# Agent-based Simulation プラットフォーム最新動向

2016年9月

筑波大学　倉橋節也

# 目次

- はじめに
- 目的
- 比較対象
- 評価結果
- まとめ

# 目的

- 国際的に使用されているABSプラットフォームは何か
- どのような機能を比較しているか
- 機能比較
- 推奨?

# 国際的に使用されている
# プラットフォーム比較対象

- Agent Factory
- Agent Builder
- AgentScape
- AGLOBE
- AnyLogic
- Cormac
- Cougaar
- CybelePro
- EMERALD
- GAMA
- INGENIAS
- JACK
- JADE
- Jadex
- JAMES II
- JAS
- Jason
- JIAC
- NetLogo
- MASON
- Repast
- Swarm

# 開発元・最新版・オープンソース

| 製品名 | 開発元 | 最新版 | オープンソース |
|---|---|---|---|
| Agent Factory | University College Dublin | 2011 | Yes |
| Agent Builder | Acronymics Inc. | 2004 | No |
| AgentScape | Delft University of Technology | 2013 | Yes |
| AGLOBE | Czech Technical University | 2008 | Yes |
| AnyLogic | The AnyLogic Company | 2014 | No |
| Cormas | Cirad research centre | 2014 | Yes |
| Cougaar | Raytheon BBN Technologies | 2012 | Yes |
| CybelePro | Intelligent Automation Inc | 2013 | No |
| EMERALD | LPIS Group, Aristotle University of Thessaloniki | 2012 | Yes |
| GAMA | IRD/UPMC International Research Unit UMMISCO | 2013 | Yes |
| INGENIAS | grasia! research group, Universidad Complutense Madrid | 2013 | Yes |
| JACK | AOS | 2010 | No |

# 開発元・最新版・オープンソース

| 製品名 | 開発元 | 最新版 | オープンソース |
|---|---|---|---|
| JADE | Telecom Italia (TILAB) | 2013 | Yes |
| Jadex | Hamburg University | 2013 | Yes |
| JAMES II | University of Rostock | 2014 | Yes |
| JAS | Universita` di Torino | 2006 | Yes |
| Jason | Universities of Rio Grande do Sul and Santa Catarina | 2013 | Yes |
| JIAC | Technische Universität Berlin | 2014 | Yes |
| MaDKit | Institut universitaire de technologie | 2014 | Yes |
| MASON | George Mason University | 2013 | Yes |
| NetLogo | Northwestern University | 2013 | No |
| Repast | University of Chicago | 2013 | Yes |
| SeSAm | Örebro University | 2012 | Yes |
| Swarm | Swarm Development Group | 2005 | Yes |

# GUI・難易度・スケーラビリティ

| 製品名 | GUI | 学習難易度 | スケーラビリティ | パフォーマンス |
|---|---|---|---|---|
| Agent Factory | Poor | 平均 | Good | Good |
| Agent Builder | Poor | 簡易 | Good | Good |
| AgentScape | Poor | 平均 | Good | Good |
| AGLOBE | Poor | 簡易 | High | High |
| AnyLogic | Rich | 簡易 | High | High |
| Cormas | Useful | 簡易 | Good | High |
| Cougaar | Unfamiliar | 平均 | High | High |
| CybelePro | Useful | 平均 | High | High |
| EMERALD | Useful | 簡易 | High | High |
| GAMA | Useful | 簡易 | Good | Good |
| INGENIAS | Friendly | 簡易 | Good | Good |
| JACK | Rich | 簡易 | High | High |

# GUI・難易度・スケーラビリティ

| 製品名 | GUI | 学習難易度 | スケーラビリティ | パフォーマンス |
|---|---|---|---|---|
| JADE | Useful | 簡易 | High | High |
| Jadex | Many feature | 平均 | High | High |
| JAMES II | Friendly | 簡易 | High | High |
| JAS | Poor | 簡易 | Average | Good |
| Jason | Friendly | 簡易 | Good | Good |
| JIAC | Useful | 平均 | High | High |
| MaDKit | Rich | 平均 | Good | Good |
| MASON | Useful | 平均 | Average | Good |
| NetLogo | Useful | 簡易 | Good | Good |
| Repast | Limited | 簡易 | Good | High |
| SeSAm | Rich | 簡易 | Good | Good |
| Swarm | Limited | 平均 | Average | Average |

# 言語・OS・サポート・人気・コスト

| 製品名 | 言語 | OS | サポート | 人気 | コスト |
|---|---|---|---|---|---|
| Agent Factory | Java, AFAPL, AgentSpeak | JVM | Good | Low | Free |
| Agent Builder | KQML, Java, C++ | Win, Linux, Sun | Good | Medium | $99~ |
| AgentScape | Java+XML | JVM | Good | Low | Free |
| AGLOBE | Java | JVM | Average | Medium | Free |
| AnyLogic | Java, UML-RT | JVM | High | Medium | $485~ |
| Cormas | SmallTalk | Win, Linux | Good | Medium | Free |
| Cougaar | Java | Win, Linux | Good | Low | Free |
| CybelePro | Java | JVM | Good | Low | $600~ |
| EMERALD | Java, JESS, RuleML, Prolog | JVM | Average | Low | Free |
| GAMA | GAML | Mac, Win, Linux | Good | Low | Free |
| INGENIAS | Java+XML | JVM | Average | Medium | Free |
| JACK | Java, JACK Agent Language | Win, Mac, Unix, g-Java | High | High | Free Trial |

# 言語・OS・サポート・人気・コスト

| 製品名 | 言語 | OS | サポート | 人気 | コスト |
|---|---|---|---|---|---|
| JADE | Java | JVM | High | High | Free |
| Jadex | Java+XML | JVM | Average | High | Free |
| JAMES II | Java | JVM | Average | Medium | Free |
| JAS | Java | JVM | Good | Medium | Free |
| Jason | Java, AgentSpeak | Win,Mac | High | High | Free |
| JIAC | Java+XML | JVM | Good | Low | Free |
| MaDKit | Java, C/C++, Python | JVM | Good | Medium | Free |
| MASON | Java | Win, Mac | Average | Medium | Free |
| NetLogo | NetLogo | JVM | Good | High | Free |
| Repast | Java, C#, C++, Lisp, Prolog, Python, ReLogo | JVM | Average | Medium | Free |
| SeSAm | Java+XML | JVM | Good | Medium | Free |
| Swarm | Java, Object C | Win, Linux | Good | Medium | Free |

# 開発言語

| 言語 | プラットフォーム |
| --- | --- |
| Java | JADE, SeSAm, Jadex, JAS, AgentBuilder, EMERALD, Repast, MaDKit, CybelePro, Cormas,AGLOBE, Couggar, Swarm, MASON, INGENIAS, AnyLogic, JAMES II |
| C/C++ | AgentBuilder, Swarm(Object C/C++), Repast, MaDKit |
| Python | Repast, MaDKit |
| SmallTalk | Cormas |
| NetLogo | NetLogo |
| 宣言型 | Repast(Lisp, Prolog), JADE(JESS), EMERALD(JESS) |
| GAML | GAMA |

# 使用目的

| 目的 | プラットフォーム |
| --- | --- |
| ABS | JAS, AgentBuilder, Aget Factory, Swarm, MASON, INGENIAS, Repast, MaDKit, AgentScape, SeSAm, AnyLogic, NetLogo, GAMA, JAMES II |
| Real World GIS | AGLOBE, Cougaar, Repast, CybelePro, SesAm, AnyLogic, GAMA |
| Large scale | Cougaar, CybelePro, JIAC, AgentScape, GAMA, JAMES II |
| Schedulling & Planning | Jadex, CybelePro, SeSAm, AnyLogic, NetLogo, GAMA |
| Biological & social studies | JADE, SeSAm, Jadex, Jason, EMERALD, JACK, Cougaar, CybelePro, MaDKit, Repastm AnyLogic, NetLogo, GAMA, JAMES II |
| Economics/eCommerce | Jade, EMERALD, JACK, Cougaar, CybelePro, MaDKit, AnyLogic |

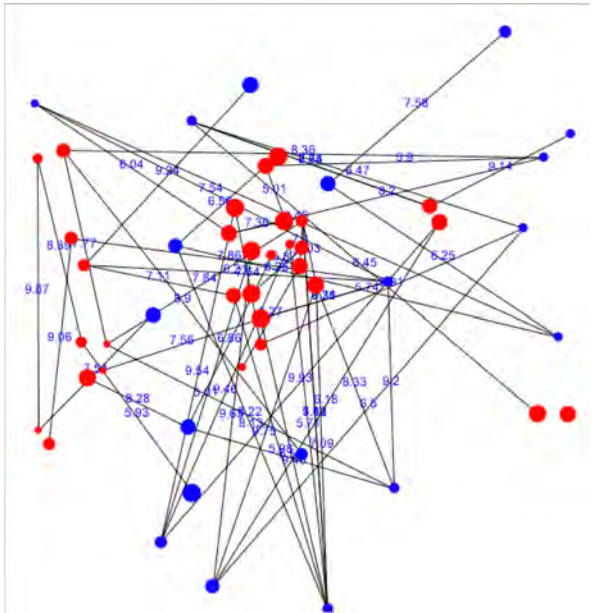
K. Kravari, N. Bassiliades, A Survey of Agent Platforms, Journal of Artificial Societies and Social Simulation, 18 (1) 11, 2015

