

CARMEL Versus FLAKEY

A Comparison of Two Winners

Clare Congdon, Marcus Huber, David Kortenkamp, Kurt Konolige, Karen Myers, Alessandro Saffiotti, and Enrique H. Ruspini¹

■ The University of Michigan's CARMEL and SRI International's FLAKEY were the first- and second-place finishers, respectively, at the 1992 Robot Competition sponsored by the American Association for Artificial Intelligence. The two teams used vastly different approaches in the design of their robots. Many of these differences were for technical reasons, although time constraints, financial resources, and long-term research objectives also played a part. This article gives a technical comparison of CARMEL and FLAKEY, focusing on design issues that were not directly reflected in the scoring criteria.

The University of Michigan's CARMEL and SRI International's FLAKEY were the first- and second-place finishers, respectively, at the 1992 Robot Competition sponsored by the American Association for Artificial Intelligence (AAAI) (see the Dean and Bonasso article in this issue). Interestingly, the two teams used vastly different approaches in the design of their robots. Many of these differences were for technical reasons, although time constraints, financial resources, and long-term research objectives also played a part.

The final scores for the robots, based solely on competition-day performance, constitute only a rough evaluation of the merits of the various systems. This article provides a technical comparison of CARMEL and FLAKEY, focusing on design issues that were not directly reflected in the scoring criteria. Space limitations preclude detailed descriptions of the two approaches; further details can be found in an upcoming AAAI Technical Report by the authors.

The Two Robots

CARMEL (computer-aided robotics for maintenance, emergency, and life support) is based on a commercially available Cybermotion K2A mobile robot platform. CARMEL is a cylindrical robot about a meter in diameter, standing a bit less than a meter high. It has a top velocity of 780 millimeters/second and a top turning rate of 120 degrees/second; it moves using three synchronously driven wheels. For sensing, CARMEL has a ring of 24 Polaroid sensors and a single black-and-white charge coupled device camera. The camera is mounted on a rotating table that allows it to turn 360 degrees independently of robot motion. Three computers work cooperatively while the robot is running: First, an IBM PC clone runs a 33-MHz, 80486-based processor that performs all top-level functions and contains a frame grabber for vision processing. Second, a motor-control processor (Z80) controls the robot's wheel speed and direction. Third, an IBM PC XT clone is dedicated to the sonar ring. All processing and power are contained on board CARMEL.

CARMEL's software design is hierarchical in structure. At the top level is a supervising planning system that decides when to call subordinate modules for movement, vision, or the recalibration of the robot's position. Each of the subordinate modules is responsible for doing low-level error handling and must return control to the planner in a set period of time, perhaps reporting failure; the planning module then determines whether to recall the submodule with different parameters or resort to another course of action.



CARMEL avoids obstacles using a point-to-point, goal-directed algorithm called VFH.

CARMEL avoids obstacles using a point-to-point, goal-directed algorithm called VFH (Borenstein and Koren 1991a, 1991b). Object recognition is done using a single camera and a one-pass algorithm to detect horizontally striped, bar-code-like tags on each of the 10 objects. A distance and a heading for each object are returned. Recalibrating the robot's position is done by triangulating from three objects with known locations.

The software system of CARMEL was kept modular to allow for a team design, whereby small groups could work independently on each module. Using this approach, the team of 30 students was able to write its winning system in six months. Only the low-level object-avoidance modules existed before work on the competition began. CARMEL's software system was also kept simple so that it could be run completely on board, allowing CARMEL to navigate at high speeds while it smoothly avoided obstacles. (Many of the other robots in the competition were sending information to off-board processors and, as a result, operated in a jerky, stop-and-go fashion, moving a bit but then having to stop and wait while sensor information was sent off board and processed and the results transmitted back to the robot.)

