Process Algebra Approach for Action Recognition in the Maritime Domain

Terry Huntsberger, *Senior Member, IEEE*

***Abstract*—The maritime environment poses a number of challenges for autonomous operation of surface boats. Among these challenges are the highly dynamic nature of the environment, the onboard sensing and reasoning requirements for obeying the navigational rules of the road, and the need for robust day/night hazard detection and avoidance. Development of full mission level autonomy entails addressing these challenges, coupled with inference of the tactical and strategic intent of possibly adversarial vehicles in the surrounding environment. This paper introduces PACIFIC (Process Algebra Capture of Intent From Information Content), an onboard system based on formal process algebras that is capable of extracting actions/activities from sensory inputs and reasoning within a mission context to ensure proper responses. PACIFIC is part of the Behavior Engine in CARACaS (Cognitive Architecture for Robotic Agent Command and Sensing), a system that is currently running on a number of U.S. Navy unmanned surface and underwater vehicles. Results from a series of experimental studies that demonstrate the effectiveness of the system are also presented.**

I. Introduction

THE use of unmanned vehicles (example shown in Fig. 1) in maritime environments is increasing for a wide variety of military and commercial missions such as wide area scientific surveys, long duration monitoring, and deep underwater repairs. Survivability of the vehicles is an issue that needs to be explicitly addressed through robust hull designs, appropriate sensor suites, and a high level of onboard autonomy. The Jet Propulsion Laboratory (JPL) has developed a system called CARACaS (Cognitive Architecture for Robotic Agent Command and Sensing) [1, 2, 3] running onboard a number of US Navy unmanned surface vehicles (USV's) and underwater vehicles (UUV's) over the last seven years. CARACaS integrates the key components of an onboard autonomy system including an onboard Dynamic Planner for on-the-fly mission changes, a Behavior Engine that is able to adapt to rapidly changing conditions and learn about new situations, and a Perception sub-system that combines local sensing for hazard detection /avoidance with long range sensing for situational awareness. The core Behavior Engine is based on a Finite State Machine (FSM) framework that is running under the R4SA (Reconfigurable, Robust, Real-Time, Robotic Software Architecture), an embedded system derived from JPL flight technology that ensures a deterministic response to the dynamic maritime environment.

Fig. 1. US Navy Unmanned Sea Surface Vehicle (USSV) technology prototype used for the on-water testing of autonomy algorithms (Photo courtesy of NAVSEA, Carderock).

In order to process the incoming sensory information in a manner that is aligned with the CARACaS World View, a formal process algebra based on the Cost-Calculus ($-Calculus) of Eberbach [4] is used to map behaviors into a framework that is efficient for onboard reasoning and learning. A similar approach was used previously to develop the CCL (Common Control Language) [5, 6] that is running onboard a UUV at the Naval Undersea Warfare Center in Newport, Rhode Island. The reasoning/learning portion of the Behavior Engine for interpretation of the actions of other agents in the environment is called PACIFIC (Process Algebra Capture of Intent From Information Content). The process algebra is used to project the observed actions/activities of other agents onto the vehicle's internal behavior base for a "like-me" analysis [7]. For example, if two boats are obeying the international maritime Rules-of-the-Road (also called COLREGS [8]), during normal operations the behavior of the other boat is to a certain extent predictable. COLREGS violations are traceable to a number of possible factors, including limited relative visibility, driver impairment, adverse weather/sea conditions, and ignorance of the rules; or in the event of an adversarial environment, possible hostile actions. The current implementation of the COLREGS behaviors under CARACaS is described in a paper in these proceedings [9].

An example of the behavior mapping under CARACaS for COLREGS of a Head-On encounter between two boats is shown in Fig. 2. The competing behaviors for the encounter are *Waypoint Navigation* and *COLREGS*. An arbitration mechanism is used to decide the appropriate action to take given the sensory inputs. Among the arbitration mechanisms commonly used are subsumption [10], weighted voting [11], and multiple objective decision making [12], all of which are supported under CARACaS through the $-Calculus.

Manuscript received March 28, 2011. This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and was supported in part by the Office of Naval Research under Contract N00014-09-IP-2-0008.

Terry Huntsberger is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA (phone: 818-354-5794; fax: 818-393-5007; e-mail: terry.huntsberger@jpl.nasa.gov).

© 2011 California Institute of Technology. Government sponsorship acknowledged.

