

Knowledge Representation and Reasoning — A History of DARPA Leadership

Richard Fikes, Tom Garvey

■ *A fundamental goal of artificial intelligence research and development is the creation of machines that demonstrate what humans consider to be intelligent behavior. Effective knowledge representation and reasoning methods are a foundational requirement for intelligent machines. The development of these methods remains a rich and active area of artificial intelligence research in which advances have been motivated by many factors, including interest in new challenge problems, interest in more complex domains, shortcomings of current methods, improved computational support, increases in requirements to interact effectively with humans, and ongoing funding from the Defense Advanced Research Projects Agency and other agencies. This article highlights several decades of advances in knowledge representation and reasoning methods, paying particular attention to research on planning and on the impact of the Defense Advanced Research Projects Agency's support.*

A fundamental goal of artificial intelligence (AI) is the creation of machines that demonstrate intelligent behavior, including, for example, the ability to assimilate and interpret information from the environment; adapt to environmental changes; formulate goals, and the procedures to accomplish them; communicate effectively with both people and other machines; and learn from and interact with the environment to effect desired changes. Effective knowledge representation and reasoning (KR&R) methods are a foundational requirement for achieving those capabilities whether the knowledge is represented in symbolic world models, in procedures, or in neural nets, and whether the reasoning methods are based on symbolic deduction or deep learning. Accordingly, the development of methodologies for representing knowledge in intelligent machines and of reasoning processes that operate over those representations has been a central activity in the field of AI throughout its history. This article highlights the development of symbolic KR&R technology with particular attention to planning and to the important ways in which the Defense Advanced Research Projects Agency (DARPA) has shaped that development.

The selection of KR&R methods for an AI system is driven by the characteristics of the problems to be solved by the system and the desired properties of the solutions. Desiderata

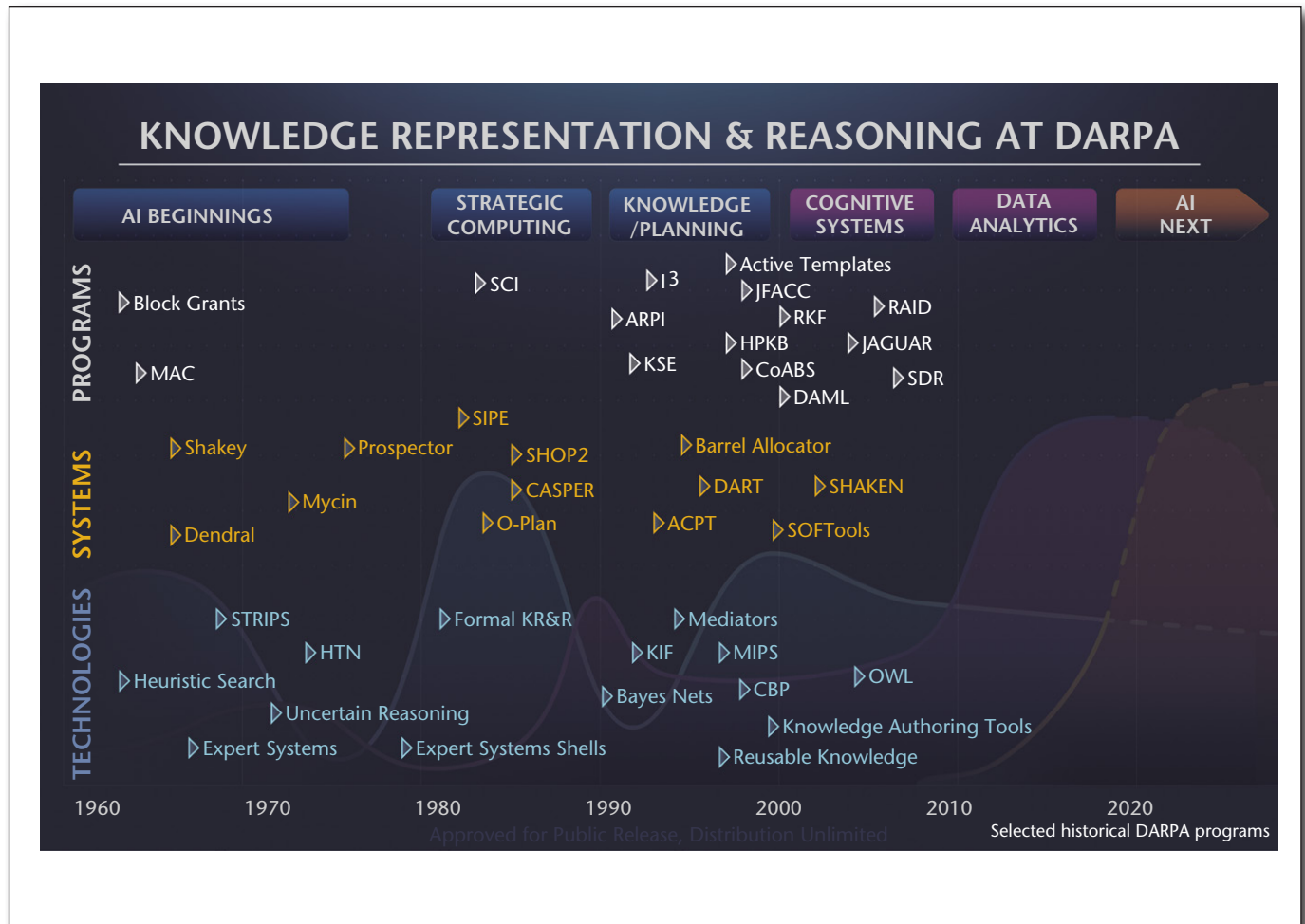


Figure 1. The Key Historical KR&R Programs, Systems, and Technologies that DARPA has Contributed to from the Beginning of AI Research into the Early 2000s.

Figure courtesy of DARPA.

include the ability to prove solution qualities such as correctness, completeness, and optimality; generate multiple hypotheses to explain observations; learn from a set of examples; capture problem-solving experience and bring it to bear on new problems; handle approximate or uncertain information and knowledge; and capitalize on massive amounts of data for solving problems.

KR&R is a broad field, and we only touch on a sampling of prominent steps in its development in which the results of DARPA's investment have been significant. Key historical KR&R programs, systems, and technologies that are discussed in this article are highlighted in figure 1. Advances in KR&R have been driven by interest in new challenge problems, the demands of more complex domains of interest, and the availability of improved computational support. In several cases, research success has led to the identification of new subfields and the attraction of additional support from other government agencies as well as commercial investors.¹

Knowledge Representation and Reasoning in Early AI Systems

Early AI systems tended to focus on broad-based reasoning methods (e.g., heuristic search, problem decomposition, means-ends analysis, Alpha-Beta game tree search, resolution theorem proving) that operated independently with little or no domain or task expertise. Representations of worlds of interest were minimal with highly restrictive expressive power (e.g., legal moves in a search tree, distance to a goal, if-then rules).

The academic centers of this early AI research were the Massachusetts Institute of Technology (MIT), The Carnegie Institute of Technology (later Carnegie Mellon University or CMU), and Stanford University. Researchers, including John McCarthy, Marvin Minsky, Allen Newell, and Edward Feigenbaum, could conduct their institutions' research during the first 10 to 15 years of DARPA's AI funding essentially unfettered by immediate applications. Thus, AI was ideally suited to graduate education, and enrollments at

each of the AI centers grew rapidly during the first decade of DARPA funding.