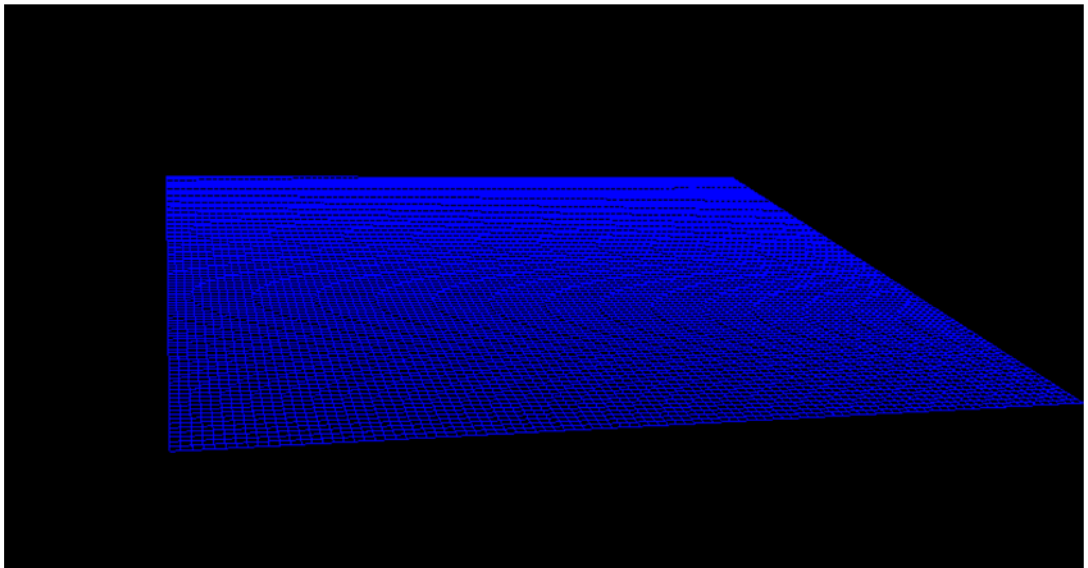
# 主要プラットフォーム比較

- マルチプラットフォームおよび使用フリーで，更新が定期的に行われており，マニュアルやサンプルなどのドキュメントがよく整備され，比較的人気の高いABSプラットフォームとして，レビューがよくされている4種類＋最近人気が上昇している１種類を加えたもので比較.
- MASON, Repast, NetLogo, Swarm, GAMA

# MASON

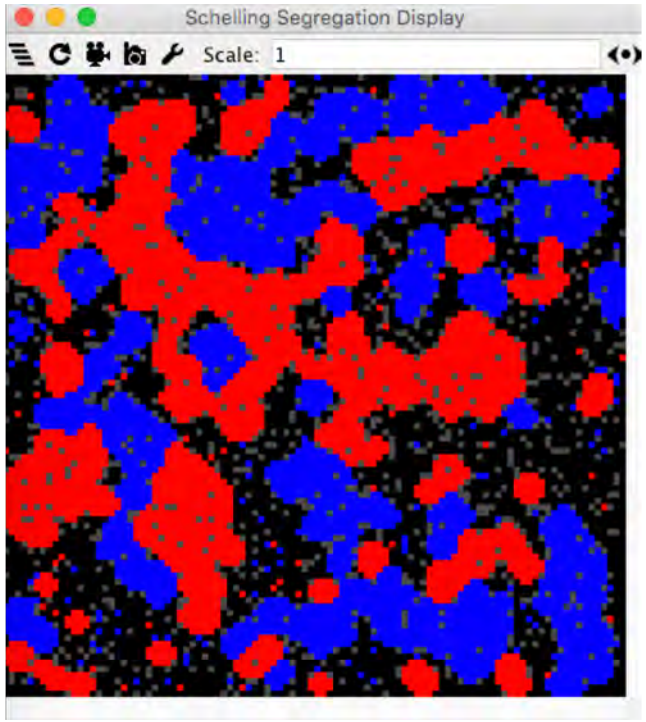
- George Mason Universityで2003年に開発された.
- Multi-Agent Simulator Of Neighborhoods/Network(MASON)
- 100%Javaベース，オープンソース，フリーライセンス
- アドオン
  - 3D toolkit, jFreeChart, Quicktime movies