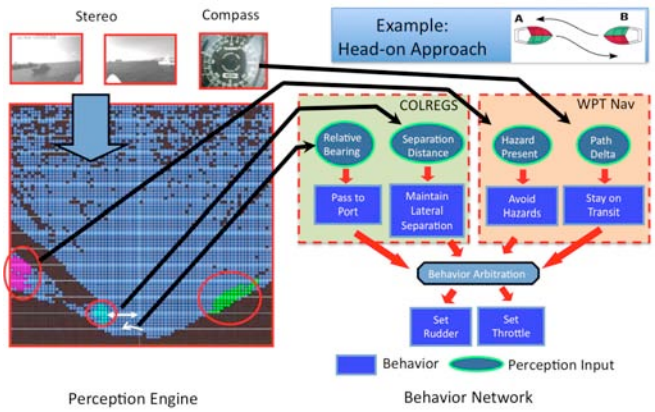

Fig. 2. Illustration of the links between the sensory inputs and action selection mechanism in CARACaS. This example is the behavior network for the satisfaction of COLREGS for the head-on encounter of two boats. The two possibly competing behaviors are *Waypoint Nav.* and *COLREGS*, which are arbitrated to determine which action to take. A single composite map for hazard detection (shown in lower left hand corner) is made from the four stereo cameras. The approaching boat for the head-on COLREGS is the blue blob circled in red at the bottom left of the composite map.

PACIFIC contains the reasoning mechanism within the process algebra to analyze the observed behavior of other agents within the current mission context, determine if there is a match to expected behaviors, and learn about new behaviors if there is not a match.

The next section briefly reviews the $-Calculus and the onboard learning mechanism. The following section describes some experimental studies that were run to test the framework reasoning and learning capabilities. The final section summarizes the results and describes the current research directions.

II. Process Algebra

A. Background

State formalisms and process algebras are at the core of many proposed approaches to representing, learning and explaining visual behaviors. These approaches have advantages in expressive power, especially with respect to representing interaction, intuitive notation and a strong mathematical foundation, and have been used successfully in representing and analyzing spatiotemporal patterns of behavior, primarily for surveillance of vehicles and humans. The Cost-Calculus ($-Calculus) [4], based on Milner's π-calculus of mobile concurrent processes, enriched with the concept of cost calculation and composition operators, is one example of a process algebra.

A Cost Calculus ($-Calculus) [4] is a model for resource bounded computation based on process algebras that:

- provides a means for generating incremental solutions for computationally hard, real-life problems,
- provides a uniform representation for the use of uncertain/unobservable information during the cost-optimization process (kΩ-optimization), and
- encapsulates most currently used search algorithms.

Behaviors are written as Cost-expressions ($-expressions) built using the algebraic operators of send/receive, cost assignment, defined simple/process calls such as choice, and sequential/parallel composition. An example of the mapping of the behavior network from Fig. 3 into the $-Calculus is given by:

$$\text{NAVIGATE} = (\text{WPT_Nav} \mid\mid \text{Satisfy COLREGS}), \quad (1)$$

where the parallel composition operator (| |) indicates the two behaviors are running with all allowable interleavings.

The process algebra in the $-Calculus is formally equivalent to a Labeled Transition System (LTS), which is convenient for the mapping of actions within the FSM of CARACaS into a framework that explicitly captures the link between vehicle behaviors and actions. A LTS is defined as a triple $(S, A, \{\xrightarrow{a} \mid a \in A\})$, where S is a set of states, A is a set of actions (labels), and each $\xrightarrow{a}$ is a subset of S x S called an action relation over S. A transition system with an initial state is called a *process*. A process algebra can be defined as a quadruple $(S, A, \{\xrightarrow{a} \mid a \in A\}, P)$, where S, A, $\xrightarrow{a}$, are defined as in the LTS, and $P \in S$ is the initial state. Each transition arc is also labeled with the cost ($) associated with the state-action pair in the transition. This cost is used for the onboard learning to weight the transition probabilities. Multiple possible transitions out of a state represent a *nondeterministic choice*, which is included in the $-Calculus using choice operators based on cost (minimal probability), adversary (maximal probability), or general (random).

The mapping of the behavior network shown in Fig. 2 into an LTS is given in Fig. 3, with transitions labeled as *Behavior*.{*action*}, where *Behavior* is either WPT or COLREGS, and the *actions* are Port, Stbd, Straight, or Continue indicating possible turn directions or execution of the *Behavior*. Each action-transition pair has a probabilistic cost associated with it (not shown in Fig. 3 for clarity), that is updated using standard Bayesian update rules.

