

# An Introduction to the Repast Software

**Recursive Porous Agent Simulation  
Toolkit**

By Andrew Crooks

# Outline

- What is Repast
- The different types of Repast.
- Repast core features.
  - Space in Repast
- Who is using Repast
- Useful websites and tools.

# What is Repast?

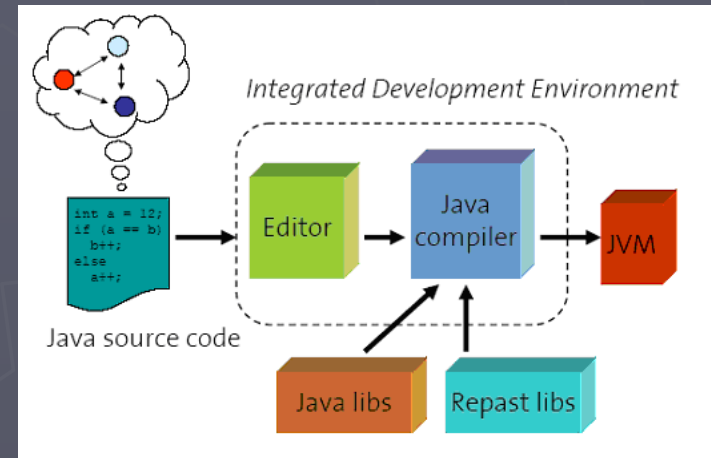
- Free, open source library of classes for creating, running, displaying and collecting data from agent based simulations.
- Fully Object Oriented.
- Created by University of Chicago's Social Science Research Computing division.
- Guaranteed to be maintained for 5 years.
- Most popular simulation toolkit at present.

# What is Repast?

It is a toolkit as it provides:  
A collection of tools & structures that are useful for simulation but don't have to be used (=> flexibility and extendable).

Standardised tools/features:

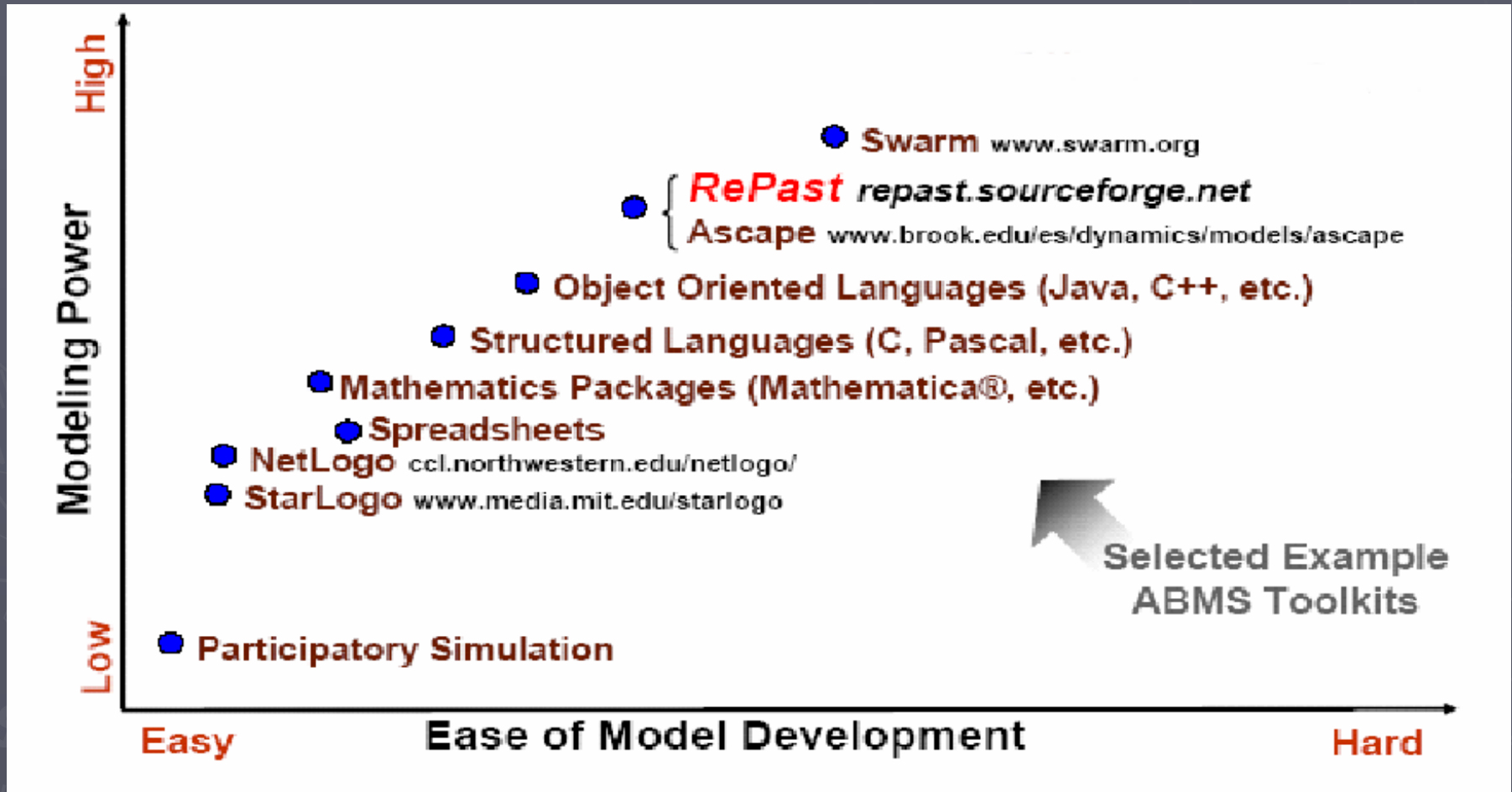
- The Toolbar.
- GUI for manipulating parameters.
- Output data (histogram, time series).
- Displaying agent interaction.



