

Describing MANTA project using the GAIA methodology

Jakub Boligłowa, Jakub Mojsiejuk

Environment	1
Environmental and organisational rules	2
Agent dependencies (organisation)	2
Agent types	2
Agent Activities	2
Agents and their available tasks	4
Interaction model	5
Role Models	6

Environment

The MANTA project has been designed using EMF model (EthoModelling Framework) which allows for setting up the simulations that included different species of individuals and an environment. By convention in this model, the species are called classes and the individuals — the agents. The EMF models determines the agent's operation and the interactions between them.

The environment is defined as a set of places which are the squares of equal size. There are two categories available — free places and obstacles (obstructed places) which are not usable by the agents and cannot propagate agent stimuli.

The communication between the agents is defined as a propagation of agent's signature in the environment — in the case of the ants we call it a pheromone-like stimuli or a personal stimuli in general terms. A change of state happens by the propagation of a stimulus to adjacent places. The stimulus is left there by the agent and is defined by a tuple (name, strength).

Given a sufficient strength of a stimulus, one that exceeds the activation threshold of an agent, an agent might undertake a specific task.

The environment parameters can to some extent regulate how much stimuli of a given time will be produced for instance: light, humidity and food. Those parameters influence the activity of the agents and the general development of the whole system.

Environmental and organisational rules

There are a couple of rules that define how the simulation proceeds and when it ceases.

- The queen produces an egg every six hours
- If the queen dies, the simulation stops
- If there is not enough food, the *killEgg* task dominates, destroying the ant colony
- There are time intervals at which the environmental agents (Light, Humidity, Food, DeadAnt) drop their stimuli, and there is a limit to a maximum number of dropped stimuli for a given resource

Agent dependencies (organisation)

The environmental agents are independent of the rest of the agents and only respond to the environment parameters (production interval, maximum number of a given resource).

The brood agents require the environmental stimuli (produced by the Environmental agents) to produce their own stimuli: egg, larvae and cocoons.

The remaining agents: ants and the queen are reliant on the all above agent types. Each of their tasks require either an environmental resource or a brood resource.

Agent types

The diagram below depicts the agent types defined in the MANTA project and the stimuli that they produce or respond to.

As mentioned above, there are 3 main agent classes:

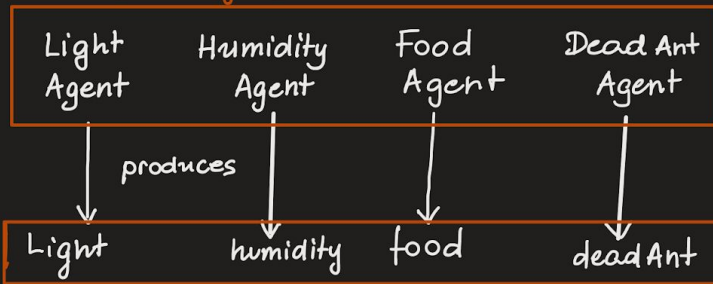
- Environment agents
- Brood agents
- Ant agents

Each class depends on a different stimuli. That stimuli determines what types of interactions are available for the agent. The interactions happen indirectly by performing a task, given a sufficiently strong presence of a stimuli (for further description of the interaction model, please see the section [Interaction model](#)).

Agent Activities

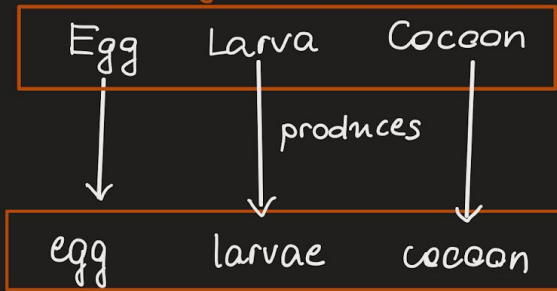
1. *Sensing* - collect the stimuli
2. *Selection* - compute activation for each task
3. *Activation* - choose the greatest activation

