

# A Hybrid Symbolic and Sub-Symbolic Intelligent System for Mobile Robots

Troy D. Kelley<sup>1</sup> and Eric Avery<sup>2</sup>  
*U. S. Army Research Laboratory, Aberdeen Proving Ground, MD 21005*

Lyle N. Long<sup>3</sup>  
*Pennsylvania State University, University Park, PA, 16802*

*and*

Eric Dimperio<sup>4</sup>  
*Indiana University, Bloomington, IN, 47405*

## **Abstract**

This paper describes our latest efforts to extend the Symbolic and Sub-symbolic Robotics Intelligence Control System (SS-RICS). Our initial development included five design guidelines for a generalized intelligence system and those guidelines are discussed in relationship to the current improvements and extensions. We also discuss recent exercises at the National Institute of Standards and Technology (NIST) robotics test facility as well as the addition of Decision Field Theory (DFT) into the architecture. We conclude that the hybrid nature of SS-RICS allows for some improvements over traditional architectures by allowing for both sub-symbolic and symbolic problems to be addressed. However, the challenges of generalization across problem spaces still remains a topic for further research.

## **I. Introduction**

**T**HIS paper will describe our latest efforts at developing a hybrid symbolic and sub-symbolic intelligent system for mobile robots. The concept of using a combination of symbolic and sub-symbolic algorithms has been discussed before [1], but we will present a new implementation that has been tested on mobile robots. At the present time this system is being applied in an indoor environment, where use of the global positioning system (GPS) is not available.

The Symbolic and Sub-symbolic Robotic Intelligence Control System (SS-RICS) [2] is intended to be an implementation of robotic control based on human cognition. Development of SS-RICS has concentrated on the integration of theories within the field of cognitive psychology - primarily theories of knowledge representation and organization. The production system within SS-RICS is based upon ACT-R [3]. ACT-R is a production-system-based cognitive architecture designed to simulate the human thought processes. ACT-R also has continuously varying algorithms which simulate human memory mechanisms and make ACT-R a hybrid symbolic and sub-symbolic architecture. SS-RICS is also a hybrid cognitive system that allows for a continuum of knowledge that includes both symbolic as well as sub-symbolic constructs. We believe that this integrated approach is the best way to represent the complete spectrum of cognition.

In order to guide the creation of SS-RICS, five developmental principles were established [4]:

- 1) The lowest level of perception includes algorithms running in a parallel fashion, while the highest levels of cognition are algorithms operating in a serial fashion.

---

<sup>1</sup> Engineering Psychologist, Human Research and Engineering Directorate.

<sup>2</sup> Computer Scientist, Human Research and Engineering Directorate.

<sup>3</sup> Distinguished Professor of Aerospace Engineering, Bioengineering, and Mathematics, and AIAA Fellow.

<sup>4</sup> Ph.D. Candidate, Psychological and Brain Sciences.

- 2) At both the low levels and the high levels of cognition, the algorithms are relatively simple. It is the interaction, processing, and results of simple algorithms which produce complex, intelligent behavior.
- 3) Pre-programming SS-RICS is guided by the algorithms that are recognized as part of the human evolutionary process (for example, algorithms for edge detection, auto-focus of eyes, pupil dilation in different lighting environments). The pre-programming that is done should allow for the emergence of complex behavior, but not be the complex behavior itself.
- 4) Cognitive development within SS-RICS is principally about the reorganization of memory elements through increasing and decreasing their respective strengths.
- 5) Cognitive development and change can occur after allowing for specialized internal processing (i.e. dreaming) or after the necessary low level elements (i.e. features) are in place to allow for higher level symbolic extraction.

We have recently implemented a dynamic model of human decision making based on Decision Field Theory (DFT) [5] into the production system to allow for more complex decisions to be considered. For example, this allows SS-RICS to incorporate information about the context of a decision in a more human-like fashion than traditional utility-based methods. This will be implemented as a conflict resolution algorithm for complex choices. We are also utilizing a Simultaneous Localization and Mapping (SLAM) algorithm [6] for path planning and localization algorithms for navigation. At the sub-symbolic level we have also begun implementing neural networks (both traditional, rate-based, artificial, neural networks and time-dependent, spiking, neural networks). These will be used to analyze low-level sensor data, such as LIDAR and image data. The neural networks quickly classify perceptual data and then transmit relevant symbolic information about the data to the production system. Perception is still a difficult issue; even in the human brain this is a complex task, roughly 25% of the human brain is used for vision. We are investigating the use of multiple types of algorithms, representations, and standard computer vision techniques for perceptual tasks. Object recognition is an especially important task. Long and Gupta [7-9] have developed spiking neural networks that mimic the human vision system. This approach is being incorporated into SS-RICS for perception.

