State Machine Execution Traces for Verifying and Validating Robot Behaviors

Tyler Errico[a], Kristin Giammarco[b], Pamela Dyer[b], Michael (Misha) Novitzky[a], John James[a], Rob Semmens[a], Michael Collins[c], and Stuart Harshbarger[c]

[a]USMA Robotics Research Center, West Point, NY, USA
[b]Naval Postgraduate School, Monterey, CA, USA
[c]Department of Defense, Fort Meade, MD, USA

ABSTRACT

This work demonstrates that a novice user of the Monterey Phoenix (MP) tool can encode their expertise of robot mission tasks in the form of a finite state machine. State machine diagrams are among several approaches available for describing behavior. They are appropriate for capturing and understanding behavior rules of systems that can be described in terms of possible states and state transitions. Tools that offer state machine automation enable modelers to test the activation of states and trace the transitions between states, providing assistance with verification and validation of the behavior logic. The United States Military Academy (USMA) at West Point's Robotics Research Center (RRC) in collaboration with the Naval Postgraduate School (NPS), developed a minimal executable state machine model to test its implementation of behavior logic for a single robot agent in Project Aquaticus, a human-robot teaming capture the flag competition. Monterey Phoenix, a formal behavior modeling language, approach, and tool was used to express the behavior logic of the robots participating in this game as a finite state machine model. MP was used to generate every possible trace through the state machine behavior logic up to the specified scope limit. This model informed the team's understanding of the desired states for the single robot competitor in Aquaticus, assisted in identifying behavior improvements, and identifying dependencies among behaviors and where the process could possibly go wrong. In essence, this work demonstrates that encoding the finite state machine of the single robot tasks in MP allows experts to verify and validate expected robot behaviors and adjust as needed. Future work will look to expand from the single agent modeling to multi-agent modeling in MP to generate and verify possible sequences of events for the Aquaticus competition between competing teams. By using this modeling method to understand the state of each actor and their response to inputs from other actors, the team expects to improve the communication within the human-robot team and ultimately achieve a better understanding of command intent in complex, contested environments. Today's mission-critical systems are actually, "systems of systems" with complexity that will surpass the designer's cognitive ability to anticipate all interactions. Without tools, languages, and methodologies to analyze system-level behavior early in the design, mission systems can be at risk of emergent behaviors that may negatively impact safety and security. Our team believes that the MP environment can assist commanders in anticipating potentially unsafe or insecure emergent behaviors of human-robot teams in complex, contested environments and adjust human intent/command intent accordingly for allowable combat crew drill behaviors required to achieve assigned tasks and missions.

Keywords: event streams, finite state machine, formal methods, scenario generation, behavior modeling, Monterey Phoenix, Aquaticus, human-robot teaming, human intent, command intent, verification and validation, emergent behaviors

1. INTRODUCTION

The exhaustive state machine execution investigations described in this article were carried out during the 2022 Monterey Phoenix Virtual Internship Program (MPVIP), a summer program started by the National Security

Further author information: (Send correspondence to Tyler Errico)
Tyler Errico: E-mail: Tyler.Errico@westpoint.edu, Telephone: 1 845 938 5062

Open Architecture/Open Business Model Net-Centric Systems and Defense Transformation 2023, edited by Raja Suresh, Proc. of SPIE Vol. 12544, 125440W · © 2023 SPIE
0277-786X · doi: 10.1117/12.2662585

Agency (NSA) National Cryptologic School (NCS) in 2020 and 2021 and supported by the Naval Research Program (NRP) in 2022 and 2023. The 2022 MPVIP projects sought to develop improved behavior models of artificial intelligence/ machine learning (AI/ML)-enabled platforms competing in the Project Aquaticus capture-the-flag (CTF) opposing force robotic games[1,2] at the multi-domain operations (MDO) human-robot teaming sandbox (MDO-HuRT-S) at Lake Popolopen on the United States Military Academy, West Point, NY military reservation. The other two projects from the 2022 MPVIP are MOOS-IvP Integration[3] and SAR Expansion,[4] which are discussed in detail in separate papers of this conference. The State Machine team objectives follow.

HUMAN-ROBOT TEAMING – STATE MACHINE team analysis objective:

- ***Motivation and Context***: The finite state machine diagrams generated by Monterey Phoenix (MP)[5] models provide a familiar capability for humans to accurately understand and verify allowable sequences of events for the process at hand. By understanding the state of each actor in the capture-the-flag scenario and how they will respond to inputs from the other various actors, we aim to increase the communication between the robot human team. The expected result is that humans will have a better understanding of their robot teammates, and be able to make more informed decisions.

- ***Analysis Objective***: Modify an existing MP model of capture-the-flag game play to generate a finite state machine for a single agent from the behavior logic (behavior dynamics/event streams of the logical dynamics of moving from state to state), so that the model more closely resembles the usual finite state machines with nodes (states) and edges (transitions).

- ***Typical Behavior***: A finite-state machine generated from a set of all possible event streams for a single agent playing the CTF Aquaticus game.

