Evaluating Knowledge and Representation for Intelligent Control

Elena R. Messina, John M. Evans, James S. Albus
Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8230

ABSTRACT

Knowledge and the way it is represented have a tremendous impact on the capabilities and performance of intelligent systems. There is evidence from studies of human cognitive functions that experts use multiple representations in problem solving tasks and know when to switch between representations. In this paper, we discuss the issues pertaining to what types of knowledge are required for an intelligent system, how to evaluate the knowledge and representations, and provide examples of how representation affects and even enables functionality of a system. We describe an example of an intelligent system architecture that is built upon multiple knowledge types and representations and has been applied to a variety of real-time intelligent systems.

1. INTRODUCTION

Various definitions of intelligence, whether pertaining to artificial or biological, make reference to knowledge. The American Heritage Dictionary defines intelligence as "the capacity to acquire and apply knowledge." Newell and Simon stated that "a physical symbol system has the necessary and sufficient means for general intelligent action." [20] Despite this, there is a paucity of literature that provides guidance to developers in terms of what is the needed knowledge within an intelligent system and how to decide on appropriate representations. This is especially true when it comes to building real-time intelligent systems, such as those for controlling autonomous mobile robots and advanced manufacturing equipment.

2. STATUS OF KNOWLEDGE AND REPRESENTATION

In 1989, Wah stated that "despite a great deal of effort devoted to research in knowledge representation, very little scientific theory is available to either guide the selection of an appropriate representation scheme for a given application or transform one representation into a more efficient one." [24] There is little evidence to repudiate this statement in 2001, particularly for real-time control.

The most basic aspect of representation design is based on pairing it to the algorithms that use it. It is well known in computer science that there is a relationship between the representation of data and the algorithms that operate on it. Efficiency of algorithms is highly dependendent on the organization of the data, therefore a starting point for design and evaluation of knowledge representation should be based on broader computer science tenets, such as those described in [16].

Davis et al. argue for a broader understanding of what knowledge representation entails [7]. Certainly representation, in any form, is a surrogate for things that exist in the real world. The issue of required fidelity of representation therefore arises. They also see knowledge representation as a set of ontological commitments, meaning that the representation choice serves as a "strong pair of glasses that determine what we can see, bringing some part of the world into sharp focus, at the expense of blurring other parts." The focussing/blurring effect is crucial because of "the complexity of the natural world is overwhelming." They conclude that knowledge representation researchers ought to characterize the nature of the glasses they are supplying, thus making the ontological commitments explicit, and that the field ought to develop principles for matching representations to tasks.

In general, most of the literature describes the use of a single representation for all the knowledge within a given system. In mobile robotics, one sees three main approaches. The first is geometry-based, where sensors or probabilistic models are used to build maps. The second is feature-based, where the topology of the environment and high-level objects of significance are stored. The third is a symbolic approach, where first-order logic or rule-based systems are used. Examples of geometry-based approaches include occupancy grids [18] and sensor-based map building [23]. Feature-based systems include [14] and [25]. Symbolic systems include STRIPS [9] and GOLOG [15]. Exceptions to this "monomodeling" design do exist, such as the hybrid intelligent systems of Devedzic [8], the multimodeling system of Chittaro [6], and the qualitative and quantitative representations of Kuiper's semantic spatial hierarchy [13]. In most cases, these multirepresentational approaches have not been applied to functioning real-time controllers.

Evidence from the cognitive science field indicates that human problem solving capabilities rely heavily on the ability to switch between representations as required [5]. Chittaro et al. [6] note that systems that reason about physical systems require

- representation adequacy
- problem solving power
- problem solving economy
- multiple uses of knowledge (for multiple problem-solving tasks)
- cognitive coupling
- efficiency

They also claim that "efficiency cannot be achieved, in general, using only one model: an appropriate problem decomposition and the cooperation of a variety of knowledge sources organized at different levels of aggregation and accessible under appropriate views is possibly the only way of adequately coping with complexity issues."

