A Formal Approach to Self-configurable Swarm-based Space-exploration Systems

Emil Vassev[1], Mike Hinchey[2], Paddy Nixon[3]
[1, 3]*Lero–the Irish Software Engineering Research Center, University College Dublin, Ireland*
[2]*Lero–the Irish Software Engineering Research Center, University of Limerick, Ireland*
emil.vassev@lero.ie, mike.hinchey@lero.ie, paddy.nixon@lero.ie

Abstract

Intelligent swarms draw their inspiration from biology where many simple entities act independently, but when grouped, they appear to be highly organized. NASA is currently investigating swarm-based technologies for the development of prospective exploration missions to explore regions of space where a single large spacecraft would be impractical. The main emphasis of this research is to develop algorithms and prototyping models for self-managing swarm-based space-exploration systems. This article presents our work on formally modeling self-configuring behavior in such systems. We present a formal model for team formation based on Partially Observable Markov Decision Processes and Discrete Time Markov Chains along with formal models for planning and scheduling.

1. Introduction

Biologically-inspired software systems adopt biological approaches to effective problem solving, where solutions developed by nature through evolution are applied in the computing milieu. Concepts in biology inspired the autonomic computing initiative [1, 2], which has arisen for self-management of complex systems. The idea is that biological systems (in particular the autonomic nervous system) are capable of doing autonomous self-regulation activities, thus inspiring principles for software autonomic systems that are capable of self-management.

NASA is currently investigating biologically inspired swarm technologies for the development of prospective exploration missions that will be autonomous and exhibit autonomic properties. The Autonomous Nano-Technology Swarm (ANTS) [3, 4] is a new class of concept missions based on swarm intelligence [5, 6] attained through cooperative interactions of the swarm entities. Conceptually, an ANTS swarm system is composed of many little spacecraft of different classes often grouped in *exploration teams* (sub-swarms). Note that a swarm-based system offers many advantages compared with the single-spaceship system, such as greater redundancy, reduced costs and risks, and the ability to explore regions of space where a single large spacecraft would be impractical.

Research Problem & Approach. ANTS is envisioned to operate autonomously, which involves autonomous team formation. This helps special teams be formed on-the-fly by allocating spacecraft of different classes. Such a task requires comprehensive knowledge of the global system's state and that of the individual spacecraft together with the goal state. Such knowledge may be obtained through exhaustive monitoring and reason that bring ANTS to an appropriate level of self-awareness. There are several factors that may have an impact on the team formation algorithm including swarm size, number of spacecraft per class and that of freelance spacecraft, existing teams and their tasks, communication range, etc.

This paper tackles an algorithm for self-configuring of ANTS teams. A formal model for self-configuring behavior is presented where both the system as a whole and individual spacecraft are analyzed from a state-goal perspective. In this approach, the global system goals may trigger team formation or reformation driven by team goals derived from the system's ones. The team goals are realized by individual spacecraft performing assigned tasks. In this paper, we present a formal model for team formation based on the so-called Partially Observable Markov Decision Processes and Discrete Time Markov Chains. Note that task scheduling is a separate problem tackled by this research and described in greater detail by another paper of ours [7].

The rest of this paper is organized as follows. In Section 2, we review intelligent swarms and present the swarm-based ANTS. Section 3 presents in detail our formal approach to self-configuring in ANTS, where formal models for *team formation*, *planning*, and *scheduling* are presented. Finally, related work, conclusions and future work are outlined in Section 4.

978-1-4244-5889-9/10/$26.00 ©2010 IEEE

2. Intelligent Swarms

Intelligent swarms [6] are complex swarm systems where the individual members of the swarm have independent intelligence. Multi-agent systems may be considered as swarm-based systems where many *intelligent* agents interact with each other [8]. These agents are considered to be autonomous entities that interact either cooperatively or non-cooperatively (on a selfish base). Multi-agent systems differ in factors, such as system organizational models, interaction models, the individual agent complexity, etc. As it is stated in [9], considering the multi-agent organizational models two main approaches can be distinguished: *agent-centered* and *organization-centered*. The agent-centered approach is more complex due to the more complex nature of the agents, which reason on their tasks and relations such as *join-intentions*, *join-commitments*, *dependencies*, etc. The organization-centered approach exposes agent relations defined a priori and imposed on the agents.

