Multi-Agent System that Attains Longevity via Death*

Megan Olsen, Hava Siegelmann
University of Massachusetts Amherst
Computer Science Department
molsen@cs.umass.edu, hava@cs.umass.edu

Abstract

We propose a novel approach to self-regenerating systems which require continuous operation, such as security surveillance. For that aim we introduce HADES, a self-regenerating cooperative multi-agent system with local monitoring. When agents of HADES find local failures they repair them. However, in extreme cases repair may not be possible and irregular aggressive agents will multiply. These irregular agents may use all of the system's resources and thus take over the system. To optimize system longevity, we identify protocols for killing these irregular agents. Our primary contribution is a double communication protocol of alert and death signals among the agents, making the multi-agent system robust to failures and attacks.

1 Introduction

Continuously functioning systems have the desired property of being able to survive damage and regenerate as necessary. We propose a biologically-inspired, self-developing and regenerative system in which each agent contains the same protocols for behaving and decision making, but presents the possibility of having failures in its protocols. Such failures may occur due to environmental effects or naturally during the regeneration process. Agents that have acquired unrepaired failures and are no longer aiding the environment are considered irregular. If this malfunctioning escalates to the point that the irregular agents take over the system, it is beneficial for them to be taken away. Our solution to this problem involves enticing agent death via communication. We develop such communication protocols and test their robustness with this system, called HADES (Healing and Agent Death Encouraging Stability). The purpose of HADES is to examine the role of death in self-regenerative systems as means to achieve general longevity. Typically, an agent's death is irreversible and causes functional shortcomings [Klein *et al.*, 2003; Dellarocas *et al.*, 2000].

Our work is applicable to various distributed systems. Since HADES is based on a 3-dimensional structure, the systems of interest are not limited to 2-D. Consider a distributed system of computer-camera pairs, the architecture of a distributed sensor network that acts in response to queries. Queries can start at each node and send processes to the various computers to tune the cameras appropriately. If some of these processes start behaving irregularly they may choke the system, especially if the irregularity involves excessive querying. This may be initiated by a hostile environment or developed naturally with a series of untreated failures. Removing the irregular processes quickly is best in either case, and regenerating will occur automatically to replace the missing agents. The protocol must assure that most agents killed are irregular and that healthy agents can regenerate after death in the system.

Aggressive treatment of this kind is also appropriate in the case of cooperative robots. An irregular robot may begin to attack the other robots, or damage resources needed to attain the system's goal. In this case the other robots must have a way to end the irregular behavior so that the system goal may still be achieved. If the robot is unable to recognize that its behavior is unacceptable, there must be a secondary mechanism of citizenship. The other robots could therefore convince the problematic robot to power itself off. Due to the amount of damage the robot is capable of causing, the system is healthier without it. The assumption is that there are methods for outside help to repair the malfunctioning robot, or that the system goal is such that the other robots can still achieve it. Death in this case is a crucial step to preventing irreversible system damage.

These examples demonstrate the fine line required in the communication protocols, since irregular agents can also send death messages to healthy agents. Irregular and healthy agents may compete in regenerating to replace killed agents as well. In the rest of this paper we outline a general framework and describe the citizenship and communication protocols necessary for achieving longevity via death.

2 Previous Work

For a multi-agent system to be working continuously it must adapt on-line to changes in the environment and internal failures. Diagnosis of a problem is a key requirement, as well as having plans to react to problems [Hamscher *et al.*, 1992]. Various frameworks exist for diagnosis in multi-agent systems, including domain independent diagnosis [Horling *et al.*,

*Supported by NSF grant and DHS Fellowship

2000]. Diagnosis for pre- and post-failure analysis for causal tasks can allow the system to both prevent failure and recover from it [Toyama and Hager, 1997]. Fault tolerance can allow a system to recover from agents that die [Kumar *et al.*, 2000]. Our system, is built on the principle of regenerative agents where all agents follow the same basic life protocol. Hence, for HADES diagnosis consists of four steps. The first is the agent's ability to determine whether its own life protocol has been damaged and then to restore the healthy one. The second step of diagnosis is for agents to note that their neighbors are irregular, which causes the third step of communication with neighboring agents. These messages are passed along and remain active for the relevant time period. The third step is designed for an agent that is so damaged that it cannot diagnose and repair itself. If this agent receives enough messages requesting it to die it must eventually do so to keep the system healthy. In the fourth step the irregular agent maintains some level of citizenship and therefore before killing itself communicates its decision via signaling, causing neighboring agents to raise their alert level.

