Keywords: Fault Tolerance; Medicine and Science; Multiagent Systems; Biology and Genetics

Abstract
In cancer biology, it is known that cancer cells can disappear without therapy, but not how. We propose that cells communicate such that primarily malfunctioning cells (tumors) die. We also propose that this same communication can be used as inspiration for a fault-tolerance mechanism for multi-agent and distributed systems to remove faulty agents using only local information. I examine the communication protocols necessary for removing these faults in both systems.

1. Introduction
My thesis is a multi-disciplinary approach to investigate fault tolerance for cooperative agent systems that have some equivalent of self-replication and self-death. Utilizing biologically-inspired mechanisms, I increase multi-agent system robustness for faulty agents when it is unknown exactly which agent is malfunctioning. In addition, I apply these techniques to the related systems of biological cells in a tissue being invaded by cancer, and neurons affected by neurodegenerative diseases.

For a multi-agent system, it is important to determine new ways to increase robustness as otherwise it cannot be guaranteed to function in all situations and thus cannot be relied upon. Robustness of a system allows agents to recover from errors and function continuously, an increasingly important trait as agent systems are deployed in real world scenarios such as sensor networks or surveillance systems where faulty or malicious nodes could disrupt application performance. To achieve robustness, there must either be prevention of all errors, or a technique for recovering from errors after they have occurred.

For a multi-agent system to function continuously it must adapt on-line to failures. The problem must be diagnosed and a plan provided [1]. Various frameworks exist for diagnosis in multi-agent systems, including domain independent diagnosis where an agent should also be able to determine a new plan if its expectations are not met [2]. Diagnosis for pre- and post-failure analysis for causal tasks can allow the system to both prevent a failure and recover from it. It is argued that post-failure protocols are less domain dependent and thus more crucial for the design of robust systems [3]. I propose a real-time inter-agent messaging scheme to find and remove agents that have failed. These robustness protocols enable local communication among agents to remove malfunctioning agents. My work suggests that by using this local communication it can be possible to fix the problem without first diagnosing exactly which agents are malfunctioning, thus removing the need to decide who is wrong before reacting.

One potential scenario that could benefit from our robustness protocols is a sensor network that must be robust to node failure and corruption [4]. Sensor nodes are typically deployed in multi-hop mesh networks whose links provide the spatial structure for our algorithms. In these networks nodes have only a local view of the system, and while it may be possible to detect that a fault has occurred, it may not be possible to know whether faulty data was caused by the source node or an intermediate node in the network path. Many sensor platforms support remote rebooting or reloading of application code, providing an analogue to the self-death and rebirth in our algorithm [5]. Our robustness protocol could thus provide a new automated way to effectively repair systems such as this, without requiring any global information or knowledge of exactly which nodes are faulty.

Cancer can be examined similarly, as this problem is analogous to events that occur during cancer development if we analyze a biological cell as we would analyze an agent — a correctly functioning agent corresponds to a healthy cell while a faulty agent represents a tumor. There is still much work to be done in understanding interactions between normal cells and cancerous cells within a tissue, and how their interactions affect the growth of a tumor. Mathematical and computational approaches for studying cancer have been primarily focused on the chemistry of cancer, involving breaking the system into its smallest parts. Both mathematical models [6-8] and agent-based simulations [9,10] tend to delve only into the cancer cell’s movement and creation, with little regard to the surrounding normal cells. However, these surrounding cells play a vital role in the growth and possible containment of cancer cells. Generally these models are based on the ideas for the future of cancer research put forth by Hanahan and Weinberg, such that research should be based on the set of
inspired by cancer biology, and provides robustness for a analysis to determine vulnerabilities uses both static and system after a malfunction has occurred. One way to obtain such post-failure robustness for agents is via self-repairing. understanding of how those interactions play a role in the development of these diseases, specifically Alzheimer’s disease. To this end, we will examine both the chemical and electrical communication between neurons in a similar way to the agent and cancer cell situations, to create hypotheses of why a lesion in the brain may grow over time.

