Signal Regulated Systems and Networks

Terence L van Zyl, Elizabeth M Ehlers

Academy of Computer Science, University of Johannesburg, University Rd., Auckland Park, South Africa, tvzyl@csir.co.za, emehlers@uj.ac.za

Abstract

The paper presents the use of signal regulatory networks, a biologically-inspired model based on gene regulatory networks. Signal regulatory networks are a way of understanding a class of self-organising IT systems, signal regulated systems. The paper builds on the theory of signal regulated systems and introduces some formalisms to clarify the discussion. An exemplar signal regulated system that can be evaluated using signal regulatory networks is presented. Finally an implementation of an adaptive and robust solution, built on a theory of signal regulated systems and analysed as a signal regulatory network, is shown to be plausible.

1 Introduction

Self-organising complex adaptive systems (CAS) exhibit many favourable characteristics including extensive robustness, scalability and adaptability [1]. These characteristics are desirable in many other systems such as software systems. Repeated themes begin to emerge as self-organising CAS are explored. One of these themes is the existence of self-regulating signalling systems as an underlying enabler of self-organisation [2, 3, 4, 5, 6]. Examples of self-organising signal-regulatory systems include stigmergy, gene-regulatory systems, signal-transduction networks, and neural networks [7]. Of particular interest are those cases where nature has used these signal-regulatory systems as a processor of information [8]. The paper describes research into the use of one of these information processing signal regulatory systems, gene regulation, as a solution to the complexity challenge of information technology (IT) systems [3, 9].

A key characteristic of self-organising systems is that they exhibit a decrease in entropy [10, 11]. Often self-organisation, as in a decrease in entropy, at the macro level comes at the cost of an increase in entropy at the micro level [7, 12]. In addition, self-organising systems are often characterised by flows of energy. These energy flows result in the system not stabilising at equilibrium but rather at an attractor or steady state far from it [13, 12]. Further, self-organising systems are dissipative in that the continuous flows of energy move entropy outside of the system [14]. In the case of information systems the energy takes the form of datum [7].

A further condition for self-organisation is non-linearity. Systems consisting of a large number of autonomous components interacting in a distributed and decentralised manner are a characteristic of such non-linearities and are referred to as complex adaptive systems [15, 16, 17].

Complex IT systems are increasingly common as a result of the large number of distributed and decentralised, interacting components within these systems [18]. That engineers try to control these systems in a top down manner comes as no surprise [19]. IT systems are built for a specific purpose and are expected to behave within certain parameters. In order to ensure the correct functioning of these systems, engineers try to write code that accounts for all eventualities. The net effect of this additional code is that it adds to the original problem with systems remaining brittle lacking both adaptability and robustness [20, 21].

In addition there is a growing realisation that complex IT systems are beyond top down orchestration [20, 22]. Further as the complexity of the components, their sheer quantity, and the extent of their interaction grows, both self-organising and emergent characteristics are likely to appear [23].

Various solutions have been proposed to mitigate the inherent complexity of a large software systems, the most prevalent of these being autonomic computing [24]. Autonomic computing tries in part to replicate the efforts of control theory within the field of software engineering [25]. Adaptability is explicitly coded with alternate paths for each eventuality [26, 22]. This results in a control that is at least as complex as the plant that it is trying to regulate and an overall system equally if not more complex.

An alternate solution to the complexity challenge is to use the principles of self-organisation in the engineering of complex systems. It is theorised that using self-organising principles will help in overcoming some of the challenges faced in the development of complex systems[27, 28]. One self-organising solution proposed by Holland is that of complex signal networks [29, 30, 31]. Holland proposed the implementation of these networks through the use of *Classifier Systems* and the *Holland Broadcast Language* [32]. The idea of *Artificial Signal Networks* has been extended by Decraene [**?**, 33]. The work presented here builds on this body of knowledge.

The research problem is twofold. On one hand building IT systems that exhibit the properties of self-organising CAS is required. On the other hand IT systems may unintentionally exhibit emergent characteristics just by virtue of their complexity. What is required are the tools and abstractions to explicate the self-organising characteristics of IT systems and in so doing allow for proper analysis and design.

In order to facilitate the creation of CAS, that are self-organising, engineers need to realise an environment, with relevant abstractions and interactions, that allows self-organisation to occur. These abstractions include non-linearity, attractors as drivers of function rather than state variables, and explicit flows of information [10, 34, 21]. Further, a system that is self-organising should be underpinned by a signal regulatory mechanism and should be understood in terms of its signal landscape [29, 30]. The paper explores signal regulatory networks, as a model of signal regulated systems (SRS), that explicitly realise these self-organising requirements.

In order to test the theory of SRS and signal regulatory networks a prototype system was designed and implemented. The resultant system is shown to be both adaptive and robust whilst explicitly adhering to the characteristics of self-organising systems.

Section 2 describes the processes used by nature to enable cell functioning and how these processes can be modelled as gene regulatory networks. In section 3, the biological inspiration taken from gene regulatory networks is presented from a computer science and information theoretical perspective in the form of signal regulatory networks and signal regulation machines. Section 4 formalises the notion of SRS. Section 5 describes the method. Section 6 looks at an implementation of a signal regulatory network and machine example. The paper is concluded in section 7.

2 Gene Regulatory Networks

This section provides background to the theory presented in the paper. Firstly, how biological systems maintain their existence through protein production is discussed. Secondly, the mechanism of protein production is linked to gene regulation, the cell's DNA and how these in combination provide a self-organising system that underpins the creation and maintenance of cells.

2.1 Protein Production In The Cell

A cell consists of a number of heterogeneous proteins functioning together to form a single cohesive entity. These proteins enable all cell function including replication, migration, differentiation, maintaining form, messaging and reacting to external and internal stimulus. In the simplest model, the cell's nucleus enables protein production through DNA transcription into RNA and from RNA translation into protein [35]. The exact form, molecular structure and chemical reactions that enable the DNA to protein transition are not discussed here. The simplified process is described as follows.

The DNA consists of a four coded base sequence that, once executed through the correct cell processes, produces a specific protein. In the first instance the subsequence of DNA that codes for a specific protein is transcribed, by RNA polymerase, base by base to the more manageable form, messenger RNA (mRNA). The mRNA then acts as a tape that is translated together with transfer RNA (tRNA) into proteins by the ribosomes [36, 37]. For a detailed discussion the reader is referred to [35].

Being able to produce proteins is not sufficient for cellular existence. What is required is that the correct protein types are produced timeously and in the required quantity. This orchestrated protein production is enabled by the cell's DNA. The DNA contains the template for the production of proteins and instructions for when, which, where, and how much of those proteins should be produced. These characteristics of DNA ultimately direct the type and function of the cell [35].

How a cell regulates protein production involves a set of proteins known as transcription factors (TF). TF bind to the DNA upstream from the coding sequence and are able to promote or inhibit the transcription of further mRNA sequences including further copies of themselves. Another place where regulation occurs is at the point where the mRNA is translated into the protein. Regulating the production of proteins by proteins is termed gene regulation [35]. When looking at cell function in terms of gene regulation, it makes sense to think of proteins as computational elements that form part of an information processing network [8, 38]. The idea of an information processing network is further explored below.

2.2 Information Processing Networks

The ability of a protein to regulate the production of further proteins including proteins of the same type as itself, allows for a complex network of regulatory relationships between proteins. The resultant network forms various positive and negative feedback loops, cascades and sequences that are modelled as a gene regulatory network (GRN) [4].

DNA is not only acted upon by proteins created within the cell but by external entities as well. These external entities include proteins produced by other local cells or even transported from other parts of the organism in the form of hormones. External entities to the organism such as various chemicals, enzymes, and environmental factors, like radiation, may also act indirectly upon the cell's DNA so as to influence the GRN [39].

In the oversimplified view presented here it should not be misconstrued that the DNA provides some centralised control of cell function. Rather it should be understood that a dynamic, non-linear system, consisting of many interacting parts, that influences and is influenced by its environment, is being described in a simplified model [8].

