Multi-phase Updating - A Practical Approach to Simulating Animat Agents

C.J. Scogings and K.A. Hawick
Computer Science, Institute for Information and Mathematical Sciences,
Massey University, Albany, NS 102-904, Auckland, New Zealand
{c.scogings, k.a.hawick}@massey.ac.nz
Tel: +64 9 414 0800 Fax: +64 9 441 8181

March 2012

ABSTRACT
A growing number of applications can be modelled using spatial agent systems or animats. Typical animat simulations model collective macroscopic phenomena using encoded individual behaviours for microscopic agents. In spatial agent systems there are well known problems that occur if agent cells are updated in a sequential order. This is known as the sweeping problem and it also occurs in other numerical simulations such as differential equation solvers and leads to observable macroscopic effects in the simulated system that are solely artifacts of the implementation algorithm. We explore various strategies to remove these artifacts including multi-phase updating. We find that in some models that include non-diffusional effects such as predator-prey interactions, a two-phase update is not sufficient and a three-phase update strategy is necessary to preserve model semantics, particularly when concurrency is used to speed up the model. We discuss the computational implications of this for animat agent simulations.

KEY WORDS
agent-based models; updating agents; multi-phase update; animats; model semantics.

1 Introduction

Agent-based modelling [21, 22, 29] is a powerful construct for tacking many complex-systems problems [11] in: physics; sociology; finance; and other areas where emergent properties [37] arise from relatively simple individual agent properties. Simulations of multi-agent systems can be readily constructed in general purpose programming languages. Spatial agents - or animats as they are often described - are agents that have some spatial position and therefore can move around and interact with other agents. The key idea for spatial agents is usually that individual agents only interact with those local to them rather than with the entire population.

An important aspect of modelling spatial animat agents is to unambiguously specify how the agents are updated or evolved in the model. Updating might mean each agent is selected in turn or at random and given the option to exercise its microscopic behavioural rules. It might move, eat, kill, breed, grow, die, buy, sell, communicate, or exercise whatever other individual actions are open to it in the particular model being studied. The update is typically applied iteratively to evolve the whole model system through its phase or state space [31, 40].

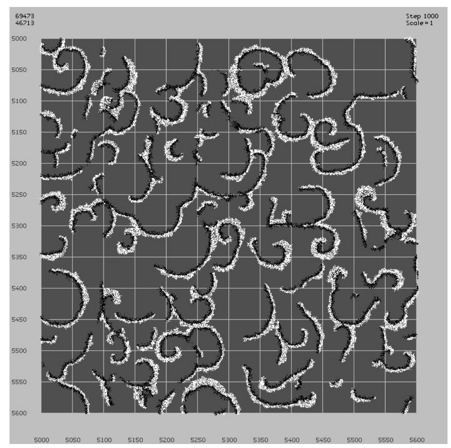

Figure 1: Animat model configuration step 1000 of a multi-phase run of a predator-prey model. Predators are black and prey are white.

Usually a particular multi-agent model will have some constraints such as conservation laws, or other global laws that cannot be sensibly violated by individual agent updates. Different models can be expressed using different sorts of update algorithmic procedures. These are often categorised as synchronous - where every agent is effectively updated at once or asynchronous - where individual updates are only loosely coupled to model time.

In automaton models [41] such as Conway's Game of Life [14] a fully synchronous update is part of the model definition. In stochastic models such as the Metropolis Monte Carlo dynamics applied to the Ising model of a magnet [18, 27] other factors allow a partially synchronous update. Other models require an asynchronous updating scheme. The idea of using asynchronous updating is not new [4, 10] and has been explored and debated in the complex systems literature [16] for a number of different models including multi-agent systems [9]; random boolean networks [17]; automata [28]; and asynchronous cellular automata [24, 25]

Many large scale models make use of parallelism and therefore need an appropriate form of concurrency control [13] to exploit parallel hardware without changing the model semantics. Asynchronous update issues also arise in some numerical methods such as successive over-relaxation [2].

In this paper we explore the practical issues behind using a multi-phase update for agent-based models such as one of the various artificial life [3] animat models. Agent-based models have contributed significantly across a wide range of distinct areas, from Artificial Life [1, 23, 38] through ecosystems [33] and trading and economics [8, 26] to military combat [6, 7].

