



Jul 1st, 12:00 AM

MR POTATOHEAD Framework – A Software Tool for Collaborative Land-Use Change Modeling

Mike Livermore

Follow this and additional works at: <https://scholarsarchive.byu.edu/iemssconference>

Livermore, Mike, "MR POTATOHEAD Framework – A Software Tool for Collaborative Land-Use Change Modeling" (2010).
International Congress on Environmental Modelling and Software. 72.
<https://scholarsarchive.byu.edu/iemssconference/2010/all/72>

This Event is brought to you for free and open access by the Civil and Environmental Engineering at BYU ScholarsArchive. It has been accepted for inclusion in International Congress on Environmental Modelling and Software by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

MR POTATOHEAD Framework – A Software Tool for Collaborative Land-Use Change Modeling

Mike Livermore

*Department of Computational Social Science and Center for Social Complexity, George
Mason University, USA*

Abstract: Agent-based land-use cover/change models (ABM/LUCC), which can be used for various types of complex human-environment interactions, recently have been gaining popularity. ABM/LUCC models combine a spatial representation of the landscape with key decision-makers (e.g. farmers and institutions) and an environment for interaction. There is inherent complexity in modeling human-environment interactions which need to be assessed at multiple spatial and time scales, and often multiple models are integrated; however, current methods of developing integrated models are often cumbersome and communicating design and results can be a great challenge and would be aided by a common framework in which different models can be compared to and collaborate with each other. Attempting to address these concerns is the MR POTATOHEAD framework, which encapsulates many of the principal elements generally used in ABM/LUCC models, providing a common medium for comparison and collaboration. The MPH would easily allow multiple models to share both data and functions so that each designer can concentrate on their area of expertise. This paper presents a Java-based implementation of the MR POTATOHEAD framework, giving a modeler a user-friendly environment in which to build a wide range of collaborative ABM/LUCC models. Through a web interface, the model can be designed with little or no coding, if desired; this would open up complex model building to researchers without any computational background. Alternatively, a model can be built in a language of choice, as long as the proper interface is provided to an MPH-compliant model; in a later phase the ability for models to be connected in real time, so that both models are running simultaneously feeding each other results, will be added. Based on Java, the framework can also be used directly and extended by experienced computational scientists.

Keywords: land-use change; agent-based modeling; collaborative modeling

1. INTRODUCTION

Considering the increasing interconnected human-caused problems of resource depletion, climate change and loss of biological diversity, there has been a growing need for researchers to model these types of complexities. One type of modeling becoming more common is agent-based land-use cover/change models (ABM/LUCC), which can be used for the above as well as other types of human-environment interactions. ABM/LUCC models use a spatial representation of the landscape (either virtual or realistic using GIS) along with important decision-makers (farmers, policy-makers, institutions, residents, owners) who directly impact that same landscape and an environment or space to allow the agents to interact with each other and the landscape.

ABM/LUCC models may impact a wide variety of resources such as water usage and quality, air quality, forests, soils, and species biodiversity. There is inherent complexity in modeling human-environment interactions which need to be assessed at multiple spatial scales, such as local, regional and global, as well as multiple time scales,

such as daily, annually or across decades. It is difficult to incorporate everything into one model; however, current methods of developing integrated models are often cumbersome and require either tight fits (which can limit flexibility) or loose fits which often have only minimal integration (e.g. one model providing input values for another model's execution). Additionally, communicating design and results can be a great challenge and would be aided by a common framework in which different models can be compared to and collaborate with each other.

Addressing these concerns is the MR POTATOHEAD ([Model Representing Potential Objects That Appear in The Ontology of Human-Environmental Actions & Decisions](#)) or (MPH) framework, a conceptual design framework which encapsulates many of the principal elements generally used in ABM/LUCC models into related sections, which each model can optionally use, providing a common medium for comparison and potentially collaboration. Introduced by Parker et al. [2006], the first use of this framework compared five somewhat diverse ABM/LUCC models. As explained by Parker et al. [2006], the exercise helped to reveal important common elements across the models, while also unveiling the unique contributions of each model. Additionally, elements that were missing from all models were uncovered. The analysis, being a first step toward collaboration, pointed the way towards sharing and integration of compatible models that could enhance all models involved.

