What Program language should this be written in?

I have come across a range of different languages used for IBM’s. There is netlogo <https://ccl.northwestern.edu/netlogo/>, parcel <https://github.com/OceanParcels/parcels> and the C++ <https://github.com/trophia/sna1>.

So which language should we go with?

I feel C++ would be the most commonly known language to fishery scientists so could be the most useful for future adaptation and collaboration.

What behaviour should we apply to each agent?

**Growth**

-growth explicitly (von bertalanffy) length is determined by some function of time or age

-Model growth as a consequence of its environment (bio-energetics model), food/energy availability, life stage

**Movement**

-Box-transfer movement – Markovian

-Advection-diffusion movement

-Ontogenetic movement spawning migrations

**Reproduction**

-Find a mate near by?

-Population level sum up the Mature biomass over all individuals and calculate number of eggs based on some Stock recruitment function

**Mortality**

-Natural Mortality

-Fishing Mortality?

What Flexible should the spatial and temporal resolution of the model be?

I say completely user defined so people could model daily temporal scales or others could look at seasonal temporal scales

Temporal model

The model consists of

1. Annual steps
   1. User defined time steps within a year e.g. day, season, annual timestep

Defining processes in a time step, each time step is responsible for sorting the processes that occur in each time step. This makes processes the key dynamics. An example of this is say we have an annual time step with the following processes

* M
* F
* Growth
* Recruitment

We would write process code for all of these processes and each year they would get called and iterate over the partition and executing there functionality. The problem with this it is not as fast as hard-coded examples which usually put processes in the individual class iterate over the partition and call a pre-defined set of processes.

So I can think of 2 methods for dealing with this

1. Each process is written out as a separate class and gets passed the partition by reference to alter as the process algorithm says. This could be threaded, you could block up the partition and sent each block to a different thread. This would make it nicely modular-so others can add their own processes. Negative is you would need to give a lot of access from agents to the processes classes
2. Individuals have processes within their own class and are executed in a pre-defined order. The time step manager would then be essentially passing bool variables whether that process is being executed in this time step.

I like the 1) option but both should be explored, the first will have tidier code but may come at a speed cost.

Spatial component

We want a user defined environment, I think something like what SPM does, where the user defines a grid of cells and defines reporting definitions but can also be queried by a process to ask where an individual is and where it wants to go based on the individual’s characteristics and the surrounding environment.

Partition structure?

Partition

Egg

Undeveloped

larvae

sub developed

Juvenile-Adult

developed

The reason we want to split the partition over some discrete entities is so we can parallelise the code easier, it is better to have independent components. When I talk about independent components I mean I process acting on one individual does need to reference another. Why I am thinking of not splitting out Juvenile-Adult is that fisheries will catch both of these so will need access to both these categories.

Time Step Manager

Environment

Process

Derived Quantities

individual

Partition

individual

individual

Random number generator

Parameters

Reports

Selectivity’s

A conceptual diagram of how objects in the code will interact, the bottom objects will be global classes, or connected to in the doBuild phase.