This block contains four screenshots of the Repast GUI, each with a caption below it:

- Controlling simulations:** A screenshot of the 'Enns' toolbar with buttons for play, stop, and other simulation controls. The 'Tick Count: 0.0' is visible.
- Displaying behavior:** A screenshot of the 'Enns Display' window showing a grid of red and blue pixels representing agent interactions.
- Charting:** A screenshot of the 'Agent Wealth Distribution' histogram, showing a distribution of wealth values.
- Managing parameters:** A screenshot of the 'Enns Settings' dialog box, showing various parameters like 'AgentWage', 'DesShuffle', 'GridDensity', etc., with input fields and dropdown menus.

# Selected ABM tools

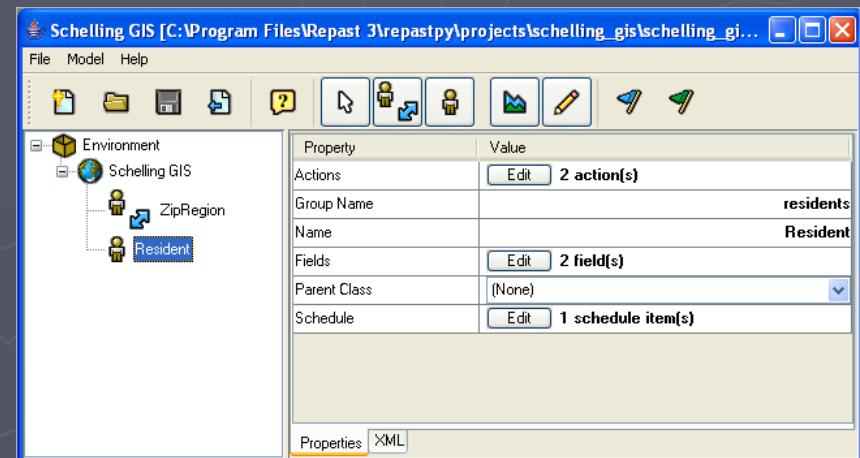
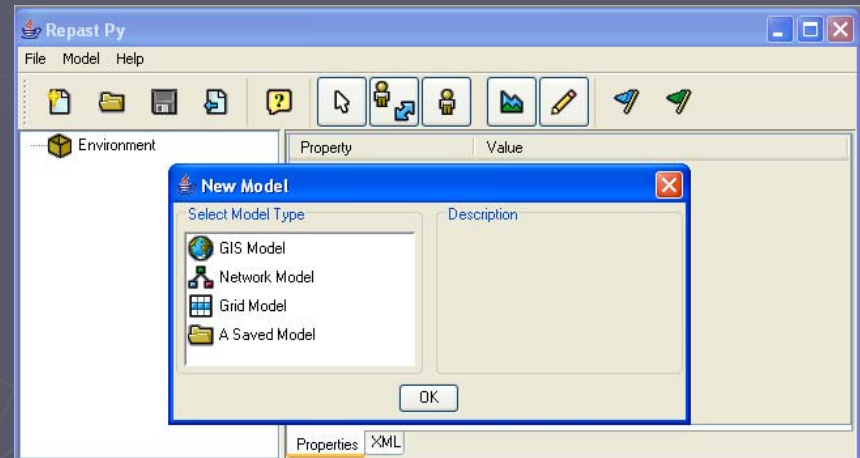


# Different strands of Repast.

- 3 Flavours of Repast:
  - RepastPy
  - RepastJ
  - Repast.Net
- All have the same core services, which allow ABM to be developed on top of.
- They differ on their underlying platform and development languages.
- Basic programming knowledge is needed.

# RepastPy

- Most graphical way to create models via GUI and Python scripting language.
- Designed for rapid development and prototyping of basic models, to learn Repast.
- Basis of Agent Analyst.
- Models Exported to Java if more complex models are desired.

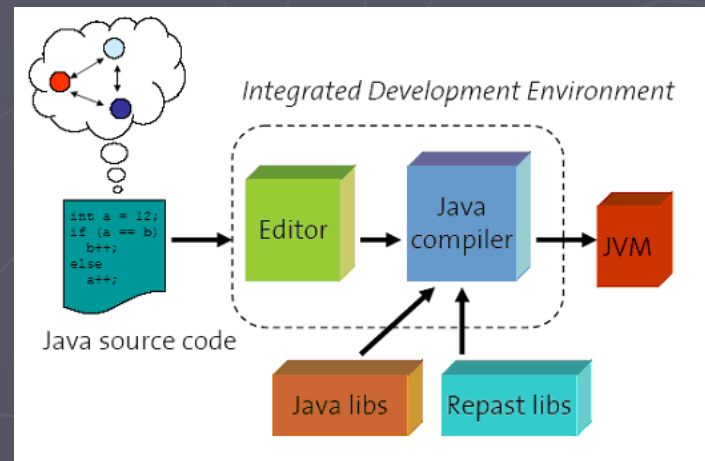


# RepastJ

- Java based (platform (OS) independent).
- Contains an extensive list of core classes (GIS).
- Recommend an IDE (Eclipse) to write, edit, compile and debug.
- Once compiled => run on any computer with JVM.
- Most models are being developed in RepastJ.

```
public class Model extends SimModelImpl {  
    // SECTION AAA FROM DEONIFICATION Template  
    // For FORMAL PRINTS and PRINTS  
    protected Parameters p = new Parameters(this).mark();  
    protected Parameters p = new Parameters();  
  
    // Generic parameters  
    protected String      initialParametersFileName = "";  
    private String        reportFileName = "report";  
    private String        outputDirName = "/";  
    private int           runNumber = 0;  
    private int           script = 100;  
    private int           debug = 0;  
    private int           saveRunEndState = 0;  
}
```