At this time, SS-RICS consists of a production system architecture with declarative memory elements which decay at rates similar to human cognition. The system is capable of using Light Detection and Ranging (LIDAR) data to identify salient features in the environment (lines, corners and gaps) and move to them. The system is capable of renaming environmental features (a gap may be a door) and then referring to them as objects. The system has been developed using C++ object oriented programming using Microsoft .Net. The system can be applied to essentially any mobile robot, but we have primarily used Pioneer mobile robots from Mobile Robots Inc. The Pioneer DX robot is a small robot with a Pentium III computer on board with two differential drive wheels and a caster wheel (<http://www.mobilerobots.com>). Our current platform uses a wireless connection to SS-RICS running on a Dell Precision laptop with a 2.6GHz Core2 duo processor.

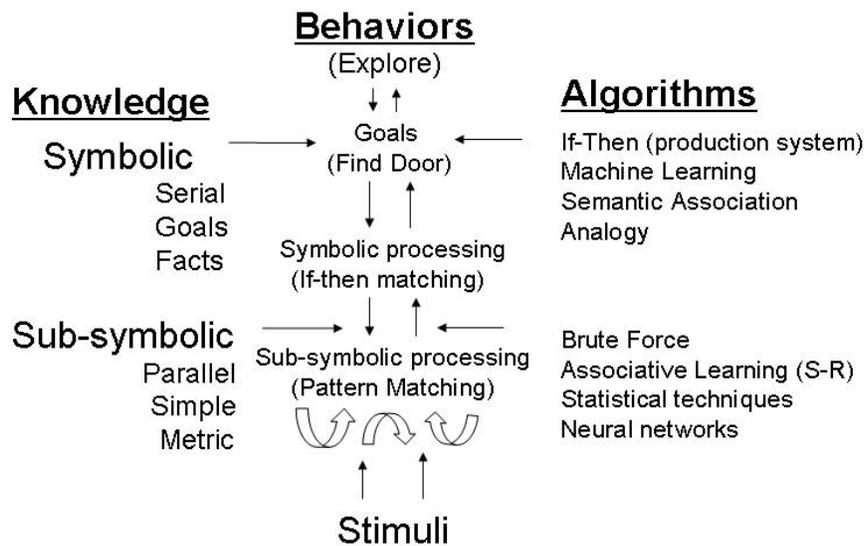
Figure 1 shows a notional version of SS-RICS emphasizing the symbolic and sub-symbolic distinction. The Figure graphically displays the different use of algorithms within different representational structures (symbolic and sub-symbolic). As Figure 1 depicts, behaviors are at the top and are a collection of goals. Some example behaviors are: explore, keep battery charged, interact with humans, and pay attention to operator. Under behaviors, a goal might be to "Find Door" in order to explore. The goals and behaviors compete for attention and attention is still a complex issue and a difficult construct to implement computationally. The algorithms listed on the right are broken into two halves. The top half represents symbolic knowledge and the bottom half represents sub-symbolic knowledge. The algorithms on the right are dependent on the ways in which the knowledge is represented and operate on the corresponding knowledge representation. The symbolic knowledge within SS-RICS tends to be serial, however, we have explored the idea of having multiple, asynchronous goals which would be a departure from the ACT-R architecture on which SS-RICS is based. Certainly having the ability to solve multiple symbolic goals at the same time would be beneficial for a robotic system. Some departures from human cognition will have to be made and carefully considered for SS-RICS to improve its behavior.

The previously mentioned DFT was integrated [10, 11] into the SS-RICS architecture. DFT was developed, due in large part, to the recognition that people do not always make optimal decisions. Humans regularly violate axiomatic principles of expected utility theory that state that the relative preference between two options should be independent of other additional options available when making choices [5]. Beyond accurately predicting context effects, DFT also makes predictions about how long decisions take and can show a speed-accuracy tradeoff under time pressure [10, 11]. We selected DFT over competitive models because of a better ability to scale up for decisions amongst many actions. Initial tests on simple decisions show that decisions can be made in about 0.1 seconds, which should allow for reasonable scalability with DFT.