Abstracting the above system of gene regulation so that all the specific detail surrounding chemical processes are removed describes a mechanism whereby genes regulate genes. Further abstraction away from the physical phenomenon of proteins and specifically the idea of genes, allows the process to be viewed as one in which signals regulate signals, where a gene is seen as a message or signal. Signals regulating signals is a core idea that underpins the concept of signal regulatory networks, expounded in later sections. How this process of signals regulating signals enables the self-organisation of cells is discussed in the following section.

2.3 Self-organisation Creating Cells

In the case of cells, self-organisation is brought about by a reduction in entropy at the macro level as a result of an increase in entropy in the form of signals (proteins) at the micro level. Various feedback loops and the continuous flow of information through the cell, allow self-organisation to occur at on of the system's "far from equilibrium" steady state attractors [14]. Genomics theorises that the various states that a cell may find itself in and the various functions that a cell undertakes can be viewed as the attractors in its gene regulatory network [36, 40, 41, 42, 43]. The function of the cell is not described in terms of a set of states and state transitions. Rather the function is described in terms of a gene regulatory network or gene landscape where the attractors in the landscape can be linked to the more mechanistic view presented by cell states [43, 44].

It should be noted that the idea of gene regulatory networks and other biological regulatory networks have been cast in terms of information theory previously [45, 46].

3 Signal Regulated Systems, Networks And Machines

A signal regulatory network (SRN) mimics the key concepts that underpin a GRN. Like a GRN, a SRN allows us to model a class of systems, signal regulated systems (SRS), that exhibit self-organising characteristics. The network abstraction is as a result of the regulatory relationships between signals. This regulatory relationship is best described by stating: signals may in combination with other signals regulate the rate of production of further signals including themselves.

Definition 1. A signal is any temporal- and, possibly, spatial-varying quantity that carries information.

Signals can exhibit both temporal and spatial variation, examples of these being dissipation and dispersion. In the context of information systems the spatial aspect of a signal can be used to identify the container of the signal or some virtual cell where the signal is located. The spatial variation of the signal describes how the signal changes spatially, e.g. the inverse law in a field. The temporal variation of a signal describes how the signal changes with time e.g. duplication into adjacent signal containers or decreasing and disappearing with time. The properties of SRS that can be modelled as a SRN are:

1. There exists a process that produces signals of various types.

2. The dynamics of the system stem from the fact that the signals produced into the environment at time t are able to regulate the signals produced at time $t+\Delta t$. Regulating includes both promoting and inhibiting the rate of signal production along with other spatial and temporal properties of the signal.

3. A signal may directly or indirectly promote or inhibit the production of further signals of its own type (cyclic relationships between signals) and thus produce positive and negative feedback loops.

4. The relationship between the signals produced at time $t+\Delta t$ and those at time t may not be just trivial one to one or linear mappings. For example it may hold that signal S_1 promotes a linear increase in the production of signal S_3. Signal S_2 promotes a exponential increase in the production of S_3. However combing both signal S_1 and signal S_2 completely stops the production of signal S_3.

5. SRS are self-organising systems resulting from the dynamic interplay of signals. A signal regulated system's behaviour is characterised by moving between steady states which represent attractors in the signal space.

Rich signal interactions, together with positive and negative feedback, allow for additional properties including multi-stability, oscillations, state-dependent responses, echoes, randomness, and signal cascades in which a relatively small signal can elicit a large response through signal amplification in various signalling pathways or chains [47]. The continuous flow of signals in their various complex networks of interaction form a signal landscape or space that is characterised by various attractors. Attractors may be fixed points, bi-stabilities, multi-stabilities, cyclic, and limit cycles[16].

Definition 2. A signal regulation machine (SRM) is a process that is able to perform the function "signal in signal out" as required by SRS. All SRS consist of one or more interacting signal regulation machines.

3.1 From State Based To Attractor Based Systems

Typical views of IT systems and the abstractions used in their design, are modelled around their states and the transitions between these states. Many self-organising and multi-agent systems are also viewed as state based machines. The state based view is amplified in the models used to understand these systems where tools such as Markov models, Bayesian Networks, and Cellular Automate are examples.

A departure from state as the primary abstraction in IT systems is presented in signal based systems. Here the primary abstraction driving function is not state variables but rather attractors. In a state based system a change in the system's variables indicate a change in the system's state. In an attractor based system the system's variables should be continuously changing so that the system can be in an attractor. If the variables dynamics should stop this would indicate that the flow of energy has ceased and as such the far from equilibrium steady state indicative of an attractor would cease. In order for a attractor based system to change attractors a large enough perturbation is required to move the system from its current attractor into another. The view presented here is consistent with the

"Being versus Becoming" view presented by Priogogine and others in which it is theorised that systems need to be understood in terms of process and not state [21, 34].

SRS not only allow for the explicit creation of an entropy sink in self-organising systems but also allows for a more grounded view of self-organising systems. The functionality of the system is a by-product of the signal landscape as with cells and gene regulation. Further to this there are a number of tools that can be used to analyse SRS include Boolean Networks, Probabilistic Boolean Networks, Rate Differential Equations, and Stochastic Models [48]. The tools also allow us to assess the existence and stability of various attractors within the signal landscape. Whether those attractors are by design or a by product of complexity is not important. Other tools such as graph theory include the ability to assess robustness of the network, or potentially sensitive nodes, using toplogy analysis [49, 50, 51]. Further, modelling as a graph allows for a topological analysis of the attractors and a qualitative assessment of their stability amongst other mathematical analysis [52, 53].

4 Formalisation Of Signal Regulatory Networks

A SRN can be formalised as follows. N signals are denoted by:

$$(\sigma_1, \sigma_2, \dots \sigma_N) \qquad i = 1, 2, \dots, N$$

where

$$\sigma_i \to g_i(\bar{x}, t) \qquad i = 1, 2, \dots, N$$

The function g_i takes this form since signals are temporal and spatial varying quantities where $\bar{x}$ is a n-dimensional spatial vector and t is a temporal scalar. In the simplest case $\dot{g}_i = 0$ I.e. doesn't change with respect to time. Here g_i takes on one of two constant values $\{0, 1\}$ and has no spatial varying attributes, this is the same as the case modelled by a Boolean Network. The regulation of a signal is time dependant such that the value of σ_i at the following time step is regulated by K regulatory signals $\sigma_{i_1}, \sigma_{i_2}, \dots, \sigma_{i_k}$ where the discrete temporal evolution is

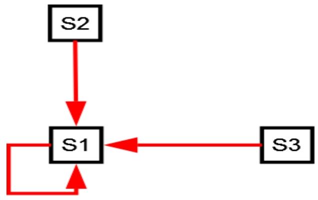

Fig. 1: *A basic SRN showing three signals, where the signal s1 is regulated by three signals including itself.*

determined by:

$$\sigma_i(t+1) = f_i(\sigma_{i_1}(t), \sigma_{i_2}(t), \ldots, \sigma_{i_k}(t)) \tag{1}$$

The function f_i may be stochastic in nature as is often the case in self-organising systems. Further it is noted that one of the K regulatory signals may be σ_i itself. A set R of J equations of the form given by equation (1) model a discrete SRS:

$$R = \begin{cases} \sigma_1(t+1) = f_1\left(\sigma_{1_1}(t), \sigma_{1_2}(t), \ldots, \sigma_{1_k}(t)\right) \\ \sigma_2(t+1) = f_2\left(\sigma_{2_1}(t), \sigma_{2_2}(t), \ldots, \sigma_{2_k}(t)\right) \\ \vdots \\ \sigma_J(t+1) = f_J(\sigma_{J_1}(t), \sigma_{J_2}(t), \ldots, \sigma_{J_k}(t)) \end{cases} \tag{2}$$

Different signal regulation machines may have different functions modelling the time evolution of the same signal σ_i in this case a SRN across these signal regulation machines would provide a superposition of the regulatory effect of the various machines.