All agent-based models require some form of interaction between agents, for example in a predator-prey model [19, 35] predators attempt to catch prey and also to breed with other predators. It is these micro-interactions that lead to the well-known emergent macro-properties of such models, for example the emergence of spiral patterns in the predator-prey model shown in Figure 1. We are primarily interested in animat models of this category in this present paper (Section 3), although we do draw comparisons with simpler deterministic models such as the Eden/Epidemic model in Section 2

An important part of the interaction between agents is the order in which such interactions are executed and this can have a significant effect on the observed patterns of behaviour. The interaction between agents becomes more complicated when agents are developed to perform a range of "higher-order" behaviours such as trading [34] or signalling [36]. In this article we critique the known solutions to this problem and suggest some practical approaches to how best to incorporate this important process into agent-based models.

Our paper is structured as follows: In Section 2 we review some update effects and artifacts that arise from a single phase update approach in a well defined example model such as the Eden/Epidemic growth model. We discuss similar issues in animat models such as our own in Section 3. We review some key update algorithmic ideas in Section 4. We discuss some statistical ideas concerning updates and multi phase algorithms in Section 5 and summarise some pragmatic advice to model practitioners in Section 6.

2 The Eden-Epidemic Model

The Eden model [12] of an epidemic or cancerous growth consists of an initial pattern - usually a single infected seed agent in a system of uninfected agents - and whose boundary agents become infected or grow according some probability parameter p at each update step. There are a number of variations, but at each time step of the model, the state of every agent site is updated. This can be done in sequential order or as a two phase update.

In a sequential update, every agent is updated *in-situ*. This means that if an agent is updated after its neighbour then it has up to date information of any changes that have occurred in the states of other agents within the same time step. The order in which the agents are chosen to be updated can be random or fixed. A fixed order is often referred to in the literature as a "sweep". Iterating through the population in a fixed order is computationally efficient as it makes good use of memory caches, which can be highly significant if the model size is very large. However this approach introduces some definite sweeping behaviours that are neither physical nor correct in the sense of being what was intended by the modeller. Often it is the temporal behaviour that the modeller is interested in measuring. This might be in the form of periodic cycles of behaviour or a growth or shrinkage exponent. If the update algorithm is therefore unphysical the experimental simulation results will be at best biased or at worst just wrong.

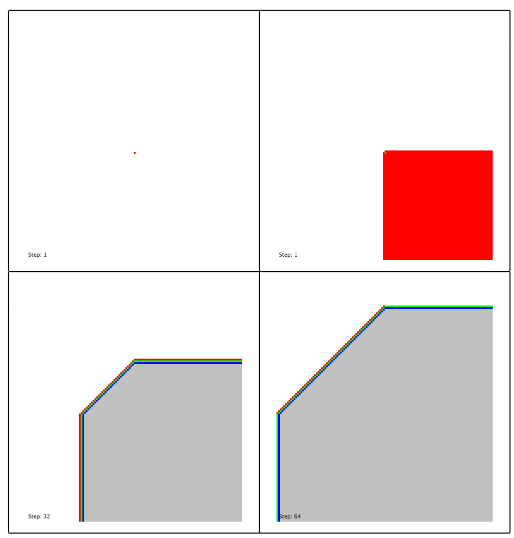

Figure 2: The effects of a sweeping *in-situ* update algorithm for the Eden Epidemic Model 128×128 across successive time steps (cells: white empty; dark live; grey dead, Infection probability 1.0). The simulation starts with a single infected cell at the centre and progresses to the right.

Figure 2 shows the effects when a sequential sweep algorithm is applied to a single central infected cell with infection (or growth) probability $p = 1.0$. The sweep is essentially a row-major raster and the *in-situ* updating and sweeping effect causes the infection to propagate rapidly to all the neighbouring cells that are updated strictly after the infected cell. Infection information travels across the model system at the maximal possible speed. When the probability of infection is substantially less than 1, for example with $p = 0.25$, the skewed results of the model are more subtle. Figure 3 shows the 'correct' result on the left and the skewed results on the right. The correct results have been produced using a two-phase update algorithm.