In a second use of the MPH framework, four ABM/LUCC models specifically used in different frontier regions (such as Amazon and Mexican rainforests) were compared through the common "eyepiece" of the MPH framework. As described by Parker et al. [2006], the comparison was fruitful in both suggesting what kinds of experiments could be conducted for further exploration and showing what may be missing or could be emphasized more on a model which was included or emphasized on another model.

The examples above have shown the usefulness of the MPH framework in providing cross-site comparisons of LUCC models, identifying both commonalities and differences between the models. Furthermore, they have allowed us to glimpse a logical extension into a future where models and sub-models may not only be compared but shared and easily integrated. This next step, the implementation of the MPH framework into a usable modeling environment, is detailed below and is the focus of this paper.

2. MR POTATOHEAD FRAMEWORK MODELLING ENVIRONMENT

2.1 Overview

The MR POTATOHEAD (MPH) conceptual framework is envisioned to allow agent-based & land-use cover/change (ABM/LUCC) models to more easily share components and interface, thereby streamlining the model building process while providing flexibility and allow expansion of the model designer base into areas of participants with little or no computational science expertise, but with no loss of modeling power. The MPH easily allows multiple models to share both data and functions so that each designer can concentrate on their area of expertise; for example, water quality, air quality, land-use change, labor market, etc. These could all be separate models or sub-models within other MPH-compliant models. Any type of model/sub-model may be linked to an MPH model by providing a suitable interface. The interface could be as simple as providing a function for one element (e.g. soil quality change); or as complex as sharing major sections with one model providing one portion, another model supplying another large portion, and even a third section combining elements of multiple models in the same function.

The MPH modeling environment consists of two integrated systems, a web-based interface for describing the model (the Model Builder), and a Java-based modeling environment for execution and output generation, which is actually an extension of University of Chicago's [2000] Recursive Porous Agent Simulation Toolkit or [Repast Symphony](#) (the Simulation Environment). In the model builder, the user specifies all of the model parameters representing elements in the MPH framework, such as landscape, spatial, non-spatial and agent details; this meta-model of model details is then saved to a database or XML file. The MPH simulation environment then loads the meta-model, builds the

model to the specifications of the meta-model, and executes the model. The next two sections explain in detail these two interconnected systems.

2.2 Meta-Model Builder Graphical User Interface

2.2.1 MP Head Build Parameters

The parameters appearing in the web-based builder GUI (listed in Table 1) provide the definition for generating the model in the RePast/MPH framework. Each parameter causes various actions to take place in the model initialization in the MPH. These actions can include setting properties, instantiating classes and disposing of other classes, loading data from files, accessing/executing other models and/or setting up links between them, or any combination of these. Since understanding how these build parameters are processed in the MPH framework is important in comprehending how the MPH/RePast executes and manages the model, we will go into detail what actions these parameters represent in the MPH framework. The following sections describe each of the different information areas.

Information Area	Parameter Item	Usage Notes in GUI and/or MPH RePast
Agents	Agent Decision Model	Based on data fields, formula is defined or loaded from external model
Agents	Agent Resource	Data fields (usually agent properties) are turned on/off
Agents	Agent Trait	Data fields (usually agent properties) are turned on/off
Agents	Agent Type	References agent class which automatically has certain traits
Demographic Dynamics	Demographic Dynamic	Formula can be standard function defined in Utility class or custom entered in GUI
Interfaces	Interfaces	MPH data fields are picked to link to external model's input/output
Landscape	Agent Parcel Relation	Affects agent and/or parcel properties
Landscape	Data Layer	Added to MPH Landscape as DataLayer
Landscape	Land Decision Unit	Parcel property turned on/off
Landscape	Land Realism	Changes underlying grid and data layer formats
Landscape	Land Structure	Changes movement and access methods for grid and data layers
Landscape	Parcel Structure	Parcel property
Land Exchange	Buyer Model	Buyer model/formula
Land Exchange	Buyer Parcel	Property of Buyer class which affects the land buyer formula
Land Exchange	Buyer Terms	Property of Buyer class which affects the land buyer formula
Land Exchange	Exchange Mechanism	Formula can be entered for land allocation
Land Exchange	Exchange Triggers	Property affecting land exchange process
Land Exchange	Supplier Model	Land supplier model/formula
Land Exchange	Supplier Parcel	Property affecting land supplier formula
Land Exchange	Supplier Terms	Property affecting land supplier formula
Non-Spatial Data Structures	Economic Data	Can be per Land Use, per Parcel, or single value; may affect land use and/or parcel properties