Approaches currently exist to react to agent death in a multi-agent system following survivalist or citizen concepts [Klein *et al.*, 2003; Dellarocas *et al.*, 2000; Smith, 1980]. Both approaches are aimed at increasing the adaptability of the system to minimize the impact of agents' death on the overall functionality of the system. The citizen approach utilizes an external system that is alerted when an agent dies and it then reallocates tasks so that the overall system continues to function correctly [Klein *et al.*, 2003]. The survivalist approach requires all agents to be capable of dealing with all problems internally [Klein *et al.*, 2003; Dellarocas *et al.*, 2000; Smith, 1980]. Each agent must therefore be built with a great deal of error handling for any problem that might occur [Dellarocas *et al.*, 2000]. The survivalist concept is part of the basic framework of the CNet protocol [Smith, 1980]. Our system naturally combines principles of both the survivalist and the citizenship approaches, and adds the communication protocols to handle irregular agents in a novel manner.

Since our system is regenerative, agents will regenerate automatically; the external communication process to alert for irregular agents follows citizenship principles. Regenerative systems have been investigated for at least the last 50 years, and include minimalist ideas on what is needed for a system to regrow [von Neumann, 1966], the use of chemicals to control the differentiation and growth [Miller, 2004], the ability to use gene structures for regeneration [Meinhardt and Gierer, 1980], and regenerative agents [Fedoruk and Deters, 2002], to mention the basics. Our main addition to these systems is communication protocols as well.

HADES investigates removing malicious agents. As Klein et al points out, killing a malicious agent may be difficult as agents usually are not given the ability to directly kill each other. However, it may be beneficial to the system overall for problematic agents to die [Klein *et al.*, 2003]. We study the option in which messages sent to an agent can only convince it to die, thus alleviating the problem of agents directly killing others. This approach helps protect against irregular agents sending death messages to healthy ones as well.

3 The System

HADES is a cooperative multi-agent system on a 3-D lattice. It is arbitrarily bounded to a size of 40x40x20, therefore the total number of functioning agents cannot exceed the healthy equilibrium point of 4000. The system is created from a single agent that generates new agents until this equilibrium is reached. Agents can replicate up to 70 generations before it is considered damaging, as each replication carries a possibility of damage to the agent. Each agent has a goal to stay healthy, therefore it will not replicate after 70 generations. Since the agents share a system goal of keeping the equilibrium, replication is a priority to all agents. These capabilities form the basis of the system's self-regenerating property.

We consider each agent to have life protocols that control their actions and define their current state. The communication among agents occurs differently for each signal type. A signal is emitted into the environment to be diffused equally in all directions for presence signals. Death signals travel a specific distance after being emitted, but do not diffuse.

4 Application Details

4.1 Healthy Agents

Life Protocols

There are four internal life protocols modeled in each healthy agent. They control both adherence to goals as well as actions taken. The first protocol is replication, which controls the frequency that an agent generates new agents and enables self-testing prior to replication. The second protocol called suppression controls the replication protocol by stopping replication if damage is sensed. Repair is the third protocol, which fixes any damage in the individual agent. The last protocol enables an agent to induce self-death. Details of these protocols will follow.

Goals

Healthy agents have multiple goals. The system level goal is to maintain the system equilibrium by replicating when necessary. Personal goals include: maintaining space, maintaining self-health, maintaining system health, and maintaining the shortest possible distance between itself and the center of the system. Each of these goals have specific motivations and interactions.

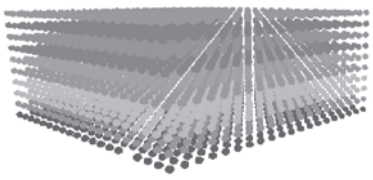

Figure 1: 3D view of the system at its stable state. Shading is used to make the rows more visible.

