

# Using Stochastic Situation Calculus to Formalise Danger Signals for Autonomic Computing

M. Randles, A. Taleb-Bendiab, P. Miseldine

*School of Computing and Mathematical Science, Liverpool John Moores University, Byrom St.  
Liverpool, L3 3AF, UK*

*{cmsmrand, a.talebbendiab, cmppmise}@livjm.ac.uk*

## Abstract

*In order to maintain and promote a robust and dependable system it is required that agents, representing system services, are allowed some autonomy in their operations. It is also required that the system be as open as possible, without compromising security, to allow the introduction of novel procedures and performance improving changes or additions. This paper uses concepts from Artificial Immune Systems (AIS) engineering and the Stochastic Situation Calculus dialect of predicate logic, to formalise the detection of system novelty, based on danger signals as an immune (self-healing) and evolutionary (self-adaptive) reaction trigger. In this way any threat or potential enhancement to the system can be monitored for and the appropriate action taken to facilitate system self-governance, ensuring predictable and safe self-adaptation.*

## 1. Introduction

The convergence of telecommunications, data networking and broadcasting has further accelerated the need for highly distributed autonomous systems. This encompasses the emerging technologies of wireless systems, sensor networks, networked appliances and pervasive, ubiquitous, on-demand computing, creating a need for high dependability. This cannot be attained without addressing software dependability issues [1], which are exacerbated by systems' complexity, their inherent diversity and frequently changing technology and users' requirements [2, 3].

Accepting the hypothesis that adopting biologically inspired situated autonomic systems design [4] can underpin our quest towards more dependable – self-adaptive distributed software, the authors argue that developers of such systems will be required to generate and deploy, together with the systems, their associated formal model. This can be reasoned upon, for instance during

governance/control of systems' autonomic operations including; self-management and adaptation.

The formal model for meta-control of system behaviour, through distributed agency, is of special importance because it permits the separation of self-governance rules from the functional norms of the applications. This approach has been recognised as beneficial from initial policy based accounts [5] through to more flexible deliberative architectures [6]. In this way the nature of the system can be adapted dynamically without the need to recode functionality and modifications can be performed without halting the system.

Research into system self-management has focussed on the “sensor-effector” mechanism for autonomic behaviour [7]. This paper, however, presents a formal semantics for the event-situation-condition-action sequence, via the stochastic situation calculus [8, 9] and danger theory [10], which is used to formalise the “adjustable” governance models for autonomic software behaviour and system evolution.

The situation calculus is a well established formalism to model dynamic systems [11]. It facilitates the representation of time, continuous processes, actions performed by agents with free will, actions performed by nature, non-deterministic actions of chance, knowledge producing actions and the mental state of service agents. It can handle concurrent actions, prediction, planning, diagnosis and hypothetical reasoning [12, 13, 14]. It is a (mostly) first order predicate calculus language. Thus it can be widely understood even by those unfamiliar with the actual language of situation calculus itself. It gives a truly logical specification of a system so that items of interest, such as undefined novel behaviour, as well as parameters used for the governance, follow as logical consequences. The situation calculus is well suited to solving the problem of providing a formal and computational account of the complex dynamic processes in system self-governance and adaptation.

The use of biologically inspired concepts, such as the Artificial Immune System (AIS), stems from the perceived robustness and self-healing properties of natural systems [15]. In this paper a theory of how the immune system operates, namely danger theory, is used to provide a formalisable mechanism to promote self-adaptation and self-healing. Danger theory challenges the received view that the immune system's main concern is the distinction of self from non-self [16]. Alternatively it sees danger detection as its primary function. Whilst these views are not mutually exclusive [17] it is not the purpose here to debate the merits of the models but rather to take useful notions, from these biological insights and appropriate them for a computer system artificial immune mechanism.