Environment agents



Environment stimuli

Brood agents



Brood stimuli

Ant agents

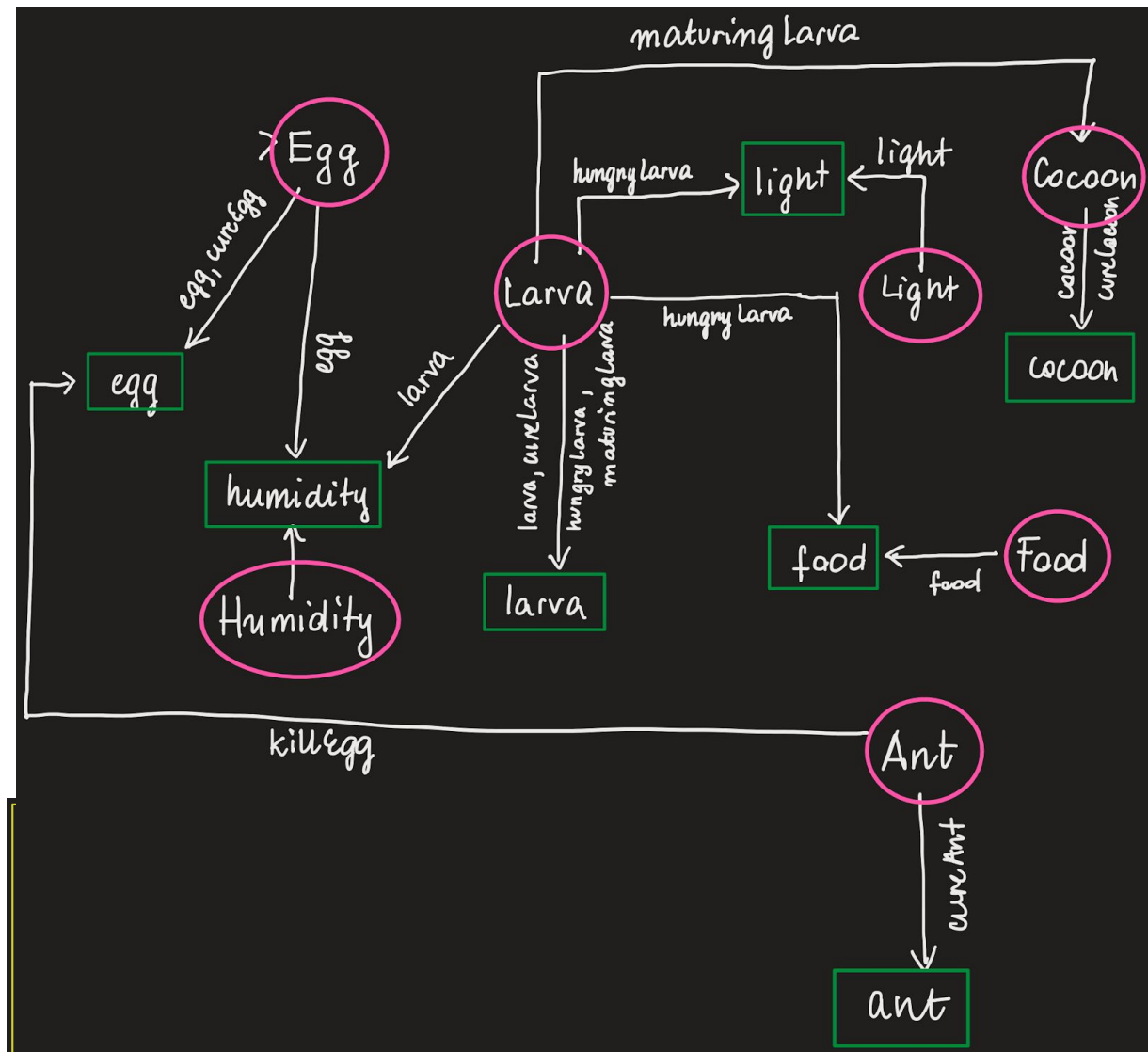


Agents and their available tasks

Agent	Tasks
Eggs	<ul style="list-style-type: none">• egg• cureEgg• Inactivity
Larva	<ul style="list-style-type: none">• larva• cureLarva• hungryLarva• maturingLarva• feeling• careOfEggs
Cocoon	<ul style="list-style-type: none">• cocoon• cureCoccon
Ant	<ul style="list-style-type: none">• cureAnt• hungryAnt• killEgg• killLarva• careOfEggs• careOfEggs
Food	<ul style="list-style-type: none">• food
Light	<ul style="list-style-type: none">• light

Interaction model

Because in the MANTA project the agents do not communicate or interact directly, the communication and interaction happens indirectly via the stimuli left on the places. The diagram below illustrates that principle. Each agent (circled in pink) can interact with a stimuli (put in the green rectangle) via the task (which is listed above or below the indicating arrow). The indirect interaction of the two agents happens when they both react to the same stimuli. For example, in the diagram the Ant agent interacts with the brooding Egg agent via the stimulus *egg*. All subsequent interactions are performed with this schema.



Role Models

Role	Eggs Nurse
Description	This role distinguish by a high level of care of eggs and a low level of inactivity. Main responsibility of this model role is to take care of health every individual in ant colony.
Protocols and activities	careOfEggs, inactivity
Permissions	read checkEggNeedHelpStatus //check if egg need to get health improvement change eggHealthStatus // agent can change destination egg status
Responsibilities	Liveness: EggNurse = (Environment.Population < maxPopulation == true && totalLarvae > 0 && Environment.Resources.TotalFood > 0) Safety: eggHealthLevel <= eggHealthLowLimit => careOfEgg

Role	Larvae-Nurse
Description	This role distinguished by a high level of care of larvae and low level in the other activities. The main responsibility of this role model is to cure larvae.
