UML for ABM

Hugues Bersini
IRIDIA/CODE
Plan

• Why modelling socio-economical complex systems?
• How to model them?
Social simulations – The crowd behaviour

Dirk Helbing
Sociological emergent effect

- Thomas Schelling experiment
- How weak racism creates strong segregation
1: move()

alt 1: assessNeighbours
[neighbours > 2]

1.1.1: setAgent(null)

1.3: setAgent(this)

[else]

1.2: findFreeSite()
UML -> MDA

→ Third generation programming language
• UML could be the language adopted by natural scientists to express their knowledge
• Could help to homogenize and cross-fertilize existing models.
• We need to stop with “write once run only once !!!”
• Could improve their own understanding of their own field.
• Could disambiguate some of their knowledge
• Just three diagrams need to be understood
Introduction to class, sequence and state diagrams

1) class TrafficLight { private ArrayList<Car> carsInFront; public TrafficLight () { carsInFront = new ArrayList<Car>();} public void addCar(Car c) {carsInFront.add(c);}} –
2) class Car{ private Engine myEngine; public Car() {myEngine = new Engine();}} –
3) class Engine{}
4) class CityCar extends Car{}
5) class SportCar extends Car{}. 
```java
public Collection<Instrument> getInstrument()
{
    return instrument;
}

/**
 * Returns an iterator over the elements in this collection.
 * @return an <tt>Iterator</tt> over the elements in this collection
 * @see java.util.Collection#iterator()
 */
```
The antigen-specific activation of these effector T cells is aided by co-receptors on the T-cell surface that distinguish between the two classes of MHC molecule; cytotoxic cells express the CD8 co-receptor, which binds MHC class I molecules, whereas MHC Class II molecules specific T cells express the CD4 co-receptor, which has specificity for MHC Class II molecules.
T cells are activated to produce armed effector T cells when their encounter their specific antigen in the form of a peptide: MHC complex on the surface of an activated antigen-presenting cell (APC) ... The most important APC are the highly specialized dendritic cells ... Macrophages can also be activated to express co-stimulatory and MHC class II molecules ... B cells can also serve as APC in some circumstances...
Dendritic cells, macrophages and B cells are often known as professional antigen presenting cells
Sequence Diagrams
public class O1 {
    private int attribute1;
    private O2 lienO2;
    private O3 lienO3;
    public void jeTravaillePourO1(int a) {
        while (i<100) {
            if (j > 20) {
                lienO2.jeTravaillePourO2();
            } else {
                lienO3.jeTravaillePourO3(a);
            }
        }
        if (j < 50) {
            lienO4.jeTravaillePourO4();
        }
    }
}

class O2 {
    private O3 lienO3;
    public void jeTravaillePourO2() {
        lienO3.jeTravaillePourO3(6);
    }
}
State-transition diagram

David Harel
Creator of State Diagrams and IBM Rhapsody

Generated by UModel
www.altova.com
From the diagram to XMI

<subvertex xmi:type="uml:State" xmi:id="U2d38031e-88c0-4487-a358-10033319382f" xmi:uuid="2d38031e-88c0-4487-a358-10033319382f" name="running">
  <region xmi:type="uml:Region" xmi:id="U9975393a-c217-402a-8c28-b4ad044f4921" xmi:uuid="9975393a-c217-402a-8c28-b4ad044f4921" name="Region1">
    <subvertex xmi:type="uml:State" xmi:id="U0d02c7cb-1cc5-4cdb-9951-f98db987fa6b" xmi:uuid="0d02c7cb-1cc5-4cdb-9951-f98db987fa6b" name="yellow">
      <outgoing xmi:idref="U255848fd-71c6-4c37-9320-49f74f21bf91"/>
      <incoming xmi:idref="U89943d6b-b6dd-4e18-aa2d-1998dd625af9"/>
    </subvertex>
  </region>
</subvertex>