FLAKEY is a custom-built mobile robot platform approximately 1 meter high and .6 meter in diameter. There are two independently driven wheels, 1 on each side, giving a maximum linear velocity of 500 millimeters/second and a turning velocity of 100 degrees/second. Like CARMEL, FLAKEY has ultrasonic sonar sensors good to about 2 meters, but instead of a uniform ring, FLAKEY has 4 sensors facing front, 4 facing back, and 2 facing each side. Additionally, FLAKEY has 8 touch-sensitive bumpers around the bottom perimeter of the robot and a structured-light sensor that is a combination of a light stripe and a video camera that is capable of providing a dense depth map over a small area in front of FLAKEY. FLAKEY has 3 computers: (1) a Z80 motor and sonar controller, (2) a SUN 3 dedicated to the structured-light sensor, and (3) a SPARCSTATION responsible for high-level routines. During the competition, all computation was performed on board.

FLAKEY's basic software design is distributed: The modules work in parallel and communicate through a blackboardlike structure called the *local perceptual space* (LPS). LPS is a geometric egocentric map of the area within two meters of the robot. Modules contribute information to, and draw information from, LPS. The loosely linked structure makes it

possible to have tasks running in parallel that have different reaction-time and information requirements. On the perception side, modules add raw sonar and structured-light information to LPS, treating it as an occupancy grid. Other interpretive processes use this information to construct and maintain higher-order structures, parsing the data into surface segments, recognizing objects, and so on. All this information is coordinated geometrically so that an action module can use whatever form is appropriate, for example, the occupancy grid for obstacle avoidance, surface segments for path planning, and object tags for task planning.

On the action side, there are three main types of modules. At the lowest level, reactive-action modules called *behaviors* guide the robot's movements. The input to these modules is the occupancy grid for obstacle avoidance plus artifacts (such as a path to follow) that are put into LPS by higher-level navigation routines. FLAKEY was unique in using fuzzy rules as the building block for behaviors (Saffiotti and Ruspini 1993), giving it the ability to react gracefully to the environment by grading the strength of the reaction (for example, turn left) according to the strength of the stimulus (for example, the distance of an obstacle on the right).

More complex behaviors, such as moving to desired locations, use surface information and artifacts to guide the reactive behaviors; they can also add artifacts to LPS as control points for motion. At this level, fuzzy rules allow FLAKEY to blend possibly conflicting aims into one smooth action sequence. At a higher level, the navigation module autonomously updates FLAKEY's global position by comparing it to a tolerant global map, which contains prior, approximate spatial knowledge of objects in the domain. Finally, task-level modules continuously monitor the progress of the complex behaviors, using information from the navigation module to plan sequences of behaviors to achieve a given goal.

The distributed architecture and loosely coupled control structure enable FLAKEY to simultaneously interpret sensory data, react to the local environment, and form long-range plans. The modular and distributed design of FLAKEY means that it is both flexible and extensible. The SRI team incorporated large portions of software previously written for an office environment, including almost all the perceptual routines and the low-level behaviors. The team started working on the competition one month before it began and



The distributed architecture and loosely coupled control structure enable FLAKEY to simultaneously interpret sensory data, react to the local environment, and form long-range plans.

CARMEL was distinguished by its graceful motion around obstacles in open terrain and was pleasant to watch

produced and integrated modules for complex behaviors, tasks, and navigation.

Issues in Moving

In stage 1, both robots had to roam the arena, avoiding people and obstacles. Both robots used sonar sensors as their primary obstacle-avoidance sensors. CARMEL used a two-step process in which the sonar readings were first filtered to reduce noise and imprecision (Borenstein and Koren 1992), and then an occupancy grid-style map, similar to that introduced by Moravec and Elfes (1985), was updated. CARMEL used this sonar map to navigate. Similarly, FLAKEY used LPS to integrate sonar readings and fuzzy-control rules to control motion.

Both robots performed obstacle avoidance remarkably well despite attempts by the judges to surprise and contain them. FLAKEY's use of fuzzy rules resulted in extremely smooth and reliable movement. FLAKEY uses two-part obstacle-avoidance rules: Longer-range rules deflect FLAKEY away from distant obstacles, and collision-avoidance rules force emergency maneuvers when an object is suddenly detected nearby. These rules are typically combined with rules for purposeful motion, such as following a wall. This combination of rules ensures that strategic goals are achieved as much as possible while a high reactivity is maintained. In this phase of the competition, speed was limited to 200 millimeters/second, primarily because of the blind spots on the diagonal: Objects in these positions had to be relatively close before they could be seen by the sonar. FLAKEY's reliable behavior is best summarized by one judge's comment: "only robot I felt I could sit or lie down in front of" (which he actually did).

