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.

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 ($-\text{Calculus}$) 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 $-\text{Calculus}$. © 2011 California Institute of Technology. Government sponsorship acknowledged.
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 $\pi$-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 ($k\Omega$-optimization), and
- encapsulates most currently used search algorithms.

Behaviors are written as Cost-expressions ($S$-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:

$\text{NAVIGATE} = (\text{WPT.Nav} \mid \text{Satisfy COLREGS})$, (1)

where the parallel composition operator $\mid$ 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 $(S, A, \{ \rightarrow \mid a \in A \})$, where $S$ is a set of states, $A$ is a set of actions (labels), and each $\rightarrow$ is a subset of $S \times S$ 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 $(S, A, \{ \rightarrow \mid a \in A \}, P)$, where $S$, $A$, $\rightarrow$, 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.$\{\text{action}\}$, 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 $k\Omega$ optimization algorithm built into the $\$\$-Calculus. The $k\Omega$ 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
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 S-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:

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

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 S-Calculus as:

$$S(a,i) = S(a,i) + S(\cup_j (\psi_{a,i}(\alpha^a j))),$$

$S$ - cost operator
\$ - general choice operator
$\cup$ - cost choice operator
$\circ$ - sequential composition operator

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 S-Calculus provides an $\epsilon$ silent process that is used for the unknown states within the look-ahead horizon. The $\epsilon$ 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 Ryo [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 $S$-Calculus was previously used in the development of the CCL (Common Control Language) for control of UUVs [5, 6]. This work used the $S$-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.

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.

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:

\[ R(t) = \alpha [\text{rel}_\text{bear}(t-1) - \text{rel}_\text{bear}(t)] + \beta [\text{lat}_\text{sep}(t) - \text{speed}(t)], \]

where speed(t) is the speed of the USV at time t, and \( \alpha \) and \( \beta \) are weights used to prioritize the relative importance of the two terms. For the purposes of this study, \( \alpha \) is fixed at 0.25 and \( \beta \) 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.
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 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


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.

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-7 and back for the violation cycle from state-0 to state-7 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.

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 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.
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 S-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.


