NRC Publications Archive
Archives des publications du CNRC

Using a Blackboard to Integrate Multiple Activities and Achieve Strategic Reasoning for Mobile-Robot Navigation

Liscano, R.; Fayek, R.; Manz, A.; Stuck, E.; Tigli, J.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

Publisher's version / Version de l'éditeur:

IEEE Expert, 10, 2, 1995

NRC Publications Record / Notice d'Archives des publications de CNRC:
https://nrc-publications.canada.ca/eng/view/object/?id=8b3a59ab-c1de-449e-829d-97bf7a181781
https://publications-cnrc.canada.ca/fra/voir/objet/?id=8b3a59ab-c1de-449e-829d-97bf7a181781

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at
https://nrc-publications.canada.ca/eng/copyright
READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site
https://publications-cnrc.canada.ca/fra/droits
LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

National Research Council Canada | Conseil national de recherches Canada

Using a Blackboard to Integrate Multiple Activities and Achieve Strategic Reasoning for Mobile-Robot Navigation

Ramiro Liscano, Allan Manz, and Elizabeth R. Stuck, National Research Council
Reda E. Fayek, University of Waterloo
Jean-Yves Tigli, Université de Nice

THE ACTIVITY-BASED BLACKBOARD SYSTEM CONSISTS OF TWO HIERARCHICAL LAYERS FOR STRATEGIC AND REACTIVE REASONING: A BLACKBOARD DATABASE TO KEEP TRACK OF THE STATE OF THE WORLD AND A SET OF ACTIVITIES TO PERFORM REAL-TIME NAVIGATION.

DEVELOPING A MOBILE ROBOT gives us the opportunity to investigate issues in the design of intelligent systems because the robot's mobility forces us to deal with many unpredictable environmental situations. One dictionary definition of intelligence is the ability to deal with new or trying situations. Thus, a mobile robot that reliably navigates in unknown environments gives the appearance of intelligent behavior. The main idea of an autonomous vehicle is quite simple: Given a task to perform, it must have the ability to perceive the environment and act appropriately. This ability requires a feedback control system to link the vehicle's sensing and control. Unfortunately, autonomous robots have characteristics not yet satisfactorily addressed by the classical control community:

- Solving the problems encountered by the mobile robot generally requires the integration of several methodologies.
- The robot's decision space is discrete and composed of distinct elements as opposed to continuous functions.
- The system must react to the environment in an appropriate time period.
- Due to the limitations of the sensors and sensory processing, most of the knowledge the robot acquires is either incomplete or uncertain.

To address these problems, we have designed a mobile-robot system architecture that uses a blackboard to coordinate and integrate several real-time activities. An activity is an organizational unit, or module, designed to perform a specific function, such as traversing a hallway, going down steps, crossing over an open channel on the floor, or tracking a landmark. An activity resembles a behavior in that it controls the robot to perform a specific task. It differs from a behavior in that it is designed to perform the specific task in a narrow application domain, whereas a behavior generally resembles a biological response—that is, an organism's response to a stimulus.

Payton[1] defined the term activity as an instance of an activation set, where an activation set is composed of a number of behaviors. Payton's activities are a way to specify a combination of behaviors to achieve a more complex behavioral pattern. In contrast, we make no attempt to define our activities as a combination of basic behaviors. In our system, several activities are necessary for the robot to perform simple tasks such as moving around a factory bay. Some of these basic navigation activities are traversing open space, crossing over floor anomalies (cables or channels), and avoiding collisions.

The system architecture must define a mechanism to coordinate the mobile robot's activities since they cannot all drive the robot simultaneously. Most mobile-robot control systems are hybrid systems combining approaches from hierarchical, behavioral, and blackboard-based systems. Be-

0885-9000/95/$4.00 © 1995 IEEE

havior-based systems have recently become more prevalent for controlling mobile robots.

Hierarchical architectures, such as Nasrem (NASA/NIST Standard Functional Architecture for Telerobot Control System Architecture)[3,4] and IMAS (Intelligent Mobile Autonomous System),[5] offer a nice paradigm for breaking down a global task into subtasks, but the hierarchical structure quickly degrades into a hierarchical command structure combined with distributed sensing similar to that implemented on Mobot III.[6] Most of these systems use *sense-think-act* cycles that are difficult to implement in real time when the robot must deal with diverse sensing conditions. Lumia[4] has demonstrated the use of Nasrem-style architecture for real-time tracking and catching a ball falling through a maze of pegs, but this system has a well-defined, narrow scope of operation and does not require the sensing diversity that a mobile robot needs.

In an attempt to add reflexive ability to a hierarchical system, Payton[1] proposed a vertical decomposition along with the more classical horizontal, hierarchical decomposition, resulting in a hierarchical structure composed of reflexive behaviors at the lowest level. The effect is a hierarchical system capable of managing diverse sensing conditions—and therefore a more robust system. Arkin[7] also emphasized the importance of a nonhierarchical broadcast of information. Arkin chose to design a system using agents that manipulate a unified representation of the world based on potential fields. Our system does not use a unified representation to coordinate activities but relies on procedural knowledge and sensory data posted on the blackboard to make a decision.

Similar to the Codger[8] system architecture used to drive CMU's Navlab,[9] our system does not use its blackboard as a problem-solving mechanism but primarily as the supervisor and coordinator of several real-time activities. These activities continually post their current state and the current state of the environment to the blackboard. The perception and sensing components of an activity are designed to run concurrently. Not all communications between an activity's modules go through the blackboard, thus reducing communications bandwidth and making reactive behavior possible. Like a traditional blackboard system, our system uses a central database to store information accessible by a number of modules, but it differs functionally because the modules do not work cooperatively to solve a common problem.

In our system architecture, the blackboard's rule set and knowledge determine which activity controls the mobile robot's actuators. A production system facilitates experimental determination of adequate conditions for selecting the activity controlling the vehicle, and the blackboard database serves as a repository of state data and sensory data from the activities. Strategic reasoning is the ability to process sensory inputs, stored information, and long-term goals so that the robot can make decisions with a global view of the environment. The decision about which activity will control the vehicle is based on sensory and state information from the activities and therefore is a form of strategic reasoning.

OUR SYSTEM DOES NOT USE A UNIFIED REPRESENTATION TO COORDINATE ACTIVITIES BUT RELIES ON PROCEDURAL KNOWLEDGE AND SENSORY DATA POSTED ON THE BLACKBOARD TO MAKE A DECISION.

Mobile-robot system architectures

The question of an adequate system architecture for an autonomous vehicle is an ongoing research problem that can be resolved only after numerous systems have been designed and tested. This research is still in its infancy, so there is no common agreement on which system architecture is most suitable for controlling a mobile robot. Almost all existing systems are continually undergoing changes as the mobile robots are fitted with more capabilities. Although many research institutes and several companies are pursuing this research, their goals are not all the same. It is important to differentiate between research into using mobile robots as tools and research into the design of autonomous mobile robots. A system intended to perform a single function as a tool is designed as a conventional control system comprising a single planning module. An autonomous mobile robot, on the other hand, should function in different environments by adopting different strategies and mechanisms for solving problems. Thus, we are concerned with designing a system architecture capable of integrating several functions necessary for reliable operation.

Although research into mobile robotics has yet to reach maturity, we can list a set of desired attributes of a mobile-robot control system. We have divided these attributes into two groups: attributes describing the desired behavior of the mobile robot, and attributes describing a successful design of the control architecture. Some of these have been listed previously by Brooks[10] and have been used as justification for other system architectures, but no formal approach has been adopted by the robotics community.

