AI Magazine Volume 19 Number 4 (1998) (© AAAI)

The DARPA High-Performance Knowledge Bases Project

Paul Cohen, Robert Schrag, Eric Jones, Adam Pease, Albert Lin, Barbara Starr, David Gunning, and Murray Burke

■ Now completing its first year, the High-Performance Knowledge Bases Project promotes technology for developing very large, flexible, and reusable knowledge bases. The project is supported by the Defense Advanced Research Projects Agency and includes more than 15 contractors in universities, research laboratories, and companies. The evaluation of the constituent technologies centers on two challenge problems, in crisis management and battlespace reasoning, each demanding powerful problem solving with very large knowledge bases. This article discusses the challenge problems, the constituent technologies, and their integration and evaluation.

Although a computer has beaten the world chess champion, no computer has the commonsense of a six-year-old child. Programs lack knowledge about the world sufficient to understand and adjust to new situations as people do. Consequently, programs have been poor at interpreting and reasoning about novel and changing events, such as international crises and battlefield situations. These problems are more open ended than chess. Their solution requires shallow knowledge about motives, goals, people, countries, adversarial situations, and so on, as well as deeper knowledge about specific political regimes, economies, geographies, and armies.

The High-Performance Knowledge Base (HPKB) Project is sponsored by the Defense Advanced Research Projects Agency (DARPA) to develop new technology for knowledge-based systems.[1] It is a three-year program, ending in fiscal year 1999, with funding totaling $34 million. HPKB technology will enable developers to rapidly build very large knowledge bases—on the order of 10^6 rules, axioms, or frames—enabling a new level of intelligence for military systems. These knowledge bases should be comprehensive and reusable across many applications, and they should be maintained and modified easily. Clearly, these goals require innovation in many areas, from knowledge representation to formal reasoning and special-purpose problem solving, from knowledge acquisition to information gathering on the web to machine learning, from natural language understanding to semantic integration of disparate knowledge bases.

For roughly one year, HPKB researchers have been developing knowledge bases containing tens of thousands of axioms concerning crises and battlefield situations. Recently, the technology was tested in a month-long evaluation involving sets of open-ended test items, most of which were similar to sample (training) items but otherwise novel. Changes to the crisis and battlefield scenarios were introduced during the evaluation to test the comprehensiveness and flexibility of knowledge in the HPKB systems. The requirement for comprehensive, flexible knowledge about general scenarios forces knowledge bases to be large. Challenge problems, which define the scenarios and thus drive knowledge base development, are a central innovation of HPKB. This article discusses HPKB challenge problems, technologies and integrated systems, and the evaluation of these systems.

The challenge problems require significant developments in three broad areas of knowledge-based technology. First, the overriding goal of HPKB—to be able to select, compose, extend, specialize, and modify components from a library of reusable ontologies, common domain theories, and generic problem-solving strategies—is not immediately achievable and requires some research into foundations of very large knowledge bases, particularly research in knowledge representation and ontological engineering. Second, there is the

problem of building on these foundations to populate very large knowledge bases. The goal is for collaborating teams of domain experts (who might lack training in computer science) to easily extend the foundation theories, define additional domain theories and problem-solving strategies, and acquire domain facts. Third, because knowledge is not enough, one also requires efficient problem-solving methods. HPKB supports research on efficient, general inference methods and optimized task-specific methods.

HPKB is a timely impetus for knowledge-based technology, although some might think it overdue. Some of the tenets of HPKB were voiced in 1987 by Doug Lenat and Ed Feigenbaum (Lenat and Feigenbaum 1987), and some have been around for longer. Lenat's CYC Project has also contributed much to our understanding of large knowledge bases and ontologies. Now, 13 years into the CYC Project and more than a decade after Lenat and Feigenbaum's paper, there seems to be consensus on the following points:

The first and most intellectually taxing task when building a large knowledge base is to design an ontology. If you get it wrong, you can expect ongoing trouble organizing the knowledge you acquire in a natural way. Whenever two or more systems are built for related tasks (for example, medical expert systems, planning, modeling of physical processes, scheduling and logistics, natural language understanding), the architects of the systems realize, often too late, that someone else has already done, or is in the process of doing, the hard ontological work. HPKB challenges the research community to share, merge, and collectively develop large ontologies for significant military problems. However, an ontology alone is not sufficient. Axioms are required to give meaning to the terms in an ontology. Without them, users of the ontology can interpret the terms differently.

Most knowledge-based systems have no common sense; so, they cannot be trusted. Suppose you have a knowledge-based system for scheduling resources such as heavy-lift helicopters, and none of its knowledge concerns noncombatant evacuation operations. Now, suppose you have to evacuate a lot of people. Lacking common sense, your system is literally useless. With a little common sense, it could not only support human planning but might be superior to it because it could think outside the box and consider using the helicopters in an unconventional way. Common sense is needed to recognize and exploit opportunities as well as avoid foolish mistakes.

Often, one will accept an answer that is roughly correct, especially when the alternatives are no answer at all or a very specific but wrong answer. This is Lenat and Feigenbaum's breadth hypothesis: "Intelligent performance often requires the problem solver to fall back on increasingly general knowledge, and/or to analogize to specific knowledge from far-flung domains" (Lenat and Feigenbaum 1987, p. 1173). We must, therefore, augment high-power knowledge-based systems, which give specific and precise answers, with weaker but adequate knowledge and inference. The inference methods might not all be sound and complete. Indeed, one might need a multitude of methods to implement what Polya called plausible inference. HPKB encompasses work on a variety of logical, probabilistic, and other inference methods.

It is one thing to recognize the need for commonsense knowledge, another to integrate it seamlessly into knowledge-based systems. Lenat observes that ontologies often are missing a middle level, the purpose of which is to connect very general ontological concepts such as *human* and *activity* with domain-specific concepts such as *the person who is responsible for navigating a B-52 bomber.* Because HPKB is grounded in domain-specific tasks, the focus of much ontological engineering is this middle layer.

The Participants

The HPKB participants are organized into three groups: (1) technology developers, (2) integration teams, and (3) challenge problem developers. Roughly speaking, the integration teams build systems with the new technologies to solve challenge problems. The integration teams are led by SAIC and Teknowledge. Each integration team fields systems to solve challenge problems in an annual evaluation. University participants include Stanford University, Massachusetts Institute of Technology (MIT), Carnegie Mellon University (CMU), Northwestern University, University of Massachusetts (UMass), George Mason University (GMU), and the University of Edinburgh (AIAI). In addition, SRI International, the University of Southern California Information Sciences Institute (USC-ISI), the Kestrel Institute, and TextWise, Inc., have developed important components. Information Extraction and Transport (IET), Inc., with Pacific Sierra Research (PSR), Inc., developed and evaluated the crisis-management challenge problem, and Alphatech, Inc., is responsible for the battlespace challenge problem.

Challenge Problems

A programmatic innovation of HPKB is challenge problems. The *crisis-management* challenge problem, developed by IET and PSR, is designed to exercise broad, relatively shallow knowledge about international tensions. The *battlespace* challenge problem, developed by Alphatech, Inc., has two parts, each designed to exercise relatively specific knowledge about activities in armed conflicts. *Movement analysis* involves interpreting vehicle movements detected and tracked by idealized sensors. The *workaround* problem is concerned with finding military engineering solutions to traffic-obstruction problems, such as destroyed bridges and blocked tunnels.

Good challenge problems must satisfy several, often conflicting, criteria. A challenge problem must be challenging: It must raise the bar for both technology and science. A problem that requires only technical ingenuity will not hold the attention of the technology developers, nor will it help the United States maintain its preeminence in science. Equally important, a challenge problem for a DARPA program must have clear significance to the United States Department of Defense (DoD). Challenge problems should serve for the duration of the program, becoming more challenging each year. This continuity is preferable to designing new problems every year because the infrastructure to support challenge problems is expensive.

A challenge problem should require little or no access to military subject-matter experts. It should not introduce a knowledge-acquisition bottleneck that results in delays and low productivity from the technology developers. As much as possible, the problem should be solvable with accessible, open-source material. A challenge problem should exercise all (or most) of the contributions of the technology developers, and it should exercise an integration of these technologies. A challenge problem should have unambiguous criteria for evaluating its solutions. These criteria need not be so objective that one can write algorithms to score performance (for example, human judgment might be needed to assess scores), but they must be clear and they must be published early in the program. In addition, although performance is important, challenge problems that value performance above all else encourage "one-off" solutions (a solution developed for a specific problem, once only) and discourage researchers from trying to understand why their technologies work well and poorly. A challenge problem should provide a steady stream of results, so progress can be assessed not only by technology developers but also by DARPA management and involved members of the DoD community.

The HPKB challenge problems are designed to support new and ongoing DARPA initiatives in intelligence analysis and battlespace information systems. Crisis-management systems will assist strategic analysts by evaluating the political, economic, and military courses of action available to nations engaged at various levels of conflict. Battlespace systems will support operations officers and intelligence analysts by inferring militarily significant targets and sites, reasoning about road network trafficability, and anticipating responses to military strikes.

Crisis-Management Challenge Problem

The crisis-management challenge problem is intended to drive the development of broad, relatively shallow commonsense knowledge bases to facilitate intelligence analysis. The client program at DARPA for this problem is Project GENOA—Collaborative Crisis Understanding and Management. GENOA is intended to help analysts more rapidly understand emerging international crises to preserve U.S. policy options. *Proactive crisis management*—before a situation has evolved into a crisis that might engage the U.S. military—enables more effective responses than reactive management. Crisis-management systems will assist strategic analysts by evaluating the political, economic, and military courses of action available to nations engaged at various levels of conflict.

The challenge problem development team worked with GENOA representatives to identify areas for the application of HPKB technology. This work took three or four months, but the crisis-management challenge problem specification has remained fairly stable since its initial release in draft form in July 1997.

The first step in creating the challenge problem was to develop a *scenario* to provide context for intelligence analysis in time of crisis. To ensure that the problem should require development of real knowledge about the world, the scenario includes real national actors with a fictional yet plausible story line. The scenario, which takes place in the Persian Gulf, involves hostilities between Saudi Arabia and Iran that culminate in closing the Strait of Hormuz to international shipping.

Next, IET worked with experts at PSR to develop a description of the intelligence analysis process, which involves the following tasks: information gathering—what happened, situation assessment—what does it mean, and sce-

III. What of significance might happen following the Saudi air strikes?
B. Options evaluation
Evaluate the options available to Iran.
Close the Strait of Hormuz to shipping.
Evaluation: Probable
Motivation: Respond to Saudi air strikes and deter future strikes.
Capability:
(Q51) Can Iran close the Strait of Hormuz to international shipping?
(Q83) Is Iran capable of firing upon tankers in the Strait of Hormuz? With what weapons?
Negative outcomes:
(Q53) What risks would Iran face in closing the strait?

Figure 1. Sample Questions Pertaining to the Responses to an Event.

nario development—what might happen next.

Situation assessment (or interpretation) includes factors that pertain to the specific situation at hand, such as motives, intents, risks, rewards, and ramifications, and factors that make up a general context, or "strategic culture," for a state actor's behavior in international relations, such as capabilities, interests, policies, ideologies, alliances, and enmities. Scenario development, or speculative prediction, starts with the generation of plausible actions for each actor. Then, options are evaluated with respect to the same factors for situation assessment, and a likelihood rating is produced. The most plausible actions are reported back to policy makers.

These analytic tasks afford many opportunities for knowledge-based systems. One is to use knowledge bases to retain or multiply corporate expertise; another is to use knowledge and reasoning to "think outside the box," to generate analytic possibilities that a human analyst might overlook. The latter task requires extensive commonsense knowledge, or "analyst's sense," about the domain to rule out implausible options.

The crisis-management challenge problem includes an informal specification for a prototype *crisis-management assistant* to support analysts. The assistant is tested by asking questions. Some are simple requests for factual information, others require the assistant to interpret the actions of nations in the context of strategic culture. Actions are motivated by interests, balancing risks and rewards. They have impacts and require capabilities. Interests drive the formation of alliances, the exercise of influence, and the generation of tensions among actors. These factors play out in a current and a historical context. Crises can be represented as events or as larger episodes tracking the evolution of a conflict over time, from inception or trigger, through any escalation, to eventual resolution or stasis. The representations being developed in HPKB are intended to serve as a crisis corporate memory to help analysts discover historical precedents and analogies for actions. Much of the challenge-problem specification is devoted to sample questions that are intended to drive the development of general models for reasoning about crisis events.

Sample questions are embedded in an analytic context. The question "What might happen next?" is instantiated as "What might happen following the Saudi air strikes?" as shown in figure 1. Q51 is refined to Q83 in a way that is characteristic of the analytic process; that is, higher-level questions are refined into sets of lower-level questions that provide detail.

The challenge-problem developers (IET with PSR) developed an answer key for sample questions, a fragment of which is shown in figure 2. Although simple factual questions (for example, "What is the gross national product of the United States?") have just one answer; questions such as Q53 usually have several. The answer key actually lists five answers, two of which are shown in figure 2. Each is accompanied by suitable explanations, including source material. The first source (*Convention on the Law of the Sea*) is electronic. IET maintains a web site with links to pages that are expected to be useful in answering the questions. The second source is a fragment of a model developed by IET and published in the challenge-problem specification. IET developed these fragments to

Answer(s):
1. Economic sanctions from {Saudi Arabia, GCC, U.S., U.N.}
   - The closure of the Strait of Hormuz would violate an international norm promoting freedom of the seas and would jeopardize the interests of many states.
   - In response, states might act unilaterally or jointly to impose economic sanctions on Iran to compel it to reopen the strait.
   - The United Nations Security Council might authorize economic sanctions against Iran.
2. Limited military response from {Saudi Arabia, GCC, U.S., others}…

Source(s):
- The *Convention on the Law of the Sea*.
- (B5) States may act unilaterally or collectively to isolate and/or punish a group or state that violates international norms. Unilateral and collective action can involve a wide range of mechanisms, such as intelligence collection, military retaliation, economic sanction, and diplomatic censure/isolation.

