

# Do Artificial Ants March in Step? Ordered Asynchronous Processes and Modularity in Biological Systems

David Cornforth, David G. Green, David Newth & Michael Kirley

School of Environmental and Information Sciences

Charles Sturt University, Albury, Australia

Email: {dcornforth|dgreen|dnewth|mkirley}@csu.edu.au

## Abstract

Processes that occur in many biological systems are not synchronous, but are governed by asynchronous updating. Although some previous models of multi agent systems have incorporated asynchronous updating in an informal or implicit way, the importance of asynchronous behaviour has been largely overlooked. However, the update scheme chosen is very important in determining the overall system behaviour. We illustrate this point using several updating schemes in simple models. The implication is that care should be given to selecting an update scheme that is appropriate for the modelling application. We also observe that certain ordered asynchronous processes play a role in emergent modularity, an important process in the evolution of complexity in living systems.

## Introduction

Most biological systems are composed of many interacting elements. The way in which elements update their internal states plays an important role in many processes. Most published models of such systems update internal states in parallel and synchronously. However, we show that state updating is asynchronous in biological systems, and may be described as either Random Asynchronous (RAS) or Ordered Asynchronous (OAS). The recognition of such processes has far reaching implications for the way in which we model biological systems. We also show that OAS processes have a role in emergent modularity. This in turn may contribute to the formation of dynamical hierarchies in biological systems. Understanding such mechanisms is an open problem in Artificial Life (Bedau *et al.* 2000).

Models of multi agent systems have rarely focused on the temporal pattern of interactions within the system. In particular, updating is usually assumed to occur synchronously and in parallel. That is, all agents in the system are assumed to update their state in a single pass, and before any of the new states are allowed to influence other agents. This assumes the existence of a “global clock” that dictates the pace of all local processes in the system. However, several authors (Thomas 1979; Kanada 1994; Di Paolo 2000) (amongst others) have

pointed out that a global clock is not indicative of any observed natural phenomenon. This raises the question of how state updating should be handled in models.

We examine some different updating schemes in simple models, and shows that the scheme chosen has a very strong effect on the type of global behaviour observed. We conclude that the choice of updating scheme is an important part of the specification of the model, and should reflect the purpose to which the model will be used.

Finally, we observe that models using OAS updating can be used to demonstrate the emergence of modularity. Modularity is of increasing interest, since it is becoming apparent that biological systems employ modularity to cope with the exponential increase in complexity as these systems grow in size (Green *et al.* 2001).

## Asynchronous Processes in Biological Systems

In many living systems, there is abundant evidence that agents update their state asynchronously. The examples below also show that these processes are ordered, and demonstrate self-synchronisation as a mechanism that forms and perpetuates modules.

Ants participating in a colony exhibit complex collective activity. Ants do not work constantly, but spend between 55% and 72% of their time resting, depending on species (Cole 1991; Delgado & Sole 1999; Franks *et al.* 1990). Individual ants separated from the colony display active and resting periods with an aperiodic pattern. However, a whole colony displays a synchronised periodic pattern of active and resting behaviour, with period of between 15 and 30 minutes.

Individual ants update their activity state autonomously, but are able to adjust the update frequency in response to interactions with other ants. Ants living in a colony were observed to wake each other (Franks *et al.* 1990). From these observations, it appears that global synchronous behaviour in the colony arises from local asynchronous behaviour. This behaviour has been recognised in models, for example, Goss and Deneubourg (1988).

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 asynchronous updating as in the previous example.

Bushfires are propagated when fuel is heated until it ignites. The rate of heating depends on the fire intensity, the distance from the flames, the fuel moisture, and so on. When a plant ignites, its neighbours ignite asynchronously, with the order determined by heat accumulation.

The competition of different species within a forest system, coupled with catastrophe such a forest fire, leads to a complex system of interactions that have been studied as a means of developing better management strategies, among other reasons. Transitions between different community classes (e.g. rainforest, open sclerophyll woodland) require vastly different times to complete (Noble & Slatyer 1980), so are examples of asynchronous updating.

### Asynchronous Processes in Models

The examples in the previous section suggest that asynchronous updating is the default, rather than the exception, in natural multi agent systems. This raises the question of what update method should be chosen in models of such systems. This will depend on the purpose of the model. Asynchronous models may provide more realistic representations of natural systems than synchronous models, and may provide a deeper understanding of system behaviour.