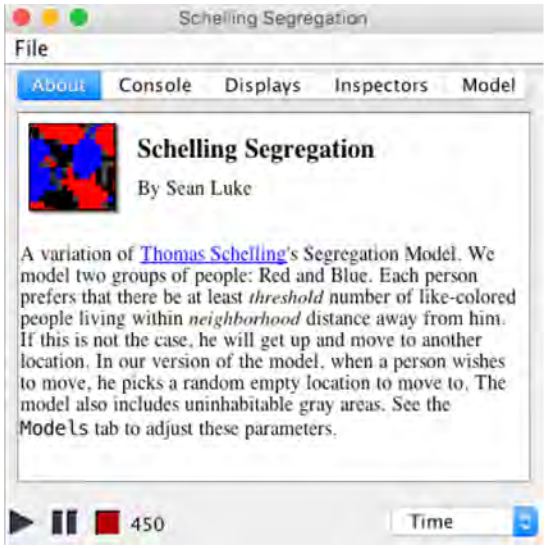


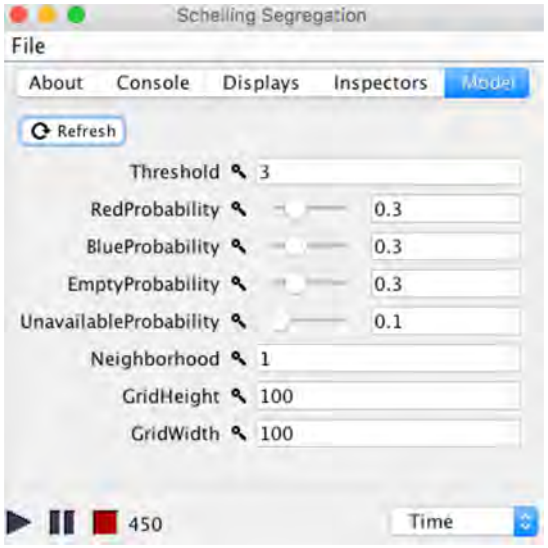


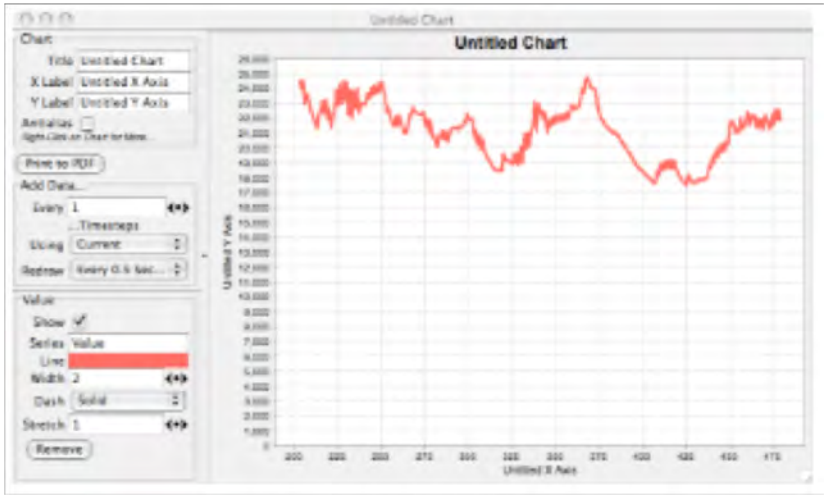
# MASON

- 実行画面


Schelling Segregation Display
Scale: 1

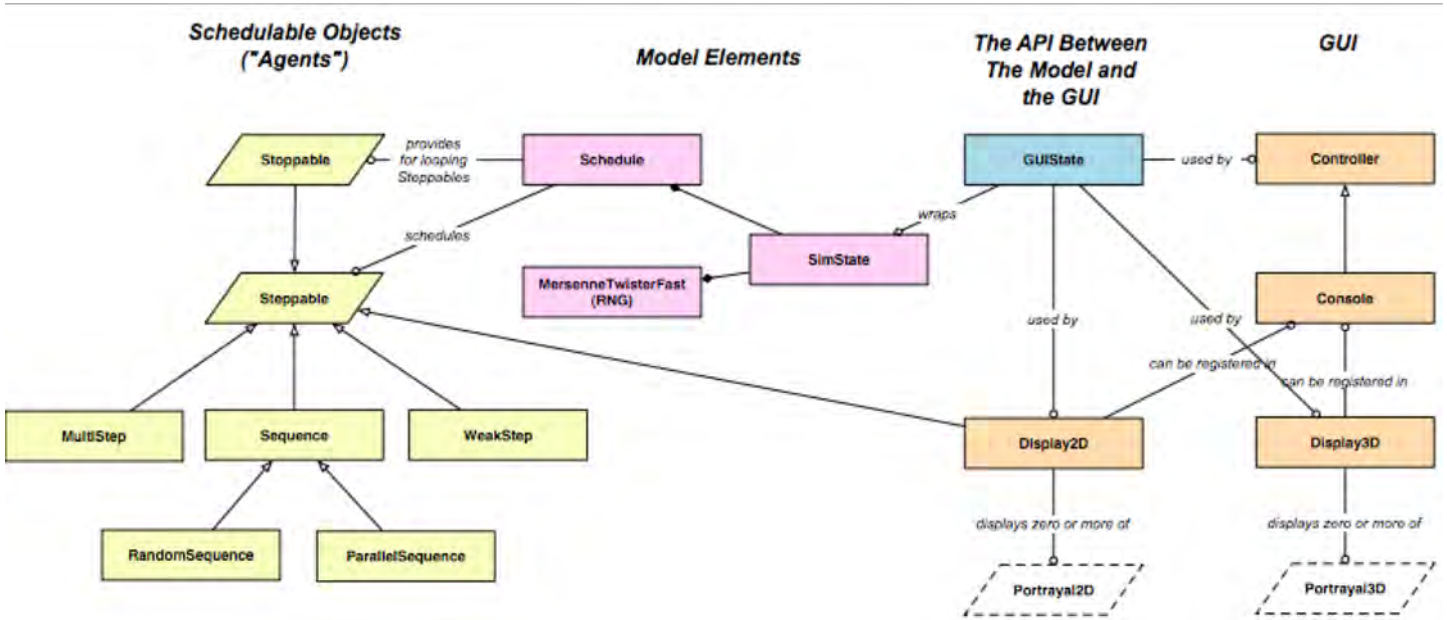

Schelling Segregation
File
About
Console
Displays
Inspectors
Model
Schelling Segregation
By Sean Luke
A variation of Thomas Schelling's Segregation Model. We model two groups of people: Red and Blue. Each person prefers that there be at least threshold number of like-colored people living within neighborhood distance away from him. If this is not the case, he will get up and move to another location. In our version of the model, when a person wishes to move, he picks a random empty location to move to. The model also includes uninhabitable gray areas. See the Models tab to adjust these parameters.
450
Time


Schelling Segregation
File
About
Console
Displays
Inspectors
Model
Refresh
Threshold 3
RedProbability 0.3
BlueProbability 0.3
EmptyProbability 0.3
UnavailableProbability 0.1
Neighborhood 1
GridHeight 100
GridWidth 100
450
Time


Untitled Chart

# MASON

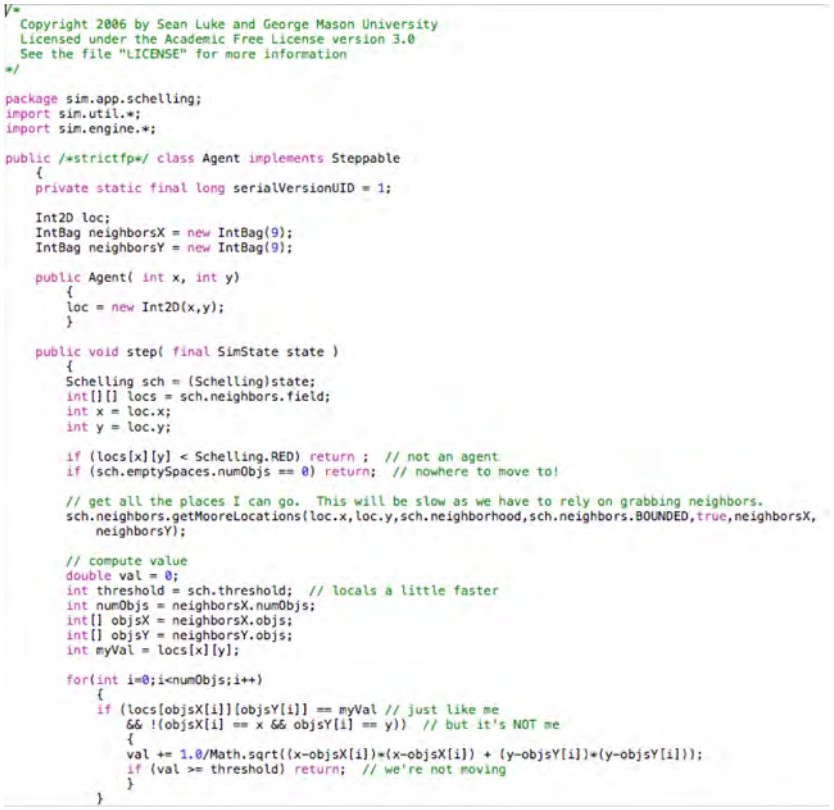
- 開発環境
  - Javaパッケージを利用し，好みのエディターで編集




| Package |
| --- |
| ec.util |
| sim.display |
| sim.display3d |
| sim.engine |
| sim.field |
| sim.field.continuous |
| sim.field.grid |
| sim.field.network |
| sim.portrayal |
| sim.portrayal.continuous |
| sim.portrayal.grid |
| sim.portrayal.network |
| sim.portrayal.simple |
| sim.portrayal3d |
| sim.portrayal3d.continuous |
| sim.portrayal3d.grid |
| sim.portrayal3d.grid.quad |
| sim.portrayal3d.simple |
| sim.util |
| sim.util.distribution |
| sim.util.gui |
| sim.util.media |
| sim.util.media.chart |

# MASON

- 開発言語



```
/*
  Copyright 2006 by Sean Luke and George Mason University
  Licensed under the Academic Free License version 3.0
  See the file "LICENSE" for more information
*/

package sim.app.schelling;
import sim.util.*;
import sim.engine.*;

public /*strictfp*/ class Agent implements Steppable
    {
    private static final long serialVersionUID = 1;

    Int2D loc;
    IntBag neighborsX = new IntBag(9);
    IntBag neighborsY = new IntBag(9);

    public Agent( int x, int y)
        {
        loc = new Int2D(x,y);
        }

    public void step( final SimState state )
        {
        Schelling sch = (Schelling)state;
        int[][] locs = sch.neighbors.field;
        int x = loc.x;
        int y = loc.y;

        if (locs[x][y] < Schelling.RED) return ;  // not an agent
        if (sch.emptySpaces.numObjs == 0) return;  // nowhere to move to!

        // get all the places I can go.  This will be slow as we have to rely on grabbing neighbors.
        sch.neighbors.getMooreLocations(loc.x,loc.y,sch.neighborhood,sch.neighbors.BOUNDED,true,neighborsX,
            neighborsY);

        // compute value
        double val = 0;
        int threshold = sch.threshold;  // locals a little faster
        int numObjs = neighborsX.numObjs;
        int[] objsX = neighborsX.objs;
        int[] objsY = neighborsY.objs;
        int myVal = locs[x][y];

        for(int i=0;i<numObjs;i++)
            {
            if (locs[objsX[i]][objsY[i]] == myVal // just like me
                && !(objsX[i] == x && objsY[i] == y))  // but it's NOT me
                {
                val += 1.0/Math.sqrt((x-objsX[i])*(x-objsX[i]) + (y-objsY[i])*(y-objsY[i]));
                if (val >= threshold) return;  // we're not moving
                }
            }
```

```
package sim.app.schelling;
import sim.util.*;
import sim.engine.*;

public /*strictfp*/ class Agent implements Steppable
    {
    private static final long serialVersionUID = 1;

    Int2D loc;
    IntBag neighborsX = new IntBag(9);
    IntBag neighborsY = new IntBag(9);

    public Agent( int x, int y)
        {
        loc = new Int2D(x,y);
        }

    public void step( final SimState state )
        {
        Schelling sch = (Schelling)state;
        int[][] locs = sch.neighbors.field;
        int x = loc.x;
        int y = loc.y;
```