2. BACKGROUND

Monterey Phoenix[6] is being used to enable human experts in human-robot teaming activities to specify, test, and control behaviors of these teams with the goal of capturing human intent for a given task or mission. MP generates event streams generated from an executable schema that is shaped with constraints to allow desired robot behaviors and flag or reject undesired behaviors during human-robot teaming activities. We assume that every human-robot team can be adequately modeled as a system of systems (SOS) composed as a hierarchy of compositions of asynchronous logical (discrete-time) components/models (such as MP models) and synchronous physical (continuous time) components/models.

In collaboration with the MDO-HuRT-S development efforts, a MPVIP event was held in the summer of 2022 to guide each intern team in stating their analysis objective in terms of (1) the motivation and context of the MP model being constructed, (2) the analysis objective of the MP model being constructed, and (3) a typical behavior of the MP model being constructed. The NSA Laboratory for Advanced Cybersecurity Research (LACR) supported this work along with basic research efforts for development of tools, languages, and methodologies necessary to analyze system level behavior early in the design of future systems.

Our set of three MP model experiments discussed at this SPIE conference were all supported by the NSA as a small part of a considerably larger set of efforts seeking to extend current capabilities for estimating distributed states and controlling distributed behaviors of sets of distributed Cyber-Physical Systems (CPS). These efforts of improving current capabilities of state estimation and control of distributed CPS are summarized in the NSA roadmap in Figure 1 for moving from resilient state estimation and control of CPS to understanding (predicting) emergent behaviors of swarms of Artificial Intelligence / Machine Learning (AI/ML) -enabled platforms.

There is a recurring interplay between the use of Monterey Phoenix discrete-event models of command intent/human intent for system dynamic behaviors and the questions which systems engineers need to answer for acceptable emergent behaviors of a SOS. These behaviors are compositions of logical system components (e.g., computers, software components, networks) and physical system components (e.g., electromagnetic waves, robots, ships, planes). The analysis questions in focus are:[7]

- What does the system NEED to do?
- What does the system actually do?

