A universal simulator for ecological models

Niels Holst
Aarhus University
Denmark
niels.holst@agrsci.dk


A probable scenario for future ecological modelling is that standardized objects for at least some components of ecosystems will be developed and widely distributed in a form that can be used in many different models. If

In a sense the use of standard objects would bring theoretical ecologists closer to experimentalists; when a laboratory scientist needs a mouse or a population of *E. coli* for an experiment, he doesn’t develop his organisms from scratch. Instead he gets standard animals from a recognized supplier. There is no basic reason why the developer of ecosystem models should not eventually do the same.
Successful model software

- Systems Biology Markup Language (SMBL)  
  *Hucka et al. (2003)*
- NetLogo for individual-based modelling  
  *Tisue and Wilensky (2004)*
- R: A language and environment for statistical computing  
  *R Development Core Team (2012)*

Universal Simulator classes

```
Simulation       Component
  Factory        Model
     Insect     Et cetera
     Plant      Table
```

Plot
Populating the environment

```cpp
Factory::create(recipe) {
    createComponents();
    for each component[i] do component[i]->amend();
}
```
Universal Simulator objects

Simulation::run() {
    for each component[i] do component[i]->initialise();
    for each iteration do {
        for each component[i] do component[i]->reset();
        for each time step do {
            for each component[i] do component[i]->update();
        }
        for each component[i] do component[i]->cleanup();
    }
    for each component[i] do component[i]->debrief();
}

Running the simulation
Running the simulation

```cpp
Simulation::run() {
    for each component[i] do component[i]->initialise();
    for each iteration do {
        for each component[i] do component[i]->reset();
        for each time step do {
            for each component[i] do component[i]->update();
        }
        for each component[i] do component[i]->cleanup();
    }
    for each component[i] do component[i]->debrief();
}
```

Recursive methods

```
Simulation meadow
```

```
Plant nettle
```

```
Stage egg
Stage larva
Stage pupa
Stage imago
Stage age
Organ leaves
```

```cpp
update to
```
Recursive methods

- Stage egg
- Stage larva
- Stage pupa
- Stage imago
- Stage age
- Organ leaves

Simulation meadow

Insect io

Plant nettle
Recursive methods

Insect io
- Stage egg
- Stage larva
- Stage pupa
- Stage imago

Simulation meadow
- Plant nettle
- Stage age
- Organ leaves

Running the simulation

```
Simulation::run() {
    for each component[i] do component[i]->initialise();
    for each iteration do {
        for each component[i] do component[i]->reset();
        for each time step do {
            for each component[i] do component[i]->update();
        }
        for each component[i] do component[i]->cleanup();
    }
    for each component[i] do component[i]->debrief();
}
```
The R environment

Universal Simulator environment
Universal Simulator resources

- www.ecolmod.org
  - Regular web-based PhD course (10 ECTS)
  - www.github.com/NielsHolst