Neurodegenerative diseases affect neurons in a way that leads to neuron death, but complete understanding of how and why these diseases affect neurons is unknown. This problem is similar to the cancer cell and agent problem with respect to the interactions between neurons, and through our analysis we would like to build a better understanding of how those interactions play a role in the development of these diseases, specifically Alzheimer’s disease. To this end, we will examine both the chemical and electrical communication between neurons in a similar way to the agent and cancer cell situations, to create hypotheses of why a lesion in the brain may grow over time.

Although these three systems may at first seem unrelated, we can ask similar questions and analyze them using similar techniques. Overall, I make two main hypotheses:

1. The death/removal of an entity in a system, whether biological or artificial, can be influenced by neighbors in such a way that it can be utilized as a mechanism to remove irregularity in the system.
2. The type of death examined in the first hypothesis may explain unwanted cell death in neurodegenerative diseases, and desired cell death in cancer.

2. Related Work

The approach for fault tolerance in agent systems is inspired by cancer biology, and provides robustness for a system after a malfunction has occurred. One way to obtain such post-failure robustness for agents is via self-repairing mechanisms. Software components can self-monitor to determine vulnerabilities and thus remove them [15]. The analysis to determine vulnerabilities uses both static and dynamic techniques, and software validation techniques can be used to identify causes of vulnerabilities [15]. Wireless sensor networks are also being designed with self-healing capabilities inspired by immunology. For example, [16] mimics B-cells in the immune system with scripts on monitor nodes that can find failure on sensor nodes by examining the statistical properties of the sensor readings, and then counteract those failures. This system can thus adapt to changes in the network [16]. My focus is on self-regenerating systems, which enable either agents or the system itself to create new agents. In biology, this is used to both recover from dying cells as well as develop and expand the system [17]. In synthetic agent systems, agents can replicate to create agent clusters that will make joint decisions, thus improving fault tolerance [18]. A simple robot has also been composed of building blocks that can generate a new copy of itself by assembling additional blocks [19], and self-regeneration has also been used for general development [20]. In this latter system, artificial chemicals are used to control movement of cells on a lattice, and grouped with the regenerative abilities the system is able to “self-repair” by retaining a specific shape despite dying cells [20]. This use of the term self-repair is to be differentiated from the more fundamental definition described above and as is used in biology: the cellular level repair. One example of cellular level repair is when a self-replicating cell or agent uses error correction to make changes to itself to ensure that it grows into the proper form [21]. The system self-repair is but one consequence of the cellular repair.

Although we utilize agent death to enable our post-failure robustness, it is generally believed that the death of an agent may be harmful in a multi-agent system [22]. Various fault tolerant algorithms exist to react to agent death following either the survivalist or citizen concepts [23,24]. Both approaches are aimed at increasing the adaptability of the system and at minimizing loss of its overall functionality due to agent death. The citizen approach utilizes an external system that is alerted when an agent dies and then reallocates tasks so that the overall system continues to function correctly [24]. The survivalist approach requires each agent to be capable of dealing with all problems as an individual following a prepared set of actions for each specific problem [23,24].

Our system HADES entices death via the use of ranged communication protocols, neither requiring agents to know exactly which other agents are faulty, nor requiring a global monitor [25,26]. Since irregularity is not the norm in most systems, instead of having self-death as a default that occurs if an override is not received [27], our system messages are sent when irregularity has occurred and only to agents in that area. Thus, we decrease the overall messages that must be sent. We improve on the approach in [28] as well by giving more flexibility to agent movement, as we do not require spare agents to be kept to the side until needed, but create them as necessary. HADES
utilizes similar mechanisms overall, such as internal repair, regeneration, and death, but allows the agents to decide when to die themselves. We are thus able to maintain our system by suggesting death to surrounding agents and over time removing all faulty ones. Overall, HADES uses a unique combination of death, repair, regeneration, movement, and communication to maintain itself by giving agents the utmost control over these actions.

When examining cancer systems, our related solution provides a possible explanation for already observed but previously unexplained behavior: cancer cells being removed without treatment, as well as message passing between cells for unknown reasons. We assume that even highly mutated cells maintain residual apoptotic (self-death) abilities, and that those skills require communication with other cells in the system, both by signal emission on viability status (alive or dead) and the compliance with external apoptotic commands. These assumptions are supported by increasing evidence on the significance of the tumor environment in cancer occurrence, with several data pointing directly to our assumptions: the role of high mobility group box 1 (HMBOX1) protein in reporting cancer cell death to the immune system as an essential part of tumor death by chemotherapy agents [12], and findings showing that AP2L/TRAIL (tumor-necrosis-factor-related apoptosis induced ligand) binding to "death receptors" DR4 and DR5 induces selective tumor cell apoptosis via the external apoptosis pathway [29,30].

