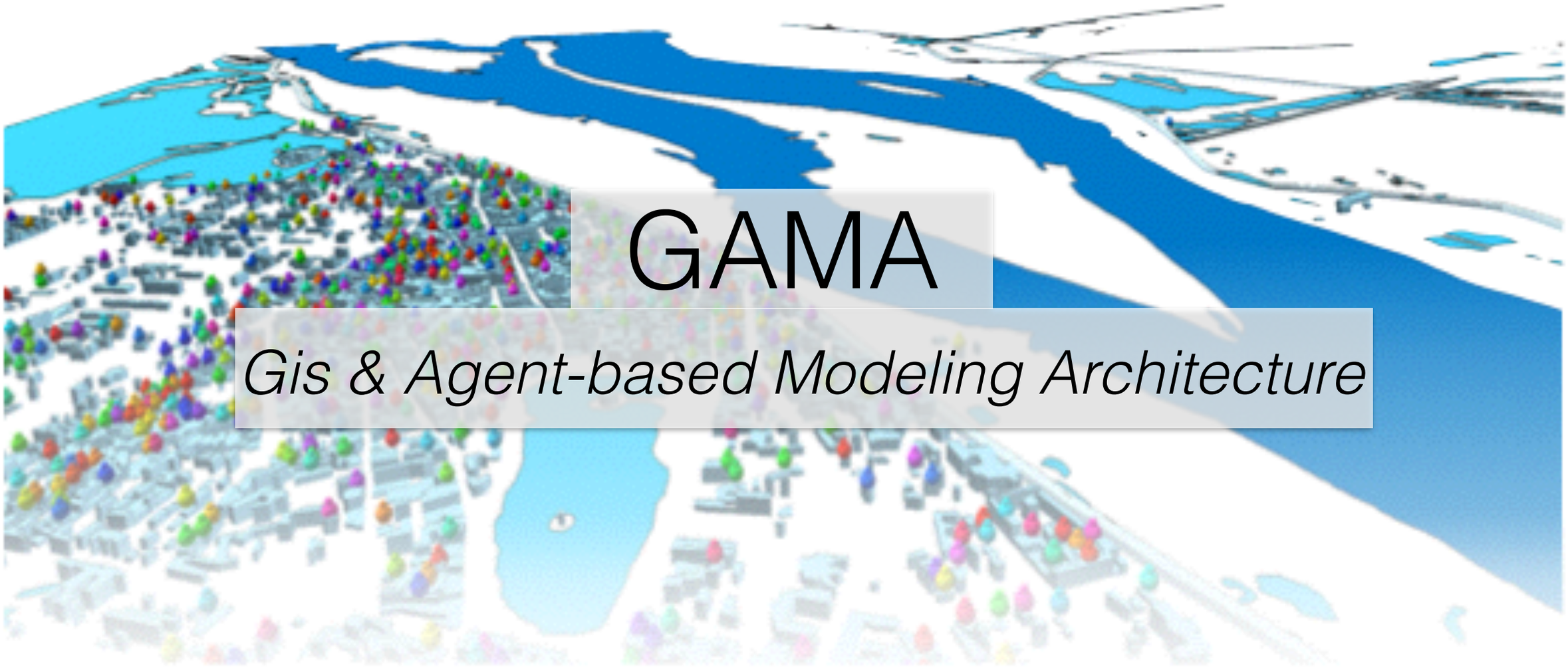
Simulation Models for Economics

A.Y. 2014/2015



GAMA

Gis & Agent-based Modeling Architecture

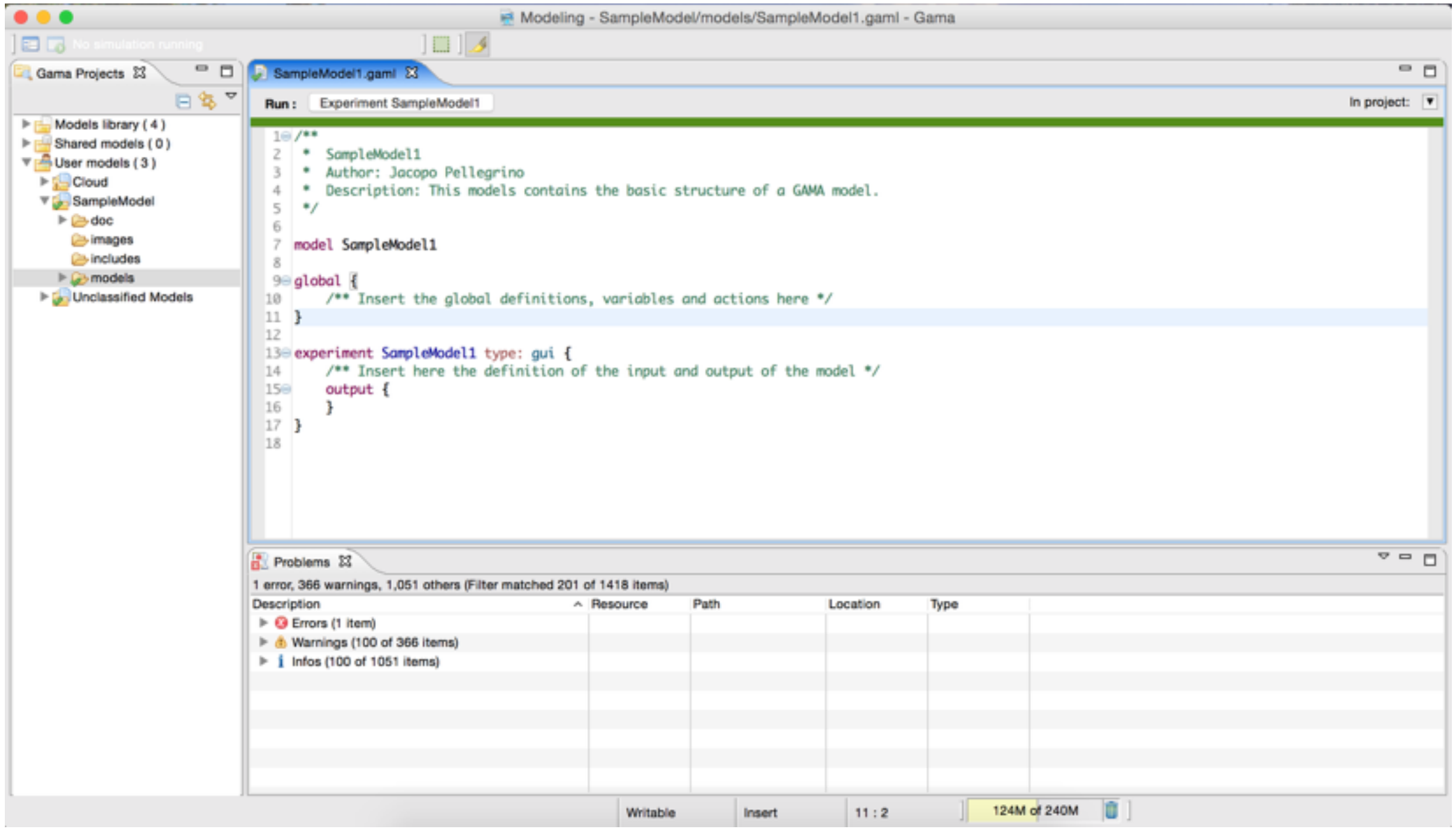
Dr. Jacopo Pellegrino

UNIVERSITÀ
DEGLI STUDI
DI TORINO