Figure 2. Part of the Answer Key for Question 53.

indicate the kinds of reasoning they would be testing in the challenge problem.

For the challenge-problem evaluation, held in June 1998, IET developed a way to generate test questions through parameterization. Test questions deviate from sample questions in specified, controlled ways, so the teams participating in the challenge problem know the space of questions from which test items will be selected. This space includes billions of questions so the challenge problem cannot be solved by relying on question-specific knowledge. The teams must rely on general knowledge to perform well in the evaluation. (Semantics provide practical constraints on the number of reasonable instantiations of parameterized questions, as do online sources provided by IET.) To illustrate, Q53 is parameterized in figure 3. Parameterized question 53, PQ53, actually covers 8 of the roughly 100 sample questions in the specification.

Parameterized questions and associated class definitions are based on natural language, giving the integration teams responsibility for developing (potentially different) formal representations of the questions. This decision was made at the request of the teams. An instance of a parameterized question, say, PQ53, is mechanically generated, then the teams must create a formal representation and reason with it—without human intervention.

PQ53 [During/After <TimeInterval>,] what {risks, rewards} would <InternationalAgent> face in <InternationalActionType>?

<InternationalActionType> =
{[exposure of its] {supporting, sponsoring} <InternationalAgentType> in <InternationalAgent2>, successful terrorist attacks against <InternationalAgent2>'s <EconomicSector>, <InternationalActionType>(PQ51), taking hostage citizens of <InternationalAgent2>, attacking targets <SpatialRelationship> <InternationalAgent2> with <Force>}

<InternationalAgentType> =
{terrorist group, dissident group, political party, humanitarian organization}

Figure 3. A Parameterized Question Suitable for Generating Sample Questions and Test Questions.

Battlespace Challenge Problems

The second challenge-problem domain for HPKB is battlespace reasoning. *Battlespace* is an abstract notion that includes not only the

The second challenge-problem domain for HPKB is battlespace reasoning. Battlespace *is an abstract notion that includes not only the physical geography of a conflict but also the plans, goals, and activities of all combatants prior to, and during, a battle and during the activities leading to the battle.*

physical geography of a conflict but also the plans, goals, and activities of all combatants prior to, and during, a battle and during the activities leading to the battle. Three battlespace programs within DARPA were identified as potential users of HPKB technologies: (1) the dynamic multiinformation fusion program, (2) the dynamic database program, and (3) the joint forces air-component commander (JFACC) program. Two battlespace challenge problems have been developed.

The Movement-Analysis Challenge Problem The *movement-analysis* challenge problem concerns high-level analysis of idealized sensor data, particularly the airborne JSTARS moving target indicator radar. This Doppler radar can generate vast quantities of information—one reading every minute for each vehicle in motion within a 10,000-square-mile area.[2] The movement-analysis scenario involves an enemy mobilizing a full division of ground forces—roughly 200 military units and 2000 vehicles—to defend against a possible attack. A simulation of the vehicle movements of this division was developed, the output of which includes reports of the positions of all the vehicles in the division at 1-minute intervals over a 4-day period for 18 hours each day. These military vehicle movements were then interspersed with plausible civilian traffic to add the problem of distinguishing military from nonmilitary traffic. The movement-analysis task is to monitor the movements of the enemy to detect and identify types of military site and convoy.

Because HPKB is not concerned with signal processing, the input are not real JSTARS data but are instead generated by a simulator and preprocessed into vehicle *tracks*. There is no uncertainty in vehicle location and no radar shadowing, and each vehicle is always accurately identified by a unique bumper number. However, vehicle tracks do not precisely identify vehicle type but instead define each vehicle as either light wheeled, heavy wheeled, or tracked. Low-speed and stationary vehicles are not reported.

Vehicle-track data are supplemented by small quantities of high-value intelligence data, including accurate identification of a few key enemy sites, *electronic intelligence* reports of locations and times at which an enemy radar is turned on, *communications intelligence* reports that summarize information obtained by monitoring enemy communications, and *human intelligence* reports that provide detailed information about the numbers and types of vehicle passing a given location. Other input include a detailed road network in electronic form and an order of battle that describes the structure and composition of the enemy forces in the scenario region.

Given these input, movement analysis comprises the following tasks:

First is to distinguish military from nonmilitary traffic. Almost all military traffic travels in convoys, which makes this task fairly straightforward except for very small convoys of two or three vehicles. Second is to identify the sites between which military convoys travel, determine which of these sites are militarily significant, and determine the types of each militarily significant site. Site types include battle positions, command posts, support areas, air-defense sites, artillery sites, and assembly-staging areas.

Third is to identify which units (or parts of units) in the enemy order of battle are participating in each military convoy.

Fourth is to determine the purpose of each convoy movement. Purposes include reconnaissance, movement of an entire unit toward a battle position, activities by command elements, and support activities.

Fifth is to infer the exact types of the vehicles that make up each convoy. About 20 types of military vehicle are distinguished in the enemy order of battle, all of which show up in the scenario data.

To help the technology base and the integration teams develop their systems, a portion of the simulation data was released in advance of the evaluation phase, accompanied by an answer key that supplied model answers for each of the inference tasks listed previously.

Movement analysis is currently carried out manually by human intelligence analysts, who appear to rely on models of enemy behavior at several levels of abstraction. These include models of how different sites or convoys are structured for different purposes and models of military systems such as logistics (supply and resupply). For example, in a logistics model, one might find the following fragment: "Each echelon in a military organization is responsible for resupplying its subordinate echelons. Each echelon, from battalion on up, has a designated area for storing supplies. Supplies are provided by higher echelons and transshipped to lower echelons at these areas." Model fragments such as these are thought to constitute the knowledge of intelligence analysts and, thus, should be the content of HPKB movement-analysis systems. Some such knowledge was elicited from military intelligence analysts during programwide meetings. These same analysts also scripted the simulation scenario.

The Workaround Challenge Problem

The *workaround challenge problem* supports air-campaign planning by the JFACC and his/her staff. One task for the JFACC is to determine suitable targets for air strikes. Good targets allow one to achieve maximum military effect with minimum risk to friendly forces and minimum loss of life on all sides. Infrastructure often provides such targets: It can be sufficient to destroy supplies at a few key sites or critical nodes in a transportation network, such as bridges along supply routes. However, bridges and other targets can be repaired, and there is little point in destroying a bridge if an available fording site is nearby. If a plan requires an interruption in traffic of several days, and the bridge can be repaired in a few hours, then another target might be more suitable. Target selection, then, requires some reasoning about how an enemy might "work around" the damage to the target.

The task of the workaround challenge problem is to automatically assess how rapidly and by what method an enemy can reconstitute or bypass damage to a target and, thereby, help air-campaign planners rapidly choose effective targets. The focus of the workaround problem in the first year of HPKB is automatic workaround generation.

The workaround task involves detailed representation of targets and the local terrain around the target and detailed reasoning about actions the enemy can take to reconstitute or bypass this damage. Thus, the input to workaround systems include the following elements:

First is a description of a target (for example, a bridge or a tunnel), the damage to it (for example, one span of a bridge is dropped; the bridge and vicinity are mined), and key features of the local terrain (for example, the slope and soil types of a terrain cross section coincident with the road near the bridge, together with the maximum depth and the speed of any river or stream the bridge crosses).

Second is a specific enemy unit or capability to be interdicted, such as a particular armored battalion or supply trucks carrying ammunition.

Third is a time period over which this unit or capability is to be denied access to the targeted route. The presumption is that the enemy will try to repair the damage within this time period; a target is considered to be effective if there appears to be no way for the enemy to make this repair.

Fourth is a detailed description of the enemy resources in the area that could be used to repair the damage. For the most part, repairs to battle damage are carried out by Army engineers; so, this description takes the form of a detailed engineering order of battle.

All input are provided in a formal representation language.

The workaround generator is expected to provide three output: First is a *reconstitution schedule,* giving the capacity of the damaged link as a function of time since the damage was inflicted. For example, the workaround generator might conclude that the capacity of the link is zero for the first 48 hours, but thereafter, a temporary bridge will be in place that can sustain a capacity of 170 vehicles an hour. Second is a *time line of engineering actions* that the enemy might carry out to implement the repair, the time these actions require, and the temporal constraints among them. If there appears to be more than one viable repair strategy, a time line should be provided for each. Third is a *set of required assets* for each time line of actions, a description of the engineering resources that are used to repair the damage and pointers to the actions in the time line that utilize these assets. The reconstitution schedule provides the minimal information required to evaluate the suitability of a given target. The time line of actions provides an explanation to justify the reconstitution schedule. The set of required assets is easily derived from the time line of actions and can be used to suggest further targets for preemptive air strikes against the enemy to frustrate its repair efforts.

A training data set was provided to help developers build their systems. It supplied input and output for several sample problems, together with detailed descriptions of the calculations carried out to compute action durations; lists of simplifying assumptions made to facilitate these calculations; and pointers to text sources for information on engineering resources and their use (mainly Army field manuals available on the World Wide Web).

Workaround generation requires detailed knowledge about what the capabilities of the enemy's engineering equipment are and how it is typically used by enemy forces. For example, repairing damage to a bridge typically involves mobile bridging equipment, such as armored vehicle-launched bridges (AVLBs), medium girder bridges, Bailey bridges, or float bridges such as ribbon bridges or M4T6 bridges, together with a range of earthmoving equipment such as bulldozers. Each kind of mobile bridge takes a characteristic amount of time to deploy, requires different kinds of bank preparation, and is "owned" by different echelons in the military hierarchy, all of which

The challenge problems are solved by integrated systems fielded by integration teams led by Teknowledge and SAIC.

affect the time it takes to bring the bridge to a damage site and effect a repair. Because HPKB operates in an entirely unclassified environment, U.S. engineering resources and doctrine were used throughout. Information from Army field manuals was supplemented by a series of programwide meetings with an Army combat engineer, who also helped construct sample problems and solutions.

Integrated Systems

The challenge problems are solved by integrated systems fielded by integration teams led by Teknowledge and SAIC. Teknowledge favors a centralized architecture that contains a large commonsense ontology (CYC); SAIC has a distributed architecture that relies on sharing specialized domain ontologies and knowledge bases, including a large upper-level ontology based on the merging of CYC, SENSUS, and other knowledge bases.

Teknowledge favors a centralized architecture that contains a large commonsense ontology (CYC) ....

Teknowledge Integration

The Teknowledge integration team comprises Teknowledge, Cycorp, and Kestrel Institute. Its focus is on semantic integration and the creation of massive amounts of knowledge.

Semantic Integration Three issues make software integration difficult. *Transport* issues concern mechanisms to get data from one process or machine to another. Solutions include sockets, remote-method invocation (RMI), and CORBA. *Syntactic* issues concern how to convert number formats, "syntactic sugar," and the labels of data. The more challenging issues concern *semantic integration*: To integrate elements properly, one must understand the meaning of each. The database community has addressed this issue (Wiederhold 1996); it is even more pressing in knowledge-based systems.

The current state of practice in software integration consists largely of interfacing pairs of systems, as needed. Pairwise integration of this kind does not scale up, unanticipated uses are hard to cover later, and chains of integrated systems at best evolve into stovepipe systems. Each integration is only as general as it needs to be to solve the problem at hand.

Some success has been achieved in low-level integration and reuse; for example, systems that share scientific subroutine libraries or graphics packages are often forced into similar representational choices for low-level data. DARPA has invested in early efforts to create reuse libraries for integrating large systems at higher levels. Some effort has gone into expressing a generic semantics of plans in an object-oriented format (Pease and Carrico 1997a, 1997b), and applications of this generic semantics to domain-specific tasks are promising (Pease and Albericci 1998). The development of ontologies for integrating manufacturing planning applications (Tate 1998) and work flow (Lee et al. 1996) is ongoing.

Another option for semantic integration is software mediation (Park, Gennari, and Musen 1997). This software mediation can be seen as a variant on pairwise integration, but because integration is done by knowledge-based means, one has an explicit expression of the semantics of the conversion. Researchers at Kestrel Institute have successfully defined formal specifications for data and used these theories to integrate formally specified software. In addition, researchers at Cycorp have successfully applied CYC to the integration of multiple databases.

The Teknowledge approach to integration is to share knowledge among applications and create new knowledge to support the challenge problems. Teknowledge is defining formal semantics for the input and output of each application and the information in the challenge problems.

Many concepts defy simple definitions. Although there has been much success in defining the semantics of mathematical concepts, it is harder to be precise about the semantics of the concepts people use every day. These concepts seem to acquire meaning through their associations with other concepts, their use in situations and communication, and their relations to instances. To give the concepts in our integrated system real meaning, we must provide a rich set of associations, which requires an extremely large knowledge base. CYC offers just such a knowledge base.