A signal regulatory network can be modelled as a directed graph:

$$G = \{V, E\}$$

The directed graph consists of $V \in \{\Sigma, R\}$ where Σ is the set of signals, and R is the set of regulatory functions, and $E \in \{V \times V\}$ is the set of ordered pairs representing the directed regulatory relationships. Where regulatory edges enters a signal vertex some function from R of those regulatory signals defines the regulatory effect. Figure 1 shows a basic SRN where two input signals s_2 and s_3 regulate a self regulating signal s_1.

5 Method

In order to evaluate the effectiveness of systems that are underpinned by a signal regulatory mechanism an experiment is constructed. The experiment consists of first designing and understanding the attractors and characteristics of the signal landscape that underpin the intended system. A system is then built that uses the defined signal landscape as a basis for self-organisation. The system is allowed to run and data is collected as to the functioning of the system. During runtime the system is exposed to disruptions and stresses to assess its robustness, scalability and adaptability.

The data collected takes the form of the number of work items that each agent within the system is able to perform along with how many work items fail and are completed successfully. In addition the total CPU usage over the interval is also recorded. The data is then averaged over one minute intervals to remove excess noise. The data will be analysed to see how the system degrades as a result of disruptions, noise in the form of false signals and purposefully removed and added agents. The system will also be evaluated as to how well it is able to recover when those stresses are removed. Additionally the system will be run using different size communities to evaluate its scalability.

6 Implementation And Evaluation

The experiment is well suited to the distributed blackboard architecture of Cougaar [1]. There exist two observation source agents that face the web. The observation sources are named Observation Source 1 (OS1) and Observation Source 2 (OS2). Each observation source agent uses one or more observation processing agents to generate its observations. In the case of OS1 there are two possible observation processing agents to choose from, those being Observation Processor 1 (OP1) and Observation Processor 2 (OP2). OS2 has a single observation processing agent, Observation Processor 3 (OP3). Every observation processing agent uses various sensor resources as input. Sensor resources provide raw data about the environment. In our case OP1 uses a Sensor Resource 1 (SR1), and

[1] http://www.cougaar.org/

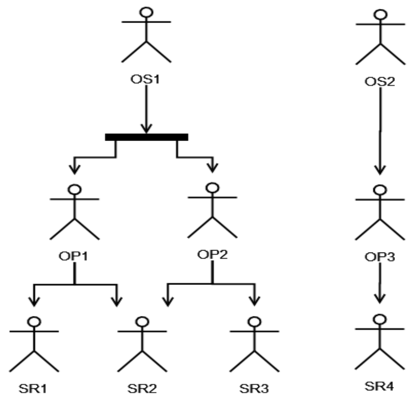

Fig. 2: *Describes the dependencies between the components encapsulated by each of the agents. The bold horizontal line indicates that OS1 can use either OP1 or OP2.*

Sensor Resource 2 (SR2). OP2 uses SR2 and Sensor Resource 3 (SR3). OP3 uses Sensor Resource 4 (SR4). The dependencies between agents described here is made clearer in figure 2.

The agents described here exist on the computing platforms, Platform-(1-5), and duplicates of each agent can exist on each platform.

In order to further restrict our prototype some constraints are made. Firstly the platforms are fully connected, that is every platform is connected to every other platform. Secondly Platform-1 and Platform-2 are outward facing, i.e. Internet facing, and thus deployments of OS1 and OS2 are constrained to these. Platform-3 and Platform-4 have the sensor resources connected to them such that sensor resources are constrained to these platforms. The above constraints leave deployments of OP1, OP2, and OP3 free to exist on any one of Platform-1 to Platform-5. Figure 3 shows a deployment diagram to clarify where the various agent types may exist.

For our purposes both OP1 and OP2 provide observation processing of the same quality. However OP1 is a CPU intensive algorithm and OP2 is a memory intensive algorithm.

The SRM present in each agent is tailored to the functionality that the agent encapsulates, is stochastic in nature and is called a Probabilistic-SRM (P-SRM). The P-SRM evaluates the signals in the agent's signal space and stochastically

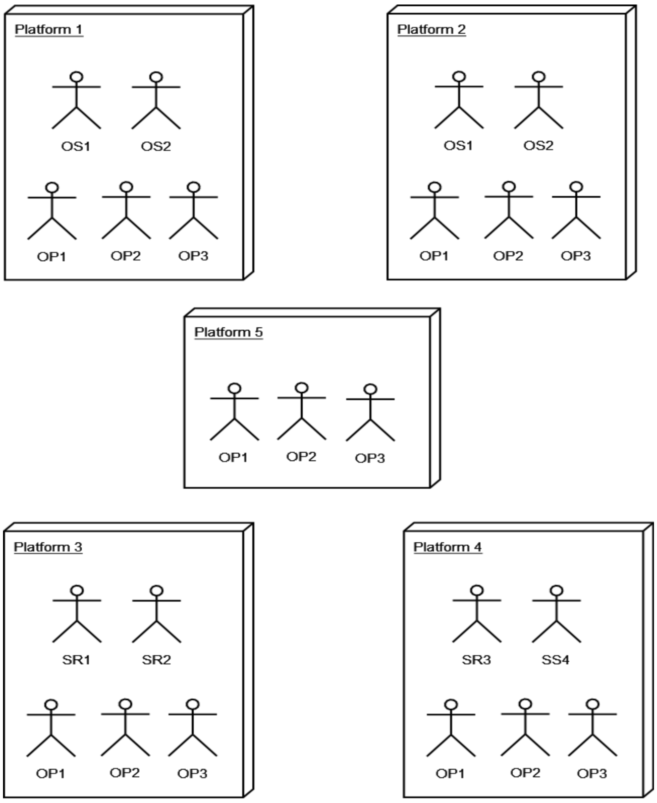

Fig. 3: *Deployment diagram for agent types. Agents are represented as UML actors so as to indicate their autonomy and to differentiate them from components.*

produces appropriate signals according to probabalistic rules. The produced signals may then be relayed to other agents' signal spaces according to the signals' spatial functions.

The signal regulatory system is composed of the following signal-types:

1. A_{os1}, A_{os2}, A_{op1}, A_{op2}, A_{op3}, A_{sr1}, A_{sr2}, A_{sr3}, A_{sr4}: The "available signal" indicates that an agent is free to provide its components service.

2. W_{os1}, W_{os2}, W_{op1}, W_{op2}, W_{op3}, W_{sr1}, W_{sr2}, W_{sr3}, W_{sr4}: The "work-request signal" is used by an agent to request work from another agent.

3. K_{os1}, K_{os2}, K_{op1}, K_{op2}, K_{op3}, K_{sr1}, K_{sr2}, K_{sr3}, K_{sr4}: The "keep-alive signal" is used to indicate that an agent is still busy with work that has been requested.

4. T_{os1}, T_{os2}, T_{op1}, T_{op2}, T_{op3}, T_{sr1}, T_{sr2}, T_{sr3}, T_{sr4}: The "time-out signal" indicates that a certain time has expired.

5. C_{os1}, C_{os2}, C_{op1}, C_{op2}, C_{op3}, C_{sr1}, C_{sr2}, C_{sr3}, C_{sr4}: The "complete signal" is produced when a work request completes.

6. F_{os1}, F_{os2}, F_{op1}, F_{op2}, F_{op3}, F_{sr1}, F_{sr2}, F_{sr3}, F_{sr4}: The "failed signal" is produced when a work request fails.

