Complex systems engineering from nano- to macro-scale

Geoff James*, Jiaming Li, Ying Guo, Geoff Poulton
CSIRO ICT Centre, Locked Bag 17, North Ryde, NSW 1670, Australia

ABSTRACT

The design of complex systems to achieve desired outcomes – complex systems engineering – is achievable in numerous natural systems and in some systems of human construction. This paper concerns multi-agent complex systems that comprise a large number of autonomous, interacting elements. Emergence presents a rich variety of behaviours for the designer to use; however, the unpredictability of emergence is a barrier to conventional engineering methodology. By probing examples of engineered systems and looking for common features, a design methodology may be sought.

Keywords: complexity, emergence, top-down, bottom-up, multi-agent, coordination, self-assembly, distributed energy.

1. INTRODUCTION

The study of complex systems is now over two decades old and much has been discovered about such systems, whose best-known characteristic is the emergence of self-organised global properties from the aggregate behaviour of its constituent components [1]. A good definition of self-organisation is given by Hermann Haken [2]: *a system is self-organizing if it acquires a spatial, temporal or functional structure without specific interference from the outside. By "specific" we mean that the structure or functioning is not impressed on the system, but that the system is acted upon from the outside in an non-specific fashion*. Quantitative measures of complexity have been formulated [3].

Most complex systems research has employed the traditional scientific method of hypothesis and test, a "bottom-up" approach which works from fundamental knowledge and experimentation to derive new principles about the world. However, the very property which characterises complex systems also works to make difficult the formulation of general theories: the inherent unpredictability of emergent behaviour. There is no reason to suppose that this situation will change in the near future.

In contrast, engineering design is a "top-down" process which starts with a global goal and seeks methods to achieve it using available scientific knowledge. For complex systems this has proven to be difficult for reasons stated above: general scientific principles are hard to come by.

There are two approaches to the design of complex systems. The first is to use evolutionary computation or similar methods to obtain designs for specific instances [4, 5]. However, such designs rarely generalise to allow generic design principles, again because of the unpredictability of the underlying system. The second approach is to modify or restrict the system to remove emergence, allowing traditional design methods to apply [6]. Solutions found this way are likely to be inferior because of the loss of the rich solution space provided by complexity, which biological systems use to such advantage. A methodology that balances these approaches is desirable.

Growing experience in designing complex systems has led to useful results in several application domains and suggests that such a methodology might be developed. Two examples of multi-agent complex systems, self-assembly for nanotechnology and management of distributed electrical loads and generators, are presented below for illustration and comparison. These are research activities at different stages on the path to adoption. Self-assembly is a recognised feature of the nano-scale world, presently in natural systems and the laboratory rather than the foundry, and self-assembly studies have given valuable insights towards new methodologies. Distributed energy management is an immediate practical concern, with some existing algorithms using complex-systems techniques, and it remains to be seen whether similar methodologies may be applied. In these and other domains, the coming few years are likely to see many cultural and technical barriers disappear as complex systems engineering begins to gain wide acceptance.

* geoff.james@csiro.au; phone +61 2 9325 3276; fax +61 2 9325 3280; www.ict.csiro.au/as

Complex Systems, edited by Axel Bender, Proc. of SPIE Vol. 6039
603902, (2006) · 0277-786X/06/$15 · doi: 10.1117/12.673480

2. NANO-SCALE: SELF-ASSEMBLY SIMULATION ENVIRONMENT

A number of research groups have developed means for the self-assembly of small components into regular lattice-like arrangements as analogues of natural systems [7]. The size of components ranges from millimetres to centimetres, sometimes referred to as "meso-scale", so they may called "mesoblocks". Such small scale components may be regarded as agents that interact due to their edge properties. They are interesting in themselves but also function as simplified analogues of nano-scale chemical or biochemical components, and it is hoped that knowledge of self-assembly gained at meso-scale, in simulation or in the laboratory, may lead to advances in nanotechnology.

2.1 Self-assembly of stable structures

A simulation environment has been developed [8, 9, 10] to explore self-assembly of two-dimensional mesoblocks, and in particular to discover how to produce stable self-assembled structures to serve as primitive building blocks for the self-assembly of more complex objects. This multi-agent environment has a large number of identical two-dimensional "sea" blocks, moving randomly in a liquid suspension, and interacting by sticking together or becoming unstuck. The blocks' edges have a positive (+1), negative (-1) or neutral (0) polarity, and blocks can stick together at edges with opposite polarity. A rich variety of self-assembled objects is possible by allowing edge polarities to change following a sticking or unsticking event under the control of an internal state machine. For simplicity all "sea" blocks are assumed to have the same fixed physical shape and the same state machine.