Behavior attributes. A behavior is a mobile robot's response to a stimulus from the environment. We have identified six behavior attributes important to a mobile robot:

- *Reactivity*: Because the real-world environment is unstructured, the mobile robot can make few assumptions about its dynamics. It must react to sudden changes in the environment within a specified time frame. Reactive behavior is generally a well-defined action in response to a narrow domain of sensing, which gives the robot a very limited scope of understanding. In our system all activities are examples of reactive behavior.
- *Intelligence*: By intelligence we mean the ability to cope with new or trying situations. A robot's ability to manage diverse situations and manipulate the environment to achieve a goal gives the appearance of intelligence. Although true intelligence is yet to be understood, all mobile-robot projects strive to achieve some intelligent behavior.
- *Centralized global reasoning*: A global, high-level decision-making module is crucial for the robot to understand its overall situation. This is particularly important when many independent activities, each with a narrow understanding of the situation, are trying to control the system. A reasoning agent with a global view

can act as an arbiter among the activities to improve the system's performance. A distinguishing feature of the system is whether the global reasoning comes from a centralized, uniform representation or from cooperation among distributed modules. As an example of the use of global reasoning, consider a mobile robot traversing a narrow hallway and discovering that an object has blocked its way. Most reactive navigation algorithms would have the robot stop, turn back, or pace from side to side waiting for an opening. With a global understanding of the situation, the robot could try to find another path or push the object out of the way. All the alternatives require knowledge from other sources integrated into a unified representation.

- *Multiple-goal resolution*: For mobile robots, situations requiring conflicting concurrent actions are inevitable, and the system should provide the means to fulfill multiple objectives. An example is the possible conflict between touching an object and avoiding a collision with the object.
- *Robustness*: A system's robustness is its ability to handle imperfect inputs, unexpected events, uncertainties, and sudden malfunctions. This is a crucial characteristic of a system operating in a real-world environment.
- *Reliability*: A system's reliability is measured by its ability to operate without failures or performance degradation over a certain period of time. The navigation system should maintain a constant level of competence and perform similarly in a number of different environments. For example, we would expect that a collision avoidance algorithm would operate reliably in several environments and that any performance degradation could be corrected without altering the system dramatically. In most cases, misinterpreted sensor data causes performance degradation, which we can correct by either adding more sensors or improving the sensor-processing algorithms.

Design attributes. Design attributes are requirements of the mobile-robot system architecture. Except for the fifth attribute in the following list, they resemble the requirements of conventional software engineering. Unlike most conventional software engineering projects, which begin from a well-specified set of requirements, a mobile-robot project must constantly reevaluate its requirements. Therefore, systems that can be modified easily are highly desirable. We have identified the following design attributes:

- *Modularity*: Following a general requirement of most complex systems, the control architecture of autonomous vehicles should be divided into smaller subsystems that can be designed, implemented, and debugged separately. Modularity is also crucial to incremental design, maintenance, and failure detection and correction.
- *Flexibility*: Experimental robotics, in general, and sensor-based intelligent control, in particular, require continuous design changes during the implementation phase. Flexible control structures allow the design to be guided by the demonstrated success or failure of individual design elements. Modular design is a first step toward flexibility in software design.
- *Expandability*: Because it takes a long time to design, build, and test the individual components of the control system, an expandable architecture is desirable. An expandable architecture facilitates incremental implementation of the many skills the robot needs to cope with diverse situations.
- *Adaptability*: Each activity is governed by a control strategy. Since the state of the world is unpredictable and rapidly changing, the control system must switch smoothly and rapidly between control strategies to adapt to the current situation. Adaptability makes it possible for the mobile robot to reconfigure its mode of operation to perform other functions and operate in different environments.
- *Multisensor integration*: This ability, more important in real-time mobile-robot control systems than in conventional software, is crucial to reliable robot behavior. The system architecture must compensate for the limited accuracy, reliability, and applicability of individual sensors by integrating several complementary sensors.

UNLIKE MOST CONVENTIONAL SOFTWARE ENGINEERING PROJECTS, WHICH BEGIN FROM A WELL-SPECIFIED SET OF REQUIREMENTS, A MOBILE-ROBOT PROJECT MUST CONSTANTLY REEVALUATE ITS REQUIREMENTS.

An autonomous system's three main functions are perception, reasoning, and action. These functions comprise basic control loops that drive the robot to perform specific tasks. The greatest difficulty in the design of the control loops is building reliable perception mechanisms. The types of processes used to accomplish these functions and the dataflow organization help determine the robot's behavior.

The classical systems. With the behavior and design requirements just discussed as criteria, we analyzed the traditional architectures—hierarchical, behavioral, and blackboard—to assess their suitability for controlling the navigation of an autonomous mobile robot. By hierarchical we mean systems similar to the Nasrem style,[3] by behavioral we primarily mean the subsumption architecture,[11] and by blackboard we mean a system similar to BB1.[12]

Table 1 shows the relative strengths and weaknesses of the three architectures. We see that none of them can be used alone for the problem at hand. Only a combination can meet our behavior and design specifications.

Hierarchical systems. Hierarchical architectures have been used for autonomous and semiautonomous systems for many years. Their common distinctive feature is that the control problem is divided along functional lines into progressive levels of data abstraction and in some cases into levels of processing time. The primary design consideration is that the interfaces between levels must be well defined and the cycle times in a level must be well known. The architecture supports intelligent behavior primarily through sequential composition of sensory data and decomposition of commands.

As we suggested earlier, the primary drawback of classical hierarchical structures, particularly those strictly decomposed in the horizontal direction, is their sluggishness in reacting to rapid changes in the environment because the sensory data must pass through several layers in the hierarchy.

Moreover, hierarchical systems are not very robust because of their sequential processing. The tight interdependence of successive modules means the failure of any component results in a complete system breakdown.

In terms of design requirements, hierarchical systems are easily divided into functional modules. But once the hierarchy has been defined, the flexibility to move these modules around is reduced, and reconfiguring the structure to adapt to different operation modes is almost impossible.

Behavioral systems. Behavioral systems consist of a set of sensing, reasoning, and action loops, which operate concurrently to drive the mobile robot. Their dominant characteristics are their decomposition of the control problem into the robot's behaviors and their direct control of the robot's actuators. The system combines a number of robust and reactive behaviors that do not require complicated world models (if any at all) to achieve some goal.

There is no standard set of behaviors accepted by the robotics community. Behaviors range from simple motions to more complicated docking actions. Behavior-based systems' low-level amalgamation of sensing and action makes them perform reactively and reliably. Since each sensor has some control over the action, these systems are very robust to failures in sensory and actuator components. Since all behaviors can drive the robot's actuators, one behavior's failure removes that particular ability from the robot but has a minimal effect on the other behaviors' performance. Another distinctive feature of behavioral systems is the absence of a central intelligent-reasoning agent; the robot's perceived intelligence stems from the interaction of behaviors.

The well-known subsumption method developed by Brooks[11] combines several behaviors, with higher layers of behaviors subsuming lower ones. Ironically, this leads to a hierarchical structure with a great deal of interdependence of behaviors, breaking down the system's design flexibility and modularity. Although the various behavior levels support multiple types of operations, the behavior interdependence makes reconfiguring the system to adapt to different tasks very difficult. The system's strict arbitration mechanism does not provide for the handling of multiple goals simultaneously.

Table 1. Comparison of classical control structures.

	HIERARCHICAL	BEHAVIORAL	BLACKBOARD
Reactivity	Medium	High	Medium
Intelligence	Sequential	Emergent	Distributed
Global reasoning	Yes	No	No
Multiple goals	Yes	Difficult	Yes
Robustness	Low	High	Medium
Reliability	Low	High	Medium
Modularity	Yes	No	Yes
Flexibility	No	No	Yes
Expandability	Yes	Yes	Yes
Adaptability	No	No	Yes
Sensor integration	Difficult	Yes	Yes

In a subsumption system, the priority levels of activities are determined by the structure of the architecture and are not easily modified. One reason for using a blackboard system for vehicle control is that its production rules give us flexibility to investigate what an activity's priority level should be.

Blackboard systems. The most appealing features of blackboard systems are their flexibility in handling the control problem and their suitability for the integration of sensory data. Blackboard systems use distributed expert modules for sensing, action, and reasoning. Each of these agents has its own inference mechanism and local knowledge to perform a specific task and thus is not hampered by the time and effort of extracting information from a global database. Blackboard systems facilitate highly parallel design approaches, allowing the testing of each individual expert module.

