Abstract: Models of multi-agent systems with fixed network structure usually update the states of all agents in synchronous fashion. Examples include Cellular Automata, Random Boolean Networks and Artificial Neural Networks. Some recent studies have shown that the behaviour of such models can change dramatically if random asynchronous updating is used. Here we show that many real systems, both natural and artificial, undergo updating that is asynchronous, but ordered in some way. We use examples to demonstrate some of the properties of ordered asynchronous updating in Lsystems and cellular automata. In many cases, models of such processes effectively hide both their asynchronous nature, and the ordering, by embedding them in the model's details. This practice has prevented earlier recognition of ordered asynchronicity as well as some important implications. Among these implications are its role in the rise of modularity within complex systems. As an example, we introduce the “spotlight model” of gene regulation, a random Boolean network in which controller nodes create modules by unfreezing different sets of nodes in turn. We argue that such models are not only more realistic representations of nature, but have potential advantages for solving complex problems.

Author Address: dnewth@csu.edu.au

URL: http://researchoutput.csu.edu.au/R/-?func=dbin-jump-full&amp;object_id=6454&amp;local_base=GEN01-CSU01


CRO identification number: 6454
Ordered Asynchronous Processes In Natural And Artificial Systems

David Cornforth¹, David G. Green¹ David Newth¹ & Michael Kirley¹

¹ School of Environmental and Information Sciences
Charles Sturt University, Albury, Australia
Phone: +2 6051 9652 Fax: +2 6051 9897
Email: dcornforth@csu.edu.ac.nz

Presented at the 5th Australasia-Japan Joint Workshop
University of Otago, Dunedin, New Zealand
November 19th-21st 2001

ABSTRACT
Models of multi-agent systems with fixed network structure usually update the states of all agents in synchronous fashion. Examples include Cellular Automata, Random Boolean Networks and Artificial Neural Networks. Some recent studies have shown that the behaviour of such models can change dramatically if random asynchronous updating is used. Here we show that many real systems, both natural and artificial, undergo updating that is asynchronous, but ordered in some way. We use examples to demonstrate some of the properties of ordered asynchronous updating in L-systems and cellular automata. In many cases, models of such processes effectively hide both their asynchronous nature, and the ordering, by embedding them in the model’s details. This practice has prevented earlier recognition of ordered asynchronicity as well as some important implications. Among these implications are its role in the rise of modularity within complex systems. As an example, we introduce the “spotlight model” of gene regulation, a random Boolean network in which controller nodes create modules by unfreezing different sets of nodes in turn. We argue that such models are not only more realistic representations of nature, but have potential advantages for solving complex problems.

Keywords and phrases: ordered asynchronous processes, modularity, spotlight model, modular Boolean networks

1.0 INTRODUCTION
Evolutionary Computation has provided models that can assist in the understanding of natural processes. However, a large class of behaviours has been overlooked. Here we draw attention to the largely unknown behaviour associated with ordered asynchronous systems. These systems arise in such diverse contexts as rainforest dynamics, gene expression, telecommunication networks, intercellular signalling, neural networks, social networks and multi-processor arrays. The salient features of such systems are agents capable of simple information processing, with links between agents forming a network, and information flow between agents (Holland 1996). Such systems can be modelled by networks of simple processing elements that interact via connections. This class of systems includes such well-known models as Cellular Automata, Random Boolean Networks and Artificial Neural Networks. Many examples of such models have appeared in the literature. They illustrate how complex behaviour may arise from the interaction between many simple processors that obey simple local rules. An evaluation of the global states of such networks reveals the presence of both cyclic and point attractors (Wolfram 1984), the former being regarded as an explanation for cyclic processes observed in natural and artificial systems.

Traditionally, the above systems have been treated as parallel processes. The models use synchronous updating of the nodes. That is, all nodes in the network are updated in a single pass, and before any of the new states are allowed to influence other nodes. However, several authors (eg. Thomas 1979, Kanada 1994, and Di Paolo 2000) have pointed out that a global clock is not indicative of any observed natural phenomenon. In real life, there are
many systems that do not undergo synchronous updating. Instead the nodes possess their own “clock”. Also, the time scales on which nodes are updated sometimes vary considerably over the network.