Introduction

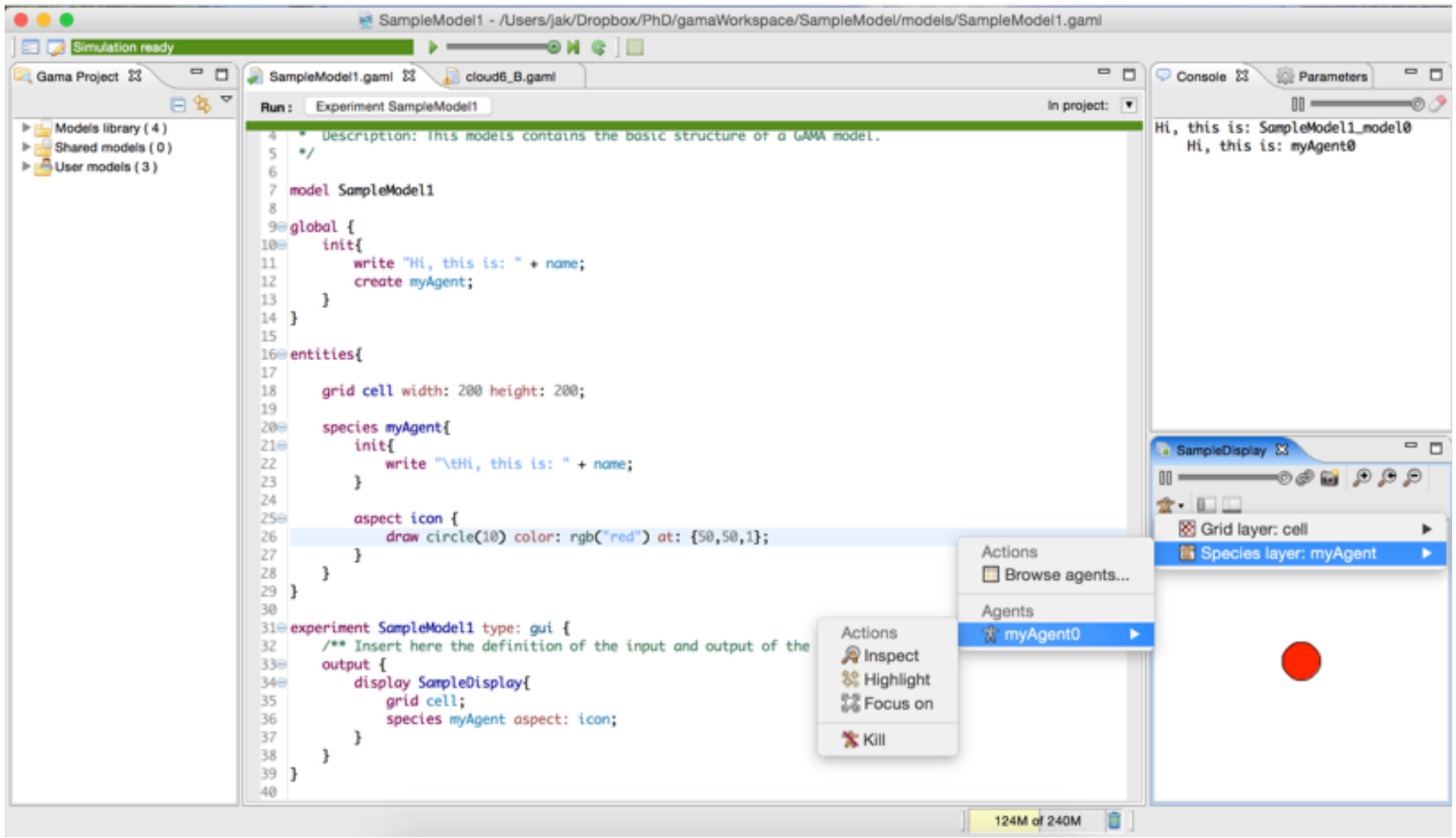
- Gis & Agent-based Modeling Architecture
- Agent-based, spatially explicit, modeling and simulation platform.
- Free and Open Source tool (GAMA webpage).
- Initially developed as an Eclipse plug-in, now is an independent tool.
- GAML (Gis & Agent-based Modeling Language) agent-oriented language, close to Java.
- Instantiation of agents from any kind of dataset, including Gis data (e.g.: road traffic model).

