Resilience of the Human Immune System and Software Systems: A Comparative Study

Abdullah Alshammari¹, Peter Keiller DSc¹, Hassan Karim¹, Shifaa Alshammari²
Electrical Engineering and Computer Science ¹, Department of Biology ², Howard University, Washington, DC, USA
abdullah.alshammari@bison.howard.edu

Abstract - The human immune system is an advanced information processing system. Many studies show the similarities between the biological immune system and software systems. Biological researchers increasingly rely on software reliability models to generate explanations for their findings. Software engineers increasingly look towards human and animal biology for inspiration on which to base their models. In this paper, we present a comparison between the human immune system and software systems. Further, we present an argument with examples, that show that by leveraging software engineering resilience modeling techniques, biological resiliency might be better understood. We introduce a new process, Left Shifted Immunization, and we compare it to the Akka Actor model to evaluate which better emulates immune system behavior, predicts natural resilience, and approximates the capabilities and limits of biological systems.

Keywords: biological, immune, software, resilience, anti-malware.

1. Introduction

Biological systems are important in the sphere of software engineering as they enable software engineers to describe and model software. Further, software engineering models, are now at the heart of most biological scientific experiments. Using software resilience modeling techniques, the natural resilience of biological systems can be determined. And software engineering researchers aim to produce algorithms and models that are as repeatable and resilient as the biological systems that have developed over millions of years. Artificial Intelligence and Anti-virus immunization software are two primary examples of software engineering’s attempt to model biological systems. This current work is a literature review that compares resilient software system models to resilient biological systems in order to evaluate the potential efficacy of models in the former over the latter. This paper represents an early look at research that seeks to establish the strong correlation between biological immunity and system software resilience.

2. Resilience

Resilience is one of those aspects of an evolutionary system that enables it to survive by either avoiding harm or overcoming harm when encountered [1]. We use this definition for resilience modeling in biological systems as well as software systems. Such a definition, and related models, are important because they enable us to monitor system performance based on predefined expectations, characterize workload, and provide clues on methods to maintain and improve performance.

The proposed research will present a set of applicable software engineering models that biological researchers can leverage in order to get more reliable, and predictable scientific experiments.

3. Software Engineering Resilience Models

There are several software system reliability models that can be used to illustrate the auto-immune process. For example, the Akka actor model [2], and Chandra Disturbance Modelling [3]. Computer software is increasingly exhibiting self-healing properties and are being used to automate operational tasks [4]. In effect, software automation, that includes embedded testing and problem resolution is the goal of resilient systems [5]. Techniques like software simulation experiments, which have been useful in transforming engineering methodologies with failure prediction [6], have also been shown to be useful in Biological research [7], [8].

4. The Human Immune System

The immune system refers to the set of mechanisms [9] that protect organic life. The human immune system employs an interconnection of advanced genetic level mechanisms to protect the body from intruders. Defense of the human body against attacks from external microorganisms is done through approximately four layers of protection namely the physical barriers, the physiological barriers, the chemical barriers and cellular protection [10]. The cells charged with the responsibility of ensuring immunity of the human body are produced in the bone marrow and subdivided into three basic cell types namely Macrophages, B cells, and T cells [11]. Biological resiliency is reflected in natural organisms, as each one constantly encounters threats from other organisms, and each organism develops a set of mechanisms to protect themselves from harm.

The information processing capacity of the immune system has made it an area of interest for researchers. It has the capability to extract unique information from microorganisms and classify harmful pathogenic peptides [7]. It also employs memory to recall signature patterns that it has previously interacted with. The overall behavior of
the human immune system results from the information exchanged between several local systems. The various immune cells which make up the immune system move between numerous secondary and primary lymphoid body organs. The cells are appropriately controlled to ascertain that there is enough distribution of B and T cells across different locations of the body. Antigens, information bioproducts of this distribution, are formed in the body via the process of differentiation and clonal expansion. Unlike typical computing models, the human immune system does not have any central point of control as the system is self-regulatory. In Fig. 1 the Alshammari-Keiler block diagram depicts biological immune system components in a manner like typical software systems.

![Image of the Alshammari-Keiler diagram](image)

**Fig. 1:** The AlShammari-Keiler diagram describes the human immune system based on standard block elements used in software engineering process design.