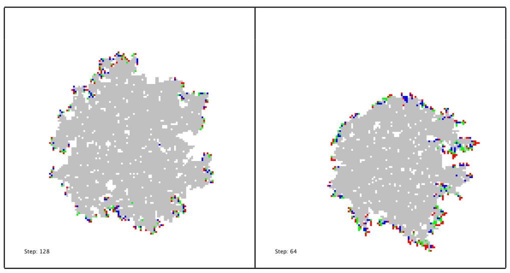

Figure 3: Two-phase update algorithm (left) with a sweeping *in-situ* update algorithm (right) for the Eden Epidemic Model 128×128 across successive time steps (cells: white empty; dark live; grey dead. Infection probability 0.25)

If constrained to a sequential update method a better way of updating the system is to randomise the choice of sites to update. This can either be done by randomly shuffling the list of sites to update (perhaps using a pair-wise shuffle). An even more random approach can be taken by performing Monte-Carlo hits on the sites: on average all sites will be updated once every n time steps (where n is the number of sites in the system), but as the update sites are being chosen randomly, there is a possibility some sites will be updated more frequently than others over a short time period. This has the effect of slightly blurring the concept of 'time' in the simulation (see Figure 4).

Figure 4a) illustrates the cellular growth behaviour of a variation of the Eden Epidemic model [12] when a single infected cell at the centre of the pattern infects nearest neighbouring cells with probability $p = 0.3$ at each time step. In the model shown, infected cells die after two time steps after being infected. Figure 4b) shows how a random algorithm can recover spatial symmetry in the growth model.

Figure 5 shows, by way of contrast, the highly regular pattern that results from the Eden model when a sequential sweep update is used. This is microscopically correct in some sense, but is clearly dominated by the sweeping artifact - and is not representative of the modeller's intentions.

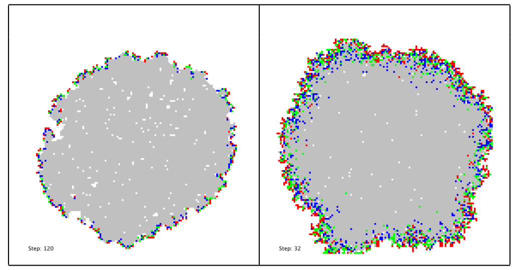

Figure 4: A variation of the Eden Epidemic model is used to show growth time scales and symmetries on a square lattice. Sites are infected from any live nearest neighbour with a probability $p = 0.3$ (left) or $p = 1.0$ (right), and once infected, die after two time steps. The cluster is grown from a single central infected cell. The left hand cluster shows the two phase update algorithm and the right hand uses a random algorithm so cells are updated once per time step on average. Some cells are hit more often and although spatial symmetry is largely recovered, the time scale is accelerated.

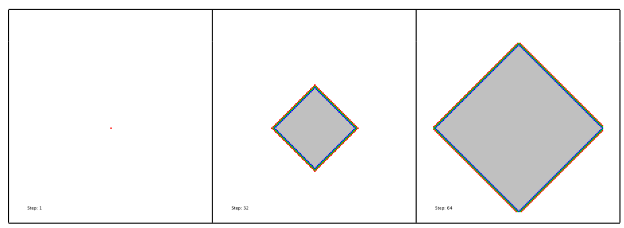

Figure 5: Sequential sweep update algorithm for the Eden Epidemic Model 128×128 cells: white empty; dark live; grey dead. (Infection probability 1.0). The simulation starts with a single infected cell at the centre and progresses to the right.

3 The Animat Model

The animat model we discuss in this section is somewhat more complicated than the Eden model, but is still implemented on a grid of agent-hosting cells, and can have a range of various sorts of update algorithm applied to each microscopic agent. An agent-based prey-predator model has been developed [35] based on "artificial animals" or *animats* [39]. Predators need to catch prey to survive and both species can breed to produce new animats. Animats can "die" from a lack of food (i.e. if health reaches zero) or due to "old age" (i.e. if age reaches a set maximum for the species). Prey animats can also die through being consumed by a predator.