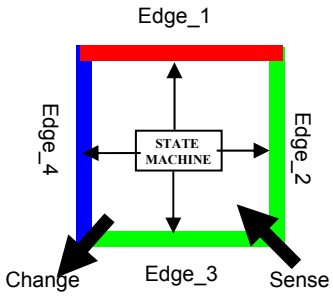

Figure 1: Structure of each agent in the mesoblock system. Edge_1 is positive (+1), Edge_4 is negative (-1), and Edge_2 and Edge_3 are neutral (0). The state machine implements rules for changing these polarities in response to "sticking" or "unsticking" events.

An example mesoblock is illustrated in Figure 1. In the simulation environment used in [8, 9] the blocks interact in the following manner:

1) Edges of opposite polarity stick together, generating a "sticking" event which is passed to the internal state machines of both sticking blocks.
2) Like polarities repel and will cause connected blocks to unstick, generating an "unsticking" event which also may be passed to the blocks' internal state machines, although this event has not been used by rule sets developed thus far.
3) Edges of neutral polarity will neither attract nor repel any other edges.
4) Blocks move randomly in the two-dimensional environment under discretised Brownian-like motion by steps which are multiples of the block width, with rotation of multiples of 90˚.

The state machine implements a set of rules dictating how each edge's polarity changes in response to sticking and unsticking events at any edge. For a given rule set many different structures may self-assemble; however, the simulation focuses on only one randomly chosen structure in the environment. Implementing state machines will require nano- or molecular-scale analogues with rudimentary intelligence; several authors have suggested mechanisms which may allow such properties, for example, those cited in [9].

A block will separate from a structure if there is no net "sticking force" to hold it on. The stability of a structure may be estimated in several ways. A measure of the proportion of neutral exterior edges indicates the likelihood of further structural growth, but doesn't account for the development of internal fractures. A measure of the total number of stuck

pairs of edges compared to repelling edges indicates crudely the likelihood of internal fractures. A correct estimate of the stability of structures of any size is a difficult problem which is hardly worth addressing due to the artificial nature of the simulation environment. Crude stability measures are sufficient to demonstrate principles of self-assembly. Correct measures should be developed when a realistic self-assembly system can be properly modelled; for example, Gerasimov et al. [10] report a simulation environment that accurately represents Newtonian motion of particles with separated charges, and such an environment can be enhanced by introducing other forms of well-characterised interactions.

Genetic algorithms with fitness functions chosen to target stability are used as a tool for exploring the range of possible rules sets for the state machine. Figure 2 shows some structures arising from one such rule set.

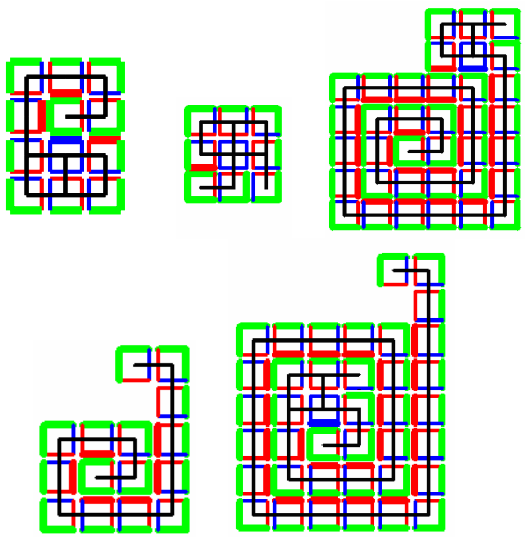

Figure 2: Several stable and semi-stable structures produced by a rule set obtained by a genetic algorithm. A line is drawn across pairs of "stuck" faces so the pattern of connections can be seen. This rule set favours spiral-type structures.

2.2 Top-down bottom-up methodology

In contrast to this rather undirected search for mesoblock rules that lead to stable and semi-stable structures, engineering design begins with a system goal and employs a top-down approach to formulate more achievable intermediate goals. In the mesoblock environment it is possible to meet in the middle – to use a bottom-up emergent process to create intermediate structures that are useful in fulfilling an overall goal. The mechanism chosen to achieve this is a structure that repeatedly creates another structure when immersed in a sea of mesoblocks. This is termed an "enzyme" structure and its properties are defined as follows:

1) An enzyme is a stable structure comprising several blocks.
2) Each block in the same enzyme has the same rule set.
3) Blocks in the environment ("sea" blocks), whilst identical, can have a different rule set from the enzyme blocks.
4) An enzyme must remain unchanged after the self-assembly process. During the process it may change in size and shape, but it must return finally to its initial state.

