

Autonomous Self Organized UAV Swarm Systems

Dustin J. Nowak and Gary B. Lamont

Dept. of Electrical and Computer Engineering, Graduate School of Engineering and Management,

Air Force Institute of Technology

WPAFB, Dayton, Ohio, USA

(dustin.nowak@afit.edu, gary.lamont@afit.edu, gilbert.peterson@afit.edu)

Abstract—Interest in Self-Organized (SO) unmanned aerial vehicles (UAVs) systems is increasing because of their flexibility, versatility and economics. Many countries and industries are developing autonomous and swarming UAVs for reconnaissance, surveillance, intelligence gathering, and target engagement and neutralization. The processes for effectively developing these systems are still in their infancy. Currently, little effort is focused on building simple agent rules with low-level SO systems communication in order to facilitate emergent behaviors. Note that only with the use of effective control structures can the full potential of these systems realized. Presented is an innovative new paradigm for developing SO-based autonomous vehicles. Using a formal design model, the Interactive Partially Observable Markov Decision Process, a full understanding of this SO domain is possible. With this design model and a focused effort on the minimization of computational and informational complexity, emergent entangled control hierarchies allow the SO rules to operate efficiently and effectively. This work extends the formal model decomposition technique, and in doing so ties in the information theoretic optimization to develop emergent structures. Preliminary computational results reflect limited success.

I. INTRODUCTION

State of the art autonomous control systems use self-organized (SO) based techniques [1]. These systems are “popping up” in a large spectrum of industrial applications, from network and data storage, to medical systems, to engineering applications, and of course the military. Water Vehicles, medical agents, and ‘Cybercraft’ security agents. Although arguments exist over the particulars of SO! (SO), shown in figure 1, fundamentally these type autonomous systems use “simple” agents synergistically with localized knowledge, neighborhood effect operators and stigmergy, to develop complex behaviors or structures. In the natural/bio-inspired world examples such as mold, ants, and birds inspire techniques applied in a myriad of engineering disciplines [2].

Currently, developing these systems focuses on simple rules and interaction sets. However, effective engineering SO systems requires the establishment of a process and toolbox set that look at the system as a whole. However, this toolbox should include three key pieces:

- Proper decomposition of the problem domain into implementable pieces.
- Exacting approach to development of simple low-level operators. (Biologically and Naturally Inspired)
- A construct in which the operators cannot only develop external behaviors but also allows internal control structures to emerge.

This effort outlines the initial steps for thrusting SO development into common practice and the establishes the processes in which development of these types of SO systems is possible.

This paper provides a structured approach to organizing a SO development technique that can be crossed utilized in multiple disciplines. First, in Section II, the background of human investigation into nature’s SO system forms the foundation of what makes a system a SO system. This includes discussions on emergent behaviors and structure and informational complexity. In Section III, a new twist to an familiar decomposition approach defines the way that the development of SO solutions take shape. Third, in Section IV, a discussion on facilitating *emergent* control structures is presented. A small scope implementation, in Section V, with UAV swarms, using these techniques gives an example of how these techniques are structured. Finally a vector for the future of SO engineering is presented.

Category	Cazamine	Heylighen	Cottam
Level of Origin	Agent	Any	Component
Level of Organization	Global	System	System
Environment Interaction	Stimulus	Perturbation	Very little
Operators	Component Interaction	Subsystem Attractors	Inter-level Transformation
Result	Global Patterns	Reduced Entropy	Unified System

Fig. 1. SO definitions [3] [2] [4]

II. OTHER WORKS

Initially, a discussion of SO literature is presented. It provides the background on autonomous control, Emergent Structure, Information Complexity and a swarm problem domain math model called Interactive Partially Observable Markov Decision Process (I-POMDP). The desired goal is the articulation of an SO system development process. All of these pieces set the stage for designing an SO system in an efficient and effective manner.

There are three foundational works on SO: Camazine’s book on Biological SO systems [2], Bonabeau’s book on Swarm Intelligence [5] and Heylighen’s work on Cybernetics [4], [6]. Camazine’s work provides a survey of biological investigations in SO developed operators and the describes them using mathematical models. (Some of these and other operators

are described in 10.) Bonabeau dealt more heavily with the integration of the benefits of the natural systems, especially swarms, into computational implementations. Heylighen sets out with a different goal in mind. Using the knowledge gained through cybernetics, he describes the way the world developed and works through SO principles. All of these works outline the ideas of Self Organization and an attempt to the them together in order to explain the structure and order that exists in the world.