2.1. NASA Swarm-based Projects

The Autonomous Nano-Technology Swarm (ANTS) concept sub-mission PAM (Prospecting Asteroids Mission) is a novel approach to asteroid belt resource exploration. By its virtue, ANTS necessitates multi-agent systems following the *organization-centered* approach and based on extremely high autonomy, minimal communication requirements to Earth, and a set of very small explorers with a few consumables [3, 4]. These explorers forming the swarm are *pico-class*, *low-power*, and *low-weight* spacecraft units, yet capable of operating as fully autonomous and adaptable agents. The spacecraft units in an ANTS swarm are able to interact with each other, which helps them to self-organize.

Figure 1 depicts the ANTS concept mission. A transport spacecraft launched from Earth toward the asteroid belt carries a laboratory that assembles the tiny spacecraft. Once it reaches a certain point in space, where gravity forces are balanced, the transport releases the assembled swarm, which will head for the asteroid belt. Each spacecraft is equipped with a solar sail, thus it relies primarily on power from the sun, using only tiny thrusters to navigate independently. Moreover, each spacecraft also has onboard computation, artificial intelligence, and heuristics systems for control at the individual and team levels.

As Figure 1 shows, there are three classes of spacecraft - *rulers*, *messengers* and *workers*. By grouping them following exploration goals, ANTS forms teams that explore particular asteroids. Hence, ANTS exhibits self-organization since there is no external force directing its behavior and no single spacecraft unit has a global view of the intended macroscopic behavior. The internal organization of a swarm depends on the global task to be performed and on the current environmental conditions. In general, a swarm consists of several sub-swarms, which are temporal teams organized to perform a particular task. Each swarm team has a team leader (*ruler)*, one or more *messengers*, and a number of *workers* carrying a specialized instrument. The messengers are needed to connect the team members when they cannot connect directly, due to a long distance or a barrier.

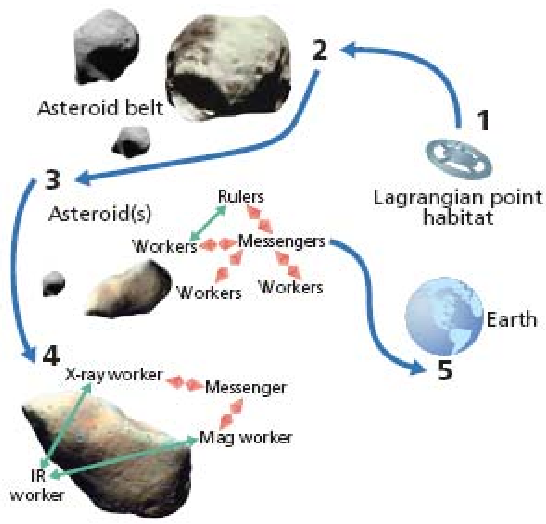

Figure 1. ANTS Mission Concept [3]

The following elements briefly describe the three different classes of spacecraft in ANTS.

- *Messengers* coordinate communication between the rulers and workers, and communications with the Earth ground station. For example, they can alert NASA to send replacement spacecraft from Earth or one with additional instruments.
- *Rulers* are coordinators that have rules that decide the types of asteroids and data the mission is interested in and that will coordinate the efforts of the workers.
- *Workers*, up to 80 percent of the swarm, bear the instruments and gather data. Instruments can include a magnetometer, *x*-ray, gamma-ray, visible/infrared, or neutral mass spectrometers.

3. Self-configurable Behavior in ANTS

We envision self-configuring in ANTS as an automatic team-formation technique where teams are formed and spacecraft are assigned new tasks on the fly.

In general, the self-configuring task may be divided into three phases:

(i) **Team formation**. In this phase, an idle ruler is delegated with the task of team formation. This phase has two steps:

1) *ruler self-initiation* to perform a team-formation task;
2) *team formation* to find potential team members among the idle (freelance) spacecraft of the ANTS swarm.