An example of an enzyme enabling a self-assembly process is shown in Figure 3. A different enzyme is put to work in in Figure 4 to produce regular structures of a desired size and shape. There are two large L-shape enzymes in this environment. These enzymes, and the rule sets of the sea blocks, are designed to ensure that L-shape structures of determined size are generated with neutral exterior edge polarities apart from two sites that function as open ends. When these structures, moving freely in the environment, meet at their open ends, they stick together and generate a rectangle whose dimensions are determined by the L-shaped structures produced by the enzymes. These dimensions are set by the positions of the "terminal" blocks in the enzymes, shown in black in Figure 4. Without external intervention only rectangles of a fixed size will be produced. However, external global messages may be used to shift the positions of the terminal blocks, and hence the size and shape of the rectangles produced. A combination of emergent outcomes and

simple engineering has created a rectangle factory. It is not a great stretch of the imagination to extend this idea to the manufacture of nano-scale electrical circuits or micro-electromechanical machines (MEMS).

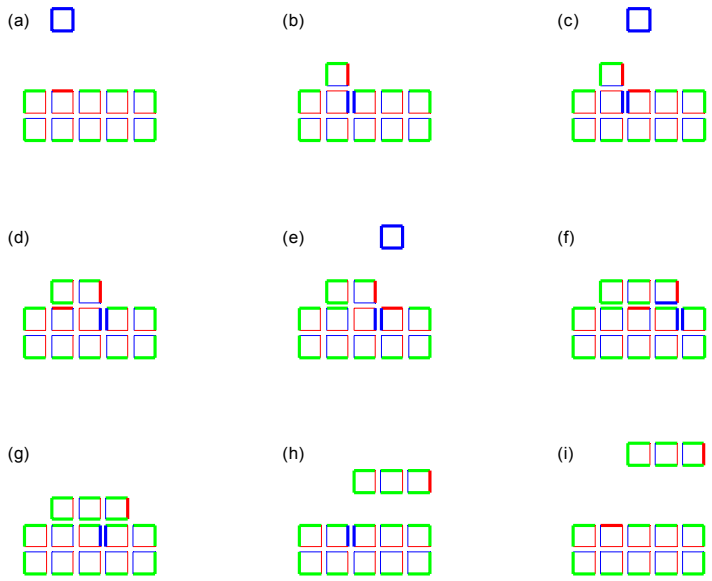

Figure 3: A simple example of an enzyme. This is an artificially constructed 10-block structure sitting in a "sea" of blocks with all negative sides and an identical rule set. This enzyme produces linear groups of three blocks with all exterior polarities neutral except for one end, which is positive.

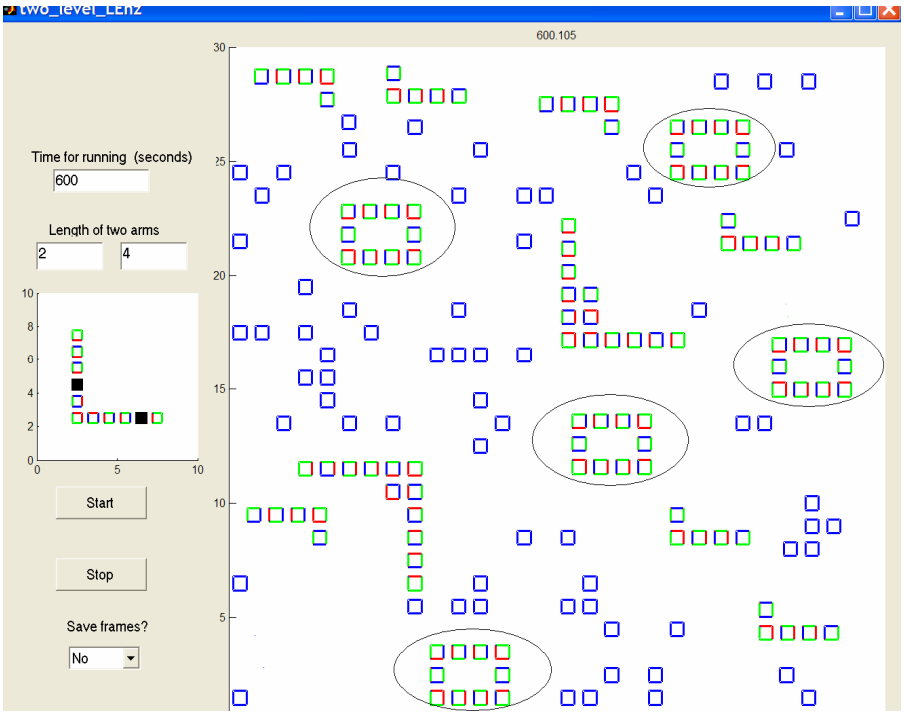

Figure 4: Self-assembled rectangles of size 3 by 4 (circled). Self-assembly was assisted by an enzyme (shown on the left panel) that generates L-shaped objects. These then joined when random motion brought them close enough to each other.

