Formalization of Emergence in Multi-agent Systems

Ba Linh Luong¹, Yong Meng Teo¹, Claudia Szabo²

¹ National University of Singapore
² University of Adelaide

21 May 2013

ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS)
Montreal, Canada
Outline

- Motivation
- Objective
- Related work
- Grammar-based Approach
  - Formalization
  - Emergent Property States
  - Example: Boids Model
- Evaluation
- Summary
Motivation

• **Emergence**: *system properties that cannot be derived from the properties of the individual entities*
  – Desirable or undesirable

• **Challenges**
  – Advance understanding of emergence
  – Lack of consensus on emergence

• **Propose Formalization**
  – Set of emergent property states
  – Reason about cause-and-effect of emergence
Objective

A *formal approach* for determining the *set of emergent property states* in a given system.
## Emergence Perspectives

<table>
<thead>
<tr>
<th>Perspective</th>
<th>How</th>
</tr>
</thead>
<tbody>
<tr>
<td>Philosophy [6, 30]</td>
<td>Surprise - Limitations of our knowledge</td>
</tr>
<tr>
<td></td>
<td>Observer with correct scale</td>
</tr>
<tr>
<td>Natural &amp; Social Science [2, 11, 16]</td>
<td>Observer-independent</td>
</tr>
<tr>
<td></td>
<td>Self-organization, hierarchy</td>
</tr>
<tr>
<td>Computer Science [6, 8, 22]</td>
<td>Derived from entity interactions (weak emergence)</td>
</tr>
<tr>
<td></td>
<td>Simulation</td>
</tr>
</tbody>
</table>
Types of Emergence

Emergence

DC: downward causation

No DC

Simple Emergence
Natural & Social Science

Positive or Negative DC

Single Weak Emergence
Computer Science

Multiple Weak Emergence

Positive and Negative DC

Complex DC

Strong Emergence
Philosophy
# Emergence Formalization

<table>
<thead>
<tr>
<th>Approach</th>
<th>Prior Knowledge</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable-based [15, 26, 35]</td>
<td>required</td>
<td>post-mortem</td>
</tr>
<tr>
<td>Event-based [10]</td>
<td>required</td>
<td>post-mortem</td>
</tr>
<tr>
<td>Grammar-based [22]</td>
<td>not required</td>
<td>on-the-fly</td>
</tr>
</tbody>
</table>
Outline

• Motivation

• Objective

• Related work

• Grammar-based Approach
  – Formalization
  – Emergent Property States
  – Example: Boids Model

• Evaluation

• Summary
Grammar-based Approach

• Kubik’s approach: “The whole is greater than the sum of its parts.”

• Main idea: determine the set of system states that are in the whole but not in the sum ($L_\xi$)

$$L_\xi = L_{\text{whole}} \setminus L_{\text{sum}}$$

- $L_{\text{whole}}$: set of all system states obtained by simulation
- $L_{\text{sum}}$: set of all permutations of states of individual parts
Limitations

• Suffers from state-space explosion \((L_{\text{whole}}, L_{\text{sum}})\)
  [ next slide ]

• Cannot model agent types
  [ introduce agent type, \(A_{ij} \) type \(i \leq i \leq m\) ]

• No support for mobile agents
  [ define mobility as attributes of agents:
    \[ P_i = P_{i_{\text{mobile}}} \cup P_{i_{\text{others}}} \] ]

• Closed systems with fixed number of agents
  [ agents can enter and leave system ]
Proposed Approach – Reduce State Space

\[ L_{\xi} = L_{\text{whole}} \setminus L_{\text{sum}} \]

\[ L_{\text{whole}} = L_{\text{whole}}^{I} \cup L_{\text{whole}}^{\text{NI}} \]

\[ L_{\text{sum}} = L_{\text{sum}}^{P} \cup L_{\text{sum}}^{\text{NP}} \]

I: interest

NI: not interest

P: possible

NP: not possible
Proposed Formalization

A system consisting of \( m \) agent types and \( n \) agents \( A_{11}, \ldots, A_{mn_m} \) (\( n = n_1 + \ldots + n_m \), \( n_i \) agents of type \( i \)) interacting in an environment (2D grid) consisting of \( c \) cells is defined as

\[
GBS = (V_A, V_E, A_{11}, \ldots, A_{mn_m}, S(0))
\]

- \( A_{ij} \): agent type \( i \) (\( 1 \leq i \leq m \)) of instance \( j \) (\( 1 \leq j \leq n_i \))
- \( V_A = \bigcup_{i=1}^{m} V_{Ai} \): set of possible agent states for all agent types, and \( V_{Ai} \) denotes the set of possible states for agents of type \( i \)
- \( V_E \): set of possible cell states
- \( V = V_A \cup V_E \)
- \( S(t) \in V^{c+n} \): system state state at time \( t \)
Environment