Here, depending on the specifics of the goal of the new team, arbitrary numbers of idle workers carrying task-required instruments and messengers are allocated by the ruler to form the new team.

(ii) **Planning.** In this phase, the ruler comes up with a *plan of action* to achieve the team goal. Such a plan consists of a *sequence of tasks* broken down by instrument (instrument tasks), where some tasks could be performed in parallel, but maybe need to be synchronized at the end. A task can be performed by workers equipped with the instrument this task is related to.

(iii) **Scheduling.** In this phase, the ruler schedules the tasks among the workers. Since the workers can perform different tasks, there are two concerns here:

- Find an idle worker with appropriate instrument. If no worker is available, ask the other teams for additional worker(s).
- When more workers are available, break down the task into sub-tasks and assign to each idle worker a sub-task.

3.1. Formal Approach

In this section, we give an overview of our formal approach to the Team Formation. Both Planning and Scheduling phases are briefly described as well. For more details on these two phases, please, refer to [7].

3.1.1. Self-initiation for Team Formation. In the first step of the Team Formation phase, an idle ruler interacts with the swarm to come up with self-initiation of team formation. For this, a behavior model based on the so-called Partially Observable Markov Decision Processes (POMDP) [10] is considered. Note that this model is appropriate when there is uncertainty and lack of information needed to determine the state of the swarm. For example, in ANTS spacecraft are supposed to operate under harsh conditions in space limiting the communication. In addition, an idle ruler does not actively participate in the swarm's activities, and thus, it is not well informed of the swarm states. Therefore, the POMDP model helps an idle ruler reason on the current swarm state and thus, to self-initiate itself when a new team is needed to be formed for the exploration of new asteroid. According to our POMDP-based model, an idle ruler takes as input the state of the swarm and generates as output actions initiating team formation, i.e., the generated actions affect the state of the swarm. Formally, this model is a tuple $\boldsymbol{M = \langle S; A; T; R; Z; O \rangle}$, where:

- $\boldsymbol{S}$ is a finite set of states of the swarm that are not observable.
- An initial belief state $\boldsymbol{s_0 \in S}$ is based on $\boldsymbol{p_0 (s_0; s_0 \in S)}$, which is a discrete probability distribution over the set of swarm states $\boldsymbol{S}$, representing for each state the ruler's belief that is currently occupying that state.
- $\boldsymbol{A}$ is a finite set of actions that may be undertaken by the idle ruler.
- $\boldsymbol{T: S \times A \rightarrow \Pi(S)}$ is the state transition function, giving for each swarm state $\boldsymbol{s}$ and ruler action $\boldsymbol{a}$, a probability distribution over states. Here, $\boldsymbol{T (s; a; s')}$ computes the probability of ending in state $\boldsymbol{s'}$, given that the start state is $\boldsymbol{s}$ and the ruler takes action $\boldsymbol{a}$, $\boldsymbol{p (s' \mid s; a)}$.
- $\boldsymbol{O: A \times S \rightarrow \Pi(Z)}$ is the observation function giving for each swarm state $\boldsymbol{s}$ and ruler action, a probability distribution over observations $\boldsymbol{Z}$. For example, $\boldsymbol{O (s'; a; z)}$ is the probability of observing z, in state $\boldsymbol{s'}$ after taking action $\boldsymbol{a}$, $\boldsymbol{p (z \mid s'; a)}$.
- $\boldsymbol{R : S \times A \rightarrow R}$ is a reward function, giving the expected immediate reward gained by the ruler for taking an action in a state $\boldsymbol{s}$, e.g., $\boldsymbol{R (s; a)}$. The reward is a scalar value in the range [0..1] determining, which action (among many possible) should be undertaken by the ruler in compliance with the swarm goals.