DARPA's early support launched a golden age of AI research and rapidly advanced the emergence of a formal discipline. Much of DARPA's funding for AI was contained in larger program initiatives. For example, as Director of DARPA's Information Processing Techniques Office, J.C.R. Licklider considered AI a part of his general charter of Computers, Command, and Control; and Project Mathematics and Computation (Project MAC), a project on time-shared computing at MIT, allocated roughly one-third of its \$2.3 million annual budget to AI research with few specific objectives (Committee on Innovations in Computing and Communications 1999).

Commenting in 1984 about the migration of AI-related research from Carnegie Institute of Technology's Graduate School of Industrial Administration to an autonomous department (and later a college) of Carnegie Mellon University (CMU), Allen Newell captured the transformation wrought by DARPA:

... the DARPA support of AI and computer science is a remarkable story of the nurturing of a new scientific field. Not only with MIT, Stanford, and CMU, which are now seen as the main DARPA-supported university computer-science research environments, but with other universities as well ... DARPA began to build excellence in information processing in whatever fashion we thought best.... The DARPA effort, or anything similar, had not been in our wildest imaginings.... (Newell 1984)

As AI research progressed, systems tackled increasingly complex and broad-based task domains, thus creating the need for representing domain knowledge more thoroughly and accurately to better support sophisticated computations using the knowledge. The first significant efforts to integrate domain knowledge into AI systems were in human language understanding, mobile robotics (*AI Magazine* DARPA leadership in AI, 2020), and expert systems. The perceived potential of expert system technology made its development a major focus of AI research starting in the early 1960s.

Expert Systems

Expert systems emulate the decision-making ability of human experts by using if-then inference rules and if-then situation-action rules obtained directly from experts. Prominent early expert systems that motivated further DARPA investment include: DENDRAL, for analyzing chemical compounds, developed by Edward Feigenbaum, Joshua Lederberg, and Bruce Buchanan at Stanford University starting in 1965 (Feigenbaum and Buchanan 1994); MYCIN, for diagnosing and treating blood infections, developed primarily by Ted Shortliffe and Bruce Buchanan at Stanford University starting in 1972 (Shortliffe 1976); and PROSPECTOR, for consultation with exploration geologists about mineral deposits, developed primarily

by Richard Duda, Peter Hart, and Nils Nilsson at SRI International (formerly called Stanford Research Institute - SRI) starting in 1974 (Benson 1986).

The simple if-then rule formulation used in early expert systems allowed for the rapid development of impressive capabilities; however, limitations in expert systems were equally rapidly discovered. As Bruce Buchanan and Ted Shortliffe observed,

“A reasoning program using only homogeneous rules with no internal distinctions among them thus fails to distinguish among several things, chance associations, statistical correlations, heuristics based on experience, cause of associations, definitions, knowledge about structure, taxonomic knowledge” (Buchanan and Shortliffe 1984).

If-then rules produced useful results, but they could not take the place of a human expert's decision-making.

The PROSPECTOR system notably overcame some of these limitations by expanding the if-then rule formulation to include semantic networks, pioneered in early natural language understanding systems, to represent taxonomic knowledge and to directly link the *if* side of each rule to the *then* side of other rules. Taxonomic knowledge enabled PROSPECTOR to infer the answers to interview questions it would otherwise need to ask the geologist. For example, if the geologist had already noted the presence of pyrites, a standard PROSPECTOR question “Are sulfides present?” would be skipped as taxonomic knowledge would show pyrites as a member of the class sulfides. The linking of rules enabled the system to quickly identify rules that were relevant to a given task.

Representation of Uncertainty

Expert systems highlighted the need for understanding uncertainty in domain knowledge because the if-then rules provided by domain experts for use in expert systems often stated conclusions that were only suggestive. Early attempts to represent uncertainty include MYCIN, which used certainty factors valued between -1 and 1 to represent confidence, and PROSPECTOR, which introduced Bayesian odds likelihood ratios.

These early methods for reasoning with uncertain information produced unpredictably inadequate and anomalous results in important situations. Research began to focus increasingly on Bayesian reasoning that uses evidence to update prior probabilities to new probabilities based on the evidence. Estimating prior probabilities, typically required for Bayesian reasoning, is often challenging and default estimates are often made from an assumption of a particular distribution (typically uniform). Non-Bayesian methods were also explored, including evidential reasoning based on the Dempster-Shafer generalization of probabilistic techniques for combination of evidence (Ruspini, Lowrance, and Strat 1992) and fuzzy sets (Zadeh 1975).

Bayesian networks (Pearl 1988) brought a rigorous foundation to reasoning from uncertain information and eventually emerged as the formalism of choice. A Bayes net is a directed acyclic graph that represents

probabilistic variables with their value or distribution as nodes and their conditional dependencies as values on directed arcs between nodes. For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms — given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Similar networks could be used to confirm the presence of a disease (or rule out other hypotheses) by computing the likelihood that certain symptoms should be observed when a particular disease is present.

The study and development of Bayes nets has become its own subfield. Efficient algorithms can perform inference and learning in Bayesian networks. Dynamic Bayesian networks can model sequences of variables (for example, speech signals or protein sequences). Generalizations of Bayesian networks, called influence diagrams, can represent, and solve, decision problems.

Strategic Computing Initiative (SCI)

DARPA's emphasis on real-world relevance in AI research grew during the late 1970s and 1980s and moved toward "mission-oriented direct research, rather than basic undirected research" (Fleck 1982). The changes led to increased funding for AI research as exemplified by the Strategic Computing Initiative (SCI; Roland, Shiman, and Aspray 2002), announced in 1983. DARPA committed \$1 billion over the planned 10-year course of the program to "create machine intelligence," and established three specific applications as research and development (R&D) objectives for SCI: a pilot's associate designated for the Air Force, an autonomous land vehicle for the Army, and a maritime battle management system for the Navy. The applications were intended to spark the military services' interest in developing AI technologies based on fundamental research. The SCI differed from most other large-scale national efforts in that its extremely ambitious goals required major advances in the underlying technology.

Because expert systems appeared promising in the late 1970s, they were slated for a prominent role in SCI: Expert systems were to be the reasoning agents that would give intelligence to the SCI applications. SCI provided significant funding to support the continuing development of expert system technology, including the addition of facilities for explaining conclusions and maintaining logical consistency as new knowledge is added. In particular, a number of environments (shells) to facilitate expert system construction were developed, including the Knowledge Engineering Environment (IntelliCorp 1985), ABE for building distributed expert systems (Erman, Lark, and Hayes-Roth 1988), and OPS5 (Forgy 1981).

Formalization of KR&R

Throughout the 1980s, DARPA supported the development of AI systems that tackled increasingly challenging problems, requiring representing knowledge of greater complexity and reasoning with greater

sophistication. Those requirements prompted the emergence of KR&R as a substantial subfield of AI R&D and, in 1989, the initiation of an ongoing series of biannual international conferences.² Much of the focus of the subfield was, and continues to be, on formal specifications of knowledge representation languages (KRLs) and reasoning methods, and on the design of KRLs for which core reasoning problems can be solved efficiently. As knowledge representation study progressed, a KRL came to be thought of as comprising the following components: A logical formalism providing a syntax for expressions in the language, a vocabulary of logical symbols, and an interpretation semantics for the logical symbols; a proof theory specifying the reasoning steps that are logically sound in the language; and an ontology that specifies a vocabulary of nonlogical symbols (that is, relations, functions, and object classes) for expressing knowledge in the language.