CYC (Lenat 1995; Lenat and Guha 1990) consists of an immense, multicontextual knowledge base; an efficient inference engine; and associated tools and interfaces for acquiring, browsing, editing, and combining knowledge. Its premise is that knowledge-based software will be less brittle if and only if it has access to a foundation of basic commonsense knowledge. This semantic substratum of terms, rules, and relations enables application programs to cope with unforeseen circumstances and situations.

The CYC knowledge base represents millions of hand-crafted axioms entered during the 13 years since CYC's inception. Through careful policing and generalizing, there are now slightly fewer than 1 million axioms in the knowledge base, interrelating roughly 50,000

atomic terms. Fewer than two percent of these axioms represent simple facts about proper nouns of the sort one might find in an almanac. Most embody general consensus information about the concepts. For example, one axiom says one cannot perform volitional actions while one sleeps, another says one cannot be in two places at once, and another says you must be at the same place as a tool to use it. The knowledge base spans human capabilities and limitations, including information on emotions, beliefs, expectations, dreads, and goals; common everyday objects, processes, and situations; and the physical universe, including such phenomena as time, space, causality, and motion.

The CYC inference engine comprises an epistemological and a heuristic level. The *epistemological level* is an expressive *n*th-order logical language with clean formal semantics. The *heuristic level* is a set of some three dozen special-purpose modules that each contains its own algorithms and data structures and can recognize and handle some commonly occurring sorts of inference. For example, one heuristic-level module handles temporal reasoning efficiently by converting temporal relations into a before-and-after graph and then doing graph searching rather than theorem proving to derive an answer. A truth maintenance system and an argumentation-based explanation and justification system are tightly integrated into the system and are efficient enough to be in operation at all times. In addition to these inference engines, CYC includes numerous browsers, editors, and consistency checkers. A rich interface has been defined.

Crisis-Management Integration The crisis-management challenge problem involves answering test questions presented in a structured grammar. The first step in answering a test question is to convert it to a form that CYC can reason with, a declarative decision tree. When the tree is applied to the test question input, a CYC query is generated and sent to CYC.

Answering the challenge problem questions takes a great deal of knowledge. For the first year's challenge problem alone, the Cycorp and Teknowledge team added some 8,000 concepts and 80,000 assertions to CYC. To meet the needs of this challenge problem, the team created significant amounts of new knowledge, some developed by collaborators and merged into CYC, some added by automated processes.

The Teknowledge integrated system includes two natural language components: START and TextWise. The START system was created by Boris Katz (1997) and his group at MIT. For each of the crisis-management questions, Teknowledge has developed a template into which user-specified parameters can be inserted. START parses English queries for a few of the crisis-management questions to fill in these templates. Each filled template is a legal CYC query. TextWise Corporation has been developing natural language information-retrieval software primarily for news articles (Liddy, Paik, and McKenna 1995). Teknowledge intends to use the TextWise knowledge base information tools (KNOW-IT) to supply many instances to CYC of facts discovered from news stories. The system can parse English text and return a series of binary relations that express the content of the sentences. There are several dozen relation types, and the constants that instantiate each relation are WORDNET synset mappings (Miller et al. 1993). Each of the concepts has been mapped to a CYC expression, and a portion of WORDNET has been mapped to CYC. For those synsets not in CYC, the WORDNET hyponym links are traversed until a mapped CYC term is found.

Battlespace Integration Teknowledge supported the movement-analysis workaround problem.

Movement analysis: Several movement-analysis systems were to be integrated, and much preliminary integration work was done. Ultimately, the time pressure of the challenge problem evaluation precluded a full integration. The MIT and UMass movement-analysis systems are described briefly here; the SMI and SRI systems are described in the section entitled SAIC Integrated Systems.

The MIT MAITA system provides tools for constructing and controlling networks of distributed-monitoring processes. These tools provide access to large knowledge bases of monitoring methods, organized around the hierarchies of tasks performed, knowledge used, contexts of application, the alerting of utility models, and other dimensions. Individual monitoring processes can also make use of knowledge bases representing commonsense or expert knowledge in conducting their reasoning or reporting their findings. MIT built monitoring processes for sites and convoys with these tools.

The UMass group tried to identify convoys and sites with very simple rules. Rules were developed for three site types: (1) battle positions, (2) command posts, and (3) assembly-staging areas. The convoy detector simply looked for vehicles traveling at fixed distances from one another. Initially, UMass was going to recognize convoys from their dynamics, in which the distances between vehicles fluctuate in a characteristic way, but in the simulated

… SAIC has a distributed architecture that relies on sharing specialized domain ontologies and knowledge bases ….

data, the distances between vehicles remained fixed. UMass also intended to detect sites by the dynamics of vehicle movements between them, but no significant dynamic patterns could be found in the movement data.

Workarounds: Teknowledge developed two workaround integrations, one an internal Teknowledge system, the other from AIAI at the University of Edinburgh.

Teknowledge developed a planning tool based on CYC, essentially a wrapper around CYC's existing knowledge base and inference engine. A plan is a proof that there is a path from the final goal to the initial situation through a partially ordered set of actions. The rules in the knowledge base driving the planner are rules about action preconditions and about which actions can bring about a certain state of affairs. There is no explicit temporal reasoning; the partial order of temporal precedence between actions is established on the basis of the rules about preconditions and effects.

The planner is a new kind of inference engine, performing its own search but in a much smaller search space. However, each step in the search involves interaction with the existing inference engine by hypothesizing actions and microtheories and doing asks and asserts in these microtheories. This hypothesizing and asserting on the fly in effect amounts to dynamically updating the knowledge base in the course of inference; this capability is new for the CYC inference engine.

Consistent with the goals of HPKB, the Teknowledge workaround planner reused CYC's knowledge, although it was not knowledge specific to workarounds. In fact, CYC had never been the basis of a planner before, so even stating things in terms of an action's preconditions was new. What CYC provided, however, was a rich basis on which to build workaround knowledge. For example, the Teknowledge team needed to write only one rule to state "to use something as a device you must have control over that device," and this rule covered the cases of using an M88 to clear rubble, a mine plow to breach a minefield, a bulldozer to cut into a bank or narrow the gap, and so on. The reason one rule can cover so many cases is because clearing rubble, demining an area, narrowing a gap, and cutting into a bank are all specializations of IntrinsicStateChangeEvent, an extant part of the CYC ontology.

The AIAI workaround planner was also implemented in CYC and took data from Teknowledge's FIRE&ISE-TO-MELD translator as its input. The central idea was to use the scriptlike structure of workaround plans to guide the reasoning process. For this reason, a hierarchical task network (HTN) approach was taken. A planning-specific ontology was defined within the larger CYC ontology, and planning rules only referenced concepts within this more constrained context. The planning application was essentially embedded in CYC.

CYC had to be extended to represent composite actions that have several alternative decompositions and complex preconditions-effects. Although it is not a commonsense approach, AIAI decided to explore HTN planning because it appeared suitable for the workaround domain. It was possible to represent actions, their conditions and effects, the plan-node network, and plan resources in a relational style. The structure of a plan was implicitly represented in the proof that the corresponding composite action was a relation between particular sets of conditions and effects. Once proved, action relations are retained by CYC and are potentially reusable. An advantage of implementing the AIAI planner in CYC was the ability to remove brittleness from the planner-input knowledge format; for example, it was not necessary to account for all the possible permutations of argument order in predicates such as *bordersOn* and *between*.

SAIC Integrated System

SAIC built an HPKB integrated knowledge environment (HIKE) to support both crisis-management and battlespace challenge problems. The architecture of HIKE for crisis management is shown in figure 4. For battlespace, the architecture is similar in that it is distributed and relies on the open knowledge base connectivity (OKBC) protocol, but of course, the components integrated by the battlespace architecture are different. HIKE's goals are to address the distributed communications and interoperability requirements among the HPKB technology components—knowledge servers, knowledge-acquisition tools, question-and-answering systems, problem solvers, process monitors, and so on—and provide a graphic user interface (GUI) tailored to the end users of the HPKB environment.

HIKE provides a distributed computing infrastructure that addresses two types of communications needs: First are input and output data-transportation and software connectivities. These include connections between the HIKE server and technology components, connections between components, and connections between servers. HIKE encapsulates information content and data transportation through JAVA objects, hypertext transfer protocol (HTTP), remote-method invocation (JAVA

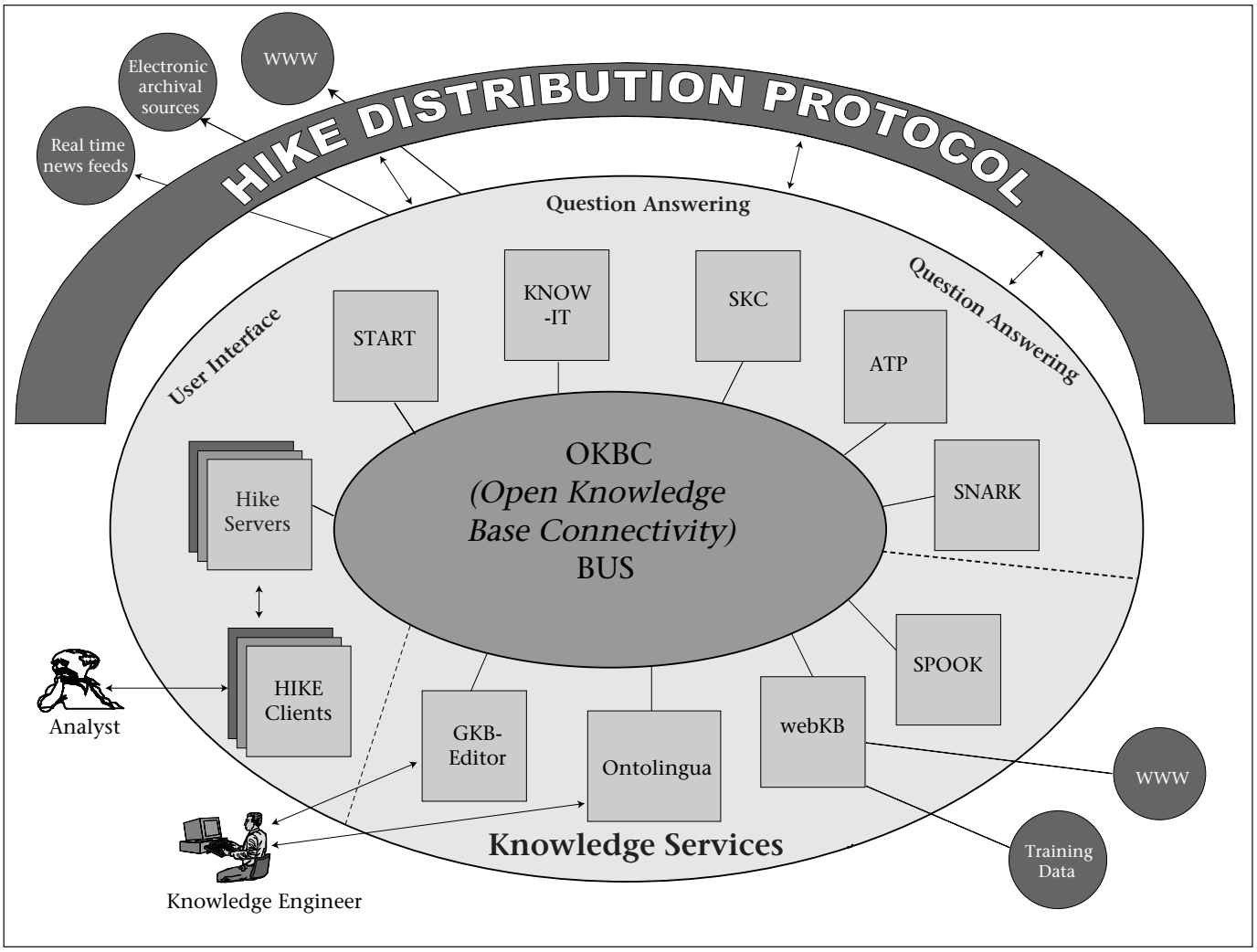

Figure 4. SAIC Crisis-Management Challenge Problem Architecture.

RMI), and database access (JDBC). Second, HIKE provides for knowledge-content assertion and distribution and query requests to knowledge services.

The OKBC protocol proved essential. SRI used it to interface the theorem prover SNARK to an OKBC server storing the Central Intelligence Agency (CIA) *World Fact Book* knowledge base. Because this knowledge base is large, SRI did not want to incorporate it into SNARK but instead used the procedural attachment feature of SNARK to look up facts that were available only in the *World Fact Book*. MIT's START system used OKBC to connect to SRI's OCELOT-SNARK OKBC server. This connection will eventually give users the ability to pose questions in English, which are then transformed to a formal representation by START and shipped to SNARK using OKBC; the result is returned using OKBC. ISI built an OKBC server for their LOOM system for GMU. SAIC built a front end to the OKBC server for LOOM that was extensively used by the members of the battlespace challenge problem team.

With OKBC and other methods, the HIKE infrastructure permits the integration of new technology components (either clients or servers) in the integrated end-to-end HPKB system without introducing major changes, provided that the new components adhere to the specified protocols.

Crisis-Management Integration The SAIC crisis-management architecture is focused around a central OKBC bus, as shown in figure 4. The technology components provide user interfaces, question answering, and knowledge services. Some components have overlapping roles. For example, MIT's START system serves both as a user interface and a question-answering component. Similarly, CMU's

WEBKB supports both question answering and knowledge services.