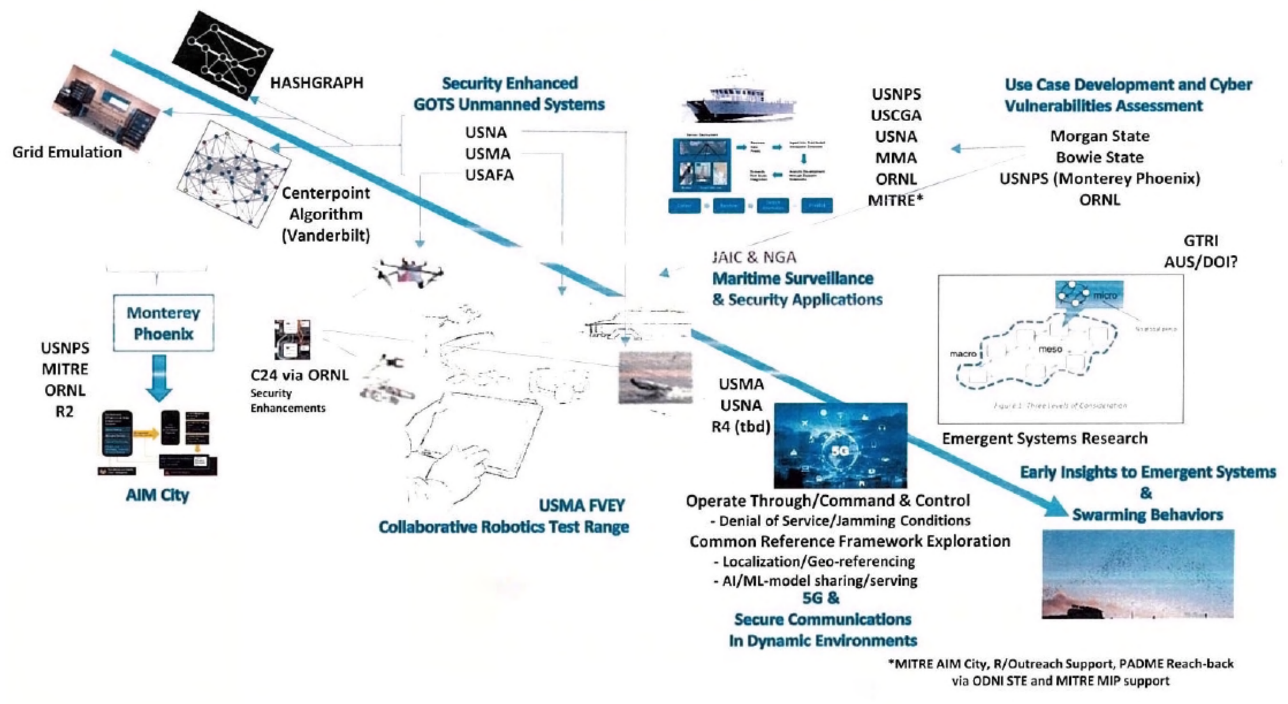

HASHGRAPH
Grid Emulation
Centerpoint Algorithm (Vanderbilt)
Security Enhanced GOTS Unmanned Systems
USNA
USMA
USAFA
USNPS
USCGA
USNA
MMA
ORNL
MITRE*
Use Case Development and Cyber Vulnerabilities Assessment
Morgan State
Bowie State
USNPS (Monterey Phoenix)
ORNL
JAIC & NGA
Maritime Surveillance & Security Applications
GTRI
AUS/DOI?
Monterey Phoenix
USNPS
MITRE
ORNL
R2
AIM City
C24 via ORNL
Security Enhancements
USMA
USNA
R4 (tbd)
Emergent Systems Research
micro
meso
macro
Early Insights to Emergent Systems & Swarming Behaviors
USMA FVEY
Collaborative Robotics Test Range
Operate Through/Command & Control
- Denial of Service/Jamming Conditions
Common Reference Framework Exploration
- Localization/Geo-referencing
- AI/ML-model sharing/serving
5G & Secure Communications In Dynamic Environments
*MITRE AIM City, R/Outreach Support, PADME Reach-back via ODNI STE and MITRE MIP support

Figure 1. NSA Roadmap for moving from resilient control of distributed CPS to insights into emergent behaviors of swarms and emergent behaviors of systems.

- What does the system NEED NOT to do?
- What does the system still allow to be done?

Understanding this interplay can be viewed as a central goal of team future emergent behavior research for human-robot team interactions for the MDO-HuRT-S facility. A four-step methodology for emergent behavior analysis[8] has been synthesized from past MP modeling efforts, and provides a foundation for emergent behavior analysis and research.

3. METHOD

This section outlines the approach to verifying and validating the maritime robot behaviors for a single agent playing Aquaticus CTF using a finite state machine in Monterey Phoenix. We used MP-Firebird,[9] a web-based Monterey Phoenix program developed by the Naval Postgraduate School and MP-Gryphon[10], a locally installable graphical user interface. MP-Firebird was used in the original analysis and MP-Gryphon was used to generate the graphs in Figures 3, 4, and 5. Both tools have similar features for creating and exploring models of behavior.

3.1 Describe the Behavior Logic

One first needs to understand the various states/behaviors that the robots could be in. As a very simple example: A robot has two states ON and OFF. Then list the states/behaviors in the logical order in which they would be experienced. i.e. The robot starts in the OFF state and is then turned ON. Draw branches from one behavior to the other where it is possible for one behavior to lead into the other given the appropriate input or stimulus. Label those branches (AKA Transitions) with the appropriate input or stimulus that would cause one state to transition into the other. i.e. branch can be labeled user presses ON/OFF button, because pressing that button would cause the robot to transition from one state to the other. This process resulted in the draft finite state machine (FSM) for a single robot agent playing the Aquaticus CTF, see Figure 2.

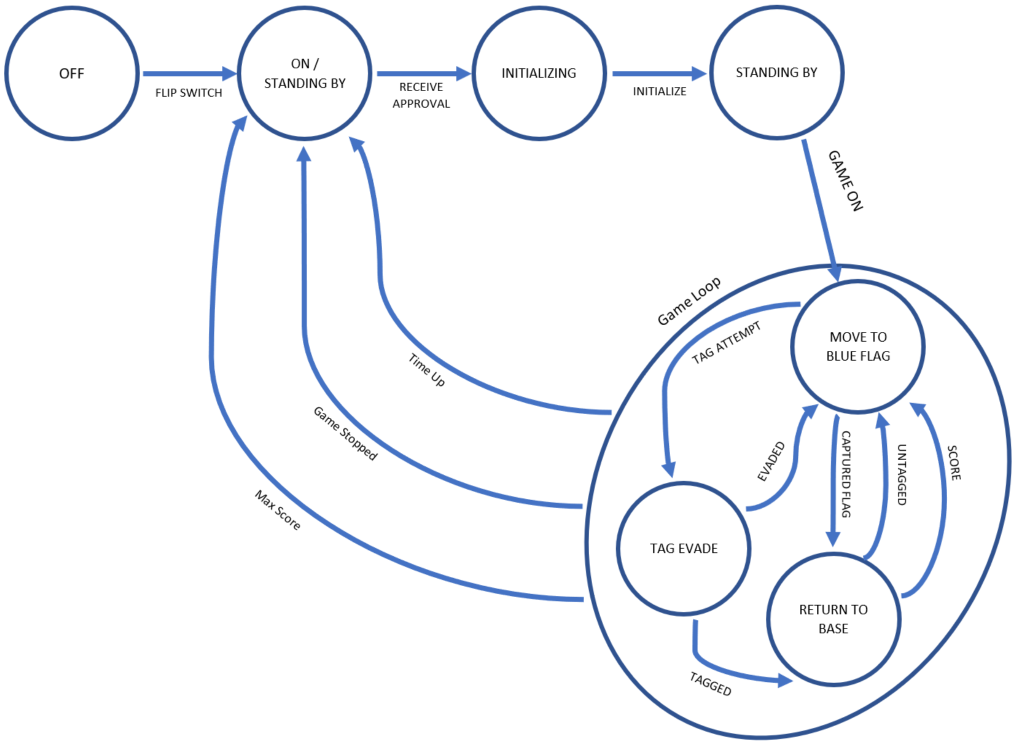
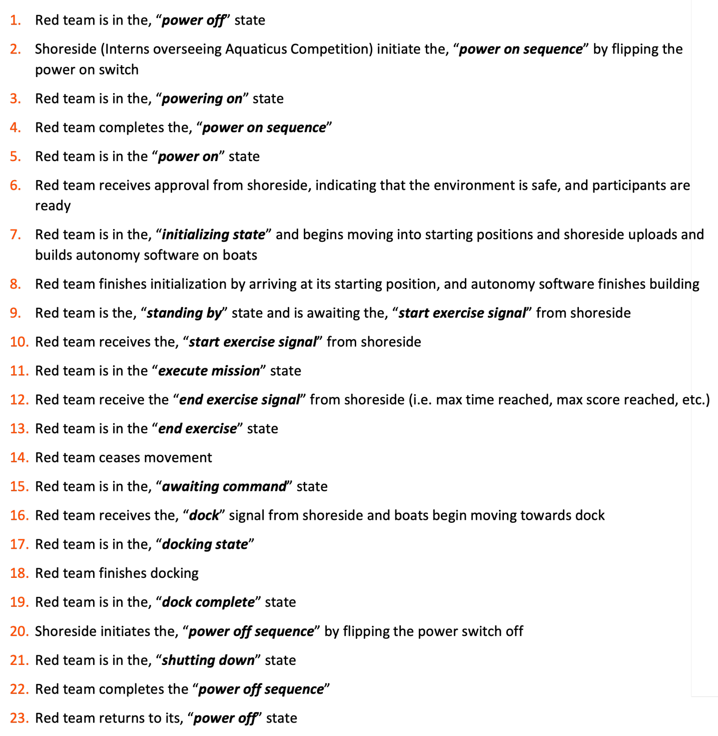