CARMEL was distinguished by its graceful motion around obstacles in open terrain and was pleasant to watch. It moved at a speed of 300 millimeters/second, noticeably faster than FLAKEY. However, under prodding from the judges, CARMEL touched two obstacles and grazed a judge. CARMEL touched objects in part because many variables in CARMEL's obstacle-avoidance code need to be tuned for the environment in which it is running. The Michigan team had assumed an environment with dynamic but much more benign obstacles.

FLAKEY placed ahead of CARMEL in this stage of the competition and was only one point behind the first-place (at this point) entry, TJ2 from IBM. Part of the reason why CARMEL

did not do as well was because it was so goal oriented; that is, it was always trying to get somewhere in particular. CARMEL could not be "shepherded" about by the judges because it had a dogged persistence in trying to achieve its goal location. Both teams noticed that behavior could be improved markedly by tuning the parameters of the avoidance routines.

Issues in Object Recognition

Stages 2 and 3 both required the ability to detect and visit objects (specifically, poles of a fixed diameter) scattered throughout the arena. The rules permitted teams to modify poles to facilitate the recognition process, although a small bonus was awarded for full autonomy, that is, no altering of the environment. Michigan took advantage of the object-modification rule by attaching a distinct omnidirectional bar-code tag to each pole. CARMEL's vision algorithm was designed to extract the bar codes from an image.

SRI, however, was one of only two teams (the other being Brown University) that did not modify the arena or poles in any way. Instead, FLAKEY used only the physical characteristics of the poles themselves in the detection process. FLAKEY used a two-tiered approach, whereby sonar input was monitored during navigation to detect candidate poles, and candidates were actively verified by having FLAKEY navigate to a position where the structured-light sensor could be applied. This hybrid approach was necessary because of the limitations of the two sensing modalities: Structured-light verification is highly accurate but applies only to a small perceptual space (less than two meters) directly in front of the robot; sonar input covers a much larger space during navigation but is not nearly as reliable for object recognition. Both the structured-light and sonar routines were built using low-level perceptual routines that FLAKEY has used for some time.

The recognition components of both teams performed extremely well during the competition. CARMEL never saw a false object, and it never missed seeing an actual object; similarly, FLAKEY's structured-light routine was perfectly reliable. CARMEL's performance was surprising because its long-range vision created the added difficulty of dealing with false objects outside the arena, a problem that FLAKEY's short-range sensors did not have. FLAKEY's candidate generation techniques based on sonar input also performed well, picking out only two nonpoles (box corners)

as candidates and only failing to detect one pole in its perceptual space as a candidate (because the robot passed too close to the pole).

The SRI team demonstrated that reliable object-type recognition was possible using only physical characteristics of the objects and simple domain constraints (such as non-proximity to other objects) without having to modify the environment. As such, FLAKEY, in contrast to CARMEL, was able to perform recognition for classes of objects rather than specially marked individuals in the class. One consequence of doing class recognition was that individual objects had to be identified based solely on information about the object's location. In contrast, the individualized bar codes used by the Michigan team provided immediate identification information to CARMEL.

CARMEL's use of long-range sensing made it possible to locate objects from as far away as 12 meters (over half the diameter of the arena). In contrast, FLAKEY could only recognize poles and candidates within its local perceptual space. As discussed later, this difference had a major impact on the methods used by the two teams for mapping and navigation.

Issues in Mapping

To be competitive in stage 3, it was necessary for the robots to generate maps of the environment during stage 2. At a minimum, these maps contain the location of discovered poles, but they could also encode further information, such as the positions of obstacles or walls. A complementary problem to map construction is *self-localization*, which involves having the robot determine where it is relative to the map. A critical issue faced by both robots in solving these problems was the inaccuracies inherent in *dead reckoning*, the robot's internal calculation of its location based on wheel movements.