This illustrates a balance between science and engineering approaches: science to develop new knowledge of what is achievable by working from the bottom-up, and engineering to reduce overall goals to simpler goals using existing knowledge. The expansive solution space created by emergent behaviour is not entirely removed by reductionism which, at the same time, is necessary to some extent for robust engineering. Successful design of complex systems is possible when this balance can be achieved, which depends on identifying an intermediate layer of entities at the meeting point. This theme is explored in more depth by Poulton et al. [9].

3. MACRO-SCALE: DISTRIBUTED ENERGY MANAGEMENT

Electricity networks are stressed by peak summer and winter loads due largely to domestic heating and cooling. The State of NSW in Australia, for example, expects to run out of peak generating capacity by 2010/11 and base (average) capacity by 2016/17. An alternative to installing new centralised generation, most likely coal-fired and increasing Australia's greenhouse gas output still further, is to manage demand more intelligently and incorporate greenhouse-friendly distributed generation technology. Flexibility, engagement with the customer, and scalability would all be enhanced by the use of intelligent software agents, installed locally, to control customer loads and generators [11]. The use of deployed multi-agent technology, measuring, acting, and communicating in the real world rather than a simulated environment, is increasingly of interest in distributed energy [12] and other domains, offering the prospect of harnessing complexity to achieve new and useful emergent behaviours. At the same time, harmful behaviours must be avoided; the failure of the north-eastern US grid in August 2003 demonstrated that electricity networks with centralised generation and regional interconnections are complex systems subject to instability when under stress [13]. Increased levels of distributed generation, located close to load centres, and local intelligence designed to encourage the formation of sustainable "minigrids" are seen as important contributors to the security of future electricity networks.

3.1 Coordination by planning

The purpose of coordinating distributed energy agents is to aggregate sufficient quantities of distributed capacity to be of strategic value to the electricity industry. Retailers exposed to volatile wholesale prices and network businesses making infrastructure investment decisions would be significant beneficiaries of such aggregates. One way to express the aggregation goal is as a problem of cost minimisation across a set of loads and generators controlled by local agents and connected to the electricity grid at market rates. A cap on the total power drawn from the grid that can be a local offering to a large-scale aggregation of distributed capacity.

Simulated experiments have used genetic algorithms to optimise load and generator operation during a planning interval based on predicted environmental and market conditions [14]. Predicted outdoor temperature determines the behaviour of heating/cooling loads, predicted solar intensity and wind speed determine the capacity of photovoltaic and wind generators, and predicted market price determines the most cost-effective mix of generation technologies. The resulting switching plans for loads and generators would be in force through one or several market cycles during which the achieved cost saving or demand cap would be valuable. This is not a real-time algorithm because genetic optimisation of plans requires a significant part of a market cycle (30 minutes in Australia). To achieve soft-real-time operation, plans are optimised for future market cycles while executing the previously optimised plans.

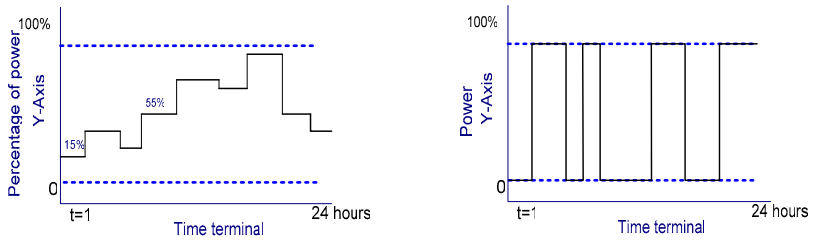

Figure 5: Resource (generator or load) actions during one day as parameterised for the genetic algorithm, a continuously variable resource on the left and a switched resource on the right.

