AI Magazine Volume 15 Number 1 (1994) (© AAAI)

AAAI–93 Workshops

Summary Reports

■ The American Association for Artificial Intelligence sponsored a number of workshops in conjunction with the Eleventh National Conference on Artificial Intelligence held 11–15 July 1993 in Washington, D.C. This article contains reports of four of the workshops that were conducted: AI Models for System Engineering, Case-Based Reasoning, Reasoning about Function, and Validation and Verification of Knowledge-Based Systems.

The American Association for Artificial Intelligence sponsored a number of workshops in conjunction with the Eleventh National Conference on Artificial Intelligence (NCAI) held 11–15 July 1993 in Washington, D.C. This article contains reports of four of the workshops that were conducted: AI Models for System Engineering, Case-Based Reasoning, Reasoning about Function, and Validation and Verification of Knowledge-Based Systems.

AI Models for System Engineering

The AI Models for System Engineering Workshop included 11 presentations divided into 2 broad categories: (1) the need for using AI in system engineering and (2) existing AI applications in system engineering.

A morning discussion centered on large system engineering problems that could benefit from AI: model-based system engineering, the monitoring of the effects of change in large systems, organizational aspects of system engineering, and the integration of software into large systems. Presenters described the basic issues in system engineering that need to be solved and then made suggestions about the kinds of AI technique that could be used to solve them. The general consensus was that AI can make significant contributions toward solving major system engineering problems.

The AI applications described in the afternoon ranged from innovative to traditional; all of them, though, made possible the solution of difficult system engineering problems. Two of the systems used rules and constraint propagation to help the user solve complex systems of constraint equations that described satellite systems. Another system described how formal specifications could be used by a theorem prover to prove the correctness and completeness of the design of a complex digital circuit. A prototype of an integrated design assistant system used case-based reasoning and rules to describe information security systems. A truly innovative system used text-understanding methods to automatically read requirement documents and extract from them the exact requirements for a system. Finally, a system was presented that was used to solve scheduling and resource-allocation optimization problems. The system combined many traditional search techniques and, at each stage of the search process, would use domain heuristics to decide which search technique to apply.

Without a doubt, the liveliest part of the workshop was the lunch-time debate on the role of formal methods in the validation and verification of KBSs.

The conclusions drawn during the final discussion of the workshop were that although there are many grand challenges in system engineering that will require innovative AI techniques, there are also many smaller system engineering problems that have been solved successfully by traditional AI methods.

The workshop was attended primarily by industry researchers and developers who voiced the concern that the academic AI community has ignored system engineering as an application domain either because it is unfamiliar with the area or because it is no longer interested in applications. The participants agreed to try to increase the awareness of the AI community about system engineering and, at the same time, educate the system engineering community about the usefulness of AI techniques.

Costas Tsatsoulis
University of Kansas

Case-Based Reasoning

Case-based reasoning (CBR) is a method for solving new problems by retrieving and adapting relevant solutions from a memory of past cases. The NCAI-93 Workshop on Case-Based Reasoning, chaired by David Leake of Indiana University, was a two-day workshop that brought together over 60 investigators from academics and industry to share results, assess the state of the art in CBR, and define directions for future progress. The workshop included paper sessions and poster presentations on fundamental theoretical issues in CBR—case representation, indexing and retrieval, and learning—and on the application of CBR to task areas such as planning, scheduling, teaching, and decision support. The workshop also highlighted four invited addresses given by Kris Hammond, the University of

Chicago; Roger Schank, the Institute for the Learning Sciences at Northwestern University; Janet Kolodner, the Georgia Institute of Technology; and Chris Riesbeck, the Institute for the Learning Sciences. The workshop closed with a panel and open discussion on deploying CBR technology.

