Modeling Metacognition for Learning in Artificial Systems

Darsana P. Josyula, Harish Vadali, Bette J. Donahue, Franklin C. Hughes
Department of Computer Science
Bowie State University, Bowie, MD 20715
darsana@cs.umd.edu

***Abstract*—Evidence suggests that metacognition—the ability to monitor the cognitive processes and regulate them—exists not only in humans but also in some animals. In nature, humans and animals use metacognition to self-regulate their learning process. This paper gathers evidence of metacognition in nature from research in various disciplines. It also shows how metacognition can be modeled in artificial systems and how the model is applied in an Air Traffic Control Simulator system.**

Keywords: metacognition, cognitive modeling, intelligent agents, fault diagnosis, learning

I. Introduction

Metacognition is not as mysterious as it sounds. In fact, we see evidence of it in everyday thought processes such as deciding to make a grocery list so that you can remember the items easier, planning to type your lecture notes to help study better or knowing that the answer is on the "tip of your tongue". All are examples of how people think about their cognitive processes, develop strategies to improve their cognitive skills and generally evaluate the information contained in their memory.

The term metacognition has slightly different definitions depending on the author and the discipline. J. H. Flavell [1] defines it as one's knowledge concerning one's own cognitive processes or anything related to them. Ridley [2] gives a more precise definition of metacognitive skills as " taking conscious control of learning, planning and selecting strategies, monitoring the progress of learning, correcting errors, analyzing the effectiveness of learning strategies, and changing learning behaviors and strategies when necessary." Independent of the definition used, the actions involved in improving the cognitive processes —monitoring, assessing and guiding—remain uniform.

The study of metacognition has provided educators and psychologists with insights into the processes involved in learning, the benefits of self-regulated learning and the skills that distinguish expert learners from their less successful peers. Recent work [3], [4], [5] on human learning has suggested that the best learners are the ones who practice self-regulated learning. Simply put, metacognition empowers learners. Self-awareness allows the learner to decide *what to learn, when to learn* and *how to learn*. Metacognition provides a means to accurately assess one's current knowledge and skill levels, identify when new knowledge is needed as well as provide strategies to acquire new knowledge.

II. Evidence of Metacognitive Learning

Evidence suggests that both humans and animals use metacognition to help the learning process. Research has shown how humans use many metacognitive techniques to learn a new second language or train on a new task. Research has also shown that animals use metacognition in evaluating their knowledge level in prospective and retrospective confidence judgment experiments. Experimental results show that animals choose to opt out of tests that they believe that they cannot pass. This section presents some of the evidence on the existence of metacognitive monitoring and control in humans and animals.

A. Human Studies

1) Academic Performance: Isaacson and Fujita's study [5] used eighty-four undergraduate students enrolled in an introductory educational psychology class to investigate the role of metacognitive knowledge monitoring in self-regulated learning and academic performance. Metacognitive knowledge monitoring (MKM) is the ability of learners to recognize whether or not they have mastered an academic task. Self-regulated learning requires students to be active, goal-directed learners with self-control over their behavior, motivation, and cognition.

In this study, the students were given ten weekly tests that were designed to reveal and substantiate student metacognitive awareness during testing. Each test included 40 true-false and multiple choice questions of varying difficulties: (i) 18 questions emphasized knowledge and comprehension and were worth 1 point each, (ii) 18 were moderately difficult questions that emphasized application of knowledge and were worth 2 points each and (iii) 4 were difficult test questions that emphasized analysis and synthesis and were worth 3 points each.

On each test, students were only allowed to answer 30 of the 40 test questions. Their test grades were dependent on both the accuracy of their answers and the type of questions they chose and answered correctly. The program was constructed such that only those students who chose more difficult test questions (worth 2 or 3 points) and got them correct could obtain an A in the course. If students attempted the harder questions but got a higher percentage wrong or attempted only the easier (worth 1 or 2 points) questions and did get a higher percentage correct, they would still receive a lower overall grade than an

A. According to Isaacson and Fujita, the key to success in the course was not only correctly answering test questions, but also choosing the test questions you could answer correctly.