In the case of BB1,[12] information transfer is performed through the blackboard and managed through the coordinator. As a result, all state information, no matter how trivial, is kept on the blackboard so that all such data is accessible to all knowledge sources. This generally leads to a system well suited for task planning and capable of managing several distributed processes but with little or no capability for reactive behavior. For that reason, most mobile-robot projects using blackboard systems[9,13,14] have modified the blackboards to manage the real-time issues peculiar to this field.

An activity-based blackboard architecture

With the ambition of building a system with reliable behaviors and a centralized arbitrator, we have developed a system composed of activities arbitrated by a blackboard system controller. The system design separates strategic reasoning from the reactive responses inherent in activities and produces a more modular structure than the subsumption architecture. Our main system design principle is to switch the vehicle's focus of control from one activity to another and to equip those activities with certain decision capabilities.

Figure 1 shows the overall system architecture. This structure implements two levels of reasoning, one through the activities and the other through the blackboard. The thick arrow running from an action module to the environment represents a physical interaction with the environment, primarily by the motion of the robot. The thick arrow running from the environment to the sensor represents the gathering of information about

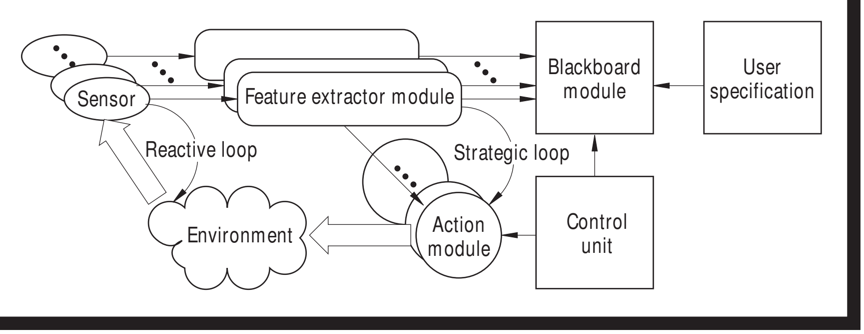

Figure 1. The activity-based blackboard control scheme for a mobile robot.

the environment by the sensor. The system combines a central controller ensuring strategic reasoning and several activities in charge of low-level behaviors. The activities run concurrently, but only one activity action module directly commands the robot's actuators. The blackboard control unit determines the activity that controls the robot, using its knowledge of the state of activities and user's specifications. The control unit can also influence the action module by specifying constraints on its actions. For example, the passageway activity, described in more detail later, is designed to avoid collisions with obstacles. The control unit can constrain it to try to maintain a certain direction of motion and a maximum speed.

An activity is an organizational unit that uses a basic skill to perform a specific function—for example, following a person, passing through a door, avoiding collision. Each activity includes sensors, one or more feature extractor modules, and an action module necessary for it to carry out its task. To perform reliably, the activity uses relevant sensors to focus on specific features of the environment.

The blackboard control unit monitors the environment and assesses the performance of enabled activities on the basis of information reported by the activity and its feature extractor modules. The feature extractor modules have two functions: filtering data to be passed to the blackboard control unit and focusing sensory perception on the action being performed. The feature extractor modules of disabled activities do not control the vehicle but rather monitor the environment for possible conditions that their activities could easily handle. For the vehicle to perform a complete task, it must combine the appropriate activities in the proper sequence, primarily driven by the events posted on the blackboard by the activities.

The blackboard system supports high-level symbolic decision making and central intelligent reasoning through the knowledge in the control unit. The only difference of our system from classical blackboard systems is that our activities do not use opportunistic problem solving but are sequenced and orchestrated by the control unit to perform a global task in reaction to the changing environment.

Feature extractor module. Although a feature extractor may depend on more than one sensor, it provides information on a specific attribute in the environment. Sensors and their associated perception modules are continuously operating. The feature extractor processes sensory data and expresses it in a suitable representation for the action or blackboard module.

The feature extractor expresses the relevant attributes of the environment symbolically and then relays them to the blackboard. Since the feature extractors are continuously operating, the blackboard reasoning module is instantly notified about the dynamic changes in the environment. Each feature extractor is responsible for ensuring the consistency of the information it sends to the blackboard, relieving the blackboard of the time-consuming truth maintenance task.

FOR THE VEHICLE TO PERFORM A COMPLETE TASK, IT MUST COMBINE THE APPROPRIATE ACTIVITIES IN THE PROPER SEQUENCE, PRIMARILY DRIVEN BY THE EVENTS POSTED ON THE BLACKBOARD BY THE ACTIVITIES.

Action module. During navigation, the enabled action module determines the mobile robot's direction and speed in response to information acquired from the corresponding feature extractor. Therefore, low-level decisions are sensor-driven, ensuring tight, reactive control.

To intelligently guide real-time operation, the high-level reasoning module (that is, the blackboard module) provides the action module with the desired direction and speed based on its global understanding of the situation. Nonetheless, it is the action module's responsibility to provide the actual control signals and maneuver the mobile robot to meet the reasoning module's recommendations. The high-level commands may also include constraints on the action modules' operations to ensure smooth transition from one activity to another.

Blackboard module. The blackboard module, consisting of the blackboard and the blackboard control unit, assesses the robot's global situation and provides sound decisions about the appropriate action to meet user-specified goals safely. The blackboard control unit performs this reasoning by processing symbolic information such as the sequence of specified goals, the current control strategy, and the acquired knowledge of the states of the world and of the mobile robot. Symbolic and qualitative information about the features of interest are continuously posted by the feature extractors to the blackboard.

Blackboard control unit. The blackboard control unit uses traditional forward-chaining production rules for the reasoning process. The typical case involves the selection of a suitable activity to satisfy a set of conditions derived from the goal requirements or from the safety requirements. The control unit's role is to arbitrate between the different activities competing to take control of the mobile robot's actuators.

The production rules activate or deactivate the appropriate activities according to the current state and the adopted control strategy. They may also modify certain operation parameters of the currently enabled activity. Since the activation-deactivation cycle is performed continuously in real time, this module appears to an external observer to be explicitly commanding the mobile robot by smoothly switching to the appropriate activities without interruption.

Blackboard essential states. The numeric-to-symbolic conversions are a challenge; we manage them mostly by means of the concept of essential states and events developed by Tigli et al.[15] (These conversions are a mapping from numeric values to linguistic predicates.) To avoid overloading the blackboard with information and to minimize the bandwidth requirement between modules, we limit the representation of the environment's state to symbolic attributes from the feature extractor modules and the vehicle's state to symbolic information from the action modules. Generally, the decision-making process is sensitive only to coarse measures of these attributes. For instance, the decision-making process would be interested in the fact that the mobile robot is close to a wall rather than the exact numeric distance.

The essential states of any type of sensory data are defined by the partition of the numerical range of values of that sensory data into a small number of mutually disjoint partitions. The boundaries between these partitions are chosen intuitively to facilitate symbolic reasoning. Similarly, the action modules' states can be expressed symbolically by nonoverlapping descriptions (for example, in the simplest case, "activated" or "deactivated"). The essential events are then defined as the changes in the corresponding essential states. The essential events cause the essential states to be modified on the blackboard.

The current state of the environment belongs to the Cartesian product of the essential states of all the sensory data. Similarly, the current states of all activities belong to the Cartesian product of all their essential states. These are represented by two vectors in the two Cartesian spaces illustrated in Figure 2, which shows the sensory and action states that must be accounted for when defining the rule set. We achieve completeness of the rule set by exploring and covering all the regions of the two Cartesian spaces, thus ensuring safe operation of the robot in reaction to changes in its environment. Besides these rules, there are also rules that decompose the user-specified goals into sequences of activities that guide the robot.