HIKE provides a form-based GUI with which users can construct queries with pull-down menus. Query-construction templates correspond to the templates defined in the crisis-management challenge problem specification. Questions also can be entered in natural language. START and the TextWise component accept natural language queries and then attempt to answer the questions. To answer questions that involve more complex types of reasoning, START generates a formal representation of the query and passes it to one of the theorem provers.

The Stanford University Knowledge Systems Laboratory ONTOLINGUA, SRI International's graphic knowledge base (GKB) editor, WEBKB, and TextWise provide the knowledge service components. The GKB editor is a graphic tool for browsing and editing large knowledge bases, used primarily for manual knowledge acquisition. WEBKB supports semiautomatic knowledge acquisition. Given some training data and an ontology as input, a web spider searches in a directed manner and populates instances of classes and relations defined in the ontology. Probabilistic rules are also extracted. TextWise extracts information from text and newswire feeds, converting them into knowledge interchange format (KIF) triples, which are then loaded into ONTOLINGUA. ONTOLINGUA is SAIC's central knowledge server and information repository for the crisis-management challenge problem. ONTOLINGUA supports KIF as well as compositional modeling language (CML). Flow models developed by Northwestern University (NWU) answer challenge problem questions related to world oil-transportation networks and reside within ONTOLINGUA. Stanford's system for probabilistic object-oriented knowledge (SPOOK) provides a language for class frames to be annotated with probabilistic information, representing uncertainty about the properties of instances in this class. SPOOK is capable of reasoning with the probabilistic information based on Bayesian networks.

The guiding philosophy during knowledge base development for crisis management was to reuse knowledge whenever it made sense.

Question answering is implemented in several ways. SRI International's SNARK and Stanford's abstract theorem prover (ATP) are first-order theorem provers. WEBKB answers questions based on the information it gathers. Question answering is also accomplished by START and TextWise taking a query in English as input and using information retrieval to extract the answers from text-based sources (such as the web, newswire feeds).

The guiding philosophy during knowledge base development for crisis management was to reuse knowledge whenever it made sense. The SAIC team reused three knowledge bases: (1) the HPKB upper-level ontology developed by Cycorp, (2) the *World Fact Book* knowledge base from the CIA, and the Units and Measures Ontology from Stanford. Reusing the upper-level ontology required translation, comprehension, and reformulation. The ontology was released in MELD (a language used by Cycorp) and was not directly readable by the SAIC system. In conjunction with Stanford, SRI developed a translator to load the upper-level ontology into any OKBC-compliant server. Once loaded into the OCELOT server, the GKB editor was used to comprehend the upper ontology. The graphic nature of the GKB editor illuminated the interrelationships between classes and predicates of the upper-level ontology. Because the upper-level ontology represents functional relationships as predicates but SNARK reasons efficiently with functions, it was necessary to reformulate the ontology to use functions whenever a functional relationship existed.

Battlespace Integration The distributed HIKE infrastructure is well suited to support an integrated battlespace challenge problem as it was originally designed: a single information system for movement analysis, trafficability, and workaround reasoning. However, the trafficability problem (establishing routes for various kinds of vehicle given the characteristics) was dropped, and the integration of the other problems was delayed. The components that solved these problems are described briefly later.

Movement analysis: The movement-analysis problem is solved by MIT's monitoring, analysis, and interpretation tools arsenal (MAITA); Stanford University's Section on Medical Informatics' (SMI) problem-solving methods; and SRI International's Bayesian networks. The MIT effort was described briefly in the section entitled Teknowledge Integration. Here, we focus on the SMI and SRI movement-analysis systems.

For scalability, SMI adopted a three-layered approach to the challenge problem: The first layer consisted primarily of simple, context-free data processing that attempted to find important preliminary abstractions in the data set. The most important of these were *traffic centers* (locations that were either the starting or stopping points for a significant number of vehicles) and *convoy segments* (a number of vehicles that depart from the same traffic center at roughly the same time, going in roughly the same direction). Spotting these abstractions required setting a number of parameters (for example, how big a traffic center is). Once trained, these first-layer algorithms are linear in

the size of the data set and enabled SMI to use knowledge-intensive techniques on the resulting (much smaller) set of data abstractions.

The second layer was a repair layer, which used knowledge of typical convoy behaviors and locations on the battlespace to construct a "map" of militarily significant traffic and traffic centers. The end result was a network of traffic connected by traffic. Three main tasks remain: (1) classify the traffic centers, (2) figure out what the convoys are doing, and (3) identify which units are involved. SMI iteratively answered these questions by using repeated layers of heuristic classification and constraint satisfaction. The heuristic-classification components operated independently of the network, using known (and deduced) facts about single convoys or traffic centers. Consider the following rule for trying to identify a main supply brigade (MSB) site (paraphrased into English, with abstractions in boldface):

If we have a **current site** which is unclassified
  and it's in the **Division support area**,
  and the **traffic** is **high enough**
  and the **traffic** is **balanced**
  and the site is persistent with no major **deployments** emanating from it
then it's probably an **MSB**

SMI used similar rules for the constraint-satisfaction component of its system, allowing information to propagate through the network in a manner similar to Waltz's (1975) well-known constraint-satisfaction algorithm for edge labeling.

The goal of the SRI group was to induce a knowledge base to characterize and identify types of site such as command posts and battle positions. Its approach was to induce a Bayesian classifier and use a generative model approach, producing a Bayesian network that could serve as a knowledge base. This modeling required transforming raw vehicle tracks into features (for example, the frequency of certain vehicles at sites, number of stops) that could be used to predict sites. Thus, it was also necessary to have hypothetical sites to test. SRI relied on SMI to provide hypothetical sites, and it also used some of the features that SMI computed. As a classifier, SRI used tree-augmented naive (TAN) Bayes (Friedman, Geiger, and Goldszmidt 1997).

Workarounds: The SAIC team integrated two approaches to workaround generation, one developed by USC-ISI, the other by GMU.

ISI developed course-of-action–generation problem solvers to create alternative solutions to workaround problems. In fact, two alternative course-of-action–generation problem solvers were developed. One is a novel planner whose knowledge base is represented in the ontologies, including its operators, state descriptions, and problem-specific information. It uses a novel partial-match capability developed in LOOM (MacGregor 1991). The other is based on a state-of-the-art planner (Veloso et al. 1995). Each solution lists several engineering actions for this workaround (for example, deslope the banks of the river, install a temporary bridge), includes information about the sources used (for example, what kind of earthmoving equipment or bridge is used), and asserts temporal constraints among the individual actions to indicate which can be executed in parallel. A temporal estimation-assessment problem solver evaluates each of the alternatives and selects one as the most likely choice for an enemy workaround. This problem solver was developed in EXPECT (Swartout and Gil 1995; Gil 1994).

Several general battlespace ontologies (for example, military units, vehicles), anchored on the HPKB upper ontology, were used and augmented with ontologies needed to reason about workarounds (for example, engineering equipment). Besides these ontologies, the knowledge bases used included a number of problem-solving methods to represent knowledge about how to solve the task. Both ontologies and problem-solving knowledge were used by two main problem solvers.

EXPECT's knowledge-acquisition tools were used throughout the evaluation to detect missing knowledge. EXPECT uses problem-solving knowledge and ontologies to analyze which information is needed to solve the task. This capability allows EXPECT to alert a user when there is missing knowledge about a problem (for example, unspecified bridge lengths) or a situation. It also helps debug and refine ontologies by detecting missing axioms and overgeneral definitions.

GMU developed the DISCIPLE98 system. DISCIPLE is an apprenticeship multistrategy learning system that learns from examples, from explanations, and by analogy and can be taught by an expert how to perform domain-specific tasks through examples and explanations in a way that resembles how experts teach apprentices (Tecuci 1998). For the workaround domain, DISCIPLE was extended into a baseline-integrated system that creates an ontology by acquiring concepts from a domain expert or importing them (through OKBC) from shared ontologies. It learns task-decomposition rules from a domain expert and uses this knowledge to solve workaround problems through hierarchical nonlinear planning.

We claim that HPKB technology facilitates rapid modification of knowledge-based systems. This claim was tested in both phases of the experiment because each phase allows time to improve performance on test problems.

First, with DISCIPLE's ontology-building tools, a domain expert assisted by a knowledge engineer built the object ontology from several sources, including expert's manuals, Alphatech's FIRE&ISE document and ISI's LOOM ontology. Second, a task taxonomy was defined by refining the task taxonomy provided by Alphatech. This taxonomy indicates principled decompositions of generic workaround tasks into subtasks but does not indicate the conditions under which such decompositions should be performed. Third, the examples of hierarchical workaround plans provided by Alphatech were used to teach DISCIPLE. Each such plan provided DISCIPLE with specific examples of decompositions of tasks into subtasks, and the expert guided DISCIPLE to "understand" why each task decomposition was appropriate in a particular situation. From these examples and the explanations of why they are appropriate in the given situations, DISCIPLE learned general task-decomposition rules. After a knowledge base consisting of an object ontology and task-decomposition rules was built, the hierarchical nonlinear planner of DISCIPLE was used to automatically generate workaround plans for new workaround problems.

Evaluation

The SAIC and Teknowledge integrated systems for crisis management, movement analysis, and workarounds were tested in an extensive study in June 1998. The study followed a two-phase, test-retest schedule. In the first phase, the systems were tested on problems similar to those used for system development, but in the second, the problems required significant modifications to the systems. Within each phase, the systems were tested and retested on the same problems. The test at the beginning of each phase established a baseline level of performance, but the test at the end measured improvement during the phase.

We claim that HPKB technology facilitates rapid modification of knowledge-based systems. This claim was tested in both phases of the experiment because each phase allows time to improve performance on test problems. Phase 2 provides a more stringent test: Only some of the phase 2 problems can be solved by the phase 1 systems, so the systems were expected to perform poorly in the test at the beginning of phase 2. The improvement in performance on these problems during phase 2 is a direct measure of how well HPKB technology facilitates knowledge capture, representation, merging, and modification.

Each challenge problem is evaluated by different metrics. The test items for crisis management were questions, and the test was similar to an exam. Overall competence is a function of the number of questions answered correctly, but the crisis-management systems are also expected to "show their work" and provide justifications (including sources) for their answers. Examples of questions, answers, and justifications for crisis management are shown in the section entitled Crisis-Management Challenge Problem.

Performance metrics for the movement-analysis problem are related to recall and precision. The basic problem is to identify sites, vehicles, and purposes given vehicle track data; so, performance is a function of how many of these entities are correctly identified and how many incorrect identifications are made. In general, identifications can be marked down on three dimensions: First, the identified entity can be more or less like the actual entity; second, the location of the identified entity can be displaced from the actual entity's true location; and third, the identification can be more or less timely.

The workaround problem involves generating workarounds to military actions such as bombing a bridge. Here, the criteria for successful performance include coverage (the generation of all workarounds generated), appropriateness (the generation of workarounds appropriate given the action), specificity (the exact implementation of the workaround), and accuracy of timing inferences (the length each step in the workaround takes to implement).

Performance evaluation, although essential, tells us relatively little about the HPKB integrated systems, still less about the component technologies. We also want to know why the systems perform well or poorly. Answering this question requires credit assignment because the systems comprise many technologies. We also want to gather evidence pertinent to several important, general claims. One claim is that HPKB facilitates rapid construction of knowledge-based systems because ontologies and knowledge bases can be reused. The challenge problems by design involve broad, relatively shallow knowledge in the case of crisis management and deep, fairly specific knowledge in the battlespace problems. It is unclear which kind of problem most favors the reuse claim and why. We are developing analytic models of reuse. Although the predictions of these models will not be directly tested in the first year's evaluation, we will gather data to calibrate these models for a later evaluation.

Results of the Challenge Problem Evaluation

We present the results of the crisis-management evaluation first, followed by the results of the battle-space evaluation.

Crisis Management

The evaluation of the SAIC and Teknowledge crisis-management systems involved 7 trials or batches of roughly 110 questions. Thus, more than 1500 answers were manually graded by the challenge problem developer, IET, and subject matter experts at PSR on criteria ranging from correctness to completeness of source material to the quality of the representation of the question. Each question in a batch was posed in English accompanied by the syntax of the corresponding parameterized question (figure 3). The crisis-management systems were supposed to translate these questions into an internal representation, MELD for the Teknowledge system and KIF for the SAIC system. The MELD translator was operational for all the trials; the KIF translator was used to a limited extent on later trials.

The first trial involved testing the systems on the sample questions that had been available for several months for training. The remaining trials implemented the "test and retest with scenario modification" strategy discussed earlier. The first batch of test questions, TQA, was repeated four days later as a retest; it was designated TQA' for scoring purposes. The difference in scores between TQA and TQA' represents improvements in the systems. After solving the questions in TQA', the systems tackled a new set, TQB, designed to be "close to" TQA. The purpose of TQB was to check whether the improvements to the systems generalized to new questions. After a short break, a modification was introduced into the crisis-management scenario, and new fragments of knowledge about the scenario were released. Then, the cycle repeated: A new batch of questions, TQC, tested how well the systems coped with the scenario modification; then after four days, the systems were retested on the same questions, TQC', and on the same day, a final batch, TQD, was released and answered.

