

VERIFICATION OF NASA EMERGENT SYSTEMS

Christopher Rouff¹, Amy KCSVanderbilt¹, Walt Truskowski² James Rash²
and Mike Hinchey²
SAIC¹; NASA Goddard Space Flight Center²

Abstract: *NASA is studying advanced technologies for a future robotic exploration mission to the asteroid belt. This mission, the prospective ANTS (Autonomous Nano Technology Swarm) mission, will comprise of 1,000 autonomous robotic agents designed to cooperate in asteroid exploration. The emergent properties of swarm type missions make them powerful, but at the same time are more difficult to design and assure that the proper behaviors will emerge. We are currently investigating formal methods and techniques for verification and validation of future swarm-based missions. The advantage of using formal methods is their ability to mathematically assure the behavior of a swarm, emergent or otherwise. The ANT mission is being used as an example and case study for swarm-based missions for which to experiment and test current formal methods with intelligent swarms. Using the ANTS mission, we have evaluated multiple formal methods to determine their effectiveness in modeling and assuring swarm behavior.*

Key words: emergent behavior; formal methods; spacecraft; swarms; verification

1. INTRODUCTION

NASA is studying advanced technologies for a future robotic exploration mission to the asteroid belt. One mission, the prospective ANTS (Autonomous Nano Technology Swarm) mission, will comprise 1,000 autonomous robotic agents designed to cooperate in asteroid exploration. Since the ANTS and other similar missions are going to consist of autonomous spacecraft which may be out of contact with the earth for

extended periods of time, and have low bandwidths due to weight constraints, it will be difficult to observe improper behavior and to correct any errors after launch. Because of this proper verification of these kinds of missions is extremely important. One of the highest possible levels of assurance comes from formal methods¹. Once written, a formal specification can be used to prove properties of a system (e.g., the underlying system will go from one state to another or not into a specific state) and check for particular types of errors (e.g. race conditions). The authors have investigated a collection of formal methods techniques for verification and validation of spacecraft using swarm technology. Multiple formal methods were evaluated to determine their effectiveness in modeling and assuring the behavior of swarms of spacecraft^{2, 3}. The ANTS mission was used as an example of swarm intelligence for which to apply the formal methods.

The ANTS mission^{4, 5} will have swarms of autonomous pico-class (approximately 1kg) spacecraft that will search the asteroid belt for asteroids that have specific characteristics (Figure 1). To implement this mission a high degree of autonomy is being planned, approaching total autonomy. A heuristic approach is being considered that uses a social structure to the spacecraft in the swarm. Artificial intelligence technologies such as genetic algorithms, neural nets, fuzzy logic and on-board planners are being investigated to assist the mission to maintain a high level of autonomy. Crucial to the mission will be the ability to modify its operations autonomously to reflect the changing nature of the mission and the distance and low bandwidth communications back to Earth.

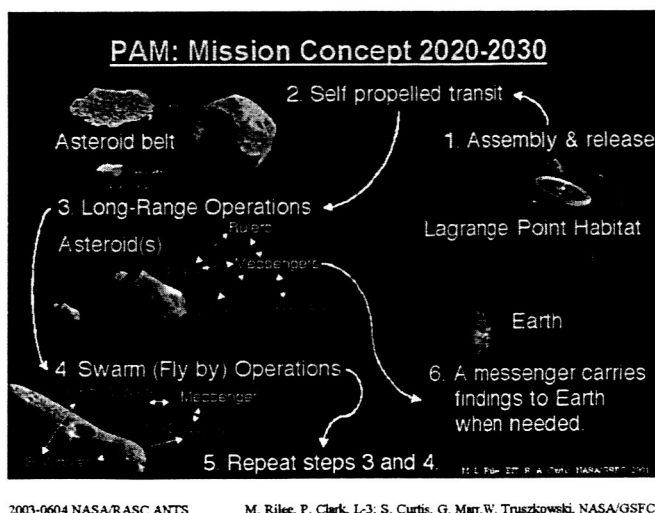


Figure 1. ANTS Mission concept.