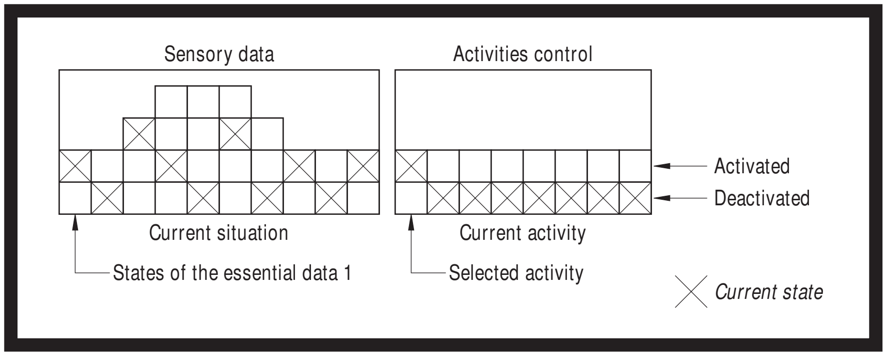

Figure 2. Sensory and action essential states as defined in the blackboard.

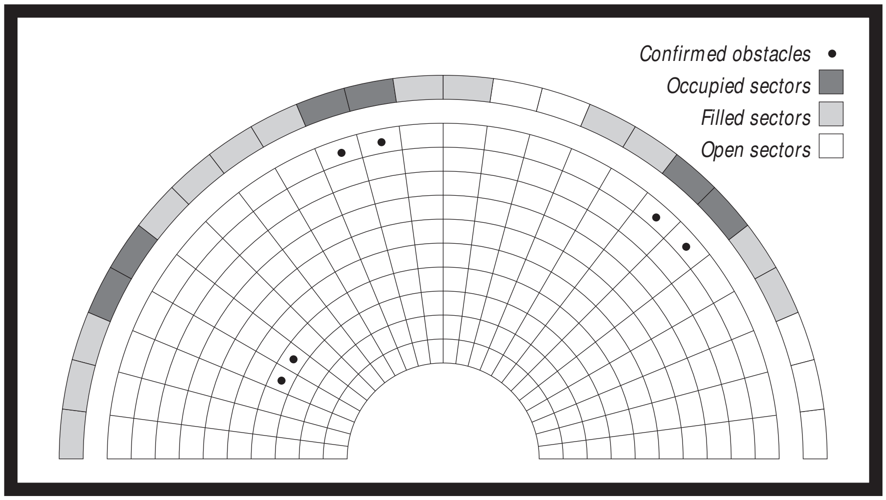

Figure 3. Sectors determined from sonar range data on a certainty grid.

Implemented activities

The activities we have implemented for the mobile-robot system include a passageway activity, a floor anomaly activity, and a dynamic path-planning activity.

Passageway activity. The passageway activity's main focus is navigating the vehicle toward open space. In effect it is a collision avoidance activity, except that one can specify a desired direction, velocity, and destination. The robot uses the activity for traversing through doorways, navigating down hallways, and generally avoiding collision with any objects.

The passageway activity uses data from sonar range sensors to construct a two-dimensional Cartesian grid representation of the robot's environment. The histogram grid method[16] of representing this information minimizes the effects of spurious data typically produced by sonar range sensors. The computation of a grid cell's value is based on the number of times an object has been detected in that cell; the larger the value, the more confidence that the cell is occupied. The activity then uses a sector elimination method to select a direction for collision-free motion that most closely matches the desired direction. The sector elimination algorithm divides the sensed environment into 48 sectors covering a discoid centered about the robot, with a hole in the center, and extending approximately 1 meter from the vehicle.

Figure 3 shows the front half of a polar certainty grid and its associated sectors. Sectors with obstacles are marked in black and denoted as "occupied." If the vehicle's width is ignored, the sectors hashed out are sufficient to prevent a collision. In reality, the vehicle will collide with an object in a sector if commanded to move in the direction of an open sector next to that vector. This is primarily due to the lack of clearance between the object and the vehicle's extremities. To account for this feature, we must increase the size of a group of "occupied" sectors. The figure shows these sectors in gray and denotes them as "filled" sectors. Those sectors that are open for passage appear as white and are denoted as "open" sectors. The robot is steered toward the bisector of the open sector that best matches the desired direction.

The essential-events information sent to the blackboard by the passageway activity's feature extractor is not as precise as the certainty grid, but can be represented in the computer with less effort and memory. The open region around the vehicle is divided into four zones: "ahead," "right," "behind," and "left." The blackboard thus has a low-resolution view of the world in terms of quadrants around the perimeter. In addition to the data events posted by the feature extractor the activity's action module reports its status to the blackboard. The action's only states are "enabled" or "disabled," corresponding to whether or not the activity has control of the vehicle. In summary, the passageway activity posts one data event consisting of four states and one status event consisting of two states.

Floor anomaly activity. The floor anomaly activity's responsibility is to safely navigate the vehicle around or across anomalies, or obstacles, on the floor. Currently, work is under way on the development of a sensor and methodology, called FAD (Floor Anomaly Detector) for object detection using a pair of Biris laser range finders.[17] (The computation of range is based on a principle of replacing the single iris in the camera with a couple of irises, hence its name, bi-iris.)

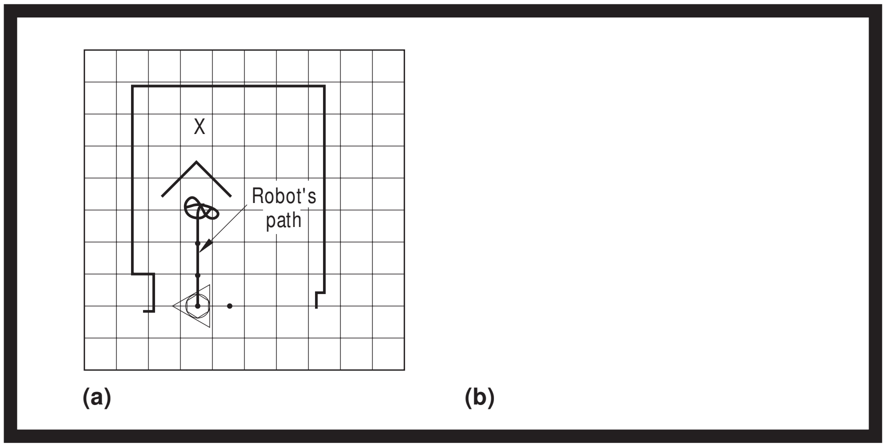

Figure 4. Mobile robot's path: (a) passageway activity only; (b) passageway and path-planning activities.

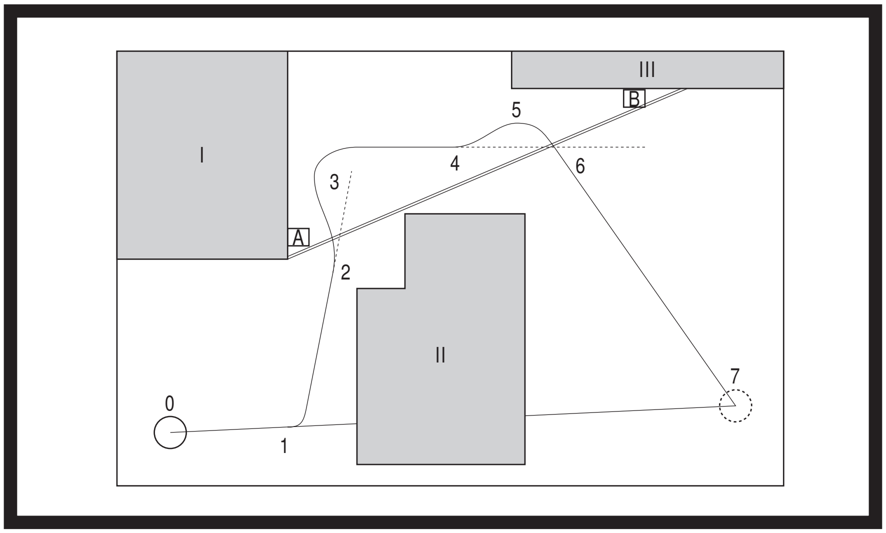

Figure 5. Possible workshop scenario.

