Park, CA: Benjamin Cummings.

Chaudhri, V. K.; Bredeweg, B.; Fikes, R.; McIlraith, S.; Wellman, M. 2010. A Categorization of KRR Methods for Requirement Analysis of a Query Answering Knowledge Base. In *Proceedings of the Sixth International Conference on Formal Ontologies in Information Systems*. Amsterdam: IOS Press.

Chaudhri, V. K.; Greaves, M.; Hansch, D.; Jameson, A.; and Pfisterer, F. 2008. Using a Semantic Wiki as a Knowledge Source for Question Answering. In *Symbiosis of Semantic Web and Knowledge Engineering: Papers from the Spring AAAI Symposium*, ed. D. Sleeman and M. Musen. Technical Report SS-08-07. Menlo Park, CA: Association for the Advancement of Artificial Intelligence..

Chaudhri, V. K.; John, B.; Mishra, S.; Pacheco, J.; Porter, B.; and Spaulding, A. 2007. Enabling Experts to Build Knowledge Bases from Science Textbooks. In *Proceedings of the Fourth International Conference on Knowledge Capture Systems (KCAP)*. New York: Association for Computing Machinery.

Chaudhri , V. K.; Murray, K. S.; Pacheco, J.; Clark, P.; Porter, B.; Hayes, P. J. 2004. Graph-Based Acquisition of Expressive Knolwedge. In *Proceedings of the 14th European Knowledge Acquisition Conference (EKAW)*. Lecture Notes in Computer Science 3257. Berlin: Springer.

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

Clark, P.; Chaudhri, V.; Mishra, S.; Thomere, J. 2003. Enabling Domain Experts to Convey Questions to a Machine: A Modified, Template-Based Approach. In *Proceedings of the 2nd International Conference on Knowledge Capture Systems (KCAP)*. New York: Association for Computing Machinery.

Clark, P.; Chaw, J.; Barker, K.; Chaudhri, V.; Harrison, P.; John, B. 2007. Capturing and Answering Questions Posed to a Knowledge-Based System. In *Proceedings of the Fourth International Conference on Knowledge Capture Systems (KCAP)*. New York: Association for Computing Machinery.

Clark, P.; Thompson, J.; Barker, K.; Porter, B.; Chaudhri, V.; Rodriguez, A.; Thomere, J.; Mishra, S.; Gil, Y.; Hayes, P.; Reichherzer, T. 2001. Knowledge Entry as Graphical Assembly of Components. In *Proceedings of the First International Conference on Knowledge Capture Systems (KCAP)*. New York: Association for Computing Machinery.

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

Felbaum C. 1998. *Wordnet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press.

Friedland, N.; Allen P.; Matthews, G.; Witbrock, M.; Baxter, D.; Curtis, J.; Shepard, B.; Miraglia, P.; Angele, J.; Staab, S.; Moench, E.; Oppermann, H.; Wenke, D.; Israel, D.; Chaudhri, V.; Porter, B.; Barker, K.; Fan, J.; Chaw, S. Y.; Yeh, P.; Tecuci, D.; Clark, P. 2004a. Project Halo: Toward a Digital Aristotle. *AI Magazine* 25(4): 29–47.

Friedland, N.; Allen, P.; Whitbrock, M.; Matthews, G.; Salay, N.; Miragalia, P. 2004b. Toward a Quantitative Platform-Independent Analysis of Knowledge Systems. In *Proceedings of the Ninth International Conference of Knowledge Representation and Reasoning*. Menlo Park, CA: AAAI Press.

Gaines, B. R. 1991. An Interactive Visual Language for Term Subsumption Languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence,* ed. J. Mylopoulos and R. Reiter. San Francisco: Morgan Kaufmann Publishers.

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

Grosof, B.; Dean, M.; and Kifer, M. 2009. The SILK System: Scalable Higher-Order Defeasible Rules. In *Rule Interchange and Applications: Proceedings of the International RuleML Symposium on Rule Interchange and Applications* (RuleML-2009). Berlin: Springer.

Klenk, M., and Forbus, K. 2009. Analogical Model Formulation for Transfer Learning in AP Physics. *Artificial Intelligence* 173(18): 1615–1638.

Lenat, D. 1995. Cyc: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM* 38(11): 33–38.

Mukherjee, A., and Garain, U. 2008. A Review of Methods for Automatic Understanding of Natural Language Mathematical Problems. *Artificial Intelligence Review* 29(1): 93–122.

Pfisterer F.; Jameson, A.; Barbu, C. 2008. User-Centered Design and Evaluation of Interface Enhancements to the Semantic Media Wiki. Paper presented at the Workshop on Semantic Web User Interaction, CHI 2008, Florence, Italy, 5–10 April.

Popescu, A.; Etzioni, O.; Kautz, H. 2003. Towards a Theory of Natural Language Interfaces to Databases. In *Proceedings of the Eighth International Conference on Intelligent User Interfaces*. New York: Association for Computing Machinery.

Reddy, R. 2003. Three Open Problems in AI. *Journal of the Association of Computational Machinery* 50(1): 83–86.