DFT is a framework for applying decisions to an evidence accumulation model. It is defined as a continuous time dynamic process whereby at every point in time, a stochastic attention mechanism focuses on one of the features/outcomes of the choice situation. All choice alternatives are compared against each other along this particular feature dimension and a preference state for each alternative is incrementally adjusted. This process continues until a preference value crosses some threshold, at which point the alternative with the maximum preference is selected. Although the DFT framework has been used to explain specific context effects, temporal discounting, and speed/accuracy tradeoffs, these effects have not been demonstrated in a single implementation until the incorporation into SS-RICS.

The parameter settings for the DFT framework are especially important and one problem has been determining the initial estimates for these preferences. Some parameters are attached to learned knowledge and act as biases on options based on experience. We are hoping to use some kind of relational semantic network (like ConcepNet [12]) to seed DFT with the initial preferences and biases. Other parameters that can be manipulated determine lower-level capabilities. This is analogous to altering the ‘personality’ of the robot. A robot with a certain high threshold parameter would make slow but accurate decisions. A robot with a fast discounting rate will be focused on the present and would make fast, impulsive decisions.

DFT can naturally incorporate another concept that is observed in human decision makers as well as used in many planning algorithms to limit searches through state space: temporal discounting. The basic idea of temporal discounting is that the magnitude of payoffs is reduced as a function of the distance between the current time and the time of receiving the payoff. People often prefer receiving \$100 now rather than a promise of \$200 a year from now. Although such discounting is used regularly even in robotic planning algorithms, we encountered an unforeseen problem when testing in an actual dynamic environment. In a situation where the robot has to pick between two options with negative payoffs (the lesser of two evils), it can get stuck in an indecisive loop. Imagine a hallway with a negative choice at each end and the robot has to choose one. As it approaches the first option, the negative effects of the second choice are reduced since it is far off (in spatial and temporal distance) and the robot will switch its course of action. However, as it approaches the second option, its danger seems more eminent while the distant first option looks less intimidating. With a robot that can regularly re-plan its actions, temporal discounting will result in these vacillations. To stop the indecisive behavior, we included a bias for continuing with actions along a previously selected plan. This is supported by sunk cost effects observed in people. A sunk cost effect occurs when someone is willing to pursue a less desirable plan of action because they have already contributed resources toward that goal.



**Figure 1. Flowchart of SS-RICS System.**

We have experimented with DFT to use it for path planning and navigation where several paths must be chosen. For example, we have used DFT to select whether or not to take a long path without enemies or a shorter path with enemies. We did find the application of DFT useful for the selection of paths which contained different preferences. We are continuing to incorporate DFT into SS-RICS and may use the theory as an attention selection mechanism; however, work in this area is still continuing (this will be reported in a future paper).

## II. NIST Navigation Experiment

During a recent exercise at the National Institute of Standards and Technology (NIST), SS-RICS was required to perform three navigation tasks: the navigation of a collection of rooms, the navigation of a figure eight, and the navigation of a maze. These three environments were useful for exploring several issues in the architecture, especially learning, generalization, and mapping. To accomplish the navigation associated with all three of the environments (rooms, figure 8 and maze) we used a total of 63 rules and 15 goals within the production system of SS-RICS.