Generators and loads are represented by their physical model and a sequence of states due to switching actions. The physical model for a cool room or refrigerator, for example, includes the thermal mass and the heat-transfer capacity of the compressor implicitly in a linear recursion on discrete time steps. Figure 5 illustrates the two possible kinds of switching permitted in plans: continuously variable between fully on and off, represented as a fraction of fully on, and discretely switched between fully on and off, represented as a binary value. When acting independently, heating/cooling loads operate a control loop to maintain temperature between given limits, as shown in Figure 7 for three loads, and there is no management of the total power drawn. Figure 8 shows what is achieved by planned coordination. Genetic optimisation of this kind has been applied, with some adaptation, to systems of up to 500 generators and loads.

3.2 Real-time coordination

Coordination by planning suffers from a lack of real-time responsiveness and the need for a central agent or computer to run the genetic algorithm for each market cycle. As an alternative, a multi-agent algorithm has been developed to maintain a capped demand by continuously coordinating switching actions within a group of agents. This algorithm does not yet include any optimisation of agent responses, so the resulting group behaviour may be inferior when compared to the planned approach. However, its speed is more than adequate compensation.

Regrettably, details of this algorithm cannot be published yet due to commercial sensitivities, but some preliminary results can be shared. The upper 6 graphs in Figure 9 show temperature and power for three refrigeration loads. The bottom graph shows the total power consumption averaged into 5-minute market-dispatch cycles, which are intervals during which generation in Australia is planned to meet the predicted load. The desired load cap of 5 kW is exceeded in 3 out of 6 cycles. Figure 10 shows the result of the multi-agent algorithm which causes agents to shift their times of operation away from cycles these 3 cycles without causing new excesses in the other cycles.

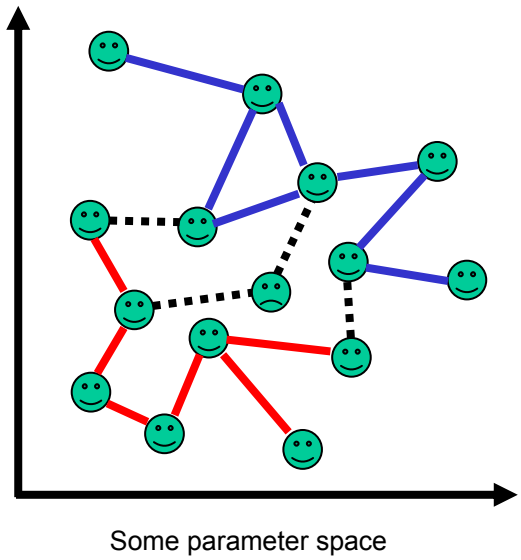

Figure 6: Decentralised clustering in action: two clusters are forming and one unhappy agent is not yet committed to either one.

Another direction of present research is to adapt decentralised clustering to coordinate distributed energy agents. This is an attractive multi-agent algorithm because it has been shown to scale better than a centralised algorithm performing the same task [15]. Through local comparison of agents' properties in a connected network, as shown schematically in Figure 6, clustering forms groups of agents that are similar with respect to some properties. Coordination of switching actions within such groups may be achieved using algorithms such as those discussed above; it may also be that clustering alone, in its more sophisticated forms, is capable of achieving detailed coordination. Whichever direction this research takes, a key challenge will be to find a balance of (top-down) engineering and (bottom-up) scientific principles like that achieved in at least some instances of mesoblock self-assembly.

