Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

UNDERSTANDING COMPLEX SYSTEMS: USING INTERACTION AS A MEASURE OF EMERGENCE

Claudia Szabo

School of Computer Science
The University of Adelaide
Adelaide, Australia

Yong Meng Teo Gautam K. Chengleput

Department of Computer Science
National University of Singapore
Singapore

Abstract

Understanding the behavior of complex systems is becoming a crucial issue as systems grow in size, and the interconnection and geographical distribution of their components diversifies. The interaction over time of many components often leads to emergent behavior, which can be harmful to the system. Despite this, very few practical approaches for the identification of emergent behavior exist, and many are unfeasible to implement. Approaches using interaction as a measure of emergence have the potential to alleviate this problem. In this paper, we analyse absolute and relative methods that use interaction as a measure of emergence. Absolute methods compute a degree of interaction that characterizes a system state as being emergent. Relative methods compare interaction graphs of the system state with interaction graphs of systems that have been shown previously to exhibit emergence. We present these approaches and discuss their advantages and limitations using theoretical and experimental analysis.

1 Introduction

Complex systems often exhibit behavior that cannot be reduced only to the behavior of their individual components, and component interactions often result in new and unexpected properties (Davis 2005, Johnson 2006, Mogul 2006). These *emergent properties* are becoming crucial as systems, and in particular software systems, grow both in size (with respect to the number of components and their behavior and states), but also in coupling and geographic distribution (Bedau 1997, Holland 1999, Johnson 2006, Mogul 2006). A plethora of emergent properties examples have been observed and studied, from flocks of birds, ant colonies, to the appearance of life and traffic jams. In software systems, connection patterns have been observed in data extracted from social networks (Chi 2009) and trends emerge in big data analytics (Fayyad and Uthurusamy 2002). More malign examples of emergent behavior include the Ethernet capture effect in computer networks (Ramakrishnan and Yang 1994), and load-balancer failures in a multi-tiered distributed system (Mogul 2006). As emergent properties may have undesired and unpredictable consequences, systems that exhibit them become less credible and difficult to manage.

While emergent properties have been the focus of research since the 1970s (Bedau 1997, Cilliers 1998, Gardner 1970, Holland 1999, Seth 2008), very few methods are known for their identification, classification, and analysis (Chen et al. 2007, Kubik 2003, Seth 2008, Szabo and Teo 2012). Moreover, existing methods are usually employed only on simplified examples that are not often found in real life. Approaches can be classified broadly from two orthogonal perspectives. In the first perspective, approaches propose to identify emergence as it happens (Kubik 2003, Szabo and Teo 2012), and aim to use formal models of calculated composed model states. This requires the identification of attributes that describe the system components, or the *micro level*, and the system as a whole, or the *macro level*, and the relationships and dependencies between these levels. This allows the specification of emergence as the set difference between macro level and the micro level, but they are difficult to capture and computationally expensive. In contrast, the second

perspective uses a definition of an observed emergent property and aims to identify its cause, in terms of the states of system components and their interaction (Chen et al. 2007, Seth 2008). A key limitation of this *post-mortem* perspective is that a prior observation of an emergent property is required, and that emergent properties need to be defined such that the macro level can be reduced or traced back to the micro level. The above approaches are demonstrated using simple models such as flocks of birds or predator-prey but have limiting assumptions and constraints when applied to more complicated systems. For example, most approaches do not consider mobile agents (Kubik 2003), assume unfeasible a-priori specifications and definitions of emergent properties (Szabo and Teo 2012), or do not scale beyond models with a small number of agents (Chen et al. 2007, Kubik 2003). In the multi-agent systems community, approaches focus more on the engineering of systems to exhibit beneficial emergent behavior and less on its identification and validation (Bernon et al. 2003, Jacyno et al. 2009, Salazar et al. 2011).

An appealing approach for the formalization of emergence is proposed by Kubik (Kubik 2003), who introduces the idea of array grammar systems as a formalism for capturing agent behavior. An agent's behavior is defined as a language produced by a grammar, and the behavior of the entire system is obtained by multiple grammars writing symbols on a common tape. Emergence is then defined as the set difference between the language of all agents written on the common tape (L_{whole}) and the superimposition of all individual agent languages (L_{sum}). Following the above classification, L_{whole} represents the macro level as system states due to agent interaction, while L_{sum} represents the superimposition of all micro levels as systems states composed from all combinations of agent attributes. In our previous work (Teo et al. 2013), we extended this formalism to model mobile agents and agents of different types. Our study showed that the calculation of L_{sum} is very prohibitive in terms of computational cost and, in most cases, this calculation is unnecessary as the superimposition of individual agent behaviors often leads to illegal states. Instead, we focus on identifying the states from L_{whole} that are not in L_{sum}, and thus in the set of emergent states.