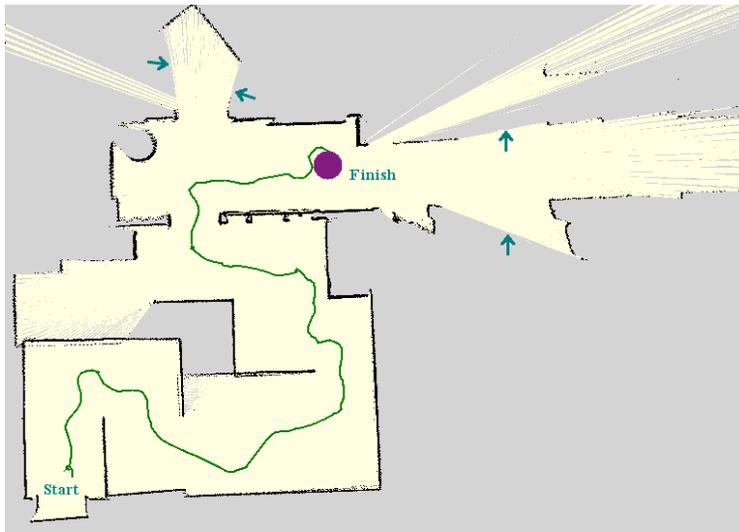
Tremaux's maze algorithm [13] was used. The algorithm works like this: As the agent proceeds down a passage, a line is drawn to mark the path (our robot didn't actually draw lines on the ground, it just remembered paths). When the agent hits a dead end, it turns around and goes back the way it came. When the agent encounters a junction that the agent hasn't visited before, a new path is picked at random. If the agent is proceeding down a new passage and encounters a junction never visited before, it is treated like a dead end and the agent will have to go back the way it came. The last step is the key which prevents the agent from going around in circles or missing passages in certain types of mazes. If the agent is going down a passage that it hasn't visited before (i.e. marked once) and it encounters a junction, it will take any new passage if one is available, otherwise it will take an old passage (i.e. one that has been marked once). All passages will either be empty, meaning the agent hasn't visited it yet, marked once, meaning the agent has gone down it exactly once, or marked twice, meaning the agent has gone down it and was forced to backtrack in the opposite direction. When the agent finally reaches the solution, paths marked exactly once will indicate a direct way back to the start. If the maze has no solution, the agent will find itself back at the start with all passages marked twice.

We implemented Tremaux's algorithm by defining a collection of goals and rules within the SS-RICS production system. The rules and goals generated declarative memories which represented each location as an intersection. The intersections contained four possible exit paths (North, South, East, and West) relative to the entering vector. Each path contained a start location and an end location bidirectionally linking the two adjacent intersections. These paths were then selected for traversal based on Tremaux's algorithm. Loop closure was accomplished by comparing the locations of new intersections to all previous intersections traversed. If the previous intersection was within 500 mm (determined by size of maze cells) of the current location it was considered the same location. This algorithm assumes that each intersection is physically marked by the agent therefore we implemented perfect memory recall when the goals were executed. Other researchers might be interested in predicting human maze behavior using the activation algorithms included in SS-RICS.

### A. Collection of Rooms

The SS-RICS architecture mapped the room environment using typical SLAM methodology; however, we occasionally lost connectivity with the robot due to the concrete walls in the NIST test course so a complete evaluation of the rooms was not conducted. We were using the Stanford Research International (SRI) Karto Software Development Kit (SDK) for mapping (<http://www.kartorobotics.com>). While the SLAM methodology certainly doesn't mimic human cognitive capabilities, we are viewing it as a necessary technology in order to allow other, more human-like capabilities, to be used. For example, with a SLAM methodology in place this can enable our robot to move around our offices and interact with people and learn from this interaction without getting lost. The metric information developed from the SLAM methodology is computationally intensive and error prone, but it is useful none the less for further cognitive development within SS-RICS.

Figure 2 shows a SLAM map for the Collection of Rooms case. In this case SS-RICS was not used; instead we utilized Karto's Exploration and Path Planning modules to freely explore the rooms. The exploration module utilizes "frontier points" to find places to explore. Frontiers are the boundaries between explored and unexplored space and frontier points are points on this boundary that may be successfully reached by the robot. Some examples of these are marked with arrows in Figure 2.