In the neurodegenerative case, our proposed mechanism could help explain why certain neurons undergo the shrinking and eventual death that is common in diseases such as Alzheimer’s. When there is a lesion in the brain, the resulting hole can continue to grow. However, there is research suggesting that communication between neurons affects this neuron death [31]. We propose that a mechanism similar to our communication scheme can account for this affect, and thus we examine the role of chemical and electrical connections between neurons in the growth of holes in the brain.

3. Methods

My solutions will be tested in a system capable of self-replication (cloning), self-death (removal), malfunctioning that may be passed to “children,” and message passing. Assuming that replication, death, and message passing are available, I propose that a “robustness protocol” can be utilized to support our hypothesis of using entity death to enable continuous functioning of our systems. In each system there are interactions between entities (whether agents or cells) that could indicate that an error has occurred, or that a neighboring entity is malfunctioning. Through preliminary simulations I determined that two specific types of messages can be sent among entities that when combined could potentially remove all malfunctioning entities without adversely affecting the normal entities. One message is sent when a neighbor is behaving abnormally; this can be determined based on interactions between entities, such as one neighbor spatially violating another. We call this message “Please Die,” as the violated entity would like the violating entity to leave (Figure 1a). Once an entity has received enough of these messages over time (as defined by a threshold), it will choose to die. In essence, neighbors are convincing annoying neighbors to die. However, preliminary results showed that this message alone was not enough, as the middle surrounded entities in a cluster of entities would not be easily affected.

Thus, a second message called “I’m Dying” was developed to aid in propagation of messages inside a cluster of entities. This message is sent by entities just before they die from the robustness protocol messages (Figure 1b). Similar to the first message type, it can also cause death in entities who have received enough of them to cross a threshold. The combination of these two message types gives both a meaningful start to the message passing, and a reasonable way to propagate messages throughout an area of malfunctioning agents.

I analyze my proposed solution both computationally and mathematically, to determine how well this technique will work for each of my relevant systems. The computational model and simulation will be defined by multiple sets of parameters: agent parameters, and robustness parameters. The agents/cells/neurons are controlled probabilistically, so their parameters determine their rate of replicating, dying naturally (not due to robustness mechanisms), and malfunctioning. The robustness parameters relate to the thresholds for when an entity will die due to the message passing, how far a message will travel, and how quickly the potency of the message will fade as it travels.

4. Results

I have primarily tested this system using a simulation of self-replicating, self-dying, and self-organizing agents in a
Figure 2. When agents may ignore robustness messages (diamond) or fail to send I’m Dying messages before dying (square), we still see high success rates. For up to 60% failure rates for either failure, we still see 100% success in systems where the correctly functioning robustness protocol also succeeded 100% of the time. The system cannot handle either failure occurring more than 80% of the time. However, these results show that my robustness mechanism is robust to failures within itself. Note the modified x-axis (percent protocol failure).

3D grid. Correctly functioning agents will only be created in the neighborhood of another correctly functioning agent if there is room for it, and will only occur when the probability of replication allows it. Malfunctioning agents will however be created at a more frequent rate once a single agent has begun to malfunction. The system is tested starting at the point where an agent begins to malfunction.

I tested HADES with hundreds of agent parameters in combination with hundreds of robustness protocol parameter sets. Each set of parameters is tested 19 times with different pseudo-random number generator seeds to mimic different potential behaviors given the same parameters. This is necessary to the results are due to the parameters tested, and are not merely artifacts of the chosen random number generator seeds.

