

A method for validating and discovering associations between multi-level emergent behaviours in agent-based simulations

Chih-Chun Chen¹, Sylvia B. Nagl², and Christopher D. Clack³

¹ Department of Computer Science, University College London

² Department of Oncology and Biochemistry, University College London

³ Department of Computer Science, University College London

Abstract. Agent-based models (ABM) and their simulations have been used to study complex systems with interacting entities and to model multi-agent systems. Simulations are used to explore the dynamic consequences of these models. In many cases, the behaviours that are of interest are emergent ones that arise as a result of interactions between agents rather than the actions of any individual agent. In this paper, we propose a formalism for describing emergent behaviours at any level of abstraction based on the idea that event types can be defined that characterise sets of behavioural ‘motifs’. This provides the basis for a method for studying the associations between multi-level behaviours in simulations. There are two categories of hypotheses that we seek to address with respect to an ABM and its simulations:

- Hypotheses concerned with associations between emergent behaviours defined at various levels of abstraction.
- Hypotheses concerned with the links between parameter sensitivity / initial conditions and emergent behaviours e.g. the ABM is sensitive to a parameter x because x predisposes the system or part of the system to exhibit a particular (emergent) behaviour.

1 Introduction

Agent-based modelling (ABM) and simulation is a widely used technique for studying complex systems. Each agent in the simulation represents an instance of an entity that is being modelled, while different agent types in the ABM can be seen as representing different ‘species’ of entities. Agent types have specifications which determine the behaviour of an agent given its own state and/or the state of its environment, where an agent’s environment can consist of other agents, objects (e.g. coordination artifacts), and/or continuous media. Complex systems are those in which entities interact in a non-trivial fashion so that behaviours at different levels of abstraction can influence one another. The usual reductionist cause-effect model therefore can not be applied, since ‘causality’ can be at different levels. This has been formalised in information-theoretic interpretations of complexity and emergence e.g.

[12], [2], [14], where dynamics at lower resolutions have high predictive efficiency so as to make higher resolution states statistically redundant. In many cases, ABM and their simulations are used to investigate the behaviours that emerge at the whole system or global level given the set of rules defined at the agent level in the agent types. By observing that a particular system level behaviour emerges, we can say that the set of rules in the ABM are sufficient to generate the emergent behaviour. Sensitivity of the ABM to certain parameters can also be investigated with multiple simulations, where non-linear differences between simulations with different parameters reveal sensitivity.

However, hypotheses about the mechanisms underlying a particular system behaviour are often expressed in natural language and validated through visualising the simulation and observing the interactions. We seek to formalise this process so that such hypotheses can be formally expressed and validated computationally rather than by human observation.

At the same time, Systems Sciences e.g. Systems Biology seek to integrate several levels of abstraction in a single model. While there are formalisms for specifying hierarchically organised systems e.g. P-Systems [9], hierarchical petri nets [5], there has been little work on using such formalisms to represent spontaneously arising hierarchies (those which are emergent). This requires a more general theory, which we provide here using our formalism. By being able to formally describe behaviours at different levels of abstraction e.g. different resolutions, we can then detect these behaviours in simulation and validate hypotheses about the relationships between these multi-level behaviours (with system-level behaviours being at the highest level of abstraction). The methodology we propose treats each simulation as an individual in the population of simulations that can be generated by an ABM and uses sampling across simulations to determine associations between behaviours at different levels. Furthermore, such associations can give us insight into other properties of an ABM such as sensitivity to particular parameters.

The paper will be organised as follows:

- Section 2 describes our conception of models and simulations.
- Section 3 briefly introduces the theory of emergence on which our work is based, referring to recent work in complexity science to formalise emergent properties.
- Section 4 introduces the complex event formalism for describing multi-level behaviours in multi-agent simulations and proposes a methodology for determining associations between behaviours at different levels of abstraction based on this formalism.
- Section 5 summarises and concludes the paper.