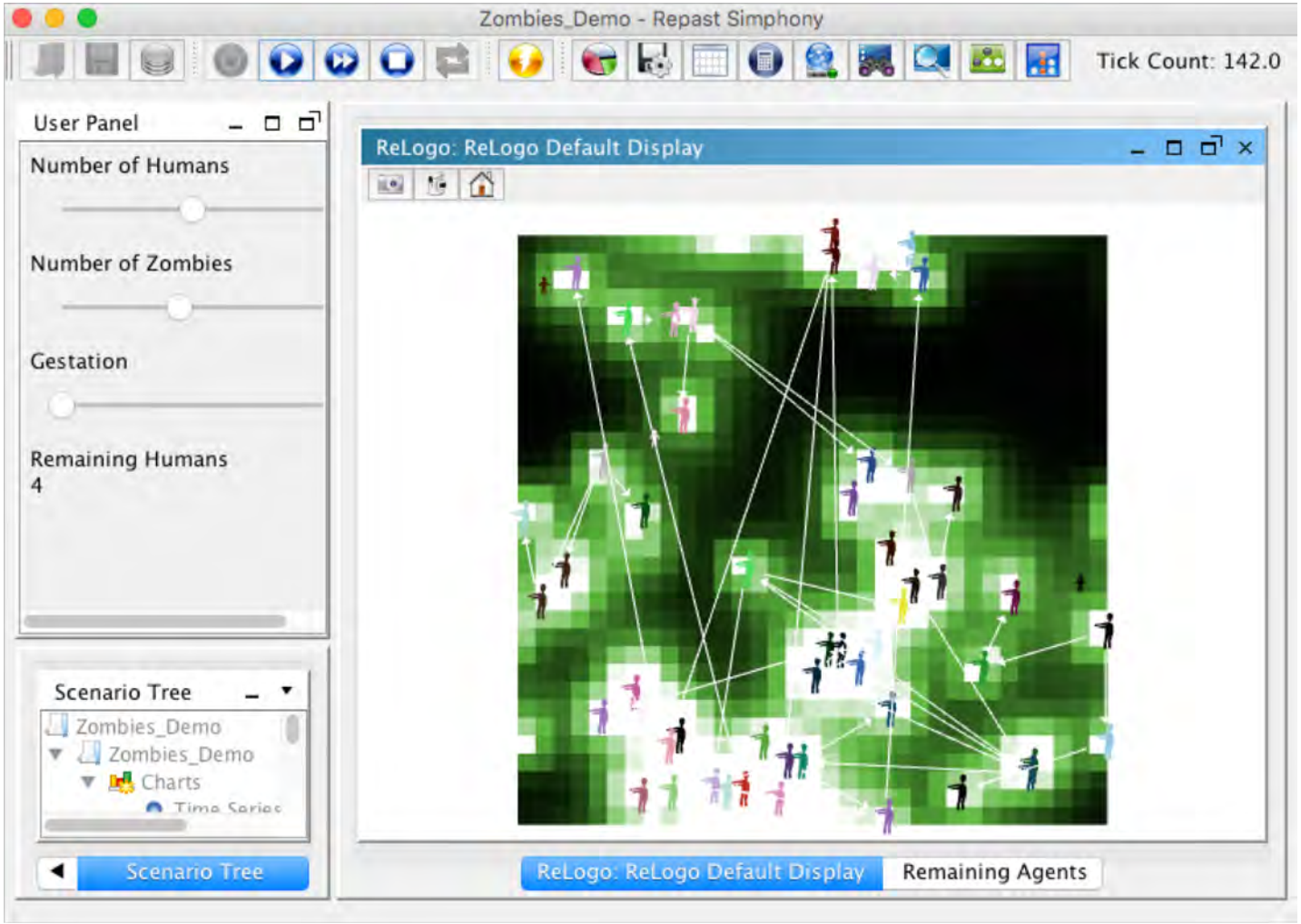
# Repast

- University of Chicagoで2006年に開発された
- Recursive Porous Agent Simulation Toolkit (Repast)
- Javaベース, Win/Mac/Linux
- Javeベース，オープンソース，フリーライセンス
- Eclipse IDEで開発
- Repast Simphony
  - Java, ReLogo, statecharts, Groovy
  - Repast Batch Runs on a Hadoop cluster

- Repast for High Performance Computing
  - C++, Logostyle C++(ReLogo)
  - マルチコア，クラスタなど，大規模並列計算
  - 実行環境：MPI-based cpde, MPICH-2, OpenMPI,

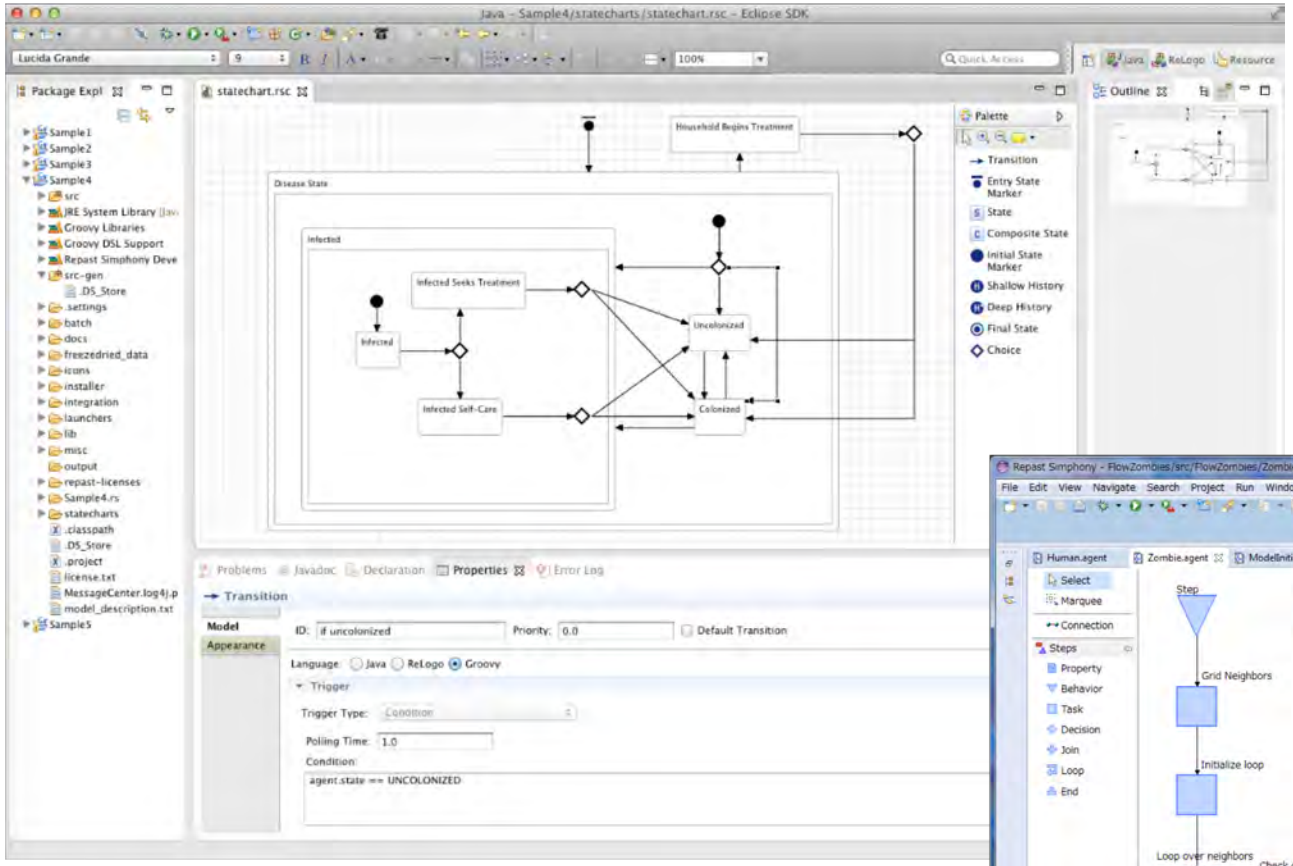
# Repast

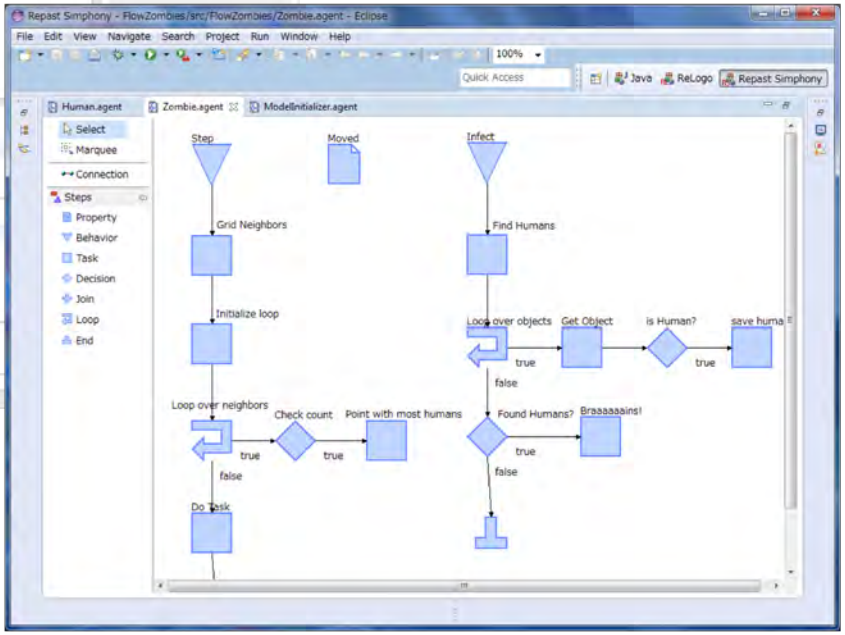
- 実行画面


Zombies_Demo - Repast Simphony
Tick Count: 142.0
User Panel
Number of Humans
Number of Zombies
Gestation
Remaining Humans
4
ReLogo: ReLogo Default Display
Scenario Tree
Zombies_Demo
Zombies_Demo
Charts
Scenario Tree
ReLogo: ReLogo Default Display
Remaining Agents

