

Identification and use of Land-Use Modelling Primitives 'LUMPS'

Derek T. Robinson,
Daniel G. Brown,
Rick Riolo, and
William Rand

Team SLUCE @ University of Michigan

Background

- The SLUCE team collaborated with Repast and ESRI to assist with theoretical developments that would facilitate ABM creation by GIS users.
- GUI, primitives, relationships between ABM & GIS

Goal of Workshop Session

- LUCC modelling is difficult and time expensive
- beyond the time-frame and skill set available to students.
- identify generalized but fundamental building blocks useful for a range of ABMs in the LUCC field.
- In this session we will try to come with a clear description of what these LUMPs are and how we can make use of their generic structures in our modeling.

Outline

- Primitive Space
- Land-use modelling primitives (LUMPs)
- The focus of existing platforms
- Overview of 3 platforms
- Constraining our LUMP discussions
- Today's direction...

Primitives

- With respect to modelling primitives can be defined a wide number of ways.
- A basic building block that can be used by others who do not require knowledge of the internal implementation structure.

Primitive Space

Coding Constructs



High Level
Components

- Data Types (e.g. string, int, etc.)
- Platform Provided (e.g. getMooreNeighbours)

- Decision-making strategies
- Empirically-based Algorithms

Primitive Space

Coding Constructs



High Level
Components

High



Low

Flexibility

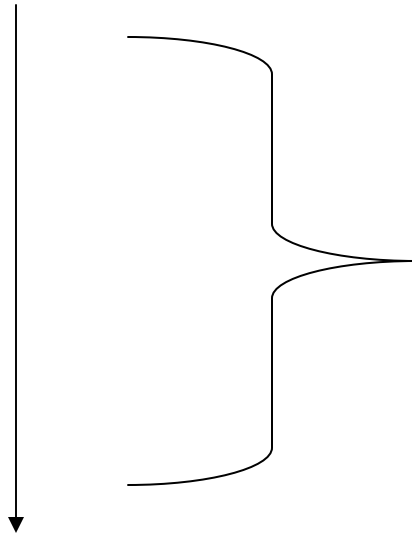
**Programming
Requirements**

High-level Components

- Produce complex agent behaviours
 - LUCITA – Heuristic decision-making strategy
 - SLUCE – Residential location utility function
- ***VERY*** useful, however...
 - they are often complex
 - require extensive analysis of
 - Empirical backing, or
 - Validation
- May be too time consuming!

Primitive Space

Coding Constructs



Land-use Modelling Primitives

High Level
Components

Land-use Modelling Primitives

- smallest modelling unit
- simple computations that complete a simple process
- Example
 - `search_agents(radius)`
 - `select_agent(value, agentList)`
 - `change_agent(attribute, value)`

LUMPs

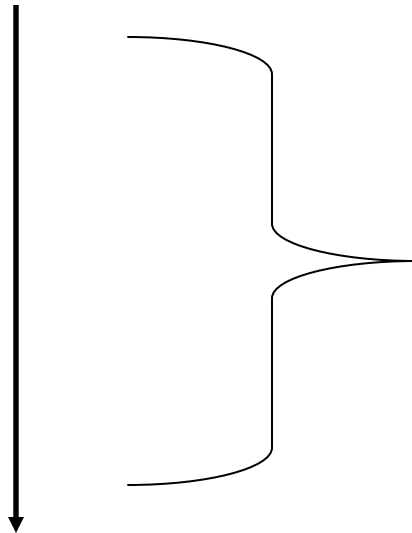
- Examples
 - rest/wait,
 - move,
 - change,
 - measure,
 - compare,
 - influence, exchange, or destroy

LUMPs

- One level above available coding and platform constructs
- Focus on verification rather than validation.
- Communication is enhanced.