Through these works several levels of SO emergent systems have been described. Although naturally occurring SO system are not constrained to these categories, the chart in Figure 2 helps in outlining the variability and capability of SO systems. Studies of Self Organization show natural examples happen internal to single species, throughout an entire population, and subsets there in.

Internal	Sub System	Global
Intra-Agent	Inter-Agent	Inter-Agent
Visual Pattern (Geometric)	Immediate Response	Structures and Repeatable Behaviors
Temporary	Temporary	Long Term
Single Emergent Property	Single Emergent Property	Several Emergent Properties
Homogeneous	Homogeneous	Heterogeneous
Animal Coat Patterns	Ant Bridges	Termite Mounds

Fig. 2. Levels of SO classification

A. Autonomous control

Translating Inspired SO operators into a usable system requires an understanding of the controls world. Figure 3 shows the commonly accepted levels of automation. Although SO techniques could be applied in almost any system a true SO implementation focuses on the 10th level of this hierarchy, complete autonomous control.

Automation Level	Automation Description
1	The computer offers no assistance: human must take all decision and actions.
2	The computer offers a complete set of decision/action alternatives, or
3	narrowes the selection down to a few, or
4	suggests one alternative, and
5	executes that suggestion if the human approves, or
6	allows the human a restricted time to veto before automatic execution, or
7	executes automatically, then necessarily informs humans, and
8	informs the human only if asked, or
9	informs the human only if it, the computer, decides to.
10	The computer decides everything and acts autonomously, ignoring the human.

Fig. 3. Sheridan & Verplank (1978) Levels of Automation [7]

Braitenburg, a psychologist, produced any interesting research into this type of control. In [8] utilizing only simple sensors and actuators he produced reactive automaton. The classification hierarchy purposed in his work focused primarily

on the 10th layer, but broke into to complexity of tasks accomplished.

In [9], Brooks took control from a linear process to a hierachal process. This allowed the system to work multiple levels of action at once. From this new style many TLAs spawned. The most notable of which is Gat's TLA [10], shown in Figure 4, but also include [11]–[13].

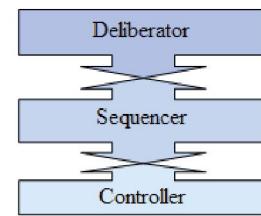


Fig. 4. Gat's Three Layer Architecture [10]

From this framework [14] defined a three layer architecture for SO. Through further research this was shown to constrain the capabilities of the SO operators and emergent structure.

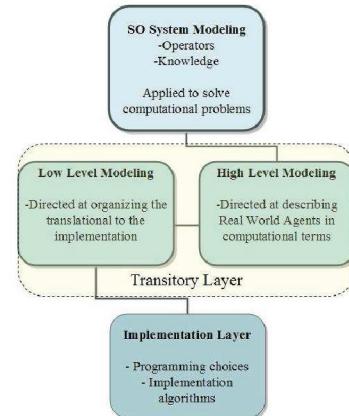


Fig. 5. Levels of SO Systems [14]

B. Emergent Structures

Synergy is the defined by the coalescing of effects in a system to create a greater result, structure, behavior or effect. Haken defined Synergetics in the early sixties. In [15] pulled these ideas into the movement, structure and control of information. Holland furthered the argument through [16], noting that complexity and structure of system effectively develop from within.

In SO systems emergence happens throughout the system. Most commonly thought of is the emergence of behaviors external to a system of agents, bird swarms, and ants trails. Emergent structures external to the system are also well-known, such as termite mounds and bee hives. Less commonly thought of are structures internal to the system or agents, such as SO agent hierarchies, Zebra strips and slime molds. This third aspect provides a leaping off point for internal control structures.

If the agent can be broken into independent sub-sections, each of those sub sections can be treated as an agent itself. Application of the SO principles now refocus the usually outward emergence to and inward development of structure in relation to the over arching agent. Object-oriented (OO) programming already lends itself to this effect. Different pieces are independent developed, utilized and reutilized through decoupling techniques. The emergent structure stems from the optimal interaction of the individual OO objects. This requires definition of optimal interaction which is interpreted many different ways to include through information theoretics.