The fast response required in the dynamic maritime environment is explicitly addressed through the kΩ optimization algorithm built into the $-Calculus. The kΩ optimization algorithm includes bounds on depth of search and tree branching during the inference matching process, and on the number of iterations for convergence to allow dynamic adaptive tailoring of processing under the $-Calculus. PACIFIC uses probability as its cost function in order to build a ranked set of hypotheses for prediction and

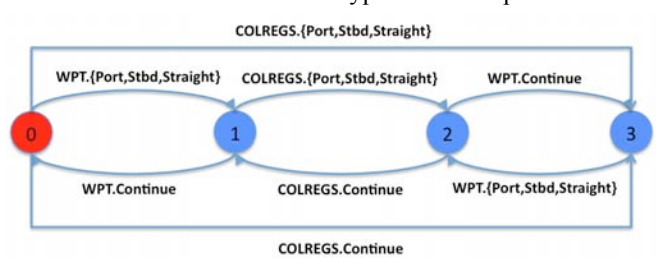

Fig. 3. LTS derived from the $-Calculus mapping of the Head-On Approach diagram shown in Fig. 2, including Waypoint Navigation (WPT) and COLREGS, with the actions associated with each type of behavior (Port, Stbd, Straight, Continue) prefixed by the behavior. Diagram generated using the LTSA (http://www.doc.ic.ac.uk/~jnm/book/ltsa/LTSA_applet.html).

interpolation. A previous study using PACIFIC extracted and identified the observed actions/activities of suited astronauts working in a simulated lunar environment with a success rate of over 80% [13]. TLTS (Timed Labeled Transition Systems) [14], which explicitly include temporal operators to capture the dynamics of state transitions between actions, are used as the underlying representation for the $-Calculus.

B. Onboard Learning

To date, there has been very little research into learning for behavior-based systems that are typically characterized by multiple, possibly conflicting goals [15]. The dominant learning strategy for single goal achievement such as robotic navigation has been reinforcement learning (RL), an unsupervised method that seeks to maximize a reward signal based on the utility of pairings of input and output states and their subsequent actions [16, 17, 18]. One of the most popular RL algorithms is Q-learning [19] and its variations such as Q-PSP [20], and hierarchical Q-learning [21]. The Q-learning rule is based on a utility function Q that has the update rule for new information:

$$Q(a,i) = R(i) + \sum_j M_{ij}^a \max_{a'} Q(a',j), \tag{2}$$

where Q is the expected utility in state i performing action a, R is the reward for being in state i, and M_{ij}^a is the probability to reach state j from state i by performing action a. This expression is rewritten in the operators of the $-Calculus as:

$$\begin{aligned} &\$(a,i) = \$(a,i) + \$(\sqcup_j(\Cup_{a'}(\circ a' j))), \\ &\$ \text{ - cost operator} \\ &\sqcup \text{ - general choice operator} \\ &\Cup \text{ - cost choice operator} \\ &\circ \text{ - sequential composition operator} \end{aligned} \tag{3}$$

The cost operator used in PACIFIC is Bayesian probability, and the RL algorithm is run using the kΩ optimization algorithm (bounded at a two step look-ahead with a branching factor of two) on the state transitions that dynamically occur as a sequence unfolds. Since the information needed to characterize an action-transition pair may not be available at each time step due to uncertainty in the sensory inputs, the $-Calculus provides an ε silent process that is used for the unknown states within the look-ahead horizon. The ε silent process does not participate in the RL analysis and serves as a placeholder until more information is available.

C. Related Work

There are a number of methods that have been proposed for action/activity recognition in sequences of sensor inputs. Among these are context free stochastic grammars [22, 23] extended grammars [24], and dynamic Bayesian networks [25]. These techniques suffer from exponential computational complexity as the number of states increases for complex environments, as opposed to the polynomial computational complexity of PACIFIC. In addition, the majority of the methods rely on fixed surveillance cameras, which is not the case for a moving USV. Good reviews of the wide range of methods can be found in Aggarwal and Ryoo [26], and in Turaga, et al. [27].