Interpretation. To illustrate this model, let's assume that an ANTS swarm is currently occupying the state $\boldsymbol{s}$ = "*new asteroid is discovered, but no exploration team has been formed yet and still no ruler is self-initiated for team formation*". Let's assume there is at least one idle ruler in the swarm ready to undertake a few actions $\boldsymbol{A}$, including the action $\boldsymbol{a}$ = "*self-initiation for team formation*". The ruler performs the following reasoning steps in order to self-initiate for team formation.

1) The ruler computes its current belief state $\boldsymbol{s_0}$ – the ruler picks up the state with the highest probability $\boldsymbol{p_0}$ and eventually $\boldsymbol{s_0 = s}$.
2) The ruler computes the probability $\boldsymbol{p_1}$ of the swarm occupying the state $\boldsymbol{s'}$ = "*new asteroid is*

discovered and a ruler has self-initiated for team formation" if the action a is undertaken from state s_0.
3) The ruler computes the probability $p_2\ (z \mid s';\ a)$ of observation z = "*there are sufficient numbers of idle workers carrying needed instruments and messengers to form a new exploration team*".
4) The ruler computes the reward $r\ (s_0;\ a)$ for taking the action a (*self-initiation for team formation*) in state s_0. If no other immediate actions should be undertaken (forced by other swarm goals), the reward r should be the highest possible, which will determine the execution of action a.

3.1.2. Probability Computation. The POMDP model for self-initiation requires the computation of a few probability values. In this subsection, we present a *model for assessing probability* applicable to the computation of POMDP probability values such as probability of the swarm being in a state and probability of observation (cf. Section 3.1.1). In our approach, the *probability assessment* is an indicator of the number of possible execution paths a ruler may take, meaning the amount of certainty (excess entropy) in the swarm's behavior. To assess that behavior prior to the swarm implementation, it is important to understand the complex interactions among the spacecraft in an ANTS swarm. This can be achieved by modeling the behavior of individual reactive spacecraft together with the team behavior as *Discrete Time Markov Chains* [11], and assessing the level of probability through calculating the probabilities of the state transitions in the corresponding models. We assume that the ruler-swarm interaction is a stochastic process, where the swarm events are not controlled by the ruler and thus their probabilities are considered equal.

The theoretical foundation for our Probability Assessment Model is the property of Markov chains, which states that, *given the current state of the swarm, its future evolution is independent of its history*, which is also the main characteristic of a reactive autonomic spacecraft.

Table 1. Transition Matrix P

	S1	S2	…	Sj	…	Sn
S1	p_{11}	p_{12}	…	p_{1j}	…	p_{1n}
S2	p_{21}	p_{22}	…	p_{2j}	…	p_{2n}
…	…	…	…	…	…	…
Si	p_{i1}	p_{i2}	…	p_{ij}	…	p_{in}
…	…	…	…	…	…	…
Sn	p_{n1}	p_{n2}	…	p_{nj}	…	p_{nn}

An algebraic representation of a Markov chain is a matrix (called *transition matrix*) (cf. Table 1) where the rows and columns correspond to the states, and the entry p_{ij} in the i^{th} row, j^{th} column is the transition probability of being in state s' at the stage following state s. The following property holds for the calculated probabilities:

$$\Sigma_j\ p_{ij} = 1 \quad (1)$$

We contend that probability should be calculated from the *steady state* of the Markov chain. A steady state (or *equilibrium state*) is one in which the probability of being in a state before and after a transition is the same as time progresses. Here, we define probability for a swarm configuration composed of k spacecraft as the level of certainty quantified by the source excess entropy, as follows.

$$\textit{Probability (ANTS)} = \Sigma_{i=1,k} H_i - H \quad (2)$$

$$H = -\Sigma_i\ v_i\ \Sigma_j\ p_{ij}\ \log_2(p_{ij}) \quad (3)$$

Here,

- H is an entropy that quantifies the level of uncertainty in the Markov chain corresponding to an ANTS swarm;
- H_i is a level of uncertainty in a Markov chain corresponding to a spacecraft (e.g., an idle ruler);
- v is a steady state distribution vector for the corresponding Markov chain;
- p_{ij} values are the transition probabilities in the extended state machines that model the behavior of the i^{th} spacecraft.