The current floor anomaly activity determines the feasibility of crossing obstacles such as channels or cables detected on the floor and takes the necessary actions to cross them safely. The activity's feature extractor relies on available sensory information to identify any obstacles present in the vehicle's vicinity. It extracts their relevant attributes (typically depth, width, orientation, and distance from the vehicle), which control the action module. The action module guarantees a safe passage over the obstacle.

Essential events the floor anomaly activity sends to the blackboard are a set of symbolic values describing the robot's distance from the obstacle ("close" or "far") and the robot's direction in relation to the obstacle ("toward," "parallel," or "from"). We have determined most of the threshold values for these sets experimentally. The action module also reports its status to the blackboard, including "starting," "orienting," "proceeding," "leaving," "finishing," and "paralleling," as well as the obvious "enabled" and "disabled." Thus the floor anomaly activity posts two data events, one consisting of two states, the other of three; and two status events, one of two states and the other of six.

Dynamic path-planning activity. After several experiments with the passageway activity, we discovered that under certain conditions the algorithm would not function satisfactorily. Collisions with objects were minimal, but the robot could easily be trapped in a corner as shown in Figure 4a.

The figure represents the robot by a combination of several shapes: a triangle, a circle, a hexagon, and a coordinate frame placed about the vehicle's center point. The triangle represents the robot's bumpers. The hexagon is the base of the robot. The circle represents the ring of sonar sensors around the base that are used by the passageway activity. The × marks the robot's destination. The grid overlaid on the figure marks off areas of 1 square meter, and the test zone is an approximately square area of 6 m × 6 m.

In contrast to the passageway activity, the dynamic path-planning activity can guide the robot around the partition, as shown in Figure 4b, because it maintains a larger map of the robot's environment, which it uses to determine possible paths to the robot's destination point. The gray-scale areas in Figure 4b represent certainty values for the existence of objects detected by the robot's sensor as it moves along its path. White indicates the highest likelihood that the cell is empty, black indicates the highest likelihood that the cell is occupied, and shades of gray indicate the various intermediate levels of likelihood of occupancy.

The path planner's map is based on a histogram grid map similar to the one used in the passageway activity, but it uses a rectangular grid centered about some point in the environment. The grid cells are 10 square cm; therefore, a map consists of 10,000 cells and takes approximately 100 ms to update. Range values for map creation come from a laser-based Biris camera and are more precise and of higher angular resolution than those from the ultrasound sensors.

The path-planning algorithm is based on Warren's modified A* algorithm,[18] extended by Stuck[19] to account for the mobile robot's width. Similar to other planning algorithms, it takes much longer to compute a free path than the passageway activity (at times taking 7 s compared to 100 ms for the passageway activity), and therefore it is not as reliable as the passageway activity. However, it maintains a much larger representation of the environment and uses a deterministic approach to find a direction for the robot. Later in this article we will demonstrate how the path-planning activity can improve the robot's performance.

Currently the path-planning activity does not communicate with the blackboard, so we cannot yet report on the essential states it will post on the blackboard. However, we think they will be similar to those generated by the passageway activity since the two activities have similar functions, differing primarily in approach.

System behavior example

The following example of a mobile robot navigating across a typical factory floor demonstrates the performance goals that mo-

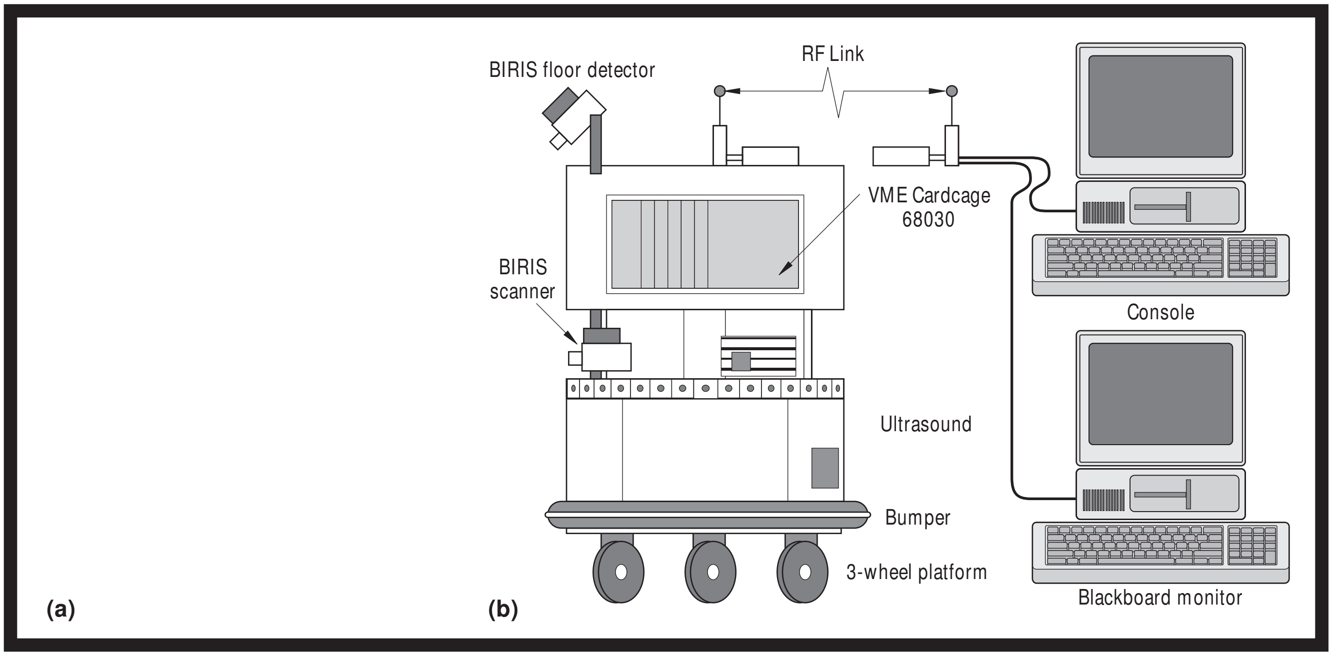

Figure 6. Photo of experimental mobile robot (a) and block diagram of robot's configuration (b).

tivated our design of the activity-based blackboard architecture. We designed the passageway, path-planning, and floor anomaly activities to address the kinds of problems the robot encounters here.

In Figure 5, the robot is originally at location 0 and is required by the blackboard to navigate autonomously to destination point 7. The robot, perceiving no obstructions, proceeds in the direction of point 7 until it reaches point 1, where the passageway activity notifies the blackboard of the presence of an obstacle. The passageway activity also realizes that the space between object II and the south wall is not wide enough for the robot to pass through, and, therefore, decides to seek an open space in another direction. The passageway activity can drive the mobile robot toward point 2, but due to limited storage capacity in the map used by the passageway activity, objects sensed earlier are no longer available and the robot moves back toward point 1, driven by point 7.

The path-planning activity, with its greater knowledge of the environment, suggests to the blackboard an alternate route, which the passageway activity uses to guide the robot toward point 2. The robot reaches point 2, where the floor anomaly activity detects channel AB. The blackboard allows the floor anomaly activity to take over, temporarily, to maneuver the robot across this channel. The vehicle crosses the channel safely and proceeds along its temporary direction. If the floor anomaly acitivity had not been allowed to take over the robot, the path taken would have been the one shown as a dashed line commencing from pont 2. This could have led to the wheels of the mobile robot having been trapped in the channel. At point 3, the blackboard returns control to the passageway activity after being notified by the floor anomaly activity of the successful completion of its task. Throughout this procedure, the path-planning activity has been computing the paths that will get the mobile robot to its destination. One of these paths is from west to east between object I and object III. The blackboard realizes that going through this passageway will bring the vehicle closer to the original goal point and commands the passageway activity accordingly.