I focus on worst case scenario parameters, in which faulty agents malfunction in such a way as to increase their likelihood of overrunning the system. I consider success to be represented by removing all malfunctioning agents without removing the majority of correctly functioning agents in the majority of experiments run for a specific set of parameters. My most recent results [13] show that in the majority of cases we are able to remove the malfunctioning agents completely without also removing a majority of the correctly functioning agents. This can be accomplished whether the rate of malfunctioning agent creation is close to the rate of normal agent creation or significantly higher. I include in my experiments parameter sets in which malfunctioning agents need to receive many more Please Die and I’m Dying messages to die. This demonstrates that my robustness mechanism is able to effectively prevent system failure in a range of scenarios, even when faulty agents are partially resistant to the recovery protocols.

In addition, there is a possibility of the robustness protocols failing. In this case, either malfunctioning agents may not always send I’m Dying messages before they die from the protocols, or they may sometimes ignore received messages. Even when malfunctioning agents fail to acknowledge my robustness mechanisms the majority of the time, we can still succeed in removing them from the system before all normal agents are also destroyed (Fig. 2).

These results are from running the simulation with biologically plausible parameters. Thus, they also indicate success for the cancer system, in that the normal cells are able to remove cancer cells but still survive themselves. The types of growth that we see in our system mimic what would be expected in breast cancer. I have also analyzed the basic system mathematically using differential equations and cellular automata. These forms of analysis have been most useful for the cancer model, indicating that the growth shown in my model is similar to what would be expected in an actual cancer system.

The neurodegenerative disease model is not yet finished, as it was only this year. I plan to have at least preliminary results and conclusions on this aspect of the thesis in time to present at the conference.

5. Conclusions

Overall, my research will show that a specific set of techniques rooted in entity death can be applied successfully in multiple biological fields to explain natural phenomena, as well as to increase the robustness of multi-agent systems. My thesis thus opens a new approach to examining these problems by proposing the following:

• A new communication mechanism that can be applied in many different computer subfields to combat errors in individual entities (agents, computers, robots, etc) that work in a group [25,26]. This communication will allow entities to remove malfunctioning entities using only local information and neighborhood messaging.

• A new form of cancer modeling as well as the first steps toward a novel solution to removing tumor cells that may explain natural phenomena and indicate future treatment [13,14]. Our solution involves message passing similar to in the artificial system, allowing cells to maintain the health of their area in the system.

• An application of the same techniques to modeling Alzheimer’s in order to give a computational model that could help explain the causes of the disease, specifically related to the growth of lesions near damaged neurons.

6. Work Plan

I plan to continue the multi-agent robustness investigation, as well as increase the work on biological analogies to be applied to human cells in regards to cancer and neurodegenerative diseases:
1. Add additional biological details such as blood vessels (angiogenesis) to increase the biological plausibility of the cancer model.

2. Continue developing a mechanism that can be easily applied to many computer systems.

3. Develop the neural network model of neurodegenerative diseases.

7. Research Philosophy

My research philosophy is strongly based on an interdisciplinary approach to problem solving. I believe that the future of computer science is in combining knowledge not only from computer science subfields, but also from relatively different fields. Although my focus has so far been on the relation between biology and computer science, I do not plan to limit myself to those two fields during my research career. I am interested in tackling big problems, and feel that my dissertation has opened up many different directions for me to pursue in my future research endeavors.

8. Advice to other Students

Be willing to work on problems you have little background for, but realize that you will have extra work initially in getting up to speed. There are many interesting problems to solve, and you can’t solve them all in your PhD. Your advisor should help you find the ones to focus on, and you should be sure to keep track of any ideas you have along the way. By your final years, you will have many potential directions to follow either during or after your PhD, based on what you learned in grad school.

Sometimes it feels difficult and overwhelming, but essentially everyone goes through those feelings; don’t let them dissuade you from doing what you want to do. When feeling frustrated, take a step back and find a way to reinvigorate yourself in your research. If you want to work on unconventional problems, be ready to have to prove that your work is worthwhile and relevant to multiple communities, and don’t back down from the challenge. It will be worth it in the end.

Acknowledgements

The author would like to thank her advisor Hava T. Siegelmann and her committee members Victor Lesser, Ramesh Sitaraman, and John Nambu for their continued advice. She would also like to thank Nava Siegelmann-Danieli for sharing her cancer expertise.

Three years of her work was supported by a Department of Homeland Security (DHS) Fellowship, nine months by the National Science Foundation (NSF), and eight months by the Office of Naval Research (ONR), thus far.

References