Note that for a transition matrix P, the steady state distribution vector v satisfies the property $v * P = v$, and the sum of its components v_i is equal to 1.

Interpretation. The level of uncertainty H is exponentially related to the number of *statistically typical paths* in the Markov chain. Having an entropy value of 0 means that there is no level of uncertainty in a Markov system for a specific spacecraft's behavior. Here, a higher value of a probability measure implies less uncertainty in the model, and thus, a higher level of predictability.

3.1.3. Modeling ANTS Behavior as Discrete Markov Chains. Note that ANTS behavior may be modeled at two levels: *individual spacecraft level* and the *level of exploration team*.

Markov Chain for Spacecraft. The behavior of spacecraft is modeled as an *extended state machine* [12]. We construct Markov chains for the participating reactive spacecraft from their extended state machines, i.e., states are mapped to the states of the corresponding Markov chain. We assume the most common *stochastic queuing model* for the arrival time of the external (swarm) events (e.g., "*a new asteroid has been discovered*"), namely *Poisson Distribution* (derived from first principles and the notion of Markov chains, which are extended to include continuous time Markov chains) [11]. Let Ext_i be the set of external events and Int_i the set of internal events triggering a change from a state s to another state s'. The transition probabilities are calculated as follows:

1) If Ext_i is empty, then the probability of a transition due to an internal event is:

$$pij = 1 / | Int_i | \qquad (4)$$

Similarly, the probability of a transition due to an external event is

$$pij = 1 / | Ext_i | \qquad (5)$$

when Int_i is empty.

2) If both Ext_i and Int_i are non-empty, the probability of a transition due to an external event is first calculated, as follows:

$$p^{ext}_{ij} = \Sigma_i \, (1/n) / | Ext_i | \qquad (6)$$

where n is the total number of external events for the spacecraft and $1/n$ is the equal probability of each external event taking place. Next, the probability of an internal event is calculated, as follows.

$$p^{int}_{ij} = (1 - \Sigma_i \, (p^{ext}_{ij})) / | Int_i | \qquad (7)$$

3) When there is more than one transition with identical source and destination states, the above transitions are substituted in the Markov model by one for which probability p_{ij} is equal to the sum of the probabilities of the corresponding transitions.

Markov Chain for Exploration Teams. The behavior model for ANTS exploration teams (consisting of a *ruler*, *workers*, and *messengers*) is a synchronous product machine of the single-state machines of the participating spacecraft, where the synchronization between the spacecraft is achieved via shared external events. The probabilities are computed as described above.

In general, the behavior of the ANTS exploration teams is driven by the *global swarm goals*. Here, team rulers (including idle rulers) determine their next action (resulting into team tasks) based on the reward function R (cf. Section 3.1.1) determining the action with the best reward (highest compliance to the swarm goals). Thus, a ruler determines its next action, by undertaking from state s the action a leading to the highest compliance to one of the m swarm goals G:

$$R\,(s, a) \approx max \,\{ Compliance\,(g_i; g_i \in G) \}_{i = 1,m} \qquad (8)$$

The process of probability and reward assessment is based on the concept of Intelligent Control Loop [1, 2] and consists of the following steps:

1) **Monitor:** continuously track evolving changes within the swarm, such as configuration changes, time constraints change, and synchronization axioms change.
2) **Analyze:** assess probability values and rewards according to the evolving changes collected at the monitoring stage, and make the decision as to whether or not those changes are acceptable.
3) **Plan:** arrange how the changes will be adapted, such as setting of the appropriate timing and checkpoints to apply the changes, and arrange how the team status and the spacecraft status will be restored if the execution fails.
4) **Execute:** apply the changes that were accepted at the Analyze stage, and follow the plans made at the Plan stage.

3.1.4. Team Formation. In the second step of the Team Formation phase, the ruler allocates idle worker and messenger spacecraft to form a team. To accomplish this task, the self-initiated ruler simply broadcast a "team formation" message to the entire swarm. Any idle worker or messenger replies with a "join" message and eventually is assigned to the new team with an "assigned" message sent by the ruler (cf. Figure 2).