Following the insight that interaction is one of the main causes of emergent behavior (Chan 2011, Melo and Veloso 2009, Jacyno et al. 2009), in this paper we evaluate methods of identifying the set of emergent states L_ξ from L_{whole}. All methods use interaction as a handle to identify emergence, in two main ways. *Absolute methods* calculate a degree of interaction between agents and consider only system states with a high degree of interaction as emergent. The degree of interaction can be calculated as a distance metric or using interaction counts. *Relative methods* compare the system state with states from other systems that have been previously shown to exhibit emergence. They use an interaction graph formalism and transform the comparison into a shape similarity problem that can be solved using various distances. We present these methods and analyse their advantages and disadvantages, both theoretically and through experimental analysis. The contributions of this work include an in-depth analysis of agent interaction as a measure of emergence and a discussion of its applicability and limitations.

2 Related Work

An emergent property can be defined as "a property of an assemblage that could not be predicted by examining the components individually" (Bedau 1997). Significant research interest in the past thirty years has led to the identification and analysis of various characteristics of emergent properties and behavior, such as *radical novelty*, as properties not previously observed in the system; *self-organization*, in which system components organize themselves without pre-defined rules; and *evolution* as the product of a dynamical process that continuously changes (Holland 1999). Research has also identified *interaction*, usually short-ranged, where the interaction happens only with close neighbors, as one of the causes of emergence. A large number of approaches distinguish between a *whole* (or *macro*) system perspective, in which system properties are captured and analyzed, and an *individual* (or *micro*) system perspective, in which properties and interactions of individual components are considered (Holland 1999). The distinction between the micro and the macro perspective aids in emergence definitions that look at the"whole as more than the sum of its parts" (Bedau 1997). More formally, *weak emergence* is defined as the properties of the whole, or the macro level, which result from the properties of the parts, or the micro level, and the interaction at the

micro level. Moreover, it is not trivial to infer the properties of the whole, and that extensive simulation studies are required to analyze and understand emergent behavior (Holland 1999).

Emergent behavior analysis follows two main perspectives. If an unexpected behavior or property has been observed then *post-mortem* or *trace* analysis can be performed on the system execution logs in order to determine its causes. This requires that detailed system logs are kept and that these logs permit analysis following an operational definition of emergence. In contrast, *live* or *on the fly* system analysis proposes to identify emergent behavior as it happens, i.e., as the system is executing, and without any system expert knowledge. Variations of this perspective analyze the system logs *after* the system has finished execution, but without any prior knowledge about emergence (Kubik 2003). We distinguish between these perspectives in our analysis, and also group existing work into three main categories, namely, *grammar-based*, *variable-based*, and *event-based*, as discussed below.

Grammar-based methods are live emergence analysis methods which aim to identify emergence in agent-based systems using two grammars, L_{whole} to describe the properties of the system as a whole, and L_{sum} to describe the properties obtained from the reunion of the parts, and a definition of emergence as the difference between L_{whole} and L_{sum} (Kubik 2003). L_{whole} and L_{sum} can be easily calculated as sets of words that are constructed following the defined output in agent behavior descriptions. This method does not require a prior observation of the system, which makes it suitable for large-scale composed models where such observations are almost impossible. However, the nature of the formalism and the computation of the composed model states make it difficult to scale, despite recent attempts at reducing state space explosion. Teo et al. proposed dividing L_{sum} and L_{whole} into smaller state spaces but the calculation of these spaces is still an open problem (Teo et al. 2013). *Event-based methods* are post-mortem analysis approaches in which behavior is defined as a series of simple and complex events that change the system state (Chen et al. 2007). Complex events are defined as compositions of simple, atomic events. Emergence is defined by a system expert as a complex event, and the approach focuses on determining its causes in terms of the sequence of complex and simple events in the system.

In *variable-based methods*, a specific variable or metric is chosen to describe emergence. Changes in the values of this variable signify the presence of emergence properties (Seth 2008). For example, the centre of mass of a bird flock could be used as an example of emergence in bird flocking behavior, as shown in (Seth 2008). The approach uses Granger causality to establish the relationships between a macro-variable and micro-variables and propose the metric of G-emergence. This has the advantage of providing an easy to implement process for emergence identification. However, the approach requires system expert knowledge. Szabo and Teo (Szabo and Teo 2013) propose the use of reconstructability analysis to determine which components *interacted* to cause a particular emergent property (defined through a set of variables). They identify the interactions that cause birds to flock (Reynolds 1987), the cells that cause the glider pattern in Conway's Game of Life (Gardner 1970) , and the causes of traffic jams. However, the accuracy of their method is heavily dependant on the choice micro and macro level variables. Other approaches use metrics such as Shannon entropy (Gershenson and Fernandez 2012, Prokopenko, Boschetti, and Ryan 2009) and variety (Holland 2007, Yaneer 2004) to measure emergence in a system. These approaches do not require system expert knowledge as they employ general definitions that are rooted in complex systems theory. However, to date they have only been applied to simple examples. Gore and Reynolds propose a taxonomy for analyzing emergent behavior based on reproducibility, predictability, and temporality (Gore and Reynolds 2007). However, the process of identifying the emergent behavior according to these criteria is not addressed.