Each question in a trial was scored according to several criteria, some official and others optional. The four official criteria were (1) the correctness of the answer, (2) the quality of the explanation of the answer, (3) the completeness and quality of cited sources, and (4) the quality of the representation of the question. The optional criteria included lay intelligibility of explanations, novelty of assumptions, quality of the presentation of the explanation, the automatic production by the system of a representation of the question, source novelty, and reconciliation of multiple sources. Each question could garner a score between 0 and 3 on each criterion, and the criteria were themselves weighted. Some questions had multiple parts, and the number of parts was a further weighting criterion. In retrospect, it might have been clearer to assign each question a percentage of the points available, thus standardizing all scores, but in the data that follow, scores are on an open-ended scale. Subject-matter experts were assisted with scoring the quality of knowledge representations when necessary.

A web-based form was developed for scoring, with clear instructions on how to assign scores. For example, on the correct-answer criterion, the subject-matter expert was instructed to award "zero points if no top-level answer is provided and you cannot infer an intended answer; one point for a wrong answer without any convincing arguments, or most required answer elements; two points for a partially correct answer; three points for a correct answer addressing most required elements."

When you consider the difficulty of the task, both systems performed remarkably well. Scores on the sample questions were relatively high, which is not surprising because these questions had been available for training for several months (figure 5). It is also not surprising that scores on the first batch of test questions (TQA) were not high. It is gratifying, however, to see how scores improve steadily between test and retest (TQA and TQA', TQC and TQC') and that these gains are general: The scores on TQA' and TQB and TQC' and TQD were similar.

The scores designated *auto* in figure 5 refer to questions that were translated automatically from English into a formal representation. The Teknowledge system translated all questions automatically, the SAIC system very few. Initially, the Teknowledge team did not manipulate the resulting representations, but in later batches, they permitted themselves minor modifications. The effects of these can be seen in the differences between TQB and TQB-Auto, TQC and TQC-Auto, and TQD and TQD-Auto.

Although the scores of the Teknowledge and SAIC systems appear close in figure 5, differences between the systems appear in other views of the data. Figure 6 shows the performance of the systems on all official questions plus a few optional questions. Although these extra questions widen the gap between the systems, the real effect comes from adding optional components to the scores. Here,

When you consider the difficulty of the task, both systems performed remarkably well. Scores on the sample questions were relatively high, which is not surprising because these questions had been available for training for several months ....
It is also not surprising that scores on the first batch of test questions (TQA) were not high. It is gratifying, however, to see how scores improve steadily between test and retest…

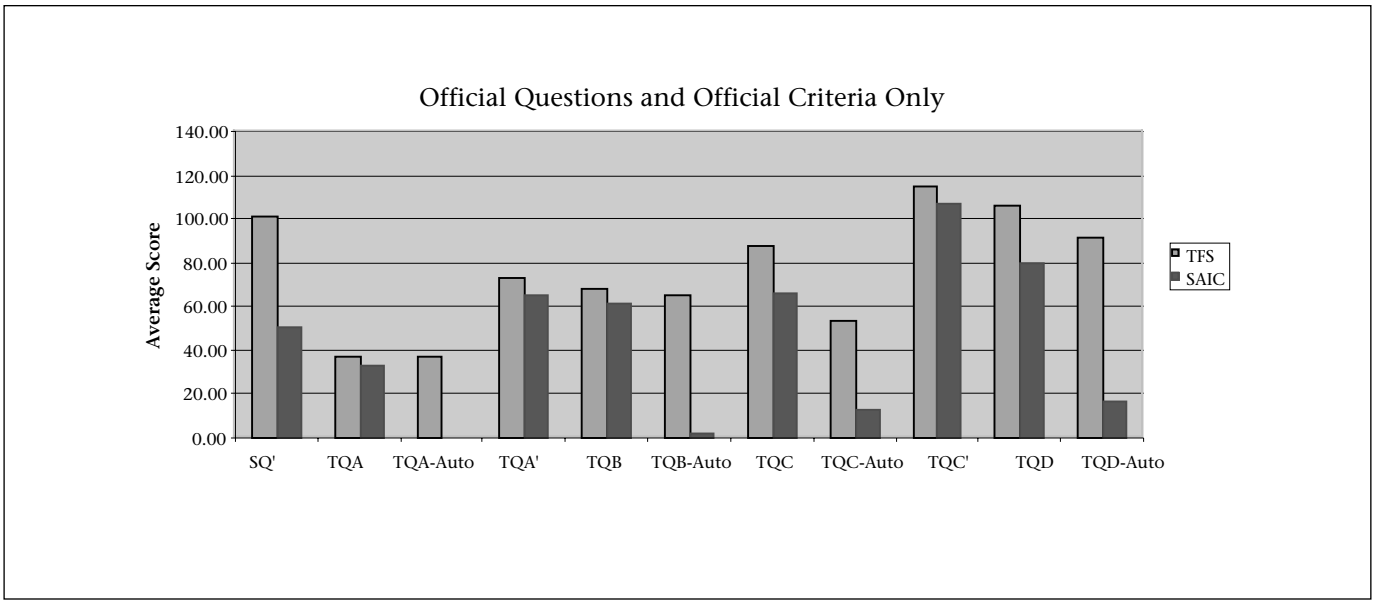

Figure 5. Scores on the Seven Batches of Questions That Made Up the Crisis-Management Challenge Problem Evaluation.
Some questions were automatically translated into a formal representation. The scores for these are denoted with the suffix *Auto*.

Teknowledge got credit for its automatic translation of questions and also for the lay intelligibility of its explanations, the novelty of its assumptions, and other criteria.

Recall that each question is given a score between zero and three on each of four official criteria. Figure 7 represents the distribution of the scores 0, 1, 2, and 3 for the correctness criterion over all questions for each batch of questions and each system. A score of 0 generally means the question was not answered. The Teknowledge system answered more questions than the SAIC system, and it had a larger proportion of perfect scores.

Interestingly, if one looks at the number of perfect scores as a fraction of the number of questions answered, the systems are not so different. One might argue that this comparison is invalid, that one's grade on a test depends on all the test items, not just those one answers. However, suppose one is comparing students who have very different backgrounds and preparation; then, it can be informative to compare them on both the number and quality of their answers. The Teknowledge-SAIC comparison is analogous to a comparison of students with very different backgrounds. Teknowledge used CYC, a huge, mature knowledge base, whereas SAIC used CYC's upper-level ontology and a variety of disparate knowledge bases. The systems were not at the same level of preparation at the time of the experiment, so it is informative to ask how each system performed on the questions it answered. On the sample questions, Teknowledge got perfect scores on nearly 80 percent of the questions it answered, and SAIC got perfect scores on nearly 70 percent of the questions it answered. However, on TQA, TQA', and TQB, the systems are very similar, getting perfect scores on roughly 60 percent, 70 percent, and 60 percent of the questions they answered. The gap widens to a roughly 10-percent spread in Teknowledge's favor on TQC, TQC', and TQD. Similarly, when one averages the scores of answered questions in each trial, the averages for the Teknowledge and SAIC systems are similar on all but the sample-question trial. Even so, the Teknowledge system had both higher coverage (questions answered) and higher correctness on all trials.

If one looks at the minimum of the official component scores for each question—answer correctness, explanation adequacy, source adequacy, and representation adequacy—the difference between the SAIC and Teknowledge systems is quite pronounced. Calculating the minimum component score is like finding something to complain about in each answer—the answer or the explanation, the sources or the question representation. It is the statistic of most interest to a potential user who would dismiss a system for failure on any of these criteria. Figure 8 shows that neither system did very well on this stringent criterion, but the Teknowledge system got a perfect or good minimum score (3 or 2) roughly twice as often as the SAIC system in each trial.

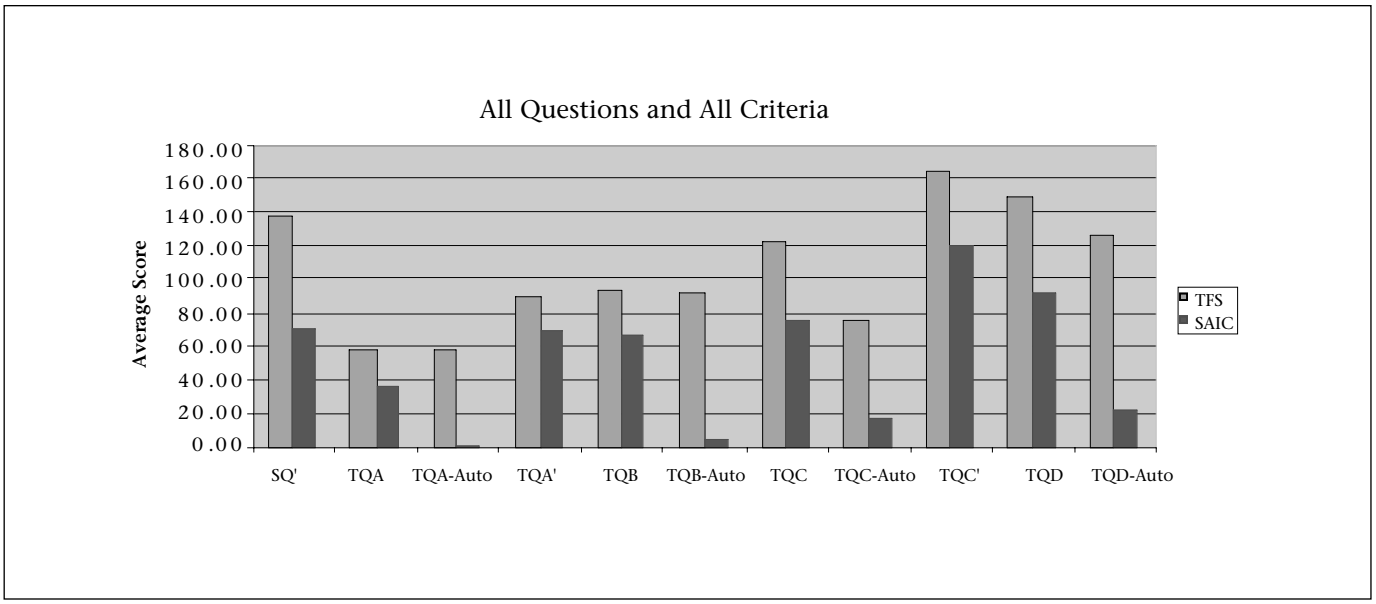

Figure 6. Scores on All Questions (Including Optional Ones) and Both Official and Optional Criteria.

Movement Analysis

The task for movement analysis was to identify sites—command posts, artillery sites, battle positions, and so on—given data about vehicle movements and some intelligence sources. Another component of the problem was to identify convoys. The data were generated by simulation for an area of approximately 10,000 square kilometers over a period of approximately 4 simulated days. The scenario involved 194 military units arranged into 5 brigades, 1848 military vehicles, and 7666 civilian vehicles. Some units were collocated at sites; others stood alone. There were 726 convoys and nearly 1.8 million distinct reports of vehicle movement.

Four groups developed systems for all or part of the movement-analysis problem, and SAIC and Teknowledge provided integration support. The groups were Stanford's SMI, SRI, UMass, and MIT. Each site identification was scored for its accuracy, and recall and precision scores were maintained for each site. Suppose an identification asserts at time t that a battalion command post exists at a location (x, y). To score this identification, we find all sites within a fixed radius of (x, y). Some site types are very similar to others. For example, all command posts have similar characteristics; so, if one mistakes a battalion command post for, say, a division command post, then one should get partial credit for the identification. The identification is incorrect but not as incorrect as, say, mistaking a command post for an artillery site. Entity error ranges from 0 for completely correct identifications to 1 for hopelessly wrong identifications. Fractional entity errors provide partial credit for near-miss identifications. If one of the sites within the radius of an identification matches the identification (for example, a battalion command post), then the identification score is 0, but if none matches, then the score is the average entity error for the identification matched against all the sites in the radius. If no site exists within the radius of an identification, then the identification is a *false positive*.

