TOWARDS IMMUNE INSPIRED FAULT TOLERANCE IN EMBEDDED SYSTEMS

Jon Timmis, Rogério de Lemos*, Modupe Ayara* and Ross Duncan***

*Computing Laboratory

University of Kent at Canterbury, UK.

{j.timmis, r.delemos, moa2}@ukc.ac.uk

** Advanced Technology and Research Group
Self Service Strategic Solutions
NCR FSG Ltd., UK.

ross.duncan@scotland.ncr.com

ABSTRACT

The immune system is a remarkable natural system that is proving to be of great inspiration to computer scientists and engineers alike. This paper discusses the role that the immune system can play in the development of fault tolerant embedded systems. Initial work in the area has highlighted the use of the immune process of negative selection, and more importantly the concept of self/non-self discrimination in the application of artificial immune systems in fault tolerance. This paper reviews those works, highlights issues relating to the way in which this area is approached, and raises important points that need to be considered before effective immune inspired fault tolerant systems can be constructed.

1. INTRODUCTION

As systems grow ever more complex, alternative sources of inspiration for solutions to the problems associated with these systems are being sought by computer scientists and engineers. Biology has been seen as a great resource by researchers in these areas, which has provided inspiration to create various biological inspired techniques such as genetic algorithms, neural networks etc. The immune system is now receiving more attention and is slowly being realized as a new biologically inspired computational intelligence approach [1]. An intuitive application of the immune system, and one that many researchers have followed, is to create artificial systems that have the ability to differentiate between self and non-self states: where *self* could be defined as many things such as the systems normal behavior, normal network traffic between computers and so on. However, it is worth noting that the immune system is capable of much more than simple detection, for example, it can remove harmful antigens by learning and adapting to invading antigens. As a computational intelligence approach, the first use of the immune system metaphor was proposed for use in security of computer systems [2].

This paper explores and challenges the current way of using the negative selection approach in fault tolerant embedded systems. The paper stems from ongoing research into the creation of immunised fault tolerant embedded systems and preliminary results that have been obtained [3]. A simple overview of artificial immune systems (AIS) and fault tolerance is provided. Then a review of the current work concerning immunised fault tolerance follows, where comments will be offered as to the major challenges this area of research faces. The paper concludes with the suitability of using negative selection for complex systems, and a perspective on the future of this research.

2. ARTIFCIAL IMMUNE SYSTEMS

The immune system is a remarkable, but complex natural defense mechanism. This paper will not detail the immune process due to space. However, readers are directed to [4] for more details on relevant immunology, in particular the process of negative selection.

Artificial immune systems (AIS) are adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to complex problems [1]. There exist a number of immune inspired algorithms such as clonal selection [5] immune networks [6] and negative selection [2]. These algorithms can be considered to be generic, and through the use of shape space (representation) and affinity measures, can be employed in a wide variety of applications, reaching far beyond their original intended domain [1].

Using the negative selection process, as inspiration, there have been a number of works that have attempted to build an artificial immune system for virus detection [2] and computer security [7]. The original work in [2] proposed the negative selection algorithm, which has formed the basis of virtually all the research in the AIS related computer security domain and has more recently, provided inspiration to build fault tolerant systems.

There are many reasons why the immune system has attracted such interest, and why it may be of particular relevance to fault tolerant embedded systems, for example: tolerance to self molecules, continual operation in a hostile environment, adaptability, self organisation, diversity, inherent distributed nature, inherent speciation of antibodies, antigenic binding, learning and memory.

3. FAULT TOLERANCE

Dependability is a vital property of any system justifying the reliance that can be placed on the service it delivers [8]. Fault tolerance is a means of achieving dependability working under the assumption that a system contains faults (e.g. ones made by humans while developing or using systems, or caused by aging hardware), and aiming at providing the specified services in spite of their presence. A fault is the adjudged or hypothesized cause of an error. An error is the part of the system state that is liable to lead to the subsequent failure. A failure occurs when a system service deviates from the behaviour expected by the user. The quality of the service provided by a system can be assessed in terms of the dependability attributes. For the purpose of this paper, two of these attributes: availability (the readiness of usage), and reliability (continuity of service) are considered.