At point 4, the floor anomaly activity identifies the second portion of channel AB, and the symbolic feature extractor notifies the blackboard that the robot can safely maneuver across it. The dashed line from point 4 towards object II identifies the path the robot would have taken if allowed to cross the channel at this point. However, the passageway activity notifies the blackboard that there is no room for safe crossing of the channel at this point because of the presence of object II on the other side. So the blackboard notifies the floor anomaly activity to go along the channel away from object II. At point 5 the blackboard allows the floor anomaly activity to drive the robot across the channel, and at point 6 the passageway activity drives the robot toward destination point 7. With these three activities guided and coordinated by the blackboard, the robot reaches its destination safely.

Implementation and experiments

The mobile-robot system shown in Figure 6 implements the activity-based blackboard architecture. It consists of a Cybermotion platform fitted with a ring of 24 ultrasound sensors around the base, two Biris laser range finders for detecting floor anomalies, a scanning Biris laser range finder, and an odometric counter. An on-board Z80 processor serves as a PID (proportional-integral-derivative) controller for the mobile robot's wheels and a counter for the odometer. All activities run on a 68030-based multiprocessing platform that is embedded in the robot. During experimentation, we used up to seven processors, but made no attempt to optimize the loads on the processors. In general, one processor must be dedicated to each external device (ultrasound ring, Biris scanner, Biris FAD system, and mobile robot), freeing the rest for processing and interpreting the sensory data and communication to the blackboard.

The system is implemented primarily by means of tasks communicating through message passing on a multitasking computer platform. We guaranteed real-time performance by using a real-time multiprocessing operating system called Harmony, which allows the activity modules to be implemented across several processors, thus ensuring concurrency.

The decision-making mechanism is a rule-based production system using the BB_CLIPS blackboard system (developed

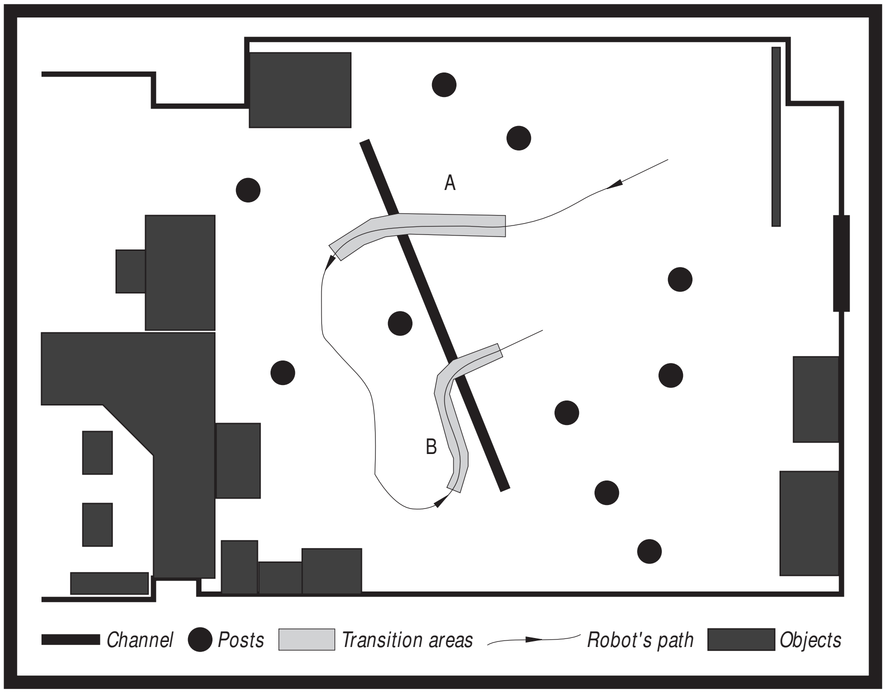

Figure 7. An experiment using the passageway and floor anomaly activities.

at the National Research Council's Institute for Information Technology). It runs on a Macintosh IIfx and communicates with the activities via an RF modem. Since information posted on the blackboard is symbolic compact data and does not directly impact an activity's performance, time delays usually associated with using an RF modem are reduced. The only time this is a crucial factor is when an activity wants control of the vehicle and the control unit is busy. In the current design of the blackboard, the control unit does not perform extensive computations but releases this chore to other activities, so it is usually waiting to process state changes that have been posted on the blackboard.

Passageway and floor anomaly experiment. The experiment diagrammed in Figure 7 was designed to investigate the interaction of the passageway and floor anomaly activities. The vehicle was commanded to navigate safely in the laboratory space, which was cluttered with stationary posts and other objects and some moving people. The laboratory also contained a simulated floor anomaly consisting of a channel of a known depth and size. During the experiment, the vehicle switched back and forth between the two activities in response to information dynamically posted on the blackboard.

In the default situation, the vehicle moved along a straight line segment (under passageway activity control) until an object or a channel entered its visibility range and caused it to change its path. The feature extractors assumed the responsibility of judging objects and channels perceived by the sensors; they notified the blackboard only when these items constituted a significant obstruction. In cases when objects obstructed the vehicle's intended path, the passageway action module decided on the corrective action with no intervention from the blackboard. Such corrections appear in the figure as changes in the direction of the robot's path. On the other hand, in cases of threatening channels, the blackboard commanded the floor anomaly activity to take over the vehicle's control, until that activity reported successful crossing of the channel (section A in Figure 7). Figure 8 shows the BB_CLIPS rules triggered in this situation.

In cases when objects on the other side of the channel made it dangerous to cross, the blackboard commanded the floor anomaly activity to go parallel to the channel until the vehicle reached a safe crossing point. An example of this situation is shown as the section marked B in Figure 7. In this example the system control unit has a combined representation of the objects and the channel on the floor. It is not possible to achieve this type of performance using a single activity because each activity is tuned to handle a certain type of condition. The BB_CLIPS rules triggered for this situation are shown in Figure 9.

```
(defrule Command_floor_anomaly_cross_0 ""
(CHANNEL_EXTRACTOR (maneuver TRUE) (closer TRUE) (parallel ?parallel))
(OPEN_SPACE_EXTRACTOR (clear TRUE) (right? right) (left ?left))
(FLOOR_ANOMALY_ACTION_MODULE_ (state ?floor_anomaly_state&~PARALLELING))
(COMMAND ?command&~"floor_anomaly_parallel")
(COMMAND ?command&~"floor_anomaly_cross")
=>
```

Figure 8. BB_CLIPS rules for channel crossing.

```
(defrule Command_floor_anomaly_cross_0 ""
(CHANNEL_EXTRACTOR (maneuver TRUE) (closer TRUE) (parallel ?parallel))
(OPEN_SPACE_EXTRACTOR (clear FALSE) (right? right) (left ?left))
(FLOOR_ANOMALY_ACTION_MODULE_ (state PROCEEDING))
```

Figure 9. BB_CLIPS rules for moving parallel to channel.

(a) (b)

Figure 10. Mobile robot's path: (a) path-planning activity only; (b) passageway and path-planning activities.

Passageway and path-planning experiment. When we used the path-planning activity to control the robot directly, the robot sometimes collided with an object, as shown in Figure 10a. Such collisions were caused primarily by delays in computing a new path. For this reason, the path-planning activity is not reactive enough to be effective and reliable unless the robot's speed is very slow.

The passageway and path-planning activities are complementary because they have identical action modules. When combined, they increase the robot's chances of safely reaching its destination. In Figure 10b, the passageway activity drove the robot away from the wall and prevented the collision shown in Figure 10a. In this experiment, the path-planning activity directly interacted with the passageway activity by using a common action module, as shown in Figure 11, and was not interfaced through the blackboard.

