Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop,
Gaithersburg, MD, August 13-15, 2002.

Towards Measuring the Performance of Architectural Components of Autonomous Vehicular Systems

Craig Schlenoff, Lawrence A. Welsch, Raj Madhavan, Noah Zimmerman
National Institute of Standards and Technology
craig.schlenoff, lwelsch, raj.madhavan, noah.zimmerman @nist.gov

Abstract

For a vehicular system to act "intelligent", the system must be able to 1) sense in a dynamic domain; 2) model the domain internally; 3) determine possible courses of action to accomplish a goal in the domain; and 4) be able to assess the various courses of actions to determine which is best. The actions that the system ultimately performs are a function of all of these components. Solely assigning performance metrics to the resultant action of the intelligent system does not evaluate any one of these components individually, and therefore leaves some doubt as to how to measure what each component contributes to the overall behavior of the system. Thus we are not looking at a single number, but a matrix of numbers that characterize the performance of the system.

In this paper, we are exploring a mechanism to assign performance metrics to the part of the system that models the domain internally, the internal knowledge representation of intelligent vehicular systems. We do not consider that part of a system that translates the raw sensory input from a vehicle's sensors to other representations. Rather we simulate a predefined set of sensory inputs, and evaluate the resulting knowledge representation based.

1 Introduction

Darwin was the first to propose the importance of natural intelligence for biological entities. He suggests that intelligence is the result of billions of years of natural selection, emerging from a competitive struggle for survival [1]. Measuring the *intelligence* of intelligent systems presents several challenges. A universal scalar value of intelligence is difficult to ascertain in a machine due to the restrictive nature of most domains.

Additionally, it is more difficult to make judgments based on the relative success of particular behaviors. However, in machines we have the advantage of being able to monitor the internal states. This enables us to make more accurate deductions about 1) the methods employed by the system to complete the task, and 2) the intermediate states that it traversed. The system can then be evaluated based on a relationship between the complexity and efficiency of the method and the precision of the final state.

There have been attempts to provide qualitative and quantitative measure to knowledge representations [10], though not, until recently, have they been applied to measuring the internal knowledge representations within autonomous vehicular systems. Gruninger and Fox have applied the concept of competency questions to formal ontologies to test their ability to answer the questions they were designed for [8]. McGuinness et al. have also explored approaches to testing the content of ontologies after multiple ontologies are merged by using a tool called Chimaera [9]. More recently work has been done to develop tests for text retrieval systems [11] and autonomous vehicle systems [12]. Research has also been done considering the performance of rule chaining in generic expert systems [13]. In this paper we are considering how to best take advantage of Real-Time Control System[1] architecture (described below) to measure the performance of the architectural components that contribute to the vehicle's behavior.

For a vehicular system to act in an intelligent manner, the system must be able to 1) sense in a dynamic domain; 2) model the domain internally; 3) determine possible courses of actions to accomplish a goal in the domain; and 4) be able to assess the various courses of actions to determine which is best. The actions that the system ultimately performs are a function of all of these components. Solely assigning performance metrics to the resultant action of the intelligent system does not evaluate any one of these components individually, and therefore leaves some doubt as to how to measure what each component contributes to the overall behavior of the system.

We have selected the Real-Time Control System (RCS)[1] as the architecture for evaluating intelligent systems. RCS is a hierarchical distributed real-time control system architecture that allows for modular and device independent algorithms to be developed for intelligent systems. A node in the RCS reference model architecture is shown in Figure 1.

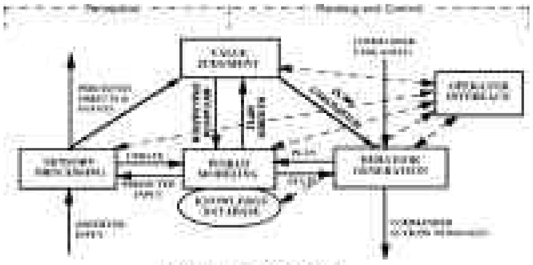

Figure 1: RCS Node

The functional elements of an intelligent system can be broadly considered to include: behavior generation (task decomposition and control), sensory processing (filtering, detection, recognition, grouping), world modeling (store and retrieve knowledge and predict future states), and value judgment (compute cost, benefit, importance, and uncertainty). These are supported by a knowledge database (KD), and a communication system that interconnects the functional models and the knowledge database. This collection of modules and their interconnections make up a generic node in the RCS reference model architecture. Each module in the node may have an operator interface.