There have been numerous studies into the common ground between cognitive processing and formal process algebras [28,29]. The Event Calculus uses first order predicate logic to characterize actions with indirect effects, actions with non-deterministic effects, compound actions, concurrent actions, and continuous change. Reasoning is done based on changes in the values of features (*fluents*) and the temporal occurrence of events through predicates (such as *Happens*, *Holds*, etc.) in the environment. The Event Calculus uses a linear time structure, as opposed to the branching time structure in the Situation Calculus [30] and variants such as the Unifying Action Calculus [31] that explicitly handle non-determinism through the *Poss* predicate. These systems have typically been used to analyze toy problems such as the Yale Shooter and BlocksWorld, and thus haven't been used extensively in real-world environments.

Early work using process algebras to define robot schemas for behavior-based control of robotic platforms was done by Lyons and Arbib [32], and further extended by Benjamin, Lyons, and Lonsdale [33]. The $-Calculus was previously used in the development of the CCL (Common Control Language) for control of UUVs [5, 6]. This work used the $-Calculus infrastructure as the backend to an interpreter in order to generate the sequence of behaviors that best match the mission needs. This sequence was then uploaded to the vehicle for execution.

Continuous valued versions of the Q-learning algorithm have been developed to address the large state space problem [34, 35]. These works used a continuous Q-value derived from neural networks or other function approximation methods. The state space concerns were also addressed for deterministic environments using a forgetting mechanism in a penalty-based hierarchical Q-learning algorithm, which reduces the amount of state information that an agent must maintain by using a low level agent to maintain local state information and a high level agent to maintain global state [36]. Most of the RL studies to date have been confined to simulations and interior navigation in 2-D environments.

Most recently, learning of sequential behaviors for goal satisfaction through a blend of static and dynamic behavioral motivation modules has been demonstrated in simulation and on a commercially available AmigoBot in a lab setting [37]. This analysis used state prediction and the use of short-term memory (STM) and long-term memory (LTM) to store successful behavioral sequences following an action to learn the sequential behaviors. However, in the case of autonomous boat operations, the relationship between an action and a subsequent state is difficult to derive since it is closer to a non-deterministic process due to interactions with the water. Memory encoding is an effective technique for limiting the time needed for on-line learning, and is used in the PACIFIC RL algorithm. An onboard system for RL learning of navigation for planetary surface rovers traveling in rough terrain was introduced and demonstrated in the field by Huntsberger, Aghazarian, and Tunstel [38].

III. Experimental Study

In order to test the ability of PACIFIC to detect new activities and to learn about them, a number of data acquisition runs were done in a boat basin at Ft. Monroe in Virginia (area shown in Fig. 4). The sensor suite mounted on the USV included a set of six cameras that give a long range, continuous 360-degree view of the environment (top boxed area in Fig. 5), and a high resolution stereo vision system (lower boxed area in Fig. 5) that gives a short range view of the area in front of the USV for hazard detection and avoidance planning. Algorithmic and performance details of these systems can be found in [39, 40]. Examples of the output from the sensors are shown in Fig. 6, with the six cameras of the 360-degree sensor shown on the top, and the four cameras of the stereo sensor shown on the bottom.

Fig. 4. Boat basin study area at Ft. Monroe, VA is circled. The boat basin is 1500 meters long, so studies with a wide range of distances and situations are easier to stage.

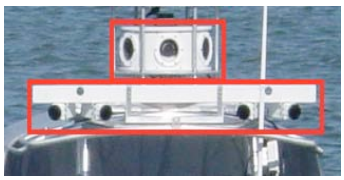

Fig. 5. Sensor suite used in the experimental studies. Inside the top boxed area is the 360-degree sensor head, and inside the lower boxed area is the high resolution stereo bar. The suite is mounted on an arch that is positioned midship.

A series of five sequences were collected, some of which had deliberate violations of COLREGS. Due to space limitations, only two of the analysis results will be presented in this paper. The reward function for the RL algorithm was defined in terms of the two measured variables: relative bearing (*rel_bear*) and lateral separation *(lat_sep)*. *Rel_bear* is defined as the bearing in degrees to a sensed boat with respect to the forward centerline of the USV, and *lat_sep* is defined as the perpendicular distance in meters between the sensed boat and the projected forward centerline of the USV. In order to obey COLREGS for a head-on encounter between two boats, both boats should stay to port (*rel_bear < 0 degrees*) of each other, and maintain a safe separation (*lat_sep = ~15 meters depending on speed*) to avoid collision.

Fig. 6. Examples of raw outputs from the six cameras of the 360-degree sensor head (top) and the four cameras of the dual stereo system (bottom).