Recall and precision rates for sites are defined in terms of entity error. Let H be the number of sites identified with zero entity error, M be the number of sites identified with entity error less than one (near misses), and R be the number of sites identified with maximum entity error. Let T be the total number of sites, N be the total number of identifications, and F be the number of false positives. The following statistics describe the performance of the movement-analysis systems:

zero-entity error recall = H / T
non-one-entity error recall = $(H + M) / T$
maximum-entity error recall
$= (H + M + R) / T$
zero-entity error precision = H / N
non–one-entity error precision
$= (H + M) / N$
maximum-entity error precision
$= (H + M + R) / N$
false positive rate = F / N

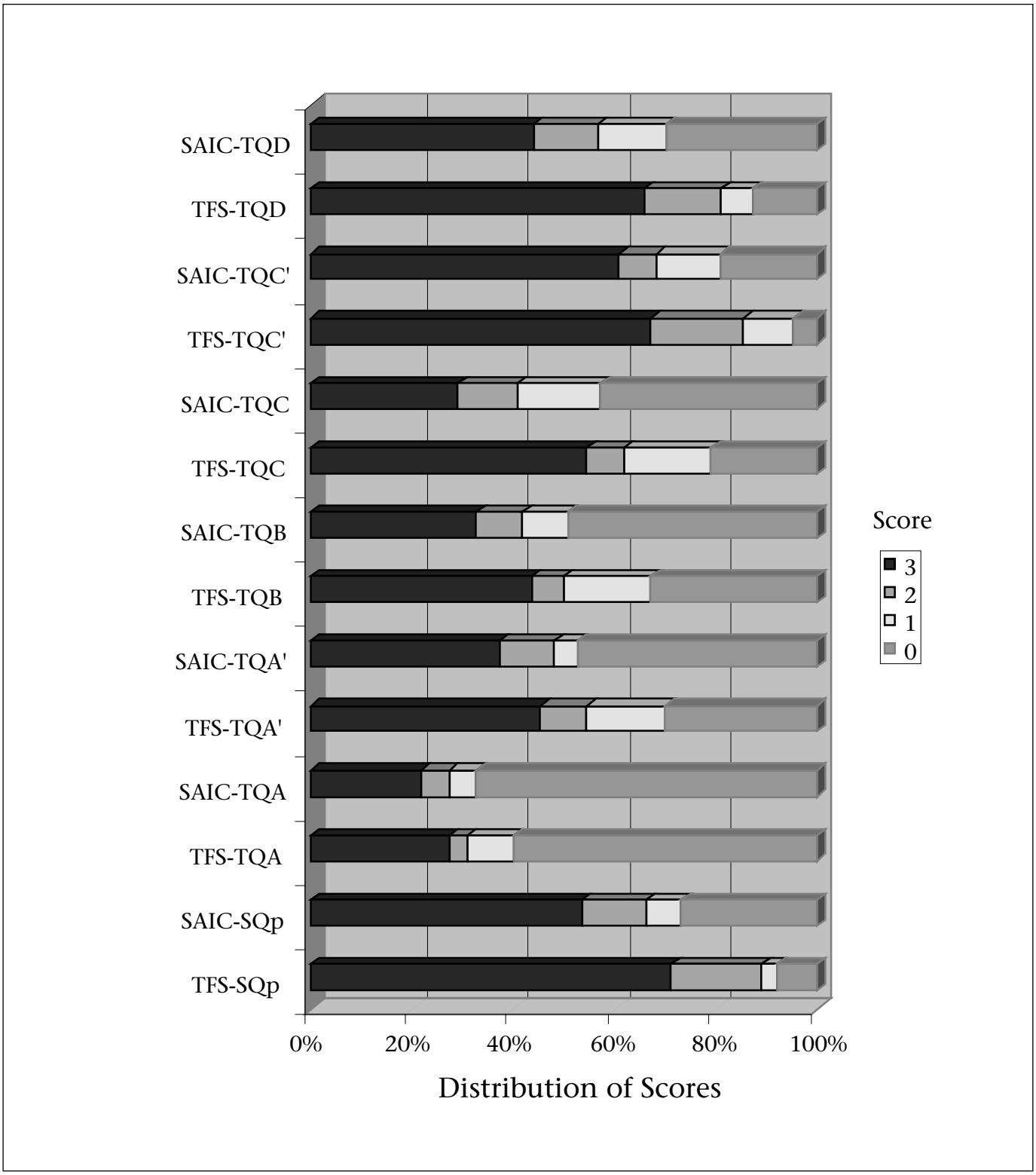

Figure 7. The Distribution of Correctness Scores for the Crisis-Management Challenge Problem.

Correctness is one of the official scoring criteria and takes values in {0, 1, 2, 3}, with 3 being the highest score. This graph shows the proportion of the correctness scores that are 0, 1, 2, and 3. The lower bar, for example, shows that roughly 70 percent of the correctness scores were 3, and roughly 10 percent were 1 or 0.

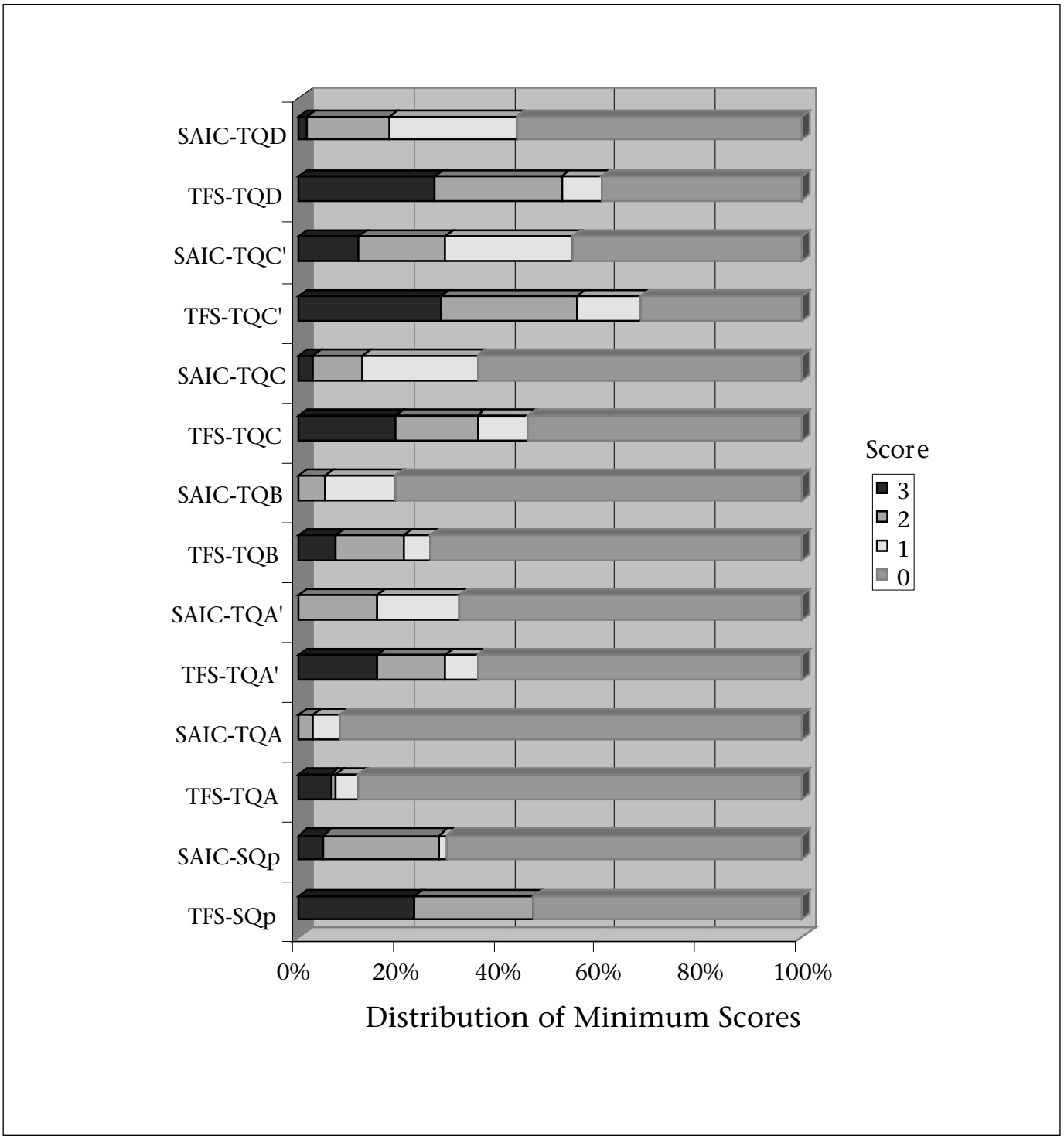

Figure 8. The Distribution of Minimum Scores for the Crisis-Management Challenge Problem.

Each question is awarded four component scores corresponding to the four official criteria. This graph shows the distribution of the minimum of these four components over questions. With the lowest bar, for example, on 20 percent of the questions, the *minimum* of the 4 component scores was 3.

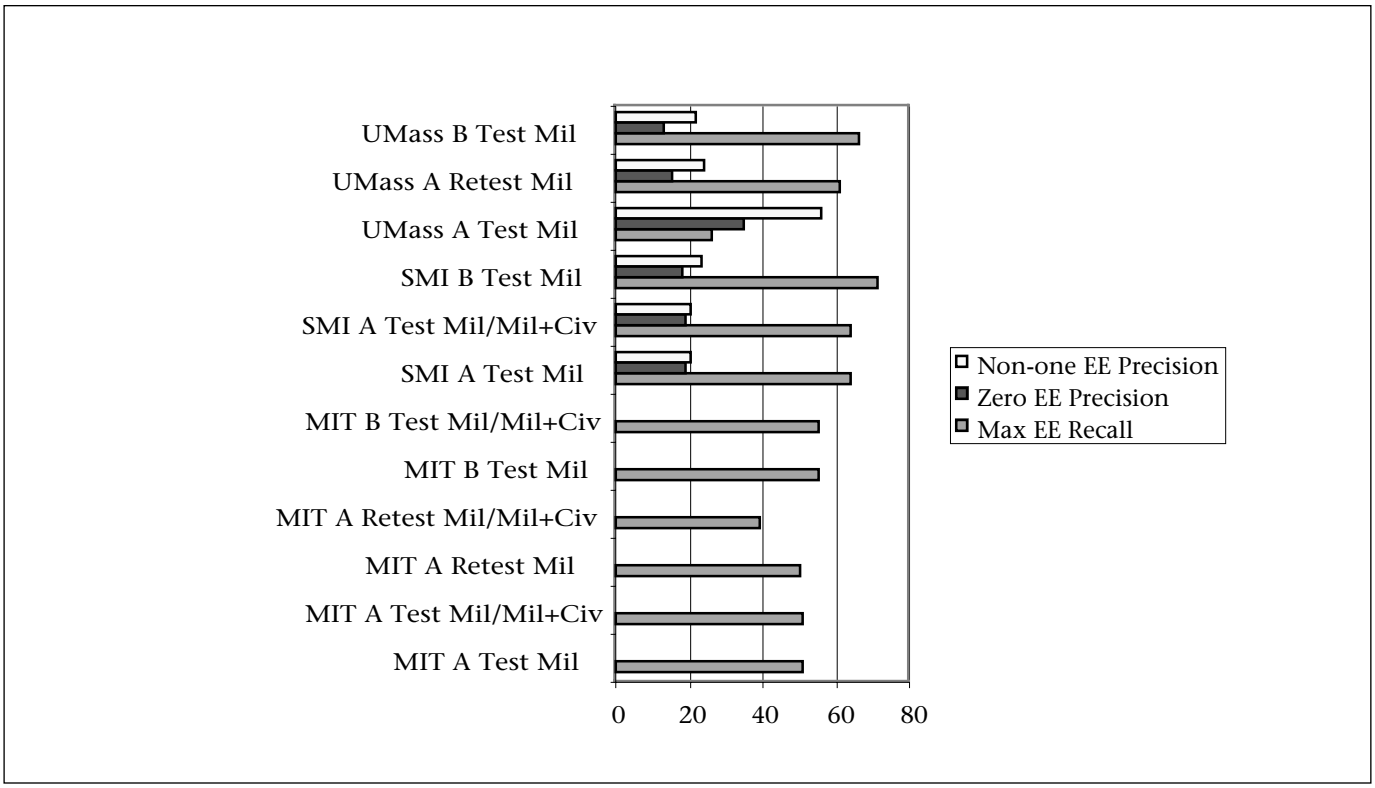

Figure 9. Scores for Detecting and Identifying Sites in the Movement-Analysis Problem.

Trials labeled *A* are from the first phase of the experiment, before the scenario modification. Trials labeled *B* are from after the scenario modification. *Mil* and *Mil + Civ* refer to data sets with military traffic only and with both military and civilian traffic, respectively.

Recall and precision scores for the groups are shown in figure 9. The experiment design involved two phases with a test and retest within each phase. Additionally, the data sets for each test included either military traffic only or military plus civilian traffic. Had each group run their systems in each experimental condition, there would have been 4 tests, 2 data sets with 2 versions of each (military only and military plus civilian), and 4 groups, or 64 conditions to score. All these conditions were not run. Delays were introduced by the process of reformatting data to make it compatible with a scoring program, so scores were not released in time for groups to modify their systems; one group devoted much time to scoring and did not participate in all trials, and another group participated in no trials for reasons discussed at the end of this section.

The trials that were run are reported in figure 9. Recall rates range from 40 percent to just over 60 percent, but these are for maximum-entity error recall—the frequency of detecting a site when one exists but not identifying it correctly. Rates for correct and near-miss identification are lower, ranging from 10 percent to 15 percent and are not shown in figure 9. Of the participating research groups, MIT did not attempt to identify the types of site, only whether sites are present; so, their entity error is always maximum. The other groups did not attempt to identify all kinds of site; for example, UMass tried to identify only battle positions, command posts, and assembly-staging areas. Even so, each group was scored against all site types. Obviously, the scores would be somewhat higher had the groups been scored against only the kinds of site they intended to identify (for example, recall rates for UMass ranged from 20 percent to 39 percent for the sites they tried to identify).

Precision rates are reported only for the SMI and UMass teams because MIT did not try to identify the types of site. UMass's precision was highest on its first trial; a small modification to its system boosted recall but at a significant cost to precision. SMI's precision hovered around 20 percent on all trials.

Scores were much higher for convoy detection. Although scoring convoy identifications posed some interesting technical challenges, the results were plain enough: SMI, UMass,

and MIT detected 507 (70 percent), 616 (85 percent), and 465 (64 percent) of the 726 convoys, respectively. The differences between these figures do not indicate that one technology was superior to another because each group emphasized a slightly different aspect of the problem. For example, SMI didn't try to detect small convoys (fewer than four vehicles). In any case, the scores for convoy identifications are quite similar.

In sum, none of the systems identified site types accurately, although they identified all detected sites and convoys pretty well. The reason for these results is that sites can be detected by observing vehicle halts, just as convoys can be detected by observing clusters of vehicles. However, it is difficult to identify site types without more information. It would be worth studying the types of information that humans require for the task.