3. MULTI-REPRESENTATION EXAMPLE

One example of a multi-representational approach to real-time intelligent systems is the Real Time Control System (RCS) and its mobile autonomous vehicle version, 4D/RCS [1][2]. A general framework for the RCS model-based control system is shown schematically in Fig. 1. This framework shows a hierarchical control structure with a world model hierarchy explicitly interspersed between the sensor processing hierarchy and the behavior generation or task decomposition hierarchy. Example labels for three of the levels (subsystem, primitive, and servo), per [1] are shown. Note that the subsystem level for locomotion is referred to as "Autonomous Mobility" in 4D/RCS implementations.

Within RCS, there are three distinctly different types of knowledge: system parameters at the servo level, maps, images and object models at the next levels, and symbolic data at the highest levels. We briefly describe each of these.

3.1 System Parameters

The lowest level for RCS, as for any control system, is the servo level. At the servo level, position, velocity, and/or torque are controlled by voltage values applied to motors or valves. Knowledge of the value of system parameters is needed to control these values. Control knowledge, such as gains and filter coefficients, is typical of the type of parametric knowledge common at this level. These are commonly represented as scalars.

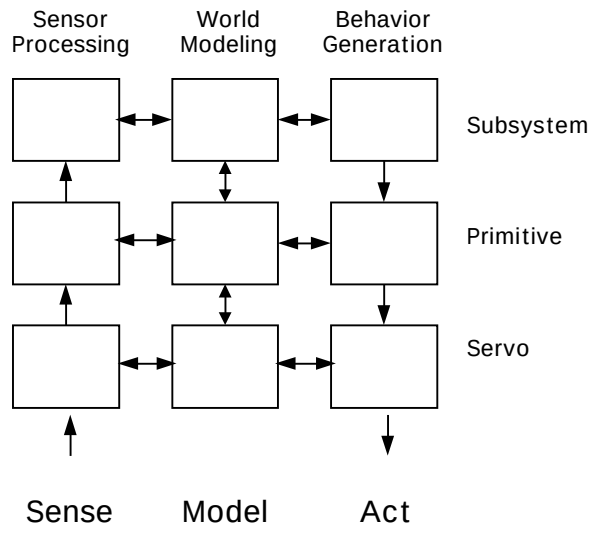


Figure 1: General Framework for Intelligent Control

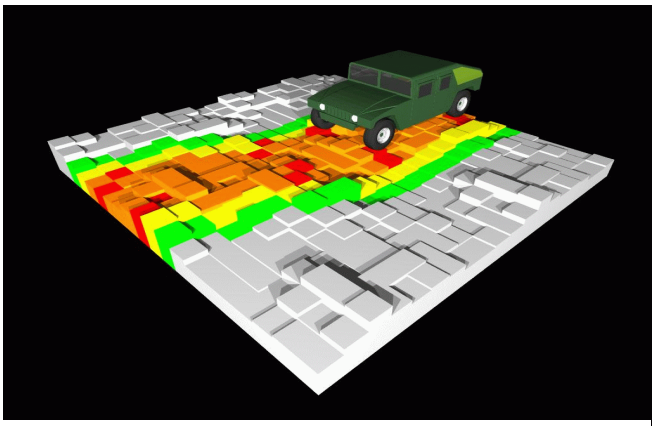
Any errors that deal with a single degree of freedom, such as ball screw lead errors, contact instabilities, and stiction and friction are best compensated for at this level.

3.2 Iconic Knowledge

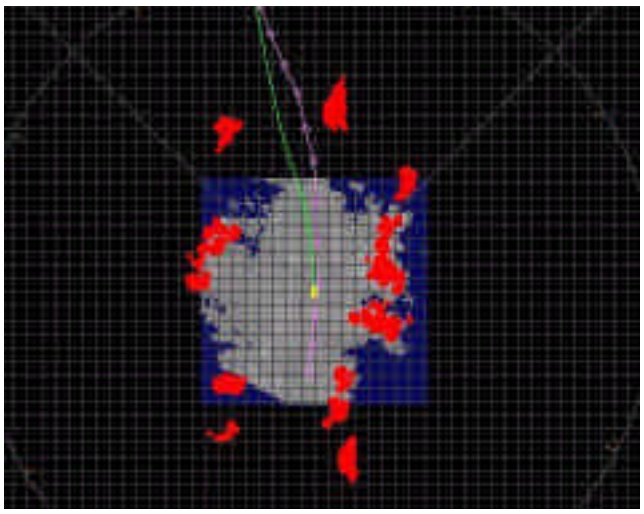
Multiple individual servo loops are coordinated at the next higher level. Interaction between axes comes into play, requiring knowledge of spatial dimensions, which we refer to as geometric or iconic knowledge. Iconic knowledge typically represents Euclidean space and includes maps, images, part models, and other geometric information. The relationship of the entities in time and space is captured through maps, images, and trajectories. Motion control for machine tool axes is computed at this level.

