Getting started with Evolvix

Laurence Loewe
Laboratory of Genetics and Wisconsin Institute for Discovery
University of Wisconsin-Madison

WISCONSIN INSTITUTE FOR DISCOVERY

What You Know...

Informal Predator-Prey Model

Your Question: Can we observe simple predator-prey cycles?

Hares breeding: **Hares** —1.5→ **2 Hares**

Lynx eating: **Hares + Lynx** —0.000225→ **Lynx** (slightly fatter)

Lynx breeding: **Hares + Lynx** —0.000045→ **2 Lynx**

Lynx dying: **Lynx** —0.225→ **0 Lynx**

CC-BY-SA http://en.wikipedia.org/wiki/File:Canadian_lynx_by_Keith_Williams.jpg
CC-BY-SA https://en.wikipedia.org/wiki/File:Snowshoe_Hare_-_Shirleys_Bay.jpg

Syntax Ideas: A ⟶ B

1. **Arrows:** `Action { A --[Rate=50]--> B }`
2. **Brackets:** `{ Stuff belonging together }`
3. **Comments:** `// One-line comment`

We keep the arrow! The code shown is valid Evolvix.
Length of arrows doesn't matter; add dashes for looks.

Curly brackets {} are like 'begin' and 'end' statements: they enclose things that belong together in a clause.

`/!! <-start of multi-line comment and its end-> !!/`

Using Evolvix ...

To investigate some aspect of our world in Evolvix, we define an **Evolvix Quest**. It contains all relevant **Models** that describe our study system, all **Queries** that describe what types of results we are interested in, and all **Tasks** that describe how the Models need to be analyzed to answer queries.

A **Model** describes all **Parts** we judge to be relevant for our study and all **Actions** that change these Parts. Simulations running in Tasks also need to know how often Actions occur and initial amounts of all Parts.

Example: In our predator-prey model the Actions are life-history events like birth, feeding and death. The Parts are 'Hares' and 'Lynx'.

As experienced modellers know, good models are built for a purpose, usually to answer a question. This question provides focus to the modeling and helps to cut irrelevant details (e.g. we do not model snowflakes in our example above). To encourage awareness of that key question, Evolvix will demand that a Quest states at least one **Question**. It is easy to document much, much more using so called **InfoBlocks**.

Simplified Evolvix. To make it easier to get started, Evolvix can define a global Model, Query and Task automatically by using the simplified syntax on this poster. A more detailed syntax is being developed for more complicated Quests.

Purpose of Evolvix: Make Modeling Easy: (i) easy to describe a model, (ii) easy to say what needs to be analyzed in a model, (iii) easy to switch between different methods of analysis, (iv) easy to process results, (v) and easy to document it all since more time is spent reading code than writing it.

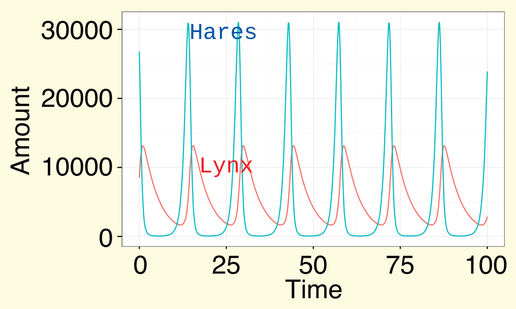
... turns into computable code

```
Evolvix Quest HareLynxDynamics  {
    Question : "Can we see simple predator prey cycles?"
}

Initial Amount of Lynx    = 8500  ["Individuals/MyArea"]
Initial Amount of Hares   = 26800 // Units are optional

Action  1  Hares_Breeding {
    Hares              -----[ Rate = 1.5       ]-> 2     Hares
}
Action  2  Lynx_Eating {
    Hares  +  Lynx  -----[ Rate = 0.000225 ]----->   Lynx
}
Action  3  Lynx_Breeding {
    Hares  +  Lynx  -----[ Rate = 0.000045 ]-----> 2 Lynx
}
Action  4  Lynx_Dying {
       Lynx  -----[ Rate = 0.225    ]-----> 0 Lynx
}
```

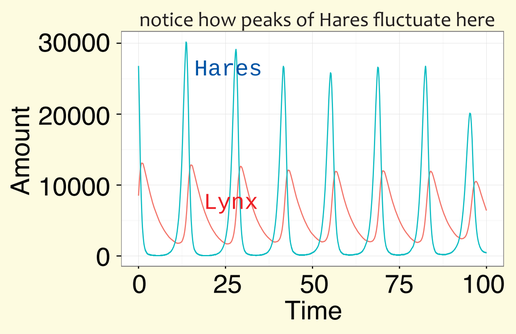
/!! "Question" is just one type of **InfoBlock**. *InfoBlocks offer many more features for documenting all aspects of Models, Queries and Tasks. !!/*

/!! One global **Model** *defined by listing all Actions and specifying the initial amounts of Parts. !!/*

```
Simulate
deterministically
until
100 ["years"]
```




```
Simulate
stochastically
until
100 ["years"]
```



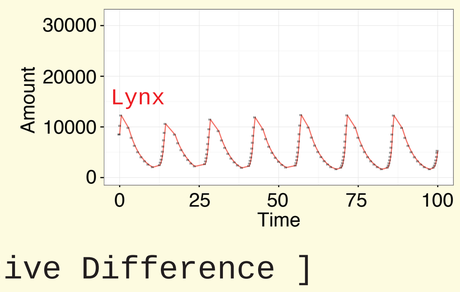
/!! "Simulate" defines a **Task** *that simulates some Model using some method until some time to answer some Query (e.g. to get a timeseries).*

Deterministic simulations often compute expected averages.

Stochastic simulations compute the random behavior that is caused by the individual nature of Parts. !!/

// One global default **Query**: *record a timeseries for each part, unless timeseries are defined explicitly.*

```
TimeSeries Predator {      Report
    Separately the Amount of Part    Lynx
    From             0    ["years"]
    Until            100  ["years"]
    Report whenever values change by 20 [ % Relative Difference ]
}
```



Thanks for helpful discussions of this poster to David Andersion, Seth Keel, Kurt Ehlert, Iratxo Flores-Lorca, Kate Scheuer and Jacob Goldfinger. Evolvix Version 0.1 http://evolvix.org/ Poster: v0r9c1 2013-12-09