Primitive Space

Coding Constructs

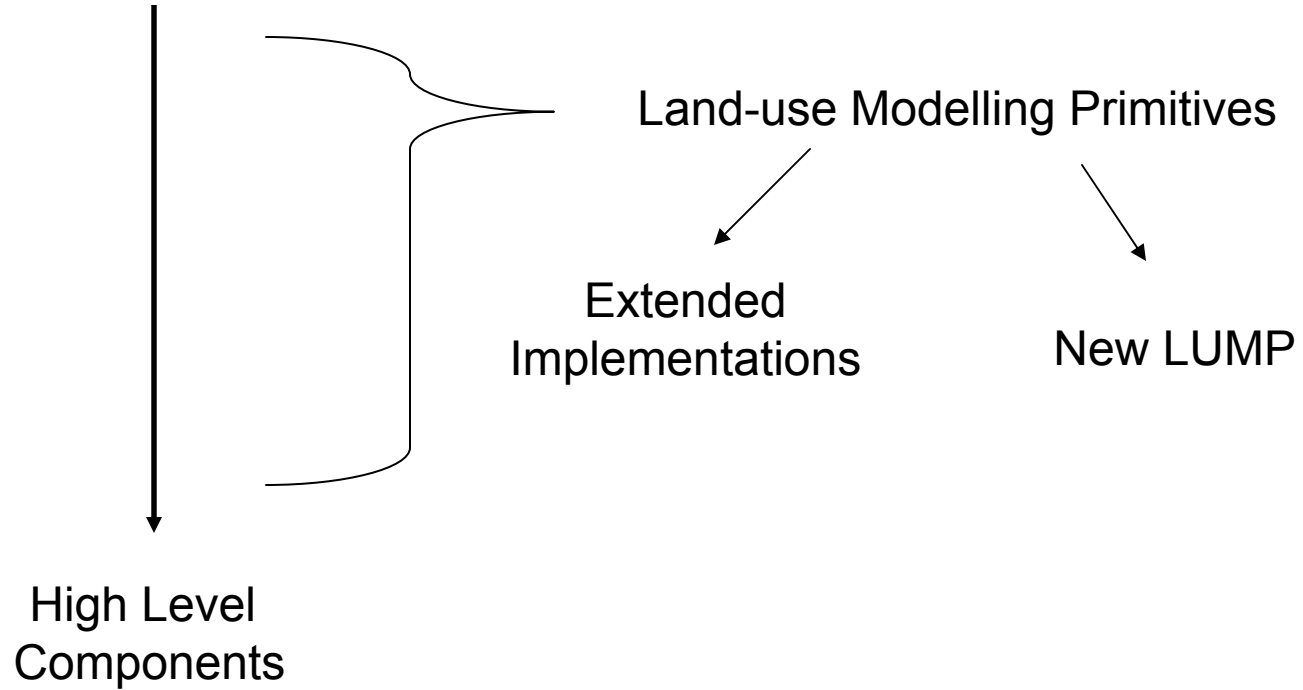


Land-use Modelling Primitives

High Level
Components

Primitive Space

Coding Constructs



Change code...

Find Nearest Neighbour

```
#set the initial min distance to the vision
```

```
min = self.vision+1
```

```
self.nearestNeighborDistance = self.vision+1
```

```
self.nearestNeighborHeading = self.heading
```

```
#loop through the agents and find the nearest one
```

```
for agent as VectorAgent in self.model.getVectorAgentGroup():
```

```
    if (not self.equals(agent)):
```

```
        if (self.the_geom.distance(agent.getThe_geom()) < min):
```

```
            min = self.the_geom.distance(agent.getThe_geom())
```

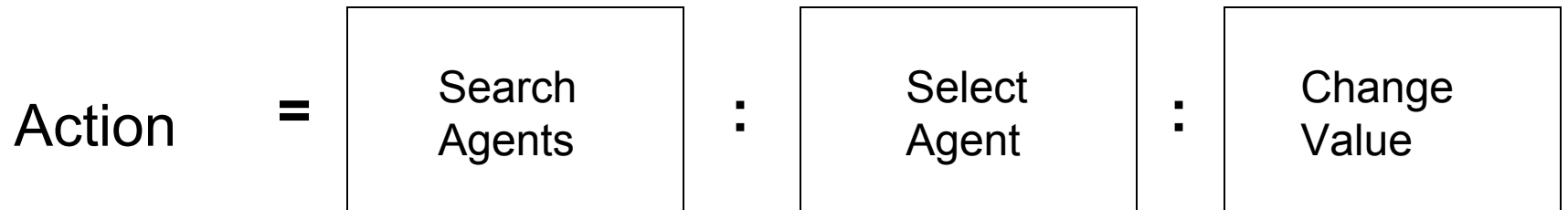
```
            self.nearestNeighborDistance = min
```

```
            self.nearestNeighborHeading = agent.getHeading()
```

to this...

Search
Agents

to this...



Combine to create more complex behaviours

Assembly of LUMPs

- has a new meaning
- may be used on its own or by other assemblies of LUMPs
- use of simple LUMPs can unite simple prototype models with the greater realism and complexity often captured in complex models

What have others done?