The majority of published models use synchronous updating. This may be because little is known about asynchronous updating, and how it may change the behaviour of the model. Asynchronous updating has been incorporated in cellular models of bushfire, for example, by using a list of cells due to ignite (Kourtz & O'Regan 1971), or by adding extra states (Green 1983). The asynchronous nature of forest succession has led to the adoption of the semi-Markov model, (Howard 1971). There have also been attempts to investigate the effects of random asynchronous (RAS) processes in more abstract models. 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, but no cyclic attractors, although they introduce the notion of loose cyclic attractors. Kanada (1994) has shown that some 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 by a network that has no constraints on the order of updates. However, this work depends on a carefully crafted network connectivity that is unlikely to be observed in natural systems.

### Cellular Automata Models of Asynchronous Processes

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. As an initial survey, we have implemented CA models using five distinct updating schemes. Source code is available from <http://life.csu.edu.au/~dcornfor/masys.html>.

The *Clock* scheme (Thomas 1979; Low & Lapsley 1999) assigns a timer to each cell. The period of each timer is set at random, as is the initial phase of the timer. The model is evolved by incrementing the values of all timers at each time step, then checking the values. Those timers that have exceeded the value of their period variable are updated in order, with new states being implemented immediately. After a cell has been updated, its timer is set to zero.

In the *Cyclic* scheme, a node is chosen at each time step according to a fixed update order. This order is decided at random during initialisation of the model. This follows Kanada (1994).

In the *Random Independent* scheme (Harvey & Bossomaier 1997), one cell is randomly selected for update at each time step. Subsequent selections are independent of each other.

The *Random Order* scheme (Harvey & Bossomaier 1997), makes a list of all cells at each time step, and sorts them into a random order. It then uses this list to decide which cell to update. In this variant, each cell is updated once only for every time step, but the order of updating is random.

The *Synchronous* scheme is included for comparison. It has been described in many papers, and its properties are well-known (Wolfram 1984). At each time step, the state of each cell is calculated, but held in a temporary store until all states have been calculated. Then the cells are all updated to their new state synchronously.

These five schemes were implemented as a one dimensional CA having 250 cells and 2 states, with each cell connected to its two neighbours and itself. Further details are provided in Cornforth *et al.*, (2001). For each rule, the five models were initialised to random states. Time space diagrams were obtained, each one representing the evolution of the model from initialisation for the first 500 time steps.

A few of the more interesting results from our prelim-

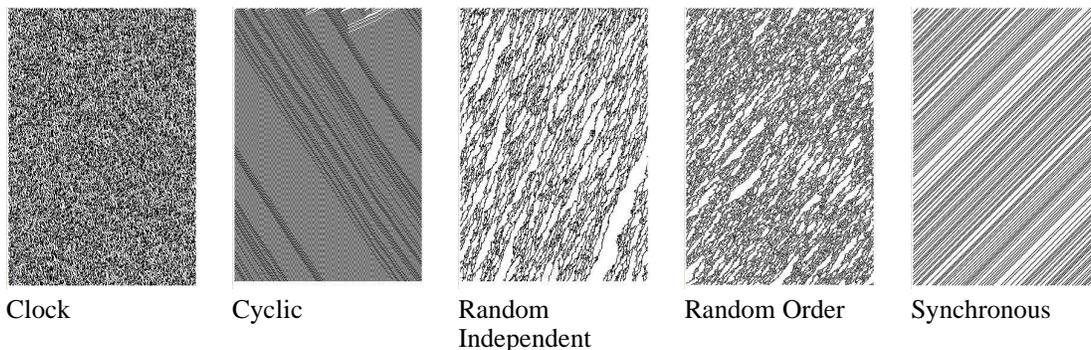


Figure 1: Time state diagrams for Cellular Automata models of different update processes for rule 38.

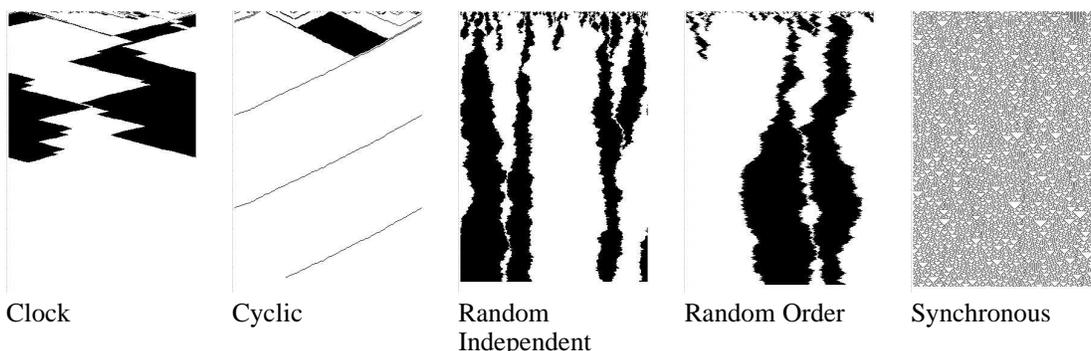


Figure 2: Time state diagrams for Cellular Automata models of different update processes for rule 146.