2 Models, simulations, agents and agent types as functions

Central to our approach is the idea that an agent-based model (ABM) is a function which, given a set of arguments (e.g. parameter values, initial agent positions and states), returns a simulation. Simulation parameters

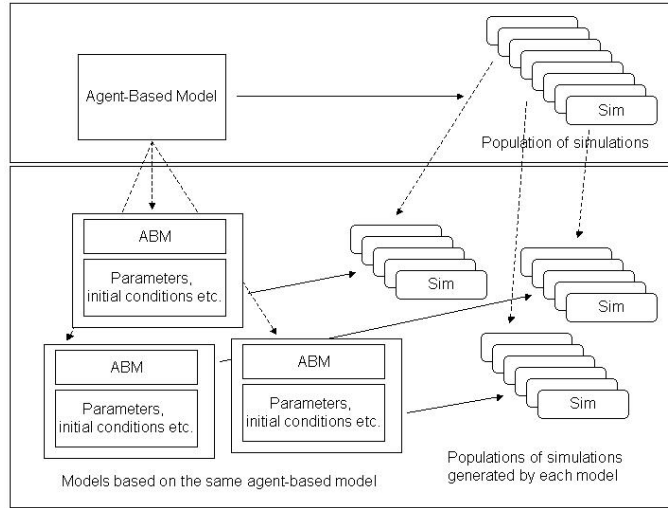


Fig. 1. An agent-based model (ABM) generates a population of simulations, Sim_{Total} . Different parametrisations/initial conditions etc. of the ABM can be regarded as different models X, Y, \dots , each of which generate a population of simulations $Sim_X, Sim_Y, \dots \in Sim_{Total}$, which are subsets of the population of simulations generated by the ABM.

can be treated as constraints that define subtypes of the ABM (i.e. more specific models). These ideas are illustrated in Figure 1.

We define an agent-based model as a function which takes the arguments: *Agents*, a set of agents and *Config*, a configuration defining the initial conditions (e.g. where each agent is situated, global and local variables etc.), and returns the simulation:

$$ABM \ Agents \ Config = Sim$$

where *Agents* and *Config* are themselves defined as functions, each agent (instance) a_0, \dots, a_n in *Agents* is a function, and each member of *Config* is a function c_0, \dots, c_n that determines the initial state of an agent and its position with respect to the other agents. Environmental objects, shared data spaces, global and local variables etc. are all members of *Agents* in this formalisation.

ABMs can also be parametrised so that given a set of parameters P , there are a set of simulations that satisfy the model i.e. there is a function *Model* for which:

$$Model \ ABM \ P(Sim)$$

returns true if the simulation belongs to the set of simulations that can be generated by $ABM \times P$ and false otherwise.

The ABM is made up of a set of functions called agent types A_0, \dots, A_n that determine which agents (instances) can be generated by the model so that each A_i returns an agent when given a c_i :

$$A \ c = a$$

An agent type is a function defining the behaviour of an agent given its own state Q_a and the state of its environment or neighbourhood Q_e (which might itself be made up of other agents' states). Usually, this is expressed as a set of rules or constraints. At a particular level of abstraction, an agent's behaviour can be viewed as an atomic state transition or event. We call these events *simple events* (*se*):

$$A \ Q \ O = se,$$

where

$$Q = Q_a \times Q_e,$$

and O is a function defining the level of abstraction at which the event is observed.

3 Emergence and emergent properties

The term 'emergence' is often used in complex systems discourse to refer to phenomena where a property or behaviour arises at some higher level of abstraction from the actions and interactions between lower level entities. However, defining the term has proved difficult, since complex systems are investigated from different perspectives and those working in different disciplines do not always agree on what counts as emergence [3]. However, we can distinguish two main categories of emergence theories:

- Information theoretic interpretations (see [13] for a review), which formalise the fact that subsets of a system's states can be statistically significantly related to one another due to the system's internal dynamics. For example, if a certain set of states is present in a time series, we may be able to predict the occurrence of some other set of states in the future.
- Definitions from the study of multi-agent systems and simulations (e.g. [1], [4], [6]), which tend to characterise emergence in terms of the specified/unspecified aspects of the system. Unspecified properties and behaviours arise from interaction between agents rather than as a consequence of a single agent's actions; these are termed emergent.