3 Interaction as a Measure of Emergence

Our approach (Teo et al. 2013) extends Kubik's grammar-based approach (Kubik 2003) to calculate L_ξ, the set of emergent property states as:

$$L_\xi = L_{whole} - L_{sum} \tag{1}$$

where L_{whole} describes all possible system states due to agent-to-agent and agent-environment interactions, and L_{sum} is the sum of all individual agent behaviors, without considering agent interactions. By representing a multi-agent system (MAS) as a cooperating array grammar system as discussed above, an agent behavior can be formalized as a grammar that produces the words that the agent writes on a tape. Thus, L_{whole} is obtained as the reunion of all words that agents write on the common tape, and L_{sum} as the superimposition of individual agent words, using a pre-defined superimposition operator that computes all possible combinations of agent words. We proposed an agent-based formalism that improves this approach by considering mobile agents and multiple agents types (Teo et al. 2013). The computation of L_ξ requires the calculation of L_{whole} by observing the common tape and recording all words written on it, and of L_{sum} by calculating all the possible combinations of individual agent behaviors, resulting in a formal approach in identifying emergence as it appears, without prior knowledge of emergent behavior. However, as in Kubik's approach, the computation of L_{sum} in unfeasible. This is because all possible combinations of individual agent states are considered following a defined superimposition operator, without including system-defined rules.

In this paper, we propose a new perspective that significantly reduces the computational cost of calculating L_ξ. Instead of deriving L_{sum} we propose to directly identify the states L_ξ from L_{whole}, using interaction-based methods. We classify these methods in two main categories, namely, absolute and relative. *Absolute methods* use interaction metrics of the system under analysis and follow a direct causality relation between interaction and emergence, without the need for comparison with other systems or measures. Since our approach in formalizing emergence is based on the interactions among entities and entities and their environment, our key idea is to differentiate the strength of these interactions. For a given system state, we propose to quantify the degree of interaction, $\mathbb{D}$, where zero denotes no interaction and one denotes the strongest interaction. This has two key advantages. Firstly, L_ξ can be directly obtained by not considering system states with $\mathbb{D} = 0$. Secondly, $\mathbb{D}$ facilitates the analysis and interpretation of L_ξ through differentiating emergent property states based on the strength of interaction. In the cases where L_ξ is large, the observer can focus on a much smaller number of emergent property states by using a high degree of interaction $\mathbb{D}$, and with a higher probability of identifying emergent properties. We investigate two approaches to determine the interaction metric, namely, distance-based and interaction counts. In contrast, *relative methods* establish a measure of interaction and compare it with pre-defined thresholds or with interaction metrics of other systems that have previously exhibited emergent behavior. We have previously proposed the use of interaction graphs (Birdsey and Szabo 2014) and Hausdorff distances for this comparison, and we present this method here for comparison. In the following, we present a multi-agent system notation and discuss each type of method and its associated metrics.

3.1 Problem Formulation

A multi-agent system consists of n agents (A) of m types interacting in an environment. The environment (E) is a part of the system that lies outside the agents and can be regarded as a platform for agent interactions. For simplicity, we assume that E has no behavior rules and changes only as a result of agent actions. Changes of the environment, in turn, impact the agent behavior. For discussion, we model E as a 2D grid[1] that is subdivided into c units called cells (e). Changes in the environment are therefore changes of the states of cells. For example, a cell turns from "occupied" to "free" when the agent that occupies the cell moves to another cell. V_e denotes the set of possible states of cell e. Similarly, V_E denotes the set of possible cell states, and $V_e \subseteq V_E$. In addition, the environment state is made up of the states of all cells. $s_e(t)$ and $S_E(t)$ denote the state of cell e and the environment E at time t respectively.

Agents are autonomous entities characterized by a set of attributes, such as the location of an agent in a spatial environment or the distance an agent travels in a time step. The attributes' values at a point in time represent the state of an agent at that time. The state of an agent changes as dictated by the behavior of the agent. Agents act and interact with other agents and the environment according to agent-specific

[1]E can be easily extended to model other topologies.

rule sets, which contain behavior rules that define the agent's state in the next time step. A rule is executed if its condition is met. Conditions may include the state of the agent, but also the states of other agents.

We formalize a multi-agent system consisting of a set of agents and their environment as an extended cooperating array grammar system where context-free grammars represent agents, and a two-dimensional array of symbols represents the global environment. The abstraction of an agent as a grammar system follows from the idea that the agent output to all other agents can be interpreted as a communication language. This communication language is generated by a grammar, which can thus abstract an agent (Kubik 2003). An agent's language is comprised of the set of words that the agent outputs on the tape. Each agent behavior or grammar has its own rewriting rules defining how the grammar cooperates with the other grammars and with the array, i.e. rewrite the symbols on the array. A grammar-based agent system (*GBS*) of m agent types and a total of n agents $A_{11},\ldots,A_{1n_1},\ldots,A_{mn_m}$ interacting in a 2D grid environment of c cells is defined as follows:

$$GBS = (V_A, V_E, A_{11},\ldots,A_{1n_1},\ldots,A_{mn_m}, S(0)) \tag{2}$$

where V_A denotes the set of possible agent states for all agent types, V_E denotes the set of possible cell states, A_{ij} denotes an agent of type i ($1 \leq i \leq m$) and instance j ($1 \leq j \leq n_i$), and $S(0)$ denotes the initial system state. A system state is composed of the state of the environment and the states of all agents.