- **Cell (e)**
  - $V_e$: set of possible states of cell $e$
  - $s_e(t) \in V_e$: state at time $t$

- **Environment (E)**
  - $V_E = \bigcup_{e=1}^{c} V_e$
  - $S_E(t) \in V_E^c$: state at time $t$
Agent

• Agent $A_{ij} \ (1 \leq i \leq m, \ 1 \leq j \leq n)$, is defined as:

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

- $P_i$: set of attributes for agents of type $i$
  
  $$P_i = P_{i\text{ _mobile}} \cup P_{i\text{ _others}}$$
  
  $$P_{i\text{ _mobile}} = \{x | x \text{ is an attribute that models mobility}\}$$

- $R_i$: set of behavior rules for agents of type $i$
  
  $$R_i = R_{i\text{ _mobile}} \cup R_{i\text{ _others}}$$
  
  $$R_i: V_{A_i} \rightarrow V_{A_i} \ // \ V_{A_i}: \text{set of possible states for agents of type } i$$

- $s_{ij}(t) \in V_{A_i}$: state of $A_{ij}$ at time $t$
Emergent Property States

• Set of emergent property states
  \[ L_\xi = L_{\text{whole}} \setminus L_{\text{sum}} \]

• Set of system states with agent coordination (GROUP)
  \[ L_{\text{whole}}^I = \{ w \in V^{c+n} | S(0) \Rightarrow^*_{\text{GROUP}} w \} \]

• Sum of states of individual agents
  \[ L_{\text{sum}} = \text{superimpose}(L(A_{11}), \ldots, L(A_{m_1n_1})) \]
  \[ L_{\text{sum}} \Rightarrow^{\text{constraints}} L_{\text{sum}}^P \]
Example

- Boids model [Reynolds87]
  - Separation (collision avoidance)
  - Alignment
  - Cohesion

- Two types of birds, five ducks and five geese, moving on a 8 x 8 grid
- Maximum speed: ducks (2 cells/step), geese (3 cells/step)
- Birds re-enter the system when they pass the grid edges
## Vector Representation of Velocity

<table>
<thead>
<tr>
<th>Direction</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>North</td>
<td>(0,0)</td>
</tr>
<tr>
<td>North-east</td>
<td>(0,0)</td>
</tr>
<tr>
<td>East</td>
<td>(0,0)</td>
</tr>
<tr>
<td>South-east</td>
<td>(0,0)</td>
</tr>
<tr>
<td>South</td>
<td>(0,0)</td>
</tr>
<tr>
<td>South-west</td>
<td>(0,0)</td>
</tr>
<tr>
<td>West</td>
<td>(0,0)</td>
</tr>
<tr>
<td>North-west</td>
<td>(0,0)</td>
</tr>
</tbody>
</table>

21 May 2013

PADS 2013, Montreal, Canada, 19-22 May

17
Agents – Ducks

• Duck instance $A_{1j}$ ($1 \leq j \leq 5$) is defined as

\[ A_{1j} = (P_1, R_1, s_{1j}(0)) \]

- $P_1 = P_{1\text{\_mobile}} \cup P_{1\text{\_others}}$
  \[ P_{1\text{\_mobile}} = \{ \text{position}(g_{1j}), \text{velocity}(v_{1j}) \}, \quad P_{1\text{\_others}} = \emptyset \]

- $V_{A_1} = \{(x,y) | 1 \leq x \leq 8; 1 \leq y \leq 8\} \times \{(\alpha,\beta) | -2 \leq \alpha \leq 2; -2 \leq \beta \leq 2\}$

- $R_1 = R_{1\text{\_mobile}} \cup R_{1\text{\_others}}$
  \[ R_{1\text{\_mobile}}, R_{1\text{\_others}} = \emptyset \]

- $s_{1j}(t) \in V_{A_1}$

21 May 2013

PADS 2013, Montreal, Canada, 19-22 May
Behavior Rules for Ducks – $R_{1\text{_mobile}}$

- For duck instance $A_{1j}$ ($1 \leq j \leq 5$) at time $t$ with position $g_{1j}(t)$ and velocity $v_{1j}(t)$:
  
  - **Update position:**
    \[ g_{1j}(t+1) = g_{1j}(t) + v_{1j}(t+1) \]

  - **Update velocity:**
    \[ v_{1j}(t+1) = v_{1j}(t) + \text{separation}(A_{1j}) + \text{alignment}(A_{1j}) + \text{cohesion}(A_{1j}) \]