Though several contemporary architectures exist in the literature for designing intelligent systems, our motivation for selecting RCS is many fold:

- In the last fifteen years, behaviorist architectures [2] [3] have gained popularity for their ease of implementation. However, within such architectures, long-term planning is not possible since only a single behavior can be selected for execution. Other disadvantages include the inability to fuse sensor data to arrive at a single best estimate of the state of the world (in some probabilistic sense) and the lack of internal representation of the world.
- RCS is a proven architecture with more than 200 person-years of research and development in intelligent control theory. It has been implemented and tested thoroughly both in the industry and academia in different operating domains under varying operating conditions. For example, RCS has been implemented as the reference model architecture for the design, engineering, integration, and testing of eXperimental Unmanned Vehicles for the DoD Demo III program [1] [4].
- RCS is supported in terms of software and updates and thus it constantly evolves through a number of versions at National Institute of Standards and Technology (NIST) and elsewhere [5]. For additional advantages, see pp. 128 of [6].

For the purpose of this paper, we are exploring a mechanism to assign performance metrics to the part of the system that models the domain internally, the internal knowledge representation of intelligent vehicular systems. We hold the sensory component constant and do not consider the behavior and value judgment components. In other words, we simulate a pre-defined set of sensory inputs, and evaluate the knowledge representation based on those sensory inputs. There would be no actions physically performed, nor would there be any value judgment implemented. In this paper, we explore developing a test harness for autonomous systems, focusing on each combination of knowledge representation components and functions. Thus, the test harness can be seen as a matrix, with the components along one axis and the functions along the other, and each cell composed of a series of questions testing the knowledge representation's ability to provide the stated function using the pertinent component, if appropriate. For example, a question such as "Where do you expect a given moving object to be at time=10?" may be appropriate to test the intelligent system's "prediction" function using its "inferencing" and "knowledge being represented" components.

In Section 2, we discuss a test harness, including the data flow through the harness and the places in the RCS hierarchy that would be appropriate to test. Section 3 discusses the typical purposes/functions of a world model. Section 4 describes the components of any knowledge representation, and discusses pertinent questions that could be asked to test those components of the knowledge representation. Section 5 brings the previous two sections together into a matrix, and discusses future work that should be done to address the development of the proposed test harness.

2 The Test Harness

2.1 Data Flow

The goal of this work is to test the world modeling capabilities of an autonomous vehicular system without requiring the system to be physically relocated to a test site, nor to require that the system have to perform any physical behaviors. The system’s world modeling capabilities would be tested by a series of questions and answers, where the answers to the questions would be assigned a score based upon a series of performance