- We surveyed NetLogo, RepastPy, and Mobidyc ABM software/libraries
- NetLogo: Focus on GUI development
- RepastPy: Model Creation
- Mobidyc: User Primitive Creation

NetLogo

The screenshot displays the NetLogo environment with the following components:

- Title Bar:** some3 - NetLogo [T:\backup\PHD\sluce\netlogomodels]
- Menu Bar:** File Edit Tools Zoom Tabs Help
- Interface:** Interface Information Procedures
- Control Panel:** Edit Delete, abc Button, abc 5 Slider, On Off Switch, abc 5 Chooser, abc 5 Monitor, abc Plot, abc 4 Output, abc def ghi jkl Text
- Simulation Area:** A green field with a yellow circular boundary. The interior contains a grid of small grey and red dots.
- Sliders and Buttons:** setup, go, go once, smoothness (1.0), Space-Quality-Tradeoff (1.0), numtests (1.5), On Off (labeled ltab7), radius (40)
- Plot:** A graph titled "Development Beyond Radius" with "# cells" on the y-axis and "time" on the x-axis. The curve shows an exponential-like growth from 0 to 100.
- Command Center:** A text area at the bottom with the prompt "observer>".

NetLogo

```
some3 - NetLogo {T:\backup\PHD\sluce\netlogomodels}
File Edit Tools Zoom Tabs Help
Interface Information Procedures
Find... | Check | Procedures
breeds [resident service circle]
resident-own [alphaq alphas utility]
patches-own [quality sddist]
globals [counter selx selx residentsperstep residentsperservice]

;; THIS IS THE SETUP STEP

;;----- Calls processes that set up the initial service center and the patches.
to setup
  ca
  setup-service
  setup-patches
  set residentsperstep 10
  set residentsperservice 100
  set-default-shape circle "box"
  ;create-circle 2000
  ;ask circle
  ; [ set heading random-int-or-float 360
  ;;   set color yellow
  ;   fd radius ]
end

;;----- Create initial service center in the middle
to setup-service
  create-service 1
```

NetLogo

Primitives Dictionary

NetLogo 3.0 User Manual

Alphabetical: [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [?](#)

observers, and vice versa. To see which agent (turtles, patches, observers) can actually run each command, consult each individual entry in the dictionary.