Approximately eighty percent of spacecraft, called workers, will have a single specialized instrument (e.g., a magnetometer, x-ray, visible/IR, neutral mass spectrometer). Other spacecraft are called rulers that have rules that decided the types of asteroids and data the mission is interested in and will coordinate the efforts of the workers. Messengers will coordinate communications between the workers, rulers and Earth. Each worker spacecraft will examine asteroids they encounter and send messages back to a ruler that will then evaluate the data and form a team to investigate it that contains the appropriate spacecraft with specialized instruments.

One of the most challenging aspects of using swarms is how to verify that the emergent behavior of such systems will be proper and that no undesirable behaviors will occur. In addition to emergent behavior in swarms, there are also a large number of concurrent interactions between the agents that make up the swarms. These interactions can also contain errors, such as race conditions, that are difficult to detect until they occur. Once they do occur, it can be difficult to recreate the errors since they are usually data and time dependent. Verifying intelligent swarms are even more difficult since the swarms are no longer made up of homogeneous members with limited intelligence and communications. Verification will be difficult not only due to the complexity of each member, but also due to the complex interaction of a large number of intelligent elements.

2. FORMAL APPROACHES AND ASSURANCE

As mission software becomes increasingly more complex, it also becomes more difficult to test and find errors. Race conditions in these systems can rarely be found by inputting sample data and checking if the results are correct. These types of errors are time-based and only occur when processes send or receive data at particular times, in a particular sequence or after learning occurs. To find these errors, the software processes involved have to be executed in all possible combinations of states (state space) that the processes could collectively be in. Because the state space is exponential (and sometimes factorial) to the number of states, it becomes untestable with a relatively small number of processes. Traditionally, to get around the state explosion problem, testers have artificially reduced the number of states of the system and approximated the underlying software using models.

Formal methods are proven approaches for assuring the correct operation of complex interacting systems^{6, 7}. Verifying emergent behavior is an area that most formal methods have not addressed. We surveyed a number of formal methods techniques to determine if there existed formal methods that

have been used or would be suitable for verifying swarm-based systems and their emergent behavior^{8, 9}. Formal methods were surveyed based on whether they had concurrency support, were based on a formal model, had tool support, and had been used to specify and verify agent-based or swarm-based systems. What was found from the survey was that there are a number of formal methods that support either the specification of concurrency or algorithms. It was also found that in recent years there have been a large number of hybrid or combination formal methods that have been developed with the hope of specifying both concurrency and algorithms with the same method. Table 1 shows part of the results of the survey for mainstream formal methods, Table 2 shows the results for hybrid formal methods and Table 3 shows a comparison of formal methods that have been used to specify swarm-based systems.

Table 1 summarizes the results of mainstream formal techniques and their use on swarm and agent-based systems. The formal methods were evaluated for concurrency support, algorithm support, tool support, their formal basis, whether they had been used in specifying agent-based systems and whether they had been used in specifying swarm-based systems.

Table 1. Comparison of candidate formal methods for intelligent swarms.

Name	Con- currency Support	Algorithm Support	Tool Support	Formal Basis	Used in Agent- Based Specs.	Used in Swarm- Based Specs.
Artificial Physics	Yes	Yes	Yes	Yes Mathem.	Yes	Yes - limited
B	No	Yes	Yes	Yes Set Theory/ Pred. Log.	Yes	No
BDI	Yes	No	Yes	Yes Logic	Yes	Yes - limited
Logic	Yes	No	Yes	Yes Algebraic	Yes	No
CSP	Yes	No	Yes	Yes Form. Lang	Yes	No
Finite State Machines	No	Yes	Yes	Yes Form. Lang	Yes	No
Game Theory	Yes	No	Yes	Yes Mathem.	Yes	Yes
I/O	Yes	Yes	Yes	Yes Form. Lang	Yes	No
Automata	Yes	No	Yes - limited	Yes Logic	Yes	No
KARO	Yes	No	Yes - limited	Yes Logic	Yes	No