For mobile autonomous robots, maps are a natural representation for the environment in which the robot must function. Maps are defined as any two (or higher) dimensional grid with attributes referenced to the grid. A simple occupancy grid may indicate whether a cell is free or not (or passable or impassable by the robot) and the path planning algorithms will use shortest distance between start and goal cells, while avoiding impassable cells. A more sophisticated world model for an outdoor mobile robot may include a variety of feature layers, such as road networks, hydrology, elevation, intervisibility, and vegetation. The various features must be taken into account when planning movement and combined according to a weighting scheme based on the mission of the robot and current situation.

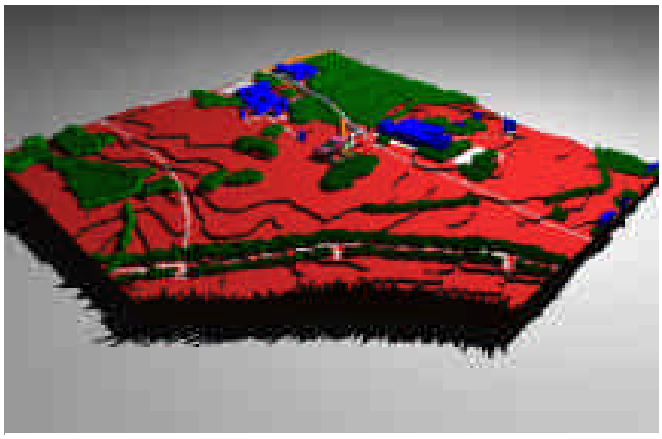
Maps used by an implementation of an outdoor mobile autonomous robot based on 4D/RCS are shown in Fig. 2.



a) Primitive Level



b) Autonomous Mobility Level



c) Vehicle Level

Figure 2: Maps at 3 Levels of 4D/RCS

[3] [19] Each level of the hierarchy concerns itself with a different spatial and temporal extent and resolution. The values listed below are representative examples for an implementation and may vary based on the computing configuration, sensors, and features supported. The features that the map contains at each level are also different, based on the area of focus for that level's planning.

Fig. 2a shows the map at the Primitive level, where planning for the robot's motion takes into account the kinematics and dynamics of the vehicle. The Primitive level of the hierarchy plans at roughly 10 Hz frequency and within a space of 5 m surrounding the vehicle (which is centered in the map) and a resolution of 20 cm. or less. This level of the hierarchy simulates the movement of the vehicle along potential obstacle-free paths and evaluates the position of the 4 wheels as they are placed along the trajectory to find the most traversable path. Terrain elevation is evaluated from range data provided by the Laser sensor, enabling computation of how stable and how rough a given path would be.

The next level up, referred to as Autonomous Mobility, plans at a frequency of 4 Hz within the 50 m surrounding the vehicle (which is again, centered in the map), with a resolution of 40 cm. Generally, this level of the hierarchy is concerned with avoiding obstacles and hazards to the navigation of the vehicle. The features that are contained in the map at this level include obstacles, cover, and roads obtained from sensory processing. Fig. 2b shows a combined Primitive level and Autonomous Mobility level map. The central square shows elevation (gray), unseen areas (blue) and obstacles (in red) detected by processing input from the vehicle's laser scanner sensor. The obstacles propagate to the Autonomous Mobility map (outside the blue and gray square). Not shown in the Fig. 2b are the precomputed feasible trajectories for the vehicle, given a starting wheel angle and velocity. The feasible trajectories that are blocked by obstacles are eliminated from consideration. Computing them offline enables the system to efficiently produce kinematically and dynamically stable steering commands.