C. Information complexity

Understanding complexity in system with emergent properties requires proper definition of complexity and emergence in relation to information. In [17], Prokopenko tries to untangle the complexity. Eight rules scope the complex system used for the analysis: 1) Complex systems are *open*; 2) Component *interaction*; 3) Non-Trivial interactions result in internal constraints; 4) Thereby creating more organization (*Self-Organization*); 5) The organization gives rise to *emergent* behaviors; 6) This response shows *adaptation*; 7) Temporal adaption is *evolution*; 8) And the resulting forms responses more effective than the single agent. Equation 1 shows Shannon's information theoretic. This gives the amount of information need to define a state as a relation to its probability.

$$H(X) = \sum_{x \in X} P(x) \frac{1}{\log P(x)} \quad (1)$$

From this the mutual information between two agents derives to the information need to assure the agents are in the same state, shown in Equation 2, known as information entropy.

$$I(X;Y) = H(y) - H(Y|X) \quad (2)$$

This information description boils down to mutual information = receiver's state diversity - state understanding of the receiver about the source. A SO systems is defined by information transfer I , where the information received from the outside the system is less than the change in information internal to the system, $I^{external} < \Delta I^{system}$. Define the complexity by the amount of information and its transfer, the system complexity change is also greater then the external complexity forces, $C^{external} < \Delta C^{system}$. Using this a complex or SO system defines it complexity through Equation 3.

$$C_i^{external} + C_i^{system}(t) < C_i^{system}(t+1) \quad (3)$$

Equation 7 efficiency of state prediction e is defined by total system entropy E and the complexity change C_i .

$$e = \frac{E}{C_i} \quad (4)$$

Simply put the efficiency of the system is determined by the total space over the ability to represent it accurately.

In order to optimize a SO system in terms of complexity the minimization of the information complexity in order to maintain effective state prediction must be accomplished. This, however is dependent on the model.

D. I-POMDP

A constrained problem model provides the framework for describing the state information and allowing the system to minimize the resulting computation complexity. Markov models are used for swarm systems in complex domains. The Partially Observable Markov Model is commonly used because it allows for uncertainty in the global state and in the transition between states. In [18] Doshi outlines the extension of a Partially Observable Markov Decision Process to include independent agent modeling. The POMDP provides a global model for autonomous agents, utilizable in complexity analysis, but the I-POMDP focuses on the agents and allows for optimization on the interaction information complexity. Through this the system uses beliefs about neighboring agent states and interactive probabilities to define the state transition. Equation 5 shows the basic tuple where IS_i defines the state through known state and interaction probability of the other agents.

$$I - POMDP_i = \langle IS_i, A, T_i, \Omega_i, R_i \rangle \quad (5)$$

Again, IS_i is the interactive state distribution, A is the action set, T_i is the transition set based on the agent, Ω_i is the individual agent observation set and R_i is the agent reward. The minimization of the interaction and state predictions from the I-POMDP provides the direct link to the efficiency calculation in Equation 7. The interactive state IS_i depends on the state S and the belief that other agents interact with that state, θ_j , using $IS_i = S \times \theta_j$. In this the belief state of other agents Θ_j effecting the state derives from Equation 6.

$$\Theta_j = \langle b_j, A, \Omega_j, T_j, O_j, R_j, OC_j \rangle \quad (6)$$

Most of the pieces of this belief state derive from the POMDP but focus on the agent j . The OC_j outlines the optimum criterion for the agents. This representation mirrors the actuality of the simulation and the stochastic nature of interaction and transitions. Further description of the I-POMDP and the proof of its mapping is seen in works by Doshi [18]. Through the POMDP the totality of the domain is represented and the I-POMDP pulls out the domain model of the individual agent and ties the two together.

The understanding of the state for any individual j is based on the belief b_j . This belief is a function of the observation, action and knowledge of the transition set. The information passed between the rules inside an agent work on this belief state. The assumptions of the optimum criterion are uniform over the agents helps constrain the state possibilities. However, the size is still the magnitude of the actions to the cardinality of agents $O(a^n)$. Abstraction of the belief state minimizes the information needed to hold the state.