Features



Modeling perspective

Features



Simulation perspective

Modeling Introduction

Model file made up of three main parts:

- `global`
  - Variables, constants that must be **accessible to every agent at any time** during the simulation.
  - Initialization, reflexes, actions that are **common to every species** in the model.
- `entities`
  - Definition of spatial environment (`grid`).
  - Definition of `species` and their **skills**, **actions** and **reflexes**.
- `experiment`
  - Possibility to display world and agents.
  - Several options for plotting data (charts, histograms, etc.).

Species Relationship

Species can be related to each other:

- **Nesting**: a species can be defined within another one. The enclosing one is referred as *macro species*, the enclosed on as *micro species*.

- **Inheritance**: a *child* species extends behavior from the *parent*, close to what happens in Java.

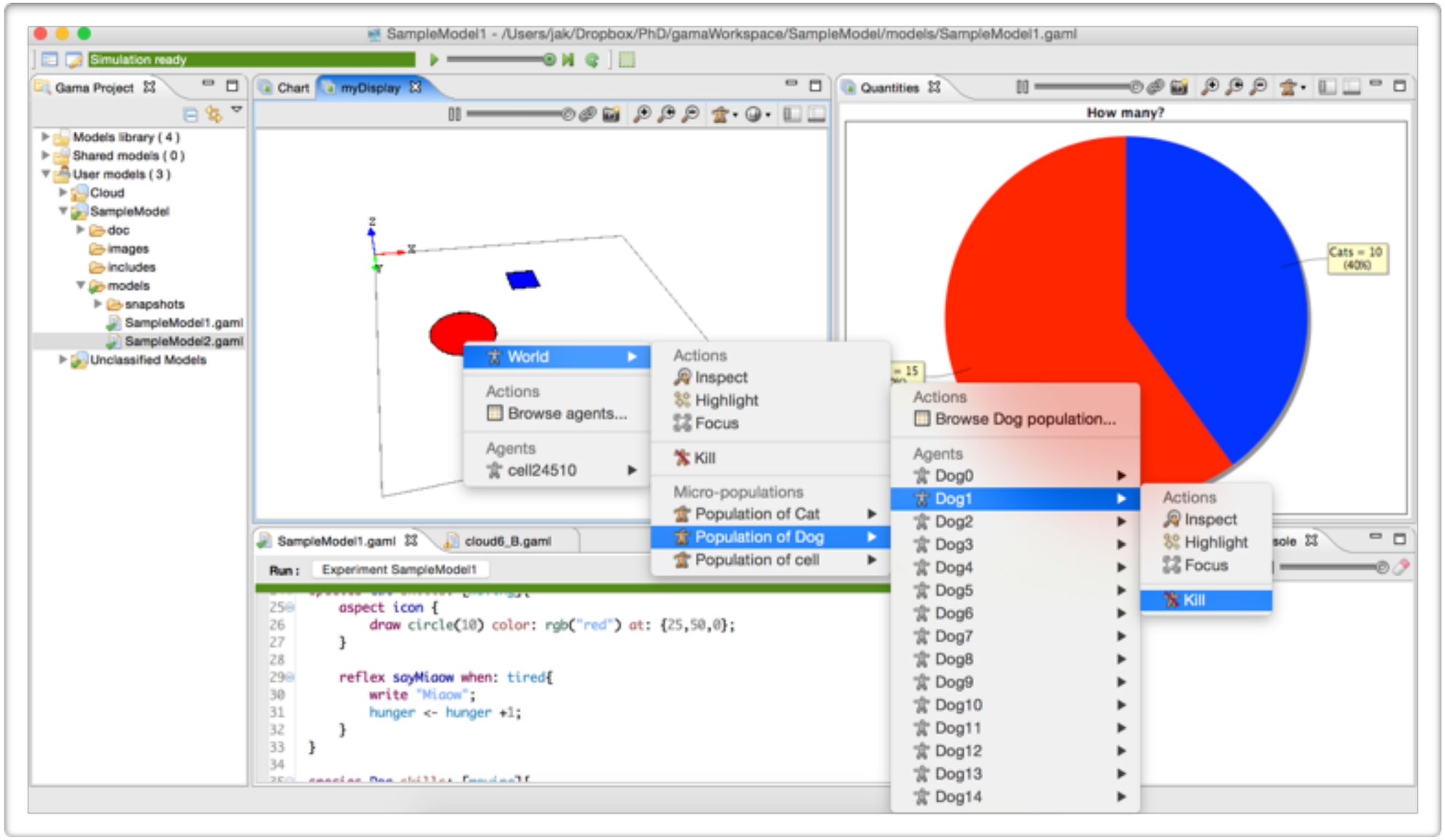
Agent Monitoring

It is possible to monitor agents:

- Agent Browser: browse population of agent species, highlight one, monitor a species.
- Agent Inspector: retrieve information related to one or more specific agent(s), e.g. position, speed, internal variables and the like.

Agent Monitoring

Agent Browser



Data input and output

Data I/O:

- Data can be imported and exported in and from the model.

- Data can be exported by means of the `save` command.

- Several common formats can be read in: .txt, .csv, .png, etc.

- The destination path must refer to an **existing folder**, otherwise an error arises.

Introduction

Modeling

In the following we will take a closer look to the implementation of a MAS in GAMA focusing on:

- The GAML language (structures, operators, etc.).
- The display of agents and data.
- The FIPA communication protocol.

Further information can be found in the documentation.

GAML - Variables

Variables Definition:

- Variables are **declared** with the following syntax:

```
datatype var_name [optional_attributes];
int myVar update: myVar + 1;
```

- If not **initialized** they get default values. To initialize:

```
datatype var_name <- value [optional_attributes];
int myVar <- 5 update: myVar + 1;
```

Variables name must not begin with a white space or digit, by convention it should be a lower case letter.

GAML - Actions

Actions Definition:

- An action embodies a **capability of an agent**, it can take from 0 to many arguments and return 0 or one variable. It is declared as follows:

```
action noArgNoReturn{
}
action noArgReturn{
  return returnVar;
}
action argNoReturn(type1 arg1, type2 arg2){
}
```

- It is possible to assign a return variable directly to a variables as follows:

```
int myVar <- argReturn(arg1::val1, arg2::"val2");
```

GAML - Reflexes

Reflexes Definition:

- A reflex can be considered as an action that the **agent automatically performs** at any time step or when a given condition occurs. In reflexes action can be called. A reflex is defined as follows:

```
reflex everyTime{
  //is executed at every time step
}
reflex someTimes when: booleanExpression{
  //is executed only when the boolean expression is true
}
action argNoReturn(type1 arg1, type2 arg2){
}
```

- The `init` is a special kind of reflex that is executed when the agent is created.

GAML - Loops

It is possible to create a *for loop* (iterating a variable between two values) with the following statement:

```
loop i from: minVal to: maxVal { … }
```

- The code between `{ … }` is executed `maxVal-minVal` times, the increasing statement is implicit.

```
loop i from: 0 to: length(temp_files) - 1 {
   write temp_files[i];
}
```

- Pay attention to the out-of-bound error.

GAML - If / Else

If / Else statement syntax:

```
if booleanCondition {
   //code to be executed if true
} else{
   //code to be executed if false
 }
```

- The compiler does not check the boolean condition, the following code will generate a run time error.

```
if 0 {
   //this is a bad written if statement
 }
```

- Use `flip(numBetween0and1)` to randomly generate `true` or `false`.

GAML - Switch

- The `switch` statement allows to evaluate more combinations than the `if{}else{}`. The syntax is the following:

```
switch expression{
   match 1 { //if expression = 1
   }
   match_one [1, 2, 3] { //if expression = 1, 2 or 3
   }
   match_between [4, 6] { //if expression = 4, 5 or 6
   }
   default { //if expression is neither of them
   }
}
```

GAML - Graphics

- A graphical representation can be useful in several modeling scenarios.

- GAMA allows the display of agents within an environment referred as the grid.

- Layers of agents can be displayed separately.

- The output of the experiments can be displayed too.

GAML - Communication

- In GAMA the communication is based upon the FIPA Agent Communication Language.

- FIPA messages are labeled with a **performative** that specifies the type of message in terms of purpose.

- Thanks to the performatives it is possible to build **interaction protocols** (patterns of behavior).