The latter of these requires that as well as being at a higher level of abstraction, emergent properties are also those that are in some way 'greater than the sum of their parts'. This has been formalised in language / grammar-based emergence definitions e.g. [4], [6] by requiring that the 'whole' language can not be generated by the same grammar that generates the 'parts' language. For example, Kubik defines a 'whole' language ($L(\bigcup_i P_i)$) and a 'sum of parts' language ($L(P_i)$), where P_i is the set of rules. Emergence corresponds to the case where an array is in $L(\bigcup_i P_i)$ (the whole language) but not in $L(P_i)$ (the sum of parts language):

$$L(\bigcup_i P_i) \supset \text{superposition}(L(P_i)) \quad (1)$$

Whereas $L(P_i)$ is derived by putting all the parts together and deriving configurations for every part separately and then putting results together, while $L(\bigcup_i P_i)$ is obtained by putting all parts together and then deriving configurations.

It is also worth clarifying the term ‘level of abstraction’, which is derived largely from information-theoretic view that different dynamics can emerge from observing a system at different scopes and resolution (as defined in [11]).

The **scope** of a representation of the system is the ‘set of components within the boundary between the associated system and its environment’ [11]⁴ (at a given resolution). The scope S of a temporally extended system can be considered to be made up of its temporal scope S_τ , which defines the set of moments of time over which the system is represented and a spatial scope S_x , which defines the set of components whose states are being considered.

Resolution is the number of states that can be distinguished: i.e., given the same scope, a higher-resolution (finer) representation will be able to distinguish a greater number of possible states. Again, there is both a spatial aspect R_x and a temporal aspect R_τ which together define the overall resolution R .

A level of abstraction is then a function of the scope and resolution, where a higher level of abstraction M has a greater scope and/or a lower resolution than a lower level of abstraction μ (see equations (2), (3) and (4)):

$$R_M \leq R_\mu \quad (2)$$

$$S_M \geq S_\mu \quad (3)$$

$$(R_M, S_M) \neq (R_\mu, S_\mu) \quad (4)$$

To summarise, we stipulate that as well as either having lower resolutions, greater scopes, or both, emergent behaviours are those that have not been explicitly specified.

4 Discovering associations between behaviours at different levels

In the complex event formalism, a behaviour corresponds to an event, where an event is a state transition defined at some level of abstraction. A distinction is drawn between the origin of an event (the set of rules from which it results) and its state transitions. As well as being able to describe state transitions at different levels of abstraction, it is also possible to describe the origins of events at different levels i.e. we can ask for a given event, how many rules were executed to give that event. This section first introduces the formalism and then describes a method for identifying, quantifying and then determining associations between behaviours at different levels.

⁴ The system’s environment is considered to be outside the scope of representation.

4.1 Complex events

In the ABM and multi-agent systems context, certain events occur due to a firing of a state transition rule (these are also known as agent ‘actions’). We call these *simple events* SE . The state transitions that result from simple events can be described at different levels of abstraction i.e. have different scopes. For example, a simple event (from a single rule) where the colour and size of an agent both change can be further decomposed into two simple events - one with the colour state change, the other with the size state change.

A complex event CE is a set of interrelated events, where relationships between events can be defined in any dimensions e.g. time, space:

$$CE :: SE \mid CE_1 \bowtie CE_2 \quad (5)$$

\bowtie denotes the fact that CE_2 satisfies a set of location constraints with respect to CE_1 . Conceptually, complex events are a configuration of simple events where the configuration can be defined in a hyperspace that includes time, physical space and any other dimensions. Location constraints define relationships between complex events, which can be represented as a coloured multi-graph, where the coloured nodes stand for event types and coloured edges stand for the different relationship types (sets of location constraints) existing between events. A simulation, therefore, is itself a (usually very big) complex event.

The state transitions of simple events can be described at different levels of abstraction. For example, a rule that causes state changes in components a , b and c can cause simple events $(q_a, q_b, q_c) \rightarrow (q'_a, q'_b, q'_c)$, $(q_a, q_b) \rightarrow (q'_a, q'_b) \dots$, $q_a \rightarrow q'_a \dots$ etc. We call this the scope of the event. Two simple events e_1 and e_2 in a system are said to be of the same type if (a) e_1 and e_2 result from the same agent rule and (b) the scope of e_1 is identical to the scope of e_2 i.e. for every component in which a state change occurs in e_1 , there is a component of the same type in which the same type of state change occurs in e_2 . Two complex events CE_1 and CE_2 are said to be of the same type if, for each constituent event e_1 in CE_1 there is exactly one event e_2 in CE_2 satisfying the same location constraints, and e_1 and e_2 are events of the same type⁵. The scope of a complex event’s state transition (the components in which state transitions occur) should be distinguished from the event scope, which is the minimal set of simple events of which the complex event is composed, i.e. the number of rule firings.