A healthy agent maintains the shortest possible distance between itself and the center of the system so that the agents will form a cohesive unit. Due to this goal, the system develops in a sphere and is only forced into a rectangle due to the tight boundaries on the system.

Healthy agents require empty space around them, as can be seen in Figure 1. It is impossible for them to impede on another agent's space unless the other agent dies. If an agent appears in the space directly next to an agent, the agent will attempt to move away if it can do so without encroaching on another agent's space.

Agents maintain their own health by monitoring any damage that occurs to their protocols. If the repair mechanism continuously fails, the agent will recognize that it may not be functioning correctly and will kill itself so that it does not damage the system. The agent is therefore preserving the system health by preserving its own health.

The agent also maintains system health by guaranteeing that it does not replicate too frequently. The replication protocol controls replication rate but if it is damaged this rate will increase, implying lack of self-testing prior to regenerating. An undamaged suppression protocol will halt this replication. An agent will also maintain system health by sending kill signals to irregular agents, as will be discussed later.

Actions

After the initial development stage is completed the system keeps the equilibrium by replacing agents that die, unless irregular agents keep healthy agents from replicating by entering their natural surrounding space. In our simulation we have chosen for all healthy agents to make their decisions before the unhealthy agents, and while decisions are made sequentially, the actions take effect at once. At each step of the simulation, an agent performs one action based on the environment:

1. Repair: occurs if the agent is damaged.
2. Death: occurs by three mechanisms, and is self-induced. Death can occur when an agent has been unable to repair its life protocols. It can also occur with a probability of 0.0024 to include other causes of death such as age. The third mechanisms is via kill signals sent by surrounding agents.
3. Replication: occurs with a probability of 0.0025, if there is available space and suppression is not activated.
4. Movement: occurs if an agent cannot replicate but there is an available adjacent space with a higher concentration of presence signals than its current spot, representing a space closer to the center of mass.

4.2 Presence Signals

Agents emit a presence signal into the environment that is diffused equally in all directions for a specific radius and speed. If the signal is strong enough to last more than one unit away, it will move by one unit each time step until it has reached its limit. Therefore, the closer areas have a stronger chemical presence as the previous signal will linger.

Presence signals are used by agents to determine their proximity to other agents, as well as the direction of the center of mass. If an agent moves or dies, the signal will slowly decrease toward the original spot at the same rate that it diffused out since it is no longer being emitted from that location. The change is therefore not immediately obvious to other agents.

4.3 Irregular Agents

If all four life protocols of a healthy agent have been damaged the agent cannot regulate itself, and will ignore all goals. The defective agent will continue to replicate, spreading its damaged life protocol to its daughters, creating a cluster of problematic agents. The probability of creating an irregular agent is incredibly low, since the processes have to be ruined in a particular order: repair damaged first, then death, then suppression, and last replication. The probability of each individual process being ruined is simulated as 0.001, and the choice of which protocol to damage is random. The probability of the damage occurring in the correct order is therefore extremely small. However, only one irregular agent is necessary for the system's behavior to change.

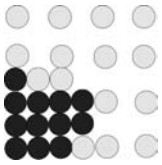

Figure 2: Irregular agents (shown in black) take over the system quickly by pushing the healthy agents out of the way and ignoring space.

Irregular agents replicate without respect to the amount of space available or the diffused signals. Although the equilibrium of healthy agents is 4000, if irregular agents completely take over the system they can grow to a size of 32,000 (Figure 2). If there is no room in any adjacent spots when an irregular agent replicates, it "pushes" a neighboring healthy agent into one of its buffer spaces. If this push causes the healthy agent to be directly next to another agent, neither it nor the adjacent agent will be able to replicate. If the agent is instead pushed into another agent, it will be considered an intruder. As the irregular agents form a cluster, they will continue to exert this physical pressure on the same area. This process may continue until the system has no more healthy agents. We propose taking advantage of this style of growth to design a communication protocol that will inhibit the problematic agents and save the healthy agents.