Also during the 1980s, knowledge representation subsystems evolved to include built-in methods for efficiently performing frequently occurring core reasoning tasks such as inheriting properties in a class-subclass hierarchy, determining the location of a newly described object class in a class-subclass hierarchy, and enforcing numerical and type constraints on the values a given property can have for a given class of objects (Brachman and Schmolze 1985).

Further development of reasoning technologies focused on various forms of logics such as temporal, context, modal, default, and nonmonotonic logics, and on methods for solving a variety of types of problems including: adding statements to and removing statements from a knowledge base (KB) in a manner that maintains the logical consistency of world models and logical support for derived sentences in the KB; asking whether a statement is entailed (theorem proving); asking for entailed instances of a statement schema (query answering and constraint satisfaction); asking for the effects of performing an action in a state (projection); asking for a plan to achieve a goal (automatic planning); and asking for a model that explains observations (diagnosis).

Scaling Up Knowledge-Based Systems

The logical formalism and proof theory of most representation languages are domain independent (for example, for predicate calculus and description logics), and are therefore applicable to a broad spectrum of application domains. Typically, therefore, the key step in developing a representation language for a given task domain is providing a domain-specific ontology to enable the expression of knowledge required to accomplish desired tasks.

As the quantity, breadth, and complexity of knowledge that AI systems needed grew, the task of building ontologies to express a system's knowledge and of building a system's KBs using those ontologies becomes a serious bottleneck in the development of large-scale AI systems.

Knowledge Representation and Reasoning

Sometime in the spring of 1978, I was contacted by William A. (Bill) Woods, who was then Director of the AI Department at Bolt Beranek and Newman (BBN) and the Principle Investigator of a large DARPA Project on Knowledge Representation for Natural Language Understanding (KRNLU). At the time, I was a 35-year-old philosophy professor and had no idea who Bill Woods was, nor what BBN was, pretty much no idea of what DARPA was, and could only guess at what was meant by “knowledge representation for NLU.” Bill was interested in hiring me as a consultant for the KRNLU project because he had heard that I knew something about Montague Grammar, a research program initiated in the late-1960s by the logician Richard Montague aimed at treating English (in principle, any natural language) as a formal language.

What did this have to do with KRNLU? One of Bill's PhD students, Ronald J. (Ron) Brachman, had joined him at BBN to work on the KRNLU Project and had been further developing a knowledge representation language (KRL) called KL-1 (or KL-ONE) (Brachman and Schmolze 1985) into which English statements would be translated and over which reasoning would be performed. My job was to develop a model-theoretical semantics for KL-1. KRLs were of course artificial languages, like programming languages and, more to the point, like the formal languages that logicians created and then analyzed such as first-order languages and Montague's intentional logic. The problem, I was told, was that most AI KRLs were not well-specified, especially not with respect to their semantics. And the KL-1 team was determined that their language would be different.

Much of the AI work at that time was influenced by then current forms of associationism in cognitive psychology. The results and insights of cognitive psychology were mixed up in AI with common-sense psychologic musings plus a good deal of armchair introspection —introspection by some very, very smart people, mind you, but still.... All of these ingredients, and more besides, were put in a blender and out came semantic networks! These really were networks; the formalisms involved graphs whose nodes and edges were typically labeled with bits and pieces of English. Given these labels, it was easy to be charitable and use your command of English to figure out what exemplars of the semantic networks were supposed to be saying; but really, there was no clear syntax. And there was nothing in the vicinity of a systematic semantic account beyond that yielded by charitable translation from semantic networks to English.

Although they were still committed to the idea of a graphical language, that is, to a semantic network formalism, Woods and Brachman deplored this state of affairs. Ron's thesis project was to develop a network formalism with a clear and well-specified syntax and a well-specified semantics that he called a structured inheritance network. (That name invokes another oddity with respect to the KL-1 team's commitments: an intense focus on taxonomies and taxonomic reasoning. This, too, I would claim, was motivated more by work in cognitive psychology; for example, on stereotypes and on priming relations between stereotypes than by any requirements analysis of the domains and applications for which KL-1 was intended.) So, Ron and Co. specified different types of nodes and edges in their networks and criteria of well-formedness for graphical (sub) structures: One could only link up a node of type *X* with one of type *Y* via an edge of type *Z*. Thus, one could define analogs of sentences or well-formed formulas, and specify their truth-claims. In sum, the KL-1 team was aiming at a formalism for which one could provide a syntax-directed semantic account and hence, provide a rationale and justification for a set of sound (validity- or truth-preserving) rules of inference.

Crucially, the aim was to treat KL-1 in the way that logicians treat formal languages. The central issues had to do with the notion(s) of logical form, of the distinction between the purely logical and the nonlogical (domain/application-specific) aspects of the overall structure of a representation, and of the specification and justification of rules of inference some of which, at least, depended solely on the logical forms of the representations. This approach to defining KR languages proved to be seminal in the development of KR&R.

At the end of the academic year 1978/1979, BBN offered me a full-time position as a computer scientist, although I had never taken a computer-science course, and I decided to accept — a decision I never regretted. Roughly 30 years later, I was the principal investigator of the SRI-led team in DARPA's machine reading program, a program that might as well have been called KRNLU.

I think my history exemplifies the interdisciplinary nature of AI research. And, from my parochial perspective, it also illustrates that DARPA's continued funding was absolutely essential to basic research in KR&R — funding that supported development of a core generic capability that underlies many of the central application areas in AI.

– David Israel

The major approaches to reducing that bottleneck were to develop technologies and methods for reusing KBs; enable multiple teams working from different sites to collaborate in building a large-scale AI system; and enable AI systems to communicate with other systems to cooperatively perform tasks.

Because vocabularies of many worlds of interest and substantial knowledge about those worlds are stable over time and support performing a wide variety of tasks in those worlds, KBs describing important aspects of many worlds of interest are potentially useable in multiple AI systems. Major efforts in the knowledge representation R&D community emerged to enable both ontologies, and KBs using those ontologies, to be constructed in a manner that allowed them to be reused in multiple systems for multiple tasks.

DARPA played a key role in catalyzing and supporting those efforts. It did so in multiple ways that illustrate DARPA's ability to bring together diverse research communities to work on important problems and then to provide continuing support of the progression of that work over an extended time frame.

Knowledge Sharing Effort (KSE) and Intelligent Information Integration (I3)

In 1990, DARPA joined with the National Science Foundation to bring together top AI and database researchers for a three-day workshop to consider how to enable the sharing and reuse of KBs and knowledge-based systems. That workshop resulted in the formation of the Knowledge Sharing Effort (KSE; Neches et al. 1991), a consortium of participants from over a dozen research centers with a goal to define, develop, and test infrastructure and supporting technologies to enable participants to build significantly bigger and more broadly functional systems. An important focus of KSE was the development of consensus-standard, domain-independent logical formalisms (for example, Knowledge Interchange Format; Genesereth and Fikes 1992) for use as the representation language for modular multiuse ontologies. DARPA jointly sponsored KSE with the Air Force Office of Scientific Research, the Corporation for National Research Initiatives, and the National Science Foundation.