The reward function for the RL algorithm is defined in terms of a weighted sum of changes to the *rel_bear* and *lat_sep* as:

$$R(t) = \alpha[rel_bear(t-1) - rel_bear(t)] + \beta[lat_sep(t) - speed(t)], \quad (4)$$

where *speed* (*t*) is the speed of the USV at time *t,* and α and β are weights used to prioritize the relative importance of the two terms. For the purposes of this study, α is fixed at 0.25 and β is fixed at 0.75. The second term includes the speed of the USV in order to indicate that higher speed encounters need more lateral separation in order to ensure safe operations. All sequences were analyzed in an off-line mode.

The first sequence (51 frames) satisfies COLREGS for a Head-On encounter and the second sequence (116 frames) violates COLREGS for a similar encounter. A 7-meter RHIB (Rigid Hull Inflatable Boat) approached the USV from the north end of the boat basin and stayed to the port side of the USV with an acceptable lateral offset during the first sequence, and initially stayed to the port side but then cut in front of the USV in the second sequence. The RHIB was detected and tracked in both sequences with the 360-degree sensors using the techniques described in [40]. A sample frame from the forward facing camera in the six-camera array and a summary sequence are shown in the top and bottom of Fig. 7 for the first sequence, and Fig. 8 for the second sequence.

For the purposes of this study, only the relative bearing directly extracted from the tracking data was used for the RL update of the utility function Q. A plot of the reward functions versus frame number for the two sequences is shown in Fig. 9. Of particular note is the distinct deviation from a uniform award in both cases – toward a higher reward for the first sequence where COLREGS was obeyed, and a sharp downward trend followed by negative reward (penalty) in the second sequence where COLREGS was violated.

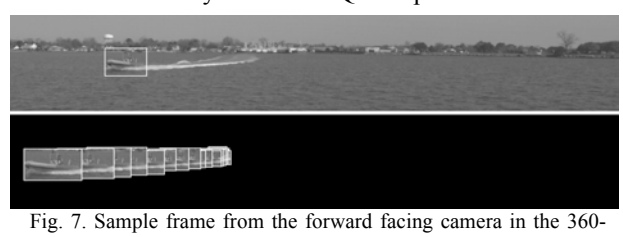

Fig. 7. Sample frame from the forward facing camera in the 360-degree sensor head (top), and a summary frame of the entire sequence sampled every fifth frame (bottom) for the first sequence. The center of the frame is aligned with the centerline of the USV.

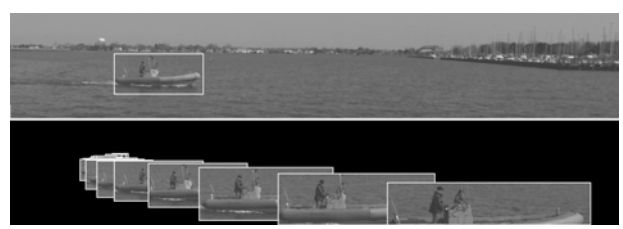

Fig. 8. Sample frame from the forward facing camera in the 360-degree sensor head (top), and a summary frame of the entire sequence sampled every fifth frame (bottom) for the second sequence. The center of the frame is aligned with the centerline of the USV.

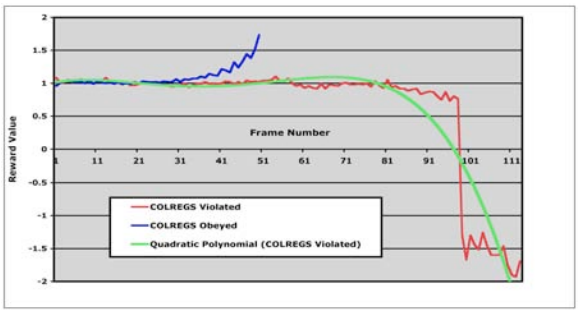

Fig. 9. Plot of RL reward function versus frame number for the first sequence (COLREGS obeyed) and the second sequence (COLREGS violated). A polynomial fit to the data is also shown for the second sequence reward function. Of particular notice is the deviation from the norm at around frame 85 and finally drop to a negative reward (penalty) in the second sequence.

The relatively simple four-state LTS shown in Fig. 2 for nominal waypoint navigation and COLREGS is expanded to eight states by the RL algorithm with the new state transitions added for the COLREGS violation. A totally nominal COLREGS sequence will transition from state-0 to state-3 and cycle in the baseline LTS shown in Fig. 3. Currently the RL algorithm adds new actions/activities using the $-Calculus parallel composition operator. The new states/transitions learned by the RL algorithm are unlabeled, but the system runs an internal search using the kΩ optimization algorithm to decide what actions should be associated for safe transitions.

