Principles for the Exploration and Construction of Reactive Swarm Systems

Karl Altenburg
Department of Accounting, Finance, and Information Systems
North Dakota State University
Fargo, ND 58102
karl.altenburg@ndsu.edu

Abstract

The study of swarms is a multidisciplinary endeavor promising a greater understanding of the complexity of nature as well as the automation of massively parallel processes. The emergence of order from the interaction of many simple agents must be understood by students of biology as well as students of computing and robotics. Students, whether building robot soccer teams, modeling animal behavior or development, or designing simulations of dynamic, fault-tolerant autonomous systems, benefit from an understanding of reactive swarm principles. Using examples of reactive swarms in robotics, biology, and agent-based computer simulations, a set of guiding principles are presented. Some of these principles include: randomness is the solution to unpredictability, minimal assumptions lead to maximum scalability, dependence on communication leads to failure, simple rules coupled with simple signals succeeds, always give yourself an out, many special agents are better than one super agent, and achieving goals through self-organization is tricky.

Introduction

Reactive swarm systems are seen as an alternative to deliberative centralized solutions. Reactive agents (i.e., robots, insects, or simulated graphical turtles) are contrasted with more deliberative agents as having a tight coupling between their sensors and actions and containing little or no internal state (Brooks 1986; Gat et al. 1994). Swarms systems are composed of many reactive autonomous agents. Work in reactive autonomous agents comes from a diversity of fields. The swarm approach discussed here draws upon graphical multi-agent systems, such as particle systems (Reeves 1983), StarLogo (Resnick 1994), and behavioral animations of flocking "boids" (Reynolds 1987). The basic control philosophy for reactive swarm agents is inspired by work in autonomous mobile robots, especially the subsumption architecture of Brooks (1986) and the collective behavioral primitives developed by Mataric (1995). Robotic swarm agents are exemplified by minimalist, multiple robotic systems, such as reported in Kube and Zhang (1992) and Werger (1999). The principles for reactive swarm system have been used by the author in the successful construction of collective robotic systems (Altenburg 1995), investigations and simulation of yellowjacket wasps and their nest construction behavior (Altenburg 1999), and the exploration of command and control models for many unmanned aerial vehicles (Altenburg et al. 2008; Lua et al. 2003; Schlecht et al. 2003).

Some characteristics making swarms attractive include: (1) scalability, (2) fault tolerance, (3) reduced cost and complexity, and (4) distributed intelligence. The design of a swarm should allow the system to scale from one to many hundreds or thousands of agents. Through redundancy, swarms should demonstrate fault tolerance and a graceful degradation of performance if problems occur. Reactive agents are less complex than more deliberative ones, and a system composed of several simple agents should be less expensive than a more monolithic system composed of few complex agents. In a swarm, no single agent is in charge; rather, the intelligence of the system is distrusted throughout the system. The goal of most research in reactive swarm systems is to discover ways for the system to self-organize and a solution to emerge from the interactions of the agents.

Principles

The practice of exploring and constructing reactive swarm systems is early in its development. Arguably, it is more art than science. However, some general principles have been discovered to aid the student of swarms. Some of these principles are:

1. Randomness is the solution to unpredictability,
2. Minimal assumptions lead to maximum scalability,
3. Dependence on communication leads to failure,
4. Simple rules coupled with simple signals succeeds,
5. Always give yourself an out,
6. Many special agents are better than one super agent, and
7. Achieving goals through self-organization is tricky.

This list of principles is not intended to all inclusive. However, they address issues of significant interest for any student or researcher of reactive swarm systems. These principles are not exclusive of each other, one principle may support another. For example, randomness supports scalability. I will discuss and provide examples of each of these principles.

Randomness is the Solution to Unpredictability

Observers of natural phenomena, such as foraging in ants, view randomness as a valid response to unpredictability in an agents' environment (Deneubourg et al. 1987). Randomness plays a role in overcoming sub-optimal solutions such as a local minima or maximum. The accidental (random) deviation from a well travelled path may lead to the discovery of a new, better (i.e., shorter, faster, richer) route.