Fault tolerance consists of error processing, which aims at removing errors from the system state before failures happen, and fault treatment, which aims at preventing faults from being once again activated. Error processing typically consists of three steps: error detection, error diagnosis and error recovery. Fault treatment consists of two steps: fault diagnosis, and fault passivation. The latter, might consider system reconfiguration if the system is not capable to deliver the same service as before. Providing system fault tolerance plays an ever-growing role in achieving system dependability as there are many evidences proving that it is not possible to rid the system and system execution from faults. These include the growing complexity of systems, operators' mistakes, and failures in the environment in which the system operates.

4. IMMUNISED FAULT TOLERANCE

There have been a number of works in the literature that attempt to make use of the negative selection approach for a variety of applications, such as software fault tolerance [9] and hardware fault tolerance [10, 11]. Earlier work in [12] proposed the use of the immune system for the development of immunised hardware, so called *immunotronics*. The work in [12] highlighted potential aspects of the immune system that could prove potentially useful when developing fault tolerance hardware systems, and will be explored in more detail later. More recent work, in the domain of network intrusion detection [13], has opened the debate as to the real usefulness of the negative selection approach for large-scale applications, and suggested possible hybrid approaches with other immune inspired algorithms.

4.1 Immunised hardware and software

Initial work in [12] made the first attempt at mapping immune system entities and processes into the domain of hard fault tolerance. Here the author suggested that it was possible to utilise the idea of self/non-self recognition from the immune system for detecting erroneous states in hardware: here erroneous states would be classified as non-self, as the hardware would have been endowed with a sense of self (acceptable states). Building on this initial work, [10] constructed a small hardware fault tolerance model implemented on a Field Programmable Gate Array (FPGA). Using the idea of state-machines, the self (or normal) behaviour of the system was captured, thus endowing the system with a sense of self. The system was exposed to possible erroneous states (antigens), and demonstrated the ability to detect these erroneous states. This work was extended to combine *embryonics* (inspired by embryonic development) with artificial immune systems [14]. These novel ideas hold great promise for the future.

Within the fault tolerant software domain, work in [9] proposed an immune network model for the detection of erroneous states of an algorithm during execution and then executing recovery procedures. The authors employed the use of a genetic algorithm combined with immune network approach to evolve sets of detectors. Here the authors employed the use of both self and non-self information, whereas negative selection approaches only typically make use of one or the other.

4.2 Addressing the way the problem is approached

From assessing the relevant literature, it appears there are plenty of thoughts suggesting that the immune system, in particular negative selection, is an attractive source of inspiration for fault tolerant systems. However, there are a number of questions and issues that have not yet been addressed: (1) Is the computational complexity for generating a set of effective detectors that important? (2) Should we be concerned with self or non-self data? (and how is it best to represent that data?) (3) Given that the artificial immune system (AIS) will be in an embedded system, how do we achieve minimal number of detectors, which achieve maximum reliability? (4) The immune system is much more than a two state classifier is the current thinking limiting the richness of the immune system metaphor? (5) Is an evolutionary strategy, such as the immune system approach, a suitable one to be adopting? This section will now attempt to address some of these issues; the goal here however, is not to answer all questions, but to raise them as important issues for consideration.

(1) Computational complexity. As previously mentioned, research outside the area of fault tolerance has highlighted the potential drawbacks of the negative selection approach when generating suitable detectors, particularly, to the time it takes to generate a suitable size repertoire of cells that are capable of reasonable detection coverage [13, 15]. The outcome of this work is further confirmed in [3] where numerous negative selection inspired algorithms were assessed for feasibility, and all shown to have the same drawback. The time it takes to generate a suitable set of detectors depends on the size of the data (assume this to be self for this discussion, but it does not necessarily have to be so). Work has shown that as self increases, the cost of generating these detectors is exponential in relation to the size of self and the size of the alphabet used to represents self [13, 15].

