

MASSPA-Modeller: A Spatial Stochastic Process  
Algebra modelling tool  
ICCSW 2011

Marcel C. Guenther, Jeremy T. Bradley

Department of Computing, Imperial College London

September 26, 2011

# Introduction

Spatial population modelling:

- ▶ Systems Biology, Ecology, Performance Analysis, . . .

# Introduction

Spatial population modelling:

- ▶ Systems Biology, Ecology, Performance Analysis, . . .

Problem:

- ▶ CTMCs with enormous state spaces

# Introduction

Spatial population modelling:

- ▶ Systems Biology, Ecology, Performance Analysis, ...

Problem:

- ▶ CTMCs with enormous state spaces

Solution:

- ▶ **High-level modelling languages:** process algebras, stochastic Petri nets, ...

# Introduction

Spatial population modelling:

- ▶ Systems Biology, Ecology, Performance Analysis, ...

Problem:

- ▶ CTMCs with enormous state spaces

Solution:

- ▶ High-level modelling languages: process algebras, stochastic Petri nets, ...
- ▶ **Moments approximating ODEs:**  $\mathbb{E}[Prey]$ ,  $Var[Predator]$  [1, 2]

# Introduction

- ▶ What if high-level descriptions **become tedious?**

---

```
Agent OnOff {
  On  = !(1.0,M,1.0).Off;
  Off =?(M,1.0).On;
};

Locations = {A,B,C,D,E,F,...};

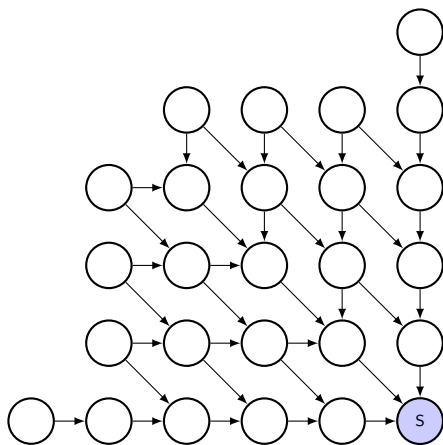
On@A  = 450; Off@B = 450;
Off@C = 300; Off@D = 300;
...

Channel(On@A,Off@B,M) = 1/450;
Channel(On@B,Off@C,M) = 1/300;
...
```

---

# Introduction

Visual modelling:



# Markovian agent models (MAM)s in MASSPA

```
Agent OnOff {
```

```
};
```



# Markovian agent models (MAM)s in MASSPA

```
Agent OnOff {  
    On = !(actionName?,1.0,M,1.0).Off;  
  
};
```

# Markovian agent models (MAM)s in MASSPA

```
Agent OnOff {  
  On  = !(actionName?,1.0,M,1.0).Off;  
  Off = ?(actionName?,M,1.0).On;  
  
};
```

# Markovian agent models (MAM)s in MASSPA

```
Agent OnOff {  
    On  = !(actionName?,1.0,M,1.0).Off;  
    Off = ?(actionName?,M,1.0).On;  
    Off = (actionName?,1.0).On;  
};
```

# Markovian agent models (MAM)s in MASSPA

```
Agent OnOff {  
    On  = !(actionName?,1.0,M,1.0).Off;  
    Off = ?(actionName?,M,1.0).On;  
    Off = (actionName?,1.0).On;  
};
```

Locations = A,B,C,D,E,F,...;

...

# Markovian agent models (MAM)s in MASSPA

```
Agent OnOff {  
    On  = !(actionName?,1.0,M,1.0).Off;  
    Off = ?(actionName?,M,1.0).On;  
    Off = (actionName?,1.0).On;  
};
```

```
Locations = A,B,C,D,E,F,...;
```

```
...
```

```
Channel(On@A,Off@B,M) = 1/450;
```

```
...
```

# MASSPA-Modeller work-flow

- ▶ Step 1: Define sequential agents

# MASSPA-Modeller work-flow

The screenshot displays the MASSPA-Modeller application window. The title bar reads "MASSPA-Modeller - /homes/mcg05/test". The interface includes a menu bar with "File" and "Edit", and a tabbed toolbar with "Agents & Variables", "Locations", "Channels", and "Evaluator".