# Repast

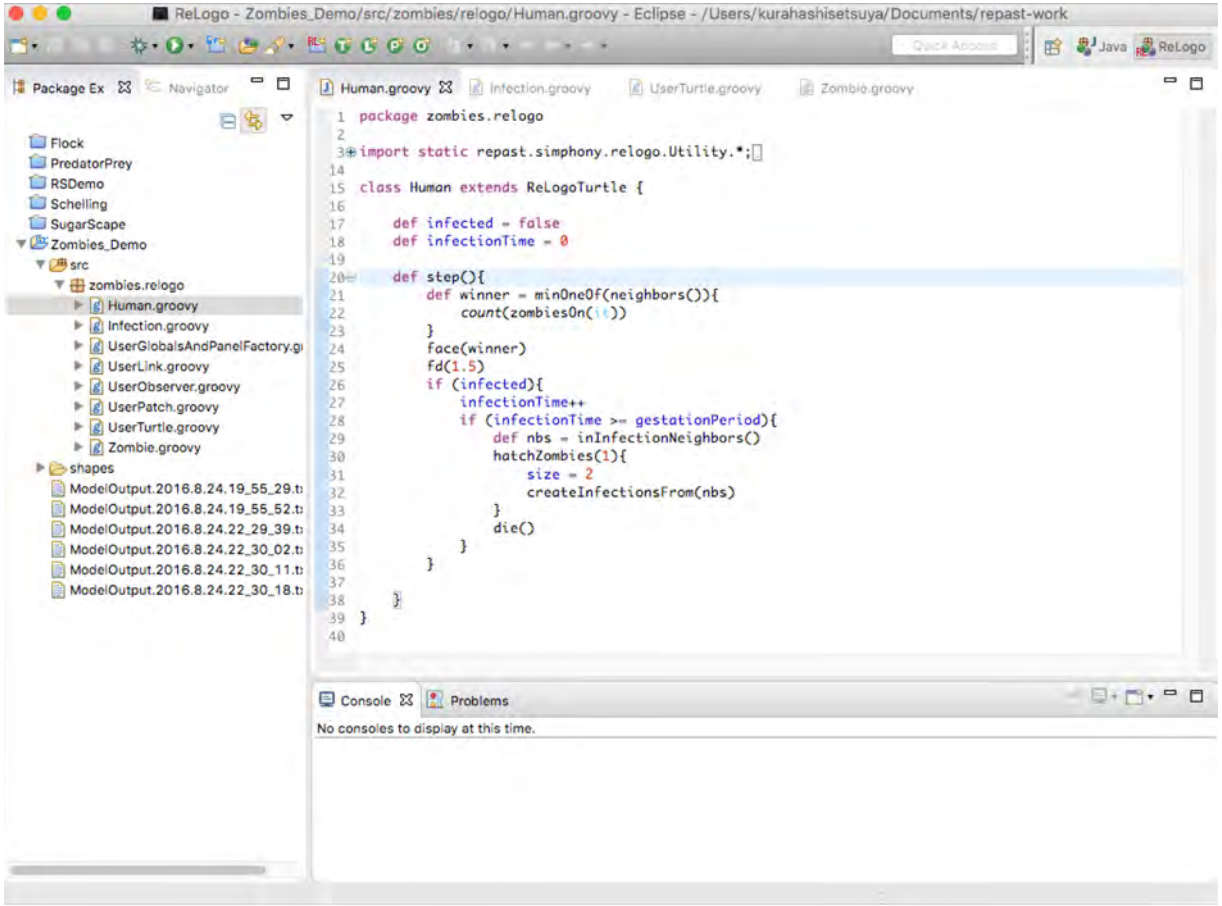
- フローチャートモデリング



# Repast

- 開発環境
  - Java Eclipse



# Repast

- 開発言語(Java)

```
public void step() {
    // get the grid location of this Zombie
    GridPoint pt = grid.getLocation(this);

    // use the GridCellNgh class to create GridCells for
    // the surrounding neighborhood.
    GridCellNgh<Human> nghCreator = new GridCellNgh<Human>(grid, pt,
        Human.class, 1, 1);
    // import repast.simphony.query.space.grid.GridCell
    List<GridCell<Human>> gridCells = nghCreator.getNeighborhood(true);
    SimUtilities.shuffle(gridCells, RandomHelper.getUniform());

    GridPoint pointWithMostHumans = null;
    int maxCount = -1;
    for (GridCell<Human> cell : gridCells) {
        if (cell.size() > maxCount) {
            pointWithMostHumans = cell.getPoint();
            maxCount = cell.size();
        }
    }
}
```

# Repast

- 開発言語(ReLogo)

```
class Human extends ReLogoTurtle {

   def infected = false
   def infectionTime = 0

   def step(){
      def winner = minOneOf(neighbors()){
         count(zombiesOn(it))
      }
      face(winner)
      forward(1.5)

      if (infected){
         infectionTime++
         if (infectionTime >= 5){
            hatchZombies(1){
               size = 2
            }
            die()
         }
      }
   }

}
```

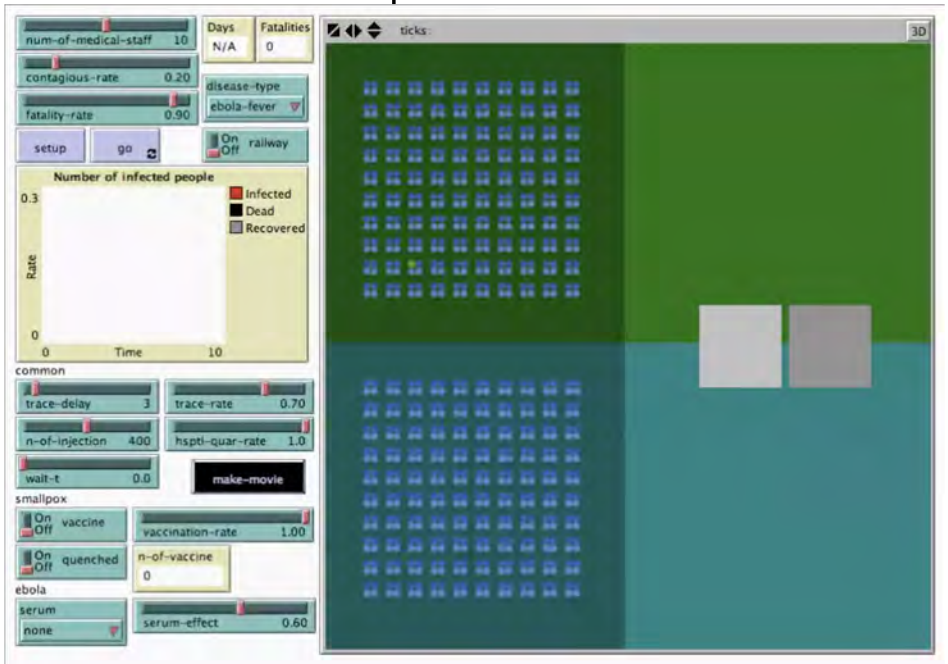
# Repast

- 開発言語(HPC: C++)

```
#include <stdio.h>
#include <boost/mpi.hpp>
#include "repast_hpc/RepastProcess.h"


class RepastHPCDemoModel{
public:
	RepastHPCDemoModel(){}
	~RepastHPCDemoModel(){}
}

int main(int argc, char** argv){

	std::string configFile = argv[1]; // The name of the configuration file is Arg 1

	boost::mpi::environment env(argc, argv);
	boost::mpi::communicator world;

	repast::RepastProcess::init(configFile);

	RepastHPCDemoModel* model = new RepastHPCDemoModel();

	model->init();

	delete model;

	repast::RepastProcess::instance()->done();

}
```