To help identify the students' ability to evaluate their own learning expectations and MKM, they were also required to complete a questionnaire partially before the exam and the remainder after they finished the exam, but before the test was graded. The questions they answered prior to the test included (i) the number of hours they studied, (ii) how many points they needed to score to be satisfied with their performance (*satisfaction goal*), (iii) how many points they needed to score to be proud of their performance (*pride goal*), (iv) and how confident they were about achieving their satisfaction goal also known as their pre-test self-efficacy. After completing the test, but before it was graded, each student was also asked to identify (i) how many points they believed they had scored on the test and (ii) how confident they were now about achieving their satisfaction goal also known as their post-test self-efficacy. Then, tests were graded and returned to the students for review before class ended each time.

According to Isaacson and Fujita, high achieving students were (i) more accurate at predicting their test results, (ii) more realistic in their goals, (iii) more likely to adjust their confidence in-line with their test results; and (iv) more effective in choosing test questions to which they knew the answers.

2) Second Language Education: O'Malley's [6] study involves analyzing the use of metacognitive, cognitive, and socio-affective learning strategies used in English as a Second Language (ESL) classes. 70 Hispanic subjects in beginning- and intermediate-level ESL classes were observed and interviewed about the specific learning techniques they used for their classes. A total of 638 instances of learning strategies were described by the subjects during their small group discussions while only an average of 3.7 strategies were identified during the 53 observation periods conducted. This difference in numbers was explained by O'Malley as most likely the result of some learning strategies lacking an observable behavior component.

The results of the interviews were grouped into 20 distinct strategy types which included 7 metacognitive, 11 cognitive, and 2 socio-affective methods. Based on strategy usage, 30% of total usage was metacognitive, 53% was cognitive and 17% was socio-affective strategies. Among the metacognitive strategies, a difference was seen between beginning- and intermediate-level students. For example, *self-management techniques* were used by the beginning- and intermediate level subjects 19.6% and 22.5% of the time, respectively. *Advanced preparation* was used 21.4% of the time by beginners versus 25.0% of the time by intermediate subjects. *Selective attention* was used 22.3% of the time by beginners and 16.3% of the time by intermediate level subjects.

The study also observed that the direct combination of cognitive strategies with metacognitive strategies were rarely (7%) reported, even though overall strategy combinations were reported as 21% of all strategies.

3) Task Performance: In 2005, van Gog's group conducted a study [7] on the ability of concurrent, retrospective and cued reporting to elicit information about the problem-solving process carried out for performing a task. Verbalizations about the actions taken, why or how something happened or metacognitive reflections used in solving the task were analyzed using each reporting method. The study involved 26 participants who completed computer-simulated troubleshooting tasks on malfunctioning electrical circuits using Crocodile Physics 1.5 software program set at the fourth-year high school or pre-university level.

In concurrent reporting, subjects were asked to think aloud or verbalize their thoughts while performing the task. Retrospective reporting required the subjects to wait until immediately after completing a task to report their thoughts and cued retrospective reporting used the original computer-based task with a superimposed record of the subject's eye fixations and mousekeyboard operations as a cue for retrospection.

The study showed that concurrent reporting resulted in more action information as well as information on why or how something happened than retrospective reporting. Metacognitive reflection information remained roughly the same using all three methods.

4) Online or E-Learning: E-Learning is typified as learner-centered environments created using network technologies to provide anywhere-anytime access for its users. Some e-learning software allow users to create their own learning strategies using existing metacognitive skills; others improve the metacognitive skills of the learners to help learn the material.

Kafai's [8] Game Design Project allows users to create their own learning strategies using existing metacognitive skills. In this project, third graders created their own fraction video game. The users ability to control their environment resulted in heightened motivation and longer periods of engagement in constructive, educational activities that teach math skills at a concrete level as well as the scientific process of discovery at an abstract level.

An example of an E-learning program that helps improve the metacognitive skills of the learners is Malcolm's KineticCity.com [9]. Set in an action-adventure story-based program, KineticCity.com combines hands-on science applications and active computer-guided learning for the fourth- and fifth-grade age level. The user is designated as the hero of the story and works through an outlined science curriculum *without realizing it* (a key component in any stealth education project which combines entertainment and educational components).

B. Animal Studies

Studying metacognition in nonhumans is difficult since such studies have to rely on non-verbal forms of communication. Two studies are presented below:

1) Rats: In Foote's [10] work with rats the subject is presented with a stimulus and then allowed to decline taking a test presumably based on its awareness of whether or not it knows the correct answer. If the subject takes the test and gives the correct response it will receive a large reward. Declining the test will result in a minimal reward and a wrong answer receives no reward.