For the environment (E),

$$V_E = \bigcup_{e=1}^{c} V_e \tag{3}$$

where V_e denotes the set of possible states of cell e and c is the total number of cells. The state of the entire environment is made up of the states of all its cells. The states of cell e and the environment E at time t are $s_e(t) \in V_e$ and $s_E(t) \in V_E$ respectively.

For the agents (A),

$$V_A = \bigcup_{i=1}^{m} V_{A_i} \tag{4}$$

where V_{A_i} denotes the set of possible states for agents of type i.

Agent of type i ($1 \leq i \leq m$) and instance j ($1 \leq j \leq n_i$), A_{ij}, is defined as follows:

$$A_{ij} = (P_i, R_i, s_{ij}(0)) \tag{5}$$

where P_i denotes attribute set for agents of type i, R_i denotes the set of behavior rules for agents of type i, and $s_{ij}(0)$ denotes the initial state of the agent. A_{ij} has an initial state $s_{ij}(0) \in V_{A_i}$. The system state at time t ($S(t)$), is composed of state of the environment ($s_E(t)$) and states of agents ($s_{ij}(t)$) at time t. Hence,

$$S(t) = s_E(t) \bigcup_{\forall i \forall j} s_{ij}(t) \tag{6}$$

3.2 Absolute Methods: Degree of Interaction

We aim to determine L_ξ without deriving L_{whole} and L_{sum}, by computing absolute metrics about the agent-based system and using interaction as a handle to define these metrics. In this section, we propose $\mathbb{D}$, a degree of interaction metric that quantifies the interaction in an agent-based systems. This has two key advantages. Firstly, L_ξ can be directly obtained by removing system states with $\mathbb{D} = 0$ from L_{whole}. Secondly, $\mathbb{D}$ facilitates the analysis and interpretation of L_ξ through differentiating emergent property states based on the strength of interaction. We analyze two methods to calculate $\mathbb{D}$, namely, a distance-based calculation and an interaction count-based calculation.

3.2.1 Distance-based Calculation

In systems where the interaction between agents is facilitated by their proximity, for a given state $S(t)$, we define the degree of interaction $\mathbb{D}$ as:

$$\mathbb{D}(S(t)) = \sum_{i,j=0}^{n,m} min\_dist_{ij} \tag{7}$$

where min_dist_{ij} is the distance from agent A_{ij} to the nearest agent. The meaning of "distance" varies depending on the system under study, and can include Euclidian distance for systems where agents are geographically close, but also similarity metrics in systems in which only agents with similar attributes and behaviors interact: a Euclidian distance metric could be used for analyzing a flock of birds model, while a similarity-based metric could be used to analyze a computer network. Considering $\mathbb{D}_{min}$ and $\mathbb{D}_{max}$ as the minimum and maximum distances for the system respectively, we can normalize $\mathbb{D}(S(t))$ as:

$$\mathbb{D}'(S(t)) = 1 - \frac{\mathbb{D}(S(t)) - \mathbb{D}_{min}}{\mathbb{D}_{max} - \mathbb{D}_{min}} \tag{8}$$

Calculating states in L_ξ For each system state $S(t)$, we calculate $\mathbb{D}'(S(t))$ according to Equation 8. States can then be grouped to different interaction subsets based on the values of $\mathbb{D}'(S(t))$, allowing the separation of L_ξ according to different interaction strengths.

Other distance-based metrics Besides Euclidian and similarity-based metrics, other distance-based metrics could be employed. For example, an approach would be to consider the initial state of the system as not exhibiting interaction or emergent behavior. Using this as a reference, a distance could be calculated from $S(t)$ to $S(0)$, $d(S(t),S(0))$, using similarity metrics that compare between all agent attributes. States in L_ξ are those for which $d(S(t),S(0))$ has a value greater than a system-specific threshold. This method is not system-specific, permitting it to be applied to a variety of systems. However, it is computationally expensive as at each time step, distances have to be computed between all agent attributes. Moreover, for the method to be relevant, only specific agent attributes should be considered, as only some attributes (at the micro level) are relevant for emergent behavior. For example, the number of kilometres flown by a bird will not be relevant when considering the emergence of flocking. The specification of such relevant attributes would thus require in-depth knowledge obtained from the system expert.

Euclidian distance metrics could also be used in clustering algorithms that identify all agents that are close to each other, and thus interacting. However, clustering algorithms are computationally expensive to execute and do not improve the accuracy of the degree of interaction calculation as presented above. Their focus is also different, in that they aim to discover which agents are interacting, rather than if the interaction is significant enough to signal the possibility of emergent behavior.