Console Output:  
GUIModel setup() called...  
Model Setup() called...  
GUIModel setup() continued...  
(GUIModel) main done --  
(GUIModel) --> enter GUIModel-begin() --  
seed was 1234567 or 0, now ==> s=1137331736897  
about to setSeed(1137331736897)

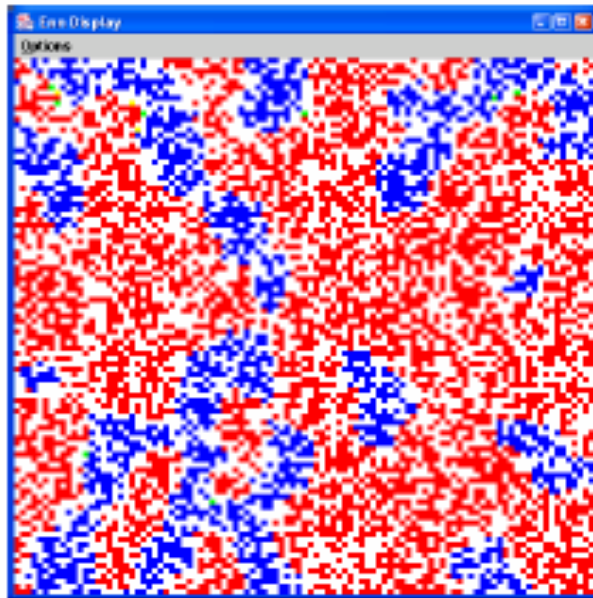




# Repast.Net

- Implemented in C# (Microsoft's answer to Java).
- Simulations can be implemented in any .Net language (VB.Net, C++, J#, C#, etc. ) and integrated to create Windows-based applications.
- No inbuilt GIS functions => need to link (e.g. ESRI MapObjects). Nor platform independent.
- Good if you don't want to know Java but still want to use Repast.

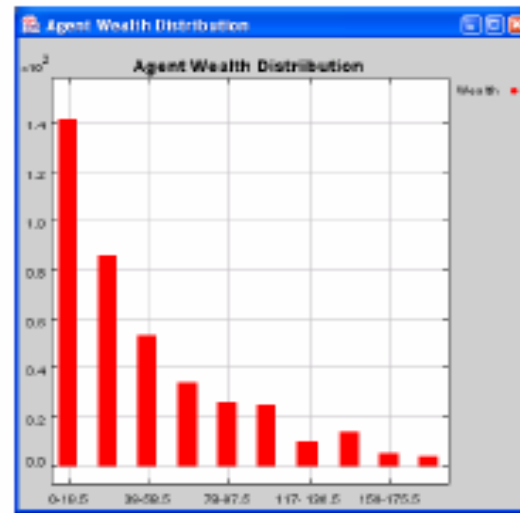
# Repast Framework



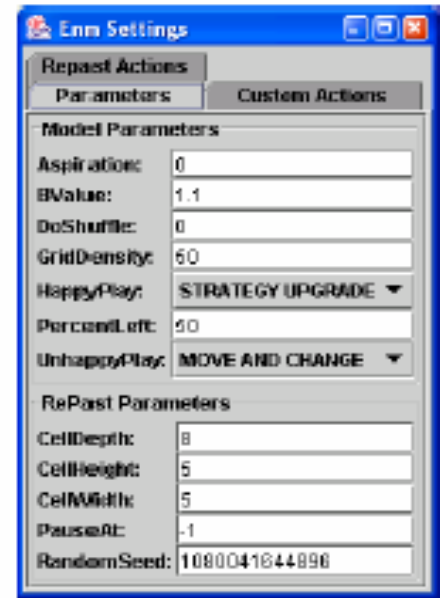
Displaying behavior



Controlling simulations



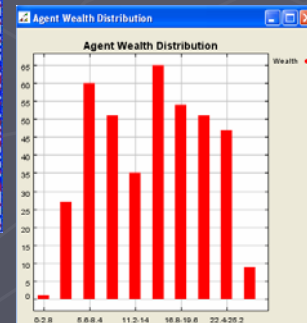
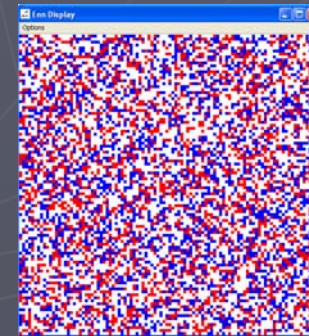
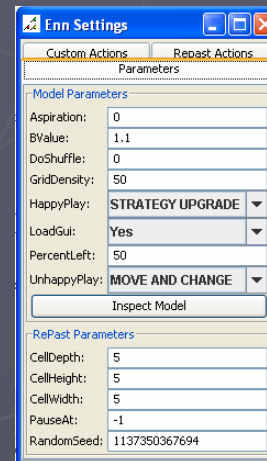
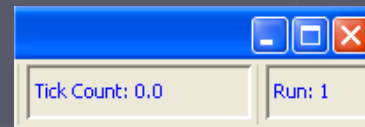
Charting