**Figure 2. SLAM map for the collection of rooms case with arrows marking possible frontier points**

## B. Figure Eight Maze

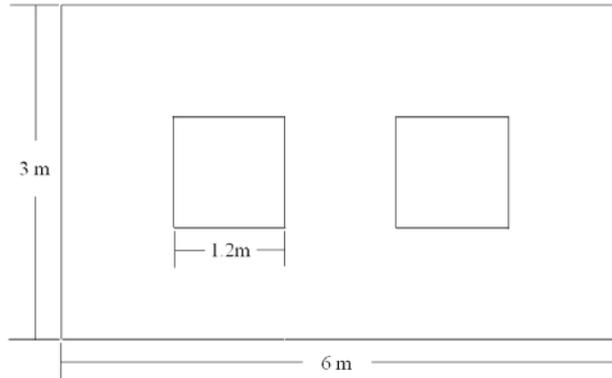
During the NIST exercise, the SS-RICS architecture was able to successfully navigate the figure eight environment (Figure 3) without using a predefined map. The goal here was to map the figure eight space completely and the robot did this successfully by completely mapping the space, even though it did not traverse the space in the middle. There was no time limit for this exercise. These intersections were then linked together into a memory for the robot to follow. For example, an intersection could be defined as a left turn, a right turn, a straight (a non-intersection), as well as combinations of each (straight AND right), which were called “perceptual primitives”. These intersections were represented as rules by using distances to the walls. For example, the robot using the laser checked the distances for three points: straight, right, and left. If the distance to a wall was less a certain range, the robot concluded that this path was blocked. If the distance was greater than a certain range, the robot concluded that this path was open. Using these three distance values SS-RICS would match to seven possible primitives to identify the intersection. We viewed this behavior as largely symbolic, rule-based behavior.



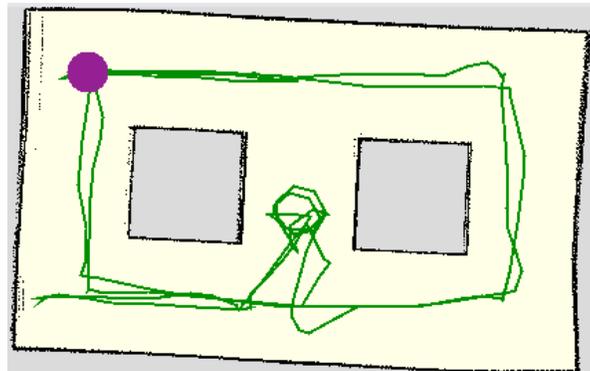
**Figure 3. Photograph of NIST Figure Eight**

Figure 4 shows a schematic of the figure eight maze. The enclosure was 3m x 6m and the inner boxes were 1.2m x 1.2m. Figure 5 shows the route the robot took in learning the maze. The robot started and ended at the dark circle. In Figure 6, the points in the maze are marked where the robot discovered new intersection types. It was able to completely map out and learn the figure eight maze autonomously. The robot was able to perform several instances of loop closure. The robot performed one full loop of the outer figure of the figure eight (shown as intersections 1-6 on Figure 6). When the robot returned to intersection 1 it began to backtrack from 1 to 6, from 6 to 5, and then correctly navigated to intersection 7, but misclassified it as a dead end. At this point the robot was lost (see loops on figure 5) but was able to recover from being lost. Eventually the robot created intersection 8, then intersection 9 as a dead end, then moved back to intersection 5, where it found a loop closure. Once the robot found intersection 5 it

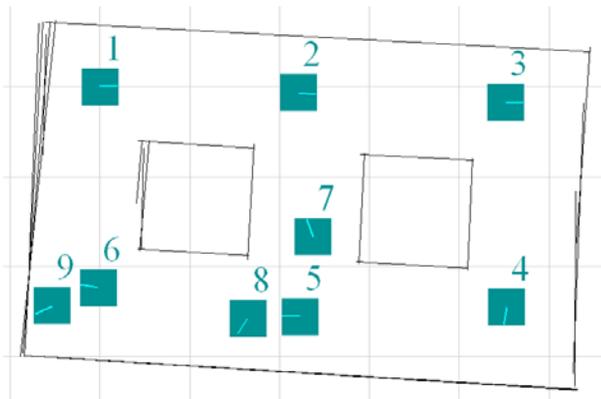
was no longer lost. At that point SS-RICS backtracked from intersection 5 to the starting point, intersection 1, and correctly stopped exploring the maze.



**Figure 4. Schematic of Figure Eight Course**



**Figure 5. Robot Route in Figure Eight Course**



**Figure 6. Robot Intersection Decision Points in Figure Eight Course**

### C. Large Maze

The behavior of the robot in the larger complex maze (Figure 7) was not perfect. One problem of using the previously described perceptual primitives is the possibility of false positives. The robot would occasionally incorrectly identify an intersection in the maze and this problem would become compounded when the robot returned to the incorrectly identified maze location. The robot would move a certain distance (1.2 meters) in order to identify each intersection and occasionally this distance was not completely accurate. Additionally, we attempted to keep the robot centered in the maze to make each intersection identification easier, but this algorithm would not always leave the robot with the correct orientation, causing a miss classification. This is a problem with robotic perception, something we have called the “perceptual problem”.