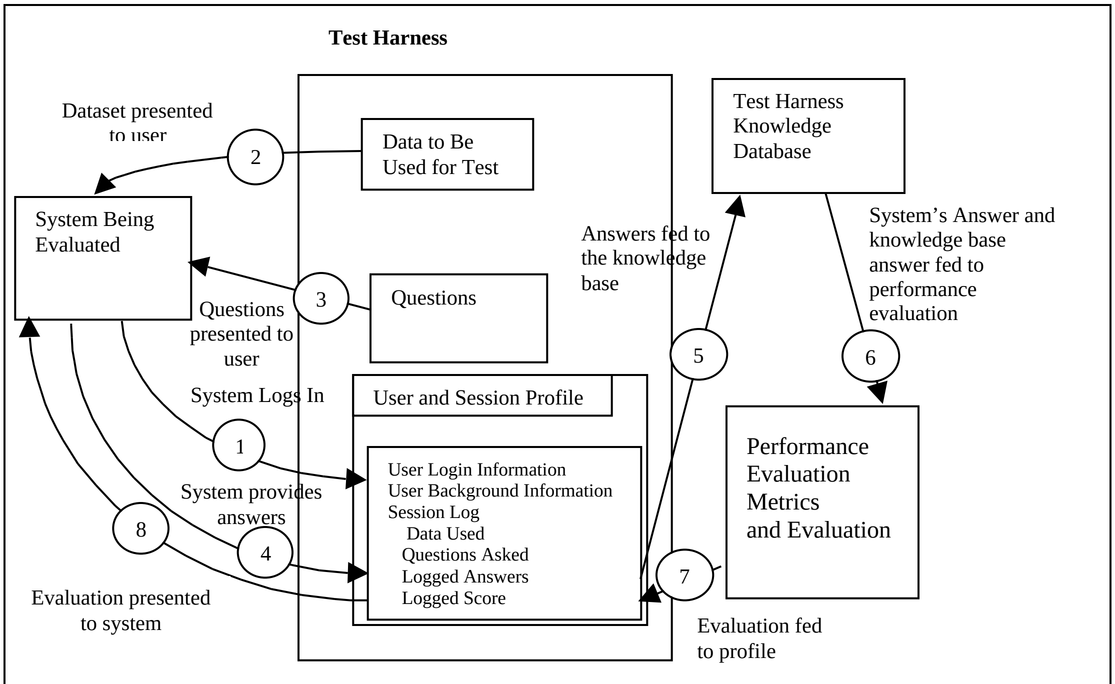

Figure 2: Data Flow in the Test Harness

evaluation metrics. Figure 2, along with the supporting text, shows the data flow pertaining to the interaction a system would have with the test harness, and is described in detail.

Figure 2 contains three main components: the system being evaluated, the test harness, and the knowledge base / performance evaluation components. The test starts when the 'system being evaluated' first registers by entering in its ID and password (number 1 in Figure 2). At this point, the user can choose between a series of sample sensory data to use for the test, sorted and rated by its level of difficulty (to be discussed in a future paper) (2). The system then has a predetermined amount of time to receive and process the data. After the data is processed, a series of questions that correspond to that data set are posed to the user (discussed in Sections 3 and 4) (3). These questions may also be rated by their level of difficulty. The user responds to these questions by providing an answer, as well as a description of how that answer was determined (4). This information, along with the amount of time that was taken to determine the answer, is noted in the user's profile. This information is the passed to the test harness knowledge base where it is compared with system's knowledge base's response to the same questions (5).

The answers from the systems and the knowledge base are then passed to the evaluation component, where predetermined metrics are used to assign a score to the system's answer (6). The score would be a function based upon the "correctness" of the answer (e.g., the answer was two, but the system thought the answer was four), the procedure used to come up with the answer (e.g., what were the equations used and the assumptions made when the answer was being determined), the amount of time it took to produce the answer, and the amount of detail provided in the answer (e.g., the answer was two, but the system responded with an answer of "between one and five"). This score is then fed back to the system's profile to be logged (7), and reported to the system (8).

There are many interesting and challenging research areas within the scope of this framework, including the types of sensor data to be presented to the user, the types of questions that should be asked to the user in response to the sensor data, the information to store in the knowledge base to evaluate the answers the system provides, the appropriate evaluation metrics to use in evaluating the answers (including the weights to put on each of the factors described in the previous paragraph), the details of the communication specifications between the system and the test harness, the interfaces and the representation of the information to be passed between the various internal components of the framework, as well as the mechanism to allow a system to supply an explanation of how an answer was produced. This paper focuses solely on the questions that are asked on the system being evaluated. Future papers will focus on the other challenges mentioned above.

2.2 Applying the Test Harness to Various Components in RCS

The test harness described above is generic and may be used to test an entire node in the RCS hierarchy (as shown in Figure 1) or just a component of a node. If the entire node is being tested, then raw sensory data would be fed to the "systems being evaluated" as input (as indicated by the bottom left arrow entering the box) and the output plan of the RCS node (as indicated by the bottom right arrow exiting the box) would be evaluated.

Instead of looking at the entire RCS node, one could only test one or more components of the node, thus focusing the attention on only a small subset of the node. In this paper, we are interested in the contribution of the World Model / Knowledge Database component as shown in Figure 3 below. In this case, we would be feeding processed sensory data to the world model (thus the sensory processing is not considered), and can query the world model about what it perceives, where is expects objects to be in the future, etc. (thus the planning is not considered since the world model is never asked to generate a plan, it is just asked to answer questions about what it is presented).