3.2.2 Interaction Count

Chan proposes to verify that it is indeed the interaction between agents in an agent-based model that causes emergence (Chan 2011). In this approach, interaction is defined as an agent-specific counter that increases as the agent interacts directly with other agents in the environment. Emergence is said to occur if the interaction measure deviates from what is deemed as normal interaction. We extend this approach for the agent-based system defined above. We separate agent behavior rules into *neighboring rules*, which define how an agent interacts with other agents, and *individual rules*, which govern the agent behavior when the agent acts individually without any interaction. For example, a bird in a flock of birds changes its position because of the neighboring rule of coordination (collision avoidance, alignment, and cohesion) with other neighboring birds. Following an individual rule, a bird may not change its speed if is alone in the environment. For a system state $S(t)$ and an agent A_{ij}, we define an interaction count $I(A_{ij}(t))$ as

$I(A_{ij}(t)) = 1$ if any of the neighboring rules for A_{ij} is fired, and $I(A_{ij}(t)) = 0$ otherwise. For the system state $S(t)$, the total interaction count $TI(S(t))$ is defined as the sum of all $I(A_{ij}(t))$.

After the simulation run completes, we can normalize the interaction count of a state $S(t)$ as:

$$\mathbb{D}'(S(t)) = TI'(S(t)) = \frac{TI(S(t)) - TI_{min}}{TI_{max} - TI_{min}} \tag{9}$$

where TI_{min} and TI_{max} are the minimum and maximum interaction counts for the entire simulation.

Calculating states in L_ξ Similar to the above, considering the normalized degree of interaction $\mathbb{D}'$ as the normalized interaction count, we calculate $\mathbb{D}'(S(t))$ according to Equation 9. From all calculated $\mathbb{D}'(S(t))$, we add the maximum value to L_ξ.

3.3 Relative Methods: Interaction Graphs

Relative methods calculate a measure of interaction and establish that emergence exists if it is comparable with that of systems that have been previously shown to exhibit emergence. We detail a relative method that computes interaction graphs from system snapshots and compares them with interaction graphs from systems known to have shown emergent behavior. Snapshots of the simulation run are taken periodically, and an interaction graph is computed for each snapshot. This interaction graph is then compared with interaction graphs of systems that have been shown to exhibit emergence, using two distance metrics (Birdsey and Szabo 2014). The interaction graph (IG) captures the interactions between agents over a given interval of time T^s, where s is the size of the interval in time units and remains the same for a simulation run. An IG is a directed acyclic graph where each vertex represents an agent and each weighted edge represents a interaction between two agents. In the following, for simplicity, we present the formal definition of an IG only for a single agent type. For systems with more than one agent the interaction graph is computed in a similar manner. Formally, $IG_{T^s}(GBS) =< Nodes_{T^s}, Edges_{T^s} >, Nodes = \{A_i | A_i \in GBS, i = 1, \ldots, n\}, Edges = \{(A_{ij}, w_{ij}) | A_{ij} \in GBS, w_{ij} \in \mathbb{Z}^+\}$, where the weight w_{ij} of the edge between A_i and A_j is incremented with each interaction between A_i and A_j.

Snapshots consist of information about agents, the environment, and instances of the specified metric formalism over the time interval T^s. Formally:

$$\mathbb{S}_{T_k}(M) = \{A_i, IG_{T_k}, \ldots | A_i \in GBS\}, \mathbb{S}_{T_k}(GBS) = \{\mathbb{S}_{T_k} | T_k \in T\}$$

where $\mathbb{S}_{T_k}(GBS)$ defines the snapshot for the time interval T_k and $\mathbb{S}_{T_k}(M)$ is the set of all collected snapshots.

Two distance metrics relying on Hausdorff distances have been proposed. The Hausdorff distance (HD) is a metric that is used to determine how much two graphs resemble each other (Huttenlocher et al. 1993). For two interaction graphs $IG(A)$ and $IG(B)$, the Hausdorff distance can be defined as:

$$HD(A,B) = \max\{h(A,B), h(B,A)\}$$

$$h(A,B) = \max_{a \in A}\{\min_{b \in B}\{d(a,b)\}\}$$

and d is the distance between vertices a and b, $a \in A$ and $b \in B$ respectively. For points in a two-dimensional Euclidian space, the distance d could be calculated as the Euclidian distance between points $a(x_a, y_a)$ and $b(x_b, y_b)$ as $d(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$, where (x_a, y_a) and (x_b, y_b) are the cartesian coordinates of points a and b respectively. Intuitively, the Hausdorff distance between A and B measures how close A and B are to each other, where "close" means that every vertex in A is close to some vertex in B.

Using the Euclidian distance d, the calculation of HD helps to determine how close two IGs are with respect to the layout of their vertices. For the flocks of birds model, determining whether a set of birds has achieved a flocking state can be transformed into a shape matching problem using HD as a measure of similarity. However, the Hausdorff distance using the Euclidian distance focuses on the *position* of the

agents and considers interacting agents only from the perspective of their inclusion in the IG, focusing on nodes and not edges and thus not considering the strength of the interaction. Moreover, as the coordinate information is recorded at the end of the interaction interval, the distance function ignores cases in which the emergent behavior happens in the middle of the snapshot interval, and makes the metric dependent on the size of the snapshot interval. An *Active Hausdorff Distance*, *HDA*, can be calculated in a similar manner as the *HD*, but following a pre-processing step: $HDA(A,B) = HD(A',B)$, where A' is obtained from A using a pre-processing algorithm to address these issues.