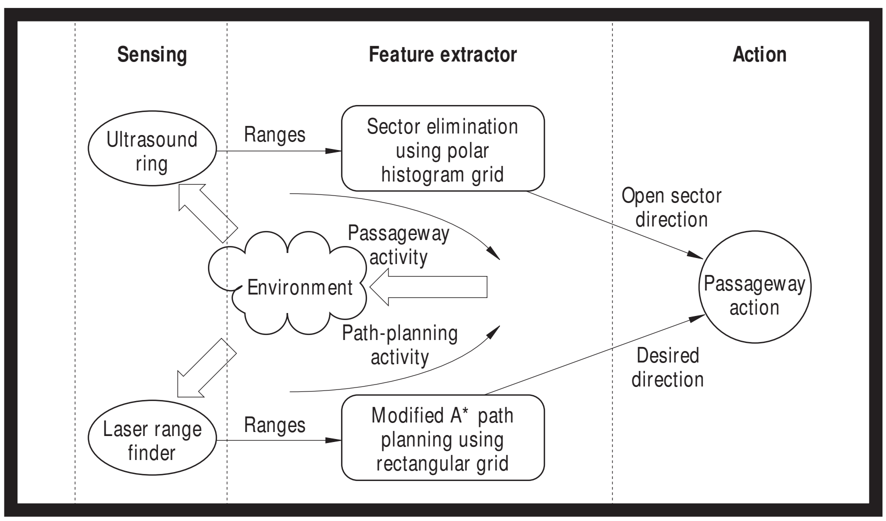

Figure 11. Interaction of passageway and path-planning activities. The large arrows signify an interaction with the environment by the robot and a sensing of the environment by the Ultrasound ring and laser range finder.

In the configuration shown in Figure 11, the path-planning activity's feature extractor module sends the passageway action module the desired direction of motion. The passageway activity's feature extractor module sends the passageway action module an open-sector direction. The passageway activity uses the two directions to compute an actual direction for the mobile robot.

This experiment demonstrates how the two activities complement one another. By working together, they minimize the need to use the blackboard to transfer data between activities and reduce the amount of trivial information on the blackboard. In this limited example, the two activities are more complementary than conflicting and thus function together rather well. In the case of the passageway and floor anomaly activities, however, each performs substantially different functions.

Summary of experimental results. We have yet to integrate all three activities into one system. Results from the combined floor anomaly and passageway activities support the claim that our architecture is suitable for reactive and strategic control of a mobile vehicle. Results from the path-planning activity combined with the passageway activity support the idea that certain activities are complementary and will function well together, bypassing the blackboard as a communication medium.

Nevertheless, we believe the blackboard is necessary for maintaining a global perspective of the mobile robot's state and also for supervising the activities. For example, the path planner computes an optimal path for the robot. If the robot must wait for an elevator door to open, the blackboard's control unit can override communications between the path planner and the passageway activity, stopping the vehicle and resuming its operation after the door opens.

The blackboard also allows the system to monitor an activity's performance and replace that activity with one with better performance. If for any reason the passageway activity fails, the system can use the path-planning activity to drive the mobile robot safely, provided the robot's operating speed is reduced.

Our two experiments show that this architecture is suitable for experimental robotics applications, particularly because of its flexibility and modularity. It allowed us to experiment with several activities and investigate their interactions, while some components were implemented and others were simulated. It also allowed us to combine two activities to achieve higher performance.

System evaluation

Earlier, we discussed a number of desirable behavior and design attributes of a mobile robot. Although expressing them quantitatively is difficult, we can summarize qualitatively how the activity-based blackboard architecture meets these requirements.

Behavior attributes

- *Reactivity*: The control architecture's reactivity is ensured by the conceptual separation of two types of decision making: strategic and tactical. The system demonstrated its reactivity by switching the control context, but not the control level, in response to dynamic and unexpected events.
- *Intelligence*: Defining intelligence as the ability to cope with new or trying situations, we have been able to demonstrate that the mobile robot can cope with unknown environment configurations. Using the blackboard as the high-level decision-making mechanism, we incorporate progressive levels of situation understanding by including rules to describe those situations.
- *Global reasoning*: The robot performs global reasoning primarily through the control unit and the blackboard. Arbitration between competing activities is accomplished by a set of rules that allows the most appropriate activity to take control of the mobile robot and resolves information conflicts. For example, as we saw in Figure 7, at the appearance of an object across the channel, the control unit commanded the floor anomaly activity to go parallel to the channel before crossing it.
- *Multiple-goal resolution*: Each activity's control algorithm is exposed to continuous information coming from the corresponding feature extractor. Therefore, each action module has its own suggestions as to the control signals to be sent to the vehicle actuators. The blackboard, however, performs selective activation and deactivation of the action packages, arbitrating the multiple goals provided by the different activities. The blackboard continuously assesses the relevance and applicability of each activity to the situation at hand.

THE BLACKBOARD ALSO ALLOWS THE SYSTEM TO MONITOR AN ACTIVITY'S PERFORMANCE AND REPLACE THAT ACTIVITY WITH ONE WITH BETTER PERFORMANCE.

- *Robustness*: More testing must be performed, and activities added, before we can comment accurately about the mobile robot's robustness. Activities with similar functions (for example, the path-planning and passageway activities) can improve robustness.
- *Reliability*: A significantly long time is required to build confidence in any system's reliability, and we have not yet tested all the system's components thoroughly. So far, we have extensively tested the passageway activity, and it performs consistently (does not allow the robot to collide with objects detected by the sensor) in many different scenarios. The addition of the path-planning activity improved the system's reliability to reach a destination.

Design attributes

- *Modularity*: The simple interconnections of the activities' components allow independent design of the modules before they are integrated to form a complete activity. For example, the feature extractor of the floor anomaly activity was developed and tested independently from the action module of the same activity. Similarly, at a higher granularity level, the activities were developed separately from the blackboard system. The architecture's modularity facilitated both design and debugging.
- *Flexibility*: The interconnection patterns of the control framework's components impose no restrictions on the information exchanged by the modules. There are no limitations on the messages coming from the feature extractors. Therefore, we changed the symbolic information generated by the feature extractors, especially that of the passageway activity, several times during the experimentation, to improve the blackboard's decision-making capability. These changes were totally transparent to the system architecture and organization. This system flexibility allowed design decisions to be based on experimental evidence and practical performance measures.
- *Expandability*: The system's modularity facilitates its expansion. For example, developers worked separately on skill-achieving activities, before these activities were integrated into the control structure. The modifications to the control architecture were limited to the addition of appropriate rules in the blackboard control unit to take advantage of the added capabilities.
- *Adaptability*: The high-level blackboard control over the activities continuously changes the system's focus of attention in response to changing world configurations. The blackboard chooses the activity that can handle the situation best: the floor anomaly activity in the vicinity of threatening channels and the passageway activity the rest of the time. The blackboard rules include implicit knowledge of the capabilities of both activities' action modules.

- *Multisensor integration*: The vehicle has four different sensors: (1) the on-board odometer, (2) the ultrasound transducers, (3) the pair of Biris laser ranging cameras for anomaly detection, and (4) the scanning Biris laser ranging camera. Although we have not yet integrated all the activities into one system, the architecture has accommodated several of these sensors during the experiments.

WE STILL HAVE MANY ISSUES to address in developing our robotic architecture. In particular, there is no rigorous method for converting some of the numeric data from the feature extractors into symbolic data for the blackboard. In our case, these conversions followed the idea of essential events and states, which minimizes the amount of data going to the blackboard, but we have not tested this concept thoroughly. The advantage of essential events and states is mainly the reduction of data, which, in turn, allows us to address the serious problem of the rule base's completeness. As the number of activities increases, the possible combination of events increases exponentially, leading to a combinatorial explosion. The solution is that not all activities should report all the time. For example, the floor anomaly activity should post on the blackboard information about a floor anomaly only when one has been detected.

On the practical side, to fully exploit the benefits of this type of design, we must have access to a clean multitasking operating system. Unfortunately, many mobile-robot systems are closed systems, designed as black boxes rather than potential programming development workstations, making experimental research difficult. A mobile robot's environment is so dynamic and difficult to constrain that an efficient and easily modifiable development system is essential. The system must be easily reprogrammed and reconfigured for the different environments it may encounter. Thus, it is imperative to simplify the creation of activities and their integration into the system.

Finally, the design of special-purpose activities allows us to create specific functions that a mobile robot can perform reliably. These activities give the appearance of robot intelligence, but only in a limited domain. We believe that to achieve more intelligent behavior, a supervisory centralized module like the blackboard is necessary, but it is not necessary that all communications go through the blackboard. We have had some success with this approach, but to fully exploit it, we must develop more activities.