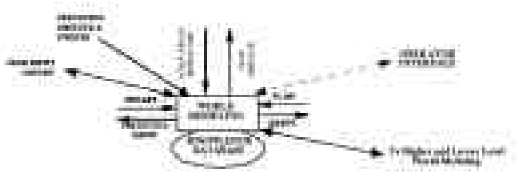

Figure 3: World Model and Knowledge Database

Although this paper solely focuses on applying the proposed test harness to systems based upon the RCS architecture, there is nothing in the design of the test harness that precludes it from being applied to other systems. The only assumption that this test harness design makes is that there is a clear place in the "system being evaluated" to which information can be fed, that there is a clear place in the "system being evaluated" from which information can be read, and there is an appropriate set of questions and evaluation metrics which can be applied to evaluate the system.

In the next section of the paper, the functionality of the test harness is exposed and test interactions proposed.

3 Functions of a World Model

The world model can be thought of as a component of the brain of the intelligent system. Just as the brain contains a representation of the environment, the world model contains a representation of its surroundings, and as such, must be able to use that representation to the benefit of the system that is immersed in that environment. The world model must inform the intelligent system on the potential results of action, similar to the way the brain informs the human body of the possible consequences of actions.

The world model can be thought to be comprised of four functions: maintenance and updating of the knowledge base, prediction of sensory input, response to queries for information required by other processes, and simulation. This is described in detail in [6]. In this section of the paper, we will provide examples of the types of queries that the world model would be expected to answer to perform these functions.

3.1 Maintenance and Updating of the Knowledge Database

The world model in its entirety is the intelligent system's best estimate of the world at the given time. The world model can be thought of as comprising a number of knowledge databases, where each knowledge base is a store of information about the world.

To ensure that the representation of the world is up-to-date, the world model must constantly be updated as new information is available. Examples of ways that the world model could be updated include:

1. As new processed sensor data is available and entities are identified, the world model must compare the actual location of the sensed images to the location in which the world model predicted that it would be. (What is the difference between the actual location of entity A and the predicted location of entity A?, How can the current prediction parameters be changed to provide more accurate predictions?)
2. As time elapses, information will move from immediate experience to short term memory, to long term memory. The world model must seamlessly allow for the migration of information into these parts of the world model, as well as transform the representation

of this information between different representation approaches. (What information should be moved to short-term memory? To long-term memory?)

3. As time elapses, new entities will appear in the intelligent system's environment, and some entities will no longer exist. The world model must be able to introduce these new entities into the knowledge database, determine which ones are most important to track, and delete those entities that no longer exist or are no longer of interest. (What new entities exist that were not previously modeled in the knowledge base? Which of these entities are important to track? What are the pertinent characteristics of those entities? What are the criteria for deleting entities from the knowledge base?)
4. In the real world, relationships exist between entities, events, and situations. It is important to maintain these relationships within the world model. (What are the important relationships in a given environment?, How should those relationships be represented?, For what time extent do those relationships hold?)

3.2 Prediction of Sensory Input

In addition to capturing the data that is passed to it by the sensors, the world model must also predict where it believes the next set of sensed data will be. Being able to accurately predict where an object is expected to be at a time in the future is essential for areas such as image processing, path planning, and collision avoidance. Accurate prediction algorithms allow the world model to better predict where an object is expected to be at some time in the future, along with a stated degree of uncertainty, and therefore make plans that account for that predicted future location.

Questions that may be asked within this function of the world model include "What is the predicted location of entity A given data pertaining to its previous location?", "What are the appropriate algorithms to provide the prediction?", "What are the criteria for updating the prediction parameters?")

3.3 Response to Queries for Information by Other Processes

The world model is the primary source for information within the intelligent system. It is designed to be an information repository, and as such, must interface with other components of the hierarchy that have a need to retrieve information from it, whether explicitly or implicitly represented. More specifically, the world model provides the following functions:

1. The world model responds to requests from the sensory processing, behavior generation, and operator interface components of the hierarchy. The sensory processing component may ask for the predicted attributes and states of an entity. The behavior generation component may request the predicted identity of entities in the environment, as well as characteristics of those entities (e.g., if the entity was a car, how fast is the car going? In what direction? What is the fastest the car can go?, etc.). The operator input may ask for the state of the intelligent system at the current time. (What is the predicted location, speed, orientation of entity A at time = t+1?, What is the object perceived by the sensors, and what are the pertinent characteristics of it?)
2. The world model performs coordinate transformations, when necessary, and accounts for the motion of the sensor platforms that affect sensor input.
3. The world model deduces additional information from the knowledge database that is not explicitly represented, but can be deduced from the information that is represented. (Given the information known about an object, what additional information can I infer about the entity that is not explicitly represented?)