Automated map generation remains a topic of current research for the field of robotics. Michigan and SRI chose two different approaches to the design of maps for their robots. CARMEL used a global Cartesian system that stores only pole locations and the current position of the robot. To track its position with reference to the map, CARMEL relied exclusively on dead reckoning: When initialized, it was given its position and orientation on the map, and subsequent movements gave an estimated position based on wheel rotation. When discovered, the poles were placed

on the map using the current estimated position together with the range and angle returned by the vision system.

Of course, errors in estimated position accumulate over time from wheel slippage and the like; CARMEL incorporated an algorithm to triangulate its position from known object locations, thus reducing the error. The vision-based triangulation system was not actually used for the competition because of last-minute changes to the system software. However, not using triangulation did not noticeably affect the performance of CARMEL for three reasons: First, the time and the distance between tasks were small; second, CARMEL's dead reckoning and its vision system were highly accurate; and, finally, the planning system was designed to deal with self-localization errors. CARMEL could be several meters away from the expected location of the pole and still be able to locate it.

In contrast to CARMEL, FLAKEY used a tolerant global map containing local Cartesian patches related by approximate metric information. Each patch contains some recognizable feature or landmark by which the robot can orient itself; the approach is similar to the work on landmark-based navigation (Kuipers 1978). The patches chosen for the competition were the walls of the arena because they were the most stable features for navigation. The approximate length and the relative orientation of the walls were given to FLAKEY as prior knowledge; FLAKEY could easily have learned this information by circumnavigating the arena.

The SRI team chose the tolerant global maps because FLAKEY accumulated dead-reckoning errors more quickly than CARMEL. Moving four or five meters, especially with some turning, can cause significant errors in estimated position, and FLAKEY must use sensed landmarks to correct its localization on the map. Compounding the problem is FLAKEY's short-range sensing, which makes it impossible to locate landmarks more than a few meters away. The tolerant global maps are a solution to FLAKEY's imprecision in large-scale sensing and movement. Within each patch, local landmarks can be sensed almost continuously (in this case, the arena walls and wall junctions) to keep localized. When going between patches, approximate metric information can be used to find the next landmark for localization. Because sensing and movement are accurate only over small distances, there is no need to keep a highly precise global geometry; further, FLAKEY would find it impossible to use this information.

*Fuzzy rules
allow FLAKEY
to blend
possibly
conflicting
aims into
one
smooth
action
sequence*

*The
recognition
components
of both teams
performed
extremely well*

The use of precise dead reckoning and long-range sensing gave CARMEL a marked advantage over FLAKEY in the competition because it made it easy to both register the poles in a global coordinate system and determine trajectories for navigating from one to another (as required for stage 3). FLAKEY's ability to rerecognize poles that it discovered previously demonstrated that the tolerant global map can successfully be used for self-localization, although FLAKEY encountered some difficulties in using this system (see the next section).

It is interesting to speculate about how well the different methods would work in other domains. FLAKEY's tolerant global maps were designed for an office environment, where navigation landmarks are plentiful (walls, corridors, doors), and long-range triangulation is difficult and of limited value. The tolerant global maps are robust in this situation, whereas a precise global Cartesian map would be hard to acquire and use. Its main advantage—navigation over open space—would be minimized because most navigation is by corridor paths.

However, FLAKEY's approach is less useful in more open areas such as outdoor navigation, where paths and local landmarks might be sparse. In this case, CARMEL benefits from the inclusion of more global positioning information.

Issues in Navigation

Task-oriented navigation played a critical role in the competition. Stage 2 required explorative navigation of the arena to detect and visit poles. Because the robots had no prior information about object locations, a general and thorough exploration methodology was required. Stage 3 involved directed navigation: Robots were to revisit three poles in a prespecified sequence and then return home.

Explorative Navigation

CARMEL's long-range object-recognition capabilities enabled the Michigan team to use a fairly simple exploration strategy. CARMEL's exploration consisted of moving to viewing positions distributed throughout the arena, executing a visual sweep for objects, and then visiting each object.