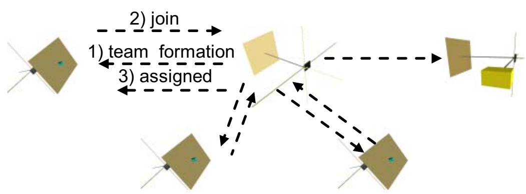

Figure 2. "Team Formation" Broadcasting

To assign new *workers* and *messengers* to the team, the *ruler* follows a simple algorithm where together with the targeted number of team members (team size) per spacecraft class, properties of individual spacecraft are considered as well. Properties of major importance are:

- team membership $\boldsymbol{m}$ (a worker member of another team may also be in idle state);
- class of spacecraft $\boldsymbol{cl}$ (worker or messenger);
- instrument on board $\boldsymbol{in}$ (cf. Section 2.1).

Thus, the formal model of the *function* $\boldsymbol{AF}$ that assigns idle spacecraft to the team is:

$$\boldsymbol{TF\,(\,Ts(in),\,m,\,cl,\,in\,)} \tag{9}$$

Here, $\boldsymbol{Ts(in)}$ is a function computing the need of more team members carrying an instrument $\boldsymbol{in}$.

3.1.5. Planning. The formal planning model for ANTS [7] is derived from work described in [13]. In this model, we consider for a given ANTS swarm:

- the initial state of the team denoted by $\boldsymbol{s_0}$;
- the goal states (set of goal states) denoted by $\boldsymbol{G}$;
- the possible *instrument tasks* that can be performed by the *workers,* denoted by:

$$\boldsymbol{R = \{r_1,\, r_2,\, \ldots.\, r_n\}} \tag{10}$$

- the time, if the goal $\boldsymbol{G}$ is time-constrained, denoted by $\boldsymbol{t}$;

to determine a plan $\boldsymbol{P}$, as a sequence of tasks $\boldsymbol{R}$ that transforms $\boldsymbol{s_0}$ to $\boldsymbol{G}$ for the time $\boldsymbol{t}$. Here, the formal definition of Planning could be:

$$\boldsymbol{P = f(s_0,\, G,\, t) = \{\, S_1,\, S_2,\, \ldots.\, S_n\,\}} \tag{11}$$

where,

$$\boldsymbol{G = \{\, g_1,\, g_2,\, \ldots.\, g_n\,\}} \tag{12}$$

and $\boldsymbol{g_1,\, g_2,\, \ldots.\, g_n}$ are intermediate goal states and $\boldsymbol{S_1,\, S_2,\, \ldots.\, S_n}$ are series of instrument tasks, needed to achieve states $\boldsymbol{g_1,\, g_2,\, \ldots.\, g_n}$ respectively. A series is a set of instrument tasks performed sequentially $\boldsymbol{S\rightarrow}$ or in parallel $\boldsymbol{S}||$. Here, the notation $\rightarrow$ means sequential execution, and the notation || means parallel execution.

$$\boldsymbol{Si \in \{\, S\rightarrow,\, S||\,\},\quad i \in N} \tag{13}$$

$$\boldsymbol{S\rightarrow\; = \{\, r_i \rightarrow r_i \rightarrow \ldots \rightarrow r_i\,\}} \tag{14}$$

$$\boldsymbol{S||\;\; = \{\, r_i \,||\, r_i \,||\; \ldots.\, ||\, r_i\,\}} \tag{15}$$

$$\boldsymbol{r_i \in R\,,\quad i \in N} \tag{16}$$

Here, $\boldsymbol{r_i}$ can be any of the possible instrument tasks $\boldsymbol{R}$.

3.1.6. Scheduling. Theoretically, workload can be partitioned according to the workers' performance. Workers, carrying identical instruments, in ideal conditions, should perform equally. But, in reality, many factors affect workers' performance. The workers work under severe conditions, so their instruments get damaged, i.e., they may still be operational, but their performance may get degraded. Therefore, some of the workers will perform more efficiently than others. For example, a less damaged *Laser Distance Meter* will need less measuring attempts to measure a distance than a more damaged one. Another factor is the distance between a ruler and a worker. Some workers will need a messenger in order to communicate with the ruler. So, the transmission time will be added to the overall performance time.