<subvertex xmi:type="uml:State" xmi:id="U95ab8769-8478-41a6-b00f-51caaa61d343b" xmi:uuid="95ab8769-8478-41a6-b00f-51caaa61d343b" name="green">
Automatic code generation: the state pattern
class TrafficLight {
    private ElementaryState currentState;
    private CompositeState currentCompositeState;
    private RunningState runningState;
    private FlashingState flashingState;
    public TrafficLight (ElementaryState currentState){
        carsInFront = new ArrayList<Car>();
        this.currentState = currentState;
        runningState = new RunningState(this);
        flashingState = new FlashingState(this);
    }
    public ElementaryState getYellowState() {
        return runningState.getYellowState();
    }
    public void changeState(ElementaryState newState){
        currentState.exitState();
        currentState = newState;
        currentState.enterState();
    }
    public void simulate() {
        while (true) {
            currentState.leaveState();
        }
    }
}
abstract class State {
    private TrafficLight theTrafficLight;
    public State (TrafficLight theTrafficLight){
        this.theTrafficLight = theTrafficLight;
    }
    public TrafficLight getTheTrafficLight(){
        return theTrafficLight;
    }
    public abstract void enterState();
    public abstract void exitState();
    public abstract void leaveState();
}

class GreenState extends ElementaryState {
    public GreenState(TrafficLight theTrafficLight){
        super(theTrafficLight);
    }
    public void enterState() {}
    public void exitState() {}
    public void leaveState() {
        if (event1){
            greenToYellow();
        }
    }
    public void greenToYellow (){
        getTheTrafficLight().changeState(getTheTrafficLight().getYellowState());
    }
}
Design Patterns: « precise » your reality

• The “prototype” DP: how a new object is created by cloning an existing one. Shallow copying (a T cell would be cloned without equally cloning the antigen receptor it is composed of) is different from deep copying (where the cloning of the container implies the cloning of the content).

• The “flyweight” DP: looking for common parts in the description of many objects. Forces the programmer and the immunologist to have a clear idea of what is unique to each clone and what is common to all of them (for instance, if the genetic sequence of their receptor is unique, it can be stored only once in the original lymphocyte and make all clones refer to it).

• The “singleton” DP: Class possessing a single object
Design Patterns (2)

The “bridge DP”

The “Decorator DP”

The “State DP”
Evolutionary Game Theory
The prisoner's dilemma

<table>
<thead>
<tr>
<th>P1/P2</th>
<th>Cooperate</th>
<th>Compete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>(1,1)</td>
<td>(-2,3)</td>
</tr>
<tr>
<td>Compete</td>
<td>(3,-2)</td>
<td>(-1,-1)</td>
</tr>
</tbody>
</table>

The winning strategy for both players is to compete. But doing so, they miss the cooperating one which is collectively better. The common good is subverted by individual rationality and self-interest.
But is competitive behaviour and collective distress avoidable?

- So far the prisoner's dilemma is lacking some crucial quality that real world situations have.
- 1) Iterated version: play several moves and cumulate your reward over these moves.
  - Cooperation can set in → Tit for Tat
- 2) Distribute spatially the players (CA): each cell just cooperates with its immediate neighbours and adapts the local best strategy. Cluster of nice individuals emerge and can prosper in hostile environments → EVOLUTIONARY GAME THEORY
The spatial cellular automata simulation

• Largely inspired by Nowak’s work on spatial prisoner dilemma
• A cellular automata in which every cell contains one agent (specialist or generalist)
• In all cells, asynchronously, an agent will subsequently:
  – interact with its neighbors (Moore neighborhood) to “consume” them.
    • Sum the payoff according to the payoff matrix
  – replicate
    • Adopt the identity of the fittest neighbor
• For a given number of iteration steps
Evolutionary strategy on networks

- Stochastic replicator dynamics:
  - Vertex $x$ plays $k_x$ times per generation and accumulates payoff $f_x$.
  - Choose a random neighbor $y$ with payoff $f_y$.
  - Replace strategy $m_x$ by $m_y$ with probability:

$$p = \max \left[ 0, \frac{f_x - f_y}{k_x(T - S)} \right]$$
Games on graphs

- Conclusions:
  - The more heterogeneous, the more cooperative.
  - Cs benefit most from heterogeneity.
The efficiency/equality trade-off in economy
If both equality and efficiency are valued, and neither takes absolute priority over the other, then, in places where they conflict, compromises ought to be struck. In such cases, some equality will be sacrificed for the sake of efficiency and some efficiency for the sake of equality. But any sacrifice of either has to be justified as a necessary means of obtaining more of the other.
Free market: the perfect concurrence