This study utilized a duration-discrimination test to train the subjects on distinguishing between sounds of short and long durations. Intermediate sounds were later added to the test and the rats were asked to classify these sounds as either long or short. During some of the trials the rats were given an option to decline taking the test after the sound was played.

The results from the experiment suggest that rats know when they do not know the answer to a duration-discrimination test. Since duration discrimination involved classifying sounds as short or long, sounds with durations near the middle of the range are more difficult to classify. The rats were more likely to decline these difficult tests. This suggests that the rats knew that they did not know the correct duration-discrimination response. Moreover, the rats were more accurate in their responses when they had chosen to take the duration test compared to trials in which they were forced to take the test.

2) Monkeys: Several studies involving metacognition in monkeys (rhesus macaques) have been done over the last several years. One such study [11] involved evaluating the ability of monkeys to transfer their ability to make retrospective confidence judgments on their performance on a perceptual task to a new perceptual task and to a working memory task.

During the training phase, a monkey is required to identify the stimulus with the longest line or the greatest number of geographical objects. After each trial, the subject is asked to select either a high or low confidence icon which affects the size of reward it receives with a correct answer. Feedback is given using a token economy, in which the monkey could earn or lose tokens from a hopper on the screen. A high confidence selection along with a correct answer will result in the highest reward; a low confidence selection with a correct answer will result in a minimal reward while an incorrect answer reduces the total tokens earned so far. Tokens are stored in a closed container and only dispensed when a total of eight tokens are acquired, this allows tokens to be removed for wrong answers before a reward is issued. The tokens are exchanged for rewards upon dispersal.

To test whether the monkeys could transfer the metacognitive ability obtained in the training phase to a new perceptual task, the monkeys were trained on new tasks like selecting the largest circle or the smallest circle. The metacognitive paradigm using the token system was then introduced to test the ability of the monkeys to make retrospective confidence judgements on their performance on the new tasks.

To test whether the monkeys could trasfer their metacognitive ability to a serial working memory task, the subjects were trained on a working memory task. The monkeys were shown six sequential sample pictures and then presented with a page containing six pictures that includes one of the sample pictures. The task was to select the picture that was seen before, from among the pool of distracters and then to make a confidence judgment on the knowledge.

> "The positive results from this study confirm that monkeys are able to transfer the ability to make metacognitive judgments from perceptual tasks to serial working memory tasks, and thus make confidence judgments about their own memories, not just psychophysical discriminations."

[12] presents a study that includes two separate phases to test the retrospective and prospective judgments of confidence of a monkey. The retrospective experiment involved the monkey performing a recall task followed by a confidence assessment. The prospective experiment required that the subject make its confidence assessment before taking the recall test based solely on how much it learned from previous experiments and the study phase of this experiment. An analysis of the results indicates that the monkey can transfer the ability to make metacognitive judgments from the serial working memory tasks in previous experiments to retrospective and prospective recall tasks.

III. MCL

The underlying conceptual apparatus of our metacognitive model MCL [13], [14], [15] is to notice anomalies, assess their importance and cause, and guide a response into place. The general MCL architecture has three sets of ontologies corresponding to the Note-Assess-Guide loop: an *indications* ontology for anomaly types to note, a *failure* ontology for use in assessment, and a *response* ontology for selecting repair types to guide, as in Figure 1.

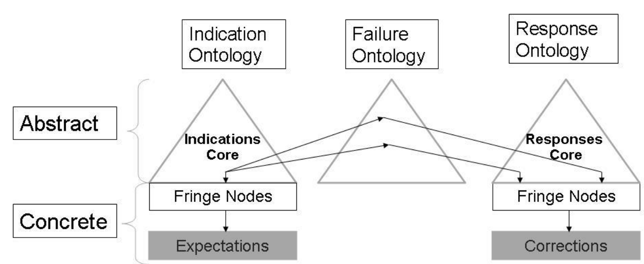

Fig. 1. An overview of the MCL ontologies

The core nodes of each ontology is implemented as Bayesian networks. These core nodes represent abstract and domain-general concepts concerning anomalies and how to respond to them. These nodes are linked within each ontology to express relationships between the concepts they represent. They are also linked between ontologies, allowing MCL to employ a number of Bayesian algorithms for reasoning over ontologies.