To specify a complex event type, we need to specify the types for each of the constituent events and the location constraints that hold between them. Depending on the semantic and expressivity requirements of different applications, it is possible to use formal expressions from different formal systems such as temporal logics to specify the location constraints (good reviews can be found in [8] and [10]). Complex event types can differ in specificity, with a fully determined complex event type CET_{Full} being defined as one whose constituent events are in a fully determined configuration. A partially determined complex event type CET_{Part} is

⁵ See [7] for a formal definition of types

then one with a partially determined configuration and therefore defines a set of complex events with fully determined configurations.

$$CET_{Part} = \{CET_{Full}\} \quad (6)$$

So a complex event type can be defined as either a simple event type, two complex event types with a defined relationship, or a set of complex event types:

$$CET :: SET \mid CET_1 \bowtie CET_2 \mid \{CET\} \quad (7)$$

The dimensions in which configurations are not fully specified lower the resolution of the complex event, with weaker constraints (greater ranges of possible values) implying a lower resolution in that dimension. More generally, the greater the number of complex event types with fully determined configurations that a complex event type contains, the lower its resolution. Given our definition of an emergent behaviour as one that (as well as being defined at a greater scope or lower resolution) is not explicitly specified in the ABM, we can say that the set of complex events that correspond to emergent behaviours are those that are *not simple events*. Similarly, the complex event types that represent emergent behaviours are those that are *not simple event types*, i.e.:

$$CET_{Emergent} = CET - SET \quad (8)$$

Having briefly introduced the formalism and its relation to emergent behaviours, we now describe how it can be used in the context of multi-agent simulations to determine correlations between behaviours at different levels.

4.2 Correlation studies to determine associations

By determining the degrees to which each complex event type (representing a behaviour at a particular level of abstraction) is observed in simulation, we can calculate the correlations between the complex event types and determine associations between behaviours at different levels. This can be done across all simulations or across classes of simulations that are themselves classified according to some behaviour exhibited (since association may itself be determined by another behaviour that is currently unknown). In other words, we are interested in what makes one simulation of a model exhibit a different behaviour to another simulation of the same model. To determine associations across simulations, the method can be summarised as follows:

1. Specify complex event types $CET_0, CET_1, \dots, CET_n$ that correspond to behaviours at different levels of abstraction.
2. Run x simulations of the agent-based model and determine the occurrence levels of each of the complex event types for each simulation $p_{CET_0}, p_{CET_1}, \dots, p_{CET_n}$. This gives x sets of n scores. The simplest occurrence level measure p_{CET_i} would be counting the number of times each complex event type occurs through the simulation, but other measures might be more suitable depending on the model and its aims.

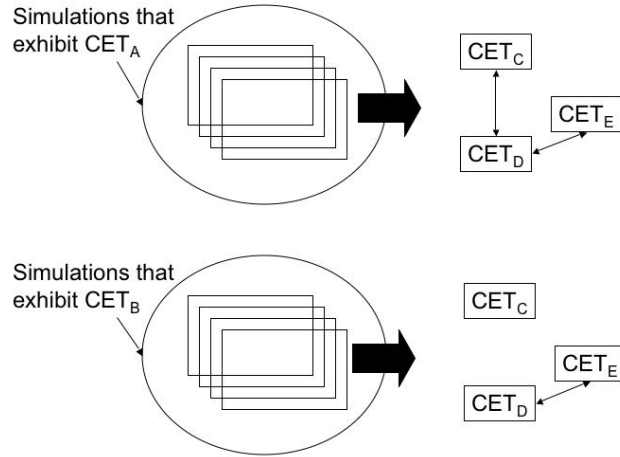


Fig. 2. Simulations with particular behavioural differences may also differ in the associations existing between behaviours at different levels. Double arrows indicate a positive correlation whereas a line connection indicates a negative correlation. In simulations that exhibit CET_A , CET_D is positively correlated with both CET_C and CET_E whereas in simulations that exhibit CET_B , CET_C is only positively correlated with CET_E while CET_E and CET_C are negatively correlated with each other.

3. Calculate the correlations between the p_{CET_i} scores across the simulations i.e. $r_{CET_0CET_1}, r_{CET_0CET_2}, \dots, r_{CET_nCET_{n-1}}$ and determine significance.