The new LTS is shown in Fig. 10, where the response of "turn-to-port" for this particular violation (cutting across the bow) was the response incrementally learned by running the RL algorithm and evaluating if the USV path intersected that of the offending boat. The transitions that were visited (circled) in this study were a cycle from state-0 to state-6 and back for the nominal COLREGS (all frames in the first sequence, and up to frame 85 in the second sequence), and a cycle from state-0 to state-7 and back for the violation (frames 86 to 116 in the second sequence). The remainder of the COLREGS violation paths in the LTS were not visited or modified by the RL algorithm due to the reward values being too low.

IV. Summary & Current Directions

Action recognition in the maritime domain was demonstrated within the PACIFIC framework, and a previously unseen action/activity was dynamically learned using a reinforcement learning algorithm running within the same process algebra framework. Analysis of the other three data sequences that were collected using the new LTS in Fig 10 converged to the same conclusion with a match to the expected transitions. These results indicate that a process algebra framework is an efficient representation to use for formal reasoning and learning within the CARACaS behavior-based intelligent autonomy system running on a USV. Live runs with the USV on the water will further validate the utility of PACIFIC for dynamically learning about interactions with other agents within its environment.

Current directions include the analysis of about forty terabytes of data logged from the USV running through various mission scenarios and under a wide variety of sea states/weather environments over the last three years to further tune PACIFIC. A stereo dataset (gives both lateral separation and relative bearing information) that was analyzed but not reported on in this paper indicated that the dominant variable for learning is relative bearing. Verification of this finding would allow further optimization of the algorithms. The large amount of states that are generated in the LTS as each new type of action is added can be somewhat mitigated by pruning states/transitions that are relatively rare, and relearning them on-the-fly.

The addition of explanation capabilities to PACIFIC would enable the USV to describe the new actions/activities that it learned about in order for a human to label the new information (and possibly add tactical/strategic information tags). A dynamic decision tree decomposition [41] of the observed behaviors can be used to generate a set of rules for explanation using information gain and pruning to limit the size of the tree.

Acknowledgment

The author would like to thank the crew at the Naval Surface Warfare Center, Carderock (in particular the boat onboard safety officer Sam Calabrese), the staff of Spatial Integrated Systems, and Yoshi Kuwata, Chris Assad, and Michael Wolf of JPL, for their support during the on-water data acquisition for this study. The author would also like to thank Dr. Robert Brizzolara of the Office of Naval Research for his continuing support of the Maritime Intelligent Autonomy technology development at JPL.

References

[1] T. Huntsberger, H. Aghazarian, A. Castano, G. Woodward, C. Padgett, D. Gaines, and C. Buzzell, "Intelligent Autonomy for Unmanned Sea Surface and Underwater Vehicles," *Proc. AUVSI North America 2008 Unmanned Vehicles Conference*, San Diego, CA, June 2008.

[2] T. Huntsberger, H. Aghazarian, D. Gaines, M. Garrett, & G. Sibley, "Autonomous Operation of Unmanned Surface Vehicles (USV's)," *Proc. IEEE ICRA Workshop on Robots in Challenging and Hazardous Environments*, Rome, Italy, April 2007.

[3] E. Hansen, T. Huntsberger, and L. Elkins, "Autonomous Maritime Navigation - Developing Autonomy Skill Sets for USVs," *Proc. SPIE Defense and Security Symposium 2006*, Orlando, FL, April, 2006.

[4] E. Eberbach, "$-Calculus of Bounded Rational Agents: Flexible Optimization as Search under Bounded Resources in Interactive Systems," *Fundamenta Informaticae*, vol. 68, pp. 47-102, 2005.

[5] C. M. Buzzell, *A Common Control Language for Multiple Autonomous Undersea Vehicle Cooperation*, Master's Thesis, University of Massachusetts, Dartmouth, 2004.

[6] C. N. Duarte, G. R. Martel, C. Buzzell, R. Komerska, S. Mapparapu, S. Chapel, D. R. Blidberg, and R. Nitzel, "A Common Control Language To Support Multiple Cooperating AUVs," *Proc. 14th Int. Symposium on Unmanned Untethered Submersible Technology*, 2005.