Figure 2. Draft state chart describing a single agent playing the Aquaticus CTF game.

The initial intent of this work was for the the draft FSM, seen in Figure 2, to be generated by a single MP model. However, the MP model created in this effort, seen in Figure 5, represents a minimal FSM where the "Game Loop" is represented as a single state called "RED TEAM EXECUTING." Future work will address implementing the "Game Loop" of the single agent playing the Aquaticus CTF by implementing another MP model which will create a hierarchy of FSMs. Thus, the states within the "Game Loop" such as "MOVE TO BLUE FLAG," "TAG EVADE," and "RETURN TO BASE" and transitions between the game playing states represented in Figure 2 will be modeled as another FSM in MP.

3.2 Model a Baseline Case

This team constructed a State Machine Diagram in MP on the topic of a single agent playing the Aquaticus CTF game, and presented[11] all compiled findings to leadership from NPS, USMA, MIT, NSA, and MARFORCYBER on the last day of the internship. This participant from the USMA created an MP model demonstrating all the detailed steps that a single robot on the Red Team could follow in the actual game of Aquaticus. This model contained all information needed to capture the full exercise, starting with the robots powering on, to then executing the mission (game) with optional pause and reset sequences, to finishing with the robots powering off. Figure 3 shows the typical event flow for what was determined to be the baseline case narrative from this model (Trace 1): the Red Team robots power on, receive approval for the mission, initialize, then execute the mission, receive the signal to dock once the mission is complete, then dock, and finally power off.



1. Red team is in the, "***power off***" state
2. Shoreside (Interns overseeing Aquaticus Competition) initiate the, "***power on sequence***" by flipping the power on switch
3. Red team is in the, "***powering on***" state
4. Red team completes the, "***power on sequence***"
5. Red team is in the "***power on***" state
6. Red team receives approval from shoreside, indicating that the environment is safe, and participants are ready
7. Red team is in the, "***initializing state***" and begins moving into starting positions and shoreside uploads and builds autonomy software on boats
8. Red team finishes initialization by arriving at its starting position, and autonomy software finishes building
9. Red team is the, "***standing by***" state and is awaiting the, "***start exercise signal***" from shoreside
10. Red team receives the, "***start exercise signal***" from shoreside
11. Red team is in the "***execute mission***" state
12. Red team receive the "***end exercise signal***" from shoreside (i.e. max time reached, max score reached, etc.)
13. Red team is in the "***end exercise***" state
14. Red team ceases movement
15. Red team is in the, "***awaiting command***" state
16. Red team receives the, "***dock***" signal from shoreside and boats begin moving towards dock
17. Red team is in the, "***docking state***"
18. Red team finishes docking
19. Red team is in the, "***dock complete***" state
20. Shoreside initiates the, "***power off sequence***" by flipping the power switch off
21. Red team is in the, "***shutting down***" state
22. Red team completes the "***power off sequence***"
23. Red team returns to its, "***power off***" state