Fig. 2c shows an example of the highest level currently implemented, the Vehicle level. This level plans within a map that is 500 m square, at a 4 m resolution, once a second. Planning at this level is concerned with generating a path between the current location of the vehicle and its goal point(s) (the operator may have specified certain waypoints or just an end location) while taking into account mission requirements. The paths generated for a mission that is stealthy versus one that gives highest priority to speed are completely different, yet the world model and the planner utilized are identical. Only the cost functions that are applied to evaluating candidate paths change. The features represented at the Vehicle level include road

networks, water, vegetation, elevation, risk (for each grid in the map, which locations can see that grid) and visibility (for each grid, which other locations can be seen). Features are typically obtained from *a priori* digital terrain maps.

3.3 Symbolic Knowledge

At the highest levels of control, knowledge will be symbolic, whether dealing with actions or objects. A large body of work exists in knowledge engineering for domains other than control, such as formal logic systems or rule based expert systems.

At the present time, symbolic knowledge has not yet been implemented in the vehicle application of RCS, but it has in manufacturing ones [17][12]. An example of a symbolic description of a solid model of a block is shown in Fig. 3. The description notation is the International Standards Organization Standards for the Exchange of Product Model Data (STEP) Part 21 [10]. Symbolic representations such as this have been used to automatically generate manufacturing process plans from part models [12]. Reasoning about a pocket feature is appropriate at higher levels of process planning. This is in contrast to having to jump directly to the geometric representation and try to derive appropriate machining sequences based solely from the surfaces of the final part geometry.

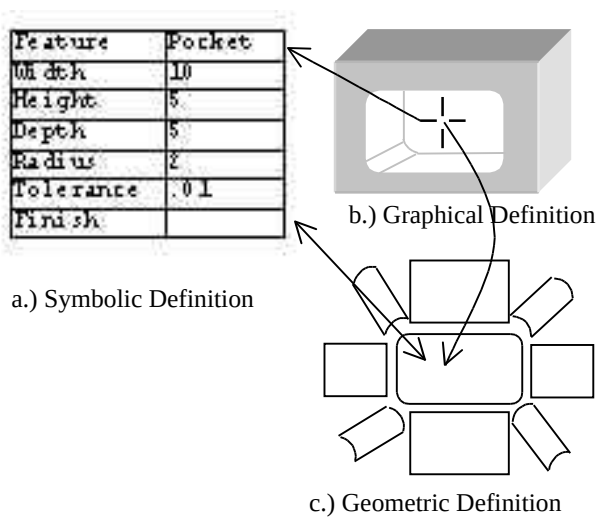


Figure 3: Pocket Feature.

Linguistic representations provide ways of expressing knowledge, expressing relationships, manipulating knowledge, and of extracting new knowledge based on knowledge already expressed, including the ability to address objects by property. Behaviors can be efficiently captured through symbolic representations. For example, in an autonomous vehicle system, entities such as "cars," "pedestrians," and "bicycles" each have certain properties and anticipated possible behaviors that affect the autonomous vehicle's planning *vis a vis* these other entities. A car can be expected to travel only on roadways (in normal circumstances) and to generally stay in a lane, whereas pedestrians may be expected to traverse roadways. Bicycles may squeeze between cars and straddle two lanes. The symbolic representation for each of these can be used in an intelligent system to derive potential behaviors in the near future and in the proximity of the autonomous vehicle. The symbolic entities may therefore be used to populate a map layer, such as the ones described in Section 3.2, based on current state information and expected potential behaviors. Higher level symbolic knowledge drives map-based (iconic) world model representations.

3.4 Other Dimensions in Knowledge

Another distinction within RCS is whether knowledge has been programmed into the system, is accessed from longer-term stores (*a priori* knowledge) or if it has been acquired or learned by the system recently during its operation (*in situ* knowledge) [17]. This distinction provides a framework for considering learning and adaptive control.