Tying the I-POMDP to the information theoretic provides entrance point for use of an optimizer. In Equation 7, efficiency

of the information representation can be applied to the belief state b_j . The entropy in Equation 7 is defined by the amount of state information known and the complexity is applied as the amount of information need to represent that state [17]. Applying this to the I-POMDP the state information is tied to the IS_i and the amount need to represent that state is defined by the complexity of the belief state θ_j . Equation 7 shows efficiency is therefore a representation of total state over the belief state complexity. This means minimizing the belief complexity maximizes that efficiency.

$$e = \frac{IS_i}{\Theta_j} \quad (7)$$

III. SO DECOMPOSITION TECHNIQUE

In order to effectively develop SO system a new twist on a combination of the conventional development models is needed. Inclusion of the mathematical model (I-POMDP) leads to definition of the state and the overall entropy of the domain or sub-domain. Decomposition from this point provides a iterative approach to addressing each sub-domain using independent rules (or operators) for independent objectives. If each rule is treated like an agent then they too utilize Prokopenko's complexity definition to minimize interaction. Minimization of the resulting interaction complexity internal to the state of the greater agent results in a streamlined, efficient, and emergent control architecture. The coalesced system is evolved providing feed back to the lowest operates. And with change the information complexity and efficiency is effected and the structure is modified.

A. Example Decomposition

To provide basic example to illustrate the how this process works, let's look at flocking (already solved). The basic intent is to group agents together through movement. Decomposing the process there are three states inside a flock: to close, to far, collision eminent. If the agent is to close to its neighbors it moves aware. Implementation for this is a separation function based on neighbor proximity. The second aspect, to far, requires the agents to move closer to its neighbors if outside a threshold. Once again this uses an operator based on the proximity of the neighbors, but his time pulls the agent towards. Finally, collision avoidance requires that if agents are on a path to colloid action to avoid must be taken. The agents must correct their vectors to cooperate, this operator is borrowed from fish lateral lines which detect neighbor heading. The result of this action matches the two agents vectors.

Building the control structure forms from these three operators. The common state information between the separation and attraction combined them in order to minimize the information needed to determine the state. The vector match (collision avoidance) is some what independent of that because of its uses of independent state information. From this point an optimizer provides the proper threshold and strength weightings for these operators, providing feedback from testing the resultant behavior. The final product gives an swarm capability to the agent.

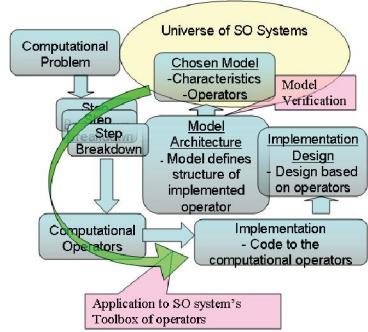


Fig. 6. 'U'-Decomposition Technique for Developing Self Organized Systems [1]

To reiterate the impetus for this approach stems from the need to allow internal structure of the agents to emerge and not force an architecture, such as the **TLA!** (TLAs) in Section II-A.

B. Emergent Entangled Control Hierarchies

As a result of this decomposition technique entangled control hierarchies are formed. Figure 7 illustrates the possibility of intertwined hierachal and operator dependencies. It is possible that an operator could join with another hierarchy of operators because the information difference is minimized in that configuration.

Emergent SO Hierarchy (Applied to UAV Swarms)

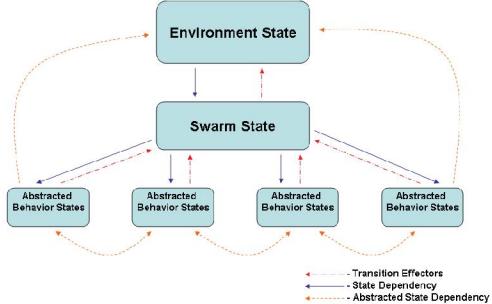


Fig. 7. Generalized Emergent Entangled Control Hierarchies [1]