It is also possible that simulations that exhibit different behaviours also have different correlation patterns. To investigate this possibility, the same method can be applied but instead of grouping all simulations together when calculating the r values, we can have different sets of r values corresponding to the sets of simulations exhibiting different behaviours (see Figure 2).

We should emphasise here that associations between behaviours at different levels across simulations do not always indicate causal relationships; nor do they translate into ‘laws’. Instead, each simulation should be seen as a unique individual which may or may not have the same associations. For example, even if there is a positive correlation between CET_A and CET_B across simulations, it does not follow that a given simulation with a high degree of CET_A will exhibit a correspondingly high degree of CET_B (this holds for all correlation relationships e.g. we can not infer from the fact that high intelligence scores in childhood IQ tests tend to be associated with higher income that a particular individual with high intelligence scores will definitely have a higher income).

5 Summary and conclusions

In this paper we have introduced a method for studying associations between emergent behaviours at different levels of abstraction in agent-based simulations. This is based on a population view of agent-based simulations, where each simulation is treated as an individual but can also be multiply classified as belonging to more specific models e.g. having the same parameter settings, initial conditions, or according to the features/behaviours it exhibits.

The complex event formalism allows us to specify the emergent behaviours that we wish to identify in a simulation so that they can be detected computationally. This formalism, together with our treatment of simulations as multiply classifiable individuals, allows us to formulate and validate two categories of hypotheses:

- Hypotheses concerned with associations between emergent behaviours defined at various levels of abstraction.
- Hypotheses concerned with the links between parameter sensitivity / initial conditions and emergent behaviours e.g. the ABM is sensitive to a parameter x because x predisposes the system or part of the system to exhibit a particular (emergent) behaviour.

We are also able to analyse behaviours in terms of the rules that give rise to them. A distinction is drawn between:

- Simple events, which represent behaviours arising purely from the single execution of an agent rule and are hence explicitly specified.
- Complex events, which represent behaviours arising from execution of more than one agent rule (these can be from the same agent) i.e. the interactions between rules.

This distinction is based on established definitions of emergence in the field of multi-agent systems engineering. The decomposition of complex events gives us a means of determining the rule-based mechanisms underlying an emergent behaviour, something that would be of considerable benefit in the design and engineering of multi-agent systems.

References

1. E. Bonabeau and J. L. Dessalles. Detection and emergence. *Intellectica*, 2(25):85–94, 1997.
2. J. P. Crutchfield and D. P. Feldman. Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos*, 13(1):25–54, 2003.
3. J. Deguet, Y. Demazeau, and L. Magnin. Elements about the emergence issue: A survey of emergence definitions. *ComplexUs*, 3:24–31, August 2006.
4. Y. Demazeau. Steps towards multi-agent oriented programming. In *First International Workshop on Multi Agent Systems*, Boston, Mass., 1997.
5. R. Fehling. A concept of hierarchical petri nets with building blocks. *Lecture Notes in Computer Science; Advances in Petri Nets 1993*, 674:148–168, June 1991.

6. A. Kubik. Toward a formalization of emergence. *Artificial Life*, 9:41–66, 2003.
7. F. W. Lawvere and S. H. Schamiel. *Conceptual mathematics: a first introduction to categories*. Cambridge University Press, 1997.
8. F. Moller and G. Birtwistle. *Logics for concurrency: Structure versus automata*. Lecture Notes in Computer Science. Springer, 1996.
9. G. Paun. From cells to computers: computing with membranes (p systems). *Biosystems*, 59:139–158, 2001.
10. C. A. Rouff, M. G. Hinchey, J. Rash, W. Truszkowski, and D. Gordon-Spears, editors. *Agent technology from a formal perspective*. Springer-Verlag London, 2005.
11. A. Ryan. Emergence is coupled to scope, not level. *Nonlinear Sciences*, 2006.
12. C. Shalizi. *Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata*. PhD thesis, University of Michigan, 2001.
13. C. R. Shalizi. *Methods and Techniques of Complex Systems Science: An Overview*, chapter Methods and Techniques of Complex Systems Science: An Overview, pages 33–114. Springer, New York, 2006.
14. C. R. Shalizi and K. L. Shalizi. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In M. Chickering and J. J. Halpern, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference*. AUAI Press, 2004.