Randomness simplifies swarm agent control. Take the task of searching as an example. Given a well-defined, well-known search area with static targets, a systematic, optimal search technique can be derived. For a flat, round search area, an expanding spiral would work very well. On the other hand, a random walk search (i.e., billiards search) can be nearly 60% as effective as an optimal search without the computational overhead of a more complex algorithm or need for *a priori* information about the shape or size of the search area (Altenburg 1995). Furthermore, a random search works as well for one agent as it does for many agents in the same search area. The algorithm for a random search may be quite simple: more forward until the agent encounters an obstacle (or a timer expires), back up a bit, turn to a new random heading, and start moving forward again. How would the expanding spiral search need to be modified to address the exceptions of many moving obstacles (i.e., other agents) in a search space that is no longer circular, rather, an irregular, many-sided polygon? The random search would need little to no modification and the search would remain relatively effective.

Random wait times are a well known device for overcoming deadlock conditions in concurrent processes such as databases and telecommunications. Likewise, random timers can be used by reactive agents by overcome deadlock conditions such as two agents trying to reach the same goal through a constrained pathway. As both robots halt, each robot sets their own random wait timer. One robot's timer will run out before the other allowing the first robot to proceed. The robots will take turns entering the pathway without any explicit coordination. The idea of internal timers was introduced by Brooks (1986) as the augmentation in his augmented finite state machine control modules used in behavior-based, reactive robots.

Minimal Assumptions Lead to Maximum Scalability

A problem that must be addressed in the construction of any agent is the nature of the agents' environment. Some environments are more predictable than others. For example, simulated worlds that exist only within a computer are more predictable than the real world. An environment occupied by a single mobile agent is more predictable than a multi-agent environment. Brooks (1986) noted one of the failures of early robotics research to move from a simulated (idealized) environment to the real world were the researchers assumption. Unrealistic assumptions lead to solutions that cannot be implemented in the real world. Brooks advocated using the world as its own model; as opposed to an internal world model. One can explore control algorithms in simulation if one makes the appropriate assumption; and the fewer the assumptions the better.

The premise for minimal assumptions is the world is not perfect: sensors are noisy or fail, actuators vary, the terrain (or winds or waves) are chaotic, and other agents may get in the way. Both sensors and actuators could be modeled probabilistically; that is, sensors and motors have a certain chance of sensing and acting correctly. Managing all desired actions as requests for actions, which may or may not be granted, is another way to model the uncertainty of the environment. For example, an agent may desire to move forward, but the environment may deny the request, thus simulating a slippery surface or a strong headwind. When working in an environment filled with other cooperating agents, an individual agent should not assume other agents are available to help. On the contrary, in crowded environments with many agents, agents disrupt each other frequently. Therefore, an agent should be designed and implemented to complete part of its tasks in the absence of assistance from other agents and in the presence of agent-based disruptions.

The random search technique described earlier provides a good example of how limited assumptions lead to maximum scalability. A single agent conducting a random search can be designed with no assumption of the size or shape of the search area. Furthermore, no assumptions are needed about the number of other agents participating in the search. Implementing a random search for many agents is just as easy as implement the search technique for a single agent; that is, it is highly scalable.

The biggest assumption to avoid is assuming any one agent knows exactly what is going in the world. No agent is omniscient. Any solution that requires one agent to know where all the other agents are is doomed for failure. Reynolds (1987), when describing his flocking animation work (i.e., boids), states, “Not only is it unrealistic to give each simulated boid perfect and complete information about the world, it is just plain wrong and leads to obvious failures of the behavior model.”(p. 31) He goes on to say, “… the aggregate motion that we intuitively recognize as ‘flocking’ (or schooling or herding) **depends** upon a limited, localized view of the world.”

Dependence on Communication Leads to Failure

Cooperation, by definition, requires communication. However, an overreliance on communication can lead to failure. A well known axiom among both military officers and emergency workers is: the first thing to fail at a time of crisis is communications. Two approaches to overcoming a dependence on explicit inter-agent communication are: the use of implicit communication and agent independence. The United States Marine

Corps has long recognized the dangers of over-dependence on explicit communication among its “agents in the field” (i.e., Marines). Their solution to this problem is implicit communication (USMC 1997). They define implicit communication as: “to communicate through mutual understanding, using a minimum of key, well-understood phrases or even anticipating each other’s thoughts” (p. 78). They emphasize a widespread familiarity with a leader’s intent as a basis for implicit communication. For the design of agents, this may translate into shared plans. Of course, having a plan is counter to the idea of a reactive agent. Another implementation of implicit communication may come from a coupling of agent interactions to internal or external stimuli imposed by design. A simple example of implicit communication is each agent sharing a common return time and location, that is, at a certain pre-specified time, all agents will return to the home base.