On the first day of the workshop, Kris Hammond presented a general framework for CBR and proposed that CBR investigations can be divided into three categories based on their goals, which, in turn determine how their results should be evaluated. The categories he proposed were (1) *true-faith CBR,* aimed at addressing fundamental theoretical issues; (2) *hard-core CBR,* aimed at applying true-faith theories to challenging tasks; and (3) *CBR light,* aimed at achieving the best possible performance by using ideas from CBR to provide a technological advantage. Workshop participants often referred to these categories in describing and evaluating CBR efforts during later discussion. Roger Schank's address advanced a case-based model of human learning and argued that the goal of education should be furthering the acquisition of appropriate cases. Schank examined cognitive ramifications and practical consequences of this view for instruction and proposed that education should be conducted through *goal-based scenarios* in which students learn skills and conceptual knowledge through activities in pursuit of compelling goals in rich learning environments.

On the second day, Janet Kolodner discussed the state of the art in CBR applications. She reported on the positive response that CBR systems have received from the user community, for both their performance and their comparative ease of development. She also identified important misconceptions that have arisen about CBR and CBR issues and argued that these misconceptions demonstrate the need to clarify the CBR paradigm, methodology, and cognitive model.

Chris Riesbeck began his address by proposing a new definition of AI: "AI is the search for answers to the eternal question, Why are computers so stupid?" Past answers have included "they don't know anything" and "they don't learn." For CBR systems, he suggested the new answer, "they're too fussy": They can retrieve a good case, have problems adapting it, and end up with no answer at all. He presented an alternative framework, nicknamed *shoot first and ask questions later,* in which the CBR system finds a good case, tells the user, and continues processing. If later retrieval or adaptation results in a better solution, it changes its mind. This approach gives the user more answers to work with and provides opportunities for the user to redirect a CBR system that is on the wrong track.

The final panel on deploying CBR was chaired by Ray Bareiss of the Institute for the Learning Sciences. Participants were Kris Hammond; Alan Meyrowitz, the Navy Artificial Intelligence Center; Hiroaki Kitano, NEC and Carnegie Mellon University; and Evangelos Simoudis, Lockheed AI Center. The questions addressed included the readiness of CBR applications to be deployed, the proper expectations for their performance, and the types of research that are likely to be most productive. Among other points, this discussion highlighted the success of many CBR systems that have already been deployed, especially as advisory systems.

The workshop revealed trends such as a significant ongoing research focus on indexing and retrieval issues, the use of CBR for design and design support, and CBR-based learning environments. It also pointed to case adaptation as the least understood aspect of CBR and, consequently, a particularly rich research area for advancing the state of the art in CBR.

David B. Leake
Indiana University

Reasoning about Function

The Reasoning about Function Workshop attracted a large audience from both academia and industry. Its agenda included a keynote speech by B. Chandrasekaran (Ohio State University), eight presentations, poster displays, a round table discussion, and a panel discussion. This report touches on some of the important issues discussed at the workshop, the conclusions drawn, and their implications.

The explicit representation and use of function of a component, either as intended by its designer or as interpreted by its user, is increasingly being used for leverage during problem solving. Explicit treatment of function has proven to be useful because of its potential to organize and provide access to causal knowledge of the component, the improved resolution it brings to reasoning, and its utility in addressing the scaling problem. The objectives of this workshop were to examine current work in reasoning about function, present a forum to develop a shared framework, and identify trends and future directions for this emerging field.

In spite of a substantial research community working in the field or, rather, perhaps because of it, one of the recurring topics was the nature of function. One of the definitions of function (promoted by Chandrasekaran) is that it is a statement about the intended state of a device, along with initial and background conditions under which the state can be achieved. This definition can be extended to include changes in the state of the environment.

Another interesting definition was brought up by S. M. Kannapan (Xerox Corporation) and M. Lind (Technical University of Denmark), who defined function as a pair of behavioral expressions that map the utilized part of a subsystem behavior to the intended part of the system behavior. This definition emphasizes the context-dependent nature of function; that is, the function of a component (of a device, organization, software, society, and so on) is relevant only with respect to its environment. In addition, the function of a component can change from environment to environment, as with a hammer, which can be used to drive nails in a construction setting or to hold papers down on an office table. However, the behavior of a device is the relation between the input and the output of the device and can be expressed in state, process, or component ontologies. Thus, the behavior of a device is independent of its environment.