3.4 Simulation

In almost any application, it is useful to simulate the results of an action before the action is physically performed. More specifically, the simulation aspect of the world model provides the following functions:

1. The world model uses the knowledge in the knowledge databases to simulate the results of possible plans generated by the behavior generation module.
2. The world model can compute all of the sets of actions which can be performed to produce a desired output.
3. The world model interfaces with the value judgment component to evaluate the cost/benefit of the proposed action based on the simulation (What are the appropriate cost algorithms?, Given a cost algorithm, what is

which plan provides the most benefit at the least cost?).

4 Knowledge Representation Measurements

The previous section described functions that the world model within an intelligent system is expected to perform. Based on those functions we posed queries that the world model are needed to support the functions. This section proposes measures for the knowledge database within the world model. By considering each measure against each query, we derive the matrix described in the conclusions.

The knowledge database can be thought of as having three attributes: 1) the formalisms for representing knowledge (i.e., how the knowledge is captured), 2) the actual knowledge the system has represented at any given time (e.g., the data that is captured within the knowledge database), and 3) the mechanism(s) available for accessing, querying, and inferencing over the represented knowledge. Each of these attributes provides a different set of measures, for the value that each brings to the overall world model.

4.1 Measuring the formalisms for representing knowledge

A KD may contain a variety of different types of formalisms for representing knowledge in its database. For example the KD may contain formalisms to represent:

- Raw sensory data collected directly from sensors;
- Map and/or geometric data where map data might provide coordinates for landmarks, roads, and topological features.
- Symbolic and/or rule data that might contain rules such as *drive on the right side of the road,* or *enter buildings through an opening*; and
- Links or associations between the different types of data.

When determining the metrics for measuring the formalisms for representing knowledge, one may consider the following criteria:

1. The number of different types of representations that the KD supports;
2. The complexity level the formalism can support. For example, in the case of symbolic representation, is the representation capable of representing Boolean algebra, first order predicate calculus, etc.;
3. The detail or granularity in which the fundamental physical units may be represented;
4. The size of the largest set that be represented – finite, countable, etc.; and
5. The number of mechanisms in which one can group knowledge.

Each measure can be considered for each question described in section 3. For example, for a particular query, the measure would be the number different types of representations of data that were involved in generating a response to the query.

4.2 Measuring the actual representation of the knowledge

At any given instant in time the world model has a set of information that is captured within its knowledge databases. One can measure the captured knowledge using the following types of metrics:

1. The quantity of different contexts/concepts[1] that are represented;
2. The quantity of contradicting knowledge, possibly organized by contexts;
3. The scale of complexity [7] of the most/least complex concept represented (not the complexity of the formalism, but rather the concept itself);
4. The numbers of links among concepts; and
6. The depth of the hierarchy tree (e.g., how many “levels” are in the representation?).

Again each measure would be considered against the each query described in section 3.

4.3 Measuring the mechanisms for accessing in and inferencing over the knowledge database

Finally we need to evaluate the performance of the mechanisms that respond to requests of the KD (the inference or query mechanism). The measures considered are:

---

[1] By concept/context is meant a collection of knowledge that is not self-contradictory. Frequently a context/concept is a way of organizing knowledge so as to make the knowledge easier to find.

1. The length of time[2] the system takes to find a particular fact, rule, assertion already in the KD.
2. The minimal time to combine a fact with an assertion;
3. The speed to switch representation formalisms (with and without links);
4. The minimum time to combine knowledge in one representation formalism with another; and
5. The quantity of different inferencing mechanisms that exist.

Again, each measure would be applied to each query. For example for the query *What is the difference between the actual location of entity A and the predicted location of entity A?*, the first measure would be, the minimal time to retrieve a fact necessary to addressing the query.

		Knowledge Representation Characteristics		
World Model Functions		formalisms for representing knowledge	knowledge being represented	mechanisms for information access and inferencing
	maintenance and updating			
	prediction of sensory input			
	response to queries for information required by other processes			
	simulation			

Table 1: Test Matrix

5 Conclusions / Future Work

In this paper, a test harness was introduced with an emphasis on the types of questions that would be needed to test the world modeling capabilities of an intelligent system. One can imagine a series of questions that would test certain expected functions of the autonomous system's world model, with respect to specific characteristics of the knowledge representation such as the way the knowledge is represented, the exact knowledge that is represented, and the mechanisms for querying that knowledge. These questions would logically fall into the matrix, as shown in Table 1, with specific questions tailored for each cell in the matrix.