IV. EXAMPLE APPLICATION, RESULTS, AND ANALYSIS

The work in [1] provides an effective example in a simple case. A UAV swarm was developed using this decomposition process. Specifically it was applied to the target engagement aspects of the autonomous swarm problem. Through the 'U'-Decomposition process the system took its shape with an emergent hierarchy shown in Figure 8. From the high level objective of effective target engagement, several steps of derivation are needed to find operators. First separating the problem into two sub-objective, getting to the target and attack target refocused the decomposition. A simple migration operator that pulled the individual agents to the target

areas through waypoints was used. Then target engagement consisted of three separate stages: local reconnaissance, target decision, and threshold based attack. In all cases the target list was known and emergent behaviors happened through the aggregated effect of each agents decision process. From this the control relations and state dependencies formed. Notice the state dependencies between the different layers of the operator tree, creating an entangled control structure.

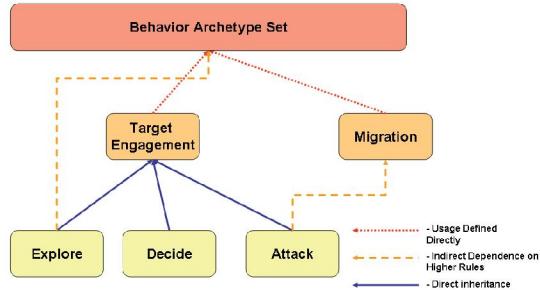


Fig. 8. Emergent Entangled Control Hierarchy for UAV swarms in target engagement scenarios. [1]

A. Experiment

To test the new attack control architecture and behavior set requires integration into the existing SO Swarm control structure originally defined in [11]. To test the new system a GA is used. Inside the GA a 6 scenario schedule allowed iterative growth of the control structures over epochs. At the end of each epoch a Pareto front defined the optimized values and an associated fitness. comparison between the old and new architectures utilizes the Pareto Fronts and full populations.

B. Results

Figure 9, shows the resulting Pareto Fronts known of the **MOEA!** (MOEA) run on the testing set. Notice in all cases the fronts of the Emergent Entangled Attack Hierarchy pushes further forward than the original control structure.

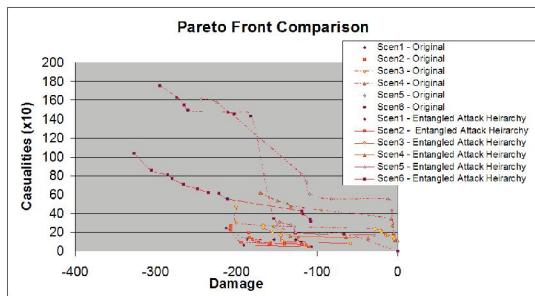


Fig. 9. Comparison of Pareto fronts. [1]

C. Analysis

Hypervolume indicators are used in MOEA comparisons where the true front is not known. The value is the area

encompassed by the curve to a given origin point, in this case 0 (no damage) and 200 (total casualties). Table I shows the hypervolume of those fronts. The difference in the control structures are emphasized in the later epochs where the new control structure out performs the old.

Scenario	Original	New
1	39871.522	37193.39
2	39334.54	35796.68
3	34729.26	31480.9
4	26172.31	28310.08
5	22354.73	26385.31
6	34132.44	47407.84

TABLE I
HYPERVOLUME VALUES FOR OLD AND NEW ATTACK CONTROL
TECHNIQUES

Another indicator used in MOEA analysis is the epsilon indicator. It shows the amount of area that would be needed to cover the other front. In this case the last epoch shows the greatest increase of dominance of the emergent control structure over the old.

Scenario	Original-New	New-Original
1	19.3	3.6
2	21.3	5
3	40	15.6
4	16.3	28.6
5	93.3	24.3
6	31.3	72

TABLE II
 ϵ -INDICATOR VALUES FOR OLD AND NEW ATTACK CONTROL TECHNIQUES

Finally the Kruskal-Wallis P-values show the independence of populations. This test is used when it is not known if the distribution is normal. Here the populations are shown to be independent in almost every case, always favoring the new control structure.