**Figure 7. Photograph of NIST Large Maze.**

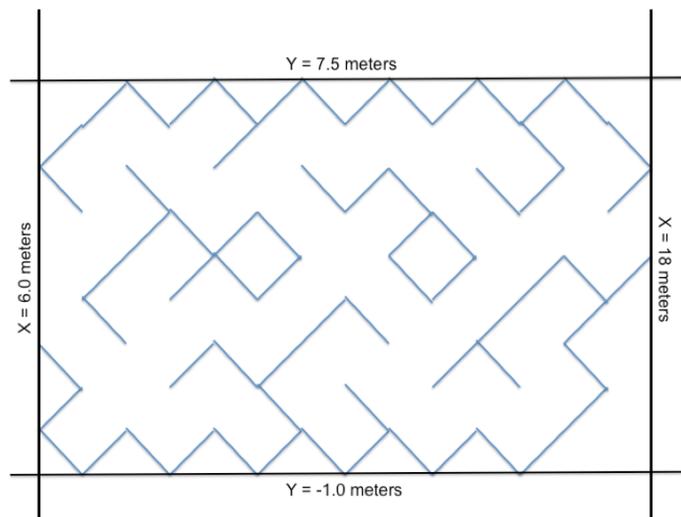
Multiple ways of perceiving each intersection would be one possible solution to this problem and the SS-RICS architecture lends itself well to multiple perceptual mechanisms (both symbolic and sub-symbolic). For instance, we experimented with using neural networks in addition to the symbolic rule based perception, but this combination approach was not fully implemented due to time constraints. This method would also require a mechanism to select amongst the possible outputs from multiple perceptual algorithms.

Another problem with the primitives system was that the method needed certain threshold values for a correct match to an intersection type. If a wall of the maze was less than or beyond an expected value, the method would not match to the correct intersection primitive. A possible solution to this problem would be that SS-RICS should learn the threshold values associated with the intersections ahead of time. This would allow SS-RICS to adapt to different, but similar, scenarios. We are finding this is a common problem - using threshold values for the definition of rules. We have found that the values end up being too specific and not flexible enough for the application of a more general rule. Learning these values would be one solution.

We were also interested in using the NIST exercise as an exploration in generalization - since we want SS-RICS to be capable of performing a wide variety of tasks in a wide variety of situations. Once we had developed the code necessary to accomplish the maze, we were interested in applying those behaviors from the maze to other tasks. We identified a few behaviors which could generalize, but not many. For example, we assumed that learning a maze would generalize to the overall task of navigation. A few things that could generalize from a maze to navigation would be: the concept of being lost, identification of landmarks, ordering of landmarks, counting landmarks, the concept of having a goal or endpoint, the concept of a grid, and the concept of a cell within a grid. These are some behaviors that can generalize from a maze to overall navigation and we intend to use the same rules and goals we developed in the production system architecture of SS-RICS for other navigation tasks. However, it could be that very little generalizes across tasks which are not similar to each other. For instance, how does learning a maze generalize to interacting with people? Lack of generalization across task domains has been a continuing problem for the field of AI. Additionally, there is also considerable evidence that generalization is difficult for humans even within similar task domains [14]. Generalization will be a continuing issue for the development of SS-RICS.

Figure 8 shows a schematic of the large maze; the enclosure was 7.5m x 18m. In the large maze we were able to run the SS-RICS architecture through the maze and find an exit point; however, this was not done completely autonomously. The robot required interactions with the operator to stay on course and had difficulty recognizing each intersection in the maze (Figure 9). To overcome this problem, after moving to a new intersection, if the orientation of the robot was not what was expected, the robot asked the operator if the current location and orientation was suitable to make an intersection determination. If the operator thought the robot was in an inappropriate location, the operator would move the robot into a location which would be suitable for an accurate perceptual reading from the sensors. Over one run of the maze, which included backtracking over locations, the