If a signal-type is a regulator of another signal-type then given our P-SRM the signal-type either increases or decreases the probability that the signal-type will be produced in the following time step. The symbol "+" is used to indicate that the existence of signals of signal-type σ increase the probability and the symbol "−" to indicate that the existence of signals of signal-type σ decrease the probability. Here is given for each signal-type the set of regulatory signal-types used by agents of type OP3:

$$\begin{aligned} A_{op3} &: \left\{-K_{op3}\right\} \\ W_{op3} &: \left\{+A_{op3};+W_{op3};+K_{op3};+F_{op3}\right\} \\ K_{op3} &: \left\{+A_{op3};+W_{op3};+K_{op3};-C_{op3}\right\} \\ C_{op3} &: \left\{+W_{op3};+K_{op3};+C_{op3}\right\} \\ T_{op3} &: \left\{+W_{op3};+T_{op3};-F_{op3}\right\} \\ F_{op3} &: \left\{+W_{op3};+T_{op3};+F_{op3}\right\} \end{aligned}$$

Figure 4 presents as a directed graph showing the signal regulatory networks modelled by the above functions and uses the same symbols, i.e. "+" to indicate increases probability and "−" to indicate decreases probability. Note that the edges do not indicate the details of regulatory function used but just the fact that the probability of producing a signal of the given type in the next time step is either decreased or increased by the presence of signals of the given signal-type.

In most cases many signal of a given type will need to be present, as a result of the probabilities involved, in order to ensure the production of signals of another given type in the following time step. In addition given that all signals dissipate with time it is essential that a signal of a given type be continuously produced if it is to effect the production of other signal-types. The probabilistic transitions increases the robustness of the SRM. In order for a set of signals to move the network from one attractor to another it is required that the signals be produced in sufficient quantity and for sufficient time so as to sustain the perturbation, otherwise the network will quickly fall back into its current attractor.

The stochastic nature of SRS and the requirement that many signals of a given type are needed parallels the situation found in GRN, where many proteins of a given type are needed to effect DNA production of further proteins.

The probabilistic functions of the P-SRM are an extension of the Holland Broadcast Language (HBL) [32]. HBL consists of a set of condition-action rules. A condition-action rule is a pair in which if the condition holds then the signal represented by the action is produced into the agent's signal container. In the P-SRM the condition is given by a function that matches a set of signals related by

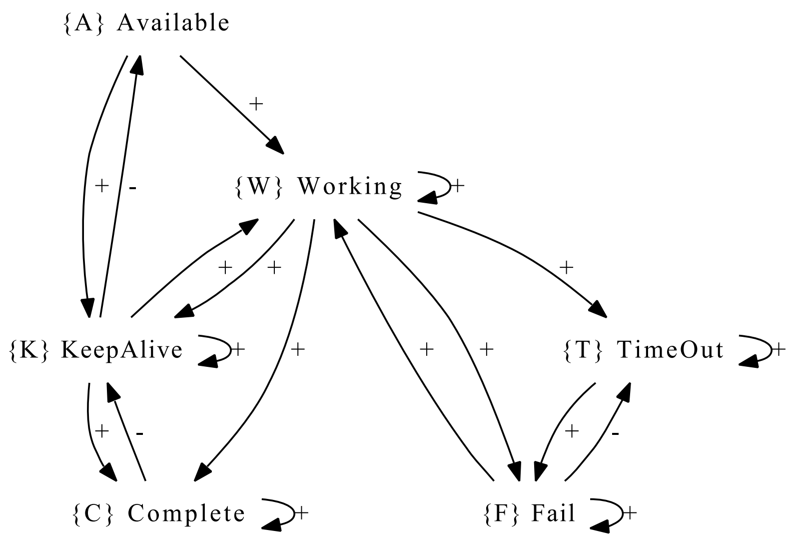

Fig. 4: *Observation Process 3 regulatory relationships between signals.*

probabilistic boolean operators. If the condition is satisfied the signal represented by the signal-type of the action is produced into the signal container with a given probability. A typical P-SRM rule would take the following form:

$$C(\Sigma_1, \Sigma_2, \ldots \Sigma_N) \rightarrow A(\sigma_i)$$

The probability is calculated based on the probabilistic condition function $C(\Sigma_1, \Sigma_2, \ldots \Sigma_N)$ and the signal represented by σ_i is produced with that probability in the next time step. The events $\Sigma_1, \Sigma_2, \ldots \Sigma_N = \Sigma_1(\sigma_1), \Sigma_2(\sigma_2), \ldots \Sigma_N(\sigma_N)$ of the condition are joined by probabilistic boolean operators for example:

$$C(\neg(\Sigma_1 \cup \Sigma_2) \cap \Sigma_2) \rightarrow A(\sigma_1)$$

Here $\cap$,$\cup$, and $\neg$ have there boolean meanings of "and", "or" and "not" respectively. The events $\Sigma_N(\sigma_N)$ are representative of the probability that the given signal σ_N would be matched[2345].

One approach to building an P-SRM would be given a signal container at time

[2] The probability of an event: $P(A) \in [0,1]$
[3] The "not" of an event: $\neg P(A) = 1 - P(A)$
[4] The "and" of two independent events: $P(A \cap B) = P(A)P(B)$
[5] The "or" of two non exclusively mutual events: $P(A \cup B) = P(A) + P(B) - P(A)P(B)$

step t run each signal rule of the P-SRM against every combination of signal in the signal container that match the condition. If the condition holds based on the probabilities of each event produce the signal represented by the action into the signal container at later time step $t+\Delta t$. Each rule should only fire once per a time step so it is necessary to perform the above operation until at least one signal is produced or there are no more signals to match the condition.

In order to reduce the computational overhead to $O\left(N^2\right)$ of the above P-SRM, a modified version that relies on the binomial probability is used. The general equation for binomial probability for getting *exactly* k successes in n trials is given by:

$$Q_n(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{3}$$

The P-SRM is concerned with the case where there is *at least* 1 success of matching the signals in the predicate in n trials for a given rule r. Where the n trial are performed on the signals in the signal container. That is the same as the case for not getting exactly 0 successes in n trial occurs, given by:

$$\begin{aligned} P_r(n) &= 1 - Q_n(0) \\ &= 1 - (1-p)^n \end{aligned}$$

In the implementation presented here all signals dissipate with a half life of one second, thus even dissipation is stochastic. An example of a rule for producing an available signal, where k is the number of K_{op3} signals in the signal container and $p = 0.3$, is given by:

$$\begin{aligned} C\left(\neg\Sigma\left(K_{op3} \cap K_{op3}\right)\right) &\rightarrow A\left(A_{op3}\right) \\ \therefore \neg P(n) \cup \neg P_r(n) &\rightarrow A\left(A_{op3}\right) \\ \therefore 2\left(0.7^n\right) - 0.7^{2n} &\rightarrow A\left(A_{op3}\right) \end{aligned}$$

Similar probabilistic rules exist for each of the signal types.

6.1 Signal Regulatory Network Boolean Networks Analysis

Boolean networks originally introduced by Kaufman provide a simple but effective mechanism for modelling gene regulatory networks [54]. They allow the expression of a protein by a specific gene to be in one of two states, either 1 or 0, akin to on and off respectively. The expression state of a gene g_i, either 1 or 0, at time $t+1$ is determined by a boolean function f_i that takes as input the expression state of K regulatory genes at time t. The regulatory genes may include g_i itself. Formula 4 shows a regulatory function for gene g_i. Given N such functions it is possible to construct a network $G = \{V, E\}$ where the vertices are the set of genes and the edges indicate the boolean regulatory relationship between them [7, 55].

$$g_i(t+1) = f_i(g_{1_i}(t), g_{2_i}(t), \ldots g_{k_i}(t)) \tag{4}$$

Boolean networks are useful as a tool to attain some qualitative information for a SRN. It is possible to ascertain what the various attractors are, how deep the basins are and how difficult it is to move from one basin to another.