The coordination of simultaneous action by several agents can be coordinated with explicit communication (Schlect et al. 2003; Lua et al. 2003). In general, an agent should view the receipt of a communicated signal as an exception rather than a requirement. For example, an agent could start a count-down timer and send out a “started waiting” signal while waiting to start a coordinated task, such as parallel sweep search or coordinated attack. The agent can then either wait until its timer expires, or, upon receipt of a “started waiting” signal from another agent, can reset its count-down timer and propagate the signal to other agents. Once one agent reaches the end of its count-down timer, it transmits a “done waiting” message that is propagated to other agents and all agents simultaneously begin their next task. This example demonstrates two important aspects of the use of explicit communication in reactive swarm system: independence and redundancy. First, the agents do not depend on communication to continue the execution of their overall task. Rather, it augments their own internal mechanisms, in this case a count-down timer. The agent may continue its overall mission independently in the absence of the communication. Second, each agents has a redundant mechanism to transition from one behavior to another; both a signal and a timer in this case.

A question to ask during the design of a reactive swarm system is: what happens if all communications fail. Does the mission fail? Do the agents enter a deadlock condition? If the answer is yes to either of these questions then it is best to consider designs that overcome these limitations due to a dependence on explicit communication.

Simple Rules Coupled with Simple Signals Succeeds

The agents in a reactive swarm system are often implemented as deterministic machines augmented with simple timers. They behaviors are prescribed by simple rules. These simple rules are one of the great benefits of reactive swarm agents. One reason for this benefit is simple rules are understandable. A lament of modern software engineers is systems are so complex no-one fully understands them and there is no way to adequately test them for the presence of fatal errors. Reactive swarms systems attempt to address this problem by ensuring the elements of the system, the agents, are simple. By decomposing the overall behavior of an agent into simple rules, each behavior can be more easily described and tested. For the sake of understandability and testability, the complexity of the system should come from the interaction of the agents and their environment, not from the agents themselves. Likewise, the communication that binds agents together should be simple.

Communications comes in a variety of forms and degrees of complexity. The simplest form of communication is indirect cues. For example, two robots approaching each other may detect the other robot's obstacle detection signal such as light from an infrared LED. Both robots will likely turn away from each other because they sense an obstacle. Although neither robot was intentionally communicating with another robot, their obstacle detection system sent indirect cues that signal an appropriate behavior by each robot. For robots whose mission involves searching an area, the simple obstacle avoidance mechanism serves double duty by dispersing them so they do not cover the same search area (Altenburg 1995).

Another form of simple signals is what may be referred to as stimulus amplification. For example, the same sensors and behaviors used to find a target may be co-opted to recruit other robots during a collective search task. Altenburg and Pavicic (1993) describe such a situation where a collection of robots sought a cylinder marked with lights. The robots were to search for the target, find it, and then return to a home base once the target was found. This task was accomplished by having a robot turn on its own set of lights which were much brighter then the target's own lights once the robot found the target. The other robots would then be attracted to the transmitting robot as though it was the target. As the finding robot carried the target to the home base, the other robots would follow it and return likewise. This method of stimulus amplification has the advantage of employing the redundant use of sensors and behaviors and, thus, reducing the overall complexity of the individual robots.

Modulating the behaviors and activity of the entire swarm systems may be accomplished through simple alarm signals. One example is described in a model of the alarm process of Australian bulldog ants (Frehland et al., 1985; Adler and Gordon, 1992). In this model, sentinel ants patrolled the boundaries of the colony territory conducting random walks or resting. Once alarmed, the sentinels will run until in contact with another ant and nearly attack the other ant. This near attack excites the second ant and it too will propagate the alarm signal and, thus, heightening the state of alertness among the whole colony. A similar alarm system has been observed in yellowjacket wasps (Altenburg 1999). When the outside of a wasp nest is even lightly tapped, several wasps exit the nest. Apparently they exit the nest in anticipation of defending the nest against a predator. An individual wasp flying near the nest may activate this defensive behavior by flying into the side of the nest after sensing a possible predator near the nest.

Alarm-based recruitment has also been used for the efficient allocation of robot resources during collective search and retrieval-type tasks (Altenburg 1994; Altenburg 1995). In this task, a collection of robots searched for targets, and, upon finding the targets, cleared them from the search area; that is, an area cleaning task. The task was complicated in that not all targets were available at once. Small batches of targets were added to the search area periodically throughout the task. Rather than having all robots search continuously, robot resources were conserved by having the majority of robots enter a resting state after no new targets were discovered after some period of time. A few robots remained active searching and waiting for the arrival of new targets. Once one of these sentinel robots discovered a new target, it would transmit an alarm signal to arouse the resting robots and all targets were quickly found and cleared.