(2) Self/Non-Self representation. In the context of embedded systems, a choice has to be made, as to the most suitable data for representing self and non-self: this may or may not be normal behaviour. If the size of erroneous data with respect to the non-erroneous states were smaller, then the smaller would be a more appropriate choice of self-data. Addressing the issue of what really is the data is also important. If one abstracts away from the system components and uses state machines, for example see [11], then one has to be careful that there is an accurate mapping between the state machine and the actual system, and ensure that the state machine adequately scopes the space to be *immunised*. Consideration here also has to be given to the way in which data is represented. The shape space paradigm, formalised in [1] proposes varying ways of data representation and interaction. However, when dealing with discrete values, such as those found in embedded systems, the method of defining affinity (i.e. seeing how similar one item is to another) is not as clear-cut as it may seem. This is coupled with the fact that mutation, even what might be thought of as a small amount, could have a huge impact on the meaning of the data. Should a binary shape space be employed, the mere flipping of one bit could indicate a huge shift in meaning of the state, rather than the small shift that may be desired. In both of these situations, domain knowledge can play a pivotal role in the success or failure of such as system. This issue has largely been ignored.

(3) Minimal detector generation. In the domain of embedded systems it is necessary to reduce the number of detectors in order to increase the run-time efficiency of an AIS. One approach to achieve this would be to ensure that the recognition space of the detectors do not superimpose, assuming that the threshold of the recognition space is optimized for reducing the amount of detectors. Another approach would be to incorporate the idea of domain knowledge into the generating procedure, through the careful selection of representation and affinity measures. It may even be possible to evolve around partial knowledge of self in order to obtain complete knowledge. This could be achieved in part by the segmentation of the search space of high level of abstractions of the self-data, thus minimising the search area.

(4) Limiting the immune approach. Current research seems to have focused on the use of negative selection and more importantly, this simple view of self/non-self discrimination. This in effect treats the whole system as a two state classifier, and the question really needs to be asked: instead can we do this with simple traditional classifier systems? In order to make richer use of the immune metaphor, approaches need to be extended away from this simple two-class problem to the more complex problem that is faced. In the immune system a certain amount of evolution is occurring (with the development of new B-Cells, antibodies etc) when changing and adapting to new input stimuli. Indeed, the immune system (with help) can even adapt to new parts of self, such as when an organ transplant occurs. These concepts should be incorporated into approaches adopted when developing AIS.

(5) Coverage. An important criterion when applying negative selection in fault tolerant embedded systems is the coverage of the detectors. In other words, the size of the detectors set is not enough to achieve a particular detection rate, if the set of detectors are not representative of the non-self set. Hence, there is a need to maximize coverage and minimise the size of the detector set.

(6) Evolution. Within the domain of fault tolerant embedded systems, what is required is an artificial immune system that can learn and adapt with the development of new systems, but also cope with a dynamic and changing environment. This therefore raises the question of the suitability of the immune inspired approach. In order to justify this approach, there needs to be the discussion addressing questions such as, where is the analogy of embedded system with evolution and immune systems. Do embedded system really evolve? Are we really dealing with a simple self/non-self recognition problem? This will be dealt with in more detail in the following section.

4.3 The agents of change

Infecting antigens drive the development of antibodies within the immune system. These *agents of change* can be considered to be external infecting antigens, which are driving the immune system to protect the host body from infection, and drive the immune system to adapt to changing antigenic infection. However, when one considers embedded systems, one has to consider whether they really evolve, in the sense previously mentioned. Embedded systems are in their very nature self-contained systems, hence they should not, in principle, be considered as evolving systems. They do however, interact with the outside world, which could affect the system. Factors such as electro-magnetic noise, radiation, vibration and temperature may affect the normal operation of the system and thus potentially causing faults. In addition to the above factors, components might fail, which can affect the operation of the system, system consumables may become exhausted, and abnormal human interaction could also affect the system in some way. Any artificial immune system for embedded systems should be able to cope with all of these agents.