In 1992, DARPA continued and expanded its support of multiuse knowledge by incorporating the objectives of KSE into a new initiative called I3 (ISX Corporation 1995). The Intelligent Information Integration objective was to enable the use of multiple heterogeneous distributed information sources (e.g., data bases, text, Web pages) in synergy with existing KBs. A significant Intelligent Information Integration initiative development was an architecture for information integration featuring a software module called a mediator (Wiederhold and Genesereth 1997). Mediators used knowledge about specific sets of data to provide information

to consumer system modules expressed in the representation language of the consumers.

High Performance Knowledge Base (HPKB)

KSE and the I3 demonstrated the importance of KB building, and in 1997 DARPA continued its support by initiating the High-Performance Knowledge Base (HPKB) program (Cohen, Schrag, and Burke 1998). With the advent of increasingly powerful computer hardware, the need arose to rapidly (within months) produce and test the technology needed to enable system developers to construct large (100K–1M axiom/rule/frame) KBs. Requirements included comprehensive coverage of topics of interest, reusability by multiple applications with diverse problem-solving strategies, and maintainability in rapidly changing environments. The program supported the development of methods for creating KBs by selecting, composing, extending, specializing, and modifying components from a library of reusable ontologies, common domain theories, and generic problem-solving strategies.

Challenge problems from two broad areas, Crisis Management and Battlespace Reasoning, were used to provide specific requirements for technical advances. They further provided shared data sources for use by the developers and a common framework for evaluation. These challenge problems were the centerpiece of the HPKB program and required participants to collaborate and focus their R&D efforts to find innovative approaches to the problems. Intensive annual evaluations assessed the completeness and correctness of the developed KBs, the time required to build the KBs, and the ease of modifying them to assimilate new or changed knowledge.

The HPKB program demonstrated both the utility and the expense of building a large KB for a given application by specially trained knowledge engineers. During the HPKB program, teams of knowledge engineers created KBs roughly at the rate of 10K axioms per year for prespecified task and evaluation criteria. The HPKB effort showed that it is possible to create KBs by reusing the content of knowledge libraries, and it demonstrated reuse rates ranging from 25 percent to 100 percent, depending on the application and the knowledge engineer. Still, it was acknowledged that the ability of a subject matter expert (SME) to directly enter knowledge is essential to improving the KB construction rates and content: Knowledge engineers lack the domain expertise to efficiently replace them, yet SMEs lack the knowledge-engineering skills to directly build the KBs. This challenge raised the question: Can knowledge-authoring technology be developed that enables the SMEs to directly create the KBs? In response, DARPA initiated the Rapid Knowledge Formation (RKF) program.

Rapid Knowledge Formation Program

In 2000, DARPA continued its support of KB building technology by initiating the RKF program with the objective of enabling distributed teams of SMEs

to quickly and easily build, maintain, and use KBs without the need for prior knowledge engineering experience. The emphasis was the rapid acquisition and integration of SME knowledge into a formal representation. Therefore, the program's primary requirement was the development of functionality enabling SMEs to understand the contents of a KB, enter new theories, augment and edit existing knowledge, test the adequacy of the KB under development, receive explanations of theories contained in the KB, and detect and repair content errors. Because of the complexity of these tasks, the approaches developed in RKF exploited the synergies among complementary AI technologies such as natural language discourse processing, problem solving and learning by analogy, and common-sense reasoning.

RKF was organized similarly to HPKB, with challenge problems administered by an evaluation contractor for the formal evaluation of technologies provided by the performers. There were two integrated teams and several component technology developers, with many supporting both teams. The program had two primary evaluation cycles: the first involved capturing biology textbook knowledge, and the second involved critiquing knowledge for military courses of action.

The SHAKEN knowledge-authoring tool developed by the SRI team is representative of the technology developed in the program (Barker and Blythe 2003). Their effort explored using SMEs, unassisted by AI technologists, to assemble models of mechanisms and processes from components using a graphical knowledge representation termed a concept map. The models were both declarative and executable, making the mechanisms and processes explainable by conventional inference methods (for example, theorem proving and taxonomic inference) and by various task-specific methods (for example, simulation, analogical reasoning, and problem-solving methods for particular tasks and domains such as time or space).

An independent group evaluated SHAKEN by having a group of four biologists independently build a knowledge system covering the content of one section of a college-level textbook on cell biology following a short course on the technology. The system was evaluated by posing a set of questions, and the answers were judged by another biologist to "be mostly correct." Although not conclusive, these results were significant in that they suggested that the basic machinery provided a method for knowledge acquisition without users directly encoding axioms. They also brought forward the intriguing possibility that relatively few components, perhaps a few thousand, independent of domain, are sufficient for SMEs to assemble models of virtually any mechanism or process.

DARPA Agent Markup Language Program and the Semantic Web

The World Wide Web was a paradigm shift in technology, and in 2000 the DARPA Agent Markup Language (DAML) program was initiated to support the

transformation of the human-readable World Wide Web into a machine-readable Semantic Web (BBN Technologies 2000). Although DAML was a continuation of technology to build KBs, it did not specifically target AI systems, but rather sought to provide principal components for a web of machine-readable data.

A primary outcome of the DAML program was the DAML specification: an ontology representation language integrated with the World Wide Web Consortium's Resource Description Framework so that the Semantic Web could be both a store of triples represented in Resource Description Framework and a set of ontologies represented in DAML to describe properties of the elements of those triples. DAML was a language based on description logic, and as such, had a formal semantics that precisely defined what reasoning could be done with the information in an ontology.

DAML contractors also worked with researchers outside the DARPA program from both the United States and Europe to produce an extension of the DAML language called DAML+OIL, submitted to the World Wide Web Consortium in 2002. That submission was the starting point for the World Wide Web Consortium formally standardizing a later version of the language as the Web Ontology Language (known as OWL; OWL Working Group 2012).

Although much of the Semantic Web vision has yet to be realized, OWL is important as a standard ontology representation language and is being used to build large-scale multiuse ontologies in multiple domains. For example, the biomedical domain has made effective use of ontologies to address the need for controlled vocabularies to support integration and joint ("cross-omics") analysis of experimental data. The Open Biologic Ontologies Foundry (Bandrowski et al. 2016) develops orthogonal interoperable reference ontologies in the biomedical domain, all represented in OWL. One such ontology, the Ontology for Biomedical Investigations, is an open-access, integrated ontology for the description of biologic and clinical investigations. Ontology for Biomedical Investigations provides a model for the design of an investigation, the protocols, instrumentation, and materials used, the data generated, and the type of analysis performed on it.

Planning

In the first part of this article, we described a progression through increasingly capable methods for representing knowledge and reasoning with that knowledge. A particularly important ongoing challenge for KR&R is to develop the capabilities necessary to use knowledge to create procedures for accomplishing goals—that is, planning.

In the US Department of Defense, as in any large enterprise, the generation, validation, and maintenance of plans for managing activities and investments to achieve important objectives is a major cost in resources and time. Planning is ubiquitous and

human centered, and its importance led DARPA to address plan generation, evaluation, execution, and management as areas in which AI systems could provide substantial assistance. Planning is also an important part of problem solving in other areas of AI, including human language communication, understanding sensed data (e.g., visual information), and robotics.

In this section we trace several key developments in AI planning that occurred with DARPA support and leadership.