Like most agent-based models, the model is executed as a sequence of cycles (called *time-steps*). Every animat is updated during each time-step and when the updating is completed, the model advances to the next time-step. The following state variables are used to maintain the state of each animat: location in terms of x, y coordinates; age which is increased each time-step; health which is decreased each time-step but increased by “eating”; neighbours in terms of the location and species of nearest neighbours; and the microscopic species rule set which is used to decide which rule to execute. The rule sets used for the experiments described in this article were as follows:

Rules for predators:

1. breed if health > 50% & mate adjacent
2. eat prey if health < 50% & prey adjacent
3. seek mate if health > 50%
4. seek prey if health < 50%
5. randomly move to adjacent position

Rules for prey:

1. breed if health > 50% & mate adjacent
2. eat grass if health < 50%
3. seek mate if health > 50%
4. move away from adjacent predator
5. randomly move to adjacent position

Most rules carry conditions usually relating to location or current health. The rules are presented in priority order and each animat executes the first rule in its list for which the conditions are satisfied. The “Breed Rule” regulates the production of new animats and when an animat is “born” it inherits the rules of its parents. The “Breed Rule” does not always succeed. Even if the necessary conditions are satisfied, there is still only a random chance that a new animat will be produced. This chance is known as the “birth rate” and is an abstraction of the cumulative effect of several unknown factors including birthing difficulties, availability of suitable shelter, etc. It would be difficult to simulate these factors separately so it is convenient to substitute one value which produces the desired effect in the model. Normally the birth rate for predators is set to 15% and the birth rate for prey is set to 40% but these can be modified to produce different effects in the simulation.

Predators need to consume prey and prey need to eat grass to survive. Whenever the animat successfully executes the “eat” rule, its health state variable is increased. Grass is placed at various points around the map and is automatically replenished. Future work will experiment with grass that is not replenished (or is replenished very slowly).

The interaction of the animats as they execute their individual rules has produced interesting emergent features in the form of macro-clusters often containing many hundreds of animats. We have analysed and documented these emergent clusters in [20]. The most fascinating cluster that consistently appears is a spiral and several spirals are visible in Figure 1.

The model uses a two-phase update system so each animat carries two versions of its state variables – the “current state” and the “future state”. A rule is selected from the rule set and applied to the current state and this generates the future state. At the end of each time-step the current state is changed to match the future state.

4 Animat Model Updates

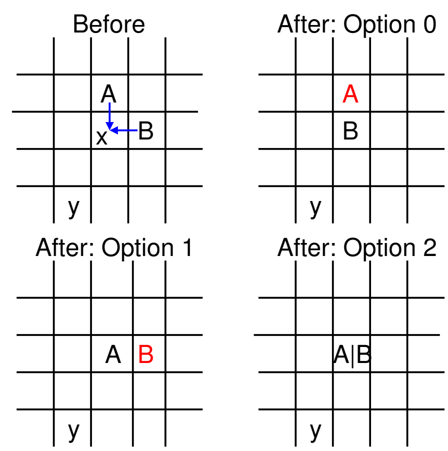

Figure 6: Using a 2-phase update, it is possible for predators A and B to “simultaneously” both locate and consume the single adjacent prey animat x. This renders the model ambiguous as it should not be possible for one prey animat to sustain more than one predator and there are three outcomes and which is the actual one is an artifact of the implementation.

Due to the known problems of a single-phase update system (discussed in section 2 above), the predator-prey model was initially constructed using a two-phase update. However it was soon discovered that the two-phase update led to a serious problem outlined in Figure 6.

The sequence of events that caused this problem can be summarised as follows:

- predator (current state) locates adjacent prey (current state)
- predator executes “Eat” Rule
- predator (future state) is updated by increasing health
- prey (future state) is updated by setting health to zero (dead)

Note that this sequence would only be executed by a predator that had less than 50% health. However in a typical model with tens of thousands of predators – often in close proximity as shown in Figure 1 – there is a strong probability that two or more predators will be adjacent to the same prey. When this occurs, both predators update their future states to indicate an increase in health, i.e. several predators can "eat" the same prey. Note that the current state of the prey (and the predators) remains unchanged, allowing other predators to repeat the process with the same prey. The problem was discovered because it was noticed that huge numbers of predators were existing off an impossibly small prey population. This update problem thus rendered any results from the model meaningless.