The pre-processing algorithm aims to move agents closer in the two-dimensional space to the agents with which they had been interacting the most. This ensures that an Euclidian distance considers the interaction strength, but also allows the inclusion of non-distance based measures, as nodes can be moved closer by using any definition of interaction. The algorithm sorts edges in the *IG A* in descending order of their weights. Agents are then moved towards other agents by considering the inverse strength of their interaction. This is done by looking at the agents' positions and determining the correct direction in which the move should take place. *HDA* can be modified further to penalise interactions that actively discourage emergent behavior, for example in the flock of birds model, if a large proportion of the birds were to use the separation interaction over a given interval, it would discourage emergent behavior from happening.

Calculating states in L_ξ For the interaction graph $IG(t)$ for the current time step t, if the distance from a similarity graph $IG_e(t)$ is smaller than a pre-defined threshold, $HDA(IG(t), IG_e(t)) < \varepsilon$, then the system state is added to L_ξ.

4 Experimental Analysis

We compare the three methods described above using a multi-agent model of a flock of birds (boids) model, which captures the motion of bird flocking and is a seminal example for studying emergence (Reynolds 1987). At the macro level, a group of birds tends to form a flock, which has aerodynamic advantages, obstacle avoidance and predator protection, regardless of the initial positions of the birds. At the micro level, each bird obeys three simple rules, namely, (i) separation - steer to avoid crowding neighbors; (ii) alignment - steer towards average heading of neighbors; and (iii) cohesion - steer towards average position of neighbors.

4.1 Absolute Methods

Previous experiments using our extension of the grammar-based approach only permitted us to identify emergent property states for models with up to ten birds. Even for such a small model, the number of states in L_{sum}, computed using the superimposition operator, exceeded ten million. This is no longer the case when using interaction as a handle to identify states in L_{whole} that are part of L_ξ, the set of emergent property states. All of the methods discussed above permit the modeling and analysis of models with a large number of birds that are flocking in environments modeled by a grid of various sizes. We have analyzed models with up to 4,096 birds in a 128 x 128 grid environment. We execute our simulator on a 2.4GHz machine with 3GB RAM and present the worst result for each method.

We analyze the two absolute interaction metrics methods for calculating the degree of interaction for a system with up to 64 birds in a 16 x 16 grid and present the results in Table 1. The results show that, when considering a degree of interaction $\mathbb{D} > 0.4$, the calculation methods achieve a significant reduction in the size of L_{whole} in terms of the number of states when compared to our previous efforts discussed above. The analysis of individual flocking states (not presented here due to space constraints) shows that all states in L_ξ show a high degree of flocking that is proportional with the values of $\mathbb{D}$.

We further analyze the scalability and feasibility of the two absolute methods when analyzing very large models. Towards this, we increase the grid size while keeping the density of birds constant at $\rho = 0.25$ birds/cells. We present the results of these experiments in Table 2.

# birds	size of L_ξ	Approach	size of L_ξ for values of $\mathbb{D}$ in				
			[0.0-0.2)	[0.2-0.4)	[0.4-0.6)	[0.6-0.8)	[0.8-0.1]
16	47	Distance	3	14	14	9	7
		Counting	7	10	13	7	10
32	73	Distance	4	14	25	21	9
		Counting	3	12	32	16	10
64	249	Distance	9	37	91	76	36
		Counting	9	36	78	77	49

Table 1: Number of States in L_ξ on a 16 x 16 Grid for $\mathbb{D}$ in Different Intervals $\in [0,1]$

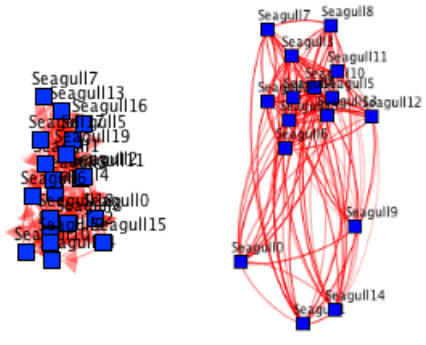
# birds	Grid size	size of L_ξ	Approach	size of L_ξ for values of $\mathbb{D}$ in				
				[0.0-0.2)	[0.2-0.4)	[0.4-0.6)	[0.6-0.8)	[0.8-0.1]
64	16x16	249	Distance	9	37	91	76	36
			Counting	9	36	78	77	49
256	32x32	587	Distance	11	94	229	200	53
			Counting	2	23	144	309	109
1024	64x64	1885	Distance	45	168	682	789	201
			Counting	2	78	95	1230	475
4096	128x128	4868	Distance	147	885	1957	1437	442
			Counting	1	97	497	1690	2583

Table 2: Number of States in L_ξ for $\mathbb{D}$ in Different Intervals $\in [0,1]$

As shown in Tables 1 and 2, the degree of interaction methods are capable of significantly reducing the number of states in L_{whole}. Both methods tend to reduce the number of states in L_ξ as $\mathbb{D}$ increases, supporting the insight that the number of states with very high interaction is low. However, an important point of future work is the analysis of the precision and recall of these methods, for various values of $\mathbb{D}$, and of understanding the relationship between specific values of $\mathbb{D}$ and the emergent states in L_ξ.