At the bottom of the indication and response ontologies are the "fringe" nodes. The fringe nodes below the indications core represent concrete, specific information about the anomaly and those below the responses core represent specific correction information.

MCL is linked to the host through two interfaces as shown in Figure 1. At the input interface, expectations are directly linked to the indications ontology through indication fringe nodes. At the output interface, the response ontology through its fringe nodes is linked to a set of possible corrections that the host could employ. When an actual perturbation occurs in the host, MCL will detect the expectation violation through the input *fringe* nodes. It will then attempt to map it into the MCL core so that it may reason about it abstractly. MCL's reasoning process then produces an output which is articulated through the output fringe nodes in the form of an action that the host is able to carry out.

IV. Air Traffic Simulator

In this section, we discuss the power of metacognition in an Air Traffic Control (ATC) Simulator system. The ATC controls the air traffic within a radar range and ensures that planes land safely by assigning an approach landing vector. The ATC acts as a server that communicates with planes over TCP/IP using implemented communication protocols. This simulation represents a 10000 by 10000 area. The ATC is situated at the center of the region and planes must move to the center to complete the simulation. The Metacognitive Loop (MCL) monitors both the planes and the ATC to find anomalies and expectation violations to guide the ATC to act effectively.

The planes can spawn at any point in the environment outside the ATC's radar range which is a 5000 by 5000 square at the center of the geographical area. Initially, when the plane connects to the ATC server, it gets a message from the ATC containing its plane ID. All the planes continually report their current location to the ATC and accordingly the ATC checks whether it is under its radar region or not. The plane is required to circle when it first hits the radar region until it gets a message from the ATC containing its assigned approach path. The ATC uses one of four strategies, as selected by the MCL, to assign an approach path. The MCL monitors overall environmental variables like number of planes circling, the number of free approach paths, and the performance of known strategies to guide the ATC to choose the appropriate strategy. The goal of this simulation is to effectively choose the strategies thereby minimizing the time spent circling and avoiding collisions between the planes.

A. Approach Landing Strategies

The ATC is equipped with the following strategies in order to choose the approach landing path for the planes.

- *Nearest Terminal Strategy*: Using this strategy, the plane is assigned an approach path by calculating the distance from the current location of the plane to all approach paths and choosing the closest one. If another plane is already assigned to this approach path, the new plane must circle and wait for the approach to be free. An approach vector becomes free once a plane has reached the ATC location $(5000, 5000)$ at which time the ATC sends a message to the next plane waiting to enter the approach vector. This process is continued until all planes reach their final destination.
- *Free Terminal Strategy*: This strategy uses the principle of the Nearest Terminal strategy with some modifications to accommodate the planes that would be left circling unnecessarily if another vector was free. In this case, if the nearest approach path is busy, it allows the system to determine if other approach paths are free in an expanding search pattern. If it finds that any path is free, that approach path is assigned to the plane that would be required to circle under the Nearest Terminal strategy. The flags on the paths are cleared once the planes reach the destination and other planes are then free to use the paths.
- *Queued Terminal Strategy*: Under this strategy, the ATC is able to assign up to 5 planes at a time to each approach vector based on their current distance to an approach path start point. When the first plane in the queue reaches the approach path starting point, then another plane is put into the queue. Allowing more planes to take a particular path may cause collisions at the terminal. So, this strategy uses a separate collision avoidance system to ensure planes land safely.

B. Collision Avoidance System

Collision avoidance is required in the Queued Terminal landing strategy which controls the actions of one to five planes at a time in a fairly small geographical area. A safe distance between aircraft must be maintained at all times which requires the ATC to reduce each vehicle's speed as necessary. For example, if planes A and B are assigned to an approach path, they are added to the queue in the order of increasing distance to the path start point. The speed of the planes may be varied according to the distance and the time needed to reach the approach path start point. If plane A is at the start point and plane B is close to A, the time to the assigned approach path is calculated for each of the planes and the difference is compared with the safest allowed time difference. If the difference is greater than the minimum safe distance, the speeds remains unchanged. If it is less than the safe distance, a new speed is calculated and assigned to the plane which is farther from the approach vector. This operation is performed on all the planes in the queue to ensure they land safely.