A final differentiation is in terms of whether knowledge pertains to things (nouns) or actions, task, or behaviors (verbs). This is akin to the distinction that the ancient Greeks made regarding "knowing that" versus "knowing what." System designers can make use of this distinction when matching sensor processing and world model specifications to the control task specification. This becomes very useful at higher levels in considering the interaction of autonomous machines with complex environments, where appropriate behaviors depend upon the nature of the objects encountered in the environment [2]. Generative process planning for machining or inspection [12] makes use of this distinction. Representations of actions will require a temporal element, unlike representation of things. An event has a time associated with it such as start, end, or duration.

4. EVALUATING KNOWLEDGE AND REPRESENTATION

Several obvious challenges exist in evaluating the knowledge that a system contains. It is difficult to isolate the world model from the sensing functions that populate and update it. The content and quality of the world model is dependent on the sensors and processes that are external to it. It is similarly difficult to separate the contribution of the world model independently from the planning subsystems that use it. There may be a very complete and efficient world model, yet the planning algorithms may be mismatched with it, poorly implemented, or inefficient.

Although it will be challenging, quantitative measures of the efficiency, completeness, and effectiveness of the representation must be developed.

Some may argue that, if a system works correctly, the particulars about the implementation are of no consequence. This is a shortsighted view of the science and engineering of intelligent systems. In order for the field to progress, successful and not so successful experiences must be shared. In this way, the capabilities of a system can be known and the best approaches can be leveraged by others in order to "raise all boats."

There are several aspects of knowledge content and representation that can be evaluated in an intelligent system, for which the community should strive to develop quantitative measures. We briefly present a few examples of evaluations without claiming this list to be exhaustive.

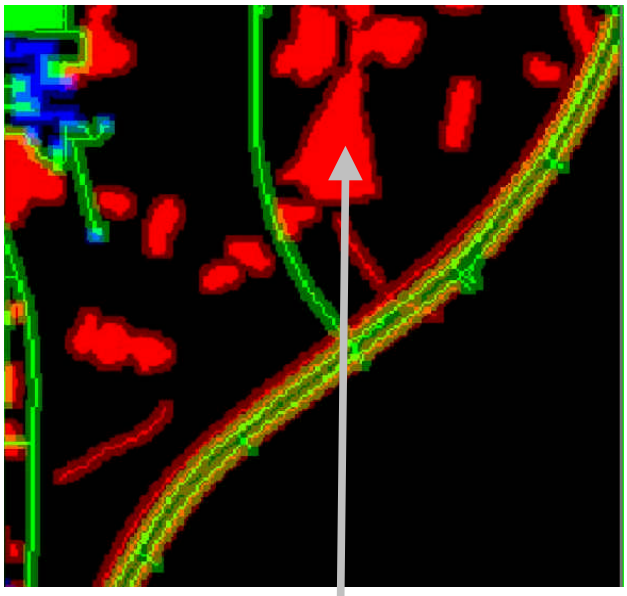
- The systems's ability to use *a priori* knowledge, and update it with newly-acquired knowledge. It is vital for most applications that the system start performing its tasks with given knowledge. That may take the form of maps of the area where an autonomous vehicle is expected to drive, a catalog of available cutter tools for machining, or an ontology to facilitate natural language interaction. When operating in the world, the intelligent system will have to sense changes in its environment and update its internal models. The new knowledge has to be placed in context of existing knowledge. Obstacles encountered during movement have to correctly update *a priori* maps. Tools that are no longer available must be deleted from the local copy of the tool catalog. Idioms or new terminology must be integrated into the language ontology.
- Mapping the environment in order to accomplish the given task. For a system that operates in the physical world, a current representation of its surroundings is crucial. Therefore, the system must be evaluated for its ability to understand and interact with a dynamic environment, including moving objects.
- Understanding general as well as specific concepts. Humans can accommodate thinking about the abstract and the concrete. Intelligent systems need to know about general classes of entities, such as "elevator" in addition to specific instances of elevators that they have to interact with. All elevators can be used to travel between floors, but the user interfaces for specific instances vary considerably. Another example is the concept of window, which may be important to a military scout robot. The general concept is important as it plans to look for windows during its mission. When it recognizes objects that fit that category, it must then plan its actions with respect to the specific instances. Windows may or not be see-through. They may be used to enter a building, but the robot needs to realize that windows at higher floors may not be useful for entering a building (unless the robot can scale the walls).
- Dealing with incomplete and imperfect knowledge. The system must accommodate and reason about partial and incorrect information about its environment. If not, it will rapidly be unable to cope.
- The correctness of the knowledge that a system holds. The system should be able to store *a priori* (given) knowledge correctly and be able to acquire correct knowledge. Correctness measures may be based on validation against ground truth or they may be evaluated based on confidence values based on multiple or redundant sensing.
- The efficiency of the knowledge representation. There are always many alternatives when implementing a system. The general representation approach (e.g., symbolic versus iconic) for a particular category of knowledge is one coarse aspect that can be examined. It may only be necessary for a system to store a structure that defines an entity as a tank and includes high level definitions such as min-max dimensions, make, model, friendly/foe, rather than an occupancy grid in three dimensional space or a solid model of the tank's geometry.