It is not possible to solve the problem and retain a pure two-phase update system. The solution adopted in our model was to introduce a hybrid of the sequential and two phase update systems. In this system every prey animat was given an extra state variable to keep track of its current status and the procedure outlined above was modified as follows:

- predator (current state) locates adjacent prey (current state)
- **if prey (current state) is "dead" then abandon this sequence**
- predator executes "Eat" Rule
- predator (future state) is updated by increasing health
- prey (future state) is updated by setting health to zero
- prey **(current state)** is updated by changing status to dead

Since this sequence allows the current state of the prey to be modified, this is no longer a two-phase update. Sequential update is acceptable in these circumstances,although it is necessary also to randomly shuffle the animats before updating. Experiments have shown that the best system is sequential updating but in a random order. Predator results are shown in Figure 7. Prey results are shown in Figure 8.

The periodic boom-bust variations seen in Figures 7 and 8, showing the populations of predator and prey agents is characteristic of this sort of spatial agent model. The period lengths are consistent between runs and are a measurable property of the experiments. Phase effects - and fluctuational noise however must be averaged out, and this is usually done by averaging over many independently seeded initial model configurations. The top trace in each of these Figures shows the normalupdate algorithm. The second has no random shuffle applied - which manifestly lowers the mean value of both populations. The third case has predators given a slight advantage, which makes them overly successful hunters and again mean populations are even lower.

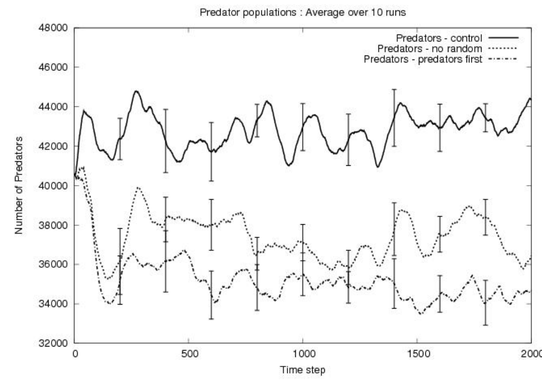

Figure 7: Predator populations using different update methods.

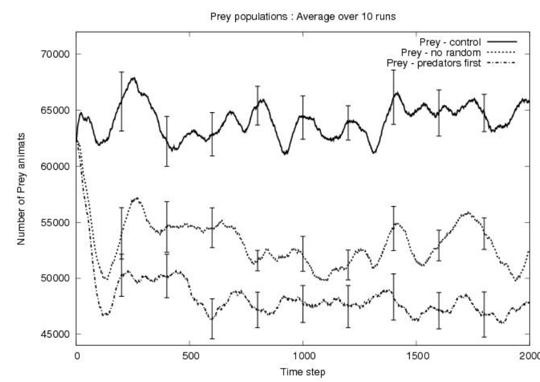

Figure 8: Prey populations using different update methods.

5 Discussion

Generally it appears that the animat model problem can be partially fixed by introduction of a 2-phase update. However, the 2-phase update causes other problems when animats are involved in (a) killing other animats or (b) bumping into objects including other animats. In these cases it is better to use a random shuffle.

We have introduced these ideas in the context of a pragmatic animat simulation. It is interesting to speculate about more general implications for other models. One theoretical approach is to consider the state space of this class of models. Each microstate $\mathbf{X}$ of the whole model system is completely specified by the set of agent state variables $\{a_i\}$. Consider a probability functional $P(\mathbf{X}, t)$ be associated with the microstate $\mathbf{X} \equiv \{a_i\}$ at time t and consider the transition probability $W_{\mathbf{X} \to \mathbf{X}'}$ giving the likelihood of a change of microstate $\mathbf{X}$ to $\mathbf{X}'$. The following master equation can be set up, requiring that the rate of change of probability of microstate $\mathbf{X}$ at time t be given by considering all transitions from $\mathbf{X}$ and all transitions to $\mathbf{X}$:

$$\frac{dP(\mathbf{X})}{dt} = -\sum_{\mathbf{X}'} W_{\mathbf{X}\to\mathbf{X}'} P(\mathbf{X}) + \sum_{\mathbf{X}'} W_{\mathbf{X}'\to\mathbf{X}} P(\mathbf{X}') \quad (1)$$