Always Give Yourself an Out

One of the challenges of using purely reactive agents is enforcing the persistence of a particular behavior over time. That is, if a behavior is activated by a particular stimulus (i.e., bumping into an obstacle, or sensing a beacon), the agent's reaction to the stimulus must persist after the stimulus is no longer sensed. A purely reflexive system would have very little persistence of action and is liable to enter into a behavioral loop; endlessly repeating the same reflexive actions. Brooks (1986) addressed this issue with the inclusion of timers in his augmented finite state machines to control his behavior-based robots. A timer allows an agent to momentarily ignore a stimulus and remain in a particular behavioral state.

A small amount of internal state can aid in overcoming behavioral loops. Gat et al. (1994) describe the use of a state variable, referred to as *frustration*, to exit dead-end allays and cul-de-sacs by a small robot named Tooth exploring an office floor and collecting Styrofoam coffee cups. Tooth keeps track of the frequency of turning during a short period of time. If this frequency becomes too high, such as when Tooth enters a cul-de-sac, Tooth will steer in a random direction in an effort to exit this local trap. Likewise, Tooth solves the problem of dead ends by successively increasing the distance traveled while backing up if repeatedly forced into the same dead end. Likewise, a similar state variable gradually diminishes the attractiveness of a possible target after several failed attempts to collect it, such as in the case when Tooth mistakenly identifies an immovable table leg for a coffee cup.

In (Schlecht et al. 2003) we introduced the use of virtual beacons which were inspired by waypoints entered into a GPS navigation system to aid in task continuity. A virtual beacon serves as a persistent stimulus in the agent memory and allows the agent to execute other behaviors without losing track of a higher-level objective. In this task, a swarm of reactive agents representing unmanned aerial vehicles (UAVs) conducted a search-and-destroy mission on static targets. When conducting a systematic search of an area, each agent set a virtual beacon as an end-of-track waypoint on the far side of a search track. If the agent is interrupted mid-search by, say, a possible collision with another search agent, a "book mark" waypoint is set and the original waypoint is ignored. The agent then executes an appropriate behavior to deal with the immediate problem. Once the problem is resolved, the agent seeks out the "bookmark" way point and, upon crossing that way point, continues on its search track en route the original end-of-track waypoint. The UAV agent does not fixate on a single goal to exclusion of all other goals or stimuli.

Although many reactive swarm agents are implemented using deterministic rules, the agents should not be blindly obligated to a particular task or stimulus. The use of timers, limited internal state information, or simple memory aids, such virtual beacons, each allow an agent to overcome blind determinism.

Many Special Agents are Better than One Super Agent

The division of labor by specialized castes (polymorphism) among social insects is considered a highly successful evolutionary strategy (Wilson 1971). Within the ants, bees, and wasps (order Hymenoptera), the most basic castes are queens, drones, workers, and soldiers. Contrary to popular belief, the queen is not the ruler or central controller of the colony, rather, her job is to lay eggs. Drones, the male caste, are relegated almost solely to gene propagation trough sex. Workers conduct the work of the colony while soldiers defend the colony. Artificial reactive swarm systems benefit greatly from following the type of agent specialization seen in social insects.

Heterogeneity of agents may come in several different forms including different behaviors or different physical capabilities. For the area cleaning robots described earlier, there were slight differences between that majority of searchers and the sentinel-searchers. The sentinel-searchers not only remained active while the other searchers rested, the sentinel-searches were equipped with signal transmitters allowing for the broadcast of the alarm signal. In Nygard et al. (2004) we described the implementation of hunter-killer UAV teams as an example of the division of labor among agents. Hunters search for and track targets while the killers attack the targets. Hunters fly low and slow and are equipped with target sensors while killers fly high and fast and are equipped with weapons. In addition, hunters work in pairs. Upon finding a target, one hunter flies low to track the target and to ensure it is not lost. The discovering hunter recruits a second hunter to fly higher, acting a communications relay, and recruits a killer. The lower flying hunter is more vulnerable to possible attack by the target. However, if the tacking hunter is lost, the relay hunter assumes the role of the tracker and attempts to recruit a new relay partner. The killer need not have a sensor to identify the target; rather, a hunter can "paint" the target with a laser which the killer can use for targeting. This simple division of labor reduces individual agent complexity and increases mission success by: 1) having redundant tracking capabilities, thus, reducing the chance of losing the target, 2) ensuring contact with the killer agents despite limited range and power of communications, and 3) limiting the exposure of killers to the risk of attack by an aggressive target.