The main workspace is divided into two panes. The left pane, titled "Define your constants, variables and MASSPA agents below", contains the following code:

```
1 mu = 1.0;
2 lambda = 2.0;
3 epsilon = 0.5;
4
5 Agent OnOff {
6   On = !(1.0,M,1.0).Off;
7   Off = ?(M,1.0).On;
8 };
9
10 Agent HeatSensor {
11   Cool = (mu).Cool + (lambda).Warm;
12   Warm = (mu).Cool + (epsilon).Hot;
13   Hot = (lambda).Warm;
14 };
```

The right pane, titled "Detected Agent(s)", shows a tree view of detected agents:

- Agents
  - HeatSensor
    - Hot
    - Warm
    - Cool
  - OnOff
    - On
    - Off

At the bottom of the right pane is a button labeled "Compile agent definition(s)".

The bottom of the window features a "Console" area with the following output:

```
[02:40:07] Compiling MASSPA agent definitions
[02:40:07] Compiling MASSPA model definition

[02:40:07] Console statistics:
[02:40:07] #WARNINGS: 0, #ERRORS: 0, #FATAL ERRORS: 0
```

# MASSPA-Modeller work-flow

- ▶ Step 1: Define sequential agents
- ▶ Step 2: Create **topology**



# MASSPA-Modeller work-flow

The screenshot displays the MASSPA-Modeller application window. The title bar reads "MASSPA-Modeller - /homes/mcg05/test". The interface includes a menu bar with "File" and "Edit", and a tabbed interface with "Agents & Variables", "Locations", "Channels", and "Evaluator".

Below the tabs, there is a "Generate Topolo..." button, a "Topology Generator:" dropdown menu set to "RectangularGrid", and two input fields: "#Locations X-axis: 2" and "#Locations Y-axis: 5".

The main workspace shows a 5x2 grid of 10 circular locations. The bottom-right location (at coordinates (1,3)) is highlighted in grey, indicating it is selected. A yellow tooltip points to this location with the text: "Add Populations for Location(s) selected in Topology edit popuations:". Above the tooltip is a button labeled "Add/Change Populations in selected Locations".

To the right of the workspace is an "Options" panel with two tabs: "Locations" and "Populations". The "Populations" tab is active, showing a "State:" dropdown menu set to "<ALL>". Below this is a list of population parameters for each location:

```
Off@(0,0)=200.0
On@(0,0)=100.0
Off@(0,1)=200.0
On@(0,1)=100.0
Off@(0,2)=50.0
On@(0,2)=200.0
Off@(0,3)=50.0
On@(0,3)=200.0
Off@(0,4)=100.0
Off@(1,2)=50.0
On@(1,2)=200.0
Off@(1,3)=50.0
On@(1,3)=200.0
```

At the bottom of the window is a "Console" area, which is currently empty.

# MASSPA-Modeller work-flow

- ▶ Step 1: Define sequential agents
- ▶ Step 2: Create topology
- ▶ Step 3: Create **communication patterns**

# MASSPA-Modeller work-flow

The screenshot displays the MASSPA-Modeller application window. The title bar reads "MASSPA-Modeller - /homes/mcg05/test". The interface includes a menu bar with "File" and "Edit", and a tabbed interface with "Agents & Variables", "Locations", "Channels", and "Evaluator". The "Channels" tab is active, showing a "Generate Channels" button and a "Channel Generator" dropdown set to "ShortestPathToSinkChannel".

Configuration options are visible below the tabs: "Sender State: <ALL>", "Receiver State: <ALL>", "Message Type: <ALL>", "Channel intensity: 1", and a checked checkbox for "Receiver Population proportional".

The central workspace contains a network diagram with eight white circular nodes and two grey circular nodes. A dashed rectangular box encloses four nodes in the center. Green arrows indicate directed edges between nodes, showing a flow from the top nodes down to the bottom nodes within the dashed box.

On the right side, the "Options" panel is open, featuring "Add channels" and "View/Edit Channels" buttons. It contains dropdown menus for "Sender State:", "Receiver State:", and "Message Type:", all set to "<ALL>". A text area for "Channel intensity:" contains the value "1". A checked checkbox for "Receiver Population proportional intens..." is present, along with "ADD CHANNELS" and "Delete channels" buttons.

At the bottom, a "Console" window shows two lines of log output: "[02:44:34] Disabled selected locations." and "[02:44:41] Disabled selected locations."