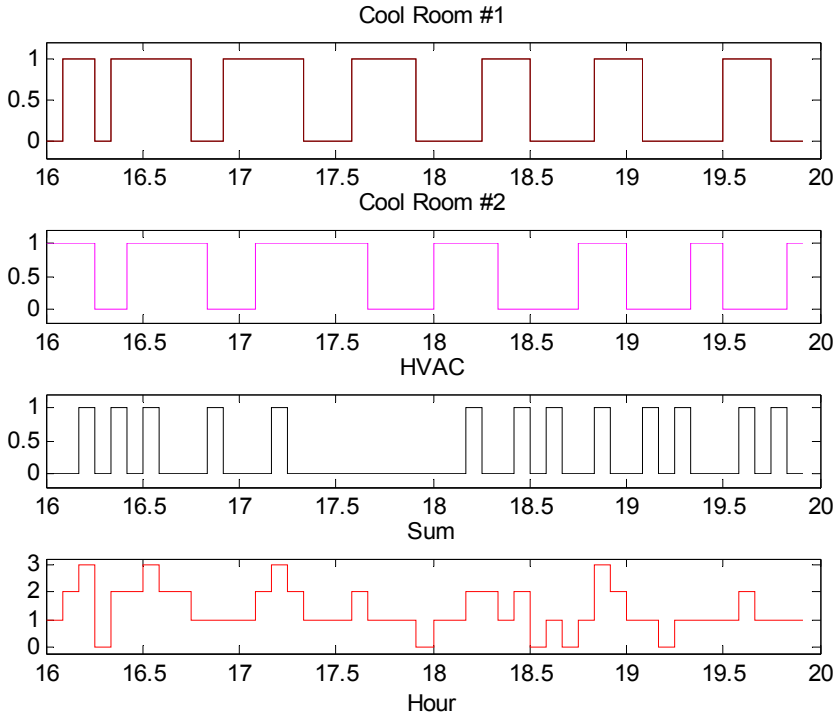

Figure 7: Uncoordinated switching of three heating and cooling loads during a four-hour period (units 0=off, 1=on).

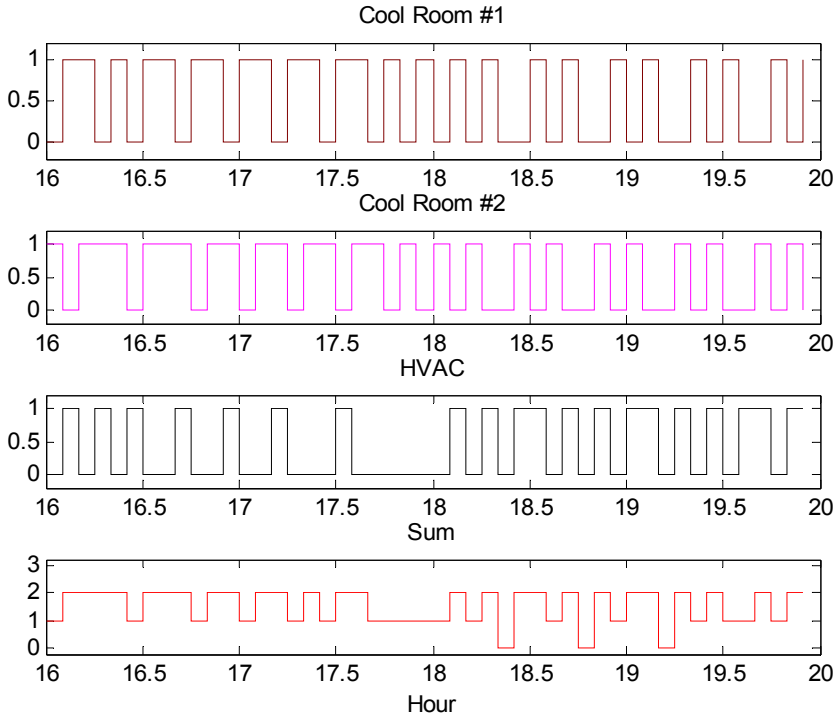

Figure 8: Coordinated switching: in the lower graph the three loads are never simultaneously on.

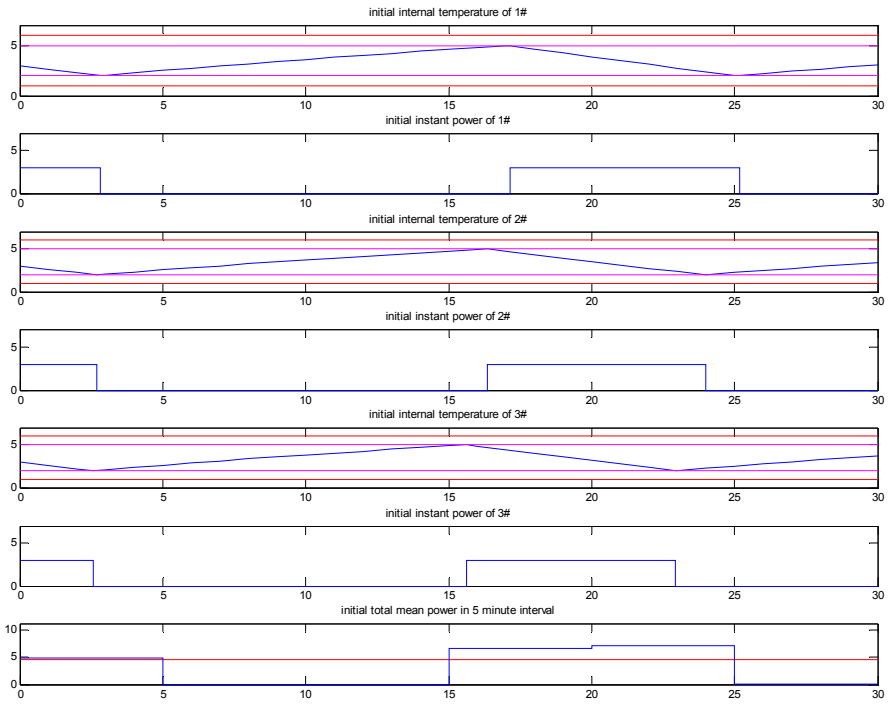

Figure 9: Three uncoordinated refrigeration loads (units kW and degrees Celsius against time in minutes). The lower graph shows total power consumption averaged into 5-minute intervals, and on this basis the specified 5 kW demand cap is exceeded 3 times.