Accordingly this paper continues, in section 2, with a background look at danger theory in the context of AIS. Section 3 provides an overview of the situation calculus, its extension to stochastic actions and how to specify danger. Section 4 explains how the formalism may be used to detect and react to danger, leading to the case study in section 5. The paper concludes with a note on future research and the usefulness of this approach.

## 2. AIS and Danger Theory

In a human body the integrity of the whole is aided by the immune system. It is usually perceived to function on three levels:

- The External Barrier: Skin, Hair, etc.
- Innate Immunity: Genetic Information Inherited from Parents
- Acquired Immunity: Learned Responses to Adapt to External Threats.

In a computer system these have direct analogies. An external barrier is the security built around the system; firewalls, password protection etc. Innate immunity is the pre-specified dangers that the designer can predict and set appropriate responses for. The acquired immunity is more difficult and complex to achieve. In this antibodies or receptors in the system are matched to foreign antigens or threats to the system. In classical immunology it is thought that an immune response results when a foreign or non-self object is encountered within the system. However more recent research, in immunology [18] suggests that the discrimination performed must be more sophisticated than simply self versus non-self. A framework [19] is proposed to analyse applications of biological models to computing. In particular meta-models to preserve openness in evolutionary systems to cope with *far from equilibrium*, heterogeneous, diverse and interacting system properties are needed. Stepney et al [19] contend that current mathematical and computational descriptions of biological models tend to be static. It is for this reason that the situation calculus is proposed here to evaluate and model

danger, in dynamical systems, which can then be placed in the context of a conceptual framework.

To enforce an immunological model on a computer system requires a dynamic or changeable version of self. In this way danger signals can be interpreted and appropriate action taken, new receptors can be instigated to cope with unforeseen danger and the system can become self-tolerant of changed but normal behaviour. So that non-self may still be considered an indicator of danger but it is not the overriding immunological response.

The idea that the immune system evolves to recognise pathogens (dangerous situations) is not particularly new [20]. What danger theory suggests is a new perspective whereby the immune system's primary motivation is to detect and protect against danger [21]. In recent work this has meant perceiving danger through the semiotics of cellular distress [10].

In line with *Aicklin et al* [22] it is necessary to move away from the mapping of objects to self/non-self. So non-self will not necessarily cause an immune response whereas the presence of danger signals will provoke a reaction. The self/non-self distinction can still be useful but it may now be viewed as a signal within Danger Theory. Furthermore a danger zone around an object emitting the danger signal can cause the immune response so that objects in proximity to danger with matching receptors participate in the stimulation of antibodies. Proximity in this case not only means spatial nearness but is a binary relation between objects that may, for example, signify file size, time stamp, connection or use of similar resources.

In order to advance these notions it is proposed to use the system's exposure to danger as a stimulus to shape its immune responses. Thus the system starts with a grounded signal of danger, specified as innate immunity, consisting of sensor readings mapped to unique concepts. So a concept takes on the role of an antigen receptor. As the system functions the instrumentation is read and the results assessed for concept mappings. If the system advances into unknown states then either a model driven simulation can be used to predict any fatal consequence and thus obtain a new receptor for danger or, given sufficient diversity, a process can be allowed to run and if it fails the danger receptor can be isolated. Thus new self can emerge and cognitive immunity be established.

## 3. Stochastic Situation Calculus and Danger

The situation calculus presented in [11] formalizes the behaviour of dynamically changing systems. It provides a particularly useful instrument to model autonomous, mobile, distributed applications, including the capturing and handling of novelty manifested as danger signals. The situation calculus is derived from the original formulation of [23]. The formalism is based on the notion of a situation, a snapshot of the state of the world. Each situation is

defined by the value of the fluents; the situation dependent functions and predicates, in the situation. A situation is transformed to a new situation by a named action that changes the value of one or more fluents. So there are three basic sorts:

- **Situations:** which all emanate from an initial situation,  $S_0$ , where no actions have yet occurred. A possible history is a sequence of actions called a situation.
- **Fluents:** are relations or functions where truth or function values change from situation to situation. They are denoted by function symbols with a situation term as their final argument
- **Actions:** change one situation to its successor situation. Each named action has an action precondition axiom stating the conditions under which the action can occur. In simple cases these can be single actions with a linear ordering. However concurrency and time can also be introduced to the representation [12].