Similarly for Geese!
$S(12) = S(4)$, hence

$L_{\text{whole}}^I = \{S(0), S(1), S(2), S(3), S(4), \ldots, S(11)\}$
Emergent Property States

- $L_\xi = L_{\text{whole}}^I \setminus L_{\text{sum}} = \{S(1), S(2), S(3), S(4), \ldots, S(11)\}$

- **Flocking** - at least 4 birds of the same type fly together
  - *Together* - each bird has at least one immediate neighbor of the same type

1. known emergent states: $S(2), S(3), S(4), \ldots, S(11)$
2. unknown emergent state: $S(1)$
Experimental Results

- Java simulator
- Equal numbers of ducks and geese

<table>
<thead>
<tr>
<th>number of birds</th>
<th>number of states</th>
<th>$L^I_{\text{whole}}$</th>
<th>$L_{\text{sum}}$</th>
<th>$L_\xi$</th>
<th>$\frac{L_\xi}{L^I_{\text{whole}}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>13</td>
<td>767</td>
<td>6</td>
<td>0.46</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>70,118</td>
<td>12</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>509,103</td>
<td>9</td>
<td>0.69</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>26</td>
<td>13,314,006</td>
<td>23</td>
<td>0.88</td>
<td></td>
</tr>
</tbody>
</table>
Summary

• Grammar-based set-theoretic approach
  – Reduce state space
  – Without a priori knowledge of emergence
  – Agents of different types, mobile agents, and open systems

• Example of boids model

• Open issues: reduce state space, reasoning of emergent property states
Q & A

Thank You!

[teoym,luongbal]@comp.nus.edu.sg
claudia.szabo@adelaide.edu.au

Separation Rule

- **Goal:** avoid collision with nearby **birds**

- **How:** if duck \( b \) is close to another bird \( a \), i.e. within \( \varepsilon \) cells, then \( b \) flies away from \( a \)

\[
\text{separation}(b) = \sum_{\text{distance}(a,b) \leq \varepsilon} (b.\text{position} - a.\text{position})
\]

```
separation(boid b)
vector c = 0;
for each boid a
    if |a.position - b.position| \leq \varepsilon then
        c = c - (a.position - b.position)
return c
```
Alignment Rule

• Goal: fly as fast as nearby ducks

• How: change velocity of duck b \( \lambda \% \) towards the average velocity of its neighbor ducks

\[
\text{alignment}(b) = \left( \sum_{\text{duck}(a)}^{k} \frac{a.\text{velocity}}{k} - b.\text{velocity} \right) / \lambda
\]

Alignment(boid b)

vector \( c = 0 \);
integer \( k = 0 \);
for each neighbor duck a
\[
k = k + 1;
\]
\[
c = c + a.\text{velocity};
\]
endfor
\[
c = c / k;
\]
return \( (c - b.\text{velocity}) / \lambda \)
Cohesion Rule

- **Goal:** stay close to nearby **ducks**
- **How:** move **ducks** b $\gamma\%$ towards the center of its neighbor **ducks**

\[
\text{cohesion}(b) = \left( \sum_{\text{neighbor}(a)} \frac{a.\text{position}}{k} - b.\text{position} \right) / \gamma
\]

```c
Cohesion(boid b)
    vector c = 0;
    integer k = 0;
    for each neighbor duck a
        k = k + 1;
        c = c + a.\text{position};
    endfor
    c = c / k;
    return (c - b.\text{position}) / \gamma
```
\[ L_{\text{sum}} \]

- \[ L_{\text{sum}} = \text{superimpose}(L(A_{11}), \ldots, L(A_{15}), L(A_{21}), \ldots, L(A_{25})) \]

- For illustration, consider two geese:
  \[ L_{\text{sum}} = \text{superimpose}(L(A_{23}), L(A_{25})) \]
  \[ = L(A_{23}) \text{ superimpose } (L(A_{25})) \cup L(A_{25}) \text{ superimpose } (L(A_{23})) \]
\( L(A_{23}) \) and \( L(A_{25}) \)

\[
L(A_{23}) = \begin{cases} 
  \text{fly east with speed of 1 cell} \\
  3, (1,0) 
\end{cases}
\]

\[
L(A_{25}) = \begin{cases} 
  \text{fly north-east with speed of 1 cell} \\
  5, (1,1) 
\end{cases}
\]

\( t = 0 \) \hspace{2cm} t = 1 \hspace{2cm} t = 7 \
\[ L_{\text{sum}} = \text{superimpose}(L(A_{23}), L(A_{25})) \]