[7] W. G. Kennedy, M. D. Bugajska, A. M. Harrison, J. G. Trafton, "'Like-Me' Simulation as an Effective and Cognitively Plausible Basis for Social Robotics," *Int. J. of Social Robotics*, vol. 1, pp. 181-194, 2009.

[8] U. C. G. Commandant, "International Regulations for Prevention of Collisions at Sea," 1972 (1972 COLREGS), US Department of

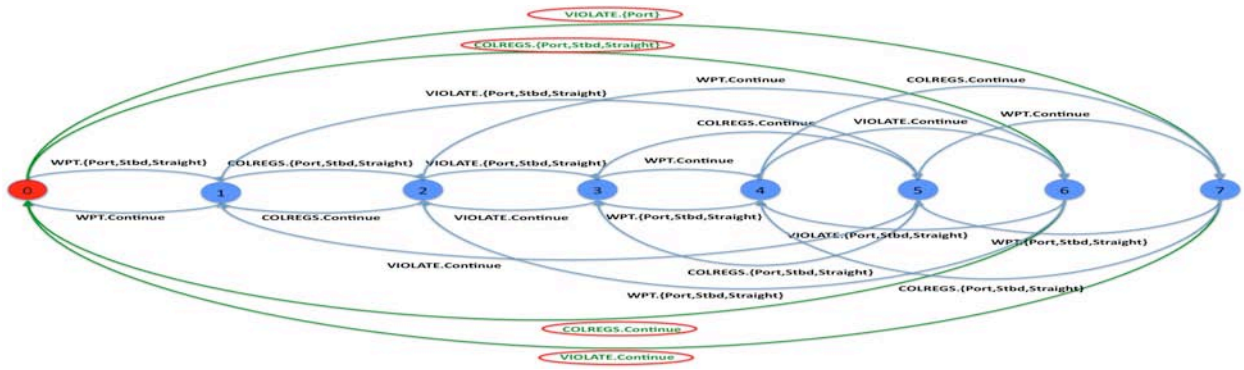

Fig. 10. LTS for the navigation and COLREGS behaviors after application of the RL algorithm to the COLREGS violation in the second sequence (sample image seen in Fig. 8). The system started with the LTS shown in Fig. 3, and new states were learned by the RL algorithm when the reward function indicated that transitions in the baseline LTS no longer matched the information retrieved from the sensors (around frame 85 in Fig. 8). The new LTS was generated using the parallel composition operator from the $-Calculus. Only the transitions (circled) visited based on the reward function values were updated as to the most probable action leading to a safe transition (in this case turn to port to avoid the boat cutting across the bow). Diagram generated using the LTSA (http://www.doc.ic.ac.uk/~jnm/book/ltsa/LTSA_applet.html).

Transportation, US Coast Guard, August 1999, COMMANDANT INSTRUCTION M16672.2D.

[9] Y. Kuwata, M. Wolf, D. Zarzhitsky, and T. Huntsberger, "Safe Maritime Navigation with COLREGS Using Velocity Obstacles," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11)*, San Francisco, CA, submitted for publication.

[10] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[11] J. Rosenblatt, "The distributed architecture for mobile navigation," *J. of Experimental & Theoretical AI*, vol. 9, no. 2/3, pp. 339–360, 1997.

[12] M. R. Benjamin, *Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control*, Ph.D. Thesis, Dep. of Computer Science, Brown University, Providence, RI, 2002.

[13] T. Huntsberger, "Cognitive Architecture for Mixed Human-Machine Team Interactions for Space Exploration," *Proc. IEEE Conf. on Aerospace*, Big Sky, MT, March, 2011.

[14] J. Y. Lee and J. Zic, "On Modeling Real-time Mobile Processes," *Proc. 25th Australasian Computer Science Conference (ACSC2002)*, Melbourne, Australia, 2002.

[15] P. Maes, (Ed.) *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, MIT Press, 1991.

[16] L.P. Kaelbling, *Learning in Embedded Systems*. MIT Press, Cambridge, MA, USA, 1993.

[17] L.P. Kaelbling, M.L. Littman, and A.W. Moore, "Reinforcement Learning: A survey," *J. of AI Research*, vol. 4, pp. 237-285, 1996.

[18] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[19] C.J. Watkins, *Learning from Delayed Rewards*, Ph.D. dissertation, Cambridge Univ., Cambridge, UK, 1989.