Recently several studies have examined the implications of alternative update schemes. Most of these studies have considered Random Asynchronous (RAS) updating, in which a single node is chosen at random and updated at any time. Such systems are also valid models of certain real life processes. One example of RAS would be a social system in which people have states that correspond to opinions (ie, YES/NO states) (Stocker et al. 2000). The state of any individual remains constant in isolation and changes only as a result of interactions with other individuals. However, because individuals interact only when they encounter one another, the timing of updating is essentially random. This system is effectively a random Boolean network with RAS updating.

Not surprisingly, RAS updating changes the characteristics of a system. For example, Harvey and Bossomaier (1997) have pointed out that stochastic updating in RBNs results in the expression of point attractors only; there is no cyclic behaviour. Kanada (1994) has shown that one-dimensional CA models that generate non-chaotic patterns when updated synchronously, generate edge of chaos patterns when randomised. Other researchers have claimed that RAS models can exhibit all the behaviour normally associated with synchronous models. For example, Orponen (1997) has demonstrated that any synchronously updated network of threshold logic units can be simulated on a network having no constraints on the order of updates. Such studies, which are not common in the literature, have largely focussed on random updating schemes.

Despite the recent interest in RAS, our experience of modelling suggests that an important class of processes has been overlooked. This class is comprised of systems in which the updating is asynchronous, but nevertheless occurs in some well-defined order. In this study we argue that Ordered Asynchronous (OAS) processes are very common and deserve close attention.

To date, OAS models have been studied only for special cases. As we will show below, the OAS nature of many systems is often obscured by the manner in which the models are implemented. Therefore, these models have not been recognised as valid alternatives to synchronous models. Our analysis of OAS models reveals that they are capable of displaying all the phenomena associated with synchronous models. We suggest that for many systems, they provide better representations of complex systems than synchronous models. In addition, we argue that studying the behaviour of this class of models will provide a deeper understanding of system behaviour. In this study, we present several examples to illustrate this point. In particular we examine asynchronicity in models of epidemic processes and plant growth, and find that the asynchronous behaviour is not random, but ordered. Finally, we examine the association between OAS behaviour and modularity in random Boolean networks using a new ordered asynchronous update model, the Spotlight Model.

### 2.0 ORDERED ASYNCHRONOUS PROCESSES

Many processes, in both natural and artificial network systems, act in ways that are both asynchronous and ordered. That is, the states of individual nodes are neither updated synchronously, according to a global clock, nor are they updated at random.

For instance, periodic behaviour has been observed in some ant colonies, but cannot be explained in terms of a global synchronising “clock” (Cole 1991). Periodic behaviour has not been observed in single ants of the same colonies. Nevertheless the whole colony is able to synchronise its activity, and possibly to control the period of working and resting phases. This synchronisation suggests that individual nodes (i.e. ants) update their states autonomously, but are able to adjust the update frequency depending on signals from other nodes.

The behaviour of interconnected neurons in the brain leads to global patterns of behaviour across the whole brain. This activity does not exhibit stationary patterns, but periodic, quasi-periodic and chaotic patterns (Freeman 1992). There is no known mechanism such as a global clock in the brain, yet neurons exhibit synchronised behaviour for a time, suggesting a mechanism of autonomous updating as in the previous example. This is borne out by the physiology of the neurons. A neuron is able to provide output signals of different periods, depending on the concentrations of neurotransmitters. These concentrations are affected by signals from other neurons, so that a neuron can change its period of firing in response to signals from other neurons.