Name	Con- currency Support	Algorithm Support	Tool Support	Formal Basis	Used in Agent- Based Specs.	Used in Swarm- Based Specs.
Mathema tical Analysis	Yes	No	Yes	Yes Mathem.	Yes	Yes
Petri Nets	Yes	No	Yes	Yes	Yes	No
Pi Calculus	Yes	No	Yes	Yes Algebraic	Yes	No
Real Time Logic	Yes	No	Yes	Yes Logic	No	No
SCR	No	Yes	Yes	Yes Form. Lang	No	No
Statechar ts	Yes	No	Yes	No Form. Lang	Yes	No
UML	Yes	Yes	Yes	No	Yes	No
X- Machines	No	Yes	No - limited	Yes Form. Lang	Yes	No
Z	No	Yes	Yes	Yes Set Theory/ Pred. Calc.	Yes	No

Table 2 compares hybrid or combination formal methods surveyed. This table also lists support for concurrency, algorithms, tool support, whether it is based on a formal foundation, has been used to specify agent-based systems and if it has been used to specify swarm-based systems. For the tool support, a yes is entered only if there was integrated tool support for the combined languages.

Table 2. Comparison of hybrid formal methods.

Name	Con- currency Support	Algorithm Support	Tool Support	Formal Basis	Used in Agent- Based Specs.	Used in Swarm- Based Specs.
Commun. X- Machines	Yes	Yes	No	Yes	Yes	Yes
CSP-OZ	Yes	Yes	No	Yes	Yes	No
Object-Z and	Yes	Yes	No	Yes	Yes	No

Name	Con- currency Support	Algorithm Support	Tool Support	Formal Basis	Used in Agent- Based Specs.	Used in Swarm- Based Specs.
Statecharts						
Temporal B	Yes	Yes	No	Yes	Yes	No
Temporal Petri Nets	Yes	No	No	Yes	Yes	No
Timed Comm. Object Z	Yes	Yes	No	Yes	Yes	No
Timed CSP	Yes	No	Yes	Yes	Yes	No
ZCCS	Yes	Yes	No	Yes	Yes	No

Table 3 compares methods that have been used for modeling or specifying swarm-based systems (computer or biological based). It lists whether each method provides support for concurrency, algorithms, has tool support, is based on a formal foundation, and if it supports the analysis of emergent behavior and whether it has been used to specify swarm-based systems (software or biological).

The following is a summary of specification techniques that have been used for specifying social, swarm and emergent behavior:

- Weighted Synchronous Calculus of Communicating Systems (WSCCS), a process algebra, was used by Tofts to model social insects¹⁰. WSCCS was also used in conjunction with a dynamical systems approach for analyzing the non-linear aspects of social insects¹¹.
- X-Machines¹² have been used to model cell biology¹³ and modifications, such as Communicating Stream X-Machines¹⁴, also have potential for specifying swarms.
- Dynamic Emergent System Modeling Language (DESML)¹⁵, which is a variant of UML, has been suggested for modeling emergent systems.
- Cellular automaton¹⁶ has been used to model systems that exhibit emergent behavior (such as land use).
- Artificial Physics¹⁷ is based on using properties from physics to model constraints and interaction between agents.
- Simulation approaches that use a modeling technique to model the behavior. These approaches do not model emergent behavior beforehand, only after the fact.

Though there were a few formal methods that have been used to specify swarm-based systems, only two had been used to analyze the emergent

behavior of swarms. One of these formal methods was WSCCS and the other was artificial physics. In addition, it was also discovered that the majority of the work in specifying swarm-based systems has been done on biological systems by biologist with the help of computer scientists that used modified formal methods^{10, 11, 13}.

Table 3. Comparison of formal methods used for swarm specifications.

Name	Con- currency Support	Algorithm Support	Tool Support	Formal Basis	Emergent Behavior Anal.	Used in Swarm- Based Specs.
Cellular Automaton	Yes	Yes	Yes	Yes (FSM)	No	Yes
Com. X- Machines	Yes	Yes	No	Yes (Formal Lang.)	No	Yes
Unity Logic	Yes	No	Yes (limited)	Yes (Logic)	No	Yes
WSCCS	Yes	No	Some (Prob. Workbench)	Yes (Process Alg.)	Yes (Markov Chain)	Yes