# MASSPA-Modeller work-flow

- ▶ Step 1: Define sequential agents
- ▶ Step 2: Create topology
- ▶ Step 3: Create communication patterns
- ▶ Step 4: Generate **MASSPA** and evaluate using **GPA** [3]

MASSPA-Modeller - /homes/mcg05/test

File Edit

Agents & Variables Locations Channels **Evaluator**

Generated MASSPA model:

```

29 Off@(0,1)=200.0;
30 On@(0,1)=100.0;
31 Off@(1,4)=100.0;
32 Off@(0,0)=200.0;
33 On@(0,0)=100.0;
34
35
36 // CHANNEL DEFINITION
37 ChannelOn@(0,4).Off@(0,3),M = (1.0)/(recvAgentPop);
38 ChannelOn@(0,3).Off@(0,4),M = (1.0)/(recvAgentPop);
39 ChannelOn@(0,2).Off@(0,3),M = (1.0)/(recvAgentPop);
40 ChannelOn@(0,3).Off@(0,2),M = (1.0)/(recvAgentPop);
41 ChannelOn@(0,1).Off@(0,2),M = (1.0)/(recvAgentPop);
42 ChannelOn@(0,2).Off@(0,1),M = (1.0)/(recvAgentPop);
43 ChannelOn@(0,2).Off@(1,2),M = (1.0)/(recvAgentPop);
44 ChannelOn@(1,2).Off@(0,2),M = (1.0)/(recvAgentPop);
45 ChannelOn@(0,2).Off@(1,3),M = (1.0)/(recvAgentPop);
46 ChannelOn@(1,3).Off@(0,2),M = (1.0)/(recvAgentPop);

```

Generate MASSPA definition

Evaluation method:

```

1 ODEs(stopTime=5.0,stepSize=0.1,density=100,closure=MASSPA_infty){
2   E[On@(1,4)].E[On@(1,4)]+Var[On@(1,4)]^0.5.E[On@(1,4)]-Var[On@(1,4)]^0.5,
3   E[On@(0,2)].E[On@(0,2)]+Var[On@(0,2)]^0.5.E[On@(0,2)]-Var[On@(0,2)]^0.5;
4 }
5 Simulation(stopTime=5.0,stepSize=0.1,replications=10000){
6   E[On@(1,4)].E[On@(1,4)]+Var[On@(1,4)]^0.5.E[On@(1,4)]-Var[On@(1,4)]^0.5,
7   E[On@(0,2)].E[On@(0,2)]+Var[On@(0,2)]^0.5.E[On@(0,2)]-Var[On@(0,2)]^0.5;
8 }

```

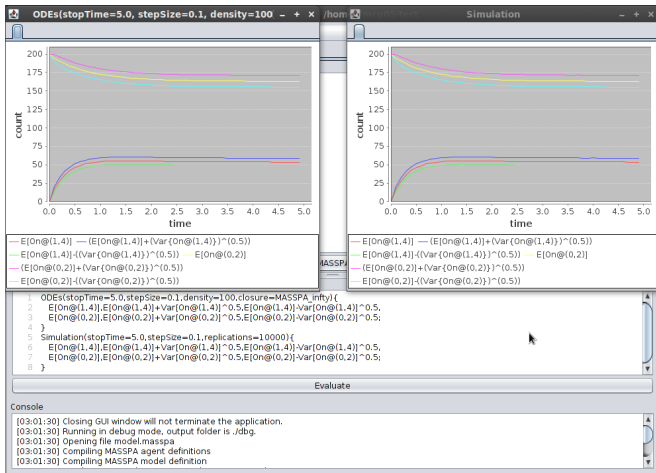
Evaluate

Console

```

[03:01:30] Closing GUI window will not terminate the application.
[03:01:30] Running in debug mode, output folder is ./dbg.
[03:01:30] Opening file model.masspa
[03:01:30] Compiling MASSPA agent definitions
[03:01:30] Compiling MASSPA model definition

```



Thank you!

- [1] J. Hillston, “Fluid flow approximation of PEPA models,” *Second International Conference on the Quantitative Evaluation of Systems QEST05*, pp. 33–42, 2005.
- [2] R. A. Hayden and J. T. Bradley, “A fluid analysis framework for a Markovian process algebra,” *Theoretical Computer Science*, vol. 411, no. 22-24, pp. 2260–2297, 2010.
- [3] A. Stefanek, R. Hayden, and J. Bradley, “A new tool for the performance analysis of massively parallel computer systems,” *Eighth Workshop on Quantitative Aspects of Programming Languages QAPL 2010 March 2728 2010 Paphos Cyprus*, 2010.