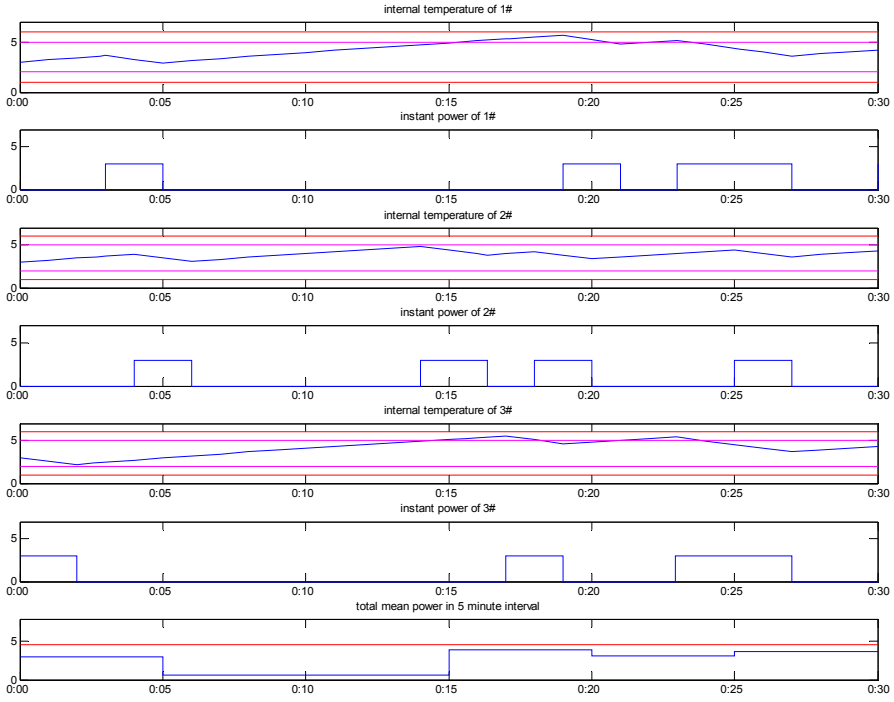

Figure 10: Three refrigeration loads coordinated using a real-time algorithm to meet the specified 5 kW demand cap.

4. CONCLUSION

Possibilities for nano-scale self-assembly can be explored using a simulation environment that idealizes the nature and interactions of nano-scale particles using a mesoblock analogue. The idealization allows design methods for complex systems to be explored in isolation from the complications of implementation in a real physical, chemical, or bio-chemical environment. A two-layer methodology shows promise while implementation of such self-assembly techniques requires much more work. Electricity networks have increasing levels of distributed generation and present an opportunity for intelligent coordination to relieve network and market pressures. Solutions to this complex-system design problem are immediately applicable using multi-agent technology that is ready for deployment to electricity customers' loads and generators, calling for a pragmatic approach. It is valuable to compare these apparently disparate environments and seek design principles that help to make good use of the diversity of possible emergent behaviours.

A comparison in broad terms is tabulated in Figure 11. In the self-assembly environment intermediate entities are created by a process of emergence that could then be used in a conventional construction process. Intermediate mesoblock entities are of two kinds: stable structures that self-assemble without assistance, and "enzymes" that assist the formation of stable structures with greater repeatability and sometimes at a much greater rate. Self-assembly is governed by random interactions among a sea of mesoblocks in a simulated liquid. Distributed energy agents do not move in physical space, yet their states change due to local imperatives determined by environment and usage, so they may be said to move independently in a state space. Their interactions are governed by a communication protocol that is random only through its lack of synchronicity, and the link between communication and state is governed by algorithms unrelated to motion in a liquid. Nevertheless, there is a desired distributed energy "structure" which is a group of agents coordinated to achieve an aggregated capacity, stability being maintained by either the repeated evolution of plans or by a real-time coordination technique. It may be that decentralized clustering is an effective means of constructing stable structures of this kind, and it may also be that comparison with self-assembly gives valuable insight to this research. This would be a sign that an engineering methodology for complex systems was imminent.