3. EVALUATION OF SPECIFICATION METHODS

Based on the results of the survey, four formal methods were selected to do sample specification of part of the ANTS mission. These methods were: the process algebras CSP¹⁸ and WSCCS¹⁰, X-Machines¹², and Unity Logic¹⁹. DESML, Cellular Automata, artificial physics and simulation approaches were not used even though they had been used for specifying or analyzing emergent behavior. DESML was not selected because it had not been used to analyze emergent behavior. Cellular Automata was not selected because it did not have any built in analysis properties for emergent behavior and because it has been primarily used for simulating emergent systems. Though not used for the specification, it too may be revisited to examine its strengths. Artificial physics, though again has possibilities, was not used due to its early stages of development and use. Lastly, simulation techniques were not used because verification can not be done only using simulation. This is because there could be emergent or other undesirable behaviors occurring that are not visible or come out in a simulation, but may be there none the less. A formal technique is designed to find exactly these kinds of errors.

The following describes the results of the sample specifications and the evaluation of the methods used.

CSP is very good at specifying the process protocols between and within the spacecraft and analyzing the result for race conditions. Being able to evaluate a system for race conditions is very important in systems, particularly swarm-based systems which are highly parallel. From a CSP specification, reasoning about the specification can be done to determine race conditions as well as converted into a model checking language for running on a model checker. Though the above is important and process algebras have been widely used for specifying agent-based systems, there is no facility for evaluating emergent behavior of the end system.

WSCCS is a process algebra that takes into account the priorities and probabilities of actions performed. It further provides a syntax and large set of rules for predicting and specifying the choices and behaviors of the Leader, as well as a congruence and syntax for determining if two automata are equivalent. All of this in hand, WSCCS can be used to specify the ANTS spacecraft and to reason about and even predict the behavior of one or more spacecraft. This robustness affords WSCCS the greatest potential for specifying emergent behavior in the ANTS swarm. What it lacks is an ability to track the goals and model of the ANTS mission in a memory.

Unity Logic has a syntax equivalent to simple Propositional Logic for reasoning about predicates and the states they imply as well as for defining specific mathematical, statistical and other simple calculations to be performed. However, it does not appear to be rich enough to allow ease of specification and validation of more abstract concepts such as mission goals. This same simplicity, however, may make it a good tool for specifying and validating the actual Reasoning portion of the ANTS Leader spacecraft, when the need arises. In short, specifying emergent behavior in the ANTS swarm will not be accomplished well using Unity Logic, though logic does provide many useful properties for reasoning about systems.

X-Machines provide a highly executable environment for specifying the ANTS spacecraft. It allows for a memory to be kept and it allows for transitions between states to be seen as functions involving inputs and outputs. This allows us to track the actions of the ANTS spacecraft as well as write to memory any aspect of the goals and model. This ability makes X-Machines highly effective for tracking and affecting changes in the goals and model. However, X-Machines do not provide any robust means for reasoning about or predicting behaviors of one or more spacecraft, beyond standard propositional logic. This will make specifying or analyzing emergent behavior difficult or impossible.

Based on the above evaluation, the following are some of the properties of a formal method needed for specifying swarm-based systems:

- Ability to model and reason about aggregate behavior based on future actions of the individual agents of a swarm (such as provided by WSCCS)
- Ability to model and reason about concurrent processes for detection of race conditions (such as provided by CSP and Unity Logic)
- Ability to model states of an agent of the swarm to assure correctness (such as provided by statecharts, X-Machines or Z)
- Ability to model and reason about persistent information so adaptive behavior can be verified (such as provided by X-Machines).

A blending of the above methods seems to be the best approach for specifying swarm-based systems and analyzing emergent behavior of these systems. Blending the memory and transition function aspects of X-Machines with the priority and probability aspects of WSCCS may produce a specification method that will allow all the necessary aspects for specifying emergent behavior in the ANTS mission and other swarm-based systems. The idea of merging the above methods is currently being furthered studied as well as adding some of the properties of logic and cellular automata.

4. CONCLUSION

Swarm-based missions are becoming more important to NASA and other government missions so new science can be performed. These types of missions have many positive attributes but represent a change in paradigm from current types of single spacecraft missions. Due to this, swarms require new types of verification and validation techniques to assure their correct operation. To overcome their nondeterministic nature, high degree of parallelism, intelligent behavior and emergent behavior, new kinds of verification methods need to be used.