Achieving Goals through Self-organization is Tricky

The product of interest in a reactive swarm system emerges through self-organization. Designing the emergence of a global phenomenon is not only difficult but is seemingly a contradiction of terms; how can something be planned to emerge? The task appears even more daunting when the elements (agents) composing the system have no perception of the global pattern they are forming. Designing reactive swarm systems to achieve a particular goal requires an ability to envision, describe, and design systems at multiple levels of interaction and through extended periods of time.

Perhaps one of the biggest challenges to designing self-organizing systems, as Resnick (1994) observed, is overcoming the centralized mindset. He noted people often assume centralized control where there is none (e.g., flocks of birds must have a leader). It is hard to think about solutions that are not centrally controlled. Clearly, it would easier to ensure system-wide behavior if each agent could communicate without error to every other agent. However, this is unrealistic outside of computer simulation. Students and researchers interested in taking advantage of the promises of reactive swarm systems must learn new ways to view problems to design decentralized solutions. One example Resnick provides is the formation of a circle with a prescribed center and radius using turtle graphics. A centralized solution is to have the turtle start at the center of circle, move forward the distance of the radius, turn right, lower its drawing pen, and then repeatedly move forward one step and turn one degree. After 360 slight moves and turns, a circle is drawn. The decentralized approach places 5000 turtles at the circle's center, has each turtle choose a random direction, then each turtle moves forward the distance of the circle's radius. In this case, the turtles *are* the circle. Observing, modeling, and exploring decentralized systems would aid students and researchers in overcoming the centralized mindset.

Two concepts that aid the design of reactive swarm system are collective primitives and the acceptance of probabilistic near-certainty. Mataric (1995) proposed a set of collective primitive (or basic interactions) for synthesizing behavior in multi-robot systems. These primitives are: avoidance, attraction, following, dispersion, aggregation, homing, and flocking. Nearly all multi-agent, system-wide behavior can be either synthesized or decomposed into one of these primitives. For example, a higher level group behavior such as herding could be synthesized by combining flocking together and homing in on a common goal. Another higher-level activity, foraging, combines dispersion and homing. The design of reactive swarm systems is facilitated through the use these types of second-order behavioral design modules.

Individual agents in a reactive swarm system often behave very deterministically, however, due to the complexity of the environment including agent-agent interaction, the system's behavior may appear chaotic. Simon (1981) noted this simplicity-complexity duality when describing the random path taken by an ant traversing a rugged landscape. Given the apparent chaos of a reactive swarm system, how can a designer state with any certainty that anything actually will be accomplished? As a system, goals can be accomplished if the designer accepts the concept of probabilistic near-certainty. For example, there is no guarantee a random walk search will find anything. There is always a chance the one spot where a target is located will never be visited by the random searcher. However, for the reactive swarm system investigator, the converse is nearly equality true: given enough time or enough agents the target will be found. Like the random search strategy, reactive agents endowed with limited communication and computation, faulty sensors and actuators, and only a loose coupling between each agent, there may be a sense that the system is doomed to failure. On the contrary, through the use of redundancy, simple well-understood behaviors, division of labor, minimal assumptions, and massive agent multiplicity, the system will near-certainly achieve its goals. Grassé (1959) coined the term *stigmergy* to explain how the random pellet-piling behavior of blind termites is coupled through a positive-feedback mechanism of the piles themselves results in the construction of an elaborate termite mound many thousands times their own size.

Summary

Reactive swarm systems are attractive because they promise several desirable characteristics including: (1) scalability, (2) fault tolerance, (3) reduced cost and complexity, and (4) distributed intelligence. Reactive swarms are multi-disciplinary in their origin and in their applications including robotics, biology, and agent-based computer simulations. Understanding or designing reactive swarm systems is challenging due to their emergent, self-organization nature.

A set of general principles for exploring and constructing reactive swarm systems was presented. Adherence to these principles is not required for the implementation or understanding of swarms; they advocate a philosophy which has led to several successful swarm studies. Is has been demonstrated that simplicity, limited communication, minimalism, independence, and the division of labor through specialization are good design and descriptive constructs. The dangers of unchecked assumptions and the centralized mindset should be self-evident. It is my hope that these principles, and the examples provided, will inspire more students and researchers to explore reactive swarm systems.