inary studies are shown below. Figs. 1 and 2 show the evolution of the CA models for rule 38 and 146. These rules were chosen for their examples of the possible divergence between the behaviour of different schemes.

For both rules, the random independent scheme and the random order scheme show similar behaviour, which appears to be complex. However, after multiple tests using many rules, both of these schemes always converged to a point attractor. This is to be expected from the results of Harvey and Bossomaier (1997). However, the results shown here suggest that the transients can be very long.

The *synchronous* scheme shows cyclic behaviour for rule 38, and chaotic behaviour for rule 146. This is to be expected, as its properties are already well known.

The cyclic scheme converges to a cyclic attractor. After multiple tests using different rules, the cyclic scheme always evolved to a cyclic or point attractor. This scheme never showed any evidence of chaotic behaviour.

The *clock* scheme seems to show the widest range of behaviour. For example, in Fig. 1, it shows evidence of chaotic behaviour when the synchronous scheme shows cyclic behaviour. In Fig. 2, it shows complex behaviour converging to a point attractor (all cells white), while the synchronous scheme shows complex behaviour.

These preliminary results suggest that the properties

of such models are changed by the particular update scheme chosen. Any researcher attempting to build a model of a multi agent system should question the type of updating used, and ensure that an appropriate scheme is chosen for the system being studied and the uses to which the model will be put.

### A Cellular Automata Model of Self Synchrony

We now extend the *clocked* update scheme to show how self-synchrony, observed in ant colonies and neural tissue, may be easily modeled. Our model is similar to the Kuramotos model of self-synchronising oscillators (Strogatz 2000), but each cell is connected only to its two adjacent neighbours.

All experiments were performed with a one dimensional CA having 250 cells and 2 states. The model was run using a variety of rules, and the network was completely homogenous. States, clock periods and phases were all initialised by drawing from a uniform distribution. At each time step, the frequency  $\omega$  of each oscillator is updated according to the phase  $\theta$  of its neighbours, using a gain term  $\beta$ :

$$\omega_{i(t+1)} = \omega_{i(t)} + \beta(\theta_{i+1(t)} - \theta_{i(t)}) + \beta(\theta_{i-1(t)} - \theta_{i(t)}) \quad (1)$$

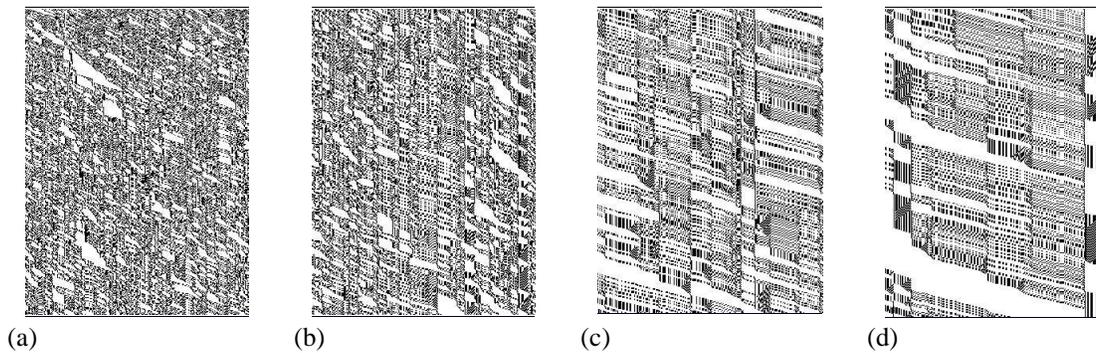


Figure 3: Time space diagrams showing four stages of the self-synchronising model. Initialisation is shown in (a), the next stage is shown in (b) and so on. Modules appear as vertical regions separated by vertical lines.