Non-Spatial Data Structures	Economic Function	Formula affects economic classes
Non-Spatial Data Structures	Institutional Political	Formula/data field affects institutional class
Non-Spatial Data Structures	Non Spatial Network	May add network data layer
Spatial Data	Factors Affecting Data Layer	Affects data layer
Spatial Data	Neighborhood Effects	Affects parcel and agent properties
Spatial Data	Network Model	Causes addition of network data layer to context
Spatial Data	Potential Land Use	Direct driver of Land Use classes
Temporal Dynamics	Temporal Iterations	Affects batch run properties
Temporal Dynamics	Temporal Scheduling	Affects agents' scheduling properties as well as charts

Table 1: MPH parameters in model builder

2.2.2 Landscape Section

In the landscape section (Figure 1), the Land Realism parameter, with Real or Abstract option values, cause a difference in how the “space” and “grid” objects are handled; for instance, for the Real option, a GIS grid layer will be used instead of the default grid. Similarly, for the Land Structure parameter, with the Cell or Vector options, the agent’s method of movement will adjust accordingly as well as the parcel definition. The parcel structure parameter will change the mutability of parcels, with the variable option allowing parcels to change in size endogenously. The decision unit will turn on or off the ability for parcels to be assigned more than one land-use. The agent/parcel relation parameter is actually a multi-select option list allowing more than one selection; specifically, the option Multiple Agents, which allows more than one agent to own a parcel can be used with either the single parcel option (forcing each agent to own only a single parcel) or the multiple parcels option (allowing an agent to own multiple parcels). Lastly, data layers are added to the context with either data loaded from a file (if filename is given) or a formula specifying how the data is calculated for each cell. Data layer resolution is not fixed; therefore it may be specified by parcel or by cell with a cell size (not shown in figure 1). The equation editor, which pops up when pressing any formula creator button, is explained in a separate section (Equation Editor) below.

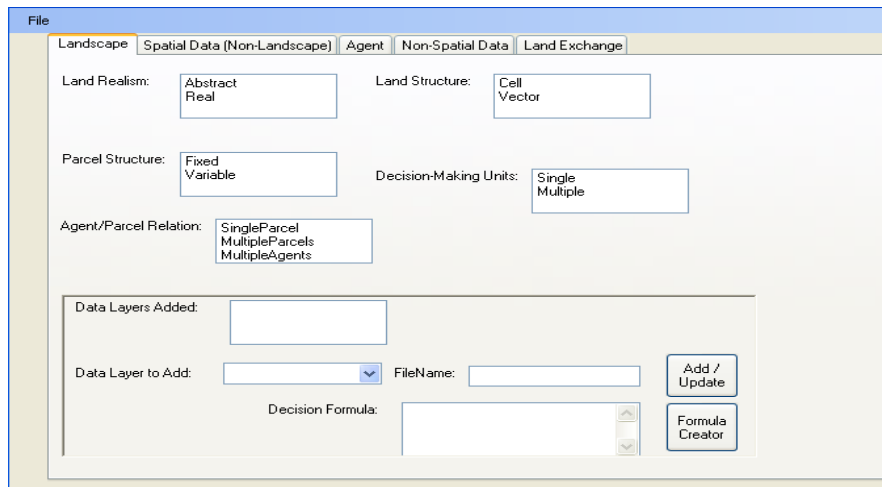


Figure 1: The Landscape section entry screen in the MPH Parameter GUI

2.2.3 Other Spatial Data

For the spatial data structures section (Figure 2), the potential land uses option defines the allowable land-use types; if a file is selected then the file is loaded along with additional data columns for each land-use type (such as conversion cost, input price). The network model will cause a network layer to be added with data loaded from a file or with a given formula specifying the data of the network. Lastly, neighborhood effects define how neighborhood effects affect agent decision making or other functions.