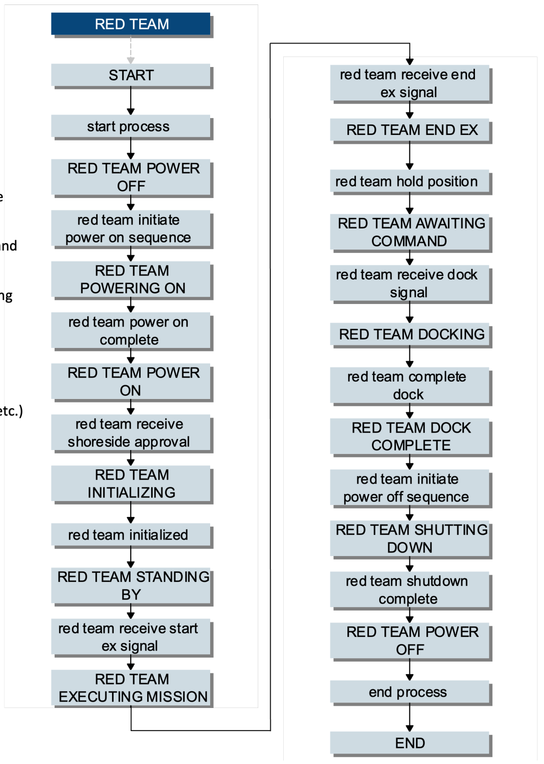

RED TEAM
START
start process
RED TEAM POWER OFF
red team initiate power on sequence
RED TEAM POWERING ON
red team power on complete
RED TEAM POWER ON
red team receive shoreside approval
RED TEAM INITIALIZING
red team initialized
RED TEAM STANDING BY
red team receive start ex signal
RED TEAM EXECUTING MISSION
red team receive end ex signal
RED TEAM END EX
red team hold position
RED TEAM AWAITING COMMAND
red team receive dock signal
RED TEAM DOCKING
red team complete dock
RED TEAM DOCK COMPLETE
red team initiate power off sequence
RED TEAM SHUTTING DOWN
red team shutdown complete
RED TEAM POWER OFF
end process
END

Figure 3. The baseline case narrative describes this baseline case and a typical event flow provides a typical sequence of events.[12]

A minimal version of the draft state chart seen in Figure 2, was then realized as a baseline Monterey Phoenix model which was used to generate the baseline case narrative and typical sequence of events shown in Figure 3.

3.3 Model Alternative Cases

The basic flow of enabling domain experts to capture the behaviors of their domains in MP models is to build alternative case narratives (scenarios) from the baseline case/scenario. Two needed alternative cases are to have the game pause and to have the game reset.

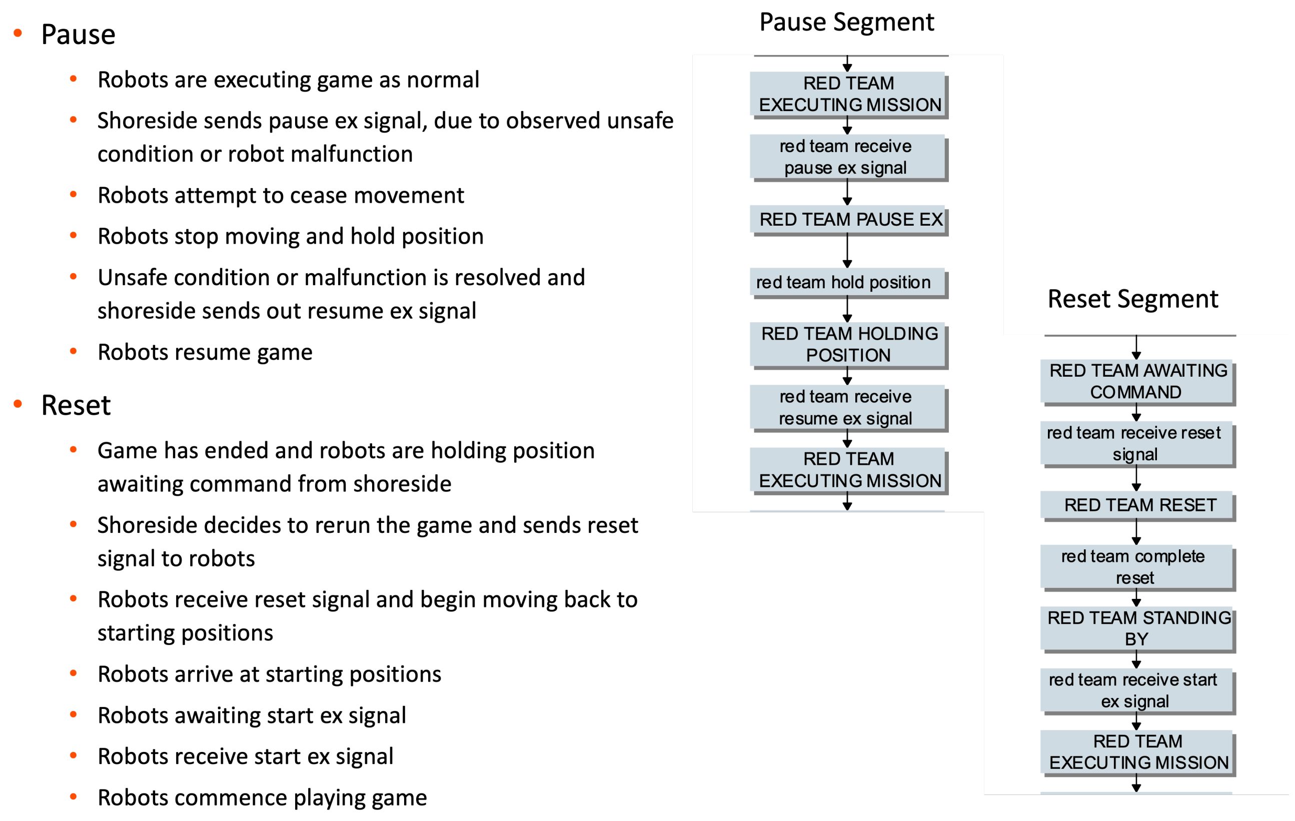
Figure 4 shows the event flow for a different trace from this MP model, which is an alternative case narrative (Trace 5). In addition to the steps from the baseline case shown in Figure 3 , this trace shows:

1) The Red Team pausing the execution of the mission, holding, and then resuming it again. The example story for this could be that there was an observed unsafe condition or robot malfunction out in the water, Shoreside sent out the appropriate assistance, the situation was then resolved, so the mission was approved to resume.

2) When the first game is over, the Red Team receives a reset signal and a second game is played. After that is finished, the robots then go through the normal docking and powering off steps.

3.4 Generate the Finite State Machine

Using MP's notation for constructing state machine diagrams, the appropriate categories were completed and added into the model. It is important to note that the names of all "state" events were capitalized, and the names of all "transition" events were given all lowercase letters. This was to make it very clear that events in this model always alternate back and forth between states and transitions, with no exceptions. Running the model in MP with the state machine additions incorporated into it produced the diagram for a single Red Team robot playing the Aquaticus CTF game that can be seen in Figure 5. All of the unique paths that the single Red



- Pause
  - Robots are executing game as normal
  - Shoreside sends pause ex signal, due to observed unsafe condition or robot malfunction
  - Robots attempt to cease movement
  - Robots stop moving and hold position
  - Unsafe condition or malfunction is resolved and shoreside sends out resume ex signal
  - Robots resume game
- Reset
  - Game has ended and robots are holding position awaiting command from shoreside
  - Shoreside decides to rerun the game and sends reset signal to robots
  - Robots receive reset signal and begin moving back to starting positions
  - Robots arrive at starting positions
  - Robots awaiting start ex signal
  - Robots receive start ex signal
  - Robots commence playing game

Pause Segment
RED TEAM EXECUTING MISSION
red team receive pause ex signal
RED TEAM PAUSE EX
red team hold position
RED TEAM HOLDING POSITION
red team receive resume ex signal
RED TEAM EXECUTING MISSION
Reset Segment
RED TEAM AWAITING COMMAND
red team receive reset signal
RED TEAM RESET
red team complete reset
RED TEAM STANDING BY
red team receive start ex signal
RED TEAM EXECUTING MISSION

Figure 4. Alternative case narratives and corresponding event flow segments for game Pause and game Reset events.[12]

Team robot could follow throughout the Aquaticus game are visible in the diagram. It is extremely beneficial to be able to verify that the state diagram is correct and has full coverage, as well as extract each path through it exhaustively - which is exactly what the traces generated from MP are able to provide to all users. The finite state machine project resulted in an MP model automatically generating the finite state machine model shown in Figure 5.

4. FINDINGS AND RECOMMENDATIONS

One of the more useful features of the Monterey Phoenix modeling environment is that the environment enables a human expert in a given domain to define the lexicon of his/her domain and then use that lexicon to declare the rules which enable/constrict the allowable events in the domain. Humans are very good at declaring rules for a given domain but very poor at clearly understanding the potential emergent behaviors which might arise from exhaustively applying those rules in different circumstances.[13]

A strength of the Monterey Phoenix environment is that it exhaustively generates all possible event traces of applying a given set of rules up to a given scope of repeatedly executing the set of rules. Monterey Phoenix can generate the various use cases of interest for system engineers in developing system architectures for complex systems and can generate the finite state machines needed by engineers to implement the interfaces necessary to integrate logical system components with physical system components.