Epidemic processes, such as spread of a fire through a fuel bed, also exhibit asynchronicity. In a simple CA model of the fire, each cell (representing part of the fuel bed), once ignited, would at the next time step ignite any neighbouring cells that were not yet ignited. However, in reality fire spread is not so simple. Fuel ignites by being heated up. The rate of heating depends on the fire intensity, the distance from the flames, the fuel moisture, and
so on. In other words, when a cell ignites, its neighbours ignite asynchronously, with the order determined by heat accumulation. Fire spread models incorporate this asynchronous nature in one of two ways (Green, 1983). One algorithm, known as Dijkstra’s algorithm, is to create a list that records all cells that are due to ignite (Weiss 1998). The list is ordered according to the time order of ignitions. Fresh cells are inserted as they become affected. They are removed as they finish burning. This algorithm, which alters the operation of a CA, allows the processing to occur in precise time order. A second approach is to include the time of ignition as an extra state of the cells in the model fuel bed. On each cycle, every cell that becomes due for ignition is processed. This practice hides the asynchronicity by embedding time in the states of the cells. In practice, the time of ignition may also vary across the model to reflect different types of vegetation.

Markov models of forest succession have suffered from the problem of how to include asynchronicity. In these models, the forest is deemed to have a state, which corresponds to a community class (e.g. rainforest, open sclerophyll woodland). These models indicate the probability of a transition from one state to another. However, a fatal flaw in simple Markov models is that the transitions between require vastly different times to complete. A fire induced transition from woodland to grass may be virtually instantaneous, whereas a transition to mature rainforest might take hundreds of years to complete. This problem led to a new kind of representation, known as a semi-Markov model, which adds time delays to each transition (Lord 1993). Semi Markov models have now been the subject of considerable research. However, they remain largely unknown by non-specialists.

A simple model of cell development shows how asynchronous processes may achieve synchronised global behaviour using asynchronous updating. Fig. 1 shows a CA model of cell development in an embryo, consisting of a two-dimensional array of cells in a biochemical gradient that changes in a horizontal direction. Depending on the setting of a global parameter, cells changing state induce neighbouring cells to also change state. In an asynchronous model, these changes may propagate rapidly, giving rise to the phenomenon of synchronised activity, as shown in Fig. 1(c).

Many ecological models “hide” asynchronicity in a similar way. Environmental gradients across a landscape often produce asynchronous behaviour. For instance, changes in soil water across a hill slope mean that plants downhill are likely to grow faster. To deal with this variation, a typical model would include age of the plants as an extra attribute of each cell’s state. Variations in external parameters, such as soil water, are not easy to handle in a synchronous model, but have typically been accommodated by hidden asynchronicity.

### 3.0 ASYNCHRONICITY IN GROWTH MODELS

As we have seen above, a common feature of many asynchronous processes is that the rate at which local events occur is mediated by some external constraint. A good example of this occurs in growth. Two schools have dominated studies of growth processes. The traditional diffusion approach models growth explicitly in relation to biochemical gradients (Thompson 1961). Models of this kind look at growth as a process that consists of biochemical reactions. Most importantly, the rates at which these reactions occur, and therefore local growth rates, are governed by the diffusion of substances across biochemical gradients. In essence, they look at the way external controls and constraints affect growth rates. The rate at which a plant grows, for instance, is limited by the rate at which water and nutrients can be supplied to the growing tissues. It is well known by botanists that different parts of a plant may grow at different rates. These rates may be controlled by the supply of nutrients by
the roots, or by the amount of light incident on different parts of the plant. Diffusion models make use of continuous variables, which allows growth rates to be different across the model.

In contrast, Lsystem models (Lindenmayer 1968) represent growth as a multi-agent system in which the agents are cells or other units. Early Lsystem models were discrete systems, which coped with asynchronicity by introducing spurious intermediate states. More recently Lsystem models have often included timings as states, much like the CA and semi Markov models described earlier. However, differing growth rates can also be accommodated by combining Lsystem rules with a suitable environmental model. The environmental model defines the local time scale for asynchronous state transitions in the Lsystem.