Managing parameters

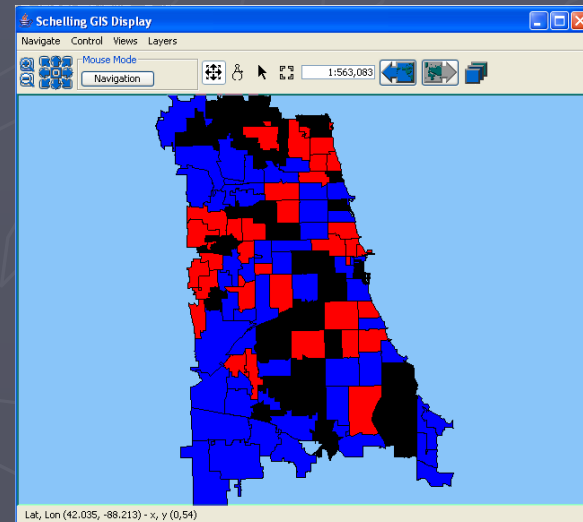
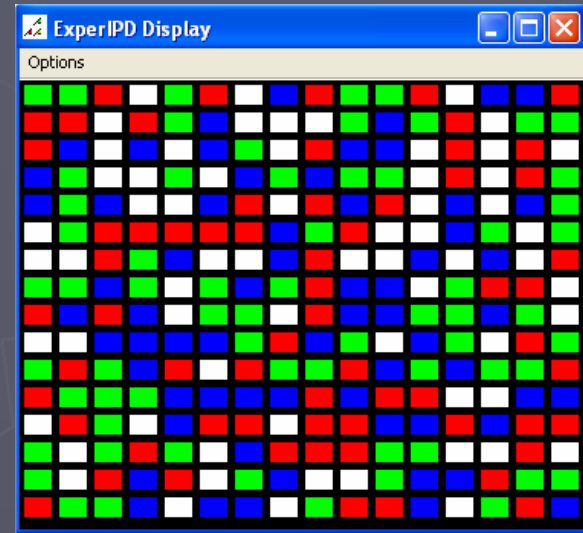
# Repast Framework: What does it offer?

- Skeletons of **agents** and their **environment** (not constrained).
- Graphical **user interface**
- **Scheduling** of simulations
- **Parameters** management
- Behaviour **display**
- **Charting**
- **Data collection**
- **Batch** and **single** runs
- **Utilities** for common tasks.
- **Sample** models (including code) and **documentation**.



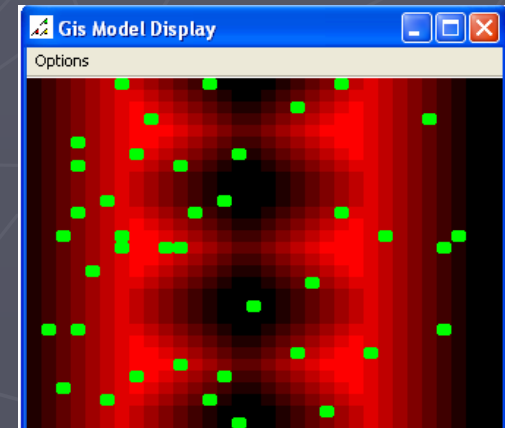
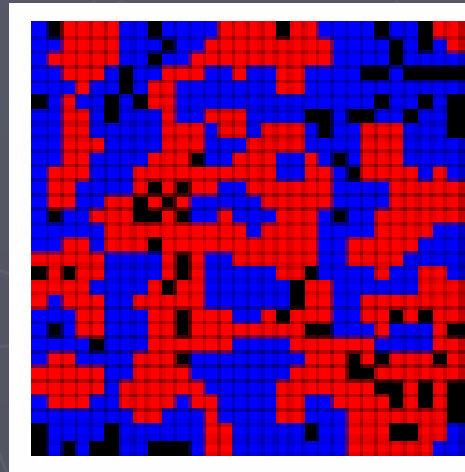
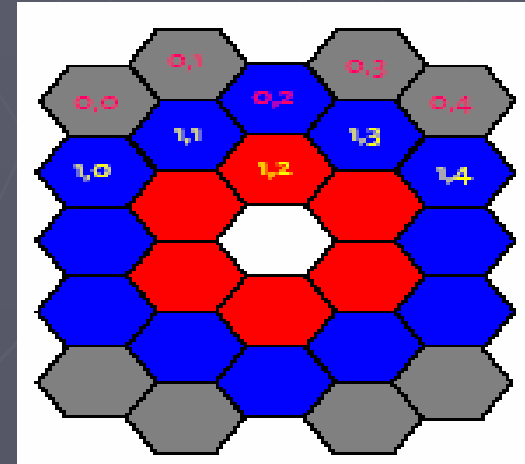
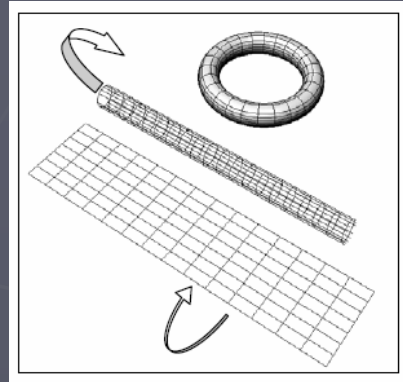
# Space in Repast

- 2 main types:
  - Cellular
  - GIS vector
- Space has Two purposes:
  - Contain a Collection of agents (e.g. the world).
  - Define Spatial relationship of agents relative to each other.
- Another type of space is Network space.