Hence, in order to perform task analysis and scheduling, the ruler should be able to evaluate the performance of each of the workers at the time of task scheduling or to ask every worker to self-estimate its performance for a particular task.

Therefore, the rulers should maintain statistics on the workers' health status. Thus, a ruler should perform external monitoring over the workers it maintains. External monitoring can be achieved proactively by having each worker send its *heartbeat* or *pulse* message regularly. A heartbeat message provides a summary of the state of a worker, thus including its health status (damage in %, is it still operational, etc.), its operational status (is it idle, percentage of complete work, etc.) and its coordinates.

ANTS workers, are designed to provide a specific set of services over their entire lifetime (cf. Section 2.1). Thus, the tasks they execute are generally fixed at some point during development, based on their instrument. Additionally, we have significant a priori knowledge regarding the tasks' characteristics and the current health status of the spacecraft units (damage in %). We derive this knowledge from the *heartbeat* messages and the restrictions the space environment imposes. Leveraging this information is essential for task-set feasibility analysis.

In general, to define formally the partition of a task that can be assigned to a worker, we need the overall remaining task $\boldsymbol{R}$, the current health status of that worker $\boldsymbol{H}$, the deadline time for the task $\boldsymbol{t}$, the current environmental conditions $\boldsymbol{C}$, the communication latency $\boldsymbol{l}$, and the communication bandwidth $\boldsymbol{b}$.

$$r_0 = 0,\ R = R - r_{i-1},\ r_i = f(R, H, t, C, l, b) \quad (17)$$

Here $f(\ )$ is a function possibly of more inputs, e.g., the number of idle *workers* carrying the appropriate instrument.

The *worker* performance self-estimation of specific sub-task r_i depends on the health status H and on the environmental conditions C. Thus,

$$Et = f(r_i, H, C) \quad (18)$$

where Et is the expected sub-task execution time.

4. Related Work and Conclusion

4.1. Related Work

A motion planning framework for a large number of autonomous robots is studied in [14]. Such framework enables robots to configure themselves adaptively into arbitrary environments (areas of arbitrary geometry). In this approach local-interaction techniques allow robots to converge to a uniform distribution by forming an equilateral triangle with their two neighbors. The basic idea is that a swarm of robots can be regarded as a liquid where robots behave like liquid particles that change their relative positions conforming to the shape of the container they occupy.

Similar research is described in [15] where physics properties of liquids, solids, and gases are used to maintain swarm formation. Other existing methods of team formation in swarms depend on a central controller [16, 17], but dependence on a single entity is prone to failure and it is not ideal for space-exploration missions where there is a high possibility of failure. Decentralized formation control methods are presented in [18, 19].

In [20], Chapelle et al. propose architecture for cooperative agents, where due to a satisfaction model and local signals, agents learn to select behaviors that are well adapted to their neighbor's activities, thus helping the system self-configure efficiently.

4.2. Conclusion and Future Work

In this paper, we have documented our formal approach to self-configuration of intelligent swarm systems. Formal models for the three self-configuration phases: *team formation*, *planning*, and *scheduling*, have been presented by using the NASA ANTS (Autonomous Nano Technology Swarm) prospective space exploration mission as a case study. The key point in our approach is the so-called "self-initiation for team formation". This is the first step of the team formation phase where an idle ruler (special ANTS spacecraft) automatically determines the need of a new team and starts the team formation procedure. Our formal model for team formation is based on the Partially Observable Markov Decision Processes and Discrete Time Markov Chains where we do not consider any central controller, but complex algorithms working on state-action relationships and considering a variety of probability values.

Our future research is concerned mainly with the implementation of the presented models. It is our intention to model the proposed self-configuring behavior with ASSL (Autonomic System Specification Language) [21] and generate an ANTS autonomic system incorporating the self-configuring properties (note that the ASSL framework has a built-in code generator). This will help us perform real experiments with the generated code and further refine our algorithms based on the obtained results.