The Shakey Project

The Shakey project, started by DARPA in 1966 at the Stanford Research Institute's Artificial Intelligence Center under the leadership of Charles Rosen, resulted in the first intelligent, mobile robot capable of problem-solving, reasoning, and planning. The Stanford Research Institute Problem Solver (STRIPS) planning system was developed to allow Shakey to plan its own methods for accomplishing tasks (Fikes and Nilsson 1971). STRIPS used a theorem prover to reason over a collection of logical statements representing a world state along with a collection of operators (actions) that could change the state of the world. STRIPS was designed to produce a plan, a sequence of actions that would transform a given starting world state to a goal world state. The STRIPS planning environment, the classic planning environment, was fully observable, deterministic, finite, static (except for Shakey's actions), and discrete (in time, actions, objects, and effects). This offered a principled basis for automated planning for Shakey's simple world but was too limited for applications in more complex worlds involving external agents, dynamics, non-determinism, resource usage, costs, and uncertainty.³

The ARPA-Rome Labs Planning Initiative (ARPI)

The SCI had programs that involved planning to support each of the services, including Pilot's Associate, Air Land Battle Management, and Fleet Command Center Battle Management. Following SCI, DARPA created the ARPA⁴ Rome Labs⁵ Planning Initiative (ARPI) to advance the state of the art in AI generative planning and scheduling (Nau 2007). Planning applications ranged from the creation of plans for future major contingencies as well as near-term tactics to real-time execution. The generation of plans required managing distributed components to collect and interpret information about situated plan objectives, manage resources for tasks, estimate plan effectiveness, control other procedures and equipment, and adapt to changes in the environment. ARPI opened this larger set of planning service requirements to the research community while also highlighting opportunities for AI approaches to problems that other DARPA programs were addressing.

A hallmark of many DARPA programs is a connection to an operational community that can identify challenges and benefit from and accept advanced

technology solutions to significant problems. Accordingly, ARPI emphasized strong connections with operational communities over the course of the program.

Dynamic Analysis Replanning Tool

ARPI's main transition partners were the US Air Force and the US Transportation Command responsible for logistics during Operation Desert Shield. A time-consuming part of the logistics process was the validation of transportation plans. Shortening this step would accelerate the whole planning process and make it more responsive to environmental changes. This led to the Dynamic Analysis and Replanning Tool (DART; Cross and Walker 1994), which integrated intelligent, expert analysis routines and database management systems to rapidly evaluate the feasibility of human-authored logistics plans. DART became the first major operational military deployment of AI. With DART, deployment plans that had previously taken four days to generate feasibility estimates were routinely completed within hours (DARPA 2015), resulting in significant savings during Operation Desert Storm.

Hierarchical Task Networks

To move beyond classic planning, ARPI investigated hierarchical task networks (HTNs), an approach first described by Sacerdoti (1973). Instead of the linear sequence of logical steps that STRIPS provided, HTNs, using templates representing precompiled strategies, recursively decompose complex tasks into subtasks terminating in primitive subtasks that can be directly performed to accomplish the objective. HTNs increased the efficiency of planning by using explicitly encoded human expertise (rather than needing to discover it), enabling more challenging real-world problems to be tackled.

HTNs have become the planning technology of choice for groups addressing real-world planning tasks. The National Aeronautics and Space Administration uses HTN planners for a variety of space missions; for example, Continuous Activity Scheduling Planning Execution and Replanning, or CASPER, is being used in a range of projects including autonomous spacecraft, autonomous rovers and uninhabited aerial vehicles (Chien et al. 2000). Domain-configurable systems such as O-Plan (Tate, Drabble, and Kirby 1994), SIPE-2 (Wilkins 1988), and SHOP2 (Nau 2003) have been used in a variety of applications, and domain-specific HTN planning systems have been built for several application domains.

Case-Based Reasoning and Planning

Interactive tools for case-based reasoning and planning, developed under ARPI, derived solutions to a problem based on solutions to past problems. In a case-based planning system, a case is a record of a situation and a resulting action stored in a case base. Case-based reasoning and planning systems offer access to past experiences for use in solving current problems, and they store the solutions to current problems as they are being made (Mulvehill 1996). Case-based

reasoning and planning was used to support force-deployment planners through the accumulation of user-built plans, query-driven browsing of past plans, and several plan-functionality-analysis primitives (Veloso, Mulvehill, and Cox 1997).

Plan-Authoring Tools

As planning techniques were being developed, it was clear that certain important problems would not be easily automated, but could be helped by the use of structured plan knowledge. The Air Campaign Planning Tool (ACPT–Maybury 1995) was developed under ARPI to capture the rules for planners that derived from new Air Force doctrine. ACPT guided human planners through the necessary steps to create effective plans, ensuring each step was tied to specific objectives and highlighting missing components. ACPT was deployed during major exercises in Korea and Europe and was used to plan missions for enforcing no-fly zones after Desert Storm.

The Active Templates program, inspired by ARPI and begun in 1996, embodied a human-driven concept in which plans for small-unit special operations would be constructed using knowledge represented as experientially derived templates containing slots that other templates, resources, and actions could instantiate. Thus, a particular mission template might include slots for the location and time frame, personnel needed, resources required, movement actions, and constraints on the way slots could be filled out. SOFTools, a product of Active Templates, allowed users to lay out plans against a timeline, understand resource capabilities and constraints, merge plans from other teams, eliminate conflicts, coordinate changes as plans solidified, and track mission execution. Studies showed that these tools decreased planning time by a factor of 4, allowing more time for rehearsal and critical decision-making. SOFTools was in active use for several years, and later served as the basis for other tool sets for mission planning.

Mixed-Initiative Planning and Scheduling

The complexity of real-world planning and scheduling problems has proven to be a formidable barrier for automation. However, even if full automation were possible, humans may not trust results unless they have a first-hand view of and input into the solution. Thus, automatic plan generation for complex domains, even when possible, is not always the best solution for problems for which user acceptance, responsibility, and control are critical.

Work in the area of mixed-initiative planning and scheduling is generally defined in terms of a division of labor into those areas in which the computer excels (e.g., extensive search through alternative possibilities to find a desired solution) and those in which human input and guidance are valuable (e.g., decision-making at higher levels of abstraction; Myers 1997; Burstein and McDermott 1994). Mixed-initiative systems must be flexible in allocating tasks between domains in which the user or computer has the initiative.

The most directly deployable technologies to emerge from ARPI were for scheduling, notably Carnegie Mellon University's Barrel Allocator (Becker and Smith 2000) used by the US Air Force Air Mobility Command (AMC) to schedule aircraft and crews needed by the AMC to meet current and upcoming airlift requirements. Importantly, the Barrel Allocator allowed a variable level of automation. For example, task requirements typically oversubscribed the available resources, preventing an automated scheduler from fulfilling all necessary tasks. However, based on knowledge unavailable to the system, a human planner could relax a noncritical constraint to create a solution. The Barrel Allocator enabled What if? exploration of the solution space: A user could selectively alter constraints to examine solutions produced by the system and interactively create a preferred solution while also gaining a better understanding of the capabilities of the allocator and trust in its decisions.