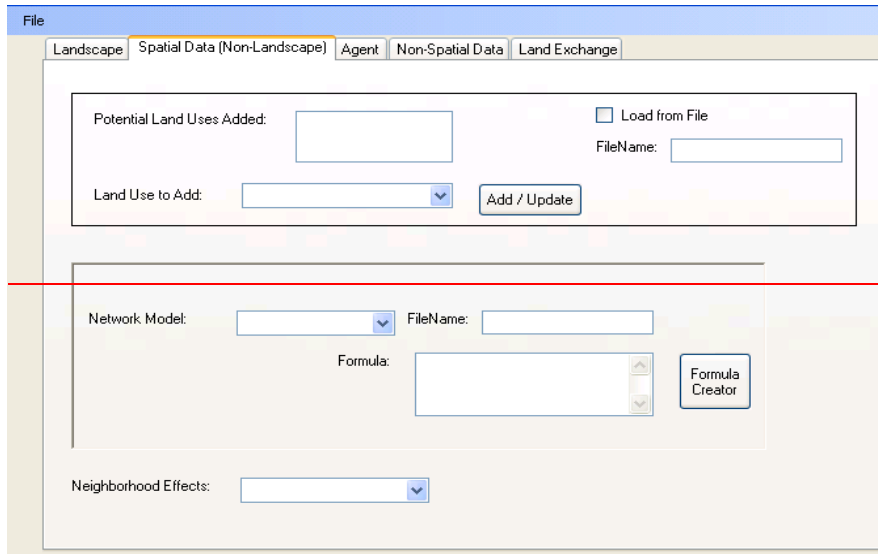


Figure 2: The Spatial Data area entry screen in the MPH Parameter GUI

Formatted: Left

2.2.4 Non-Spatial Data

In the non-spatial data section, the first of three sub-areas is the economic structures sub-area. The economic data options allow various economic-related data to be attached to land-use types, parcels or overall (single values). Economic functions may be used by agents and/or parcels in their various formulas. Likewise, institutional/political rules and constraints may be used by agents in their decision-making and/or other processes. Finally, non-spatial networks will add a network data layer with which agents will interface.

2.2.5 Agent Section

The agent area (figure 52) allows various standard as well as custom agent types to be used. Within each agent type, standard and/or custom traits may be added; additionally, for each trait the memory attribute, if selected, will allow agents to store a history of value changes to that trait. A decision model will also be entered or loaded from a file, which will define the agent's decision making for land-use change (or another function); this may be a simple heuristic sub-model or a complex learning sub-model such as genetic algorithms or neural networks.

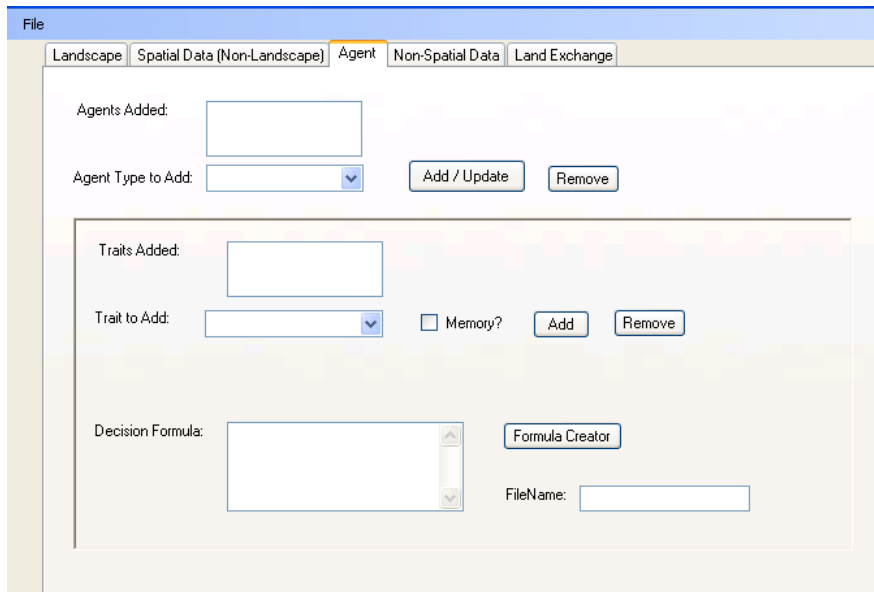


Figure 23: The Agent area entry screen in the MPH Parameter GUI

2.2.6 Land Exchange Section

Finally, the land exchange section includes three sub-areas which define the parameters connected to the exchange of land parcels between agents. The land supplier sub-area includes a supplier model parameter, which will specify a formula based on a specific motivation; alternatively, a formula may be entered or loaded from a file. This formula, along with the other supplier parameters (terms offered and parcels supplied) and the formula specified in the exchange rules, will determine the details of when and how much the supplier will offer a parcel. Likewise, for the acquirers of land sub-area, the buyer model/formula along with the other buyer parameters and the exchange rules, enforce the method the buyer uses when acquiring land parcels. In the exchange rules sub-area, the trigger will specify which property of the agent will trigger an event which causes a possible land exchange to take place. The allocation mechanism specifies a model/formula to use in conjunction with the buyer and supplier models to conclude the transfer.