The experience of SRI is instructive: The SRI system used features of movement data to infer site types; unfortunately, the data did not seem to support the task. There were insufficient quantities of training data to train classifiers (fewer than 40 instances of sites), but more importantly, the data did not have sufficient underlying structure—it was too random. The SRI group tried a variety of classifier technologies, including identification trees, nearest neighbor, and C4.5, but classification accuracy remained in the low 30-percent range. This figure is comparable to those released by UMass on the performance of its system on three kinds of site. Even when the UMass recall figures are not diluted by sites they weren't looking for, they did not exceed 40 percent. SRI and UMass both performed extensive exploratory data analysis, looking for statistically significant relationships between features of movement data and site types, with little success, and the disclosed regularities were not strongly predictive. The reason for these teams found little statistical regularity is probably that the data were artificial. It is extraordinarily difficult to build simulators that generate data with the rich dependency structure of natural data.

Some simulated intelligence and radar information were released with the movement data. Although none of the teams reported finding it useful, we believe these kind of data probably help human analysts. One assumes humans perform the task better than the systems reported here, but because no human ever solved these movement-analysis problems, one cannot tell whether the systems performed comparatively well or poorly. In any case, the systems took a valuable first step toward movement analysis, detecting most convoys and sites. Identifying the sites accurately will be a challenge for the future.

Workarounds

The workaround problem was evaluated in much the same way as the other problems: The evaluation period was divided into two phases, and within each phase, the systems were given a test and a retest on the same problems. In the first phase, the systems were tested on 20 problems and retested after a week on the same problems; in the second phase, a modification was introduced into the scenario, the systems were tested on five problems and after a week retested on the same five problems and five new ones.

Solutions were scored along five equally weighted dimensions: (1) viability of enumerated workaround options, (2) correctness of the overall time estimate for a workaround, (3) correctness of solution steps provided for each viable option, (4) correctness of temporal constraints among these steps, and (5) appropriateness of engineering resources used. Scores were assigned by comparing the systems' answers with those of human experts. Bonus points were awarded when, occasionally, systems gave better answers than the experts. These answers became the gold standard for scoring answers when the systems were retested.

Four systems were fielded by ISI, GMU, Teknowledge, and AIAI. The results are shown in figures 10 and 11. In each figure, the upper extreme of the vertical axis represents the maximum score a system could get by answering all the questions correctly (that is, 200 points for the initial phase, 50 points for the first test in the modification phase, 100 points for the retest in the modification phase). The bars represent the number of points scored, and the circles represent the number of points that could have been awarded given the number of questions that were actually answered. For example, in the initial phase, ISI answered all questions, so it could have been awarded 200 points on the test and 200 on the retest; GMU covered only a portion of the domain, and it could have been awarded a maximum of 90 and 100 points, respectively. The bars represent the number of points actually scored by each system.

How one views the performance of these systems depends on how one values correctness; coverage (the number of questions answered); and, more subtly, the prospects for scaling the systems to larger problem sets. An assistant should answer any question posed to it, but if the system is less than ideal, should it

What we might call errors of specificity, in which an answer is less specific or complete than it should be, are not inconsistent with the philosophy of HPKB, which expects systems to give partial—even common-sense—answers when they lack specific knowledge.

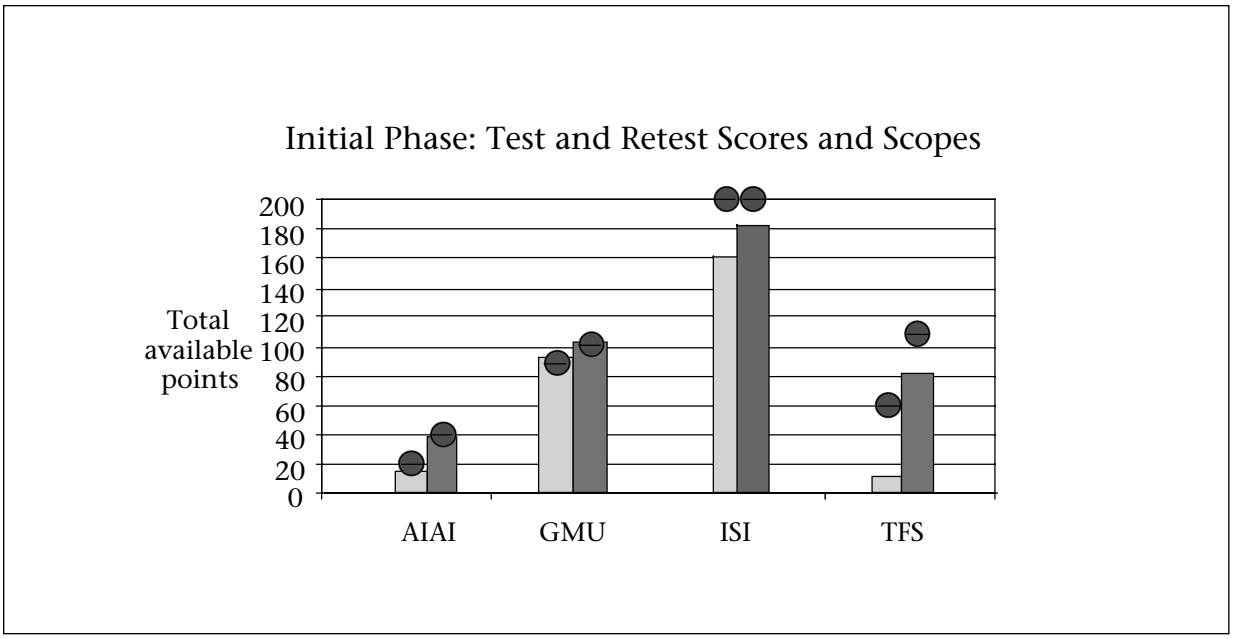

Figure 10. Results of the Initial Phase of the Workaround Evaluation.

Lighter bars represent test scores; darker bars represent retest scores. Circles represent the scope of the task attempted by each system, that is, the best score that the system could have received given the number of questions it answered.

answer more questions with some errors or fewer questions with fewer errors? Obviously, the answer depends on the severity of errors, the application, and the prospect for improving system coverage and accuracy. What we might call errors of specificity, in which an answer is less specific or complete than it should be, are not inconsistent with the philosophy of HPKB, which expects systems to give partial—even commonsense—answers when they lack specific knowledge.

Figures 10 and 11 show that USC-ISI was the only group to attempt to solve all the workaround problems, although its answers were not all correct, whereas GMU solved fewer problems with higher overall correctness. AIAI solved fewer problems still, but quite correctly, and Teknowledge solved more problems than AIAI with more variable correctness. One can compute coverage and correctness scores as follows: *Coverage* is the number of questions attempted divided by the total number of questions in the experiment (55 in this case). *Correctness* is the total number of points awarded divided by the number of points that might have been awarded given the number of answers attempted. Figure 12 shows a plot of coverage against correctness for all the workaround systems. Points above and to the right of other points are superior; thus, the ISI and GMU systems are preferred to the other systems, but the ranking of these systems depends on how one values coverage and correctness.

Two factors complicate comparisons between the systems. First, if one is allowed to select a subset of problems for one's system to solve, one might be expected to select problems on which one hopes the system will do well. Thus, ISI's correctness scores are not strictly comparable to those of the other systems because ISI did not select problems but solved them all. Said differently, what we call correctness is really "correctness in one's declared scope." It is not difficult to get a perfect correctness score if one selects just one problem; conversely, it is not easy to get a high correctness score if one solves all problems.

Second, Figure 12 shows that coverage can be increased by knowledge engineering, but at what cost to correctness? GMU believes correctness need not suffer with increased coverage and cites the rapid growth of its knowledge base. At the beginning of the evaluation period, the coverage of the knowledge base was about 40 percent (11841 binary predicates), and two weeks later, the coverage had grown significantly to roughly 80 percent of the domain (20324 binary predicates). In terms of coverage and correctness (figure 12), GMU's coverage increased from 47.5 percent in the test phase to 80 percent in the modification phase, with almost no decrement in correctness. However, ISI suggests that the complexity of the workaround task increases significantly as the coverage is extended, to the point where a perfect score might be out of reach given the

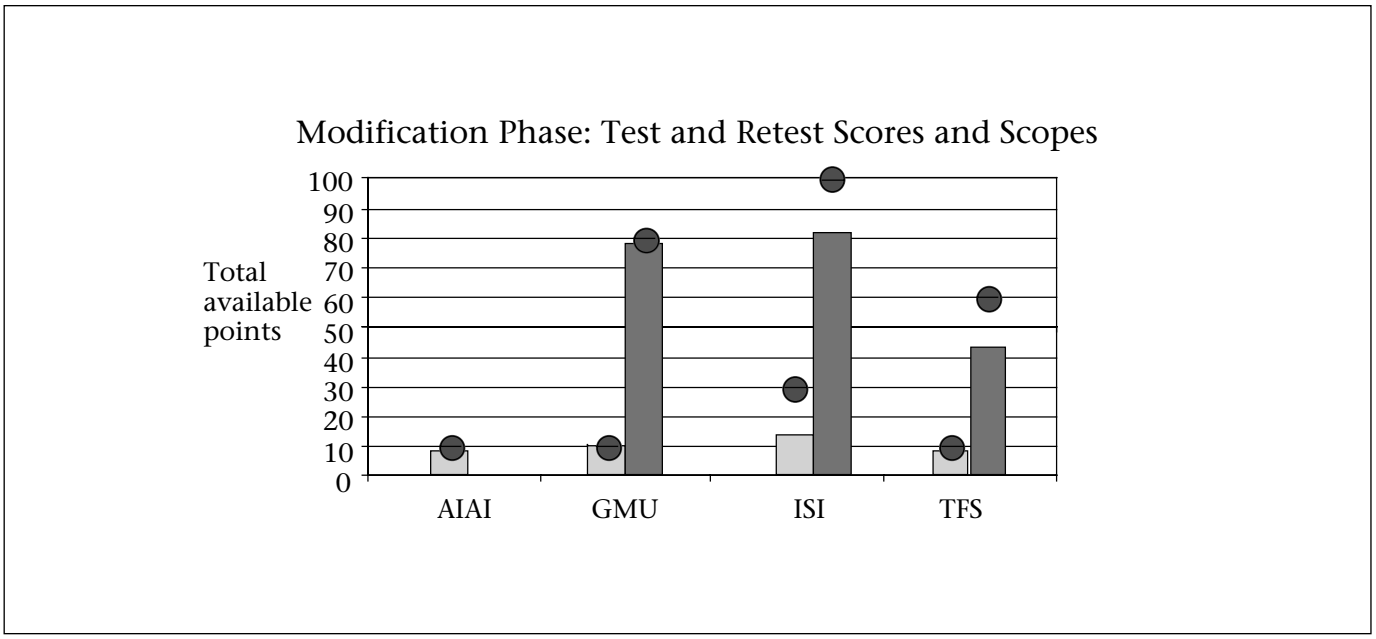

Figure 11. Results of the Modification Phase of the Workaround Evaluation.

Lighter bars represent test scores; darker bars represent retest scores. Circles represent the scope of the task attempted by each system, that is, the best score that the system could have received given the number of questions it answered. Note that only 5 problems were released for the test, 10 for the retest; so, the maximum available points for each are 50 and 100, respectively.

state of the art in AI technology. For example, in some cases, the selection of resources requires combining plan generation with reasoning about temporal aspects of the (partially generated) plan. Moreover, interdependencies in the knowledge bases grow nonlinearly with coverage because different aspects of the domain interact and must be coordinated. For example, including mine-clearing operations changes the way one looks at how to ford a river. Such interactions in large knowledge bases might degrade the performance of a system when changes are made during a retest and modification phase.

The coverage versus correctness debate should not cloud the accomplishments of the workaround systems. Both ISI and GMU were judged to perform the workaround task at an expert level. All the systems were developed quite rapidly—indeed, GMU's knowledge base doubled during the experimental period itself—and knowledge reuse was prevalent. These results take us some way toward the HPKB goal of very rapid development of powerful and flexible knowledge base systems.

Evaluation Lessons Learned

The experiments reported here constitute one of the larger studies of AI technology. In addition to the results of these experiments, much was learned about how to conduct such studies. The primary lesson is that one should release the challenge problem specifications early. Many difficulties can be traced to delays in the release of the battlespace specifications. Experiment procedures that involve multiple sites and technologies must be debugged in dry-run experiments before the evaluation period begins. There must be agreement on the formats of input data, answers, and answer keys—a mundane requirement but one that caused delays in movement-analysis evaluation.

A major contributor to the success of the crisis-management problem was the parameterized question syntax, which gave all participants a concise representation of the kinds of problem they would solve during the evaluation. Similarly, evaluation metrics for crisis management were published relatively early. Initially, there was a strong desire for evaluation criteria to be objective enough for a machine to score performance. Relaxing this requirement had a positive effect in crisis management. The criteria were objective in the sense that experts followed scoring guidelines, but we could ask the experts to assess the quality of a system's explanation—something no program can do. However, scoring roughly 1500 answers in crisis management took its toll on the experts and IET. It is worth considering whether equally informative results could be had at a lower cost.