It is necessary to state how the actions affect the world via effect axioms. These represent the change in value of a fluent when an action causes the situation to change.

However to reason about change in the system these axioms are not sufficient. It is necessary to add frame axioms that state when fluents remain unchanged by actions.

This gives rise to the frame problem [24]. The solution is to combine the frame and effect axioms into a single successor state axiom [25].

i.e.  $TRUE$  in next situation  $\Leftrightarrow$  (an action occurred to make it  $TRUE$ )  $\vee$  ( $TRUE$  in current situation and no action occurred to make  $FALSE$ ).

An action is initially specified by stating the conditions under which it can be performed.

$poss(A(x_1, x_2, \dots, x_n), s) \Leftrightarrow F_A(x_1, x_2, \dots, x_n, s)$   
 where  $F_A(x_1, x_2, \dots, x_n, s)$  is an expression specifying the preconditions for the action  $A(x_1, x_2, \dots, x_n)$ .

To encompass non-determinism the stochastic situation calculus [4] consists of the usual situation calculus plus a nature's choice space,  $C_0$ . This is the set of the sets consisting of the choices for each stochastic action. This has the property that for  $X_1, X_2 \in C_0$  and  $X_1 \neq X_2$  then  $X_1 \cap X_2 = \emptyset$ . A member of the choice space is called an *alternative* an element of the alternative is an *atomic choice*. So for example

$\{succeeds\_action(x_1, \dots, x_n), fails\_action(x_1, \dots, x_n)\} \in C_0$

There is a probability function,  $prob_o: C_0 \rightarrow [0, 1]$  such that  $\forall X \in C_0, \sum_{\alpha \in X} P_o(\alpha) = 1$

This is a probability measure of the choices controlled by nature. This is the usual standard probability definition. The probability of a proposition is the sum of the probabilities of the situations where it is true and the probability of a situation is the product of the probabilities of the atomic choices that are true in that situation. Thus the atomic

choices are probabilistically independent. This assumes a finite choice space.

This provides a rich formal language for representing and reasoning about dynamic systems. In particular it is possible to formulate danger precondition axioms whereby cognitive immunity is imparted to the system via abduction of pre-danger situations. These can be characterised by  $danger(succeeds\_a, s)$  meaning the successful enactment of action  $a$  is dangerous to perform in situation  $s$ . In general a danger precondition axiom is a sentence of the form

$danger(succeeds\_A(a_1, a_2, \dots, a_n), s) \equiv \prod_A(a_1, a_2, \dots, a_n, s)$   
 where  $A$  is an n-ary action function symbol and  $\prod_A(a_1, a_2, \dots, a_n, s)$  is a formula that is uniform in  $s$ . To be uniform in  $s$ , in this case, means that  $a_1, a_2, \dots, a_n$  do not mention terms of the sort *situation*. This ensures that the danger preconditions for the action  $A(a_1, a_2, \dots, a_n)$  are determined by the current situation,  $s$ , not by any other situation.

#### 4. Danger Situation Calculus

In modelling a system in the Situation Calculus a flexible representation of dynamic systems is gained. There is no requirement for the explicit enumeration of the tied (situation) space and action or events are not explicitly tied to specific time points. Using a Danger Theory perspective with the situation calculus allows the evolution of a dynamic self. Reasoning can then be performed on the logical representation to supply receptors for perceived danger signals (antibodies for known foreign antigens). In this way new interactions that cause no harm, and may be beneficial, are allowed.