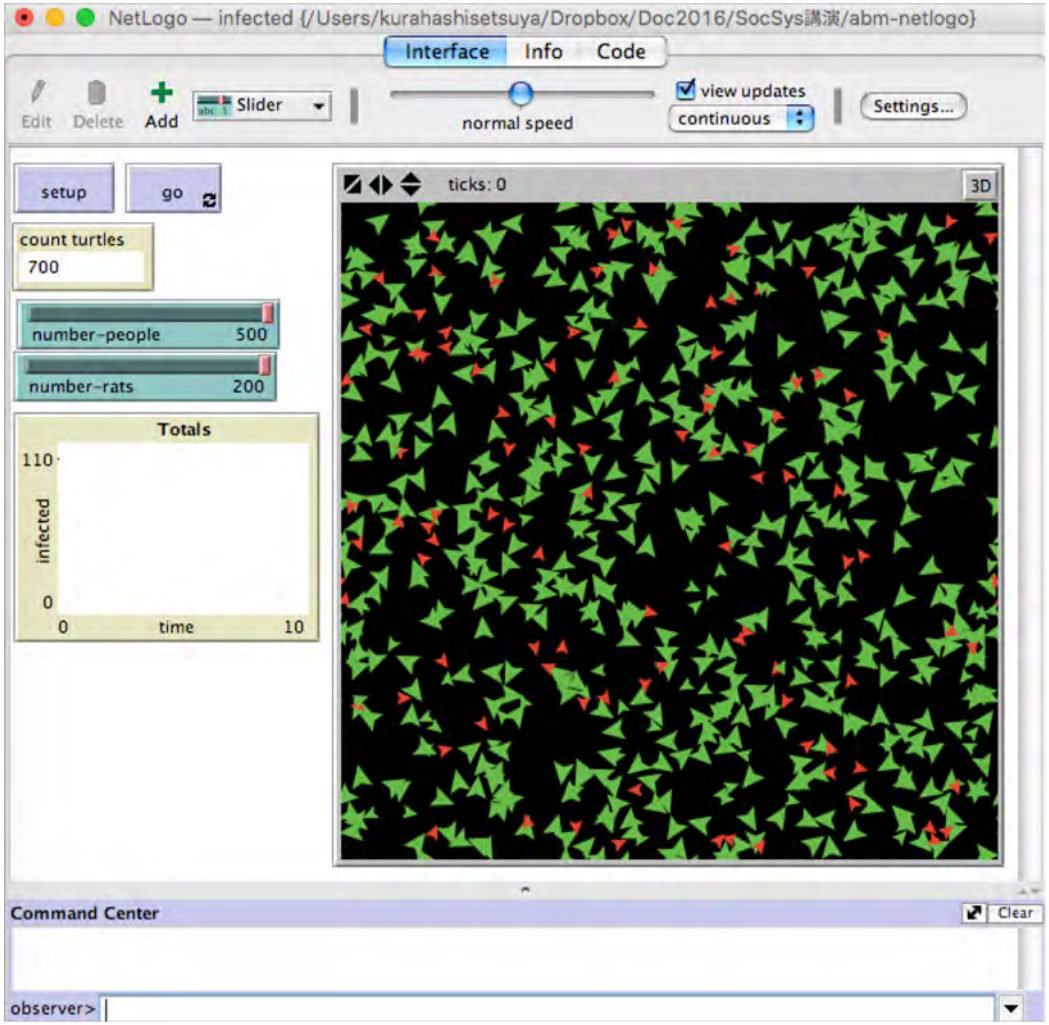
# NetLogo

- Northwestern Universityで1999年に開発された
- Javaベース，フリーライセンス
- Logoプログラミング言語の派生系言語がNetLogo
  - Logo 1967年に開発された教育用プログラミング言語
  - プログラミング経験がないドメインエキスパート向け
- turtle, patch, link, observerの４種類のエージェント
- Extentions: Behaviour space/Hubnet/Networkなど多数



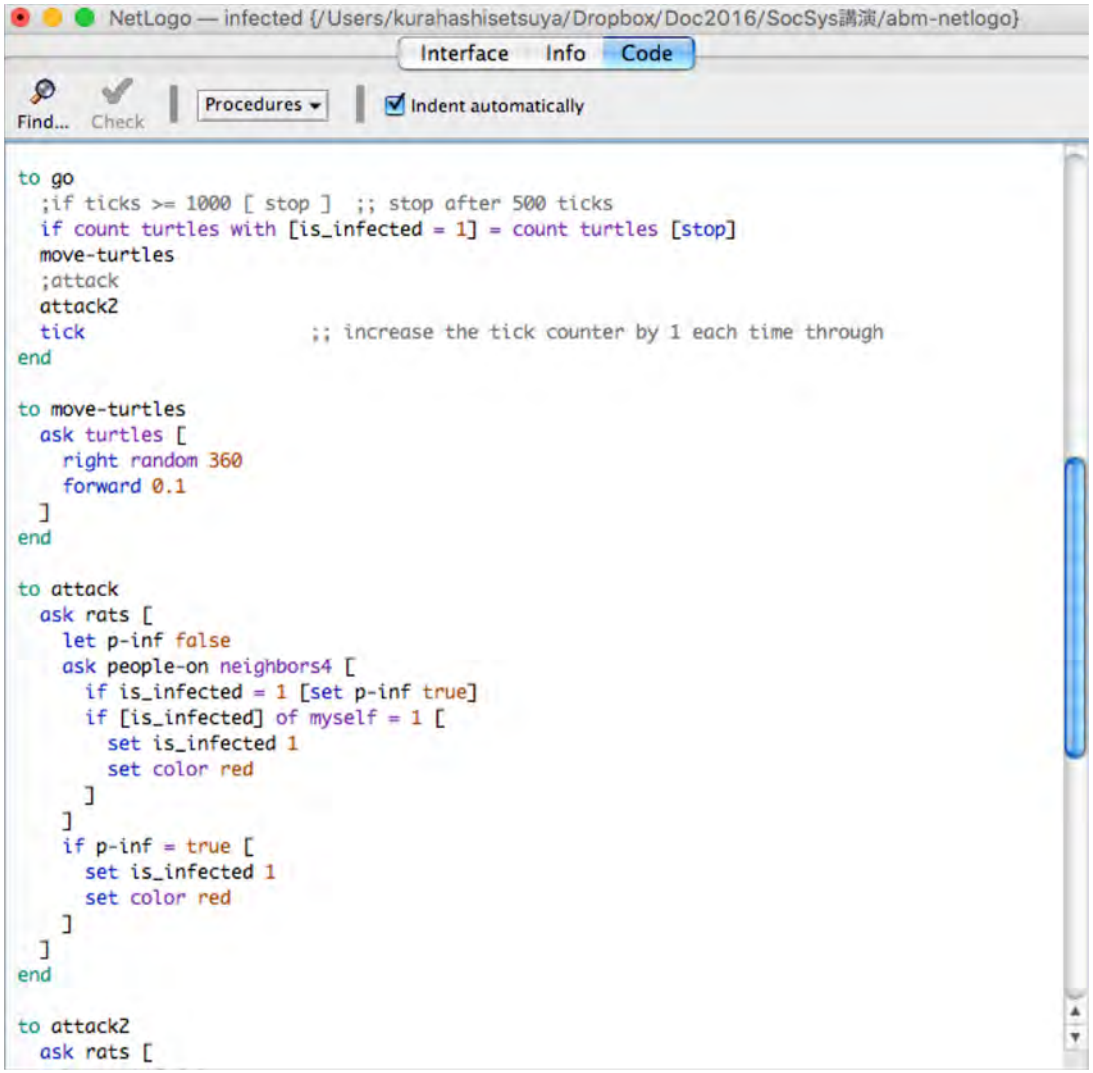
# NetLogo

- 実行環境



# NetLogo

- 開発環境



# NetLogo

- 開発言語

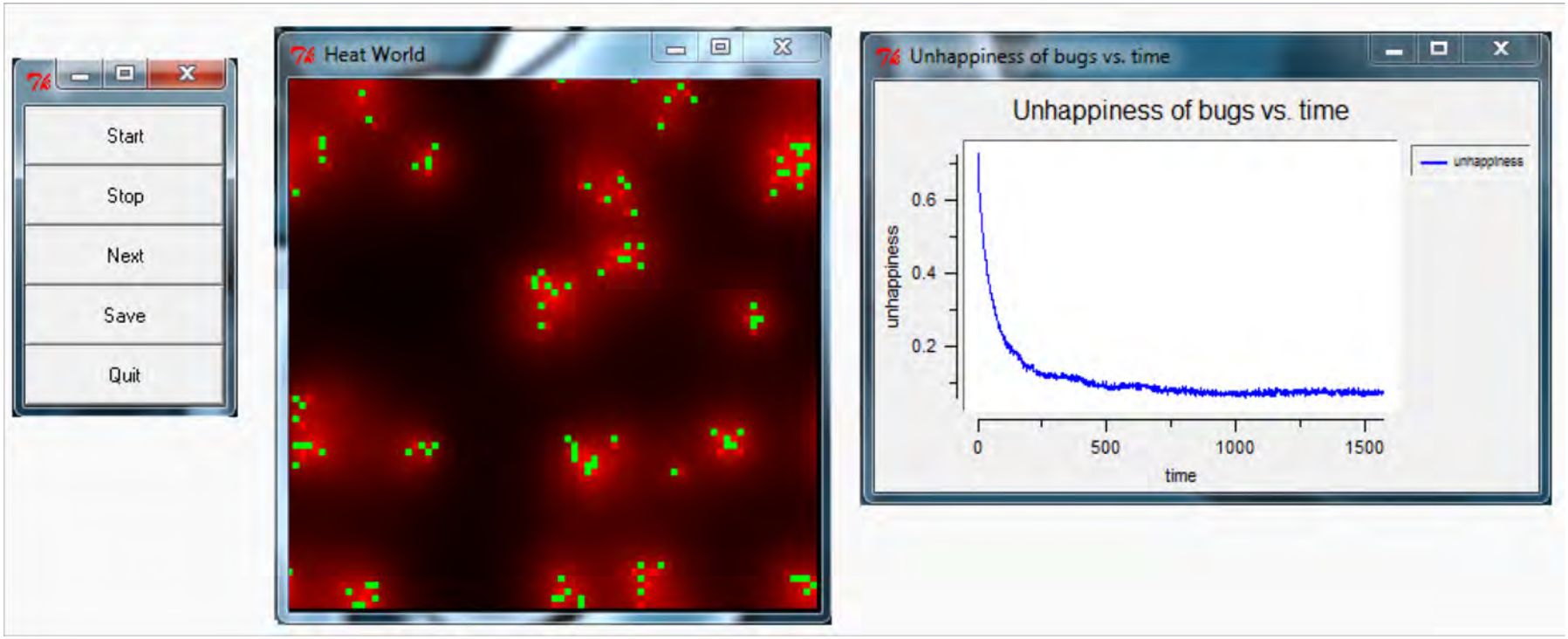
```
turtles-own [
  energy
  is_infected
]

breed [people person]
breed [rats rat]

to setup
  clear-all
  setup-turtles
  reset-ticks
end

to setup-turtles
  create-people number-people [
    set color green
    setxy random-xcor random-ycor
    set size 1.5
    set is_infected 0
 ]
end
```