To gain a better sense of the SRS presented above, the stochastic transitions of the P-SRM are removed and the set of equations are modelled as a boolean signal regulatory network as presented by these equations:

$$\begin{aligned}
A_{op3}(t+1) &= \neg K_{op3}(t) \\
W_{op3}(t+1) &= W_{op3}(t) \wedge \big(T_{op3}(t) \vee K_{op3}(t) \vee F_{op3}(t)\big) \\
K_{op3}(t+1) &= \big(A_{op3}(t) \wedge W_{op3}(t)\big) \vee \big(K_{op3}(t) \wedge \big(\neg C_{op3}(t) \vee W_{op3}(t)\big)\big) \\
C_{op3}(t+1) &= W_{op3}(t) \wedge K_{op3}(t) \wedge C_{op3}(t) \\
T_{op3}(t+1) &= W_{op3}(t) \wedge T_{op3}(t) \wedge \neg F_{op3}(t) \\
F_{op3}(t+1) &= W_{op3}(t) \wedge \big(T_{op3}(t) \vee F_{op3}(t)\big)
\end{aligned}$$

Figure 5 shows the boolean network for the above set of equations. What is immediately apparent in figure 5 is the existence of six attractors labelled 1 to 6. Attractor 1: [011100] has all signals off except for W_{op3}, K_{op3}, C_{op3} which are on. The attractor 1 is representative of a steady state indicating the agent has completed its task.

Note that the activation of the available signal, bit position one, immediately pulls the system back into attractor 1. However the introduction of either a failed or time out signal pushes the system into the failed and complete attractor as presented by 4: [011101]. Attractor 4 represents a decision point in the system as the agent must choose to accept or reject the completed work. If the agent rejects the work a transition into the failed attractor represented by Attractor 3: [011001] takes place. Attractor 2: [100000] representative of a steady state in which the agent is awaiting work and is a deep attractor and the only perturbations out of it are as a result of a work signal. Attractor 5 is the working attractor and requires both a work signal and a keep alive signal to maintain.

Attractor 6:[001000] is of some concern as it represents an attractor that can only be escaped from if a completed signal is produced. It immediately indicates that it is possible for our SRS to enter a steady state from which it cannot escape and for which the agent has no behaviour that would allow it to escape. For this reason it becomes apparent that some alternative is required to mitigate this situation. The solution is a timeout signal. Without the explicit realisation of the SRN and its evaluation this possible point of failure would not easily be detected.

Even though the boolean network based analysis of the system gives some qualitative information about the system it should be noted that due to the complex dynamics of the system as represented by interacting random functions and parallel threads it may exhibit other “anomalies” not modelled in the simplified abstraction.

6.2 Evaluation

Similar probabilistic signal regulation machines, to the Observation Process 3 P-SRM described here, were constructed for each of the agents in the system. In our test runs five of each agent type were deployed to the appropriate platform as indicated in figure 3 earlier. The system was both adaptive and robust. It is possible to remove nodes (platforms) and the system re-organises so as to compensate and continues to function.

The system was evaluated for robustness, adaptability, and scalability at vari-

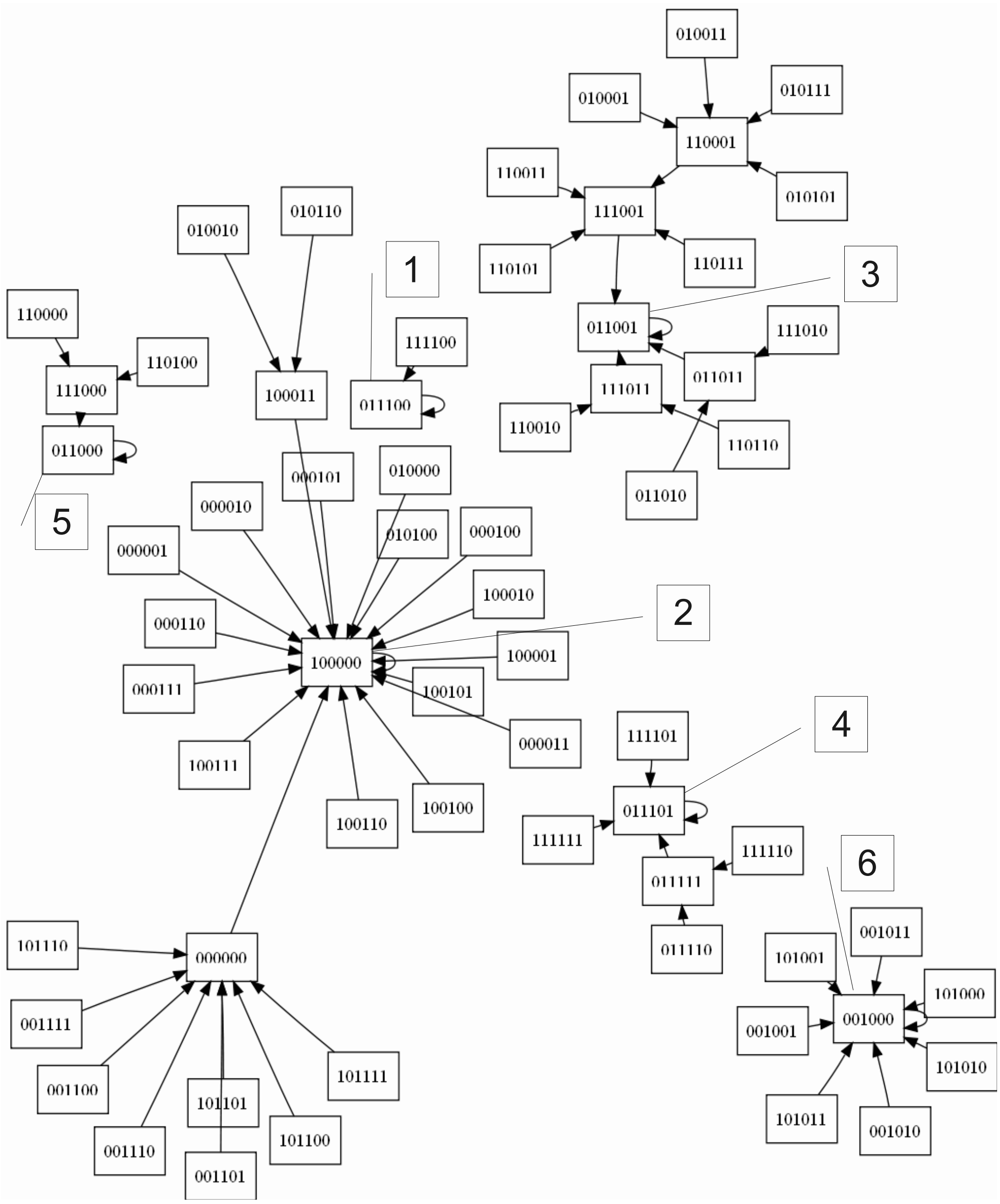

Fig. 5: *Attractor landscape resulting from the interaction of the signals between Observation Source 2 and Observation Process 3. Where the bit position of the signals is given by [A_{op3}, W_{op3}, K_{op3}, C_{op3}, T_{op3}, F_{op3}].*

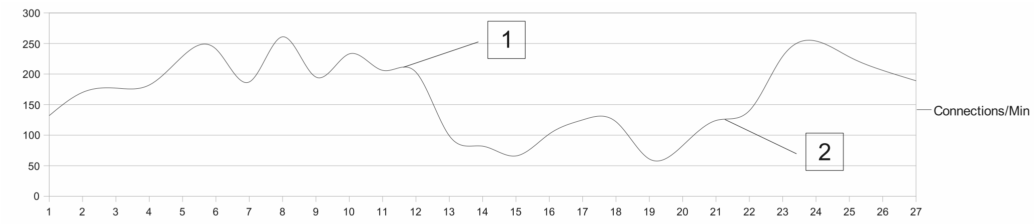

Fig. 6: *Robustness to failure and adaptability. The y-axis shows the number of connections handled per minute. The x-axis shows the number of minutes that have passed.*

ous levels. Firstly the system is evaluated at the platform level to assess how the system responds to an entire group of agents being removed. Figure 6 shows the system's robustness to failure and adaptability when a platform with agents is removed. At label 1 in figure 6 one of the platforms is purposefully failed. The system performance begins to degrade however note that there is not an instant crash but instead a gradual degradation in performance over a minute, as the system re-organises, before the system begins to oscillate within its new range. At label 2 a new platform is brought on line the system self-organises and increases the number of connections a minute that it can handle.

