An Architecture for Identifying Emergent Behavior in Multi-Agent Systems

(Extended Abstract)

Lachlan Birdsey and Claudia Szabo
School of Computer Science
The University of Adelaide
Adelaide, Australia
lachlan.birdsey, claudia.szabo@adelaide.edu.au

ABSTRACT

Multi-agent systems exhibit unexpected, emergent behavior as a result of the complexity of agent behaviors and their interactions. Despite significant research interest in the past decades, computational methods to identify and analyze emergence as it happens are still needed. This paper proposes a software architecture for identifying emergent behavior in a multi-agent system as it happens, using interval-based snapshots and emergent behavior metrics. We propose various distance functions to compare between the multi-agent system under analysis and systems that have been previously shown to exhibit emergent behavior.

Categories and Subject Descriptors

I.6.3 [**Modeling and simulation**]: Agent models

General Terms

Experimentation

Keywords

emergence, interaction, multi-agent systems

1. INTRODUCTION

Complex systems often exhibit behavior that cannot be reduced only to the behavior of their individual components and require thorough analysis once unexpected properties are observed [4]. These *emergent properties* are becoming crucial as systems grow both in size (with respect to the number of components and their behavior and states), but also in coupling and geographic distribution [1, 4]. A plethora of emergent properties examples, from flocks of birds, ant colonies, to the appearance of life and traffic jams have been observed and identified. More malign examples of emergent behavior include power supply variation in smart grids due to provider competition, the Ethernet capture effect in computer networks, and load-balancer failures in a multi-tiered distributed system. Very few methods for the identification, classification, and analysis of emergent behavior exist [2] and they usually assume prior knowledge of emergence. In our work, we propose an architecture to identify and analyze emergent behaviors as they happen [5].

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

2. ARCHITECTURE & METRICS

Our architecture focuses on employing a wide variety of metrics in the analysis of emergent or unexpected system. These metrics may include interaction and statistical complexity among others, but our approach is extensible enough to allow for the addition, removal, or inclusion of various metric combinations in the analysis. We rely on the use of multi-agent system simulation as the fundamental methodology for system analysis, divided into three main steps, namely, (i) *Modelling*, (ii) *Metric Collection*, and (iii) *Analysis and Visualization*. Using an agent-based model of the system under study, simulation metrics are captured in the *Metric Collection* step. The relevant metrics for our analysis are captured at regular simulation intervals by the *metric aggregator*. A meta-model describing these metrics and their aggregation drives this process. For example, when using an interaction metric, the agent interactions are recorded as interaction graphs and specific similarity distances are employed as discussed below. In the *Analysis and Visualization* step, the collected metrics are compared with system-specific threshold or indicator values. Depending on the metric used, the comparisons consider either previous manifestations of behavior that were considered "normal" by system experts, or specific threshold values, or combinations thereof. For example, a system expert can choose interaction as the metric for analysis. In this case, snapshots of agent interactions are captured by the metric aggregator and recorded in the form of interaction graphs.

For a multi-agent system M comprised of n agents a_i, we define an interaction graph (IG) to capture 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, $a \in M$, and each arc represents a interaction between two agents, $a_i \to a_j$, and carries a weight w_{ij}. Formally:

$$IG_{T^s}(M) = < V_{T^s}, E_{T^s} >$$
$$V = \{a_i | a_i \in M, i = 1, \ldots, n\}$$
$$E = \{(a_i, a_j, w_{ij}) | a_i, a_j \in V, w_{ij} \in \mathbb{Z}^+\}$$

The weight w_{ij} of the arc between a_i and a_j is incremented every time an interaction between a_i and a_j happens. Snapshots consist of information about agents, the environment, and instances of the specified metric formalism over the time interval T^s. We aim to make the snapshots metric agnostic to allow for the calculation of various emergent behavior metrics without the need to re-execute the simulation run. Towards this, each snapshot will include, besides the specific formalism values as defined in the meta-model, e.g., interaction graphs, all agent and environment states. We propose the use of Hausdorff distances to calculate the similarity between interaction graphs. The Hausdorff distance (HD) is a metric that is used to determine how much two graphs resemble each other [3]. For interaction graphs $IG(A)$ and $IG(B)$, we define the Hausdorff distance as: $HD(A,B) = \max\{h(A,B), h(B,A)\}$, where $h(A,B) = \max_{a\in A}\{\min_{b\in B}\{d(a,b)\}\}$ and d is the distance between vertices a and b, with $a \in A$ and $b \in B$ respectively. For points $a(x_a, y_a)$ and $b(x_b, y_b)$ in a two-dimensional Euclidian space, the distance d could be calculated as $d(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$.