Scaling Up Planning

ARPI was the first large-scale research program to develop planning technology for a wide range of problems. It was the foundation for major investments in planning technology by DARPA and other agencies over many years. Commercial interest in ARPI projects led to the formation of several startup companies which, in many cases, were driven by specific operational challenges faced by the US Department of Defense and were intended to produce technologies that could transition directly into the services. Here we discuss a sampling of DARPA planning programs that followed from ARPI.

Joint Forces Air Component Commander Program

The Joint Forces Air Component Commander (JFACC) is the senior air commander in a major military campaign and is responsible for the daily Air Tasking Order that defines missions, roles, and schedules for all aircraft in the theater of operations. ACPT helped educate the research community on some of the challenges of working in this domain. Started in 1996, DARPA's JFACC program aimed to provide AI planning technology to decrease the time it took to respond to a new mission requirement from days to hours or minutes, to evaluate the degree to which proposed missions would meet objectives, and to ensure availability of necessary resources.

Technology demonstrations indicated the feasibility of meeting these aims, but the costs and risks imposed by the need to modify and extend or replace existing system components prevented direct transition of most results other than improved versions of the Barrel Allocator and ACPT. However, concepts developed in JFACC spawned additional programs outside of DARPA.

Joint Air/Ground Operations: Unified Adaptive Replanning Program

The Joint Air/Ground Operations: Unified Adaptive Replanning (JAGUAR) program, started in 2003, aimed

to pick up where JFACC left off by integrating technologies for automating plan generation, plan assessment, and model adaptation in a consistent, model-based framework that could respond to the forthcoming transformations in air operations.

JAGUAR followed an approach similar to that of Active Templates but for a much larger and more complex set of problems. Part of the JAGUAR program focused on tools to monitor air operation planning activities and to capture knowledge about air planning processes, plans, and effects. This knowledge would allow plans to be decomposed into fragments that could then be reassembled by automated planning tools into new operational plans tailored and optimized for new situations. Like JFACC, JAGUAR never transitioned as a program to Air Force operations due to mismatches with current operational systems, but the JAGUAR core plan representation was adapted and used in later Air Force programs integrating cyber, space, and air operations.

Control of Agent-Based Systems Program

In 1998, DARPA created the Control of Agent-Based Systems program to provide enabling technology for heterogeneous software agents to solve problems. Accordingly, it covered a range of topics that included directability of agents, safety of agent actions, semantic match-making (i.e., finding agents for tasks), and agent intercommunication. An important result of the program was the Control of Agent-Based Systems Grid, which provided dynamic registration of service-providing agents, discovery of registered services, and ontological mapping of services to enable interoperability. The Control of Agent-Based Systems Grid was the integrating vehicle for a successful experimental demonstration involving 21 multiagent systems interoperating to plan a noncombatant evacuation utilizing 75 agents across nine distributed sites using multiple computing platforms (Allsopp et al. 2002). The success of Control of Agent-Based Systems led to the DAML program (described above).

Software for Distributed Robots Program

The 18-month Software for Distributed Robots program, started in 2004, aimed at eliciting desirable aggregate behaviors from a large group of small, simple robots capable of communication and limited sensing to accomplish a set of tasks within a large, previously unexplored area (Howard, Parker, and Sukhatme 2006; Konolige et al. 2006). A sequence of successful exercises using groups of 25, 50, and then 100 robots, demonstrated that robots could enter a new building and then establish and maintain a communication network; organize efficient mapping activities; avoid obstacles; collect, share, aggregate, and validate map information; use the created map to formulate paths; organize an efficient and complete search of the area for objects of interest; and other activities. The group could also recognize and fill gaps in its ad hoc communication network caused by movement or failure of a robot.

Real-Time Adversarial Intelligence and Decision-Making Program

Most US Department of Defense missions must take into account an intelligent and resourceful adversary. Real-Time Adversarial Intelligence and Decision-Making, started in 2004, was the first program to explicitly model and reason about an adversary's objectives, intent, and plans. Real-Time Adversarial Intelligence and Decision-Making used linguistic geometry (Stilman and Aldossary 2015) to characterize the space of engagements for small units in urban terrain. It then applied approximate game-theoretical and deception-sensitive algorithms (Gilpin 2009; Roughgarden, Nisan, Tardos, and Vazirani 2007) to anticipate future actions of human teams, Blue and Red, in urban conflict to suggest the best actions for Blue against the actions of Red in real-time.

Extensive experiments compared the performance of teams staffed with retired and active US military personnel operating in a four-square-kilometer urban area where Blue controlled a simulated force of 30 to 35 four-soldier groups and Red controlled about 30 groups comprised of several insurgents each. In half of the experiments, the Blue staff was assisted by Real-Time Adversarial Intelligence and Decision-Making and significantly outscored unassisted Blue teams in a preponderance of examples by about 10 percent.

The US Army and others are extending Real-Time Adversarial Intelligence and Decision-Making technology to make it more robust, applicable to diverse conditions and scenarios, and easy to use in real combat environments.

Conclusions and Future Challenges

In attempting to describe the influence DARPA has had on the development of AI representation and reasoning methodologies, we have focused primarily on declarative symbolic world modeling and on foundational reasoning methods having applicability to a broad range of tasks. Much of the research in KR&R has moved into application subfields (for example, learning, human language understanding, robotics, and perception). Other articles in this collection describe, in more detail, the KR&R approaches underlying their specific domains.

DARPA has invested a great deal in the development of planning capabilities aimed at assisting a human planner to create effective plans with less effort. Multiple programs are currently underway at DARPA to support greater autonomy for software and physical agents (DARPA 2019).

Although much of the focus of attention in AI has moved away from traditional symbolic KR&R approaches, the results of R&D for KR&R have had many successes (e.g., the R1/XCON system used to configure VAX computers [McDermott 1980], the Jeopardy champion Watson system [Ferrucci et al. 2010], and the DART logistics system used in the Gulf War) and current uses (e.g., the A* search algorithm that is the basis for routes determined by the

Global Positioning System [Hart, Nilsson, and Raphael 1968] and knowledge graphs). Knowledge representation technology has in fact become critical to many products in diverse industry settings today by enabling the building of symbolic KBs that provide structured descriptions of objects of interest and connections between them to multiple applications. For example, the search engines of Google and Microsoft are supported by knowledge graphs that enable their response to user queries about entities to include an array of facts about the entities. Commercial knowledge graphs are massive in size (for example, Google's knowledge graph currently has over 1 billion entities and 70 billion facts) and make use of ontologies, strongly typed entities, relations with domain and range inference, and so forth, developed by the KR&R research community (Noy et al. 2019).

Challenges continue to exist in most areas of KR&R as harder, more complex problems are undertaken, and knowledge is used in more sophisticated ways. Continuing challenges for knowledge graphs include entity disambiguation, managing changing knowledge, knowledge extraction from multiple structured and unstructured sources, and managing operations at scale. Important areas of active KR&R research motivated by more complex problems include reasoning from evidence about complex environments; effective use of context by intelligent systems; maintaining multiple hypotheses; explaining reasoning; developing plans to shape complex future events; linking observation, explanation, planning and execution to create truly autonomous agents; and creating agents that can act effectively in teams with humans.

In recent years, a great deal of attention has been focused on the artificial neural representations used in deep learning systems, and their capabilities have been extremely impressive and promising. Important challenges remain for current learning systems to enable them to learn procedures and processes, adapt to changing conditions, provide necessary performance guarantees, and, in particular, provide users with explanations of their results. Many researchers, including the authors, foresee a future in which those capabilities are provided by unifying symbolic KR&R technologies with data-centric machine learning methods to produce intelligent systems that are significantly more generalizable, transparent, and directable than possible today.