The HPKB evaluations were set up as friend-

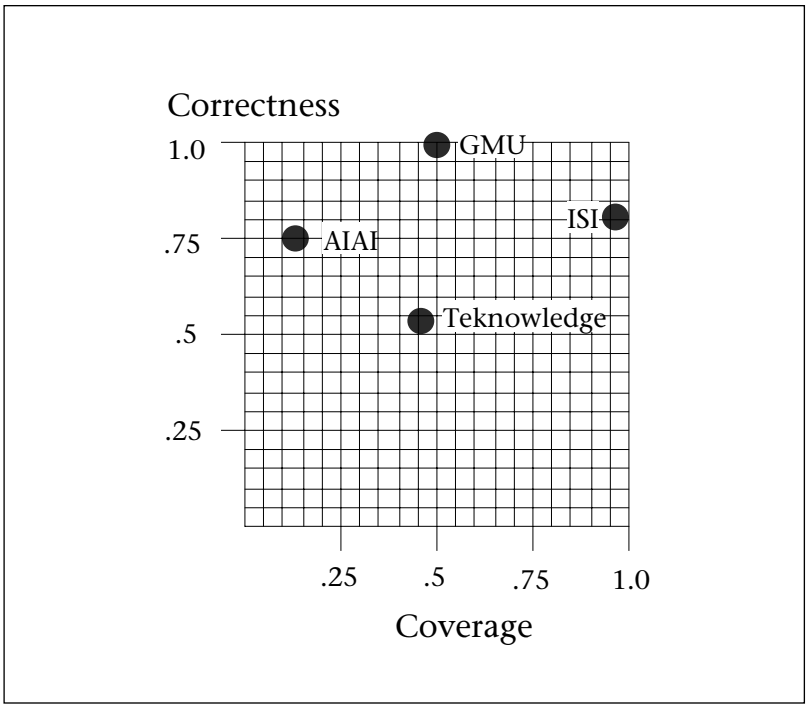

Figure 12. A Plot of Overall Coverage against Overall Correctness for Each System on the Workaround Challenge Problem.

ly competitions between the integration teams, SAIC and Teknowledge, each incorporating a subset of the technology in a unique architecture. However, some technology was used by both teams, and some was not tested in integrated systems but rather as stand-alone components. The purpose of the friendly competition—to evaluate merits of different technologies and architectures—was not fully realized. Given the cost of developing HPKB systems, we might reevaluate the purported and actual benefits of competitions.

Conclusion

HPKB is an ambitious program, a high-risk, high-payoff venture beyond the edge of knowledge-based technology. In its first year, HPKB suffered some setbacks, but these were informative and will help the program in future. More importantly, HPKB celebrated some extraordinary successes. It has long been a dream in AI to ask a program a question in natural language about almost anything and receive a comprehensive, intelligible, well-informed answer. Although we have not achieved this aim, we are getting close. In HPKB, questions were posed in natural language, questions on a hitherto impossible range of topics were answered, and the answers were supported by page after page of sources and justifications. Knowledge bases were constructed rapidly from many sources, reuse of knowledge was a significant factor, and semantic integration across knowledge bases started to become feasible. HPKB has integrated many technologies, cutting through traditional groupings in AI, to develop new ways of acquiring, organizing, sharing, and reasoning with knowledge, and it has nourished a remarkable assault on perhaps the grandest of the AI grand challenges—to build intelligent programs that answer questions about everyday things, important things like international relations but also ordinary things like river banks.

Acknowledgments

The authors wish to thank Stuart Aitken, Vinay Chaudhri, Cleo Condoravdi, Jon Doyle, John Gennari, Yolanda Gil, Moises Goldszmidt, William Grosso, Mark Musen, Bill Swartout, and Gheorghe Tecuci for their help preparing this article.

Notes

1. Many HPKB resources and documents are found at www.teknowledge.com/HPKB. A web site devoted to the Year 1 HPKB Challenge Problem Conference can be found at www.teknowledge.com/HPKB/meetings/Year1meeting/. Definitive information on the year 1 crisis-management challenge problem, including updated specification, evaluation procedures, and evaluation results, is available at www.iet.com/Projects/HPKB/Y1Eval/. Earlier, draft information about both challenge problems is available at www.iet.com/Projects/ HPKB/Combined-CPs.doc. Scoring procedures and data for the movement-analysis problem are available at eksl-www.cs.umass.edu:80/research/hpkb/scores/. For additional sources, contact the authors.

2. These numbers and all information regarding MTI radar are approximate; actual figures are classified.

References

Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian Network Classifiers. *Machine Learning* 29:131–163.

Gil, Y. 1994. Knowledge Refinement in a Reflective Architecture. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 520–526. Menlo Park, Calif.: American Association for Artificial Intelligence.

Jones, E. 1998. Formalized Intelligence Report Ensemble (FIRE) & Intelligence Stream Encoding (ISE)", memo, Alphatech.

Katz, B. 1997. From Sentence Processing to Information Access on the World Wide Web. Paper presented at the AAAI Spring Symposium on Natural Language

Processing for the World Wide Web, 24–26 March, Stanford, California.

Lee, J.; Grunninger, M.; Jin, Y.; Malone, T.; Tate, A.; Yost, G.; and the PIF Working Group, eds. 1996. The PIF Process Interchange and Framework Version 1.1, Working paper, 194, Center for Coordination Science, Massachusetts Institute of Technology.

Lenat, D. 1995. *Artificial Intelligence*. *Scientific American* 273(3): 80–82.

Lenat, D. B., and Feigenbaum, E. A. 1987. On the Thresholds of Knowledge. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 1173–1182. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Lenat, D., and Guha, R. 1990. *Building Large Knowledge-Based Systems*. Reading, Mass.: Addison Wesley.

Liddy, E. D.; Paik, W.; and McKenna, M. 1995. Development and Implementation of a Discourse Model for Newspaper Texts. Paper presented at the AAAI Symposium on Empirical Methods in Discourse Interpretation and Generation, 27–29 March, Stanford, California.

MacGregor, R. 1991. The Evolving Technology of Classification-Based Knowledge Representation Systems. In *Principles of Semantic Networks: Explorations in the Representation of Knowledge,* ed. J. Sowa, 385–400. San Francisco, Calif.: Morgan Kaufmann.

Miller, G.; Beckwith, R.; Fellbaum, C.; Gross, D.; and Miller, K. 1993. Introduction to WORDNET: An Online Lexical Database. Available at www.cogsci.princeton.edu/~wn/papers/.

Park, J.; Gennari, J.; and Musen, M. 1997. Mappings for Reuse in Knowledge-Based Systems, Technical Report, SMI-97-0697, Section on Medical Informatics, Stanford University.

Pease, A., and Albericci, D. 1998. The Warplan. CJL256, Teknowledge, Palo Alto, California.

Pease, A., and Carrico, T. 1997a. The JTF ATD Core Plan Representation: A Progress Report. Paper presented at the AAAI 1997 Spring Symposium on Ontological Engineering, 24–26 March, Stanford, California.

Pease, A., and Carrico, T. 1997b. The JTF ATD Core Plan Representation: Request for Comment, AL/HR-TP-96-9631, Armstrong Lab, Wright-Patterson Air Force Base, Ohio.

Swartout, B., and Gil, Y. 1995. EXPECT: Explicit Representations for Flexible Acquisition. Paper presented at the Ninth Knowledge Acquisition for Knowledge-Based Systems Workshop, 26 February–3 March, Banff, Canada.

Tate, A. 1998. Roots of SPAR—Shared Planning and Activity Representation. *Knowledge Engineering Review* (Special Issue on Ontologies) 13(1): 121–128.

Tecuci G. 1998. *Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool, and Case Studies*. San Diego, Calif.: Academic.

Veloso, M.; Carbonell J.; Perez, A.; Borrajo, D.; Fink, E.; and Blythe, J. 1995. Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Experimental and Theoretical* 7:81–120.

Waltz, D. 1975. Understanding Line Drawings of Scenes with Shadows. In *The Psychology of Computer Vision,* ed. P. H. Winston, 19–91. New York: McGraw-Hill.

Wiederhold, G., ed. 1996. *Intelligent Integration of Information*. Boston, Mass.: Kluwer Academic.

Paul Cohen is professor of computer science at the University of Massachusetts. He received his Ph.D. from Stanford University in 1983. His publications include *Empirical Methods for Artificial Intelligence* (MIT Press, 1995) and *The Handbook of Artificial Intelligence, Volumes 3 and 4* (Addison-Wesley), which he edited with Edward A. Feigenbaum and Avron B. Barr. Cohen is developing theories of how sensorimotor agents such as robots and infants learn ontologies.

Robert Schrag is a principal scientist at Information Extraction & Transport, Inc., in Rosslyn, Virginia. Schrag has served as principal investigator to develop the high-performance knowledge base crisis-management challenge problem. Besides evaluating emerging AI technologies, his research interests include constraint satisfaction, propositional reasoning, and temporal reasoning.

Eric Jones is a senior member of the technical staff at Alphatech Inc. From 1992 to 1996, he was a member of the academic staff in the Department of Computer Science at Victoria University, Wellington, New Zealand. He received his Ph.D. in AI from Yale University in 1992. He also has degrees in mathematics and geography from the University of Otago, New Zealand. His research interests include case-based reasoning, natural language processing, and statistical approaches to machine learning. He serves on the American Meteorological Society Committee on AI Applications to Environmental Science.

Adam Pease received his B.S. and M.S. in computer science from Worcester Polytechnic Institute. He has worked in the area of knowledge-based systems for the past 10 years. His current interests are in planning ontologies and large knowledge-based systems. He leads an integration team for Teknowledge on the High-Performance Knowledge Base Project.

Albert D. Lin received a B.S.E.E. from Feng Chia University in Taiwan in 1984 and an M.S. in electrical and computer engineering from San Diego State University in 1990. He is a technical lead at Science Applications International Corporation, currently leading the High-Performance Knowledge Base Project.

Barbara H. Starr received a B.S. from the University of Witwatersrand in South Africa in 1980 and an Honors degree in computer science at the University of Witwatersrand in 1981. She has been a consultant in the areas of AI and object-oriented software development in the United States for the past 13 years and in South Africa for several years before that. She is currently leading the crisis-management integration effort for the High-Performance Knowledge Base Project at Science Applications International Corporation.

David Gunning has been the program manager for the High-Performance Knowledge Base Project since its inception in 1995 and just became the assistant director for Command and Control in the Defense Advanced Research Projects Agency's research projects in Command and Control, which include work in intelligent reasoning and decision aids for crisis management, air campaign planning, course-of-action analysis, and logistics planning. Gunning has an M.S. in computer science from Stanford University and an M.S. in cognitive psychology from the University of Dayton.

Murray Burke has served as the High-Performance Knowledge Base Project deputy program manager for the past year, and the Defense Advanced Research Projects Agency has selected him as its new program manager. He has over 30 years in command and control information systems research and development. He was cofounder and executive vice president of Knowledge Systems Concepts, Inc., a software company specializing in defense applications of advanced information technologies. Burke is a member of the American Association for Artificial Intelligence, the Association of Computing Machinery, and the Institute of Electrical and Electronics Engineers.

The Third International Conference on Autonomous Agents (Agents '99)

■ **May 1 - May 5, 1999**
■ **Seattle, Washington—Hyatt Regency Bellevue Hotel**

Sponsored by SIGART ACM
Co-sponsored by ACM SIGGRAPH and ACM SIGCHI
Held in cooperation with AAAI

CONFERENCE DATES

- **April 1, 1999**
  Cutoff date for early registration and hotel conference rate
- **May 1-2, 1999**
  Workshops and tutorials
- **May 3-5, 1999**
  Conference technical sessions

CONFERENCE ORGANIZERS

General Chair:
Jeffrey M. Bradshaw (The Boeing Company)

Technical Program Co-Chairs:
Oren Etzioni (University of Washington)
Joerg P. Mueller (John Wiley & Sons Ltd, UK)
Craig A. Knoblock (University of Southern California)

Workshops Chair: Elisabeth Andre (DFKI GmbH, Germany)

Tutorials Chair: Carles Sierra (AI Research Institute - CSIC, Spain)

Software Demos Chair: Keith Golden (NASA Ames Research Center)

Robotics Demos Chair: Wei-Min Shen (University of Southern California)

Videos Chair: Maria Gini (University of Minnesota)

Posters Chair: Frank Dignum (Eindhoven University, The Netherlands)

Treasurer: Daniela Rus (Dartmouth College)

Local Arrangements Chair: Gene Ball (Microsoft Research)

http://www.cs.washington.edu/research/agents99

PA Expo 99

Five Major International Conferences Under One Roof

PAAM99

PAAM99 - The Practical Application of Intelligent Agents and Multi Agents
19th-21st April, 1999
www.practical-applications.co.uk/PAAM99

PAKeM99 - The Practical Application of Knowledge Management
21st-23rd April, 1999
www.practical-applications.co.uk/PAKeM99

PA JAVA 99

PA Java99 - The Practical Application of Java
21-23 April 1999
www.practical-applications.co.uk/PAJAVA99

PA CLP 99

PACLP99 - The Practical Application of Constraint Technologies and Logic Programming
19th-21st April, 1999 www.practical-applications.co.uk/PACLP99

PADD99

PADD99 - The Practical Application of Knowledge Discovery and Data Mining
21st-23rd April, 1999
www.practical-applications.co.uk/PADD99

PA Expo99 will bring together developers and users from commerce, industry and academia to examine and discuss solutions for solving real world problems found in key commercial areas such as: Transportation and Distribution, Finance, the Intelligent Internet, Decision Support, Information Management, Customer Retention, Fraud Detection, Planning and Scheduling, Business Modelling and Support.

The Commonwealth Conference and Events Centre, London, UK 19th-23rd April, 1999

SIEMENS

www.practical-applications.co.uk/Expo99