By requiring the model update algorithm to yield $P(\mathbf{X}) \to P_{eq}(\mathbf{X})$, the thermodynamic equilibrium probability of microstate $\mathbf{X}$ as $t \to \infty$, as a solution of equation 1 with $\frac{dP(\mathbf{X})}{dt} = 0$ so that:

$$\sum_{\mathbf{X}'} W_{\mathbf{X}\to\mathbf{X}'} P(\mathbf{X}) = \sum_{\mathbf{X}'} W_{\mathbf{X}'\to\mathbf{X}} P(\mathbf{X}') \quad (2)$$

This is the condition of detailed balance [5, 30]. It is common to use the stronger (but tractable) condition that:

$$\frac{W_{\mathbf{X}'\to\mathbf{X}}}{W_{\mathbf{X}\to\mathbf{X}'}} = \frac{P(\mathbf{X})}{P(\mathbf{X}')} \quad (3)$$

So that the probability of the system moving to microstate $\mathbf{X}$ is increased for highly probable microstates $\mathbf{X}$, and decreased for unlikely ones. It is then necessary to recognize that for a Boltzmann statistical weighting of the microstates, the probabilities $P(\mathbf{X})$ can be expressed in terms of the Hamiltonians $\mathcal{H}(\mathbf{X})$.

$$P(\mathbf{X}) = Ae^{-\frac{\mathcal{H}(\mathbf{X})}{k_b T}} \quad (4)$$

Where A is a normalising constant, k_b is Boltzmann's constant and T the temperature. Substituting 4 in 3 gives:

$$\frac{W_{\mathbf{X}'\to\mathbf{X}}}{W_{\mathbf{X}\to\mathbf{X}'}} = e^{-\frac{\{\mathcal{H}(\mathbf{X}) - \mathcal{H}(\mathbf{X}')\}}{k_b T}} \quad (5)$$

This does not have a unique solution but commonly used approaches are the Metropolis [32] or Glauber functions [15].

These or some other deterministic or stochastic procedure provides a way of traversing the phase space of the model. In the case of the animat model we have similar concerns and goals. The procedures become more complicated in that we have killing and births and other effects that change the number of agents involved. There is potential to develop formulations for model phase state traversal bringing these two approaches together in a unified notation.

6 Conclusion

In summary we have discussed models such as the Eden/Epidemic model and our predator/prey animat model and drawn out comparisons between the updating procedures involved in both. We have identified in particular that sweeping effects can give rise to the wrong model behaviour. This is because spatial correlations are introduced that are solely due to artifacts of the algorithm rather than from the thermal and other fluctuations that we desire to simulate.

Single-phase sequential updating is shown to cause particular problems. Furthermore, in certain circumstances (such as for more complex models like the animat system) even an alternative two-phase update causes a different but related set of problems. We have discussed in particular the problem that arises when two predators would potentially eat the same prey. There are disadvantages to the two-phase update model - not the least of which is that the two-phase model also wastes memory in storing two complete model states. This is problematic for the very large model systems sizes we generally wish to simulate.

We conclude by offering the following pragmatic advice to model practitioners. Firstly, check if a sequential update will cause sweeping problems in the proposed model, and if the answer is yes, then use a two-phase update. Secondly, check if the two-phase update will cause problems similar to the "eating" problem we described, and if the answer is yes, then introduce a full or partial sequential update.

There appear to be some quite profound and deeper philosophical issues underpinning these pragmatics and we believe there is further work to be done to unify the update algorithm semantics under a single notation. This may lead to insights into the relationships between these different classes of simulation models.