However, these are not the only agents of change, there might also be changes in the physical components of the system e.g. replacement of faulty parts, upgrade of components or the addition of new components. The concept of adaptation is therefore important. Any immunised embedded system will need to be able, firstly, to detect such consequences from the agents of change, and secondly, to adapt to them and possibly new ones. It should be noted that the AIS does not have to detect a change in components, but merely the consequences of that change. An analogy can be made with the immune system. Should a host have an organ transplant, the immune system does not know this is a new organ, merely that something has changed and it is no longer recognisable as self, i.e. it has detected the consequence of the change.

In an AIS, this can be viewed at two levels. At one level, there are minor adaptations of a system to the environment, e.g. if a component fails, the system should be able to detect the consequences of this failure and reconfigure for continuing to provide a degraded service when available redundancies do not permit the continuation of delivery of the original service. At the other level, there are the issues of possible families of embedded systems, where a whole host of similar embedded systems are developed over the cause of time, with similar or different components. What is desired here, therefore, is a system that can have an immune system capable of adapting to new components, new operating conditions etc, without the need to retrain it, but use the immune knowledge of existing embedded systems. This then naturally leads to two areas of reconfiguring the AIS. The first is at design time. A new embedded system (the first of its kind) can have a set of detectors generated that should be capable of working with that system: in essence this is a static generation of detectors, off line. However, the second area for reconfiguring the system is at run-time. The system should allow new components to be introduced, removed and so on and be able to adapt to these changes: having to re-learn a new set of detectors from scratch is not practical they need to be *evolved* from the knowledge the AIS already has and can capture from its new hardware/software or environment. It may also be possible to introduce new detectors with a new component, therefore a new component is already endowed with its own immune system, which is then integrated into the embedded systems immune system.

4.4 A role for immune network metadynamics

Discussion in previous section raises the question of allowing for the continued development of the AIS in response to replacement components, or the introduction of a whole family of systems. The idea here therefore, is to create a basic artificial immune system, which can then learn new identity when placed in any of a related family of embedded systems. This gives rise to the idea of reusing immune systems within families and also the possibility of reusing antibodies produced by one system in another, therefore eliminating the need for all systems to learn all things – information sharing may be possible. To help address this issue, it is necessary to turn back to the immunology literature to see what can be gleaned from the immune system.

An analogy can be made with the metadynamics of the immune system, which in turn is related to the immune network. Work in [16] details the idea of the self-assertion role of the immune system, where the immune network is able to identify self over a period of time, rather than be *endowed* with this sense of self at the beginning. Metadynamics is the recruitment of new individuals (B-Cells) some of which are randomly produced into the immune network structure. The immune network has the ability to maintain a structure of cells which can adapt to the changes in the environment and has what is know as a double-plastic structure, i.e. the network structure can change, as can the contents. Indeed work such as [6] have to some degree begun to capitalize on this idea, albeit in a different domain. In order to develop a learning fault tolerant embedded system, it would appear that immune networks offer a plausible solution, in part tackled by work in [9], but never fully exploited. Additionally, the domain of fault tolerance has a great deal to learn from works such as [13].