Results are shown in Fig. 3. In the early stages of the evolution of the model (a), cells states are randomly distributed, and state transitions occur independently of each other. However, further evolution of the model (b to d) reveals the formation of clusters or modules of cells that update their states together. Modules appear as vertical sections of cells displaying a similar pattern. The cells in a module have achieved synchronisations within the module. This synchronisation persists for some time, and then the modules break apart and re-form. This may be helpful in explaining the presence of patches of burning forest during bushfires, synchronous behaviour in ant colonies, and the ability of neural tissue to form transient resonant modules.

## Discussion

A consideration of some biological systems reveals that state updating in such systems is usually asynchronous. This implies that anyone building models of such systems should at least consider using asynchronous updating.

A review of the literature reveals that different updating schemes do exist, including random and ordered asynchronous updating.

Implementing these schemes in a simple Cellular Automata model show that different schemes produce different global behaviour. This implies that anyone building models should be aware of these differences, and choose an updating scheme that is appropriate to the model being constructed, and its purpose.

A modification of one of these schemes shows how easy it is to model self-synchronisation once asynchronous updating is used.

This model demonstrates a possible mechanism for the emergence of modularity in complex systems. The implications of this include a possible explanation of the origin of some types of modularity in living systems.

## References

- Bedau, M. A.; McCaskill, J. S.; Packard, N. H.; Rasmussen, S.; Adami, C.; Green, D. G.; Ikegami, T.; Kaneko, K.; and Ray, T. S. 2000. Open problems in artificial life. *Artificial Life* 6:363–376.
- Cole, B. 1991. Short-term activity cycles in ants: Generation of periodicity by worker interaction. *American Naturalist* 137:244–259.
- Cornforth, D.; Green, D.; Newth, D.; and Kirley, M. 2001. Ordered asynchronous processes in natural and artificial systems. In *Proceeding of the Fifth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, 105–112. Dunedin, New Zealand: The University of Otago.
- Delgado, J., and Sole, R. 1999. Self-synchronisation and task fulfillment in ant colonies. Forthcoming.
- Di Paolo, E. A. 2000. Searching for rhythms in asynchronous random boolean networks. In Bedau, M. A.; McCaskill, J. S.; Packard, N. H.; and Rasmussen, S., eds., *Alife VII: Proceedings of the Seventh International Conference*. Cambridge, Mass.: MIT Press.
- Franks; Bryant; Griffiths; and Hemerik. 1990. Synchronisation of the behaviour within nests of the ant *Leptothorax Acervorum (Fabricius)* — I. discovering the phenomenon and its relation to the level of starvation. *Bulletin of Mathematical Biology* 52(5):597–612.
- Freeman, W. 1992. Tutorial on neurobiology: from single neurons to brain chaos. *International Journal of Bifurcation and Chaos* 2(3):451–482.
- Goss, S., and Deneubourg, J. 1988. Autocatalysis as a source of synchronized rhythmical activity in social insects. *Insectes Soc.* 35:310–315.
- Green, D.; Newth, D.; Cornforth, D.; and Kirley, M. 2001. On evolutionary processes in natural and artificial systems. In *Proceeding of the Fifth Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, 1–10. Dunedin, New Zealand: The University of Otago.

- Green, D. 1983. Simulated fire spread in discrete fuels. *Ecological Modelling* 20:21–32.
- Harvey, I., and Bossomaier, T. 1997. Time out of joint: Attractors in asynchronous random boolean networks. In Husbands, P., and Harvey, I., eds., *Proceedings of the Fourth European Conference on Artificial Life (ECAL97)*, 67–75. MIT Press.
- Howard, A. 1971. *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes*. John Wiley and Sons.
- Kanada, Y. 1994. The effects of randomness in asynchronous 1D cellular automata. In *proceedings of ALIFE IV*.
- Kourtz, P., and O'Regan, W. 1971. A model for a small forest fire to simulate burned and burning areas for use in a detection model. *Forest Science* 17(2).
- Low, S., and Lapsley, D. 1999. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*.
- Noble, I. R., and Slatyer, R. O. 1980. The use of vital attributes to predict successional changes in plant communities subject to recurrent disturbances. *Vegetatio* 43:5–21.
- Orponen, P. 1997. Computing with truly asynchronous threshold logic networks. *Theoretical Computer Science* 174(1-2):123–136.
- Strogatz, S. 2000. From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators. *Physica D* 143(1-4):1–20.
- Thomas, R., ed. 1979. *Kinetic Logic: A Boolean Approach to the Analysis of Complex Regulatory Systems*, volume 29 of *Lecture Notes in Biomathematics*. Springer Verlag.
- Wolfram, S. 1984. Cellular automata as models of complexity. *Nature* 311:419.