Turtle-related

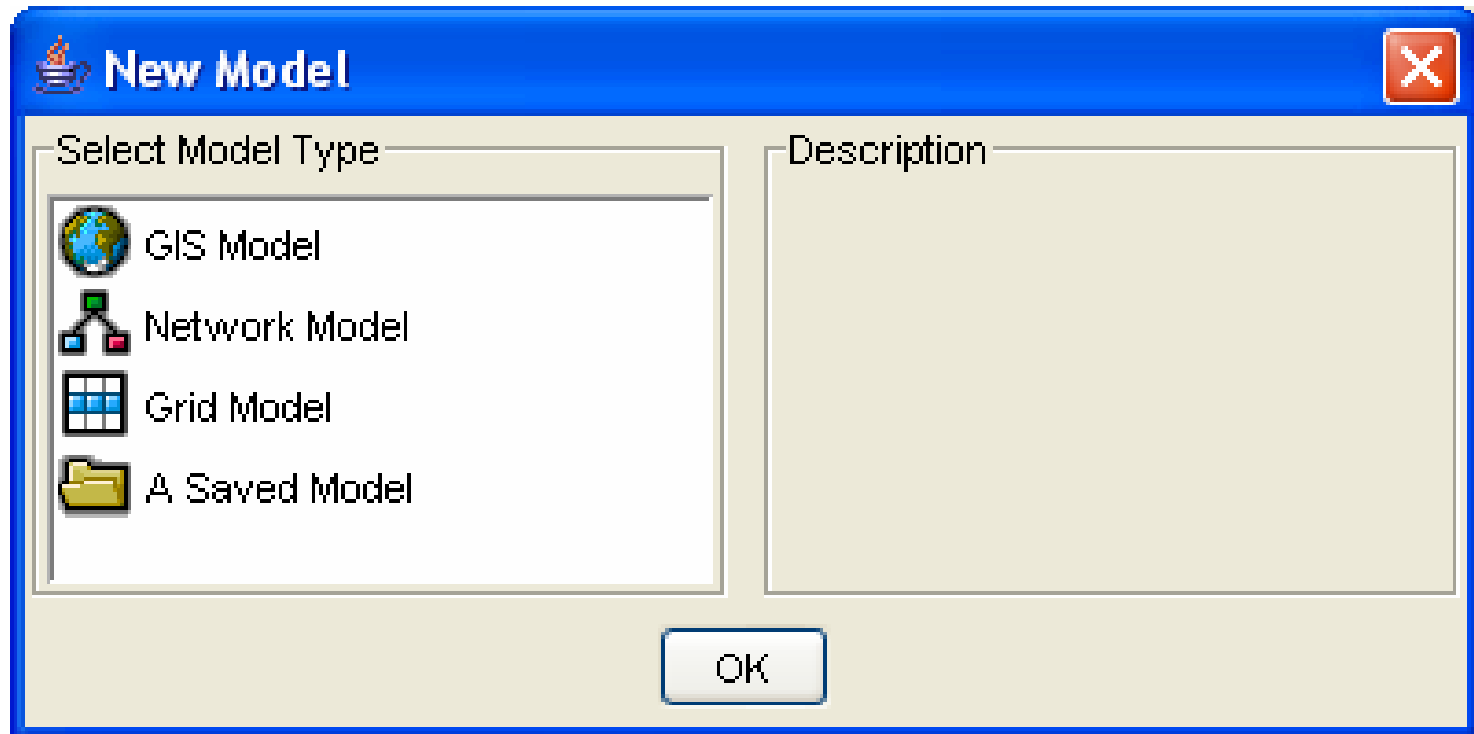
[back](#) (bk) [BREED-at](#) [BREED-here](#) [BREED-on](#) [clear-turtles](#) (ct) [create-BREED](#) [create-custom-BREED](#) [create-custom-turtles](#) (cct) [create-turtles](#) (ct) [die](#) [distance](#) [distance-nowrap](#) [distancexy](#) [distancexy-nowrap](#) [downhill](#) [downhill4](#) [dx](#) [dy](#) [face](#) [face-nowrap](#) [facexy](#) [facexy-nowrap](#) [forward](#) (fd) [hatch](#) [hatch-BREED](#) [hideturtle](#) (ht) [home](#) [inspect](#) [is-turtle?](#) [jump](#) [left](#) (lt) [myself](#) [no-label](#) [nobody](#) [-of](#) [other-turtles-here](#) [other-BREED-here](#) [patch-ahead](#) [patch-at-heading-and-distance](#) [patch-here](#) [patch-left-and-ahead](#) [patch-right-and-ahead](#) [pen-down](#) (pd) [pen-erase](#) (pe) [pen-up](#) (pu) [right](#) (rt) [self](#) [set-default-shape](#) [setxy](#) [shapes](#) [showturtle](#) (st) [sprout](#) [sprout-BREED](#) [stamp](#) [subject](#) [subtract-headings](#) [towards](#) [towards-nowrap](#) [towardsxy](#) [towardsxy-nowrap](#) [turtle](#) [turtles](#) [turtles-at](#) [turtles-from](#) [turtles-here](#) [turtles-on](#) [turtles-own](#) [uphill](#) [value-from](#)

Patch-related primitives

[clear-patches](#) (cp) [diffuse](#) [diffuse4](#) [distance](#) [distance-nowrap](#) [distancexy](#) [distancexy-nowrap](#) [import-pcolors](#) [inspect](#) [is-patch?](#) [myself](#) [neighbors](#) [neighbors4](#) [no-label](#) [nobody](#) [nsum](#) [nsum4](#) [-of](#) [patch](#) [patch-at](#) [patch-ahead](#) [patch-at-heading-and-distance](#) [patch-here](#) [patch-left-and-ahead](#) [patch-right-and-ahead](#) [patches](#) [patches-from](#) [patches-own](#) [self](#) [sprout](#) [sprout-BREED](#) [subject](#) [value-from](#)