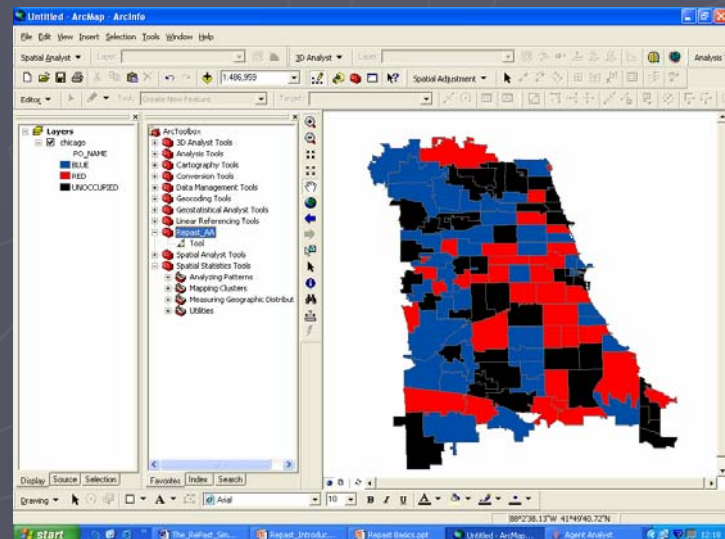
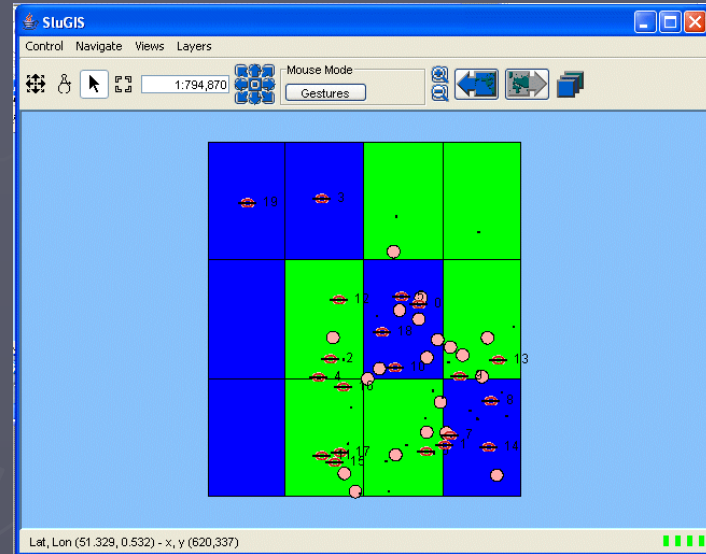
# Cellular Space

- **Boundaries**
  - Grid or Torus
- **Cell's shape**
  - Regular or hexagonal
- **Cell's content**
  - One object or a collection of agents
- Also includes Raster-Space (inbuilt functions for reading ASCII & .pgm files)



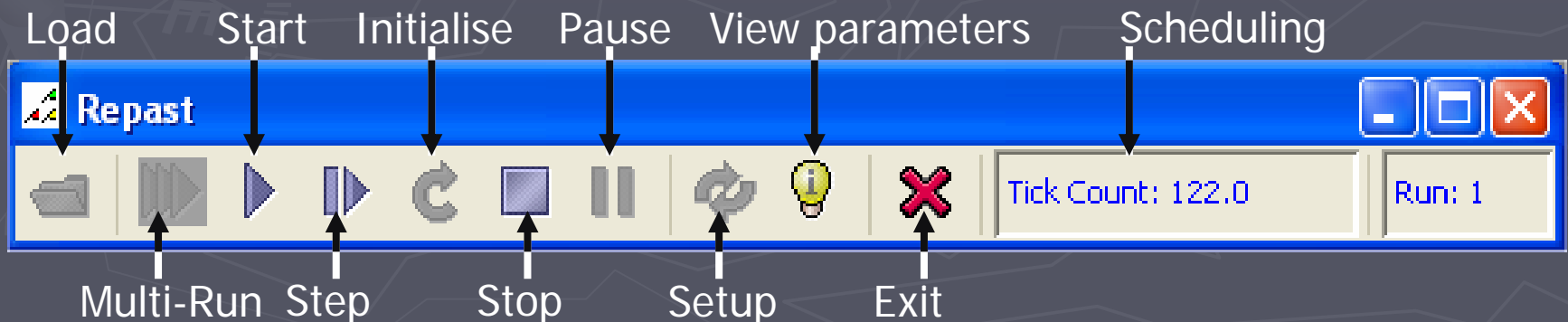
# Vector GIS

- More complicated (own packages) but highly adaptable.
- Agents either
  - Points (Generic agent).
  - Polygons (Vector agent) (static) => detailed boundaries.
- Can be combined.
- Needs its own displays: ESRI or OpenMap.

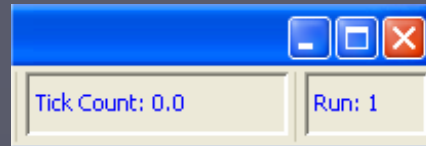


# Inbuilt GUI: Controlling Simulations

- Batch or GUI
  - Different purpose depending on context
    - **Batch mode:** allow variables to be changed through parameter files (.txt) used for multiple runs.
    - **GUI mode:** show the variables in the control panel (Gives greater user interaction).
      - GUI give:
        - Customised parameter panel
        - Dynamic Graphs
        - Graphical Displays
        - Probes



# Inbuilt functions for Controlling Simulations: Scheduling



- Built in Scheduler.
- Set of methods which allow certain actions to be carried out at a specific time and/or specific order e.g. Agent to move 1 cell each time step then update display.
- Scheduled actions can change the agents state.