4.2 Relative Methods

To showcase the relative method focused on interaction graphs, we present in Figure 1 a flocking in a boids model with 16 birds (B_{16}), for which the HDA distance is the smallest when compared with the interaction graph of a reference model with 20 birds in which flocking has been observed ($IG_e(B_{20})$).



Reference: $IG_e(B_{20})$ Comparison: $IG(B_{16})$

Figure 1: Comparison using Interaction Graphs

Similar to the absolute methods, the relative method using interaction graphs can aid in the computation of L_ξ and can successfully identify flocking in a boids model. It is also successful in identifying flocking as an emergent behavior when comparing between models with different numbers of birds, and also across models with different types of agents. Specifically, it is able to identify emergence when an agent-based model with one type of birds is compared with a model with two types of birds. This shows that the approach is promising and can be applied across domains. We have also been able to identify the emergence of flocking in boids models of 50 to 100 birds by comparing their interaction patterns to those recorded in the boids model with 20 birds, $IG_e(B_{20})$. While semantically very similar, from an automated emergence identification perspective these are very distinct systems, which makes this approach very promising and warrants future investigation. However, work still needs to be done in the comparison and analysis of interaction graphs from different domains, e.g., for comparing flocks of birds and traffic jams.

4.3 Discussion

We also analyze the above methods based on two main quantitative and qualitative criteria, namely, scalability and applicability to different domains. With respect to scalability, we have analyzed the execution time of each of the approaches for large to very large models of up to 1,024 birds and for various grid sizes. Our results show that all approaches are scalable, with runtimes in the hundreds of milliseconds for large models in terms of the number of birds and the size of the grid. Specifically, for a 64 bird model running on a 16 x 16 grid, the computational cost of finding L_ξ is 5, <0s, and 3s respectively for the distance, counting, and interaction graph methods respectively. This is because the counting method has practically no overhead in the calculation of $\mathbb{D}$, as a counter is modified every time the separation rule is called, and this rule is fired as the simulation executes. However, both distance-based and counting methods require the simulation to finish execution before calculating L_ξ, as they rely on normalization. In contrast, the interaction graph method computes similarities between interaction graphs at each snapshot.

With respect to the applicability criteria, we analyze whether the method is applicable to any application domain with little or no modification, and understand the method's limiting assumptions and constraints. In the case of the distance method for the calculation of the degree of interaction, we find that, since the distance calculations assume that some degree of closeness (with respect to geometric distances) will cause interaction and emergence, the method is applicable directly only to systems in which the closeness between individuals will cause emergence. In other systems with indirect interaction, the distance criteria may not be straightforward to define. This is the case also for the counting calculation method for the degree of interaction. Here, indirect interaction is not considered and as such the method may not be applicable to systems where indirect interaction between components is the cause of emergence (as is the case for example in social networks). The interaction graph method currently only focuses on an Euclidian distance for points in the interaction graph and thus might suffer from the same problems as the distance method. Nevertheless, the approach is more generic as it allows for the inclusion of a variety of emergence metrics and is not limited to two-dimensional environments. These metrics could include Shannon entropy and statistical complexity among others, but the definition of a process for including and applying them is still an open problem. Lastly, methods for analyzing the accuracy of various approaches need to be devised.

5 Conclusion

Understanding emergent behavior as it happens is a fundamental and challenging problem in the study of complex systems. A promising approach to formalizing emergent behavior identification proposes a definition of emergence as the set difference between the observed behavior of the system as a whole, and the superimposition of individual agent behaviors. Our previous work has shown that a key issue with using this approach is its unfeasible computational cost, which prevents it and other similar approaches to be applied to medium to large scale models. In this paper, we propose to reduce this computational cost by only considering system states that show high interaction between agents as possible candidates

for emergent behavior. We present and analyze three methods of quantifying interaction, which can be broadly classified into two classes, namely, absolute and relative. Absolute methods compute a degree of interaction that characterizes the system state, whereas relative methods compare between an interaction graph characterizing a system state and interaction graphs from other systems that have been previously shown to exhibit emergent behavior. We implement these methods on a flock of birds model of up to 4,096 birds and discuss their advantages and disadvantages. Our discussion focuses on quantitative metrics such as the number of emergent property states that the methods are able to identify, but also qualitative metrics such as their applicability to other domains and problems. Our findings show that interaction is a promising metric in identifying emergent behavior, but that challenging work remains ahead in further refining the methods and applying them to various domains.

Acknowledgements This work is supported by the National University of Singapore under grant number R-252-000-522-112.

REFERENCES

Bedau, M. 1997. "Weak Emergence". *Philosophical Perspectives* 11:375–399.

Bernon, C., M.-P. Gleizes, S. Peyruqueou, and G. Picard. 2003. "Adelfe: A methodology for adaptive multi-agent systems engineering". In *Engineering Societies in the Agents World III*, 156–169. Springer.