Scenario	Objective Damage	Objective Casualties
1	.1765	.2306
2	.1671	.0988
3	.001	.2628
4	.1578	0010
5	.0010	.0010
6	.0010	.0010

TABLE III
KRUSKAL-WALLIS P-VALUE FOR OLD AND NEW ATTACK CONTROL
TECHNIQUES

V. CONCLUSION & FUTURE WORK

This investigation developed a new paradigm for development of self organizing systems with military and industrial applications. The new formal decomposition technique is articulated for the system designer. It suggested that the approach is a quantifiable way to model and decompose the problem. Further the I-POMDP provides a model in which the information-theoretic framework can be "optimized." The

ideas of emergent structure based on these building blocks are outlined. Finally a small but effective implementation shows the feasibility of the approach.

Note that the referenced theories are built on firm mathematical models and sound reasoning. The interlinking of the information-theoretic analysis and optimization to I-POMDP provides a point in which the model can be optimized. Though this approach does need maturation, it serves as a beginning of a paradigm shift in autonomous swarms design.

Given this is a novel approach there are multiple areas in which further research can extend. Investigation into the decomposition technique would provide better understanding of the technique's capabilities. A full mathematical analysis and integration of the information-theoretic and I-POMDP aspects could provide mathematical proofs of optimization in the environment. Also further articulation of the math model would prove helpful, but would likely be problem specific. In order to truly prove the viability of the emergent control structure problems with domains of much larger magnitudes than those currently being addressed must be studied. Most current autonomous control does not truly test the capability and complexity of the emergent control structures. Finally, these techniques can be used in various agent applications such Cybercraft security swarm research. Some ideas are outlined in [19].

ACKNOWLEDGMENT

This effort is in support of the AFIT Advanced Navigation Technology (Ant) Center. The specific sponsors of this research are the Air Force Research Laboratory (AFRL) Information Directorate (Dr. Robert Ewing) and the Sensors Directorate - Virtual Computing Laboratory (Mike Foster).

REFERENCES

- [1] D. J. Nowak, "Exploitation of self organization in uav swarms for optimization in combat environments," Master's thesis, Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 2008.
- [2] S. Camazine, *Self Organization in Biological Systems*. USA: Princeton University Press, 2003.
- [3] R. Cottam, W. Ranson, and R. Vounckx, "Autocreative hierarchy ii: dynamics self-organization, emergence and level-changing," in *Integration of Knowledge Intensive Multi-Agent Systems, 2003. International Conference on*, 30 Sept.-4 Oct. 2003, pp. 766–773.
- [4] F. Heylighen and C. Joslyn, "Cybernetics and second-order cybernetics," *Encyclopedia of Physical Science & Technology*, vol. 3rd ed., 2001, academic Press, New York.
- [5] E. e. a. Bonabeau, *Swarm Intelligence: From Natural to Artificial Systems*. Sante Fe Institute, 1999, new York.
- [6] F. Heylighen, "Self-organization and complexity in the natural sciences," Principia Cybernetica Web, Dec 2006. [Online]. Available: <http://pespmc1.vub.ac.be/COMPNATS.HTML>
- [7] W. L. Sheridan, Thomas B. ; Verplank, "Human and computer control of undersea teleoperators," Massachusetts Institute of Technology Cambridge Man-Machine Systems Lab, Technical rept., JUL 1978, aDA057655.
- [8] V. Braatenburg, *Vehicles: Experiments in Synthetic Psychology*, E. C. T. e. a. Walker, Ed. Bradford Books, 1987.
- [9] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE JOURNAL OF ROBOTICS AND AUTOMATION*, vol. RA-2, pp. 14–23, 1986.
- [10] E. Gat, "On three-layer architectures," in *Artificial Intelligence and Mobile Robots*, R. P. B. D. Kortenkamp and e. R. Murphy, Eds. MIT/AAAI Press, 1997.
- [11] I. C. Price, "Evolving self organizing behavior for homogeneous and heterogeneous swarms of uavs and ucavs," Master's thesis, Graduate School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, March 2006.
- [12] J. K. Rosenblatt, "DAMN: A distributed architecture for mobile navigation," in *Proc. of the AAAI Spring Symp. on Lessons Learned from Implemented Software Architectures for Physical Agents*, Stanford, CA, 1997. [Online]. Available: citeseer.ist.psu.edu/article/rosenblatt97damn.html
- [13] C. W. Reynolds, "Steering behaviors for autonomous characters," in *Proceedings of the 2005 Winter Simulation Conference. San Jose, California*, 2005, pp. 763–782.
- [14] I. P. Dustin J Nowak and G. B. Lamont, "Self organized uav swarm planning optimization for search and destroy using swarmfare simulation," in *Proceedings of the 2007 Winter Simulation Conference*, M.-H. H. J. S. J. D. T. S. G. Henderson, B. Biller and e. R. R. Barton, Eds., 2007.
- [15] H. Haken, *Information and Self Organization*. Germany: Springer, 2003.
- [16] J. H. Holland, *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley, 1996, new York.
- [17] F. B. M. Prokopenko and A. J. Ryan, "An information-theoretic primer on complexity, self-organisation and emergence," *Advances in Complex Systems*, 2007.
- [18] P. Gmytrasiewicz and P. Doshi, "Interactive pomdps: Properties and preliminary results," 2004. [Online]. Available: citeseer.ist.psu.edu/gmytrasiewicz04interactive.html
- [19] D. Nowak, "Cybercraft swarm intelligence," Air Force Institute of Technology, Tech. Rep., 2008.