# SWARM

- Santa Fe Instituteで1990年代半ばに開発された
- Objective-C, Javaでモデル作成可能
- Unix, Windows, (Mac)対応
- MinGW(Minimalist GNU for Windows)上でのObjective-C Swarmも開発されている
- 適用例も多い
  - サプライチェーンマネジメント，クチコミ，社会ネットワーク
  - 交通モデル，職場管理，分散コンピューティング，等

# SWARM

- 実行環境


Start
Stop
Next
Save
Quit
Heat World
Unhappiness of bugs vs. time
Unhappiness of bugs vs. time
unhappiness
unhappiness
0.6
0.4
0.2
0
500
1000
1500
time

# SWARM

- 開発言語
  - Objective-C

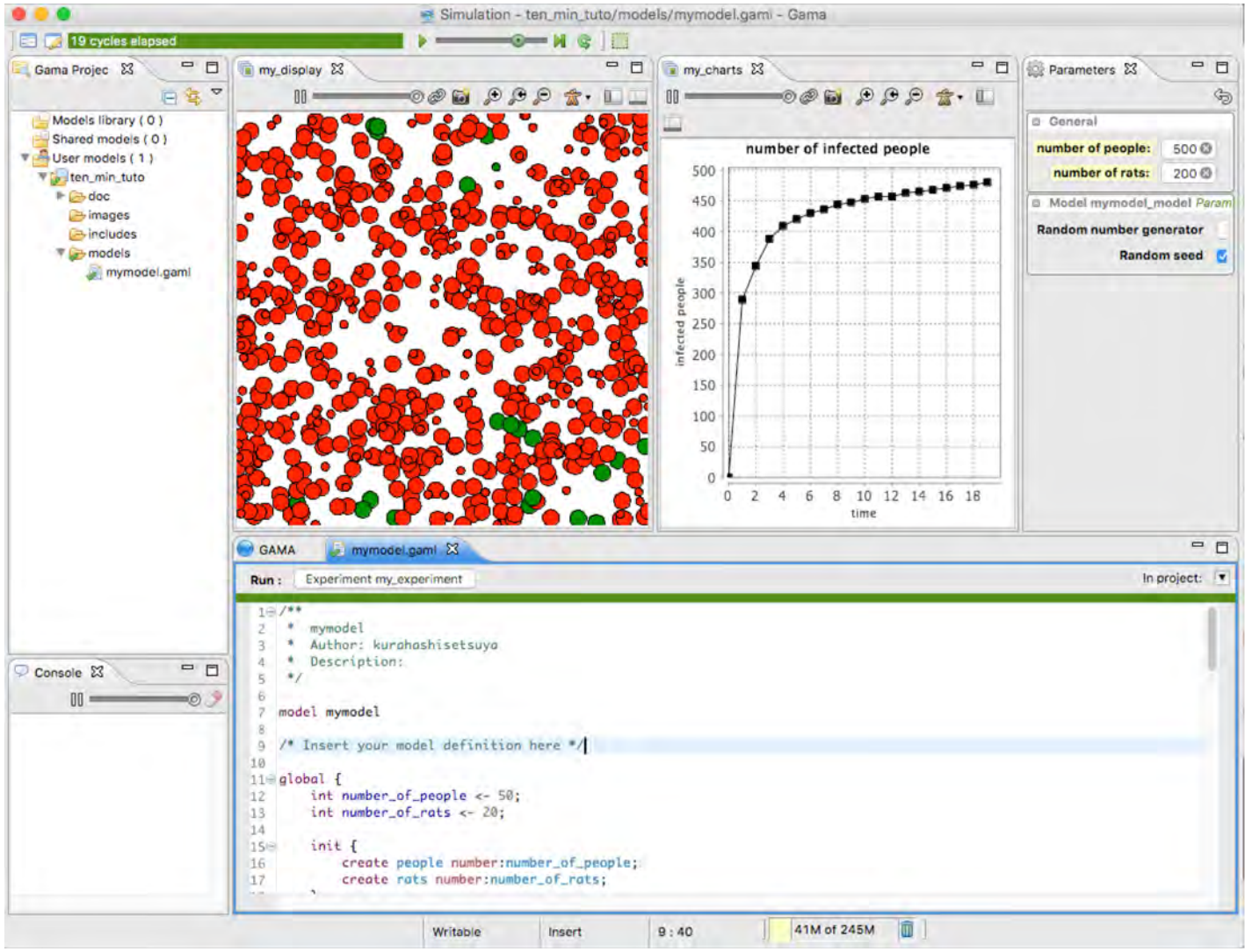
```
@interface HeatbugModelSwarm : Swarm {
 int numBugs; // simulation parameters
 double evaporationRate;
 double diffuseConstant;
 int worldXSize, worldYSize;
 int minIdealTemp, maxIdealTemp;
 int minOutputHeat, maxOutputHeat;
 double randomMoveProbability;

 id modelActions; // scheduling data structures
 id modelSchedule;

 id heatbugList; // list of all the heatbugs
 Grid2d * world; // objects representing
 HeatSpace * heat; // the world
}

-getHeatbugList; // access methods into the
-(Grid2d *) getWorld; // model swarm. These methods
-(HeatSpace *) getHeat; // allow the model swarm to be observed.

+createBegin: aZone; // extra methods you
-createEnd; // provide for Swarms
-buildObjects;
-buildActions;
-activateIn: swarmContext;
```

# GAMA

- IRD/UPMC international research unit UMMISCO（フランス，ベトナムの混成チーム）で2010年に開発された.
- GIS Agent-based Modeling Architecture (GAMA)
- Javaベース，オープンソース，フリーライセンス
- Win, Mac, Linux対応
- JavaベースのエージェントベースGAML言語
  - 2D, 3D，GIS, OSM(open street map data), R, Database
- 交通，都市，感染症，環境などのモデルが多い

# GAMA

- 実行環境＆開発環境



# GAMA

- 開発言語
  - GAML

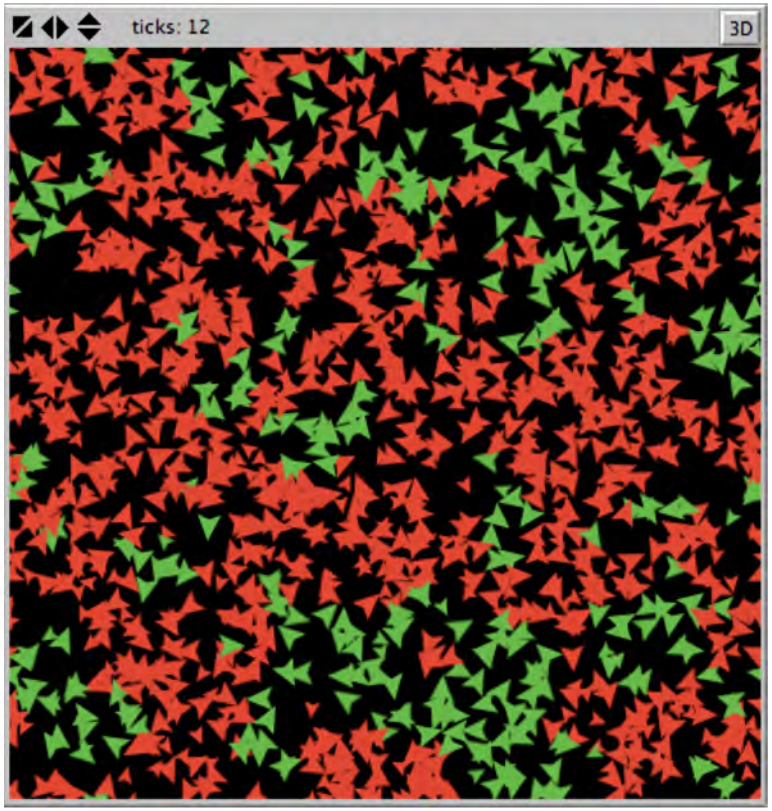
```
model mymodel

global {
	int number_of_people <- 500;
	int number_of_rats <- 200;

	init {
		create people number:number_of_people;
		create rats number:number_of_rats;
	}
}

species my_species {
}

species people skills:[moving]{
	bool is_infected <- false;
	reflex moving {
		do wander speed: 0.5;
	}
	aspect base {
		draw circle(2) color: (is_infected) ? # red : #green;
	}
}
```