In the first case the danger signal model used can be unavailability of service, within the system. Thus a locally unavailable service can trigger an immune response, at that location, that can later be used across the whole system. The immune response, from the system, would take the form of receptor (antibody) generation for the danger. In the context of the Danger Theory model for the human immune system the unavailability of a service corresponds to the danger signal generated by cells in the process of non-apoptotic necrosis. That is anything that causes abnormal cell stress.

As a facet of the evolutionary process certain situations may become apparent where a threat or uncertain state can be deduced. In order to characterise this situation a novel fluent may need to be established within the running model. At the point of system evolution where this occurs the snapshot then contains a new fluent. The model is thus updated and the initial value of the new fluent is false in the initial situation carrying through to where it is triggered in the system's evolution. In this way an unknown situation is confronted by the system encountering the new or non-determined situation, via receptors, which have previously

been defined as concepts based on the results obtained from system instrumentation.

In the situation calculus the knowledge gained in this manner does not affect any fluents other than the knowledge fluent [26] and non-knowledge producing actions only affect the knowledge fluent as appropriate. Additionally memory emerges because something known in a certain situation remains known in successor situations unless a relevant feature changes.

The result of the system running triggers a specific configuration of receptors to be matched by behaviour that can be termed threatening. These receptors are in turn just defined concepts, which may be innocuous on their own but occurring together form the basis of a new non-determined situational model.

So, for example, as the system evolves, there may be fluents monitoring for heavy CPU load or unresponsive behaviour in a situation:

$heavyLoad(service, s)$  or  $\neg responsive(service, s)$

with corresponding remedial actions. These can be thought of as innate immunity. However a situation may be encountered where there is a lack of information giving rise to a non determined state (termed here *critical*). This triggers a system's identification process of the critical state. So we may call this

$critical(service, s)$ .

Now in the regular Situation Calculus the successor state axioms take the form of, for example:

$poss(a,s) \wedge holding(letter, do(a,s)) \Leftrightarrow [holding(letter,s) \wedge a \neq drop(letter)] \vee a = pickup(letter)$

in the case of a robot mail delivery system where  $poss(a,s)$  is the action precondition axioms expressed logically,

e.g  $poss(pickup(letter),s) \Rightarrow$

$in(mailRoom, robot) \wedge numberItemsCarried \neq maxPayload$

Here, for most reasoning purposes, the values of the fluents in the next situation are deduced. However if the new type is introduced into the Situation Calculus, the danger precondition axiom:  $\neg danger(succeeds\_a,s)$ , a successor state axioms can be formulated:

$danger(succeeds\_a,s) \wedge poss(succeeds\_a,s) \wedge \neg available(service, do(a,s)) \Leftrightarrow critical(service,s)$

$poss(a,s) \wedge critical(service, do(a,s)) \Leftrightarrow (\neg available(service,s) \wedge critical(service,s) \wedge a \neq immuneResponse) \vee (available(service,s) \wedge danger(a,s))$

In this case a dangerous history can be abduced from the individual service running to give a proscribed action history or antibody for a particular danger antigen or critical abnormal behaviour.

The implementation of these results is currently achieved through an appropriate introspective scripting language, Neptune, for autonomic systems programming [27, 28]. As illustrated in figure 1, the system controller monitors service states through a distributed tuple-space. The self-governance is achieved through runtime adaptable Neptune objects. Thus the system controller may use the sensing results and danger signal responses to modify the runtime components without the need to stop and restart the system. Additionally the architecture provides the meta-structure whereby the self-governance is separated from the service concerns. Danger receptors can therefore be incorporated