5. **Software Systems**

Computer software is made up of several elements that interact together to perform certain tasks. Software systems take input, perform some processing and produce output. Output files are stored within the memory of computer systems. The software systems are susceptible to virus attacks which can compromise the integrity of information stored in these files [10]. Moreover, malfunctions in the software systems may also impact the performance of the software system. However, virus infections can be detected by antivirus programs installed in the memory of computers. When an antivirus program detects an external code trying to modify a file, the program reacts by protecting the computer resources against any harm. The software industry has even produced next generation antivirus software that applies anti-biotic like protection to software code to prevent the recurrence of pathogen-like malware.

6. **Software System and human correlates**

In this section, we demonstrate the more prominent common features that appear to have correlation.

6.1 **Detection, Classify and Defense**

The immune system uses T lymphocytes cells to extract unique information from antigen molecules, classifies it as harmful, and immediately releases cytokine protein peptides [12]. This is the exact behavior of anti-malware systems.

6.2 **Flexibility and adaptability**

Flexibility is another feature that both the immune system and software systems demonstrate a strong correlation. Fisher, et al. point out that biological systems are highly adaptive, to both internal and external changes; they use signals coming in from receptors and sensors, as well as emergent properties to fine-tune their functioning. [13] and [7] further demonstrated that both the human immune system and software systems develop over some time and adapt to several changes.

6.3 **Fault Tolerant**

Resistance to failures is another feature shared by both the systems. Bradley and Tyrrell [14] highlight that the human immune system is composed of several redundant systems which work in conjunction with each other to ensure that the body is protected against harmful diseases. They further introduce “immunotronics”, a system fault tolerance model designed to exhibit resilient characteristics. Immunotronics adequately models the immune system which is composed of tissues, special cells, proteins and organs that shield the body against the microorganisms [15].

6.4 **Response to Disturbances**

Both systems, can respond to various disturbances. They also can be engineered towards stability by leveraging automated modeling and systematic treatment. Chandra claims resilience of transportation to be one of the systems to react to any kind of disturbance [3]. Chandra further contends that immunological studies would benefit from disturbance modeling which accounts for responses to different kinds of disturbance. Chandra’s Disturbance Model relies on statistical analysis of the disturbance of internal distributions which is typical in behavior anomaly detection anti-malware systems.

6.5 **Embedded Rules**

In contrast to the disturbance modeling by Chandra, Faeder et al. make an argument for the efficacy for rule-based models of biological systems. Models generated via their BioNetGen tool enables molecules and even full cell colonies to be represented as correctly structured objects that influence the systematic interactions between proteins [16].

6.6 **Security**

Rodriguez et al., in their work, mention the other pattern of influence of resilience – security performance. Intrusion into vulnerable components can be avoided with the approach of excluding immunity faults [16]. There is even a complex package of resilience model, introduced in the article, that show where all factors and components of successful or unsuccessful management of threats are interrelated. They make a strong case that everything is founded on the connection between biological immunity and software engineering resilience system, so all these
relations can be monitored and managed in a proper way in order to improve the resilience system.

6.7 Summary of resilient correlations

Table 1 summarizes features of the two systems that show high correlation.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Biological Component</th>
<th>Software Correlate</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detect, Classify, Defense</td>
<td>T cells, Macrophages, B cells, cytokine proteins</td>
<td>Anti-malware</td>
<td>[12]</td>
</tr>
<tr>
<td>Flexibility/Adaptability</td>
<td>Cytokine proteins</td>
<td>Reinforcement learning</td>
<td>[20]</td>
</tr>
<tr>
<td>Signature pattern recall</td>
<td>Peptides</td>
<td>Regex libraries</td>
<td>[21]</td>
</tr>
<tr>
<td>Fault Tolerant</td>
<td>Cytokine proteins</td>
<td>Dual paths graphs</td>
<td>[14], [20]</td>
</tr>
<tr>
<td>Response to Disturbances</td>
<td>Cytokine proteins</td>
<td>Reinforcement learning</td>
<td>[3]</td>
</tr>
<tr>
<td>Embedded Rules</td>
<td>Cytokine proteins</td>
<td>Signature matching</td>
<td>[3], [19]</td>
</tr>
<tr>
<td>Security</td>
<td>Cytokine proteins</td>
<td>Anti-malware</td>
<td>[5], [19]</td>
</tr>
</tbody>
</table>

Table 1: Feature Comparison.