Once the dimensions of knowledge and representation that are to be evaluated are identified, the actual evaluation process is still a challenge. In this emerging new technology of intelligent systems, there are few examples of evaluation procedures that specifically target the knowledge itself, as opposed to the overall system performance. One of the key aspects of evaluations is that they be accurate and reproducible. We will describe some possible approaches to address these requirements.

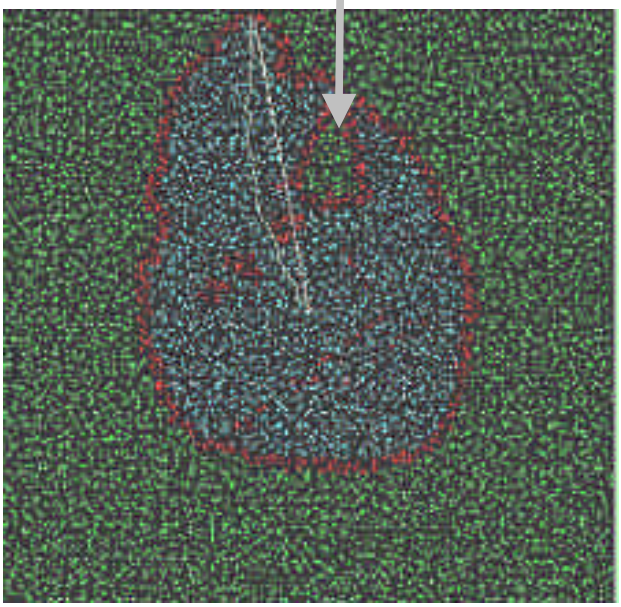
Test arenas and scenarios are already being used to test robotic system capabilities. Examples include RoboCup [11][22] and the American Association for Artificial Intelligence Competitions, such as the Urban Search and Rescue Robots and Hors d'Oeuvre Anyone [21]. In the urban search and rescue competition, robots enter arenas that represent a collapsed building and search for targets that represent victims and hazards. The robots are supposed to communicate to human supervisors the locations of each victim and hazard. This requires at minimum the ability to map the environment and localize objects within the maps. The competition arenas have second stories, hence a good representation scheme would accommodate a third dimension. An excellent competitor would produce a map of every area explored, not just coordinates of the targets.

Virtual test environments and simulators can also be used to glean the knowledge representation aspects of intelligent systems. A virtual environment is one in which an organization can "plug in" their software and have the intelligent system, such as a mobile robot, receive simulated

inputs from the environment and compute outputs to the virtual actuators. The level of interfaces from and to the virtual environment may be high level or, for high fidelity systems, could be equivalent to the interfaces to the actual sensors and servos. Isolating the world modeling databases and processes becomes feasible with the right simulation or virtual environment.



a) Map with a priori features



b) Node Status after Planning Cycle

Figure 4: Correspondence between planning space and physical features

Test harnesses that can be hooked up to knowledge bases can be used to evaluate its contents. A knowledge base that has been functioning and updating as an intelligent system performs its tasks can be isolated, either after the tasks are completed, or at certain points during operation. The harness can be used to query the contents of the knowledge base. For instance, it can check what entities have been detected in the environment and where they were estimated to be located. A harness would require defining or making known interfaces to the knowledge base.