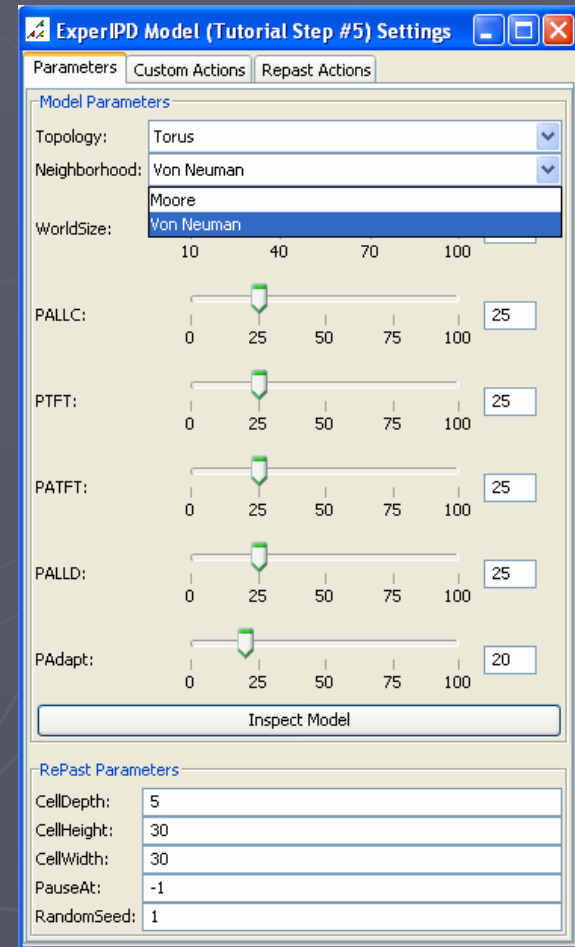
```
public void buildSchedule(){
    ActionGroup group = new ActionGroup(ActionGroup.SEQUENTIAL);
    group.addAction(new PreStep());//e.g.: get landuse & update displays
    group.addAction(new Step());// e.g.: move if unhappy
    group.addAction(new PostStep());// e.g.: add new agents & update displays
    group.addAction(new GISModelUpdateLanduseGraph());

    schedule.scheduleActionBeginning(1, group);
    schedule.scheduleActionAtInterval((double)5, this, "saveShapefile", Schedule.LAST );
    schedule.scheduleActionAtInterval((double)1, this, "createShapefileForAgents", Schedule.LAST );
    schedule.scheduleActionAtPause(recorderLanduse, "writeToFile");
    schedule.scheduleActionAtEnd(this, "saveGALFile");
    schedule.scheduleActionAtEnd(recorderLanduse, "writeToFile");
}
```



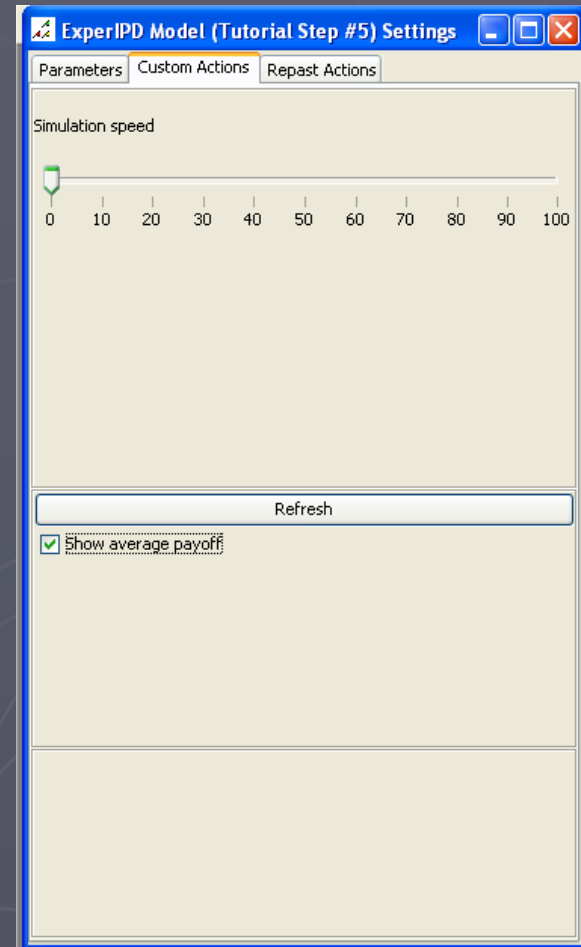
# Controlling Simulations Parameters Management : GUI

- To adjust parameters at the start-up or when the simulation is paused.
- Can be applied to both **models** (e.g. world size) and **agents** (e.g. preferences).
- Setup through predefined descriptors variable of `SimModel`
- Inbuilt functions for dropdown lists, sliders and text



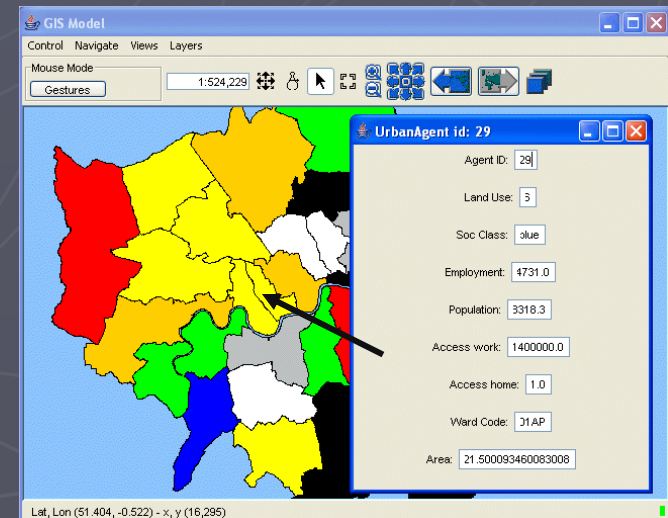
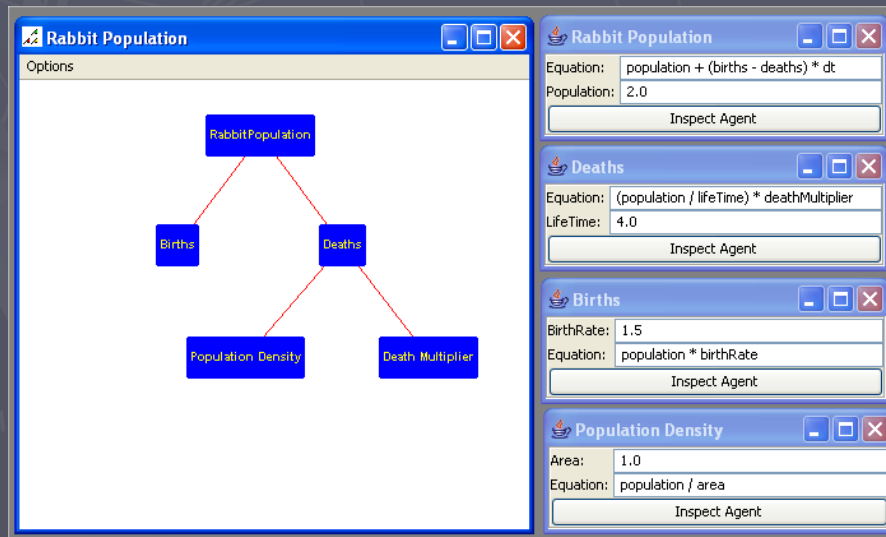
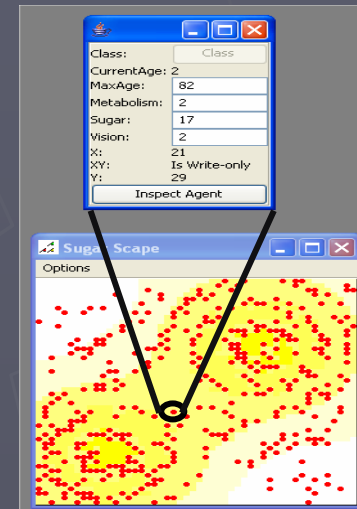
# Controlling Simulations: Custom Actions