2.2.7 Equation Editor

Appearing in many areas is a Formula Creator button attached to a particular parameter. This opens up the “Equation Editor”, which allows a function or equation to be entered using any combination of math operators (including exponents, SUM, etc.) along with “MPH data fields.” An MPH data field is actually any parameter which can be calculated for a time step and/or spatial location (parcel or cell); for instance, spatial data layers, economic data and agent decision rules. Conditional statements (such as IF/ELSE) may also be used. Execution timing (pre- or post-step) is also user-configurable.

2.2.8 Other Future Sections

Some of the sections of the MPH framework, including demographic dynamics, temporal dynamics and interfaces will be implemented in a later phase. Also, though the land exchange section is in the GUI it is not yet fully implemented so will be fully functional in a later phase. Current sections will be enhanced and built upon in later phases as well. For example, a formula creator function will be an option to be added for the memory of agent attributes.

2.3 Java-Based MPH-Repast Simulation Environment

2.3.1 MPH Object Hierarchy

Figure 43 shows the core of the object model for the MPH framework, which will be expanded in future iterations. RePast provides a master “context” object which a builder inherits from in order to access all of the spaces, grids, data layers, and agents. So our MPHContext, which inherits from the RePast master context, lies at the root of our object model as well. The three principal classes directly feeding off of MPHContext (dependent on but not a child of MPHContext) are MPH Landscape, MPHDataContext, and SimpleAgent.

MPHLandscape encapsulates the specific characteristics of the landscape, such as realism and structure, as well as all of the data layers. The MPHDataContext holds all of the MPH parameters from the model definition GUI, and thus any custom data structures not tied directly to the RePast core spaces/grids, data layers and agents. The LandUses are one prime example of MPH objects tied to the MPHDataContext; all of the economic data structures are another. Lastly, the SimpleAgent class is the MPH base agent class and is registered as such in the master RePast context. The SimpleAgent class is then split into passive and active agents, the principal difference being active are decision-making, proactive agents while passive agents cannot make decisions and can only be reactionary at best. The current example of a passive agent is the Parcel class representing a land parcel. SimpleAgent contains properties and methods which can be used by all agents, whether passive or active, while ActiveAgent contains properties and methods applicable to active agents, such as a farmer.