References

[1] Adami, C.: On modeling life. In: Brooks, R., Maes, P. (eds.) Proc. Artificial Life IV. pp. 269–274. MIT Press (1994)

[2] Adler, S.L.: Over-relaxation method for the monte carlo evaluation of the partition function for multiquadratic actions. Physical Review D 23(12), 2901–2904 (June 1981)

[3] Aleksic, Z.: Artificial life: growing complex systems. In: Bossomaier, T.R., Green, D.G. (eds.) Complex Systems. pp. 91–126 (2000), iSBN 0-521-46245-2

[4] Bandini, S., Bonomi, A., Vizzari, G.: What do we mean by asynchronous ca? a reflection on types and effects of asynchronicity. In: Proc. 9th Int. Conf. on Cellular Automata for Research and Industry (ACRI 2010). pp. 385–394. No. 6350 in LNCS, Ascoli Piceno, Italy (21-24 September 2010)

[5] Binder, K. (ed.): Monte Carlo Methods in Statistical Physics. Topics in Current Physics, Springer-Verlag, 2 edn. (1986), number 7

[6] Cares, J.R.: The use of agenbt-based models in military concept development. In: Proc. 2002 Winter Simulation Conference. pp. 935–939. San Diego, California, USA (8-11 December 2002)

[7] Cioppa, T.M., Lucas, T.W., Sanchez, S.: Military applications of agent-based simulations. In: Proc. 2004 Winter Simulation Conference. pp. 171–180. Washington DC, USA (5-8 December 2004)

[8] Cliff, D., Bruten, J.: Animat market - trading interactions as collective social adaptive behaviour. Adaptive Behaviour 7(314), 385–414 (1999)

[9] Cornforth, D., Green, D.G., Newth, D.: Ordered asynchronous processes in multi-agent systems. Physica D 204, 70–82 (2004)

[10] Cornforth, D., Green, D.G., Newth, D., Kirley, M.: Do artificial ants march in step? ordered asynchronous processes and modularity in biological systems. In: Proc. Artificial Life VIII - the 8th Int. Conf or the Simulation and Synthesis of Living Systems. pp. 28–32. Sydney, Australia (9-13 December 2002)

[11] Cornforth, D., Green, D.S.: Intelligent Complex Adaptive Systems, chap. Modularity and Complex Adaptive Systems, pp. 75–104. IGI Global (2008)

[12] Eden, M.: A two-dimensional growth process. In: Proc. Fourth Berkeley Symposium on Mathematics, Statistics and Probability. vol. 4, pp. 223–239. Univ. California Press, Berkeley (1960)

[13] Fox, G., Coddington, P.: Complex Systems, chap. Parallel Computers and Complex Systems, pp. 289–338. Cambridge University Press (2000), iSBN 0-521-46245-2

[14] Gardner, M.: Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life". Scientific American 223, 120–123 (October 1970)

[15] Glauber, R.: Time dependent statistics of the Ising Model. J. Math. Phys. 4(2), 294–307 (1963)

[16] Green, D.G.: Towards a mathematics of complexity. Complexity International 3, 98–105 (1996), iSSN 1320-0682

[17] Harvey, I., Bossomaier, T.: Time out of joint: Attractors in asynchronous random boolean networks. In: Husbands, P., Harvey, I. (eds.) Proc Fourth European Conference on Artificial Life (ECAL97). pp. 67–75. MIT Press (1997)

[18] Hawick, K.A.: An agent model formulation of the ising model. Tech. rep., Information and Mathematical Sciences, Massey University, Albany, North Shore 102-904, Auckland, New Zealand (November 2003)

[19] Hawick, K.A., James, H.A., Scogings, C.J.: Roles of rule-priority evolution in animat models. In: Proc. Second Australian Conference on Artificial Life (ACAL 2005). pp. 99–116. Sydney, Australia (December 2005)

[20] Hawick, K.A., Scogings, C.J., James, H.A.: Defensive spiral emergence in a predator-prey model. Complexity International (msid37), 1–10 (October 2008), `http://www.complexity.org.au/ci/vol12/msid37`, iSSN ISSN 1320-0682

[21] Hawick, K., Scogings, C.: Agent-Based Evolutionary Search, chap. Complex Emergent Behaviour from Evolutionary Spatial Animat Agents, pp. 139–160. No. ISBN 978-3-642-13424-1, Springer (January 2010), cSTN-067