Recognizing that powerful learning methods are a necessary but not a sufficient basis for intelligent systems, it is vital going forward that a broad range of KR&R capabilities and approaches be actively pursued and supported in the AI research community.

Acknowledgments

Although we are unable to thank them individually, the authors wish to acknowledge the generous help, keen memories, and continuing fervor of the many former DARPA program managers, staff, and

performers who helped us understand details of their vision, motivation, programs, challenges, and results. We would like to especially acknowledge the influence on our own work of two individuals, Charlie Rosen, the founder of the SRI AI Lab and father of Shakey the Robot, and Nils Nilsson, a visionary and a teacher. Nils' book, *The Quest for Artificial Intelligence — A History of Ideas and Achievements*, was an invaluable reference for this article. Early on, Nils stated that the most important challenge for AI researchers was the representation problem. His insight has held up over these many decades! This project was supported by DARPA.

Notes

1. Much of the historical material in this article has been obtained directly from existing publications, including the book by Nils Nilsson (Nilsson 2009) and *Funding a Revolution: Government Support for Computing Research* (Committee on Innovations in Computing and Communications 1999).
2. The 2020 Conference on Principles of Knowledge Representation and Reasoning will be held in Patras, Greece (kr2020.inf.unibz.it/).
3. Refinements of the STRIPS paradigm viewed it as a satisfiability problem, allowing for the use of more generic reasoning systems rather than requiring a specialized planning engine, e.g., SATPLAN (Kautz and Selman 1992).
4. An alternate reference for DARPA.
5. Rome Laboratories, housed on Griffiss Air Force Base, Rome, New York.

References

- Allsopp, D.; Beautement, P.; Kirton, M.; Bradshaw, J.; Suri, N.; Durfee, E.; and Thompson, C. 2002. Coalition Agents Experiment: Multi-Agent Co-operation in International Coalitions. *IEEE Intelligent Systems* 17(3): 26–35. doi.org/10.1109/MIS.2002.1005628.
- Bandrowski, A.; Brinkman, R.; Brochhausen, M. S.; Brush, M. H.; Bug, B.; Chibucos, M. C.; Clancy, K.; Courtot, M.; Derom, D.; Dumontier, M., et al. 2016. The Ontology for Biomedical Investigations. *PLoS One* 11(4): e0154556. doi.org/10.1371/journal.pone.0154556.
- Barker, A., and Blythe, J. 2003. A Knowledge Acquisition Tool for Course of Action Analysis. In *Proceedings of the Fifteenth Conference on Innovative Applications of Artificial Intelligence*. 43–50. Palo Alto, CA: Association for the Advancement of Artificial Intelligence.
- BBN Technologies. (August 2000). *The DARPA Agent Markup Language Homepage*. <http://www.daml.org/index.html>.
- Becker, M., and Smith, S. 2000. Mixed-Initiative Resource Management: The AMC Barrel Allocator. In *Proceedings 5th International Conference on AI Planning and Scheduling, Breckenridge, Colorado*. Published by The AAAI Press, Menlo Park, California.
- Benson, I. 1986. Prospector: An Expert System for Mineral Exploration. In *Computer Assisted Decision Making: Expert Systems, Decision Analysis, Mathematical Programming*. Mitra, G., Ed. Oxford, UK: North Holland. 17–26.
- Brachman, R. J., and Schmolze, J. G. 1985. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2): 171–216. doi.org/10.1207/s15516709cog0902_1.

- Buchanan, B. G., and Shortliffe, E. H. 1984. *Rule-Based Expert Systems — The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison Wesley.
- Burstein, M., and McDermott, D. 1994. Issues in the Development of Human-Computer Mixed-Initiative Planning Systems. In *Proceedings of the DARPA/RL Planning Initiative*. San Mateo, CA: Morgan Kaufmann.
- Chien, S. A.; Knight, R.; Stechert, A.; Sherwood, R.; and Rabideau, G. 2000. Using Iterative Repair to Improve Responsiveness of Planning and Scheduling. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.
- Cohen, P. R.; Schrag, R. C.; and Burke, M. 1998. The DARPA High-Performance Knowledge Bases Project. *AI Magazine* 19(4): 25–49.
- Committee on Innovations in Computing and Communications. 1999. *Funding a Revolution: Government Support for Computing Research*. Washington, DC: National Academy Press.
- Cross, S., and Walker, E. 1994. DART: Applying Knowledge-Based Planning and Scheduling to Crisis Action Planning. In *Intelligent Scheduling*. Zweben, M., and Fox, M., Eds. San Francisco, CA: Morgan-Kaufmann. 711–29.
- DARPA. 2015. *Breakthrough Technologies for National Security*. Arlington, VA: Defense Advanced Research Projects Agency. March. www.darpa.mil/attachments/DARPA2015FINAL.pdf.
- DARPA. 2019. Training AI to Win a Dogfight. DARPA News, May 8. www.darpa.mil/news-events/2019-05-08.
- Erman, L. D.; Lark, J. S.; and Hayes-Roth, F. 1988. ABE: An Environment for Engineering Intelligent Systems. *IEEE Transactions on Software Engineering* 14(12): 1758–70. doi.org/10.1109/32.9062.
- Feigenbaum, E. A., and Buchanan, B. G. 1994. DENDRAL and Meta-DENDRAL: Roots of Knowledge Systems and Expert System Applications. In *Artificial Intelligence in Perspective*. Bobrow, D. G., Ed. Cambridge, MA: MIT/Elsevier. 233–40.
- Ferrucci, D.; Brown, E.; Chu-Carroll, J.; Fan, J.; Gondek, D.; Kalyanpur, A.; Lally, A.; Murdock, J.; Nyberg, E.; Prager, J., et al. 2010. Building Watson: An Overview of the DeepQA Project. *AI Magazine* 31(3): 59–79. doi.org/10.1609/aimag.v31i3.2303.
- Fikes, R., and Nilsson, N. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence*. 189–20. Los Altos, CA: William Kaufmann. doi.org/10.1016/0004-3702(71)90010-5.
- Fleck, J. 1982. Development and Establishment in Artificial Intelligence. In *Scientific Establishments and Hierarchies. Sociology of the Sciences*, Vol. 6. Elias, N., Ed. Dordrecht, Holland: Reidel. 169–217.
- Forgey, C. 1981. *OPSS User's Manual*. Technical Report CMU-CS-81-135. Pittsburgh, PA: Carnegie Mellon University.
- Genesereth, M. R., and Fikes, R. E. 1992. *Knowledge Interchange Format Version 3.0 Reference Manual*. Stanford, CA: Stanford Logic Group, Stanford University.
- Gilpin, A. 2009. *Algorithms For Abstracting and Solving Imperfect Information Games*. Doctoral dissertation, CMU-CS-09-127. Carnegie Mellon University, Pittsburgh, PA. www.cs.cmu.edu/~gilpin/papers/thesis.pdf.
- Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4(2): 100–7. doi.org/10.1109/TSSC.1968.300136.
- Howard, A.; Parker, L.; and Sukhatme, G. 2006. The SDR Experience: Experiments with a Large-Scale Heterogeneous Mobile Robot Team. In *Experimental Robotics IX: The 9th International Symposium on Experimental Robotics*. Ang, M., and Khatib, O., Eds. 121–30. Berlin, Germany: Springer.
- IntelliCorp. 1985. *KEE Software Development User's Manual*. Mountain View, CA: IntelliCorp.
- ISX Corporation. 1995. I3 Initiative Home Page. www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-6/web-agent/www/i3.html.
- Kautz, H., and Selman, B. 1992. Planning as Satisfiability. In *Proceedings of the Tenth European Conference on Artificial Intelligence*. 359–63. New York: John Wiley.
- Konolige, K.; Fox, D.; Ortiz, C.; Agno, A.; Eriksen, M.; Limketkai, B.; and Vincent, R. 2006. Centibots: Very Large Scale Distributed Robotic Teams. In *Experimental Robotics IX: The 9th International Symposium on Experimental Robotics*. Ang, M., and Khatib, O., Eds. 131–40. Berlin, Germany: Springer.
- Maybury, M. T. Distributed, collaborative, knowledge-based air campaign planning. In *Proceedings of the NATO/AGARD Lecture Series 200 on Knowledge-Based Functions in Aerospace Mission Systems*. (Nov. 6•17, 1995, Torrejon, Spain, Chatillon, France, and NASA Ames, CA).
- McDermott, J. 1980. R1: An Expert in the Computer Systems Domain. In *Proceedings of the First AAAI Conference on Artificial Intelligence, AAAI 1980*. 269–71. Los Altos, CA: William Kaufmann.
- Mulvehill, A. M. 1996. Building, Remembering, and Revising Force Deployment Plans. In *Advanced Planning Technology (ARPT) Proceedings*. Tate, A., Ed. Palo Alto, CA: Association for the Advancement of Artificial Intelligence.
- Myers, K. L. 1997. Abductive Completion of Plan Sketches. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference*. 687–93. Menlo Park, CA: Association for the Advancement of Artificial Intelligence.
- Nau, D. 2003. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research* 20: 379–404. doi.org/10.1613/jair.1141.
- Nau, D. S. 2007. Current Trends in Automated Planning. *AI Magazine* 28(4): 43–58.
- Neches, R.; Fikes, R. E.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; and Swartout, W. R. 1991. Enabling Technology for Knowledge Sharing. *AI Magazine* 12(3): 6–56, 9–15.
- Newell, A. 1984. Reports on Artificial Intelligence from Carnegie-Mellon University: Introduction to the Comtex Microfiche Edition. *AI Magazine* 5(3): 35–9.
- Nilsson, N. J. 2009. *The Quest for Artificial Intelligence — A History of Ideas and Achievements*. Cambridge, England: Cambridge University Press doi.org/10.1017/CBO9780511819346.
- Noy, N.; Gao, Y.; Jain, A.; Narayanan, A.; Patterson, A.; and Taylor, J. 2019. Industry-Scale Knowledge Graphs: Lessons and Challenges. *Communications of the ACM*. 62(8): 36–43.
- Owl Working Group. (2012, December 11). Web Ontology Language (OWL). w3.org: <https://www.w3.org/OWL/>
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