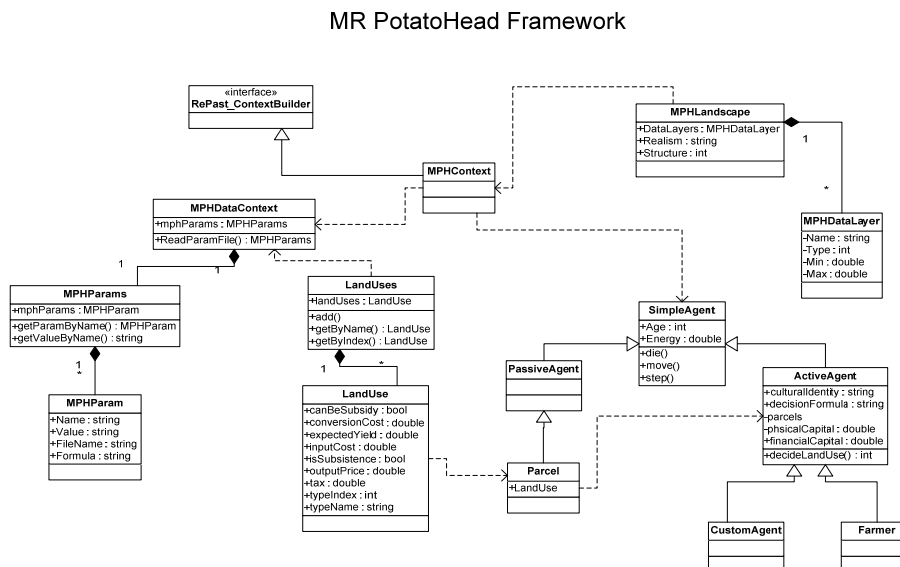


Figure 34: The object hierarchy of the MR PotatoHead Framework

2.3.2 Main Processing

The MPH simulation environment then loads the meta-model, builds the model to the specifications of the meta-model, and executes the model. When receiving the model parameters, the MPH has to first load all the MPH data fields, determine the order of precedence based on dependencies (e.g. if an economic data field uses a soil data field, soil data calculation must precede the former) as well as enabling/disabling specific events,

functions and properties. Note that sub-models, such as formulas for data layers or interfaces to other models, may use a different temporal scale and by their integration into the main model incorporate feedbacks and multi-scale effects into the model. Upon initial execution, the user may add additional output such as charts/plots and modify run-time initialization parameters (such as initial number of agents, etc.) before running the built model and analyzing the results.

2.3.3 Communicating Model Results

If the results of a model cannot be communicated effectively or clearly, the model is ultimately of little practical use. One way this is accomplished in MPH is capitalizing on the charting capability included in Repast Symphony. Another is through an option planned in a later phase to output the MPH model specifications into an ODD protocol-compliant format: Overview, Design concepts, and Details (ODD) protocol has been implemented in a number of agent-based modeling efforts as a model description and comparison tool (Polhill et al. [2008]).

2.3.3 Full Web-Enabling of Simulation Environment

Currently the Java MPH/RePast environment runs only locally. In a later phase, the ability will exist to run as an applet on the web for full web integration. This will allow more flexibility especially in team collaborations.

3. CONCLUSION

Before the MPH framework is released for public use, more extensive verification and validation needs to be performed. A series of existing models need to be implemented, and detailed comparison of results to the original model must be made. First, two simple models testing required core functional areas (such as Landscape and Agent sections) will be compared; currently the SLUDGE model (Parker et al. [2006]) is in this phase of verification. Next, the goal is for at least two models using a particular function will be validated before that section can be enabled. For example, until two models using Land Exchange can be validated, that area will be unavailable.

Once these validations have been performed and deemed reasonably consistent with the original model, then hopefully the MPH can start to be utilized by modeling teams and from these pioneering MPH modeling efforts a better understanding of the most appropriate future refinements can be gauged. Furthermore, as more collaborative models are implemented, the true usefulness of and degree of flexibility inherent in the framework can be ascertained.

REFERENCES

- Parker, D. C., S. M. Manson, M. A. Janssen, M. Hoffmann, and P. Deadman. 2003. Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review. *Annals of the Association of American Geographers* 93 (2), 2003. http://www.csiss.org/events/other/agent-based/papers/maslucc_overview.pdf.
- Parker, D., D. Brown, J. G. Polhill, S. M. Manson, and P. Deadman. Illustrating a new 'conceptual design pattern' for agent-based models and land use via five case studies: the MR POTATOHEAD framework. Pages 29-62 in A. L. Paredes and C. H. Iglesias, eds. *Agent-based Modelling in Natural Resource Management*. Universidad de Valladolid, Valladolid, Spain, 2006
- Parker, D., B. Entwisle, E. Moran, R. Rindfuss, L. V. Wey, S. Manson, L. Ahn, P. Deadman, T. Evans, M. Linderman, S. M. M. Rizi, and G. Malanson. Case studies, cross-site comparisons, and the challenge of generalization: Comparing agent-based models of land-use change in frontier regions. *Journal of Land Use Science* 3 (1): 41 –

72, 2008. <http://dx.doi.org/10.1080/17474230802048151>

[Polhill, J. G., D. Parker, D. Brown, and V. Grimm. 2008. Using the ODD Protocol for Describing Three Agent-Based Social Simulation Models of Land-Use Change. *Journal of Artificial Societies and Social Simulation* 11 \(2-3\). <http://jasss.soc.surrey.ac.uk/11/2/3.html>.](#)

University of Chicago. 2000. Repast (Recursive Porous Agent Simulation Toolkit). <http://repast.sourceforge.net>

Formatted: Indent: Left: 0.51 cm