4.4 Controlling Irregular Growth via Communication

Our communication protocol allows agents to send signals to convince other agents to induce self-death. The initial signal is known as "Please Die," and is sent by a healthy agent that senses irregularity around itself. This irregularity is represented by an invasion of space, although other systems could incorporate different representations. The invading agent will either be an unhealthy agent, or a healthy agent that has been pushed by an unhealthy agent and therefore forced to move. This signal initiates the inter-agent communication, and therefore has a low strength that was tested and chosen for optimality. The original "Please Die" signal will only reach those agents in the original buffer space of the sending agent. There is therefore a high probability that this signal will originally only be intercepted by irregular agents.

Although this mechanism allows the agent to eventually convince close problematic agents to die, it is not enough to save the entire system (Figure 3). We therefore propose a double signal system inspired by agent trust [Ramchurn *et al.*, 2004].

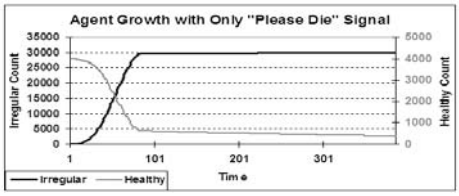

Figure 3: If only the "Please Die" signal is used and not the "I Died" signal, there is a slower exponential growth of irregular agents until they reach a total of 30,000. The number of healthy agents have already decreased to under 1000 after only 80 steps, and after 300 steps is still slowly decreasing to under 400 agents. The same result occurs when the signal limit for "Please Die" signals is either 3, 2, or 1. The final ratio of healthy to irregular agents is 0.01.

The "I Died" signal is sent by an agent when it is dying, to alert neighboring agents that they should consider dying as well. The signal is twice the strength of the "Please Die" signal, therefore affecting any agents within the buffer space or neighboring agents that are respecting the buffer space. Since the structure of irregular agents is a close cluster, this type of signal is shown to be very effective in eliminating the majority of them due to their close proximity, while not affecting as many of the further apart healthy agents due to their spacing. Recent descendants are likely to be close since irregular agents do not move, and they are therefore likely to have a high level of trust. Irregular agents only send the "I Died" signal, and therefore they only send one message during their lifetime. It is therefore impossible for a single irregular agent to flood its healthy neighbors with death signals.

The propagation of these signals is different than the presence signals. As it is not diffusion, the signal is the same to each agent it meets as opposed to being stronger to the closer agent. We chose to represent the signal this way to facilitate more efficient signal passing, therefore equally encouraging all agents to die that receive the signal.

An agent decides to die due to signals when the number of either type of signal it has received is above its limit. This limit can either be the same or different for each type of agent, can be different for each signal type, and can also change over time. The limit is determined by the trust an individual agent has toward the agents surrounding it. The higher the level of trust, the lower the limit will be. Our results tested different levels of trust to determine the optimum.

5 Results

Eight main scenarios were tested with our communication protocol. We began by testing four scenarios to determine the optimum signal limit. These scenarios included using a limit of 2 for both signal types, a limit of 3 for both signal types, and limit of 10 for both signal types, and a limit of 3 for "Please Die" and a limit of 2 for "I Died" (Figure 4). The optimum was found to be a signal limit of 2 for each signal

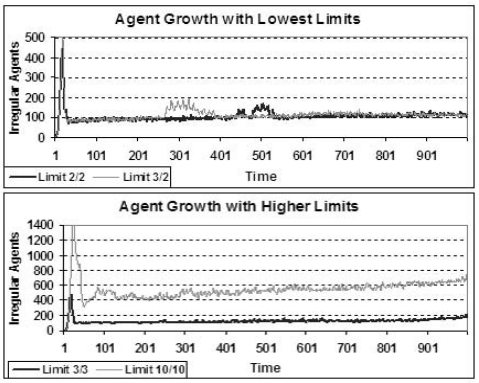

Figure 4: With a threat limit of 2 for both signal types, the number of irregular agents peak at 446, and after 1000 ticks is at 105 (3.9% of agents in system). There are 2588 healthy agents in the system at this point, giving a ratio of 24.65 for healthy to irregular. With a limit of 3 for the "Please Die" signal and a signal limit of 2 for the "I Died" signal, the highest irregular agent count is slightly lower than the previous one, at 435. The final number of irregular agents is 108 (4.13% of agents), giving a ratio of 23.21 healthy agents to each irregular one. A threat limit of 3 for both signal types has a peak of 481 irregular agents. The number of irregular agents ends at 170 (7.21% of agents), giving a final healthy to irregular ratio of 17.37. With an increased signal limit of 10 for both signal types, the irregular agents grow to 1459 (97.56% of agents), almost three times as large as they did with the other limits, with a final ratio of 0.01 after 1000 ticks.