A simple model illustrates the above idea in action. Consider the growth patterns of the two (hypothetical) trees shown in Figure 2 and governed by the following simple grammars:

\[
(1) \quad A \rightarrow 0A0 \\
(2) \quad A \rightarrow AAA
\]

Here the symbol A denotes a growing tip, and the symbol 0 denotes either a terminal or a slow-growing side branch. In the model, the symbol 0 remains constant (ie. no further growth occurs). The second tree has a much greater number of branches than the first, because the number of growing tips is multiplied by three at each growth stage (Fig. 2A). A generation is defined as one iteration of the model. The second tree takes longer to complete each generation, since more growing tips have to be serviced by a constant rate of nutrient uptake. In effect the supply of nutrients from the roots sets the time scale. Because the nutrients get spread over many more growth elements the time scale effectively slows down. The first model progresses through many more generations than the second model does in the same time. Their updating is asynchronous to one another. As can be seen from the Fig. 2B, the first tree grows at a constant rate, while the second's growth rate continually declines. Thus, at maturity, the first tree can be expected to be very tall, whereas the second remains as a small bush.

Under our assumptions, the second tree will always have about 3 times as many leaves as the first tree. This will affect the relative chance of survival of the two trees, depending on their environment. For simplicity, we ignore complications that arise because the roots grow too, and assume that both trees have identical root systems. If we make the reasonable assumption that a certain amount of water is needed to produce each branch element, then the supply of water up the stem limits the rate at which the development proceeds. This illustrates how a simple Lsystem model may incorporate asynchronicity. In other Lsystem models, different parts of a plant grow at different rates.

![Fig. 2](image)

**Fig. 2. Illustrating asynchronicity in Lsystem models of growth. A) different types of tree grown using different rules B) graph of the relationship between stage of growth reached and time.**

The basic lesson of the above growth models is that an environmental variable, not time, can set the order in which events happen. In the cell development case, it was a biochemical gradient. In the trees example, it was supply of water and nutrients. The dominance of environmental variables seems to be common in many (but not all) cases of OAS updating.
### 4.0 ASYNCHRONOUS MODELS OF COMPLEX SYSTEMS

The synchronous method for updating the states of nodes in a multi agent system model is to calculate all the new states at once, then to update all nodes to their new state simultaneously. Such models are simple, but are not able to model asynchronous features of real systems, as shown above. Several alternative schemes for node updating have been described in the literature.

Kanada (1994) described three update schemes: (1) at each time step a node is chosen at random with replacement; (2) at each time step a node is chosen according to a fixed update order which was decided at random during initialisation of the model; or (3) at each time step a node is chosen according to a fixed interlaced order. The results of this work are that randomised automata models are more likely to generate edge of chaos patterns than models using a fixed update order. This is highly significant, as systems poised on the edge of chaos produce the richest, most interesting behaviour (Langton 1990).

Harvey and Bosomaier (1997) experimented with two random update schemes, where at each time step a node is chosen at random, either with or without replacement. They concluded that synchronous RBNs may be applicable to biological systems only where there are special reasons to believe that some synchronising mechanism exists.

Low and Lapsley (1999) developed algorithms for optimising link utility and bandwidth cost across a communications network, where the goal is to calculate bandwidth for each node that maximises the sum of the utilities. Such a network is necessarily asynchronous because the nodes communicate at different times and with different frequencies. The authors used an update scheme in which each node maintained its own clock, updating autonomously at different rates. They provide evidence that the algorithm will converge to a global optimum when network conditions are static and will track the optimum when network conditions change slowly.

Sipper et al (1997) investigated the evolution of non-uniform CAs that perform specific computing tasks. These models relax the normal requirement of all nodes having the same update rule. In their models, nodes were divided into blocks. Nodes within a block were updated synchronously, but blocks were updated asynchronously. They experimented with three schemes: (1) at each time step, a block is chosen at random with replacement; (2) at each time step, a block is chosen at random without replacement; (3) at each time step, a block is chosen according to a fixed update order. They conclude that synchronous and asynchronous models can be evolved with equivalent computational properties, but models of the asynchronous type may require a larger number of nodes.