Buyers compete by raising up the buying price and sellers compete by decreasing the selling offer. Buyers are happy of the sellers competition and vice versa. But is that moral ????
Stylised simulation
Agent’s behaviour
2) Agent = welfare + money + tastes + skills

Tastes and skills are random vectors of products
• The consumer consumes what he has bought. His welfare increases.
• The producer produces either randomly or as a function of the current market. Price feedback effect on the production. The resources decrease. His money decreases.
• The seller makes a selling offer composed of the products just produced: price = fabrication price + benefit (either random in a distributive economy or invertely related to his richness in a competitive one).
• The buyer makes a buying offer. He selects a product out of the blackboard (closest to his tastes). His price is the selling price + a bid (either random in a distributive economy or related to his richness in a competitive one).
3) Offer / either selling or buying
Distributive vs competitive
Distributive:

Random choice of the transaction:
Selling offer 3 and Buying offer 4
Competitive:

<table>
<thead>
<tr>
<th>Selling Offer</th>
<th>Agent</th>
<th>Product</th>
<th>Quantity</th>
<th>Price</th>
<th>UnitPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>P1</td>
<td>100</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>P2</td>
<td>50</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>P3</td>
<td>50</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>P1</td>
<td>25</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>P2</td>
<td>50</td>
<td>100</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Buying Offer</th>
<th>Agent</th>
<th>Product</th>
<th>Quantity</th>
<th>Price</th>
<th>UnitPrice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>P1</td>
<td>50</td>
<td>150</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>D</td>
<td>P2</td>
<td>100</td>
<td>600</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>P1</td>
<td>25</td>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>P3</td>
<td>25</td>
<td>75</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>P3</td>
<td>100</td>
<td>300</td>
<td>3</td>
</tr>
</tbody>
</table>

Fig. 2. The two lists of offers which compose the blackboard, above the selling list below the buying list.

choice of the MAX transaction: SellingOffer 2 and BuyingOffer 2
Preliminary results:

100 agents (the same initial money and welfare)
Random skills and tastes
<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th>80</th>
<th>60</th>
<th>50</th>
<th>25</th>
<th>20</th>
<th>15</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility</td>
<td>5465</td>
<td>5340</td>
<td>5464</td>
<td>5452</td>
<td>5433</td>
<td>5401</td>
<td>5330</td>
<td>5246</td>
</tr>
<tr>
<td>Money</td>
<td>4182</td>
<td>3275</td>
<td>2093</td>
<td>1735</td>
<td>494</td>
<td>284</td>
<td>83</td>
<td>17</td>
</tr>
<tr>
<td>Ad. Val.</td>
<td>3680</td>
<td>3720</td>
<td>3264</td>
<td>3380</td>
<td>2675</td>
<td>2400</td>
<td>1817</td>
<td>1424</td>
</tr>
<tr>
<td>G(Util)</td>
<td>0.05</td>
<td>0.061</td>
<td>0.077</td>
<td>0.070</td>
<td>0.088</td>
<td>0.12</td>
<td>0.16</td>
<td>0.25</td>
</tr>
<tr>
<td>G(Mon)</td>
<td>0.006</td>
<td>0.007</td>
<td>0.013</td>
<td>0.010</td>
<td>0.016</td>
<td>0.017</td>
<td>0.014</td>
<td>0.08</td>
</tr>
<tr>
<td>G(AV)</td>
<td>0.08</td>
<td>0.092</td>
<td>0.11</td>
<td>0.10</td>
<td>0.13</td>
<td>0.15</td>
<td>0.17</td>
<td>0.20</td>
</tr>
<tr>
<td>MF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>26</td>
<td>270</td>
<td>914</td>
</tr>
</tbody>
</table>
Gini index
Indice de Gini
Comparison lottery/concurrence

Lottery
• Low Gini → More equality
• Low aggregate utility → Less efficient

Concurrence
• High Gini → Less equality
• High aggregate utility → More efficient

Fondamental economy dilemma

Solutions: Redistributions (But Okun), regulation, random economy