FLAKEY's reliance on local sensing necessitated an actual physical exploration of the environment: To ensure full coverage, FLAKEY had to cover the full extent of the competition arena with its local perception. The strategy adopted by the SRI team was to traverse the

perimeter of the arena, making forays into the center of the arena at certain points along the way. This strategy was designed to reconcile the conflicting objectives of providing broad coverage of the arena and keeping FLAKEY self-localized using information about wall locations.

FLAKEY did encounter some localization problems near the end of its stage 2 run, primarily because of a tactical mistake (on the part of the designers!) in the execution of forays. FLAKEY initiated its final foray before having registered the current wall. As a result, dead-reckoning errors accumulated to such an extent that FLAKEY's beliefs about its position were inaccurate. Given more time, FLAKEY would eventually have returned to the wall and reregistered itself, thus correcting the problem. The entire issue could have been avoided had forays been postponed until wall registration had taken place.

The Michigan team's use of long-range sensing easily enabled CARMEL to find all 10 poles within the allotted 20-minute search period. In contrast, the physical exploration executed by FLAKEY was time consuming. In the end, FLAKEY found and correctly registered only 8 of the 10 poles before time expired. Certainly, the Michigan approach was superior given the conditions of the competition environment. In particular, Michigan took full advantage of the fact that objects would be visible above all obstacles. Because FLAKEY's method was not based on any such assumptions, it was less efficient; however, FLAKEY's exploration method could be used in more realistic environments, where objects can be occluded.

Directed Navigation

In stage 3 of the competition, the robots were given three poles to visit in order, and then, they were to return to a home position. This stage was timed, with the robots receiving points based on their time with respect to the other robots.

FLAKEY's strategy of registering objects and itself with respect to walls meant that the robot had to navigate along the perimeter of the arena when traveling between objects. Visiting an object registered with respect to a wall W involved determining the direction of the shortest perimeter path to W (either clockwise or counterclockwise), following the perimeter in this direction until W was encountered, and then using dead reckoning within the coordinate system of W to move to the pole. CARMEL, however, used dead reckoning with its global map to proceed directly

to the recorded locations of objects. Not surprisingly, CARMEL was able to perform the stage-3 visiting task in a much shorter period of time (3 minutes versus 11 minutes for FLAKEY).

Like many other teams, the Michigan team marked the home position by placing an 11th pole there. This modification to the environment was made to provide a perceptual landmark for the home position, which compensated for the accumulation of dead-reckoning errors during the run. FLAKEY, in contrast, did not require any such modification to the environment. Instead, it was able to treat the home location in the same manner as other positions of interest (such as pole locations or foray positions) because it used the tolerant global maps to continuously correct its position.

Conclusion

Both CARMEL and FLAKEY must be considered unqualified successes, having bested 10 or so other entries in a nationwide competition. There were many reasons for this success. Both teams did all their processing on board the robot, avoiding problems with radio and video links and enabling their robots to be more reactive. Both teams inherited robots that had well-developed software systems. In addition, both teams used simulations to speed the development process. However, the approach to the competition was different for each team. Michigan looked at the competition rules and engineered a system to optimize its robot's performance at the cost of generality. SRI used the competition as a demonstration of the application of its research in a new domain, without engineering any hardware specific for this domain.

Interestingly, neither team used any geometric planning for navigation around obstacles to a goal point, although this area is a large part of robotics research (Latombe, Lazanas, and Shekhar 1991). Instead, both teams relied on the simple strategy of heading toward the goal and using reactive behavior to avoid obstacles, with simple methods for getting out of cul-de-sac situations. Geometric planning requires some sophistication in perception and mapping of obstacles and can be difficult to perform in real time. The large openings around obstacles in the competition made it easy to pursue simpler strategies, and we speculate that in other domains, geometric planning will also play a minor role in navigation.

It is interesting to try to compare the two

system architectures. At the level of reactive movement, FLAKEY perhaps had the advantage, because the fuzzy-control paradigm provides a flexible and powerful representation for specifying behavior. In less than a month, the SRI team was able to write and debug half a dozen complex movement routines that integrated perception and action in the service of multiple simultaneous goals.