Agentset primitives

RepastPy



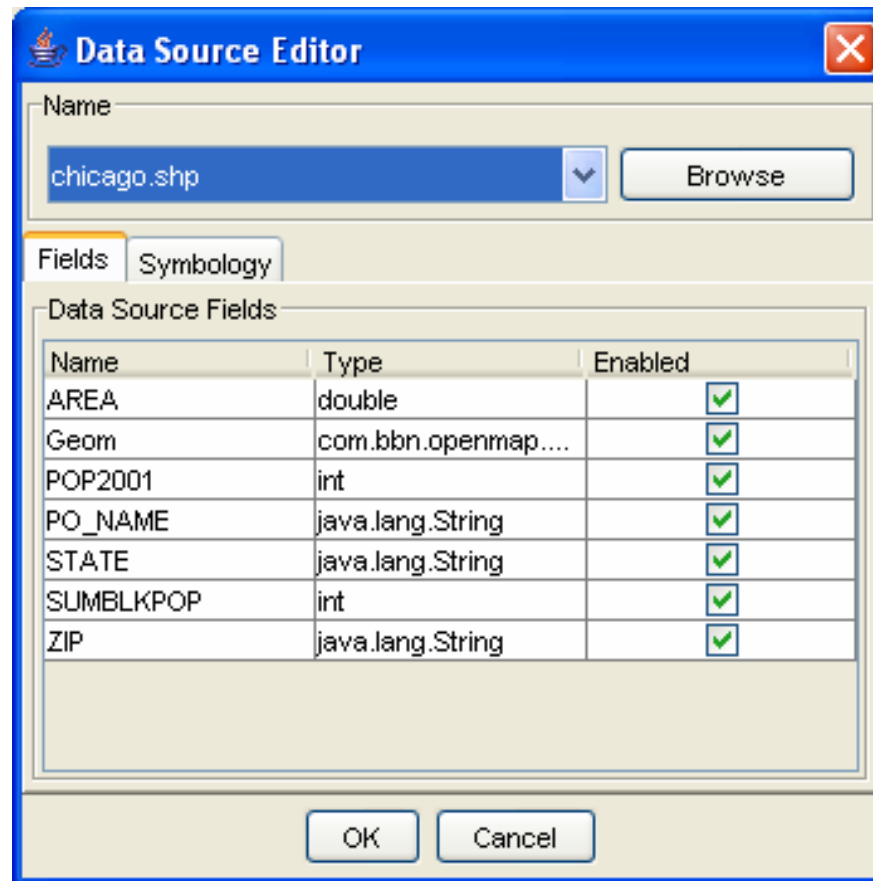
RepastPy

The screenshot displays the Schelling GIS software interface. The title bar reads "Schelling GIS [C:\Program Files\repastpy\projects\schelling_gis\schelling_gis.sbp] - R...". The menu bar includes "File", "Model", and "Help". The toolbar contains icons for file operations (New, Open, Save, Print, Help), navigation (Cursor, Move, Copy, Paste), and GIS functions (Map, Edit, Pan, Zoom). The left sidebar shows a tree view with "Environment" expanded to "Schelling GIS", which contains "ZipRegion" and "Resident". The main area is the Properties window for the selected model, showing the following details:

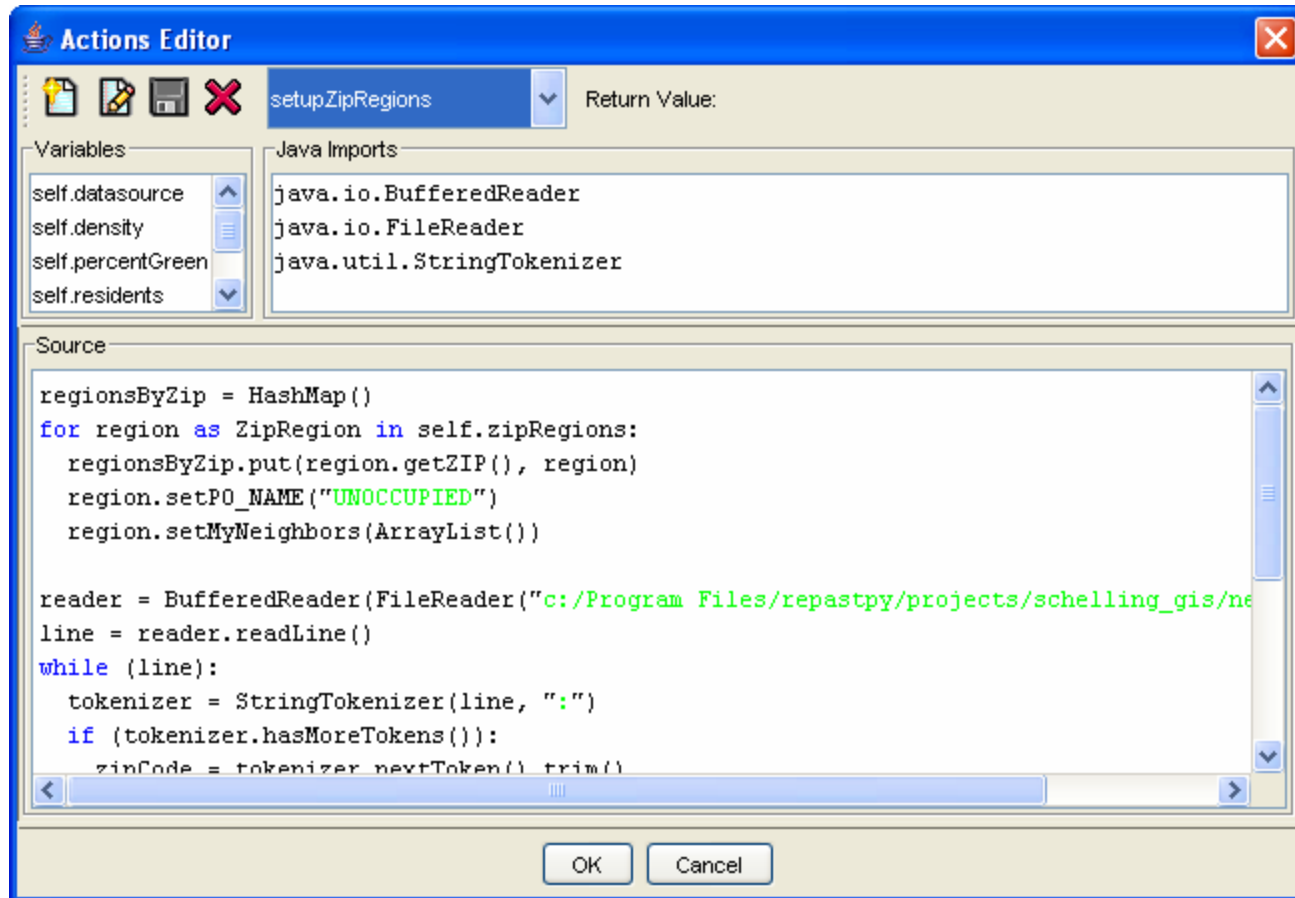
Property	Value
Actions	<input type="button" value="Edit"/> 5 action(s)
Display Name	Schelling GIS
GIS Package	Open Map <input type="button" value="v"/>
Master Schedule	<input type="button" value="Edit"/>
Model Name	GISModel
Fields	<input type="button" value="Edit"/> 4 field(s)
Schedule	<input type="button" value="Edit"/> 3 schedule item(s)