Thomas (1979) has suggested an updating scheme in which updates depend upon each node possessing its own clock. Each node updates autonomously with different period. A variant of this scheme that has not been suggested in the literature is to add noise to the update times for each node.

The relative merits of different updating schemes are not well understood. An important research question is to determine their characteristics and their suitability for representing various kinds of multi agent systems.

The work of Sipper et al discussed above brings together two specific themes that we would like to develop. These themes are asynchronicity and modularity. Modularity is a recurrent theme in biological systems. Biological systems have the competing requirements of stability, and exploration of new behaviour. Dividing the system into modules, and reducing the number of connections between modules, has the effect of isolating elements and processes from one another. Modularity provides a means for stability in some parts of the systems and chaotic, exploratory behaviour in other parts. Modularity can emerge spontaneously from asynchronous systems. As we saw earlier, one example is social networks, where people are modelled as nodes in a RBN, and nodes are updated at random (Stockers et al 2000). As the connectivity of the network changes, stable homogenous clusters emerge. This is an example of modularity emerging from asynchronous behaviour.

### 5.0 THE SPOTLIGHT MODEL

As a demonstration of the relationship between asynchronicity and modularity, we end by presenting a model that makes the relationship plain. In this example, we introduce a new update scheme that allows us to investigate the role of modularity and asynchronicity in a RBN. The model is based on the Boolean network models of genetic networks originally proposed by Kauffman (1967). Genes encode proteins that are responsible for directing the proliferation and differentiation of cells into tissues. However, Hox genes are known to regulate the expression of other genes (Cohen 2001). Such effects form a network of genes with their interactions modelled as connections. Regulator genes are responsible for turning whole blocks of other genes on or off. The resulting Homeobox
clusters appear to play an important part in introducing modularity into growth and development. Our spotlight model embodies the idea of homeobox structures. It is a Boolean network divided into modules, similar to the idea of blocks in the work by Sipper et al (1997). However, each block is associated with a regulator node, which allows or prevents state updates of nodes in the block, depending on its own state. The regulator nodes are the same as all other nodes in every other respects, and can themselves be part of a block controlled by another regulator node. Fig. 3 illustrates this organisation in the spotlight model. A block is comprised of a set of nodes whose update is controlled by the beam of a spotlight. Each spotlight is turned on or off according to the state of the regulator node. The number of blocks and the number of nodes contained in each are determined during initialisation of the model. The model allows control of the amount of asynchronicity by altering the number of blocks used. With one block, the whole network is updated synchronously, so become equivalent to the synchronous model as long as the regulator node remains activated. With the number of blocks equal to the number of nodes, the model is similar to a random asynchronous network with an extra node in the neighbourhood, and a reduced rule set.

In our implementation, nodes were represented as a linear array for the purposes of producing time space diagrams. The size of the network was 100 nodes. Each node was connected at random to three others. Multiple tests were used, each time using rule 30 for all nodes (Wolfram 1984). This rule was chosen because it is associated with complex behaviour in synchronous models. For each run of the algorithm, the assignment of regulator nodes and initial states were assigned at random. Each regulator node was constrained to control a contiguous portion of cells, so that behaviour within each single spotlight could be observed. Tests were conducted with the number of spotlights varying between 10 and 90.

Fig. 3. The Spotlight Model. Squares represent nodes in a RBN. Squares with heavy border represent regulator nodes, which control the updating of a number of other nodes within their region of influence, or spotlight (shown by dotted circle). Other links between nodes (ie. the normal RBN structure) are omitted from the diagram for clarity.