[20] T. Horiuchi, A. Fujino, A., Katai, O., and T. Sawaragi, "Q-PSP Learning: An exploitation-oriented Q-learning algorithm and its applications," *Proc. IEEE International Sympos. on Evolutionary Computation*, pp. 76-81, 1996.

[21] L.-J. Lin, *Reinforcement Learning for Robots Using Neural Networks*. Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1993.

[22] D. Minnen, I. Essa, and T. Starner, "Expectation grammars: Leveraging high-level expectations for activity recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03)*, 2003.

[23] S. Hongeng and R. Nevatia, "Multi-agent event recognition," in IEEE International Conference on Computer Vision Proceedings, Vol. 2, Vancouver, Canada, 2001, pp. 84-91, 2001.

[24] Z. Zhang, T. Tan, and K. Huang, "An Extended Grammar System for Learning and Recognizing Complex Visual Events," *IEEE Trans. On PAMI*, vol. 33, no. 2, pp. 240-255, February, 2011.

[25] T. Huang, D. Koller, J. Malik, G. H. Ogasawara, B. Rao, S. J. Russell, and J. Weber, "Automatic symbolic traffic scene analysis using belief networks," *Proc. National Conference on AI*, pp. 966–972, 1994.

[26] J.K. Aggarwal and M. S. Ryoo, "Human Activity Analysis: A Review," *ACM Computing Surveys*, 2011.

[27] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473-1488, November 2008.

[28] M. Shanahan and C. M. Witkowski, "Event Calculus Planning Through Satisfiability," *J. of Logic and Computation*, vol. 14, pp. 731-745, 2004.

[29] M. P. Shanahan, "The Event Calculus Explained," Artificial Intelligence Today, (ed. M.J.Wooldridge and M.Veloso), *Springer Lecture Notes in Artificial Intelligence*, vol. 1600, pp. 409-430, 1999.

[30] J. McCarthy, "Situations and Actions and Causal Laws," Stanford AI Project, Memo 2, Stanford University, CA, 1963.

[31] M. Thielscher, "A unifying action algebra," *Artificial Intelligence*, vol. 175, pp. 120-141, 2011.

[32] D. Lyons and M. Arbib, "A Formal Model of Computation for Sensory-Based Robotics," *IEEE Trans. on Robotics and Automation*, vol. 6, no.3, pp. 280-293, 1989.

[33] D. P. Benjamin, D. Lyons, and D. Lonsdale, "ADAPT: A Cognitive Architecture for Robotics," *Proc. 2004 International Conference on Cognitive Modeling (ICCM)*, 2004.

[34] C. Gaskett, D. Wettergreen, and A. Zelinsky, "Q-learning in continuous state and action spaces," *Proc. 12th Australian Joint Conf. on AI*, Sydney, Australia, 1999.

[35] Y. Takahashi, M. Takada, and M. Asada, "Continuous valued Q-learning for vision-guided behavior acquisition," *Proc. International Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 716-721, 1999.

[36] G. Yen and T. Hickey, "Reinforcement learning algorithms for robotic navigation in dynamic environments," In *Proc. 2002 Intern. Joint Conf. on Neural Networks (IJCNN '02), Vol. 2*, pp. 1444-1449, 2002.

[37] I.H. Suh, M.J. Kim, S. Lee, and B.J. Yi, "A Novel Dynamic Priority-Based Action-Selection-Mechanism Integrating a Reinforcement Learning," *Proc. 2004 IEEE International Conf. on Robotics and Automation (ICRA2004)*, New Orleans, LA, pp. 2639-2646, 2004.

[38] T. Huntsberger, H. Aghazarian, and E. Tunstel, "Onboard Adaptive Learning for Planetary Surface Rover Control in Rough Terrain," *Proc. 2005 IEEE International Conference on Robotics and Automation (ICRA'05),* Barcelona, Spain, April, 2005.

[39] T. Huntsberger, H. Aghazarian, A. Howard, and D. Trotz, "Stereo Vision Based Navigation for Autonomous Surface Vessels," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 2*, vol. 28, No. 1, pp. 3-18, 2011.

[40] M.T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu, and T. Huntsberger, "360-Degree Visual Detection and Target Tracking on an Autonomous Maritime Patrol Vessel," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 1*, vol. 27, No. 6, pp. 819-833, 2010.

[41] P. E. Utgoff, N. C. Berkman, and J. A. Clouse, "Decision Tree Induction Based on Efficient Tree Restructuring," *Machine Learning*, vol. 29, pp. 5-44, 1997.