At the bottom of the Properties window, there are tabs for "Properties" (selected) and "XML".

RepastPy



RepastPy



RepastPy

The image shows a screenshot of the "Schedule Editor" dialog box in RepastPy. The dialog has a blue title bar with a close button (X) in the top right corner. The main area is divided into two sections: "Edit Action Execution" and "Action Executions".

Edit Action Execution

Actions:

Execution: Tick:

Execute Last:

Action Executions

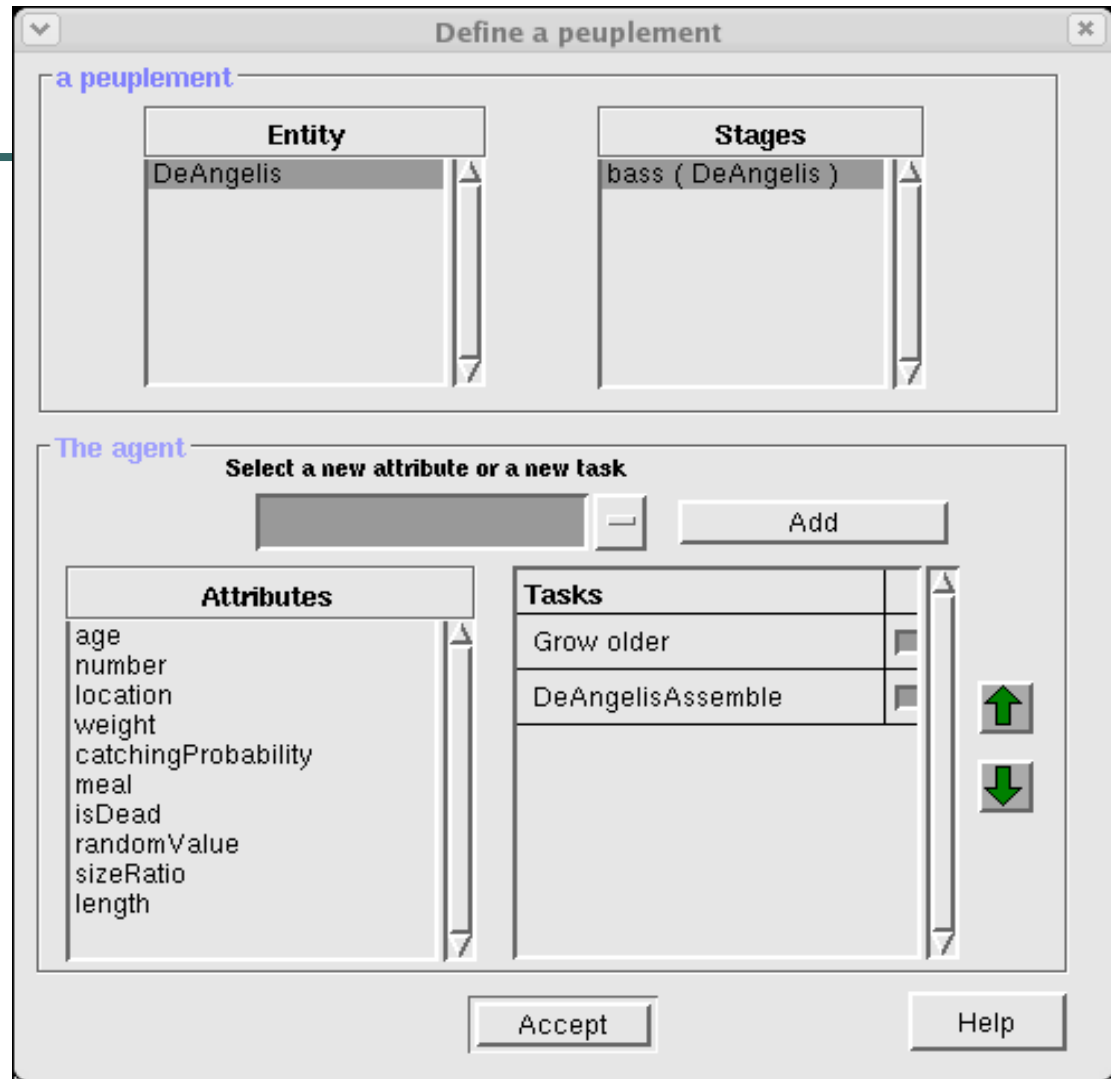