- Example:

```
reflex sendCFP when: (time = 1) {
    loop p over: Participant{
        do start_conversation with: [
            receivers :: [p],
            protocol :: ‘fipa-cfp',
            performative :: 'cfp',
            content :: ['cont']
        ];
    }
}
```

Language and Communication

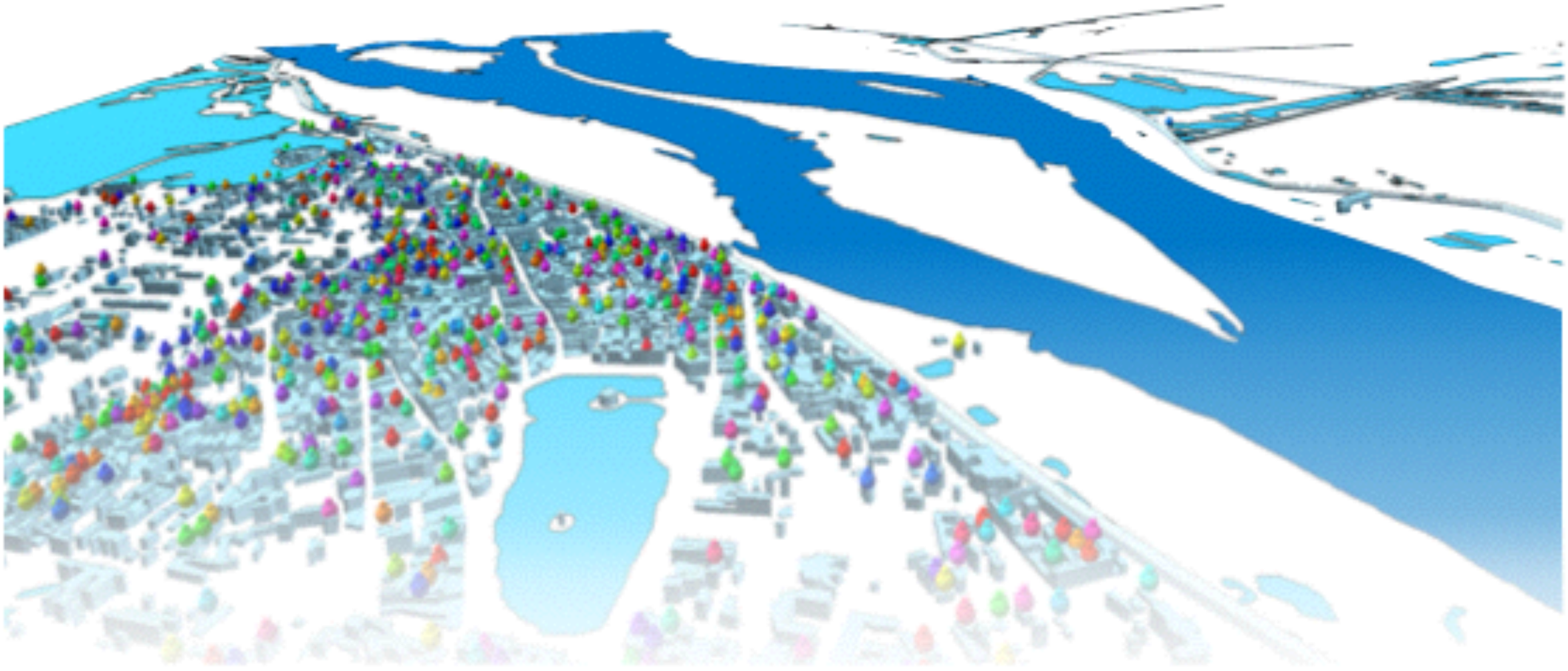
QUESTIONS?

Prey / Predator Model

The aim of this model is to simulate a natural environment in which two species of animals coexist.

- The environment is made up of a grid of cells representing the soil with grass.
- Preys look around for grass to eat.
- Predators look around for preys to eat.

Thanks for the Attention



For any further information: jacopo.pellegrino@to.infn.it