A third perspective on function is that function complements behavior. One of the consensus points was that function provides leverage during problem solving because it annotates behavior; that is, it indexes the relevant parts of behavior. This statement implies that function is an abstraction and approximation of behavior. Given that behavior can itself be expressed at several levels of abstraction (for electronic devices there are parametric, logic, and component levels, among others), function is always a level more abstract than behavior. In fact, the function of a component from one perspective can be called behavior of the component from a more abstract perspective. As an example, at the electron level, a *p* channel on an *n* substrate is a device with the behavior of a transistor: By function, it is an amplifier. At the signal level, the behavior of the device is amplification: By function, it is now the preamplifier in an audio amplifier circuit.

This discussion led to the question of whether there can be function without behavior. For example, flowers have the function of making an environment pleasant, although they have little by way of behavior to show for it. This statement is not true because flowers smell and look good, making them aesthetically pleasing. The all-important consideration here is the path(s) of interaction in which behavior manifests. In the case of flowers, the paths of interaction are visual and olfactory. Reminding ourselves that behavior is the relation between input and output, the smell emitted or the color projected by the flower are parts of its behavior. (However, this definition of behavior is not universally accepted.) Again, a device can have different functions with respect to different paths of interaction. Revisiting our example of a hammer, when used to drive nails, its momentum is in the path of interaction, whereas when used as a paperweight, its weight is in the path of interaction.

During a panel discussion on how functional reasoning could work with qualitative reasoning, it was noted that behavior, as used in qualitative reasoning, is an integral part of function description. Function helps focus behavior-based reasoning in the direction of the goal to be achieved. As a word of caution, K. Forbus (Northwestern University) noted from J. deKleer's work that although one can arrive at functional roles of devices from their behavioral envisionments, one might not always be able to start from functional roles and reason about the behavior of the device, especially in domains where the mapping between structure and function is not simple.

As in any other field, representation was voted the issue that merited further attention the most. All aspects of representation need attention. Lind pointed out that representing the relationships between structure, behavior, and function was just as important as representing structure, behavior, and function themselves. J. Malin (NASA, Houston) proposed that a function model should necessarily include descriptions of relations between intents and their effects in the world in terms of the roles of the following intermediate stages: devices, control actions, and affected entities. Further, interactions among these stages must be modeled at any granularity of representation.

Both C. Price (University of Wales, United Kingdom) and D. Allemang (Swiss Federal Institute of Technology) felt that enabling users to build function models is crucial to the success of function-based technology. It is also necessary to build and make available libraries of function components. However, compiling such libraries is not a straightforward task, considering that function models are dependent on the environment where they are used as well as the task for which they are used. Another, more lofty goal for the field would be to be able to automatically build the function model of a device from its design descriptions.

The relative infancy of the field was specially noted by Forbus and J. Sticklen (Michigan State University). Forbus cautioned that function-based research should steer clear of the agenda of earlier, not too successful efforts by the AI community to find a complete theory of causation. Sticklen thought that the issue at the top of the agenda for the field was applications and more applications. Only more experience could bolster the field and that is when all the unresolved differences of opinion would come out in the wash.

Amruth Kumar
State University of New York
at Buffalo

Validation and Verification of Knowledge-Based Systems

Since NCAI-88, each national conference on AI has played host to a workshop on the theme of assuring the quality and reliability of knowledge-based systems (KBSs) through verification and validation tools and methods. This annual forum has been instrumental in fostering the growth of the verification and validation area. Interest and activity in this area continue to grow, as evidenced by the growth in both the number of contributed papers and the number of workshop participants since the 1992 workshop. The 1993 workshop was attended by 35–40 individuals, many of whom participated actively. Furthermore, this workshop met its aim of attracting a greater degree of international participation: Over 40 percent of the registered attendees were from outside the United States.

Roughly half of the workshop was devoted to papers on the core topics of KBS validation and verification tools and methods. In addition, a panel session examined the current state of the practice in KBS validation and verification. The remainder of the workshop focused on two highly topical issues: the use of formal methods in the validation and verification of KBSs and the need to perform validation and verification on *hybrid systems* (that is, those that mix KBS technology with other kinds of AI and non-AI software technology).