7. Ideal Models

7.1 Akka Actor

Autili et al. [1] promotes the principle of collaboration of software resilience and biological immunity, since their proper maintenance can change the human approach to the environment in general. Like Autili et al., we acknowledge that the Akka actor model shows efficient mathematical models of resilience computation, which watch the behavior of actors and their approach to social situations. Being guided by human reaction, the algorithm calculates the allowable preventive recognition, sent by our immune systems.

7.2 Shift Left

Bjerke-Gulstuen’s “Shift Left” extreme programming approach is a software resiliency model that enables fault detection early in the software development life cycle (SDLC), the standard model for developing software [19] and [22] both describe an automated process of the Shift Left model based on automatic code inspection that feeds code property graphs. The code property graphs highlight faults in code like T-cells detecting vulnerabilities, pathogens, in massive code sets.

Fig. 2 demonstrates the immunization process in the SDLC. We augmented it with a Shift Left approach contending that problems, like pathogens, should be detected early and immunized early. The last step in our Left Shifted Immunization During SDLC (LSI-SDLC) process represents a functioning anti-malware system described in Fig. 3. We’re calling Fig. 3 the Left Shifted Immunization Malware Learning and Defense process. It describes the Shift Left software engineering process automated into a self-healing next-gen anti-malware process.

7.3 Akka Actor vs. Shift Left

Although Autili et al. demonstrated the value of modeling the immune system via the Akka Actor model, we argue that a Shift Left methodology applied to the full SDLC for an anti-malware system represents a more appropriate model. LSI goes further than Autili’s Akka Actor. They generalized the Akka Actor modeling approach. But Akka Actor doesn’t itself account for the probabilistic response to disturbances demonstrated by the immune system as Chandra proved and LSI includes. LSI-MLD leverages disturbance as a factor in approximating the limits of a system by summing undisturbed state ($S_u$) with the disturbed state ($S_d$). Since disturbed state is a probability, the total sum can only be approximated. Resulting an immune (I) limit equation 1 of

$$ I \approx S_u + S_d $$

Further, since LSI embedded testing the concept (which could have had embedded problems - pathogens), during the early phases of the SDLC, it more adequately addressed the evolutionary resilience aspects of the immune system.
That is, Akka Actor assumes a fixed amount of information. However, there does not appear to be a fixed amount of information in the immune system. As far as researchers can tell, the immune system has developed evolutionary defense mechanisms that get passed on to newly created cells (unlimited information). LSI incorporates this behavior via its reinforcement learning capabilities.

In Table 2, we show that based on the reasons and features listed, the LSI process better emulates immune system behavior, predicts the immune system’s natural resilience, and approximates its capabilities and limits.

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Akka Actor</th>
<th>Left Shifted Immunization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emulates immune system behavior</td>
<td>1 – YES</td>
<td>1 – YES</td>
</tr>
<tr>
<td>Predicts natural resilience</td>
<td>0 – No</td>
<td>½ - models resilience but doesn’t predict</td>
</tr>
<tr>
<td>Approximates capabilities</td>
<td>1 – YES</td>
<td>1 – YES</td>
</tr>
<tr>
<td>Approximates limits</td>
<td>0 – No</td>
<td>1 – YES</td>
</tr>
</tbody>
</table>

Table 2: Shows that when comparing Akka Aktor to Left Shifted Immunization (LSI), LSI better models human immunization.

8. Discussion

Before LSI, the primary shortcoming from most models was they didn’t account for the immune system’s ability to improve over generations, i.e. its resilience. This was probably due to the complexity of interconnected systems. We have however demonstrated that the LSI process accommodates for most of the features that make the human immune system resilient. It is based on this that we confidently conclude that there are software engineering models that adequately describe biological systems. Researchers looking to leverage the benefit of the immune system for modeling systems like high-performance computing, artificial intelligence systems, vehicular networks, brain-computer-systems can safely look to the LSI process for resilience capabilities.

9. Conclusion

The information processing capabilities of the human immune system makes it an area of immense interest for researchers. The immune system has the capacity to perform complex operations in a parallel and distributed manner. Like the nervous system, this system can acquire information, recall previously processed information and perform pattern recognition. In software engineering, immunity-based systems have been applied in many areas, some of the areas where it has been applied include in the creation of virus detection software systems, pattern recognition, and noise detection. It is these similarities, as demonstrated with our Left Shifted Immunization (LSI) process, that will continue to help establish the correlation between the human immune system and software engineered resilience.

10. References