- Modify the simulation while it is running.
- Setup through `modelManipulator`.
- Works for buttons, check boxes, sliders.



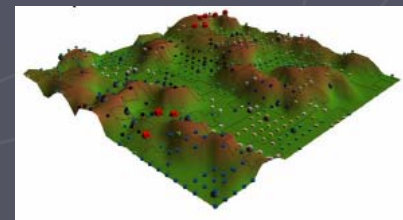
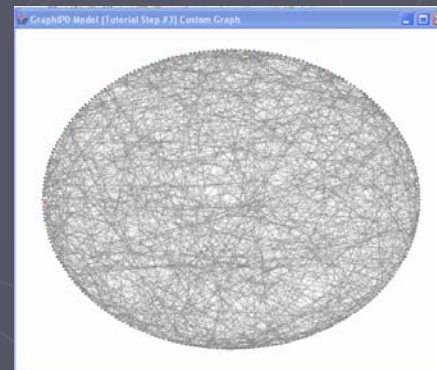
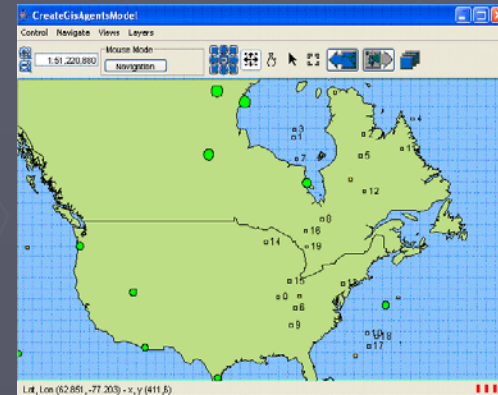
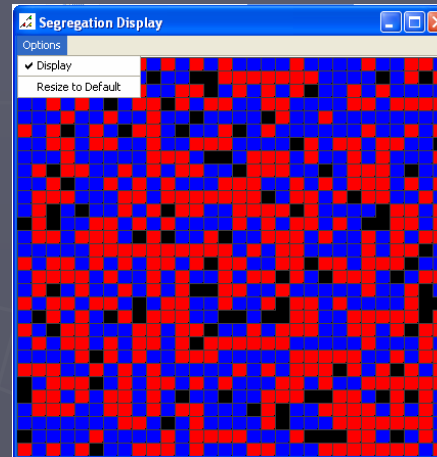
# GUI Parameters: Probes

- Allow the user to view and manipulate single agents.
- Altering simple equations.
- Feeds back into the simulation



# Displaying Behaviour in Repast

- A collection of classes for the Graphical presentations of agents and their environments,
- Can be updated whenever.
- Can be extended (Java3D).
- Don't have to be displays.



```
SegregationGUI.java x
package schelling;

/**
 *
 */

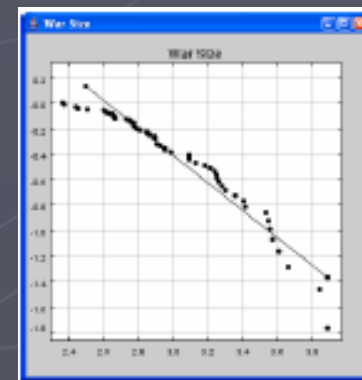
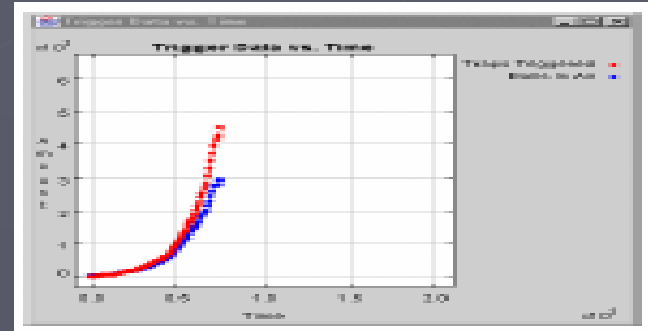
Problems Declaration Search Console
<terminated> Model (4) [Java Application]

Configuration after step 19

|##00#####0000|
|##00 #####0 00#0 00|
|## #####00000#00000|
|##### #00000#0000|
|#####00000#0000|
|## # #0#00### #0000|
|0#####00000#####000|
|00## #00000#####00|
|00 ##0000 #0###0000|
|#####000#000000000|
|##### #0#####0000000|
|## #####0#####0000000|
|#####000000000##00|
|## #000000000## 0|
|## # 00000#### 00|
|## ##00000#####00|
|##### 00000#####00|
|#####0000##### ##|
|## #0000# # # #####|
|## ##000## # #####|
```