type, so all other tests utilized this limit. Tests were done with no "I Died" signal to verify its necessity (Figure 3). An increasing signal limit was used to represent decreasing trust among agents, where the limit for each signal separately starts at 2 and increases by 2 every 100 time steps (Figure 8). Delayed signaling is a crucial property to test to examine how the protocols work if they are initiated late. Signaling begins when a specific number of irregular agents exist, tested with delays of 500, 1,000, 10,000, and 25,000 (Figure 6). A combined signal scenario where the two signals were regarded as the same message by the receiving agent was also tested; in this case, once 2 messages are received it dies even if each message is of a different type (Figure 7).

All scenarios were run for 1000 time ticks as a relative equilibrium had been reached by the best cases, and the poor cases were at a point of no possible improvement. The ratios of healthy to irregular agents were above 9 for all of the 7 scenarios with more healthy agents than irregular agents, which is not great. The best three, however, had ratios of over 20, giving the healthy agents a much better survival chance. For the worst three scenarios, the irregular agents have successfully comprised the system so that it cannot recover, with ratios around 0.03 healthy agents per each irregular agent. For the successful cases the ratio gives the healthy agents enough space to replicate away from the irregular agents since signaling is still occurring, and enables them to kill irregular agents quicker than originally. This quicker kill is possible because most irregular agents have already received some signals and are therefore closer to death.

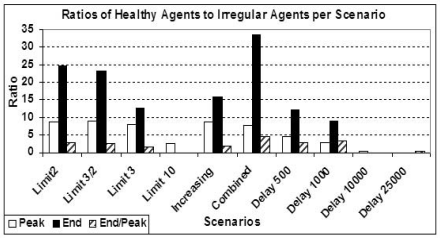

Figure 5: The ratio of healthy agents to irregular agents can change dramatically by scenario. A high ratio is ideal, as our goal is 100% healthy agents and 0% irregular agents.

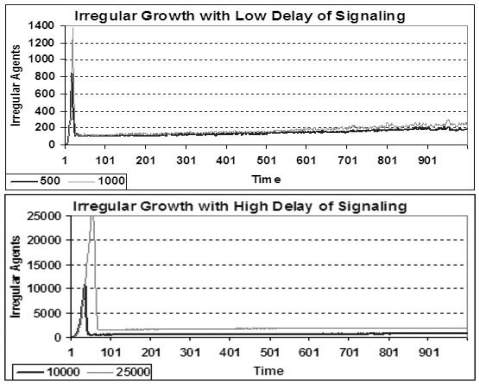

Figure 6: The "Please Die" signal can be stalled in the beginning so that it is not used until there are a certain number of irregular agents. The number of irregular agents before signaling starts only slightly affects the final amount of stable irregular agents for most delays, with around 200 irregular agents for a delay until 500 or 1000 agents and 600 for a delay of 10,000, although they comprise different percentages of the total system (7.50% for 500, 9.98% for 1000, 97.76% for 10,000). For a delay of 25000 problematic agents, we have an increase to 1900 problematic agents (96.39% of agents) and much less stability. All increases in the amount of delay significantly affects the number of irregular agents in the beginning leap. The ratios of healthy to irregular agents for each delay type (in order) are: 12.33, 9.02, 0.02, 0.04.

For the majority of the scenarios, the ratio of healthy agents to irregular agents was higher at the end than it was during the peak of irregular agents. The obvious worst scenarios are when the ratio decreased, as with a signal limit of 10, signals delayed until 10,000 irregular agents exist, and signals delayed until 25,000 irregular agents exist (Figure 5). Overall, the scenario of the combined limit performed the best, with a ratio of 7.82 at the peak and 33.66 at the end. The regular limit of 2 was the second best with a final ratio of 24.65, although its peak ratio was better at 8.67. The joint limit of 3 and 2 was third with the best peak ratio (8.97) and a final ratio of 23.21. An increasing limit took fourth with ratios of 7.99 and 12.86, the delay of 500 came in fifth with ratios of 4.55 and 12.33, and the delay of 1000 came in sixth with ratios of 2.81 and 9.02. The best improvement in ratios from peak to end was the delay of 1000.