References

[1] R. Murch, *Autonomic Computing: On Demand Series*, IBM Press, Prentice Hall, 2004.

[2] IBM Corporation, *An Architectural Blueprint for Autonomic Computing*, White Paper, 4th ed., June 2006.

[3] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff, "NASA's swarm missions: The challenge of building autonomous software", *IT Professional,* 6(5), Sep-Oct 2004, pp. 47-52.

[4] M. Hinchey, J. Rash, W. Truszkowski, C. Rouff, and R. Sterritt, "Autonomous and autonomic swarms", *Proceedings of 8th Biennial Conference on Real Time in Sweden (RTiS) 2005*, Skövde, Sweden, August 2005, pp. 65-73.

[5] G. Beni, J. Want, "Swarm intelligence", *Proceedings of 7th Annual Meeting of the Robotics Society of Japan*, RSJ Press, Tokyo, 1989, pp. 425-428.

[6] E. Bonabeau, G. Theraulaz, "Swarm smarts", *Scientific American*, March 2000, pp. 72-79.

[7] E. Vassev, M. Hinchey, and J. Paquet, "A Self-scheduling model for NASA swarm-based exploration missions using ASSL", *Proceedings of the 5th IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASe'08)*, IEEE Computer Society, 2008, pp. 54-64.

[8] K. P. Sycara, "Multiagent systems", *AI Magazine*, vol. 19(2), 1998, pp. 79-92.

[9] C. Lemaitre, C. B. Excelente, “Multi-agent organizational approach”, *Proceedings of the 2nd Ibero-american Workshop on Distributed Artificial Intelligence and Multi-Agent Systems*, Toledo, 1998.

[10] M. L. Littman, *Algorithms for Sequential Decision Making*, PhD Thesis, Department of Computer Science, Brown University, 1996.

[11] W. J. Ewens and G. R. Grant, “Stochastic processes (i): poisson processes and Markov chains”, Chapter in *Statistical methods in Bioinformatics*, 2nd edition, Springer, New York, 2005.

[12] R Achuthan, *A Formal Model for Object-Oriented Development of Real-Time Reactive Systems*. Ph.D. Thesis, Department of Computer Science, Concordia University, Montreal, Canada, 1995.

[13] D. S. Weld, “Recent advances in AI planning”, *AI Magazine*, 20(2), 1999, pp.93-123.

[14] Robotics Laboratory, *Decentralized Coordination for Robot Swarms - Adaptive Self-configuration*, School of Information Science, Japan Advanced Institute of Science and Technology, 2010, url: http://www.jaist.ac.jp/robot/research/swarm/top_swarm.html

[15] W. M. Spears, D. F. Spears, W. K. R. Heil, and S. Hettiarachchi, “An overview of physicomimetics”, *Proceedings of SAB International Workshop*, 2004.

[16] S. Zilinski, T. Koo, and S. Sastry, “Optimization-based formation reconfiguration planning for autonomous vehicles”, *Proceedings of IEEE International Conference on Robotics and Automation*, 2003.

[17] R. O. Saber, W. B. Dunbar and R. M. Murray, Cooperative Control of Multi-vehicle Systems Using Cost Graphs and Optimization, *Proceedings of the American Control Conference*, 2003.

[18] L. Chaimowicz, N. Michael and V. Kumar, “Controlling swarms of robots using interpolated implicit functions”, *Proceedings of International Conference on Robotics and Automation*, 2005.

[19] J. Fredslund and M. J. Mataric, “A general algorithm for robot formations using local sensing and minimal communication”, *IEEE Transactions on Robotics and Automation*, 18, 2002.

[20] J. Chapelle, O. Simonin, and J. Ferber, “How situated agents can learn to cooperate by monitoring their neighbors' satisfaction”, *Proceedings of the 15th European Conference on Artificial Intelligence*, 2002, pp. 68-72.

[21] E. Vassev, *ASSL: Autonomic System Specification Language - A Framework for Specification and Code Generation of Autonomic Systems*, LAP Lambert Academic Publishing, Germany, November 2009.