# Inbuilt Charting options

- **Time series:** used defined variable (s) over time.
- **Histograms:** bar chart showing a variable distribution.
- **Scatter plot:** snapshot of two variables
- Can be extended e.g. JFreeChart



# Inbuilt Data Collection

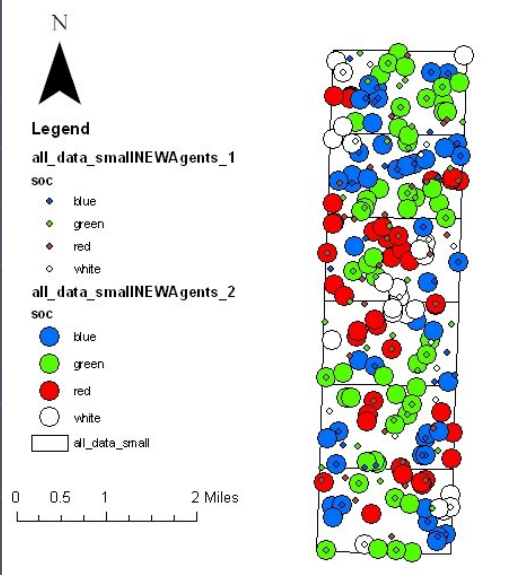
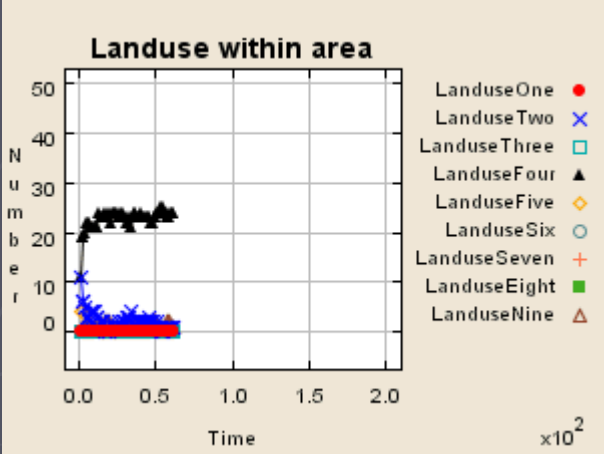
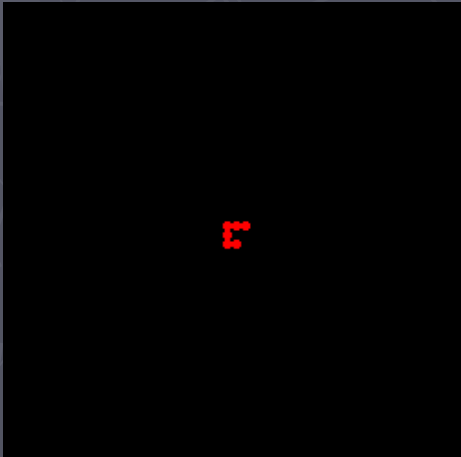
- Inbuilt data recording facilities (Data Recorder) to .txt files (as comma, space or tab delimited).
- Movies and snapshots.

```

all_data_small_landuseStat.txt - Notepad
File Edit Format View Help
Model Landuse uses
Timestamp: 10-Jan-2006 19:00:35
AddAgents: 2
AddNewAgentsOn: true
Blue_blue: 50
Blue_green: 50
Blue_red: 20
Blue_white: 50
Green_blue: 0
Green_green: 50
Green_red: 0
Green_white: 50
MoveToWithin: 200
Movement: 500
PerAgents: 1
Red_blue: 0
Red_green: 0
Red_red: 100
Red_white: 0
RemoveAgentsOn: true
Search: 200
SearchOn: true
SizeOfAgent: 0.0010
White_blue: 50
White_green: 50
White_red: 0
White_white: 50
Time
"11": "Landuse1", "Landuse2", "Landuse3", "Landuse4", "Landuse5", "Landuse6", "Landuse7", "Landuse
1.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,3.0,2.0,1.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
2.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,3.0,3.0,1.0,0.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0

```

Records parameters



# Who is using Repast?

- Designed for social scientists but not confined to it
- Examples include:
  - **Politics:** the rise and fall territories.
  - **Economics:** Banking strategies, evolution of firms, dynamics between firms, electricity markets
  - Segregation, Pedestrian modelling, disease transmission,
  - **Physical environment:** water shed modelling
  - **Landuse models:** SLUCE (change at urban rural fringe), LUCITA (Amazon farming), urbanisation.
  - Extensions also:
    - GeoGraph (3D extensions), AgentCell (biochemical reactions 3D) and parallel processing.

# Useful Java libraries

- **OpenMap**: extend the displays.
- **OpenForecast**: a regression library.
- **Log4J**: useful for debugging code.
- **Java Topology Suite**: classes and methods for 2D spatial analysis functions.
- **JFreeChart**: More sophisticated chart construction than Repast.
- Can use any Java library and choose only the Repast tools you want.



# More information / examples

- **Repast Website:** <http://repast.sourceforge.net/>
- **How to create a RePast model** (step by step instructions on the basics of repast) by John. Murphy. <http://www.u.arizona.edu/~jtmurphy/H2R/HowTo01.htm>
- **International Conflict Research** (repast examples and most linked to papers and theory, lecture slides)  
<http://www.icr.ethz.ch/teaching/archive/comppmodels/ss2004/models>

# Summary

- Why use it?
  - **Inbuilt functions** but highly **customisable**  
=> if a tool does not exist create one.
  - Need to know some programming.
  - General purpose, not constrained by the application.
  - Large user group, documentation, examples.
  - Use of freely available Java libraries.