C. Learning Strategies

The ATC has the ability to create new landing strategies during the simulation. These strategies are used to assign the approach path to the planes by running a data mining algorithm on training data. In order to create the training data, a virtual ATC with virtual planes at random points are spawned off; the designated approach path and speed using the Queued Terminal strategy are used as the desired output values for the training data. Once the training set is ready, the data mining algorithm uses it and discovers a strategy that works for the training data. This new strategy can be used by the system on the actual input data that is being run over the simulation to provide approach vectors. Thus the ATC can dynamically learn new strategies and use them on the actual data set in a dynamic environment.

V. MCL IN ATC

The MCL component plays an important role in the domain by identifying any expectation violations that might occur while the simulation is running. After a problem is identified, MCL will assess the error and its causes and then guide the ATC to make changes in its landing strategies. Possible responses to expectation violations include (i) switching strategies, (ii) refining current strategies that have learning capabilities or (iii) creating new strategies better suited to fit the situations. The expectations for the domain are stored in the knowledge base of the MCL component and include (i) circling time of the plane should not exceed 80 seconds, (ii) planes are expected to reach assigned goal locations within a defined time, (iii) the collision avoidance system should not be engaged more than 10 times per simulation and (iv) the plane must maintain the speed assigned to it by the ATC.

The default landing strategy for the ATC is the Nearest Terminal strategy which is utilized until the MCL notes an expectation violation and guides the ATC to the most effective strategy for the current air traffic situation. Once an expectation violation is noted, the MCL will evaluate the current situation variables in the domain to determine which, if any, of the strategies best suit the current situation. These variables include (i) if any other approach vectors are free and (ii) the density of the air traffic in any given approach vector region. The Queued Terminal strategy will be recommended if a circling time expectation violation is noted and either all of the terminals are currently active or the air traffic density for any approach vector exceeds the maximum planes allowed in circle mode.

A. Expectation Violation Scenarios

The following scenarios represent expectation violations identified by the MCL and the responses the metacognitive component sends to the ATC.

1) Case 1: In Figure 2, 7 planes (OXIVZ, SYUTN, P6YE2, MEGTE, BWPEE, OF0J2 and 302FL) are spawned at random points outside the radar region to simulate aircraft competing for the approach vectors. The planes BWPEE and MEGTE each reach the radar region before the other planes and are assigned to their nearest approach vectors as illustrated in Figure 2. The other planes have to circle at the edge of the radar region until they receive an approach vector message from the ATC.

In this case, the MCL will note that the maximum circling time is exceeded by one or more of the five planes waiting for the same approach vector, which causes an expectation violation. The MCL will then evaluate the current situation variables to determine if any other approach vectors are free or if the air traffic density is exceeded for any approach. Since the current situation holds some free approach vectors and density of the air traffic is low, the MCL's response will be to instruct the ATC to switch to the Free Terminal strategy as illustrated in Figure 2. Under the Free Terminal strategy, the ATC will send out new approach vector assignments to the waiting planes that will then travel to their new approach paths.

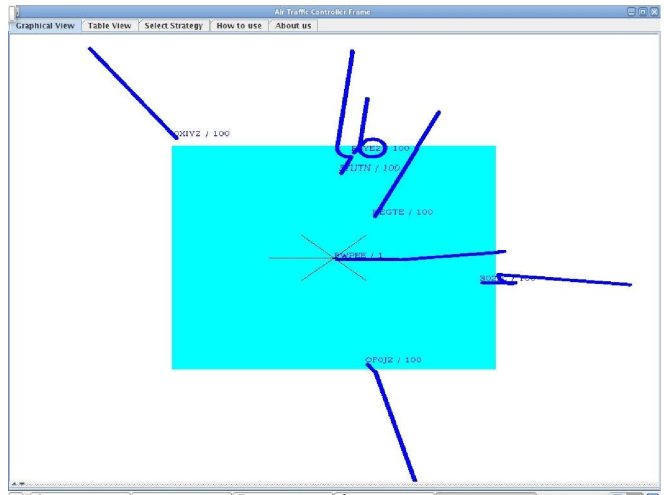

Fig. 2. MCL triggers strategy switch to Free Terminal Strategy.