Birdsey, L., and C. Szabo. 2014. "An Architecture for Identifying Emergent Behavior in Multi-Agent Systems". In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 1455–1456.

Chan, W. K. V. 2011. "Interaction Metric of Emergent Behaviors in Agent-based Simulation". In *Proceedings of the Winter Simulation Conference*, edited by S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, 357–368. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chen, C., S. B. Nagl, and C. D. Clack. 2007. "Specifying, Detecting and Analysing Emergent Behaviours in Multi-Level Agent-Based Simulations". In *Proceedings of the Summer Computer Simulation Conference*.

Chi, L. 2009. "Transplating Social Capital to the Online World: Insights from Two Experimental Studies". *Journal of Organizational Computing and Electronic Commerce* 19:214–236.

Cilliers, P. 1998. *Complexity & Postmodernism*. Routledge.

Davis, P. 2005. "New Paradigms and New Challenges". In *Proceedings of the Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Fayyad, U., and R. Uthurusamy. 2002. "Evolving Data Into Mining Solutions for Insights". *Communications of the ACM* 45.

Gardner, M. 1970. *The Fantastic Combinations of John Conway's New Solitaire Games*. Mathematical Games.

Gershenson, C., and N. Fernandez. 2012. "Complexity and Information: Measuring Emergence, Self-organization, and Homeostatis at Multiple Scales". *Complexity*.

Gore, R., and P. Reynolds. 2007. "An Exploration-based Taxonomy for Emergent Behavior Analysis in Simulation". In *Proceedings of the Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1232–1240. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Holland, J. 1999. *Emergence, From Chaos to Order*. Basic Books.

Holland, O. T. 2007. "Taxonomy for the Modeling and Simulation of Emergent Behavior Systems". In *Proceedings of the 2007 Spring Simulation Multiconference*, 28–35.

Huttenlocher, D. P., G. A. Klanderman, and W. J. Rucklidge. 1993. "Comparing Images Using the Hausdorff Distance". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15:850 – 863.

Jacyno, M., S. Bullock, M. Luck, and T. R. Payne. 2009. "Emergent service provisioning and demand estimation through self-organizing agent communities". In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 481–488.

Johnson, C. W. 2006. "What are Emergent Properties and How Do They Affect the Engineering of Complex Systems?". *Reliability Engineering and System Safety* 12:1475–1481.

Kubik, A. 2003. "Towards a Formalization of Emergence". *Journal of Artificial Life* 9:41–65.

Melo, F. S., and M. Veloso. 2009. "Learning of coordination: Exploiting sparse interactions in multiagent systems". In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 773–780.

Mogul, J. C. 2006. "Emergent (mis)behavior vs. Complex Software Systems". In *Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems*, 293–304. New York, USA.

Prokopenko, M., F. Boschetti, and A. J. Ryan. 2009. "An Information-theoretic Primer of Complexity, Self-organization and Emergence". *Complexity* 15:11–28.

Ramakrishnan, K. K., and H. Yang. 1994. "The Ethernet Capture Effect: Analysis and Solution". In *Proceedings of the IEEE Local Computer Networks Conference*. Minneapolis, USA.

Reynolds, C. 1987. "Flocks, Herds, and Schools: A Distributed Behavioral Model". In *Proceedings of ACM SIGGRAPH*, 25–34.

Salazar, N., J. A. Rodriguez-Aguilar, J. L. Arcos, A. Peleteiro, and J. C. Burguillo-Rial. 2011. "Emerging Cooperation on Complex Networks". In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 669–676.

Seth, A. K. 2008. "Measuring Emergence via Nonlinear Granger Causality". In *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, 545–553.

Szabo, C., and Y. Teo. 2012. "An Integrated Approach for the Validation of Emergence in Component-based Simulation Models". In *Proceedings of the Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 2412–2423. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Szabo, C., and Y. M. Teo. 2013. "Post-mortem analysis of emergent behavior in complex simulation models". In *Proceedings of the 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 241–252. ACM.

Teo, Y. M., B. L. Luong, and C. Szabo. 2013. "Formalization of emergence in multi-agent systems". In *Proceedings of the 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 231–240. ACM.

Yaneer, B.-Y. 2004. "A Mathematical Theory of Strong Emergence using Multiscale Variety". *Complexity* 9:15–24.

AUTHOR BIOGRAPHIES

CLAUDIA SZABO is an Lecturer at the School of Computer Science and an Associate Dean (DI) for the Faculty of Engineering, Computer Science and Mathematics at The University of Adelaide in Australia. Her email address is <claudia.szabo@adelaide.edu.au>.

YONG MENG TEO is an Associate Professor with the Department of Computer Science at the National University of Singapore, and a Visiting Professor at the Shanghai Advanced Research Institute, Chinese Academy of Sciences in China.. He heads the Computer Systems Research Group and the Information Technology Unit. His email address is <teoym@comp.nus.edu.sg>.

GAUTAM KRISHNAN CHENGLEPUT is an Honours Student at the Department of Computer Science at the National University of Singapore. His email address is <gautam@nus.edu.sg>.