The Hausdorff Euclidian distance focuses on the *position* of the agents and considers interacting agents only from the perspective of their inclusion in the interaction graph. Since 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 interval. We propose an *Active Hausdorff Distance*, *HDA*, which is 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 that aims to move agents "closer" to agents they have interacted with the most.

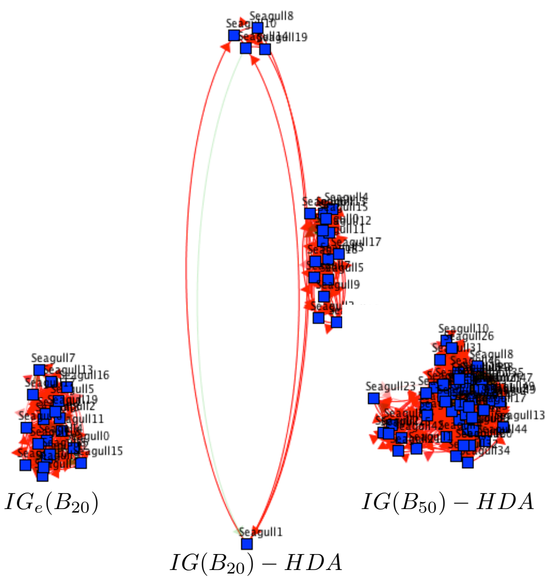

Figure 1: Flocking when compared to the Reference $IG_e(B_{20})$

3. EXPERIMENTS AND CONCLUSIONS

For simplicity, we analyse the boids model, which captures the motion of bird flocking and is a seminal example for studying emergence. We model this system as a multi-agent system in which each bird is an agent with three movement rules defined above. Other bird attributes include initial position and initial velocities. In our experiment, the initial bird positions can be either fixed or assigned randomly at startup. Bird velocities are assigned randomly.

We execute the model in Repast and collect and compute results: at each snapshot interval s we collect and record the snapshot and interaction graph. We compare between the collected interaction graphs and the interaction graphs of systems that have been shown to have emergent behavior and highlight interaction graphs of interest to the user. Our experiments use three different interval sizes, namely, $s = 2$, $s = 5$, and $s = 10$ ticks, over a simulation run of 1,000 ticks. We collect and analyze interaction graphs of boids models with sizes of 20, 50, and 100 birds, with fixed and randomly assigned position values, and randomly assigned velocity values. The results in Tables 1 and 2 are collected over 10 runs and show the values of the normal Hausdorff distance (*HD*) and Active Hausdorff distance (*HDA*) using an Euclidian distance as the distance function. The results show the min, median, mean, standard deviation and runtime on a commodity PC. All interaction graphs are compared with the interaction graph that shows a completely emergent state, in that the birds had flocked (IG_e).

Time interval	Min	Median	Mean	σ	Runtime (ms)
T_{100}	109.77	161.62	165.62	31.34	0.43
T_{500}	110.49	173.93	203.15	84.86	0.28
T_{1000}	154.44	273.62	381.78	211.88	0.31

Table 1: $HD(IG(B_{20}), IG_e(B_{20}))$: 20 birds, $s = 5$

Time interval	Min	Median	Mean	σ	Runtime (ms)
T_{100}	64.21	103.03	99.85	26.60	1.16
T_{500}	81.31	213.69	221.42	134.02	0.74
T_{1000}	3.58	91.06	161.85	212.47	1.41

Table 2: Identifying Emergence: $HDA(IG(B_{20}), IG_e(B_{20}))$: 20 birds, $s = 5$

Figure 1 captures the interaction graph with the minimum *HDA* value for 20 and 50 birds, and the system state in which only a part of the birds are flocking.

Our experiments show that the Active Hausdorff distance has two main advantages over a normal Hausdorff distance. Firstly, in addition to only considering interacting nodes, the Active Hausdorff distance analyzes the interaction strength by including edge weights. This can be further customized to give higher weights to specific interactions, depending on the system under study. Secondly, the Active Hausdorff distance captures the system behavior over the *entire* snapshot interval, and it is thus capable of identifying if the system has been in an emergent state in the middle of the interval and then ceased to be at the end. This makes the interaction metric less dependant on the snapshot interval size. Next, we have been able to identify the emergence of flocking in a boid model of 50 birds by comparing its interaction patterns to those recorded in a boid model of 20 birds. While semantically very similar, from an automated emergence identification perspective these are very distinct systems, which makes our approach very promising and warrants future investigation.

4. REFERENCES

[1] M. Bedau. Weak Emergence. *Philosophical Perspectives*, 11:375–399, 1997.

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

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

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

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