This paper gave the results of an investigation into formal method techniques that might be applicable to future swarm-based missions and that can verify their correctness. It also analyzed the properties of these methods to determine the needed attributes of a formal specification language to predict and verify emergent behavior of future NASA swarm-based systems.

We are currently working on developing a new formal method based on blending aspects of the above formal method as well as adding additional mathematical techniques from other areas of mathematics that might prove fruitful for predicting the emergent behavior of swarms. From this new formal method we will use the ANTS and another NASA swarm-based mission to test the capabilities of the resulting formal method. We expect

that the resulting formal method could become the basis of other specification languages to support specification and analysis of future swarm-based systems.

References

1. E. Clare and J. Wing. Formal Methods: State of the Art and Future Directions. Report by the Working Group on Formal Methods for the ACM Workshop on Strategic Directions in Computing Research, ACM Computing Surveys, vol. 28, no. 4, Dec. 1996, pp. 626-643.
2. E.G. Bonabeau, G. Theraulaz, et al. Self-organization in Social Insects, Trends in Ecology and Evolution, 1997, vol. 12, pp. 188-193.
3. G. Beni. and J. Want. Swarm Intelligence. In Proceedings of the Seventh Annual Meeting of the Robotics Society of Japan, pp 425-428, Tokyo, Japan, 1989, RSJ Press.
4. ANTS team. Protocol for ANTS Encounters. NASA GSFC, Code 695.
5. S.A. Curtis, J. Mica, J. Nuth, G. Marr, M. Rilee, and M. Bhat. ANTS (Autonomous Nano-Technology Swarm): An Artificial Intelligence Approach to Asteroid Belt Resource Exploration. International Astronautical Federation, 51st Congress, October 2000.
6. P. Nayak, P. Pandurang, et. al. 1999. Validating the DS1 Remote Agent Experiment. In Proceedings of the 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS-99).
7. C. Rouff, J. Rash, M. Hinchey. Experience Using Formal Methods for Specifying a Multi-Agent System. Sixth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2000) September 11-15, 2000.
8. C. Rouff, W. Truszkowski, J. Rash and M. Hinchey. Formal Approaches to Intelligent Swarms. IEEE/NASA Software Engineering Workshop, Greenbelt, MD. December 2003.
9. C. Rouff, A. Vanderbilt, W. Truszkowski, J. Rash and M. Hinchey. Verification of NASA Emergent Systems. Ninth International Conference on Engineering of Complex Computer Systems (ICECCS 2004), Florence, Italy, April 14-16, 2004.
10. C. Tofts. Describing social insect behaviour using process algebra. Transactions on Social Computing Simulation. 1991. 227-283.
11. D.J.T. Sumpter, G.B. Blanchard and D.S. Broomhead. Ants and Agents: A Process Algebra Approach to Modelling Ant Colony Behaviour. Bulletin of Mathematical Biology. 2001, 63, 951-980.
12. S. Eilenberg. Automata, Languages and Machines, Vol. A. Academic Press, 1974.
13. M. Holcombe. Mathematical models of cell biochemistry. Technical Report CS-86-4. 1986. Dept of Computer Science, Sheffield University, United Kingdom.
14. J. Barnard, J. Whitworth and M. Woodward. Communicating X-machines. Journal of Information and Software Technology, 38(6), 1996.
15. J.R. Kiniry. The Specification of Dynamic Distributed Component Systems. Masters' Thesis, CS-TR-98-08 1998 California Institute of Technology, Comp. Sci. Depart.
16. J. von Neumann. Theory of Self-Reproducing Automata. University of Illinois Press, Urbana, Illinois. 1996. Edited and completed by Q.W. Burks.
17. W. Spears and D. Gordon. (1999) Using artificial physics to control agents. Proceedings of the IEEE Conference on Information, Intelligence, and Systems (ICIIS'99).
18. C.A.R. Hoare. Communicating Sequential Processes. Communications of the ACM, 21(8):666-677, August, 1978.
19. K. Chandy and J. Misra. Parallel Program Design: A Foundation. Addison-Wesley. 1988.