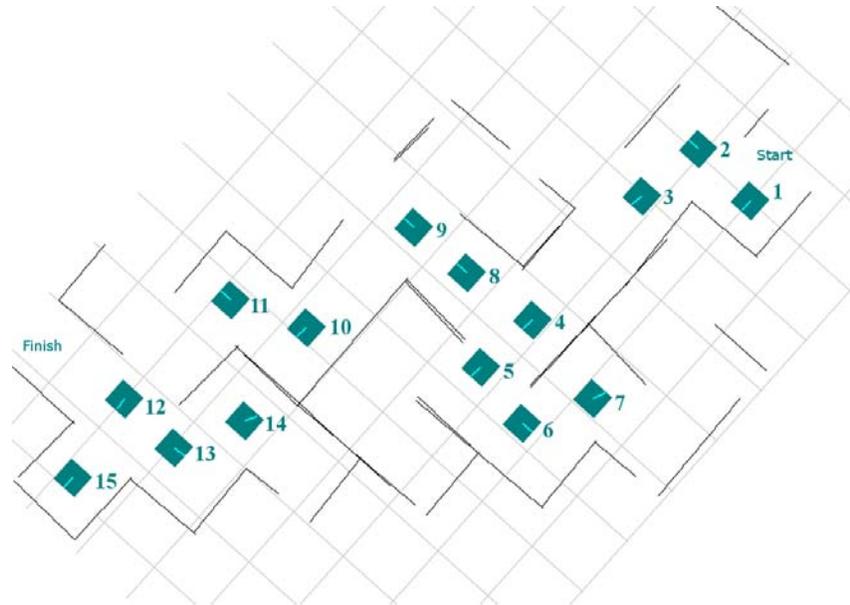
robot covered a total of 30 locations; which included intersections and straight passageways. Of those 30 locations, 11 (36 percent) required interactions with the human operators. In Figure 10, the locations in the maze are marked where the robot discovered new intersection types. The robot and operator were able to correctly move to and classify locations one through six; however, at location seven the operator did not reorient the robot correctly causing it to classify the location as a dead end. The misclassification at location seven caused a back track to location four. At location four the robot continued on and successfully created locations eight through thirteen. Again at location fourteen the robot was not correctly oriented and misclassified the location as a dead end, causing another back track to location twelve. Lastly, location fifteen was created and the robot backtracked to location twelve and finally exited the maze.



**Figure 8. Schematic of Maze**



**Figure 9. Robot Path in Maze**



**Figure 10. Intersection Decision Points in Maze**

### III. Conclusions

The initial five guiding principles (discussed in the introduction) continue to help steer the development of SS-RIC; however, some of the guiding principles have been easier to follow than others. The first two guiding principles, using simple parallel algorithms at the low levels, and simple serial algorithms at the high levels, has continued to be a real benefit as the robot has become more complex. Also, as previously mentioned we used a small number of goals to solve the NIST figure eight maze and we hope this will generalize to other navigation tasks. Principle number 3, using algorithms defined by evolution, has been difficult to follow because there are so many possible candidate algorithms. Determining which algorithms to concentrate development on has been a challenge. Principle 4, that cognition is guided by the reorganization and strengthening of memory elements, is an important principle, but we have yet to see it fully realized. We have had trouble limiting the scope of memories which get strengthened through learning. Ideally, this should be limited to the production system, or working memory. Additionally, we have found that the activation functions which were taken from the ACT-R cognitive architecture probably need to be modified for low level perceptual memories. We have found that perceptions don't need to be activated as strongly as higher level symbolic memories. This is still an area of research. Finally, principle 5, that hierarchical development can be generated once the lower levels of the hierarchy are in place, requires more work in developing the lower levels of the system. We are aware that we do not have all the necessary low level perceptual mechanisms in place in order to generate higher level symbolic relationships. And without the correct perceptions some types of symbolic knowledge can never be developed. We continue to research ways in which we can help SS-RICS understand and identify the human salient features of the world; however, this continues to be a challenge.