[22] Helbing, D., Balietti, S.: How to do agent-based simulations in the future: From modeling social mechanisms to emergent phenomena and interactive systems design. Tech. Rep. 11-06-024, Santa Fe Institute, NM, USA (June 2011), santa Fe Working Paper

[23] Holland, J.H.: Echoing emergence: Objectives, rough definitions, and speculations for echo-class models. In: Cowan, G.A., Pines, D., Meltzer, D. (eds.) Complexity: Metaphors, Models and Reality, pp. 309–342. Addison-Wesley, Reading, MA (1994)

[24] Jeanson, F.: Evolving asynchronous cellular automata for density classification. In: Proc. Artificial Life XI. pp. 282–288. Winchester, UK (5-8 August 2008)

[25] Kanada, Y.: The effects of randomness in asynchronous 1d cellular automata. In: Proc. Artificial Life IV (1994)

[26] King, A.J., Streltchenko, O., Yesha, Y.: Using multi-agent simulation to understand trading dynamics of a derivatives market. Annals of Maths and AI 44(3), 233–253 (July 2005)

[27] Laciana, C.E., Rovere, S.L.: Ising-like agent-based technology diffusion model: adoption patterns vs. seeding strategies. Tech. Rep. arXiv:1011.3834v1, Universidad Catolica Argentina (2010), to Appear in Physica A

[28] Li, W., Packard, N.H., Langton, C.: Transition phenomena in cellular automata rule space. Physica D 45, 77–94 (1990)

[29] Macal, C.M., North, M.J.: Tutorial on agent-based modeling and simulation part 2: How to model with agents. In: Proc. 2006 Winter Simulation Conference, Monterey, CA, USA. pp. 73–83 (3-6 December 2006), iSBN 1-4244-0501-7/06

[30] Manousiouthakis, V.I., Deem, M.W.: Strict Detailed Balance is Unneccessary in Monte Carlo Simulation. J. Chem. Phys. 110(2753) (1999)

[31] Margolus, N., Toffoli, T.: Cellular automata machines. Complex Systems 1, 967–993 (1987)

[32] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. 21(6), 1087–1092 (Jun 1953)

[33] Ronkko, M.: An artificial ecosystem: Emergent dynamics and lifelike properties. J. ALife 13(2), 159–187 (2007)

[34] Scogings, C.J., Hawick, K.A.: Intelligent and adaptive animat resource trading. In: Proc. 2009 International Conference on Artificial Intelligence (ICAI 09) Las Vegas, USA. No. CSTN-076 (13-16 July 2009)

[35] Scogings, C.J., Hawick, K.A., James, H.A.: Tools and techniques for optimisation of microscopic artificial life simulation models. In: Nyongesa, H. (ed.) Proceedings of the Sixth IASTED International Conference on Modelling, Simulation, and Optimization. pp. 90–95. Gabarone, Botswana (September 2006)

[36] Scogings, C., Hawick, K.: Cross-caste communication in a multi-agent predator-prey model. In: Proc. IASTED Int. Conf. on Artificial Life and Applications (AIA 2011). pp. 163–170. IASTED (February 2011)

[37] Standish, R.K.: On complexity and emergence. Complexity International 9, 1–6 (2001), `www.complexity.org.au/vol09`

[38] Tyrrell, T., Mayhew, J.E.W.: Computer simulation of an animal environment. In: Meyer, J.A., Wilson, S.W. (eds.) From Animals to Animats, Proceedings of the First International Conference on Simulation of Adaptive Behavior. pp. 263–272 (1991)

[39] Wilson, S.W.: The animat path to AI. In: Meyer, J.A., Wilson, S. (eds.) From Animals to Animats 1: Proceedings of The First International Conference on Simulation of Adaptive Behavior. pp. 15–21. Cambridge, MA: The MIT Press/Bradford Books (1991)

[40] Wolfram, S.: Statistical Mechanics of Cellular Automata. Rev.Mod.Phys 55(3), 601–644 (1983)

[41] Wolfram, S.: Theory and Applications of Cellular Automata. World Scientific (1986)