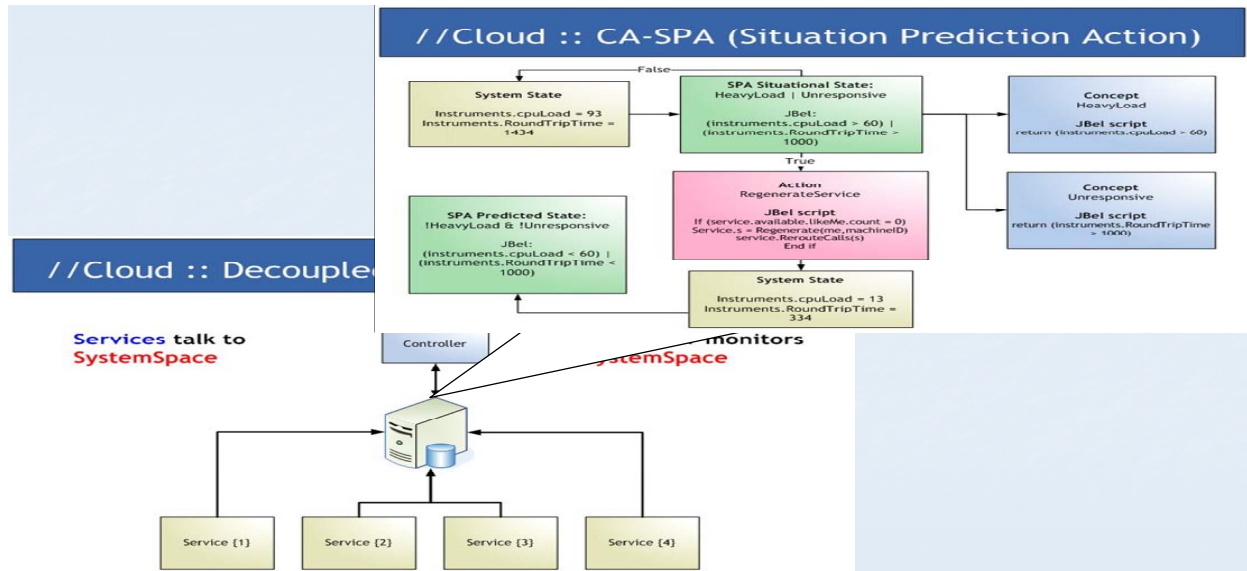


Figure 1: Self-Governance through Neptune Objects in the Clouds Architecture

into the self-governance model, as a result of the deliberation process, without any system interruption.

## 5. Case Study

The specification and details involved in a Neptune implementation, within the Clouds architecture, is beyond the scope of this paper [27, 28]. However there is a direct route from calculus to code making the logical calculus specification synonymous with executable governance structures [6]. Deliberative logic is defined through the Situation Calculus into the Neptune language with system co-ordination and storage of the common base made through the Clouds architecture. Once published onto the Cloud, a concept or action when referenced is located and deliberated upon to produce a result that determines the future action. This deliberation is controlled and specified by the underlying logical formalism provided by the Situation Calculus model adapted via the derived danger signals Vital signs of the system state are obtained via sensing actions, in the Situation Calculus, produced from the instrumentation, in the runtime system.

The deliberative functional language Neptune is used to combine concepts (the results of sensing in the Situation Calculus) and actions to produce formalised states (situations) as well as to allow actions to be executed by the system as guided by the reasoning capabilities in the logic or the perceived threat or danger.

The Neptune instrument framework exposes sensors at both a machine level and component level to the language and script. Neptune Instruments are wrappers of the Windows operating system performance counters, meaning that any performance counter is exposed to Neptune. These are used to assess the level of criticality based on the derived danger model.

This work has been applied to provide a robust self-governance meta-system in a medical decision support system [29]. For example the medical treatment decision support system's *core concern* delivers a decision based on matching patient variables to known treatments. However the final decision on which treatment to follow rests with the doctor. This raises a quality of process concern as a danger signal. There are two reasons why this situation may be a precursor of system danger. Firstly the doctor may have chosen a risky treatment path that needs more investigation. Secondly the system decision may need updating to be more in line with the current clinical practices if enough practitioners are using different treatments to the recommended one. So, using the Situation Calculus, a successor state axiom to monitor the state of compliance may be stated:

$$\begin{aligned} \text{compliance}(\text{patient}, \text{service}, t, \text{do}(a, s)) \Leftrightarrow & \\ & [\text{compliance}(\text{patient}, \text{service}, t, s) \wedge \\ & \neg \exists t_1 a = \text{treatment\_decision}(\text{patient}, t_1)] \vee \\ & \exists id [a = \text{treatment\_decision}(\text{patient}, t) \wedge \\ & \text{system\_decision}(id, t, s)] \\ & [id = (\text{session}, \text{doctor}, \text{patient}, \text{service})] \\ & \text{with } \text{poss}(\text{treatment\_decision}(\text{patient}, t), s) \Rightarrow \text{True} \end{aligned}$$

Thus the action history that led to the decision is encoded within the logical statements. Furthermore this history can be abducted and used to alert of danger in subsequent similar circumstances. So, for instance in future uses the system's output may be varied to take account of changes in medical knowledge or the doctor user may be alerted to a mistake previously detected, queried, verified and stored as an error by the system. This may be implemented via the Neptune script applied to the NICE guidelines:

```
rule compliance
{
    treatmentDecision t =
"cloud://Services.DecisionModel.NICE";
    instrument i =
"cloud://Instruments.DecisionModel.NICEResult";

    if (t.treatmentDecision(patient) !=
t.systemDecision(patient))
    {
        //flag that there is a guideline
decision conflict
        i["DecisionConflict"] = true;
    }
}
```

Similar structures are used detect and extract histories that caused, for example CPU overload or unresponsive behaviour. Indeed any sequence of actions that led to a diminution of system performance can be used as a receptor for the danger.

## 6. Conclusion

The formal treatment presented here, in line with the application of danger theory to computer systems, is an early attempt to formalize danger signals. The main result provided is the use of the stochastic situation calculus language for abducting *interesting* action histories (situations) for use in system adaptation via the newly developed Neptune language. Also a part of this research is to develop tools to support the specification, analysis and refinement of danger signals to inform system adaptation. The correctness of such models follows from the logical specification with the resulting direct and verifiable

implementation, in Neptune, easily achieved. Further research is underway to better identify the vital system components, which require monitoring, and their dependencies to better abduct danger receptors for the system.

## Acknowledgement

This paper has been partially supported by EPSRC grant reference:GR/R86782/01. 2nrich Project URL: <http://www.cms.livjm.ac.uk/2nrich/>