Abstract Model Types (AMT)							
AMT	Species / Sub-species	Operator	Parameter / Sensor	Condition / Knowledge Base	AF Uses	Rules	Remarks
Path Solver	Molds	Chemical induced movement	cAMP sensor	internal and external cAMP levels	Movement	pg 104-105 (1)	
Massing	Bark Beetle	Pheromone production, autonomous movement (larvae Density)	pheromone (chemical)	Pheromone concentration/gradient	Massing	dispersion, time, distance based pg 132 (1)	possible mass of attacks applications
Synchronization	Firefly	light, locomotion, timing mechanism	visual	pulse time constant	Communication	undefined pg 151-155 (1)	synchronization of timing sources or mass of attacks applications
Construction	Termites	retrieval & placement dirt, path follow	pheromone (chemical)	queen attributes	unknown	pg 392 (1)	
Construction	Wasp	Nest construction	touch/visual	Foundational structure, # adjacent cell walls	unknown	pg 430-432	
Dominance / Hierarchy	Wasp	Challenge	touch	Force, Rank	Structuring	460-461 (1) (Tsu)	
Foraging	Bee	Dance, Foraging	visual	dance rhythm	IR	based on number of foragers and dance length 207-208 (1)	transfer of data and resource re-supply point
Cluster control	Bee Swarm	Thermoregulation	heat	temperature range, radius, density	unknown	separation and movement inward/outward 294-297 (1)	
Classification	Bee Hive	Honeycomb fill structure	visual	oviposition, fill rates, depletion rates	Classification	pg 331 (1)	
Foraging	Ant	foraging, marking, feeding, path following	pheromone (chemical)	chemical production, pheromones interpretation	path to target	number of travelers, length, chemical deposit strength, time 229,232-234,239-241 (1)	
Offensive Mass	Ant Raiding	foraging, marking, Carrying prey, path following	pheromone (chemical)	crowding, pheromone product/interpretation	Military Mass	pg 269-274 (1)	Multiple speeds, raid size OM: 100Ks
Construction	Ant Nest	carrying, pushing, deposit	Touch (Pheromone 2nd)	structure resistance, brood size/location	Building or classifying	Pg 356-360(1)	distribution of pheromones around brood
Path Solver	Ant	locomotion, path laying, path following	pheromone (chemical)	Pheromone concentration/gradient	unknown	pg 1 (2)	Leafcutter
Classification	Ant Nest	Brood and corpse	pickup, drop	adjacent items, threshold	unk	Pg 152(2)	
Construction	Ant Weaver	chaining, weaving	touch	lava silk weaving, agent bridge construction	unknown	pg 1 (2)	
Schooling	Fish	locomotion, predator, schooling	visual, lateral line	Adjacent agent information	defense/ swarming	proximity, repulsion, matching and search; pg 180-181 (1)	Swarming, obstacle gradients
morphology	Ant	task distribution has a high plasticity	varied	agent based	UAV swarms	sub species allow for more specific task application	Multiple species (minor or minors, medium workers, sub-majors, major)

Fig. 10. Abstract Model Types - Biologically inspired low-level operators [1]