The generalized hybrid architecture of SS-RICS offers benefits over traditional symbolic or sub-symbolic architectures. We have found benefits in using simple parallel algorithms at the low levels in terms of speed. We can selectively turn off sub-symbolic processing to increase available resources. Additionally, we were encouraged at the symbolic level, where we utilized relatively few goals (15), which were able to be applied to all 3 of the NIST environments. We could also use the same 15 goals in other navigation environments. The complex maze did require some interaction with the operator, and until we improve the perceptual capabilities of SS-RICS, helping our robot understand perceptual problems will continue to be a priority. Humans will continue to have better perception than most robots for the foreseeable future so it will be important to understand how human intervention can help robots with perceptual tasks. Additionally, cognition requires a multipart approach [15, 16] and perception is one of many complex capabilities a robot must possess. Moreover, many parts of the system such as perception, memory, action, planning and decision making must be in place before complex behavior can be produced. We will continue to develop SS-RICS in the hope of creating a generalized intelligence system.

## Acknowledgments

Lyle N. Long gratefully acknowledges the support of the Army Research Laboratory (Contract No. TCN 07-305). We would also like to thank Adam Jacoff of NIST for his help during the NIST exercises.

## References

1. Boden, M. A. "New Breakthroughs or Dead-Ends?," *Transactions of the Royal Society* Vol. 349, No. 1689, 1994, pp. 1-13.
2. Avery, E., Kelley, T. D., and Davani, D. "Using Cognitive Architectures to Improve Robot Control: Integrating Production Systems, Semantic Networks, and Sub-Symbolic Processing," *15th Annual Conference on Behavioral Representation in Modeling and Simulation (BRIMS)*. Baltimore, MD, 2006.
3. Anderson, J. R., Bothell, D., Byrne, M. D., Douglas, S., Lebiere, C., and Qin, Y. "An Integrated Theory of the Mind," *Psychological Review* Vol. 111, No. 4, 2004, pp. 1036-1060.
4. Kelley, T. D. "Developing a Psychological Inspired Cognitive Architecture for Robotic Control: The Symbolic and Sub-symbolic Robotic Intelligence Control System (SS-RICS)," *International Journal of Advanced Robotic Systems* Vol. 3, No. 3, 2006.
5. Roe, R. M., Busemeyer, J. R., and Townsend, J. T. "Multi-alternative decision field theory: A dynamic connectionist model of decision-making," *Psychological Review* Vol. 108, 2001, pp. 370-392.
6. Smith, R. C., and Cheeseman, P. "On the Representation and Estimation of Spatial Uncertainty," *The International Journal of Robotics Research* Vol. 5, No. 4, 1986, pp. 56-68.
7. Long, L. N. "Scalable Biologically Inspired Neural Networks with Spike Time Based Learning," *IEEE Symposium on Learning and Adaptive Behavior in Robotic Systems*. Edinburgh, Scotland, 2008.
8. Long, L. N., and Gupta, A. "Scalable massively parallel artificial neural networks," *Journal of Aerospace Computing, Information, and Communication* Vol. 5, No. 1, 2008.
9. Long, L. N., and Gupta, A. "Biologically-Inspired Spiking Neural Networks with Hebbian Learning for Vision Processing," *46th AIAA Aerospace Sciences Meeting*. AIAA, Reno, NV, 2008.
10. Busemeyer, J. R., and Johnson, J. G., eds. *Micro-Process Models of Decision-Making*. Cambridge: Cambridge Univ. Press, 2006.
11. Busemeyer, J. R., and Townsend, J. T. "Decision Field Theory: A dynamic cognitive approach to decision making.," *Psychological Review* Vol. 100, 1993, pp. 432-459.
12. Liu, H., and Singh, P. "ConceptNet - A Practical Commonsense Reasoning Toolkit," *BT Technology Journal* Vol. 22, No. 4, 2004, pp. 211-226.
13. "<http://www.astrolog.org/labyrinth/algorithm.htm>." 2009.
14. Holyoak, K. J., and Koh, K. "Surface and Structural Similarity in Analogical Transfer," *Memory and Cognition* Vol. 15, No. 4, 1987, pp. 332-340.
15. Kelley, T. D. "Simulating Intelligent Behavior Requires a Complex Approach," *AAAI Spring Symposium on Between a Rock and Hard Place: Cognitive Science Meet AI-Hard Problems*. 2006.
16. Long, L. N., and Kelley, T. D. "The Requirements and Possibilities of Creating Conscious Systems," *InfoTech@Aerospace* AIAA Paper No. 2009-1949, Seattle, 2009.