Work has recently been started on implementing the framework of the test harness, using an agent-based infrastructure, in a web-based environment, such that the interaction with the test harness would be web-based calls with a web server located at NIST. However, much work remains to be completed.

As mentioned in Section 2, there are many interesting and challenging research areas within the scope of this test harness that have yet to be addressed, including:

- the types of sensor data to present to the user,
- the types of questions that should be asked to the user in response to the sensor data,
- the information to store in the knowledge base to help provide the information to evaluate the answers the system provides,
- the appropriate metrics to use in evaluating the answers (including the weights to put on each of the factors described in the previous paragraph),
- the details of the communication specifications between the system and the test harness, and
- the interfaces and the representation of the information to be passed between the various internal components of the framework.

However, for any of these components to be developed and tested, the overall framework must exist. Therefore, the development and implementation of the overall framework of the test harness, with initial black boxes for each of the individual components, is the first priority and thus is currently being developed.

Additional future work will focus on applying the test harness to other aspects of the autonomous system architecture (as discussed in Section 2). To be more specific, in this paper we only focused on testing the system's world model capabilities. However, we could expand the parts of the hierarchy being tested such that we allow the system to generate plans, and compare those plans to "optimal" plans as determined by the system's knowledge base which contains "perfect" world knowledge. We could also test the autonomous system's sensory processing components, by feeding in raw sensory data, and ask the

---

[2] Ideally, one would represent processing speed in independent unit, where the actual time could be based on multiplying the units by the appropriate processing speed factor.

autonomous system questions based on the processing of that data.

It would also be interesting to apply this test harness to other architectures besides RCS. Although, in theory, there is nothing RCS-specific about this architecture, it would be interesting to see how well the design holds up to other architectures for autonomous systems.

References

[1] Albus J., “4D/RCS: A Reference Model Architecture for Demo III, Version 0.1,” NISTIR 5994, National Institute of Standards and Technology, Gaithersburg MD, 1998.

[2] Arkin R., *Behavior-based Robotics*, Cambridge, MA: The MIT Press, 1998.

[3] Brooks, R. “A Robust Layered Control System for a Mobile Robot,” *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1988.

[4] Shoemaker C., and Bornstein J., “The Demo III UGV Program: A Testbed for Autonomous Navigation Research,” in the Proceedings of the IEEE ISIC/CIRA/ISAS Joint Conference, 1998, pp. 644-651.

[5] Gazi V., Moore M., Passino K., Shackleford W., Proctor F., and Albus J., *The RCS Handbook: Tools for Real Time Control Systems Software Development,* New York: John Wiley and Sons, 2001.

[6] Albus J., and Meystel A., *Engineering of Mind: An Introduction to the Science of Intelligent Systems*, New York: John Wiley and Sons, 2001.

[7] Ganter B., and Wille R., *Formal Concepts Analysis: Mathematical Foundations*, Germany: Springer, 1999.

[8] Gruninger, M., and Fox, M.S., “The Role of Competency Questions in Enterprise Engineering” in the Proceedings of the IFIP WG5. 7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway. June 1994.

[9] Deborah L. McGuinness, Richard Fikes, James Rice, and Steve Wilder. “An Environment for Merging and Testing Large Ontologies.” in the Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, USA 12-15 April 2000.

[10] A. Turing. Computing Machinery and Intelligence. *Mind*, 59, pp. 433-460. Reprinted in E. Feigenbaum and J. Feldman. (Eds.) *Computers and Thought,* McGraw-Hill, New York, 1963.

[11] D. K. Harmon (Ed.). The First Text Retrival Conference (TREC-1). NIST Spec. Pub. 500-207, National Institute of Standards and Technology, Gaithersburg MD, 1993.

[12] A. Jacoff, E. Messina, and J. Evans. A Standard Test Course for Urban Search and Rescue Robots. in Measuring the Performance of Intelligence of Systems: Proceedings of the 2000 PerMIS Workshop August 14-16 200. NIST Spec. Pub. 970, National Institute of Standards and Technology, Gaithersburg MD, 2001.

[13] William Mettrey. A Comparative Evaluation of Expert System Tools. *Computer* pp. 19-31. Feb. 1991.