5. CONCLUSIONS

This paper has considered the role of the immune system as inspiration for creating immune inspired fault tolerance in embedded systems. It is argued that the immune system is a good candidate from which to seek inspiration to develop adaptable artificial immune systems for such devices. Previous research seems to have primarily focused on the use of negative selection and the concept of self/non-self discrimination of the immune system to help solve this problem. However, there are bigger issues involved and many considerations that have to be taken into account when developing such systems, such as: what actually defines the self for an embedded system; how is data represented; what, if any, are the consequences of choosing such as representation; how does the system learn new identity when components are changed. Thought also has to be given to what happens when the environment in which the system is operating changes, or there is a desire to reuse immune systems from other machines in a whole family of machines, all with different and sometimes similar devices contained within. These are open questions, and current research in AIS in general has made a good start utilising immune inspired approaches, but a bigger picture needs to be examined and questions asked (and answered) before significant progress can be made. Also, one possible way forward, which has been hinted at in this paper, is to explore the richness of the immune metaphor and seek alternative approaches that it may offer. It is also important to bear in mind, when dealing with embedded systems, a minimal number of receptors is desired. This may be possible via the niching of detectors, which can then be used to extract clusters of abnormalities. Abnormalities that have common patterns are clustered together and can thus be detected by a wide broad detector. It is hoped that this paper has gone some way to start addressing these very important issues.

6. ACKNOWLEDGMENTS

The authors would like to thank Dr. Leandro de Castro for his thoughts and ideas during his time at UKC. We would also like to thank NCR FSG Ltd. for their continued financial support and valuable input into this research.

7. REFERENCES

[1] De Castro, L.N. and J.I. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. 2002: Springer-Verlag.

[2] Forrest, S., et al., *Self-Nonself Discrimination in a Computer.* Symposium on Research in Security and Privacy, 1994: p. 202-212.

[3] Ayara, M., et al. *Negative Selection: How to Generate Detectors.* To appear in 1st International Conference on Artificial Immune Systems (ICARIS), University of Kent at Canterbury. 2002.

[4] Staines, N., J. Brostoff, and K. James, *Introducing Immunology*. 2nd ed. 1994.

[5] de Castro, L.N. and F. Von Zuben, *The clonal selection algorithm with engineering applications.* Proceedings of Genetic and Evolutionary Computation Conference, 2000: p. 36-37.

[6] Timmis, J. and M. Neal, *A resource limited artificial immune system for data analysis.* Knowledge Based Systems, 2001. **14**(3-4): p. 121-130.

[7] Forrest, S., S.A. Hofmeyr, and A. Somayaji, *Computer Immunology.* Communications of the ACM, 1996.

[8] Laprie, J.C., *Dependable Computing: Concepts, Limits, Challenges.* Special Issue of the 25th International Symposium On Fault-Tolerant Computing, 1995: p. 42-54.

[9] Xanthakis, S., et al., *Immune System and Fault Tolerant Computing*, in *Lecture Notes in Computer Science*, J.M. Alliot, Editor. 1995, Springer-Verlag. p. 181-197.

[10] Bradley, D.W. and A.M. Tyrell, *Immunotrics: Hardware Fault Tolerance Inspired by the Immune System*, in *Lecture Notes in Computer Science*. 2000. p. 11-20.

[11] Bradley, D.W. and A.M. Tyrell, *A Hardware Immune System for Benchmark State Machine Error Detection.* World Congress on Computational Intelligence., 2002: p. 813-818.

[12] Tyrell, A.M., *Computer Know Thy Self!: A Biological Way to Look at Fault-Tolerance.* Second Euromicro/IEE Workshop on Dependable Computing Systems, 1999: p. 129- 135.

[13] Kim, J. and P. Bentley, *Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Clonal Selection with Negative Selection Operator.* Congress on Evolutionary Computation, 2001: p. 1244-1252.

[14] Bradley, D.W. and A.M. Tyrell, *Embryonics + Immunotrics: A Bio-Inspired Approach to Fault Tolerance.* 2nd NASA/DoD Workshop on Evolvable Hardware, 2000.

[15] Kim, J. and P. Bentley, *Towards an Artificial Immune System for Network Intrusion Detection: An Investigation of Dynamic Clonal Selection.* Congress on Evolutionary Computation., 2002: p. 1015-1020.

[16] Bersini, H. and F.J. Varela, *The Immune Learning Mechanisms: Reinforcement, Recruitment and their Applications*, in *Computing with Biological Metaphors*, R. Paton, Editor. 1994, Chapman and Hall. p. 166-192.