Acknowledgments

This work was partially supported by a Natural Sciences and Engineering Research Council grant to Gerald Karam, who kindly used it to support Reda Fayek as a master's student at Carleton University. We thank Marie-Claude Thomas, at Sophia Antipolis University in Nice, France, for financial and supervisory support of Jean-Yves Tigli. This work was performed in cooperation with the Precarn ARK (Autonomous Robot for a Known Environment) Project, which receives its funding from Precarn Associates Inc., the Department of Industry Science and Technology Canada, the National Research Council of Canada (NRC no. 38355), Technology Ontario, and Ontario Hydro.

References

1. D.W. Payton, "An Architecture for Reflexive Autonomous Vehicle Control," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE Computer Society Press, Los Alamitos, Calif., 1986, pp. 1838–1845.
2. C. Malcolm, T. Smithers, and J. Hallam, "An Emerging Paradigm in Robot Architecture," in *Intelligent Autonomous Systems 2*, Vol. 2, T. Kanade, F.C.A. Groen, and L.O. Hertzberger, eds., Elsevier Science Publishers, Amsterdam, Dec. 1989, pp. 545–564
3. J.S. Albus, R. Lumia, and H. McCain, "Hierarchical Control of Intelligent Machines Applied to Space Station Telerobot," *Proc. IEEE Symp. Intelligent Control*, IEEE CS Press, 1987, pp. 20–27.
4. R. Lumia, "Using Nasrem for Real-Time Sensory Interactive Robot Control," *Robotica*, Vol. 12, No. 2, March 1994, pp. 127–135.
5. C. Isik and A.M. Meystel, "Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot," *IEEE J. Robotics and Automation*, Vol. 4, No. 3, June 1988, pp. 241–255.
6. R. Hinkel, T. Knieriemen, and E. von Puttkamer, "An Application for a Distributed Computer Architecture—Real-Time Data Processing in an Autonomous Mobile Robot," *Proc. Eighth Int'l Conf. Distributed Computing Systems*, IEEE CS Press, 1988, pp. 410–417.
7. R.C. Arkin, "Motor Schema-Based Mobile Robot Navigation," *Int'l J. Robotics Research*, Vol. 8, No. 4, Aug. 1989, pp. 92–112.
8. A. Stentz, "The Codger System for Mobile Robot Navigation," in *Vision and Navigation, The Carnegie Mellon Navlab*, Charles E. Thorpe, ed., Kluwer Int'l Series in Engineering and Computer Science, Kluwer Academic Publishers, Boston, 1990, chap. 9, pp. 187–201.
9. Y. Goto and A. Stentz, "The CMU System for Mobile Robot Navigation," *Proc. Third Int'l Conf. Robotics and Automation*, IEEE CS Press, 1987, pp. 99–105.
10. R.A. Brooks, "A Layered Intelligent Control System for a Mobile Robot," *Third Int'l Symp. Robotics Research*, MIT Press, Cambridge, Mass., 1985, pp. 365–372.
11. R.A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE J. Robotics and Automation*, Vol. RA-2, No. 1, Mar. 1986, pp. 14–23.
12. B. Hayes-Roth and M. Hewett, "BB1: An Implementation of the Blackboard Control Architecture," in *Blackboard Systems*, R. Engelmore and T. Morgan, eds., Addison-Wesley, Reading, Mass., 1988, chap. 14, pp. 297–313.
13. R.E. Fayek, R. Liscano, and G. Karam, "A System Architecture for a Mobile Robot Based on Activities and a Blackboard Control Unit," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE CS Press, 1993, pp. 267–274.
14. S.Y. Harmon, "Practical Implementation of Autonomous Systems: Problems and Solutions," *Proc. Int'l Conf. Intelligent Autonomous Systems*, Elsevier, Amsterdam, 1986, pp. 47–59.
15. J.-Y. Tigli et al., "Methodology and Computing Model for a Reactive Mobile Robot Controller," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, IEEE Press, New York, 1993, pp. 17–20.
16. A. Manz, R. Liscano, and D. Green, "A Comparison of Real-Time Obstacle Avoidance Methods for Mobile Robots," *Proc. Second Int'l Symp. Experimental Robotics*, Springer Verlag, London, 1991, pp. 301–316.
17. A. Manz et al., "Floor Obstacle Avoidance for a Mobile Robot," *Proc. 1993 DND Workshop on Advanced Technologies in Knowledge Based Systems and Robotics*, Industrie Canada, Ottawa, Canada, 1993.
18. C.W. Warren, "Fast Path Planning Using Modified A* Method," *Proc. IEEE Int'l Conf. Robotics and Automation*, IEEE CS Press, 1993, pp. 662–667.
19. E.R. Stuck et al., "Map Updating and Path Planning for Real-Time Mobile Robot Navigation," *Proc. IEEE/RSJ/GI Int'l Conf. Intelligent Robots and Systems*, Munich, IEEE Press, New York, 1994, pp. 753–760.

Ramiro Liscano is a research officer at the Institute for Information Technology at Canada's National Research Council. His research interests have been in the processing of sensory data, primarily for real-time, autonomous navigation of mobile robots. He is also pursing his PhD at the PAMI Laboratory in the Systems Design Engineering Department of the University of Waterloo, Canada. His thesis research is in the development of reasoning techniques for the interpretation of sensory data from 3D active vision systems for modeling indoor environments. He graduated from the University of New Brunswick, Canada, in mechanical engineering in 1982, and earned his MS (1984) in mechanical engineering from the University of Rhode Island, specializing in robotics and computer vision. He is a member of the IEEE. He can be reached at the Institute for Information Technology, National Research Council, Ottawa, Ontario K1A 0R6, Canada; liscano@iit.nrc.ca.

Allan Manz currently works as a robotics engineer at Applied Robotics in Saskatoon, Saskatchewan, where he is investigating the autonomous navigation of underground mining vehicles. His research interests are in the processing of sensory data from ultrasound, 3D active vision, and 2D passive systems, primarily for autonomous navigation of mobile robots. He completed his bachelors (1986), post graduate diploma (1989), and MS (1993) degrees from the University of Saskatchewan in mechanical engineering. He can be reached at the Institute for Information Technology, National Research Council, Ottawa, Ontario K1A 0R6, Canada; manz@iit.nrc.ca,

Elizabeth R. Stuck is currently a research associate at the institute for information technology at Canada's National Research Council. Her research interests are in human-computer interaction, in particular the visualization and manipulation of computer information. She graduated from St. John's College with a degree in liberal arts and earned her MS (1987) and PhD (1992) in computer science from the University of Minnesota with an emphasis in artificial intelligence and a minor in cognitive science. She is a member of AAAI and ACM. She can be reached at the Institute for Information Technology, National Research Council, Ottawa, Ontario K1A 0R6, Canada; liz@ai.iit.nrc.ca.

Reda E. Fayek is a PhD candidate at the PAMI Laboratory in the Systems Design Engineering Department of the University of Waterloo, Canada. His research focus on the use of 3D computer vision and active sensing techniques in 3D modeling and mapping of natural terrain and outdoor autonomous navigation. He graduated with a degree in electrical and computer engineering in 1988 from Ain-Shams University, Egypt, and earned his MEng in systems and computer engineering in 1992 from Carleton University. He is a member of the IEEE. He can be reached at Systems Design Engineering Department, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada; reda@watnow.uwaterloo.ca.

Jean-Yves Tigli is a member of the I3S Laboratory, pursuing his PhD, at the Nice-Sophia Antipolis University. His research interests are in knowledge-based situated and behavioral agent systems for controlling mobile robotics, and also miniature mobile robots. He received a degree in computational engineering from the Nice-Sophia Antipolis University, France, in 1991. He is member of the IEEE, Robotics & Automation, and Man and Cybernetics societies. He can be reached at Laboratoire I3S - CNRS URA 1376, Université de Nice, Sophia Antipolis, Valbone 06560, France; tigli@mimosa.unice.fr.