References

Adler FR, Gordon DM. 1992. Information collection and spread by networks of patrolling ants. The American Naturalist. 140(3)373-400.

Altenburg K. 1994. Adaptive Resource Allocation for a Multiple Mobile Robot System Using Communication. NDSU Technical Report: NDSU-CSOR-TR-9404. Fargo, ND. 7 p.

Altenburg KR. 1995. COCOBOTS: Robots, Ants, and Randomness. MS Thesis. NDSU: Fargo, ND, 238 p.

Altenburg KR. 1999. Construction of Yellowjacket Wasp Nests: Internal Models of Collective Reactive Behavior? Ph.D. Dissertation. NDSU: Fargo, ND, 172 p.

Altenburg K, Hennebry M, Pikalek J, Nygard KE. 2008. Simian: A multi-agent simulation framework for decentralized UAV task allocation. ISAST Transactions on Intelligent Systems. 1(1):69-73.

Altenburg K, Pavicic M. 1993. Initial results of the use of inter-robot communication for a multiple, mobile robotics system. pp 95-100. In: Kanazawa K (editor), IJCAI 1993 Workshop on Dynamically Interacting Robots, Chambery, France, 139 p.

Brooks RA. 1986. A layered control system for a mobile robot. IEEE Journal of Robotics and Auomation. 2(1):14-23.

Deneuborg JL, Goss S, Pasteels JM, Fresneau D, Lachaud JP. 1987. Self-organization mechanisms in ant societies (II): Learning in foraging and division of labor. pp. 177-196 In: Deneuborg JL, Pasteels JM, From Individual to Collective Behavior in Social Insects, Birkhauser Verlag: Basel, 433 p.

Frehland E, Kleutsch B, Markl H. 1985. Modeling a two-dimensional random alarm process. Biosystems. 18(2):197-208.

Gat E, Desai R, Ivlev R, Loch J, Miller DP. 1994. Behavior control for robotic exploration of planetary surfaces. IEEE Journal of Robotics and Automation. 10(2): 490-503.

Grassé PP. 1959. La reconstruction du nid et les coorinations interindividuelles ches *Bellicositermes natalensis* et *Cubitermes* sp. La théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. Insectes Sociaux. 6(1):41-83.

Kube CR, Zhang H. 1992. Collective robotic intelligence. pp. 460-468. In: Meyer JA, Roitblat HL, Wilson S (editors), Second International Conference on Simulation of Adaptive Behavior. MIT Press: Cambridge, MA, 523 p.

Lua CA, Altenburg K, Nygard KE. 2003. Synchronized multi-point attack by autonomous reactive vehicles with simple local communication. In: 2003 IEEE Swarm Intelligence Symposium Proceedings. Indianapolis, IN. 95-102.

Mataric MJ. 1995. Designing and understanding adaptive group behavior. Adaptive Behavior. 4(1):51-80.

Nygard KE, Altenburg K, Jingpeng T, Schesvold D, Pakalek J. 2004. Heterogeneous swarm agents in hunter-killer teams. Presented at: 2nd Annual Swarming: Network Enabled C4ISR Conference. Arlington, VA, June 14-15, 2004.

Reeves WT. 1983. Particle Systems – A technique for modeling a class of fuzzy objects. Computer Graphics. 17(3): 359-376.

Resnick M. 1994. Turtles, Termites and Traffic Jams: Explorations in Massively Parallel Microworlds. MIT Press: Cambridge, MA, 163 p.

Reynolds C. 1987. Flocks, herds, and schools: A distributed behavioral model. Computer Graphics 21(4):25-34.

Schlecht J, Altenburg K, Ahmed BM, Nygard KE. 2003. Decentralized search by unmanned air vehicles using local communication. In: Arabnia HR, Joshua R, Mun Y (editors), Proceedings of the International Conference on Artificial Intelligence 2003 (Volume II). Las Vegas, NV. 757-762.

Simon HA. 1981. The Sciences of the Artificial. MIT Press: Cambridge, MA. 247 p.

United States Marine Corps. 1997. Marine Corps Doctrinal Publication 1: Warfighting. 106 p.

Werger BB. 199. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. Artificial Intelligence. 110(2):293-320.

Wilson EO. 1971. The Insect Societies. Belknap Press: Cambridge, MA, 548 p.