Figs. 4 shows some typical time space diagrams obtained from the spotlight model by varying the number of spotlights. Notice the tendency for particular kinds of behaviour to turn on and off in parts of the network. When the number of spotlights is small, the model exhibits a variety of behaviours (Fig. 4a–c). As the number of spotlights increases, the range of behaviour is reduced. Convergence to a homogenous state becomes very likely once the number of spotlights reaches a critical threshold. We suggest that this convergence is due to the changing connectivity properties within the model (Green et al 2001). Preliminary studies suggest that around 50-60 spotlights, sufficient decomposition has occurred to cause convergence to a homogenous state. Results for 50 spotlights and above are not shown, as the model degenerated almost immediately to stationary or cyclic states.

As the above results show, the spotlight model exhibits the typical kinds of behaviour associated with any synchronous RBN. But layered on top of that are several new features. For example, Fig. 4(c) shows several different types of behaviour occurring at the same time in different regions. In each model, transitions are apparent between these different types of behaviour, as the regulator nodes are turned on or off. To the best of our knowledge, these features have not been described before in a model of a multi agent system. The modularity built into the Spotlight Model enables different processes to occur at different rates without interference. This behaviour demonstrates the connection between modularity and ordered asynchronicity.
6.0 CONCLUSIONS

In the previous sections, we have shown that many real systems, both natural and artificial, undergo asynchronous updating. Examples such as the spread of forest fires and plant growth show that models of living systems often have intrinsic asynchronous behaviour, which has not been fully recognised. Such systems have richer and more diverse types of behaviour than is normally seen in simple models. For example, different parts of a plant may grow at different rates, depending on external variables. Different parts of a plant may serve different functions at different times, exhibiting rapid transitions between types of behaviour, such as the onset of flowering. This richness is difficult to incorporate within synchronous models, with their reliance upon the notion of a global synchronising clock. So far the investigation of asynchronous updating is at an early stage, with most studies considering only stochastic updating. However, most real systems use Ordered Asynchronous (OAS) updating, and there is evidence of the presence of timing variables in many OAS systems.

As we have shown, asynchronicity is closely related to modularity. Real systems exploit modularity to reduce complexity and increase adaptability. For example, different parts of a plant can have different functions, which occur at different rates. The limited connections between modules prevents processes occurring at different rates from interfering with each other. Here, we have demonstrated this idea in a simple RBN, namely the Spotlight Model. Examples of OAS processes in real systems have been ignored, partly because modularity dominates the study of such systems, and partly because the OAS nature is hidden within the model’s workings. However, not all OAS behaviour is modular. Therefore the understanding of OAS systems is crucial to gain an understanding of modularity.

The question of OAS and modularity has relevance to many applications. For example, the design of controllers for robots has recently begun to focus on distributed control, which is a form of modularity (Brooks 1986, 1991). This distributed control often entails asynchronous updating and activity. Distributed controllers within the robot also need to communicate with each other asynchronously. So a consideration of modularity leads to asynchronicity, and vice versa. Another example may be taken from Evolutionary Computation, where modularity can be applied to a population in order to preserve diversity during convergence (Kirley et al 2000). For instance, applying the Spotlight Model here would divide population into sub-populations, each of which are able to undergo genetic manipulation operators (eg. Mutation, crossover, selection) only when interacting with the other sub populations. This model could be used to represent the stability of populations until coming into contact with other species.

Another question for which OAS might provide useful insights lies in the formation of network structure itself. As we saw at the start, many systems of free agents are inherently asynchronous. Many Alife models, for instance, consist of agents moving about in a landscape (e.g. animals, traffic). The interactions in these systems depend on encounters between moving agents. The question is how such systems can develop organisation in which a network defines the behaviour. For instance, social hierarchies, first emerge from, and then determine the interactions between individuals in a society. We suggest that asynchronicity plays a role in this process.

In conclusion, we have described here several alternative models that incorporate OAS behaviour. The implications from this are important. These new models are able to provide more realistic representations for some aspects of nature; they exhibit behaviour that previously has been difficult to study; and they provide an
insight into the role of modularity within complex systems. Such models may be the choice when modelling appropriate systems. We should examine the properties as they could lead to new model, for example the Spotlight Model. With the rising interest in modularity, these new models have potential advantages for solving complex problems.

REFERENCES