For the three MDO-HuRT-S' MP modelling efforts, each summer intern team was tasked to investigate MP capabilities for modeling various aspects of controlling human-robot team behaviors as summarized in this finite state machine paper and the two related papers in this conference which discuss MOOS-IvP Integration and SAR Expansion. These modeling efforts will continue with the expectation of being repeated as the capture-the-flag game is expanded to approximate complex, contested environments of interest to improve safety and security

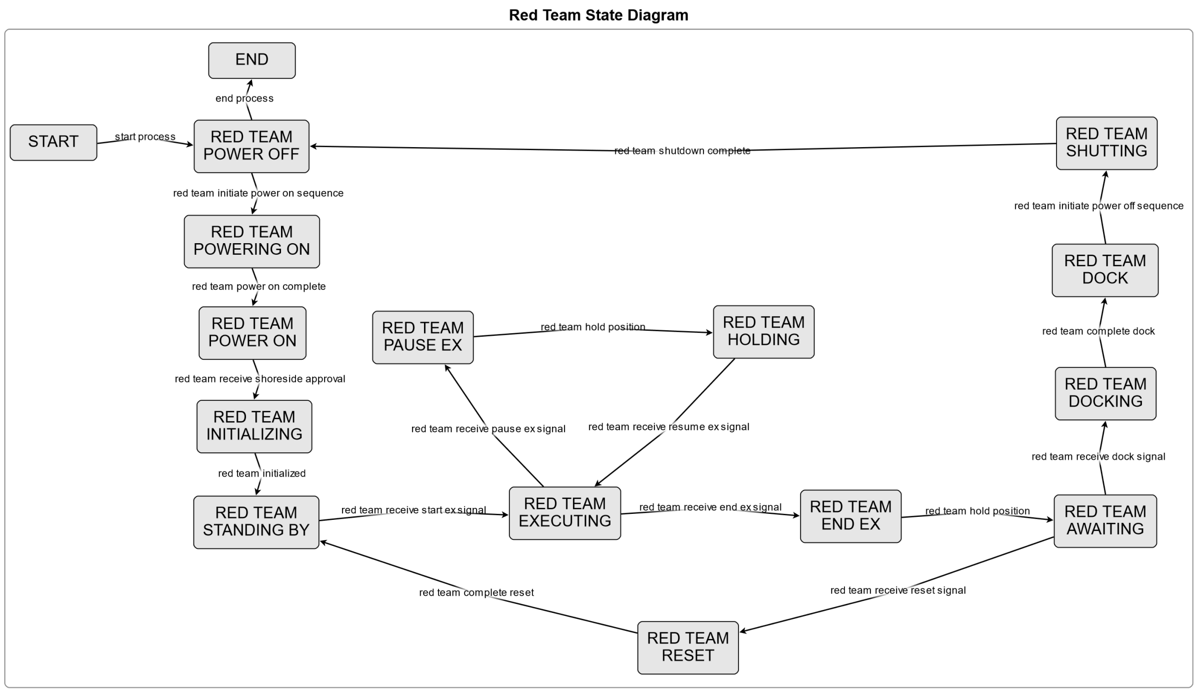


Figure 5. The finite state machine for a single agent generated from the MP model, capturing the states of a Red Team robot powering on, executing STARTEX, executing the game (departing from home station, playing the CTF game, returning to home station), and executing ENDEX.

of human-robot teams meeting human intent for assigned tasks and missions. However, the following sections describe the benefits we realized during this first iteration.

4.1 Improved understanding of the Aquaticus CTF game

- Helped understand of the process and desired states for the single agent in the Aquaticus competition.
- Assisted in identifying how the behaviors connect with each other and where the process could possibly go wrong. For example, emergent behaviors of controlling human-robot teaming tasks can be viewed as a SOS challenge for understanding/controlling CPS.

4.2 Assists in planning expansions

- Helps with planning what capabilities that could be added on.
- By creating a state diagram, it forces the designer to specifically plan out the actions/tasks a single robot playing the Aquaticus CTF game.
- Acts as a useful planning tool.

4.3 Different views highlighted

- The model also captures how the process could be iterative, such as playing the game multiple times if the robots had enough battery, which would aid in faster data acquisition.

5. CONCLUSION AND FUTURE WORK

This work specifically demonstrated that an expert in the field of autonomous robots can create a model of single robot tasks in the form of a finite state machine and then exhaustively verify and validate that model using the Monterey Phoenix tool. While the model implemented in MP for this work was minimal in terms of game play strategy, future efforts will focus on creating a hierarchy of FSMs within MP to adequately reflect game play strategy for one agent. Additional work will then focus on creating the FSMs of opposing agents such that MP can be used to exhaustively trace multi-agent competitions for Aquaticus CTF. Such efforts would reveal emergent behaviors as cooperative and competing agents interact.

The MDO-HuRT-S team is coordinating with other institutions to improve safety and security of state estimation and control of MDO-HuRT-S human-robot teaming experiments by applying recent technical results in trusting state estimation and control results of four microgrids distributed across four states[14] :

- Use of hashgraph distributed ledger technology (DLT) to trust asynchronous current and voltage values shared at 60 HZ among microgrid peers in a peer-to-peer network of microgrids in different states ,[15] and
- Use of the Vanderbilt resilient centerpoint vector consensus algorithm to construct the best continuous parameter estimates of current and voltage values from the asynchronous measured values shared using DLT and resulting in local trusted copies of global data [16] .