# Stupid Model

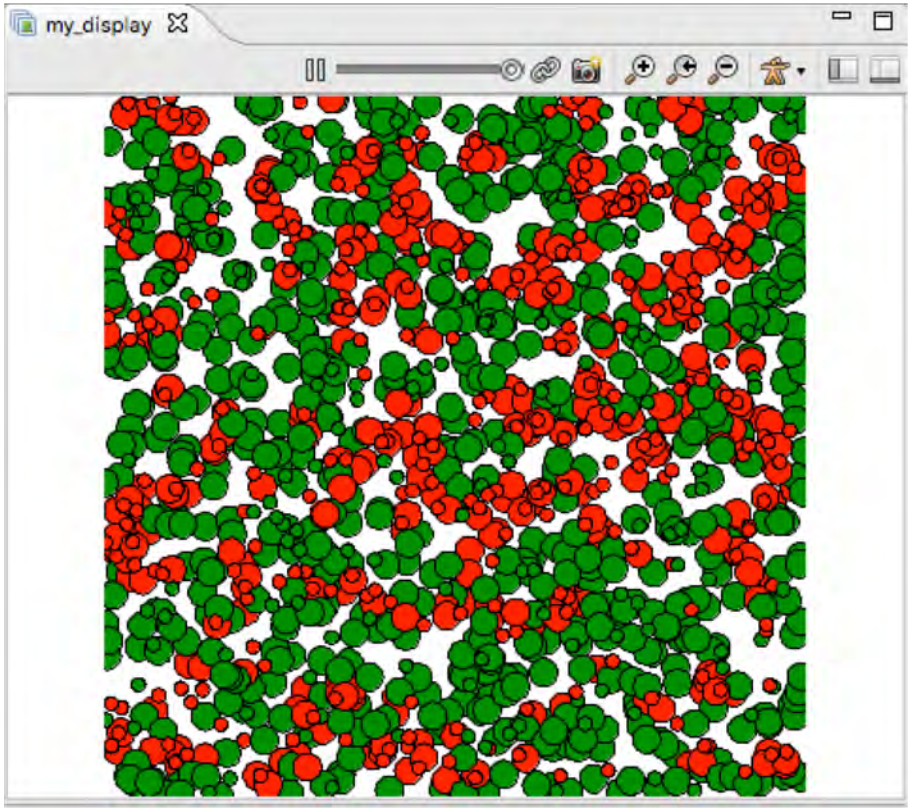
| Version | Characteristics added |
|---|---|
| 1 | 100 agents ("bugs") distributed randomly into a 100 x 100 grid space. Bugs have one action: move to a randomly chosen neighbor cell. Bug locations are displayed graphically. |
| 2 | A second bug action: growing by a constant amount. |
| 3 | Habitat cells that grow food; bug growth is equal to the food they consume from their cell. |
| 4 | "Probes" letting the user see the instance variables of selected cells and bugs. |
| 5 | Parameter displays letting the user change the value of key parameters at run time. |
| 6 | A histogram of bug sizes. |
| 7 | A stopping rule that causes execution to end when any bug reaches a size of 1000. |
| 8 | File output of the minimum, mean, and maximum bug sizes each time step. |
| 9 | Randomization of the order in which bugs move. |
| 10 | Size-ordering of execution order: bugs move in descending size order. |
| 11 | Optimal movement: bugs move to the cell within a radius of 4 that provides highest growth. |
| 12 | Mortality and reproduction: bugs have a constant mortality probability, and reproduce when they reach a size of 10. |
| 13 | A graph of the number of bugs. |
| 14 | Initial bug sizes drawn from a random normal distribution. |
| 15 | Cell food production rates read from an input file; graphical display of cell food availability. |
| 16 | A second "species": predator agents that hunt bugs. |

# Infected model

- AgentはPeopleとRats
- Ratsの半分は病原菌に感染している
- 双方が接触するとお互いに感染しあう




NetLogo




GAMA

# 実行速度

| Model | Agents | Objects | MASON | NetLogo | Repast | Java Swarm | Objective C Swarm | GAMA |
|---|---|---|---|---|---|---|---|---|
| 1 | 100 | 100 | 7.9 | 8.1 | 12 | 6.1 | 2.2 | |
| 3 | 100 | 10,100 | 10 | 30 | 12 | 379 | 68 | |
| 8 | 100 | 10,100 | 9 | 31 | 12 | 176 | 70 | |
| 11 | 100 | 10,100 | 11 | 49 | 13 | 245 | 85 | |
| 12 | 150 | 10,150 | 12 | 63 | 16 | 292 | 95 | |
| 15 | 1760 | 29,800 | 113 | 417 | 124 | 2984 | 1528 | |
| 16 | 1660 | 29,800 | 122 | 314 | 123 | 2338 | 1029 | |
| Infected* | 700 | 700 | | 11 | | | | 7 |
| Infected* | 1400 | 1400 | | 20 | | | | 10 |

*Kurahashi 2016

S. F. Railsback, S. L. Lytinen, S. K. Jackson, Agent-based simulation platforms: review and development recommendations, Simulation 82 (9), 600-623, 2006

# 実行速度

| Model | # of agents | NetLogo | ReLogo | ReLogo/NetLogo |
|---|---|---|---|---|
| 1 | 100 | 6.3 | 173 | 27.7 |
| 3 | 100 | 9.1 | 189 | 20.9 |
| 11 | 100 | 9.7 | 191 | 19.7 |
| 12 | 140 | 13.5 | 280 | 20.7 |
| 15 | 1330 | 103 | 2509 | 24.3 |
| 16 | 729 | 65.1 | 1516 | 23.3 |
| Zombie* | 55 | 30/k step | 5/k step | 0.17 |

*Kurahashi 2016　（Groovy++？）

S. L. Lytinen, S. F. Railsback, The evolution of agent-based simulation platforms: A review of NetLogo 5.0 and ReLogo, Proceedings of the fourth international symposium on agent-based modeling and simulation, 2012

# まとめ

- プログラミングに精通していない社会科学系の研究者であれば，NetLogoがおすすめ．インストールやプログラミングの簡便さ，日本語マニュアル，サンプルの豊富さは群を抜いている．そこそこの実行速度もあり，プログラミングと合わせた時間で見れば，未だに優位性を保つ．プログラミング言語と日本語対応は，要注意．

- コンピュータサイエンス系の研究者であれば，Repast，MASONがおすすめ．Javaに慣れている人であればなおさら．スケーラビリティ，実行速度ともストレスがない．ReLogoであれば，モデル作成も比較的楽，C++で書くならRepastHPCが使える．

- GAMAは独自言語だが，マニュアルもしっかりしており，比較的馴染みやすい．GIS対応やスケーラビリティも評価は高い．

- 商業ユースなら，AnyLogicの評価が高い様子．サポート，GUIも優秀．

- artisoc, SOARSは，日本人の研究者におすすめ．日本語のサンプルやマニュアルも多い．なんといっても，教えてくれる人が日本人．

# 参考文献


- K. Kravari, N. Bassiliades, A Survey of Agent Platforms, Journal of Artificial Societies and Social Simulation, 18 (1) 11, 2015
- S. F. Railsback, S. L. Lytinen, S. K. Jackson, Agent-based simulation platforms: review and development recommendations, Simulation 82 (9), 600-623, 2006
- S. L. Lytinen, S. F. Railsback, The evolution of agent-based simulation platforms: A review of NetLogo 5.0 and ReLogo, Proceedings of the fourth international symposium on agent-based modeling and simulation, 2012
- K. Bajrachaya, R. Duboz, Comparison of Three Agent-based Platforms og the Basis of a simple Epidemiological Model, Proceedings of the Symposium on Theory of Modeling & Simulation, 2013
- S. Railsback, S. Lytinen, V. Grimm, StupidModel and Extensions: A Template and Teaching Tool for Agent-based Modeling Platforms, Swarm Development Group. http://condor.depaul.edu/~slytinen/abm/
- A. G. Issac, The ABM Template Models: A Reformulation with Reference Implementations, Journal of Artificial Societies and Social Simulation, 14 (2) 5, 2011
- C, Nikolai, G. Madey, Tools of the Trade: A Survey of Various Agent Based Modeling Platforms, Journal of Artificial Societies and Social Simulation, 12 (2) 2, 2009