The session on testing expert systems offered evaluations of the effectiveness of three different testing strategies: (1) an output, partition-based black-box strategy (Imran Zualkernan, Pennsylvania State University); (2) a random probabilistic strategy (Michel de Rougement, Université de Paris-Sud); and (3) a strategy for assessing structural coverage achieved during a testing session (Cliff Grossner, Concordia University). The trend shown in this session toward performing quantitative evaluations of validation and verification techniques indicates that this field is maturing.

The workshop reflected trends in the larger fields of AI and software engineering. Prompted by the current interest in knowledge sharing and large KBSs, the session on verifying expert systems included two papers on coping with inconsistency when a large knowledge base is assembled by combining several independently consistent knowledge bases (Nicolaas Mars, University of Twente, and Windy Gambetta, University of New South Wales). The need to maintain, comprehend, and reengineer existing KBSs (especially the older rule-based systems) was addressed by papers on software metrics for rule bases (Zhisong Chen, Concordia University) and rule base clustering techniques (Mala Mehrotra, ViGYAN Inc).

The panel on the state of the practice indicated that the adoption of validation and verification tools and methods by KBS developers is proceeding slowly, albeit with some successes: Marc Dahl of Boeing reported highly positive experiences with the KB-REDUCER verification tool, and David Hamilton (IBM) described how a joint National Aeronautics and Space Administration (NASA)–IBM training initiative is assisting KBS developers. Part of the problem in getting validation and verification technology into the field is the limited availability of tools and the lack of a comprehensive methodology that incorporates the use of available tools. Steve Wells (Lloyds Register) described how the ESPRIT VIVA project is attacking this problem by developing an integrated validation and verification method and tool set, and Robert Plant (University of Miami) outlined a methodology that aims to facilitate validation and verification using formal methods by capturing several kinds of metaknowledge needed for validation and verification as a natural part of development.

Three works were motivated by the belief that complex problems in the future will be solvable only by combining a multiplicity of software technologies and that validation and verification of such systems will pose new challenges. Takao Terano (University of Tsukuba) presented experimental results from the validation and verification of a combined case-based and conventional knowledge-based system. Dan O'Leary (University of Southern California) looked at the problem of combining the judgments of the different agents in a multiagent system using statistical techniques. Chris Landauer (Aerospace Corporation) described the *wrapping approach,* which uses KBS techniques to facilitate maintenance and testing of heterogeneous software systems, integrating various kinds of AI and non-AI component.

Without a doubt, the liveliest part of the workshop was the lunch-time debate on the role of formal methods in the validation and verification of KBSs. Rose Gamble (University of Tulsa) opened the debate, reporting positive experiences in using formal methods to verify the correctness of rule-based systems. However, she also noted that determining appropriate formal specifications for KBS is a nontrivial task and that current formal methods are probably too resource intensive for practical use. Chris Landauer also expressed positive experiences in using formal mathematical models in KBS validation and verification, asserting that multiple models are typically required, each addressing a different aspect of the system. Dan O'Leary sounded a cautionary note, arguing that because KBS applications in many domains involve a high degree of human and organizational factors that cannot be formalized, formal methods are unlikely to have a significant impact in such domains. Finally, Nick Sizemore (ARC Inc.) represented the KBS developers' position, arguing that the formal method research community must provide methods that are more flexible and easy to use, or it is unlikely that developers will adopt them. The debate and discussions closed with participants offering suggestions about what must be done to promote the use of formal methods in KBSs (and AI in general). These suggestions included (1) investigate the nature of (formal) requirements for AI software; (2) make formal methods easier to use, especially for nonmathematicians; (3) allow formal methods to incorporate informal or incomplete knowledge; and (4) explore the factors that determine when a specific formal method is or isn't useful.

The workshop closed with a discussion of possible future directions for the validation and verification field, drawing on many of the themes that arose during the day, notably the growing interest in formal methods and hybrid systems.

Alun D. Preece
University of Savoie, France