Examining the actual number of irregular and healthy

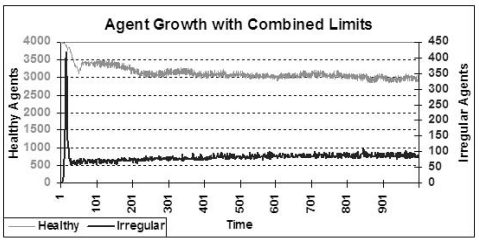

Figure 7: When the signals are interpreted identically by the agents so that once the total number of signals reaches the limit it dies, the results for irregular agents are very similar to when there is a limit of 2 for either signal as in Figure 4, except that there are more healthy agents in the end (2996). The final ratio of healthy to irregular is therefore much better, at 33.66.

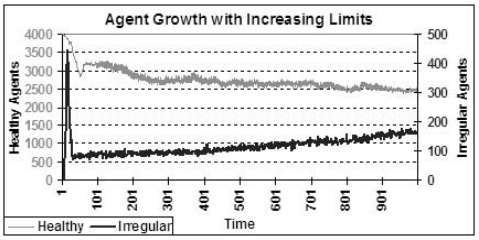

Figure 8: When the number of signals required for death increases over time for both signals by starting at 2 and increasing by 2 after every 100 steps, the irregular agent count slowly increases. After 1000 steps it is at 159 (5.95% of the system), with a healthy to irregular agent ratio of 15.80. The number of healthy agents stays around 2500.

agents at the end of the run yields the same order of optimality. The best cases are the combined limit with a healthy agent count of 2996 and irregular count of 89 (2.88% of all agents, Figure 7), followed by the limit of 2 with 2588 healthy agents and 105 irregular agents (3.90% of all agents, Figure 4). The limit of 3 and 2 was again a close third with a healthy agent count of and an irregular agent count of 108 (4.13% of agents, Figure 4), followed by the increasing limit with 2449 healthy agents and 155 irregular agents (5.95% of all agents, Figure 8). The limit of 3 had 2187 healthy agents and 170 irregular agents (7.21% of agents, Figure 4), the delay of 500 had 2367 healthy agents and 192 irregular agents (7.50% of agents), and the delay of 1000 had 2021 healthy agents ad 224 irregular agents (9.98% of all agents, Figure 6). The worst three scenarios were the limit of 10 with 18 healthy agents and 720 irregular agents (97.56% of all agents, Figure 4), a delay of 10,000 agents with 15 healthy agents and 656 irregular agents (97.76% of agents), and a delay of 25,000 agents with 72 healthy agents and 1925 irregular agents (96.39% of agents, Figure 6).

The obvious best case is the combined limit, as an agent can either get 2 of a specific type of signal before dying or it can receive 1 signal of each type. This scenario had the highest ratio at the end (33.66) as well as the highest number of healthy agents (2996) and lowest number and percentage of irregular agents (89, 2.88%). Since the limit of 2 and the limit

of 3 for "Please Die" and 2 for "I Died" were the next best and the limit of 10 was one of the worst (Figure 4), it is apparent that the key is to have a low limit overall. It is also important to start signaling as early as possible, as can be seen by the poor performance of the high delays (ratio of 0.02, Figure 6). The system cannot recover from the destruction caused by a large delay, as the signaling is not strong enough to kill over three times as many irregular agents at the beginning.

The healthy agents do not return to their initial equilibrium due to a combination of random death, a low replication probability, and spacing.