2) Case 2: In Figure 3, 6 planes (VMU1F, IP5G2, 8CY0L, V4MNF, 1OKBO and ZRT5V) are spawned in the same region to simulate higher density traffic competing for a single approach vector. The plane VMU1F reaches the radar region first and is assigned to the nearest approach vector as shown in Figure 3. All the other planes start circling when they hit the radar region and wait for approach vector assignment from the ATC.

As in the previous case, MCL notes the circling time violation and then evaluates the current situation variables. There are other approach vectors free but the air traffic density limit is exceeded for this approach vector which triggers the MCL recommendation to use the Queued Terminal strategy.

The ATC creates a new queue and assigns the nearest five planes to the queue allowing any additional planes to circle until the queue is available. In this example, a maximum of five planes (VMU1F, IP5G2, 8CY0L, V4MNF and ZRT5V) are queued according to their closeness to the approach vector. The remaining plane (1OKVO) continues to circle until the first plane reaches the center and is removed from the queue. As the number of planes increase inside the radar region, the danger of an air collision increases. Under the Queued Terminal strategy, the collision detection system is enabled which ensures the aircraft maintain a safe distance from each other. If the danger of a possible collision is detected, the collision detection system calculates a new speed for each plane and the ATC orders each to modify its speed according to the detection systems requirements.

3) Case 3: Certain situations like sabotage or a communication error can result in an aircraft changing its course.

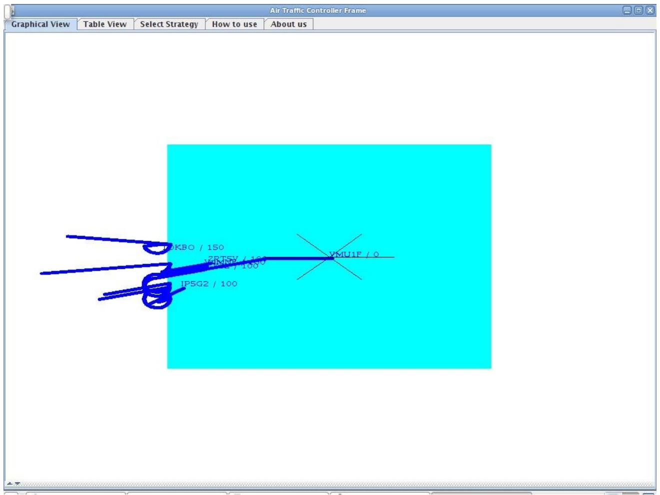

Fig. 3. MCL triggers strategy switch to Queued Terminal Strategy.

The illustration in Figure 4 shows how the MCL notes an exceeded time to goal violation for plane UVY9R on the right which alters its flight path away from the nearest approach path to which it was assigned [1]. Upon detection, the MCL recommends that the ATC send a goal update message to the plane and then continues monitoring the environment. In this case the goal update message corrected the problem and the plane changed its flight path to reach the assigned approach vector.

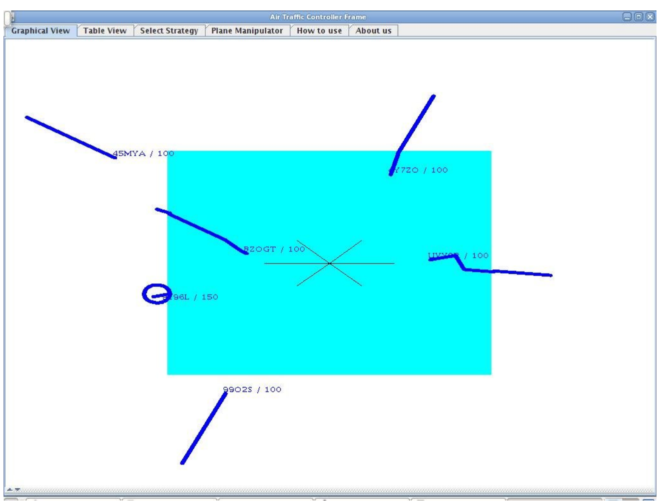

Fig. 4. MCL identified the goal violation for plane UVY9R and resent the actual goal message.