Action	Execution Type	Tick	Last
step	EVERY_TICK	1	<input type="checkbox"/>

Mobidyc

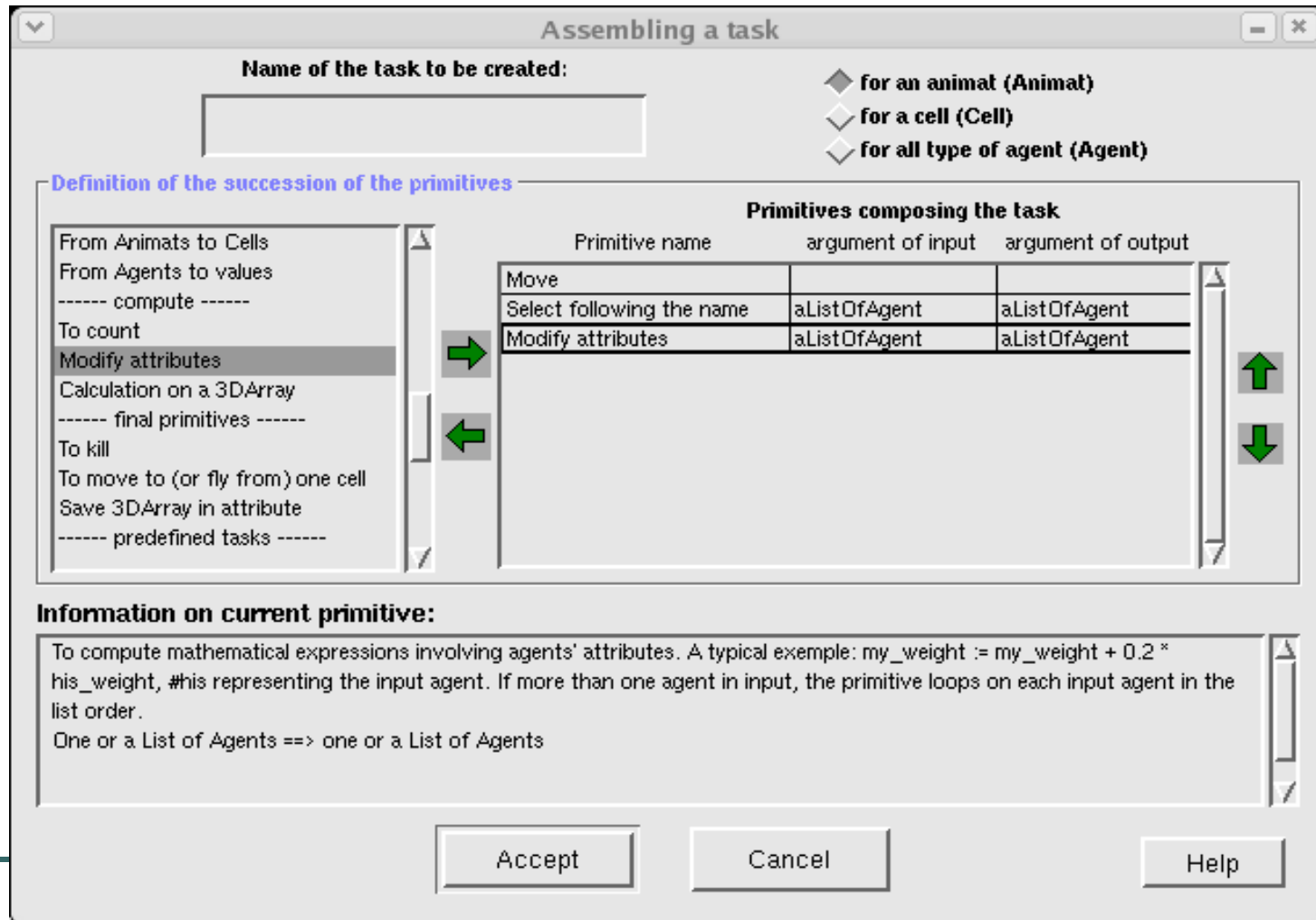
The screenshot displays the Mobidyc software interface for a project named 'project1'. The window title is 'Mobidyc [project1]'. The menu bar includes 'Program', 'Tools', 'Space', 'Templates', 'Tasks', 'Options', and 'Help'. Below the menu bar are several icons representing different simulation components. The main interface is divided into several sections:

- Control Panel:** Contains 'Launch' and 'Continue' buttons. Below them, the text 'Agents not located in space' is displayed. A blue progress bar shows '100%' completion.
- Time units:** A section with a blue header. It includes two rows of controls: 'time step' with a value of '1.00' and unit 'day', and 'simulation' with a value of 'ex' and unit 'week'. Each row has a minus button to the right.
- Batch Configuration:** A section with a blue header. It features a 'Start batch' button, a 'Simulation experiment:' label above an empty text box with a minus button, and two columns of numeric input fields: 'nb. of Exper.' (two fields, both containing '0') and 'nb. of replicats' (two fields, both containing '0'). At the bottom, there are 'Backup file:' and 'nb. max animats' labels above their respective text input fields, with the latter containing the value '5000'.

Mobidyc



Mobidyc



Mobidyc

Create a task

Name of the task to be created:

for an animat (Animat)
 for a cell (Cell)
 for all type of agent (Agent)

Source code of the method executer :

```
executer  
  "commentaires"  
  
  | "noms des variables temporaires" |  
  "texte de la méthode"
```

Choice of category :

- etat du monde
- histoire du monde
- le monde
- temps ecoule depuis le debut de la simulation

Mobidyc

The screenshot shows the 'Scheduler' window in Mobidyc. It features a 'Synchronisation' section with radio buttons for 'agents' and 'cells', each with 'synchronous' and 'asynchronous' options. A yellow box highlights the 'To define the backup...' button. Below is a 'List of the operations' section with instructions and a table of operations.

Synchronisation

agents cells

synchronous

asynchronous

To define the backup... Help

List of the operations

Use arrows to modify their order

Check the box not to carry out the operation

	<input type="checkbox"/>
run Cells	<input type="checkbox"/>
run Animats	<input type="checkbox"/>
save Cells	<input type="checkbox"/>
save Animats	<input type="checkbox"/>
run Space Display	<input type="checkbox"/>

Up arrow and Down arrow

6 Focal Areas of ABM

- Agents
- Groups
- Environment
- Reporting/Output
- Experiments
- Interoperability

6 Focal Areas of ABM

- Agents
- Groups
- Environment - CA
- Reporting/Output -
- Experiments
- Interoperability

What we need to do...

- Take the ideas from these different platforms and use them as a base for discussion and development of

“Land-Use Modelling Primitives”

What we need to do...

- Focus on Agents, Groups, & Interaction with the Environment
- Ignore Implementation
- Think Simple and Build
- What primitives would your ideal Lucc library contain?
- Are there any hooks that your model would require to work with a specific toolkit or data collection technique?

Summary

- Primitive Space
- LUMPs
- Focus on agents and their interactions