6 Conclusions

Agent death has been shown to be useful in keeping the health of systems that are prone to damage when repair is unavailable. A novelty found in our simulation is that it is not enough that agents will send "Please Die" messages, but an agent that is going to die must announce its death to the environment as a way of transferring the alert for irregularity to its neighbors. Only through this mechanism can the system rid itself of an entire cluster of irregular agents. Our results have shown that although a low signal threshold for unhealthy agents is ideal, communication can still be successful even if there are increasing limits. However, a delay or relatively high starting limit will still compromise the healthy agents in the system despite the fact that they significantly decrease the number of irregular agents.

Our solution is designed for general multi-agent systems as long as citizenship and trust is introduced where the agents share the goal of keeping the system functioning. The mechanism used to determine that an irregular agent is being invasive will also change for different systems, as well as the specific limits for each signal. For example, if introduced with defective agents the distributed camera system mentioned previously may suffer from too many processes working incorrectly, as they will tie up resources. By having a way for other processes to communicate with the damaged processes to convince them to halt, the system may be able to correct itself. The robot case from the introduction will react similarly, with the irregular robots shutting down.

This algorithm has shown success, but can be improved. The best case is great with a final of 2.88% irregular agents, as it shows that the healthy agents are in a high majority. Only slight changes may be necessary to decrease it to the target of 0%. We therefore propose three techniques to improve this percentage that will be examined in future work, all of which will use the low signal limit and will change other aspects to find a better overall protocol. The first technique is to allow healthy agents to replicate approximately 2 to 3 times more frequently under specific conditions such as a low number of neighbors. This technique will combat the problem of healthy agents dying from random death too frequently when trying to rebuild. The second technique is to modify the decision to move so that healthy agents will spread out to about 3 times their current spacing when there are low numbers, allowing interior agents more opportunities to replicate. This technique may fail as irregular agents will be able to replicate 6 times as much before pushing a healthy agent and causing signaling. The third technique is to change the signal interpretation so that agents trust signals from specific agents more than others. This would give a lower limit in some cases, enabling possibly 25% more irregular agents to be killed in the same amount of time. A combination of these techniques may also be beneficial.

References

[Dellarocas *et al.*, 2000] C. Dellarocas, M. Klein, and J. A. Rodriguez-Aguilar. An exception-handling architecture for open electronic marketplaces of contract net software agents. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 225–232, Minneapolis, 2000.

[Fedoruk and Deters, 2002] Alan Fedoruk and Ralph Deters. Improving fault-tolerance by replicating agents. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 737–744, New York, NY, USA, 2002. ACM Press.

[Hamscher *et al.*, 1992] W. Hamscher, L. Console, and J. de Kleer. *Readings in Model-Based Diagnosis*. Morgan Kaufmann Publishers Inc., 1992.

[Horling *et al.*, 2000] B. Horling, V. Lesser, R. Vincent, A. Bazzan, and P. Xuan. Diagnosis as an integral part of multi-agent adaptability. *Proceedings of DARPA Information Survivability Conference and Exposition*, pages 211 – 21, January 2000.

[Klein *et al.*, 2003] M. Klein, J. Rodriguez-Aguilar, and C. Dellarocas. Using domain-independent exception handling services to enable robust open multi-agent systems: The case of agent death. *Autonomous Agents and Multi-Agent Systems*, 7:179–189, 2003.

[Kumar *et al.*, 2000] Sanjeev Kumar, Philip R. Cohen, and Hector J. Levesque. The adaptive agent architecture: Achieving fault-tolerance using persistent broker teams. *icmas*, 00:0159, 2000.

[Meinhardt and Gierer, 1980] H. Meinhardt and A. Gierer. Generation and regeneration of sequence of structures during morphogenesis. *Journal of Theoretical Biology*, 85(3):429–50, August 1980.

[Miller, 2004] J. Miller. Evolving a self-repairing, self-regulating, french flag organism. In *Proceedings of GECCO*, 2004.

[Ramchurn *et al.*, 2004] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25, 2004.

[Smith, 1980] R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE TRANSACTIONS ON COMPUTERS*, C-29(12), December 1980.

[Toyama and Hager, 1997] K. Toyama and G. D. Hager. If at first you don't succeed... In *Proceedings of the 14th National Conference on Artificial Intelligence*, 1997.

[von Neumann, 1966] J. von Neumann. *Theory of Self-reproducing Automata*. University of Illinois Press, Urbana, 1966.