VI. Conclusion

Metacognition occurs in nature in various forms and it helps humans and animals deal with anomalies and regulate their learning. Modeling metacognition in artificial systems can improve their performance in the presence of anomalies, help them muddle through unanticipated situations and provide a mechanism for regulating their learning. The success of the MCL model of metacognition is illustrated in the air traffic control simulator program.

[1] This action was simulated by giving the plane a new goal using the plane's user interface.

VII. Acknowledgements

We would like to thank NSF and NASA for the grants that supported this research. We would like to acknowledge Donald Perlis, Michael Anderson, Tim Oates and Matthew Schmill for their contributions to the MCL model. We would also like to thank the undergraduate students (Fassil Molla, Michelle Snowden, Jalissa Miles, Ahmed Kamara and Chinmnso Maduka) in the Robust Intelligence Lab at Bowie State University for their help in this research.

References

[1] J. H. Flavell, "Metacognitive aspects of problem solving," in *The nature of intelligence*, L. B.Resnick, Ed. Hillsdale,NJ: Lawrence Erlbaum Associates, 1976, pp. 231–236.

[2] D. S. Ridley, P. A. Schutz, R. S. Glanz, and C. E. Weinstein, "Self-regulated learning: The interactive influence of metacognitive awareness and goal-setting," *Journal of Experimental Education*, vol. 60, no. 4, pp. 293–306, 1992.

[3] A. L. Wenden, "Metacognitive knowledge and language learning," *Applied Linguistics*, vol. 19, no. 4, pp. 515–537, 1998.

[4] W. P. Rivers, "Autonomy at all costs: An ethnography of metacognitive self-assessment and self-management among experienced language learners," *The Modern Language Journal*, vol. 85, no. 2, pp. 279–290, 2001.

[5] R. Isaacson and F. Fujita, "Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflections on learning," *Journal of the Scholarship of Teaching and Learning*, vol. 6, no. 1, pp. 39–55, 2006.

[6] J. M. O'Malley, A. U. Chamot, G. Stewner-Manzanares, R. P.Russo, and L. Küpper, "Learning strategy applications with students of english as a second language," *TESOL quarterly*, vol. 19, no. 3, pp. 557–584, September 1985.

[7] T. van Gog, F. Paas, J. J. Merriënboer, and P. Witte, "Uncovering the problem-solving process: Cued retrospective reporting versus concurrent and retrospective reporting," *Journal of Experimental Psychology: Applied*, vol. 11, no. 4, pp. 237–244, 2005.

[8] Y. Kafai, "Gender differences in childrens constructions of video games," in *Interacting with video games*, P. M. Greenfield and R. R. Cocking, Eds. Norwood, NJ: Erlbaum, 1996.

[9] S. Malcolm, "KineticCity," in *In Proceedings of the Annenberg Public Policy Summit on Childrens Media Policy*, St. Thomas, Virgin Islands, 2002.

[10] A. L. Foote and J. D. Crystal, "Metacognition in the rat," *Current Biology*, vol. 17, no. 6, pp. 551–555, March 2007.

[11] N. Kornell, L. K. Son, and H. S. Terrace, "Transfer of metacognitive skills and hint seeking in monkeys," *Psychological Science*, vol. 18, pp. 64–71, 2007.

[12] J. Ding, T. Kornblum, N. Kornell, and H. S. Terrace, "Prospective and retrospective metacognitive abilities in rhesus monkeys," *Columbia Undergraduate Science Journal*, vol. 2, no. 1, pp. 91–97, 2007.

[13] M. Schmill, D. Josyula, M. L. Anderson, S. Wilson, T. Oates, D. Perlis, and S. Fults, "Ontologies for reasoning about failures in AI systems," in *Proceedings from the Workshop on Metareasoning in Agent Based Systems at the Sixth International Joint Conference on Autonomous Agents and Multiagent Sytems*, 2007.

[14] M. L. Anderson, M. Schmill, T. Oates, D. Perlis, D. Josyula, D. Wright, and S. Wilson, "Toward domain-neutral human-level metacognition," in *Papers from the 2007 AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 2007. [Online]. Available: http://www.ucl.ac.uk/commonsense07/papers/anderson-et-al.pdf

[15] M. L. Anderson, S. Fults, D. P. Josyula, T. Oates, D. Perlis, M. D. Schmill, S. Wilson, and D. Wright, "A Self-Help Guide for Autonomous Systems," *AI Magazine*, 2008.