Although removing and adding platforms gives some rudimentary analysis of robustness a more in-depth evaluation is required at a finer scale. In order to gain this deeper insight the system is evaluated at the agent community level of a single platform. In the agent community level experiment individual agents are randomly selected and temporarily suspended from or reanimated to the community. The probability that an agent will be temporarily suspended or reanimated is referred to as the volatility of the system. Here the volatility, is the probability for any agent that it will be suspended or reanimated every 100 micro-seconds. In the experiment the volatility is increased every ten minutes by 0.01. By time step ten every agent is almost guaranteed to be suspended or reanimated at least once every second. The experiment is repeated five times and the averages of the total completes and failures across all five runs are recorded. Figure 7 shows the results of this experiment. Notice how the performance in terms of total completes is impacted by the random agent suspensions. However there is no significant change in the number fails in the ninety minute time period. The experiment shows the

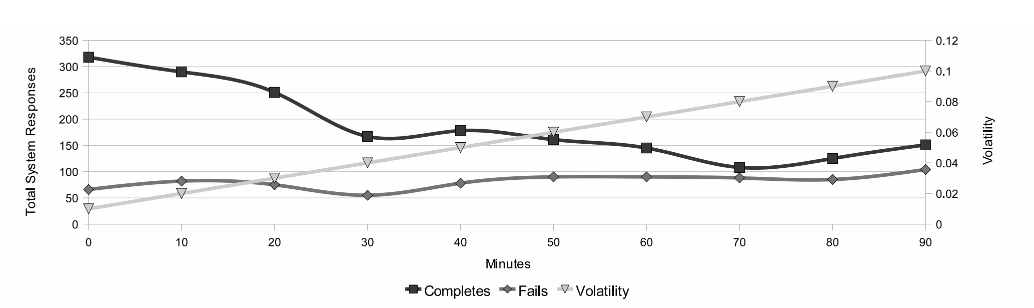

Fig. 7: *Chart showing the result of increasing the volatility of the agents and the resultant response in the number of completed versus failed responses.*

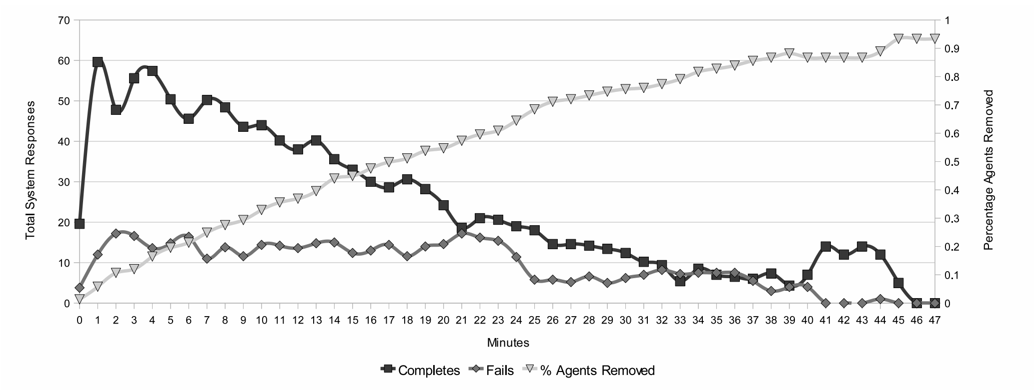

Fig. 8: *Number of removed agents before system is driven to complete failure.*

systems robustness and gracefull degradation at the community level.

The experiment in which random agents are temporarily suspended is extended by permanently suspending agents to evaluate how many agents can be removed from the system before the entire systems collapses. The experiment is repeated five times and the results averaged across all five runs. Figure 8 shows a chart with the results of the experiment. Total system failure occurs on average after forty minutes when approximately ninety percent of the agents have been removed.

Although the community level gives some indication of overall robustness and adaptability of the system, there is also robustness and adaptability at the individual agent level within its SRS. In the individual agent level experiment noise is introduced in the form of random false signals. In order to gain some perspective the experiment is run using the P-SRM and without the P-SRM. A good ninety

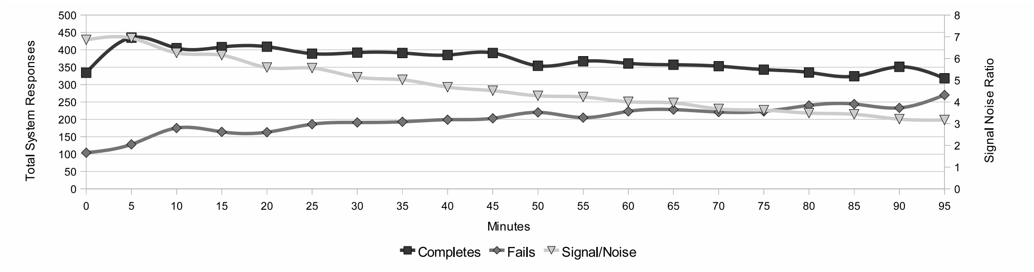

Fig. 9: *Chart shows total system responses in relationship to the signal to noise ratio of the system when a non probabilistic SRM is being used.*

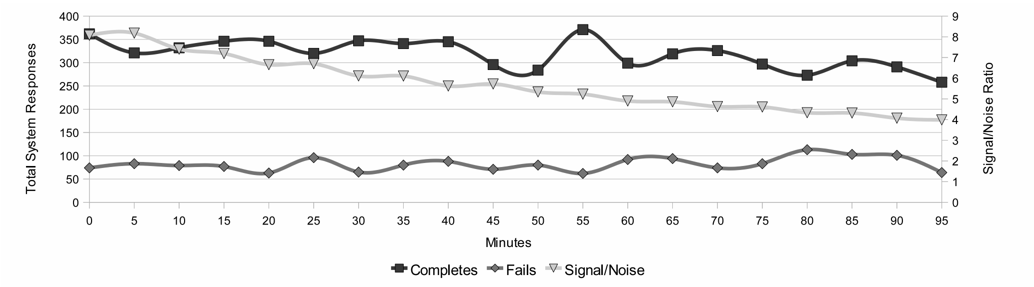

Fig. 10: *Chart shows total system responses in relationship to the signal to noise ratio of the system when a P-SRM is being used.*

percent of the signals introduced, as there are no rules that they trigger, have no effect on the agent's SRM. However approximately ten percent of the signals do affect the agent's SRM. The signal to noise ratio takes into account only the ten percent of signals that can affect the SRM, although the signal to noise ratio in truth would be much higher. The experiment is run five times and the average across all five runs is taken. Figure 9 shows the results without the probabilistic rules. Note how as the signal to noise ratio decreases the number of failures increases. In figure 10 the same experiment is performed using the P-SRM. Here it can be seen that as the signal to noise ratio decreases the number of failures remains constant within a range. What is intresting is that the additional robustness of the P-SRM comes at a performance cost as can be seen in the total complete system responses when comparing the two experiments. However the total system responses that are fails is much lower using the P-SRM.