Feature	**Self-assembly environment**	**Distributed energy management**
Agents	Mesoblocks	Loads and generators
Movement	In a liquid	In a state space
Communication	By contact	By asynchronous messaging
Senses	Sticking and unsticking	Environment and user preferences
Events	Sticking and unsticking	Switching on and off
Agent intelligence	Internal state machine implementing rules for changing face polarities	Load and generator models, control loops, rules for handling messages
System goal	Useful structures	Safe network or stable wholesale price
Intermediate goal	Small stable structures	Groups with aggregated capacity
Intermediate entities in two-layer hierarchy	Stable structures or enzymes	Perhaps aggregation groups?
Emergent process	Sticking and unsticking as mesoblocks come close enough for their edge polarities to interact	Switching on and off or joining clusters in response to local imperatives or messages
Design technique	Manual or evolutionary optimization	Manual or evolutionary optimization

Figure 11: Comparison of self-assembly and distributed energy environments. A two-layer hierarchy has not yet been rigorously developed for distributed energy management.

ACKNOWLEDGEMENTS

The authors are pleased to acknowledge their gratitude to other team members who have contributed to this work: Vadim Gerasimov, Phil Valencia, Peter Wang, and Astrid Zeman.

REFERENCES

1. V. Vemuri, *Modeling of Complex Systems: an Introduction*, New York, 1978.
2. H. Haken, *Information and Self-Organization: A Macroscopic Approach to Complex Systems*, Springer-Verlag, 1988.
3. C. Shalizi, *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*, PhD Thesis, University of Michigan, 2001.
4. H. Surmann, A. Kanstein, and K. Gosser, "Self-Organizing and Genetic Algorithms for an Automatic Design of Fuzzy Control and Decision Systems", *1st European Congress on Fuzzy and Intelligent Technologies (EUFIT 1993)*, Aachen, vol. 2, pp. 1097-1104, 1993.
5. H. de Garis, "Growing Arbitrary Shapes with Genetic Programming", *7th International Conference on Machine Learning*, pp. 132-139, 1990.
6. C. Goldman and S. Zilberstein, "Optimizing Information Exchange in Cooperative Multi-agent Systems", *2nd International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2003)*, Melbourne, 2003.
7. G. Whitesides et al., "Mesoscale Self-Assembly of Hexagonal Plates Using Lateral Capillary Forces: Synthesis Using the 'Capillary Bond' ", *J. Am. Chem. Soc.*, vol. 121, pp. 5373-5391, 1999.
8. Y. Guo, G. Poulton, G. James, P. Valencia, V. Gerasimov, and J. Li, "Designing stable structures in a multi-agent self-assembly system", *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2004)*, Beijing, pp. 405-408, 2004.
9. G. Poulton, Y. Guo, G. James, P. Valencia, V. Gerasimov, and J. Li, "Directed Self-Assembly of 2-Dimensional Mesoblocks using Top-down/Bottom-up Design", in Sven A. Brueckner, Giovanna Di Marzo Serugendo, Anthony Karageorgos, et al. eds., *Engineering Self-Organising Systems: Methodologies and Applications*, "State of the Art Survey", LNCS vol. 3464, p. 154, Springer Verlag, 2005.
10. V. Gerasimov, Y. Guo, G. James, and G. Poulton, "Self-assembly Simulation System", *4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05)*, Muroran, Japan, 2005.
11. G. Platt and G. James, "Multi-agent systems in distributed energy networks", *1st International Symposium on Energy, Informatics and Cybernetics (EIC 2005)*, Orlando, 2005.
12. J. K. Kok, C. J. Warmer, and I. G. Kamphuis, "PowerMatcher: Multiagent Control in the Electricity Infrastructure", *4th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2005)*, Utrecht, 2005.
13. U.S.-Canada Power System Outage Task Force, *Final Report on the August 14, 2003 Blackout in the United States and Canada*, U.S. Department of Energy, 2004.
14. Y. Guo, J. Li, and G. James, "Evolutionary Optimisation of Distributed Energy Resources", *18th Australian Joint Conference on Artificial Intelligence (AI 2005)*, Sydney, 2005.
15. E. Ogston, M. Van Steen, and B. A. Brazier, "A Method for Group Formation Among Decentralized Autonomous Agents", *Journal of Applied Artificial Intelligence*, vol. 18, no. 9-10, pp. 953-970, 2004.