In terms of overall design, it is difficult to compare the relative merit of the two architectures because the approaches to solving the problem were so different. FLAKEY's distributed control scheme allows various modules to run in parallel, so that (for example) self-localization with respect to landmarks occurs continuously as FLAKEY moves toward a goal location or searches for poles. However, the distributed design leads to behavior that is more difficult to predict and debug than that of CARMEL's top-down approach in which all the perception and goal actions are under sequential, hierarchical control.

Although Michigan was the winner of the competition, it is not clear that its system can easily be extended to other domains. Certainly, the obstacle-avoidance routines are necessary in any domain and are widely applicable. CARMEL's reliance on a global coordinate system and tagged objects restricts it to engineered environments that can accurately be surveyed (a reasonable assumption when you consider how much of the world in which humans operate is highly engineered). Also, CARMEL's simple exploration strategies would be naive in an environment where objects can be occluded. CARMEL's keys to victory were fast, graceful obstacle avoidance and fast, accurate vision algorithms, not cognitive smarts.

FLAKEY, moving more slowly and possessing less accurate and more local sensing, had to rely on a smart exploration strategy and constant position correction. One of the key research ideas behind FLAKEY is that natural (that is, non-engineered) landmarks are sufficient if the right map representation is used, and it was gratifying to see this approach work in a new environment. Still, FLAKEY could be more efficient in navigating open spaces if it incorporated more global geometric information, such as CARMEL used.

The fact that CARMEL, which is sensor rich and cognitively poor and FLAKEY, which is sensor poor and cognitively rich, came in as the top two robots in the competition clearly shows that fundamental trade-offs can be made in engineering mobile robots. Complex sensing can allow for simple planning; simple

sensing requires complex planning. In no sense is either robot more complex than the other; it is just that the complexity lies in different places. What was not clear from the competition was whether complex sensing and complex planning will make for a fundamentally better robot. This issue will have to be resolved at future competitions.

Acknowledgments

Support for several of the Michigan authors was from Defense Advanced Research Projects Agency grant DAAE-07-92-C-R012. Purchase of CARMEL and its computers, development of the EERUF (error elimination rapid ultrasonic firing) and VFH (vector field histogram) algorithms, and support of several Michigan authors were through Department of Energy grant DE-FG02-86NE37969. Additional financial support for the Michigan team was from The University of Michigan College of Engineering, The University of Michigan Rackham Graduate School, ABB/Graco Robotics Inc., and the American Association for Artificial Intelligence.

Support for Kurt Konolige and Karen Myers came partially from Office of Naval Research contract N00014-89-C-0095. Alessandro Saffiotti is supported by a grant from the Italian National Research Council. We gratefully acknowledge the help of Enrique Ruspini and Daniela Musto in suggesting ideas for self-localization and goal-directed movement.

Note

1. Clare Congdon, Marcus Huber, and David Kortenkamp are with the AI Laboratory at the University of Michigan, and Kurt Konolige, Karen Myers, Alessandro Saffiotti, and Enrique Ruspini are with SRI International.

References

- Borenstein, J., and Koren, Y. 1992. Noise Rejection for Ultrasonic Sensors in Mobile Robot Applications. In Proceedings of the IEEE Conference on Robotics and Automation, 1727–1732. Washington, D.C.: IEEE Computer Society.
- Borenstein, J., and Koren, Y. 1991a. Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance. *IEEE Transactions on Robotics and Automation* 7(4): 535–539.
- Borenstein, J., and Koren, Y. 1991b. The Vector Field Histogram for Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation* 7(3): 278–288.
- Kuipers, B. 1978. Modeling Spatial Knowledge. *Cognitive Science* 2:129–153.
- Latombe, J.-C.; Lazanas, A.; and Shekhar, S. 1991. Robot Motion Planning with Uncertainty in Con-

trol and Sensing. *Artificial Intelligence* 52(1): 1–47.