## References

- M.G. Hinchey, J.L. Rash, C.A. Rouff, "Requirements to Design to Code: Towards a Fully Formal Approach to Automatic Code Generation" NASA Technical Report, January 2005
- A. Aviziensis, J-C Laprie, B. Randell, C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing" IEEE Transactions on Dependable and Secure Computing 1(1) Jan-Mar 2004
- J.L. Rash, C.A. Rouff, M.G. Hinchey, "Experience Using Formal Methods for Specifying an Agent Based System" Proceedings of Sixth International Conference on Engineering of Complex Computer Systems (ICECCS 2000), Tokyo, Japan, 2000
- M. Shackleton, F. Saffre, R. Tateson, E. Bonsma, C. Roadknight "Autonomic Computing for Pervasive ICT — A Whole-System Perspective", BT Technology Journal, Vol.22 No.3 pp 191-199, July 2004
- M. S. Sloman, "Policy Driven Management for Distributed Systems," *Journal of network and Systems Management*, vol. 2, pp. 333-360, 1994.
- M. Randles, A. Taleb-Bendiab, P. Miseldine, A. Laws "Adjustable Deliberation for Self-Managing Systems", To appear in proc. *12th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS2005)*, Washington MD, April 2005
- IBM Research. *Autonomic Computing*. <http://www.research.ibm.com/autonomic>. Accessed March 2005
- M. Randles "Applications of Situation Calculus to Distributed Self-Adaptive Middleware" *MSc. Thesis* Liverpool John Moores University, 2003
- R. Reiter, "Stochastic Actions, Probabilities and Markov Decision Processes in Situation Calculus". *Proceeding of the 18th National Conference on AI*, Edmonton, Alberta, Canada July 2002
- P. Matzinger "An Innate Sense of Danger" *Semin. Immunology* 10:399, 1998
- H. J. Levesque, F. Pirri and R. Reiter (1998) 'Foundations for the situation calculus'. *Linköping Electronic Articles in Computer and Information Science*, Vol. 3(1998): nr18. <http://www.ep.liu.se/ea/cis/1998/018/>
- R. Reiter "Natural actions, concurrency and continuous time in the situation calculus". *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR '96)*, ed: L. C. Aiello, J. Doyle and S. C. Shapiro, pp 2-13. Morgan Kaufmann Publishers, San Francisco, California, 1996
- H. Levesque "What is planning in the presence of sensing?" in *Proc. of AAAI-96 Conference*, Portland, OR, Aug. 1996, 1139-1146.
- F. Pirri and R. Reiter 'Some contributions to the metatheory of the situation calculus'. *Journal of the ACM* 46(3), pp 325-361, 1999
- S. George, D. Evans, L. Davidson "A Biologically Inspired Programming Model for Self-Healing Systems" *Proc of WOSS'02* pp102-104, Charleton, SC, USA, Nov 18-19, 2002
- Burnett and Fenner "The Production of Antibodies" Macmillan, London, 1949
- R.E. Vance, "Cutting Edge Commentary: A Copernican Revolution? Doubts about the Danger Theory" *The Journal of Immunology* 165 pp1725-1728, 2000
- P. Matzinger, "The Danger Model: A Renewed Sense of Self", *Science* 296 pp301-305, 2002
- S. Stepney, R.E. Smith, J. Timmis, A.M. Tyrrell "Towards a Conceptual Framework for Artificial Immune Systems", In G. Nicosia, V. Cutello, P.J. Bentley, J. Timmis (editors) *ICARIS 2004: Third Intl. Conference on Artificial Immune Systems, LNCS 3239* pp53-64, Springer 2004
- C. A. Janeway 'The Immune System Evolved to Discriminate Infectious Nonself from Noninfectious Self' *Immunology Today* Vol.13.11, 1992
- P. Matzinger 'Tolerance, Danger and the Extended Family' *Annual Review Immunology* 12:991, 1994
- U. Aicklen, P. Bentley, S. Cayzer, J. Kim, J. McLeod "Danger Theory: The Link Between AIS and IDS?" In J. Timmis, P. Bentley, E. Hart (editors) *LNCS 2787* pp156-167, Springer, 2003
- J. McCarthy (1963) "Situations, actions and causal laws". Technical report, Stanford University. Reprinted in *Semantic Information Processing*, ed: M. Minsky, pp 410-417, MIT Press, Cambridge, Massachusetts, 1968.
- J. McCarthy and P. Hayes (1969) "Some philosophical problems from the standpoint of artificial intelligence". *Machine Intelligence 4*, ed: B. Meltzer and D. Michie, pp 463-502, Edinburgh University Press, Edinburgh, Scotland.
- R. Reiter (1991) "The frame problem in the situation calculus: a simple solution (sometimes) and a complete result for goal regression". *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, ed: V. Lifschitz, pp359-380, Academic Press, San Diego, California.
- R.B. Scherl, H.J. Levesque, "Knowledge, Action and the Frame Problem", *Artificial Intelligence* 144 pp:1-39, 2003
- P. Miseldine "Cloud Architecture Schematic" <http://www.cms.livjm.ac.uk/2nrich/> (Accessed 03/ 05)
- P. Miseldine "The Neptune User Guide" <http://www.cms.livjm.ac.uk/2nrich/> (Accessed 03/ 05)
- The 2nrich Project, Liverpool John Moores University <http://www.cms.livjm.ac.uk/2nrich/>