Protocols and activities	careOfLarvae
Permissions	read checkLarvaeNeedHelpStatus //check if larvae need to get health improvment change larvaeHealthStatus // agent can change larvae status
Responsibilities	Liveness: LarvaeNurse = (Environment.Population < maxPopulation == true && totalLarvae > 0 && Environment.Resources.TotalFood > 0). Safety: larvaeHealthLevel<= HeathLowLimit => careOfLarvae

Role	Egg Feeder
Description	This role distinguished by a high level of feeding activities. Members of this group also show an important level of inactivity. Main responsibility to feed eggs. The main responsibility of this role model is to feed eggs.
Protocols and activities	feeding, inactivity, receiveFood
Permissions	read checkHungryLevel // check if the agent need food change <i>eggHungryLevel</i> // agent can change egg hungry level feedEgg // agent can feed egg
Responsibilities	Liveness: EggFeeder = (amountOfResources < lowLimitFoodLevel => die == true && Environment.Population < maxPopulation == true && totalEggs > 0). Safety: eggHungryLevel <= lowLimit => feedEgg

Role	Larvae Feeder
Description	This role distinguished by a high level of care of eggs and a low level of inactivity. The main responsibility of this role model is to take care of larvae.
Protocols and activities	careOfLarvae, feedLarvae
Permissions	change <i>larvaeHungryLevel</i> //agent can change larvae hungry level change <i>feedLarvae</i> // agent can feed larvae read larvaeHungryLevel // agent can check if larvae need food
Responsibilities	Liveness: LarvaeFeeder = (amountOfResources < lowLimitFoodLevel => die == true && Environment.Population < maxPopulation == true && totalLarvae > 0). Safety: LarvaeHungryLevel <= lowLimit => feedEgg

Role	Larvae-Inactive
Description	This role distinguished by a high level of care of larvae, a high level of inactivity and a very low level of care eggs.
Protocols and activities	inactivity, consumeResources
Permissions	change environmentResourcesLevel read foodLevel
Responsibilities	Liveness: amountOfResources < lowLimitFoodLevel => die