The AAAI Feigenbaum Prize

The 2021 AAAI Feigenbaum Prize is awarded biennially to recognize and encourage outstanding Artificial Intelligence research advances that are made by using experimental methods of computer science. The laboratories for the experimental work are real-world domains, and the power of the research results are demonstrated in those domains.

The Feigenbaum Prize may be given for a sustained record of high-impact seminal contributions to experimental AI research; or it may be given to reward singular remarkable innovation and achievement in experimental AI research.

The prize is \$10,000 and is provided by the Feigenbaum Nii Foundation and administered by AAAI. The 2021 Feigenbaum Prize will be awarded in conjunction with the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21) in Vancouver, British Columbia, Canada.

The deadline for nominations is September 25, 2020.

www.aaai.org/Awards/feigenbaum.php

Roland, A.; Shiman, P.; and Aspray, W. 2002. *Strategic Computing: DARPA and the Quest for Machine Intelligence, 1983–1993*. Cambridge, MA: MIT Press.

Roughgarden, T.; Nisan, N.; Tardos, E.; and Vazirani, V. 2007. *Algorithmic Game Theory*. Cambridge, UK: Cambridge University Press.

Ruspini, E. H.; Lowrance, J. D.; and Strat, T. M. 1992. Understanding Evidential Reasoning. *International Journal of Approximate Reasoning* 6(3): 401–24. doi.org/10.1016/0888-613X(92)90033-V.

Sacerdoti, E. D. 1973. Planning in a Hierarchy of Abstraction Spaces. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*. 412–30. Los Altos, CA: William Kaufmann.

Shortliffe, E. H. 1976. *Computer-Based Medical Consultations: MYCIN*. New York, NY: Elsevier.

Stilman, B., and Aldossary, M. 2015. Revisiting Major Discoveries in Linguistic Geometry with Mosaic Reasoning. *Procedia Computer Science*. 46: 784–93.

Tate, A.; Drabble, B.; and Kirby, R. 1994. *O-Plan2: An Open Architecture for Command, Planning and Control*. *Intelligent Scheduling*. San Francisco: Morgan Kaufmann.

Veloso, M.; Mulvehill, A.; and Cox, M. 1997. Rationale-Supported Mixed-Initiative Case-Based Planning. In *Proceedings of the Ninth Innovative Applications of Artificial Intelligence Conference on Artificial Intelligence IAAI-97*. Palo Alto, CA: Association for the Advancement of Artificial Intelligence.

Wiederhold, G., and Genesereth, M. 1997. The Conceptual Basis for Mediation Services. *IEEE Expert* 12(5): 38–47. doi.org/10.1109/64.621227.

Wilkins, D. E. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm*. San Francisco, CA: Morgan Kaufmann.

Zadeh, L. 1975. The Concept of a Linguistic Variable and Its Application to Approximate Reasoning—I. *Information Sciences* 8(3): 199–249. doi.org/10.1016/0020-0255(75)90036-5.

Richard Fikes is professor emeritus of computer science at Stanford University. He was Director of Stanford's Knowledge Systems Laboratory where he led major projects for DARPA and other Federal Government agencies focused on developing large-scale distributed repositories of computer-interpretable knowledge, collaborative development of multi-use ontologies, technology enablement for the Semantic Web, reasoning methods applicable to large-scale KBs, and knowledge-based technology for intelligence analysts. Fikes is best known as co-developer of the STRIPS automatic planning system, the Knowledge Interchange Format, the Ontolingua ontology representation language and Web-based ontology development environment, the Open Knowledge Base Connectivity application programming interface for knowledge servers, and IntelliCorp's Knowledge Engineering Environment system.

Tom Garvey is a senior principal scientist emeritus in SRI International's AI Center where he started his career as a Stanford PhD student working on mobility, theorem proving, and, with Fikes, the STRIPS planner for Shakey the Robot. He pioneered work in active, goal-driven computer vision. His research career spanned work in computer vision, planning, approximate reasoning, command and control, human-directed autonomy, and learning. As program manager and assistant office director for planning and command and control in DARPA's Information Systems Office he was responsible for programs in command and control, logistics, intelligence, and crisis avoidance. His current interests include human-on-the-loop autonomy and anticipatory control of sensing, reasoning, and learning.