Although the above results are promising with respect to the adaptability and robustness of the system, the scalability of the approach is of some concern. In

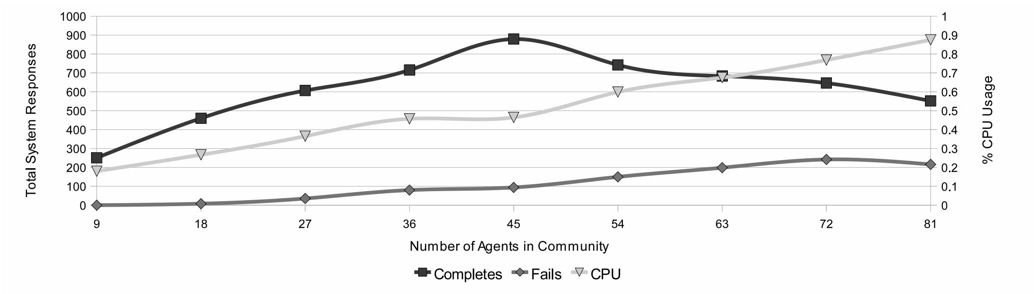

Fig. 11: *Chart showing the relationship between the number of agents on a platform the total CPU usage and how the system is able to scale.*

order to evaluate the scalability, the number of agents in the community on a platform is incremented in steps of nine starting from nine to up until eighty one. The amount of CPU being consumed is also recorded along with the total number of system responses. For each of the agent community sizes the experiment is repeated five time and the average number of completes and fails in a ten minute run is taken. Figure 11 shows the relationship between the community size and total system responses. It is clear from the chart that although the system scales well on a dual core CPU up until a certain threshold beyond that point the additional load incurred as a result of the large number of signals produce result in a performance penalty. How these effects will be mitigated on larger clusters and what the negative effects of the network I/O bottleneck will be between nodes requires further investigation.

To gain some deeper insight into the works of the system a typical individual run of the system is evaluated. Here it can be seen that the system exhibits load balancing properties. By placing the system under load it is possible to see which agents are being utilised. The system compensated for scenarios where different observation sources OS1 or OS2 were placed under load and also balanced the usage of observation process OP1 and OP2 which are CPU and memory intensive process respectively as can be seen in figure 12. Notice how in figure 12 at ten minutes the load is changed from being higher on OS1 to being higher on OS2. At this point you can see the sudden increase in failures and decrease in completions. However by twelve minutes the system is performing within range again. One anomaly which can be seen is that there is a local maximum for completions and local minimum for failures at three minutes. The locals at this point indicate that although the system is continuously self-optimising it is not obtaining the optimal. It is likely that this would be the case as, in order to maintain plasticity, some amount of inefficiency must be built into the system.

What is significant is that it is not necessary to retrieve resource availability indicators from the operating systems in order to obtain this adaptability. Instead the adaptation emerges as a result of the self-organising properties of the signal regulated agent based system. The system is built using just transitions from one attractor to another and the agents are able to effect each other and communicate only by manipulating the signal landscape.

By using the signal space, the self-organising properties that occur as a result of the micro-macro interplay of entropy between the agents and the signals is explicit. Rather than thinking about a state space the discussion centres on a signal space that represents the dynamics of information as it flows through the SRS. The various "states" are represented by the attractors in the signal space generated by the SRS. Using a signal regulatory network as a model enables both qualitative and quantitative assessment of the stability of various steady states and the explicit creation of a self-organising system.

One limitation of such systems is that is difficult to deduce failure. The system may continuously self-organise, searching through and repeatedly visiting various configurations without realising that there is no feasible solution. Although many of these may exist as multi-stabilities, in which case it is possible to identify them and prevent the cycle from re-occurring, others may be limit cycles that never repeat in precisely the same way.

7 Conclusion

In conclusion, it has been shown that it is possible to build self-organising systems upon the theory of SRS. Solutions that exhibit various properties such as self-healing and self-optimisation naturally emerge out of systems built upon the principles of SRS. Further due to the grounded nature of such systems, with explicit entropy sinks and the ability to model them using signal regulatory networks, they present a good foundation for building self-organising systems. By extending

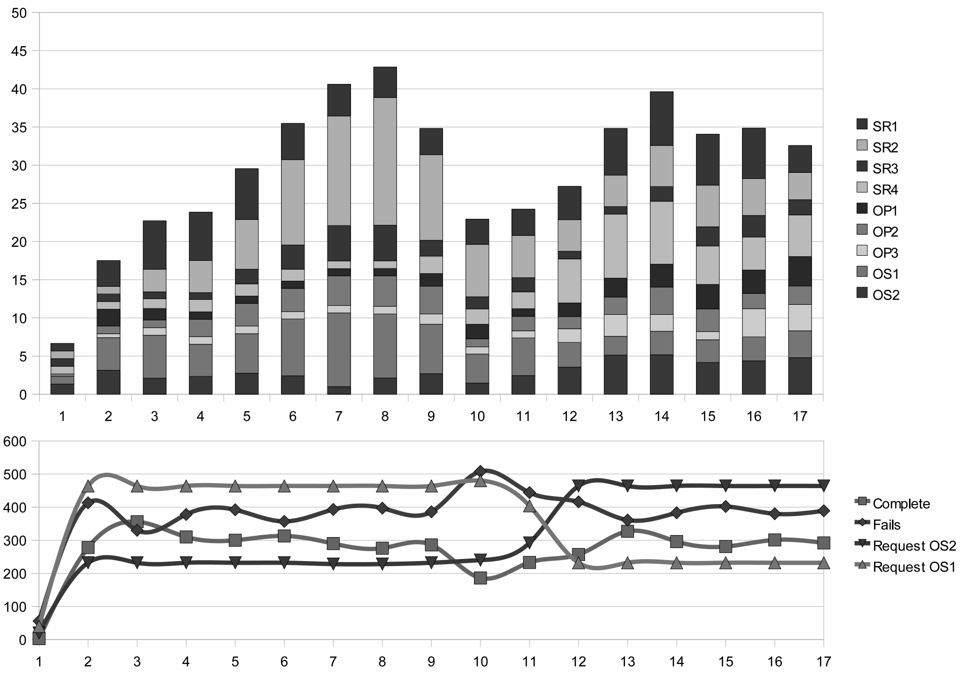

Fig. 12: *System under an abnormal load. The load forces the system to start failing certain requests. The upper chart shows the usage of agents of each type within the system. The lower chart shows the number of requests being made to OS1 and OS2 along with number of request being completed and failed. In both charts the x-axis shows the number of minutes that have passed.*

the simple examples presented here into more complex real world scenarios it is conceivable that ever more functional emergent systems will be built.

Of some concern is the scalability of the approach. However this constraint on scalability may be due to the current serial processing capabilities found in many computing systems. Nature tends to perform in a much more parallel environment than our current ones. In the future as a result of the trend in multi-cored, parallel processing capabilities it is envisioned by the authors that SRS across these multi-cored machines may have much to offer, especially given the inherently parallel nature of SRS.

Given that all of Nature, from the functioning of gene regulatory networks, to pheromone trails built by insects, to the neural firing in the brain can all be modelled as natural signal regulatory networks, software systems underpinned by the same theory should exhibit many of the same properties. A well grounded understanding of how SRS are able to self-organise and exhibit emergent properties will enable computer science to understand and model ever more complex systems. Software engineering will be enabled to build on this solid foundation and produce systems that can be qualitatively and quantitatively evaluated and analysed.

References

[1] Thomas D Seeley. When is self-organization used in biological systems? *Biol Bull*, 202(3):314–318, Jun 2002.

[2] E. Alm and A.P. Arkin. Biological networks. *Current Opinions on Structural Biology*, 13(2):193–202, Apr 2003.

[3] I. Stewart. Networking opportunity. *Nature*, 427(6975):601–604, 2004.

[4] R. Thomas. Laws for the dynamics of regulatory networks. *Int J Dev Biol*, 42(3):479–485, 1998.

[5] V. Lacroix, L. Cottret, P. ThÃ©bault, and M. Sagot. An introduction to metabolic networks and their structural analysis. *IEEE/ACM Trans Comput Biol Bioinform*, 5(4):594–617, 2008.

[6] M.D. Lewis. Self-organizing individual differences in brain development. *Developmental Review*, 25(3-4):252–277, 2005.

[7] S. Camazine, J. Deneubourg, N.R. Franks, J. Sneyd, G. Theraula, and E. Bonabeau. *Self-organization in biological systems*. Number 0691116245. Princeton University Press, New Jersey, USA, illustrated edition, 2001.

[8] D. Bray. Protein molecules as computational elements in living cells. *Nature*, 376(6538):307–312, 1995.

[9] U. Alon. Biological networks: the tinkerer as an engineer. *Science*, 301(5641):1866–1867, Sep 2003.