There are also activities during states, such as the Red team playing the CTF game in the "Red Team Executing" state of Figure 5. Future work will include creating hierarchical finite state machines (FSMs) to follow the rules to execute red team playing behaviors for playing the CTF game and also for responding to safety violations which may occur during the game. The FSM model event traces/behaviors generated by the MP model can be incrementally improved by using multiple event traces to model different systems states, such as capturing a computable model of the blue team event traces/behaviors, shoreside event streams/behaviors, and environmental behaviors. Additionally, we can expand specific SAR behaviors such as those required for sharing some information and hiding other information as the minimal data needed for cooperative execution of SAR operations are shared across national and international security boundaries to resolve a MAN OVERBOARD safety violation (i.e. share object identification and object position, velocity, acceleration and jerk vector parameters across national and international security boundaries).

In the future, the model can be expanded to simulate actual robot behaviors that occur during the game of Aquaticus in comparison to the very high-level view of the game portrayed by the state machine created in this paper. Instead of simply "executing the mission", a system architect can have different offensive and defensive strategies that can be selected based on conditions in the field. This allows one to see the interplay of those behaviors. Baker, et.al[3] created a method to convert an MP file to .moos and .bhv files. Therefore, a user can create and plan strategies in Monterey Phoenix and that MP file can then be translated into actual behaviors that the USV will execute. This allows the human planner to dictate desired behavior to the robot. A competent programmer can create a file for each behavior and the human commander can select from a set of pre-written behaviors, that will be triggered based on field conditions. Those field conditions can be set by the commander, who will only need to know the relatively simple MP language, instead of python, C++, and MOOS. Further, Monterey Phoenix can provide a view of how those behaviors will interplay over multiple iterations and game conditions, which is not available in the suite of other languages.

ACKNOWLEDGMENTS

The work reported here was partially supported by the National Security Agency (NSA). The views expressed herein are those of the authors and do not purport to reflect the position of the United States Military Academy, the Department of the Army, the Naval Postgraduate School, the National Security Agency or the Department of Defense.

REFERENCES

[1] Novitzky, M., Robinette, P., Benjamin, M. R., Fitzgerald, C., and Schmidt, H., "Aquaticus: publicly available datasets from a marine human-robot teaming testbed," in [*2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*], 392–400, IEEE (2019).

[2] MIT, "Project Aquaticus Home Page." On line https://oceanai.mit.edu/aquaticus/pmwiki/pmwiki.php?n=Main.HomePage. (Accessed: 1 April 2023).

[3] Baker, L., Sen, S., Novitzky, M., Giammarco, K., James, J., Semmens, R., Collins, M., and Harshbarger, S., "Creating an interface between behavior-modeling software and a robotic simulation environment." Conference paper, SPIE Defense + Commercial Sensing (30 April - 4 May 2023).

[4] Sagos, M., Giammarco, K., Dyer, P., Novitzky, M., James, J., Semmens, R., Collins, M., and Harshbarger, S., "Behavior analysis of search and rescue operations employing human-machine teaming." Conference paper, SPIE Defense + Commercial Sensing (30 April - 4 May 2023).

[5] NPS, "Monterey Phoenix Home." On line https://nps.edu/mp. (Accessed: 1 April 2023).

[6] Auguston, M., "Monterey Phoenix, or how to make software architecture executable," in [*Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*], 1031–1038, ACM, Orlando, FL (2009).

[7] Collins, M., "Emergent behavior analysis questions." In email correspondence between J. James, M. Collins and others, dated Dec. 7, 2022, unpublished.

[8] Giammarco, K., "Exposing and controlling emergent behaviors using models with human reasoning," in [*Emergent Behavior in System of Systems Engineering*], 23–61, CRC Press (2022).

[9] Naval Postgraduate School, "MP-Firebird Tool." Accessed Apr. 10, 2023 [Online].

[10] Naval Postgraduate School, "MP-Gryphon Tool." Accessed Apr. 10, 2023 [Online].

[11] Errico, T., "Human-robot teaming: State machine." MPVIP Cohort 4 Showcase Presentation (Summer 2022).

[12] Errico, T., *Human-Robot Teaming: State Machine*. West Point, NY, USA (Summer 2022 [Online].).

[13] Tonn, B. E. and Stiefel, D., "Anticipating the unanticipated-unintended consequences of scientific and technological purposive actions," *World Futures Review* **11**(1), 19–50 (2019).

[14] Koutsoukos, X., Eisele, S., Wutka, M., Potteiger, N., and Collins, M., "Pre-curser for fully distributed control of powergrids." Report (February 6 2023).

[15] Hedera, "Open Source at Hedera." On line https://hedera.com/learning/hedera-hashgraph/open-source-at-hedera. (Accessed: 1 March 2023).

[16] Abbas, W., Shabbir, M., Li, J., and Koutsoukos, X., "Resilient distributed vector consensus using centerpoint." Science Direct https://www.sciencedirect.com/science/article/pii/S0005109821005744. (Accessed: 1 March 2023).