5. KNOWLEDGE REPRESENTATION MATTERS

In this section we very briefly present examples of how the type of representation chosen for knowledge can affect the capabilities and effectiveness of a system. The examination of these examples is cursory and is meant to stimulate thought.

The first example is a classic taken from [20]. As an introductory exercise, a checkerboard, eight by eight squares, is to be covered by rectangular tiles. Each tile covers exactly two of the squares in the checkerboard. How many tiles are needed to completely cover the board? The solution is obvious (64/2=32) and can be easily found by a computer algorithm that searches through a grid-based representation of the checkerboard. Now, take away 2 of the squares, one from the top left corner and one from the bottom right. 62 squares remain, so one might naively assume that 31 tiles should be able to cover the remaining squares. The computer program that performs a search will have to expend a lot of compute cycles and may not be equipped to confront the fact that with this geometric configuration, there is no solution that fully covers the board with tiles. A different representation is better suited to quickly reach the correct conclusion. If the board is viewed as 2-tuples of black and red squares, since two same color squares can never be adjacent, then a tile covers each tuple of exactly one red and one black square. The missing corners took away 2 squares of the same color, hence there are more squares of one color than the other. Given this perspective, it is impossible to cover the board completely with tiles.

A second example is taken from [4]. In Balakirsky's system, a graph representation is used to solve planning problems. The LAyered World Modeling and Planning System (LAWMPS) has been applied to path planning for autonomous military vehicles. The world model in LAWMPS consists of a set of layers, organized in a grid representation. Each layer is dedicated to a particular feature, such as roads, vegetations, buildings, and sensed obstacles. The cost map is built by computing the contribution of each layer to the cost of having the vehicle traverse that location. The cost weights, which control the

contribution of each feature are variable and determined by user preferences, modes, and objectives. A subset of the grid locations is used to generate the nodes and arcs for the planning graph. The planning process proceeds on the resulting graph, where each node represents a location, and the arcs have costs associated with moving between two specific locations.

Having the graph connect nodes that align with a vehicle-centered map grid and applying a Dykstra search algorithm can lead to discovery of knowledge that is useful to a mobile robot. "Problem" areas in the graph (where the search essentially stalls) as the search progresses can be correlated with map features and used to extract rules about traversability or other aspects of the problem state. In Figure 4a, an *a priori* map is shown with trees and fences (red), buildings (blue), and roads and parking lots (green). Figure 4b shows the node states after a cycle of planning. Green ones have never been visited, blue ones are closed (all their children have been visited), and red ones are still open). Due to the spatial relationship between the planning space and the *a priori* maps, the correspondences are clear: one area that appears problematic in the graph space is shown to correspond to a fenced or treed area, which would be impassable by the vehicle. Balakirsky uses this correspondence to allow the system to learn rules about planning.

6. CONCLUSIONS

Knowledge content and representation are critical aspects of an intelligent system. In constructing intelligent systems, there is a need for more science and engineering in the area of what should be represented and how it should be represented. Work in the area of knowledge representation has not, for the most part, addressed the area of real-time intelligent control. We argue that there are several categories of knowledge and types of representations that are necessary within a system that demonstrates advanced capabilities. Much work still needs to be done in understanding how to capture, use, and build knowledge within these systems. It is imperative to capture quantitative data about systems that demonstrate intelligence so that the field can benefit and move forward.

REFERENCES

[1] Albus, J.S., Lumia, R., Fiala, J., Wavering, A., "NASREM -- The NASA/NBS Standard Reference Model for Telerobot Control System Architecture"," Proceedings of the 20th International Symposium on Industrial Robots, Tokyo, Japan, October 4-6, 1989.

[2] Albus, J., "4-D/RCS: A Reference Model Architecture for Demo III," NISTIR 5994, Gaithersburg, MD March 1997.

[3] Balakirsky, S. and Lacaze, A., "World Modeling and Behavior Generation for Autonomous Ground Vehicles," Proc. Of the IEEE International Conference on Robotics and Automation," April 2000.