Wan, H.; Grosof, B.; Kifer, M.; Fodor, P.; and Liang, S. 2009. Logic Programming with Defaults and Argumentation Theories. In *Proceedings of the 25th International Conference on Logic Programming (ICLP 2009)*. Berlin: Springer-Verlag.

Voorhees, E., and Buckland, L. 2008. *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*. Gaithersberg, MD: National Institute of Standards and Technology.

David Gunning David Gunning is a senior research program manager at Vulcan Inc., leading the AURA and HaloBook developments. Prior to Vulcan, he served as a program manager at the US Defense Advanced Research Projects Agency (DARPA), where he led numerous AI projects, including Personalized Assistant that Learns (PAL) and Command Post of the Future (CPOF). He holds MS degrees in computer science from Stanford University and psychology from the University of Dayton.

Vinay K. Chaudhri (vinay.chaudhri@sri.com) received his PhD from University of Toronto. He is currently a program director for ontology management in the Artificial Intelligence Center at SRI International. His research focuses on knowledge acquisition, ontologies, and deductive question answering.

Peter E. Clark (peter.e.clark@boeing.com) is an associate technical fellow in the Networked Systems Technology group, Boeing Research & Technology, leading research in the areas of knowledge based systems, machine reasoning, and natural language processing.

Ken Barker (kbarker@cs.utexas.edu) is a research scientist in the Department of Computer Science at the University of Texas at Austin investigating knowledge-based systems and their application to language understanding. He received a Ph.D. in computational linguistics from the University of Ottawa in 1998.

Shaw-Yi Chaw (jchaw@cs.utexas.edu) received his Ph.D. from the University of Texas at Austin. Shaw-Yi contributed to the question-answering functionality of the AURA system. He is currently working at IBM's T. J. Watson Research Center on their Watson question-answering system. His research interests center on advancing knowledge-based systems as a feasible tool for knowledge management and evolution.

Mark Greaves is director of knowledge systems research at Vulcan Inc., where he leads advanced research in large knowledge bases and semantic web technologies. Prior to working at Vulcan, he served as director of DARPA's Joint Logistics Technology Office, as a program manager in DARPA's Information Exploitation Office, and as a computer scientist in Boeing's Mathematics and Computing Technology group. He currently serves as an advisor to several semantic web-oriented organizations and research and development groups. Greaves holds a PhD from Stanford University.

Benjamin Grosof is a senior research program manager at Vulcan Inc., leading the SILK knowledge representation effort on higher-order defeasible reactive semantic web rules and their piloting in deep question answering. Previously an information technology professor at the Massachusetts Institute of Technology Sloan Institute (2000-2007) and a senior software scientist at IBM Research (1988-2000), he holds a Stanford AI PhD and a Harvard BA. He has pioneered the key semantic rules knowledge representation technologies behind the recent W3C Rule Interchange Format and OWL RL (rules profile) standards, and their application in e-commerce, business policies, and finance.

Alice Leung is a senior scientist at Raytheon BBN Technologies. Leung's main research interest is the application of immersive and virtual world technologies for shaping and measuring human behavior. She is interested in the use of games and online communities for distributed problem solving and crowd-sourcing. Currently, she leads the experimentation thrust for the ARL Network Science CTA, an effort to understand universalities among different types of networks.

David D. McDonald has more than thirty years experience in artificial intelligence and computational linguistics research and development. He has worked in both academe and industry, and is the author of more than 60 refereed publications. McDonald has made significant contributions in the area of robust semantic parsing and knowledge representation. In addition to pursuing opportunities in these and other areas, he contributes to efforts on high accuracy parsing, synthetic agents, reasoning with rich ontologies, and applications of speech to experiential training

Sunil Mishra (smishra@ai.sri.com) is an alumnus of SRI's Artificial Intelligence Center and holds a Master's degree from Northwestern University.

John Pacheco (pacheco@ai.sri.com) is a senior research engineer in the SRI's Artificial Intelligence Center. He holds a Bachelor's degree in symbolic systems from Stanford University.

Bruce Porter (porter@cs.utexas.edu) is a professor in the Department of Computer Science at the University of Texas at Austin and the chairman of the Department. His research and teaching focuses on knowledge-based systems and the contributing technologies of knowledge representation and reasoning, machine learning, explanation generation, and natural language understanding.

Aaron Spaulding (spaulding@ai.sri.com) is a senior computer scientist and interaction designer at SRI's Artificial Intelligence Center. His work centers on developing usable interfaces for AI systems that meet real user needs. He holds a Master's degree in human computer interaction from Carnegie Mellon University.

Dan Tecuci (dan.tecuci@siemens.com) is a research scientist with Siemens Corporation, Corporate Research working in the Data Modeling and Optimization group. He received a Ph.D. in artificial intelligence from The University of Texas at Austin (UT) in 2007. The work reported here was completed when he was a research associate at UT. His interests include knowledge representation and reasoning, question answering, episodic memory and its applications.

Jing Tien (tien@ai.sri.com) is an interaction designer in SRI's Artificial Intelligence Center specializing in user experience for intelligent systems. She holds a Master's degree in human computer interaction from Carnegie Mellon University. Her research focuses on identifying user needs, designing user interface related solutions for AI challenges and conducting user studies to evaluate complex interaction.