Moravec, H., and Elfes, A. 1985. High-Resolution Sonar Maps from Wide-Angle Sonar. In Proceedings of the IEEE Conference on Robotics and Automation, 19–24. Washington, D.C.: IEEE Computer Society.

Saffiotti, A., and Ruspini, E. H. 1993. Blending Reactivity and Goal Directedness in a Fuzzy Controller. In Proceedings of the Second International IEEE Conference on Fuzzy Systems. Washington, D.C.: IEEE Computer Society. Forthcoming.



Clare Bates Congdon is a doctoral candidate in the AI Laboratory of the Electrical Engineering and Computer Science Department at the University of Michigan. She holds a B.A. in mathematics from Wesleyan University and an M.S. in computer science from the University of Michigan. Her thesis work is a comparison of genetic algorithms and other machine learning approaches in a complex domain characterized by nonlinear interactions among features (the genetic and biochemical traits that predict coronary artery disease).



Marcus Huber is a research assistant at the University of Michigan AI Laboratory. He graduated with a B.S. in electrical engineering from GMI Engineering and Management Institute in 1988 and received his M.S. in computer science from the University of Michigan in 1991. His research interests include distributed AI, plan recognition, and autonomous robotics. He is a member of the Institute of Electrical and Electronics Engineers, the Association of Computing Machinery, and the American Association for Artificial Intelligence.



David Kortenkamp is a Ph.D. candidate at the University of Michigan AI Laboratory. He graduated with a B.S. in computer science from the University of Minnesota in 1988 and received his M.S. in computer science from the University of Michigan in 1990. His thesis research involves applying principles of cognitive mapping to mobile robot navigation tasks.



Kurt Konolige is a senior computer scientist in the AI Center at SRI International. He received his Ph.D. in computer science from Stanford University in 1984; his thesis, "A Deduction Model of Belief and Its Logics," developed a model of belief based on the resource-bounded inferential capabilities of agents. His research interests are broadly based on issues of commonsense reasoning, including introspective reasoning, defeasible reasoning, and reasoning about cognitive state. More recently, he has

AAAI-93 Robot Exhibition

July 13–15, 1993
Washington, DC



Following the highly successful robotics exhibition at AAAI-92, AAAI is planning to hold a robot competition at the national conference in Washington D.C. in July of 1993

Last year's competition was a three-stage event in which mobile robots demonstrated skills of reactivity, exploration, and directed search (a detailed description is in the Summer 1992 issue of *AI Magazine*).

Mobile robotics is an area where much of the research in diverse AI areas can be effectively and creatively combined to give interesting results. At AAAI-93, we would like to extend the competition to highlight as wide a range of robotic research as possible, and to stress the "intelligent" aspects of their behavior. In

addition to mobile robots, we are also considering having a competition among robotic manipulators, either stationary or attached to mobile platforms.

If you are interested in more detailed information about the competition, please contact:

Kurt Konolige, Artificial Intelligence Center
SRI International, 333 Ravenswood Avenue
Menlo Park, CA 94025, (konolige@ai.sri.com)

or

Reid Simmons, School of Computer Science
Carnegie Mellon University,
5000 Forbes Avenue
Pittsburgh, PA 15213,
(reids@cs.cmu.edu)

been working on making moving robots more intelligent.



Karen Myers is a computer scientist in the AI Center at SRI International. She completed her Ph.D. in computer science at Stanford University in 1991. Her research interests include the acquisition and use of maps by mobile robots, reactive planning, and hybrid reasoning systems.



Alessandro Saffiotti is a research assistant at the Université Libre de Bruxelles, Belgium. He received his Ph.D. in computer science from the University of Pisa, Italy, in 1988. Since January 1992, he has been at SRI International as an international

fellow, working on the use of multivalued logics in the control of autonomous mobile robots. His research interests include the formalization of partial belief and uncertain knowledge and their use in reasoning, planning, and acting.



Enrique H. Ruspini is a senior computer scientist with the AI Center of SRI International. He received his doctoral degree from the University of California at Los Angeles in 1977. He has done extensive research on fuzzy sets and their applications, having pioneered the application to classification problems in 1969.