[4] Balakirsky, S., "A Layered World Model for Intelligent Planning," journal article in review.

[5] Bauer, M. I. and Reiser, B., "Incremental Envisioning: The Flexible Use of Multiple Representations in Complex Problem Solving," Proceedings of the 12th Conference of the Cognitive Science Society, Cambridge, MA,. 1990, pp. 317-324.

[6] Chittaro, L., Guida, G., Tasso, C., Toppano, E., "Functional and Teleological Knowledge in the Multimodeling Approach for Reasoning about Physical Systems: A Case Study in Diagnosis," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 6, November/December 1993.

[7] Davis, R., H. Shrobe, et al. 1993. "What is in a Knowledge Representation?" AI Magazine, Spring 1993.

[8] Devedzic, V., "A Survey of Modern Knowledge Modeling Techniques," Expert Systems with Applications 17 (1999) 275-294.

[9] Fikes, R. and Nilsson, N., STRIPS: a new approach to the application of theorem proving to problem solving," ARTIFICIAL INTELLIGENCE, 2(3-4): 189-208, 1971.

[10] ISO 10303-21: Industrial Autonmation Systems And Integration—Product Data Representation And Exchange—Part 21: Clear text encoding of the exchange structure. Geneva, Switzerland: ISO 1994:1994.

[11] Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H., "Robocup A Challenge Problem for AI," AI Magazine 18(1): Spring 1997, 87-101.

[12] Kramer, T., Huang, H., Messina, E., Proctor, F., Scott, H., "A Feature-Based Inspection and Machining System," Computer-Aided Design; Vol. 33; August 2001; pp. 653-659.

[13] Kuipers, B., "The Spatial Semantic Hierarchy," Artificial Intelligence 119 (2000) 191-233.

[14] Kunz, C., Willeke, T., Nourbakhsh, I., "Automatic Mapping of Dynamic Office Environments," Autonomous Robots Journal. 7(2) 1999.

[15] Levesque, H., Reiter, R., Lesperance, Y., Lin, F., Sherl, R., "Golog: A Logic Programming Language For Dynamic Domains," Journal of Logic Programming, 31:59-84, 1997.

[16] Lewis, H. and Denenberg, L., Data Structures and Their Algorithms, Addison Wesley, 1991.

[17] Messina, E., Horst, J., Kramer, T., Huang, H.M., Tsai, T.M., Amatucci, E., "A Knowledge-Based Inspection Workstation," Proceedings of IEEE Intn'l Conf. on Intelligence, Information & Systems, Bethesda, MD, Oct 31-Nov 3, 1999.

[18] Moravec, H., and Elfes, A., “High Resolution Maps from Wide Angle Sonar,” Proceedings of the 1985 IEEE International Conference on Robotics and Automation, St. Louis, MO, 1985.
[19] Murphy, K., Abrams, M., Balakirsky, S., Coombs, D., Hong, T.H., Legowik, S., Intelligent Control for Unmanned Vehicles, Proceedings of the World Automation Congress Conference (WAC 2000), Maui, HI, June 11-16, 2000.
[20] Newell, A., and Simon, H., "Computer Science as Empirical Inquiry: Symbols and Searc," Communications of the ACM, Volume 19, No. 3, pp. 113-126, March, 1976.
[21] Schultz, A., “The 2000 AAAI Mobile Robot Competition and Exhibition,” AI Magazine 22(1), Spring 2001, 67-72.
[22] Stone, P., ed., “RoboCup 2000: The Fourth Robotic Soccer World Championships,” AI magazine 22(1), Spring 2001, 11-38.
[23] Thrun, S., Fox, D., Burgard, W., “Probabilistic Mapping of an Environment by a Mobile Robot,” Proceedings of the 1998 IEEE International Conference on Robotics & Automation.
[24] Wah, B., Lowrie, M., Li, G., “Computers for Symbolic Processing,” Proceedings of the IEEE, Vol. 77, No. 4, April 1989, 509-540.
[25] Yeap, W. K., Jefferies, M. E., “Computing a Representation of the Local Environment,” Artificial Intelligence 107 (1999) 265-301.