[10] T. de Wolf and T. Holvoet. Emergence and self-organisation: a statement of similarities and differences. In *Proceedings of the Second International Workshop on Engineering Self-Organising Applications*, pages 96–110, Berlin, Heidelberg, 2004. Springer-Verlag.

[11] C Gershenson and F Heylighen. When can we call a system self-organizing. In *7th European Conference on Advances in Artificial Life (ECAL 2003*, Dortmund, Germany, 2003.

[12] Francis Heylighen. The science of self-organization and adaptivity. In L. D. Kiel, editor, *Knowledge Management, Organizational Intelligence and Learning, and Complexity*, pages 253–280. EOLSS Publishers Co. Ltd., Oxford, UK, 2003.

[13] W.R. Ashby. Principles of the self-organizing system. In Heinz Von Foerster and Jr. George W. Zopf, editors, *Principles of Self-Organization: Transactions of the University of Illinois Symposium*, page 255, Oxford, UK, 1962. Symposium Publications, Pergamon Press.

[14] H. Morowitz and E. Smith. Energy flow and the organization of life. Technical Report 1, Krasnow Institute for Advanced Study and Santa Fe Institute, 2007.

[15] LAN Amaral and JM Ottino. Complex networks. *European Physical Journal B: Condensed Matter Physics*, 38:147–162, 2004.

[16] S.H. Strogatz and D.E. Herbert. *Nonlinear dynamics and chaos*. Addison-Wesley Reading, MA, 1994.

[17] W. Andy Wright, Robert E. Smith, Martin Danek, and Pillip Greenway. A generalisable measure of self-organisation and emergence. In *ICANN '01: Proceedings of the International Conference on Artificial Neural Networks*, pages 857–864, London, UK, 2001. Springer-Verlag.

[18] T. de Wolf and T. Holvoet. Towards a methodology for engineering self-organising emergent systems. *Frontiers in Artificial Intelligence and Applications*, 135:18, 2005.

[19] A.G. Ganek and T.A. Corbi. The dawning of the autonomic computing era. *IBM Syst. J.*, 42(1):5–18, 2003.

[20] G.D.M. Serugendo, M.P. Gleizes, and A. Karageorgos. Self-organisation and emergence in MAS: An overview. *Informatica*, 30(1):45–54, 2006.

[21] W. Banzhaf and N. Pillay. Why complex systems engineering needs biological development: Essays and commentaries. *Complex.*, 13(2):12–21, 2007.

[22] J. Miller and P. Thomson. Beyond the complexity ceiling: Evolution, emergence and regeneration. In *Proc. GECCO 2004 Workshop on Regeneration and Learning in Developmental Systems*, Seattle, USA, 2004.

[23] P. Maes. Modeling adaptive autonomous agents. *Artif. Life*, 1(1-2):135–162, 1994.

[24] J.O. Kephart and D.M. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

[25] Y. Diao, J.L. Hellerstein, S. Parekh, R. Griffith, G. Kaiser, D. Phung, and NY Hawthorne. Self-managing systems: A control theory foundation. In *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2005. ECBS'05*, pages 441–448, 2005.

[26] M.C. Huebscher and J.A. McCann. A survey of autonomic computing—degrees, models, and applications. *ACM Comput. Surv.*, 40(3):1–28, 2008.

[27] F. Klein and M. Tichy. Building reliable systems based on self-organizing multi-agent systems. In *SELMAS '06: Proceedings of the 2006 international workshop on Software engineering for large-scale multi-agent systems*, pages 51–58, New York, NY, USA, 2006. ACM.

[28] Danny Weyns and Tom Holvoet. An architectural strategy for self-adapting systems. In *SEAMS '07: Proceedings of the 2007 International Workshop on Software Engineering for Adaptive and Self-Managing Systems*, page 3, Washington, DC, USA, 2007. IEEE Computer Society.

[29] J.H. Holland. Exploring the evolution of complexity in signaling networks. *Complex.*, 7(2):34–45, 2001.

[30] J.H. Holland. Studying complex adaptive systems. *Journal of Systems Science and Complexity*, 19(1):1–8, 2006.

[31] J.H. Holland. A derived markov process for modeling reaction networks. *Evolutionary Computation*, 11(4):339–362, 2003.

[32] J. Decraene, G.G. Mitchell, B. McMullin, and C. Kelly. The holland broadcast language and the modeling of biochemical networks. *Lecture Notes in Computer Science*, 4445:361, 2007.

[33] J. Decraene, G.G. Mitchell, and B. McMullin. Evolving artificial cell signaling networks: Perspectives and methods. *Intelligence (SCI)*, 69:165–184, 2007.

[34] I. Prigogine. Beyond being and becoming. *New Perspectives Quarterly*, 21(4):5–12, 2004.

[35] F.M. Harold. *The way of the cell: Molecules, organisms, and the Order of Life*. Oxford University Press, New York, USA, 2001.

[36] M.B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403(6767):335–338, Jan 2000.

[37] D.S. Latchman. Transcription factors: an overview. *Int J Biochem Cell Biol*, 29(12):1305–1312, Dec 1997.

[38] R. Laubenbacher. A computer algebra approach to biological systems. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 5–6. ACM New York, NY, USA, 2003.

[39] P. Nurse. Life, logic and information. *Nature*, 454(7203):424, 2008.

[40] Sui Huang, Gabriel Eichler, Yaneer Bar-Yam, and Donald E Ingber. Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Phys Rev Lett*, 94(12):128701, Apr 2005.

[41] Ali H Brivanlou and James E Darnell. Signal transduction and the control of gene expression. *Science*, 295(5556):813–818, Feb 2002.

[42] Neil D Theise and Mark d'Inverno. Understanding cell lineages as complex adaptive systems. *Blood Cells Mol Dis*, 32(1):17–20, 2004.

[43] Alexei Kurakin. Self-organization vs watchmaker: stochastic gene expression and cell differentiation. *Dev Genes Evol*, 215(1):46–52, Jan 2005.

[44] G.T. Reeves and S.E. Fraser. Biological systems from an engineer's point of view. *PLoS Biol*, 7(1):e21, Jan 2009.

[45] A. de la Fuente, G. Fotia, F. Maggio, G. Mancosu, and E. Pieroni. Insights into biological information processing: structural and dynamical analysis of a human protein signalling network. *Journal of Physics A-Mathematical and Theoretical*, 41(22):224013–224013, 2008.

[46] A. Wagner. From bit to it: how a complex metabolic network transforms information into living matter. *BMC Syst Biol*, 1:33, 2007.

[47] B.N. Kholodenko. Cell-signalling dynamics in time and space. *Nat Rev Mol Cell Biol*, 7(3):165–176, Mar 2006.

[48] R. Vilela Mendes. Tools for network dynamics. *Int. Journal of Bifurcation and Chaos*, 15:1185–1213, 2005.

[49] T. Rohlf and Stefan B. Self-organized criticality and adaptation in discrete dynamical networks. Technical report, Max-Planck-Institute for Mathematics in the Sciences and University Bremen, Institute for Theoretical Physics, Leipzig, Germany and Bremen, Germany, 2008.

[50] G. Caldarelli and D. Garlaschelli. Self-organization and complex networks. Technical report, Centre SMC CNR-INFM and Dipartimento di Fisica, Roma, Italy, June 2008.

[51] Reginald D. Smith. The dynamics of internet traffic: Self-similarity, self-organization, and complex phenomena. Technical report, Bouchet-Franklin Research Institute, Rochester, USA, July 2008.

[52] R.V. Mendes. Network dynamics: Tools and examples. *Nonlinear Dynamics*, 44(1):181–204, 2006.

[53] D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[54] S.A. Kauffman. *At home in the universe: The search for laws of self-organization and complexity*. Oxford University Press, New York, USA, 1995.

[55] P. Smolen, D. A. Baxter, and J. H. Byrne. Mathematical modeling of gene networks. *Neuron*, 26(3):567–580, Jun 2000.