University of Leuven (KUL)
Departement of Computer Science

Multi-Agent Systems

Project 2011-2012

By:
Karsten Daemen
Michiel Staessen

For:
prof. Tom Holvoet
dr. Robrecht Haesevoets

6 juni 2012

1 Introduction

This paper will outline a *Delegate Multi-Agents System (DMAS)* solution that uses *Ant Colony Optimalisation (ACO)* techniques to solve the dynamic *Pickup and Delivery Problem (PDP)*. The performance of this solution will be compared with two other classic *Multi Agent Systems (MAS)*, *ContractNet* and *Gradient Field*.

The considered PDP assumes that a truck can only transport one package at the time and that new packages can randomly be added to the system. The infrastructure will also remain the same and congestion on road segments will not occur.

All of the agents in the presented DMAS solution are simulated in a virtual environment (e.g. a private cloud). It is not very affordable or even practical to equip every pickup and delivery location (in the physical environment) with a communication device. These locations can change a lot and these devices are necessary because agents on these locations have to be able to communice with each other. Because the whole DMAS is run on the virtual environment, truck drivers (in the physical environment) will only need some kind of GPS device to receive the planned path from the simulation in the virtual environment.

The purpose of this paper is only to outline the implementation of an ACO based DMAS for PDP. Therefore, the reader is supposed to have a decent knowledge of *Multi-Agent Systems (MAS)* and the applications of *Ant Colony Optimalisation (ACO)* techniques in *Delegate Multi-Agent Systems (DMAS)*.

2 Approach

The DMAS solution we developped defines two agents, *Package Agents* and *Truck Agents*. All Package Agents have a corresponding package and are located on the pickup location of that package. Truck Agents control a corresponding truck and move along with the truck. Both agents also own a *pheromone table*. These tables can hold multiple paths (list of Package Agents in a certain order) and a corresponding pheromone (heuristic value).

Package Agents and Truck Agents communicate with each other by sending ants. These are a sort of smart messages.

There is also a *"forwarder"* placed in each delivery location of a package. These *"forwarders"* will send all received ants to the Package Agent corresponding with the package for the local delivery location. Because of their limited intelligence, forwarders are not considered to be agents.

Like most DMAS based on ACO, three type of ants will be used. They will be used in the following way:

Feasability Ants These ants will be periodically broadcasted over a certain radius from the Package Agents to the forwarders (delivery locations). Their main purpose is to inform other Package Agents wich Package Agents are in a certain radius near their delivery location.

Exploration Ants These ants will be sent from the Truck Agents to discover and evaluate a path of Package Agents. They will update the pheromons table in the Package Agents during their return over the discovered path.

Intention Ants These ants will be sent from the Truck Agents over the path of Package Agents that they are planning to follow. These ants will decrease the pheromone values for that path in the visited Package Agents in order to discourage other agents to follow this path.

How agents find an optimal path by using the pheromones and communication over ants is further explained in the following scenarios.

Scenario I: Broadcasting of Feasibility Ants

Scenario I is illustrated in Figure 1.

In this scenario Package Agent A (depicted as a brown box 'A') is added to the simulation. It will broadcast a Feasibility Ant $\rightarrow A$ in a certain radius to all forwarders (depicted as yellow flags). Forwarders C' and B' receive the feasibility ant and send it to their corresponding Package Agents C and B. These Package Agents now know in the future that package A is near after they are delivered. C and B will therefore add path $\rightarrow A$ to their tables with a default pheromone value, they will broadcast a new Feasibility Ant $\rightarrow CA$ and $\rightarrow BA$ respectively. Only forwarder F' receives the ant $\rightarrow BA$ this time and sends it to Package Agent F. Package agent F will add $\rightarrow BA$ to its pheromone table and broadcast a new

Feasibility Ant again so the whole process can repeat itself. This goes on until the max number of hops (= number of broadcasts) is reached.

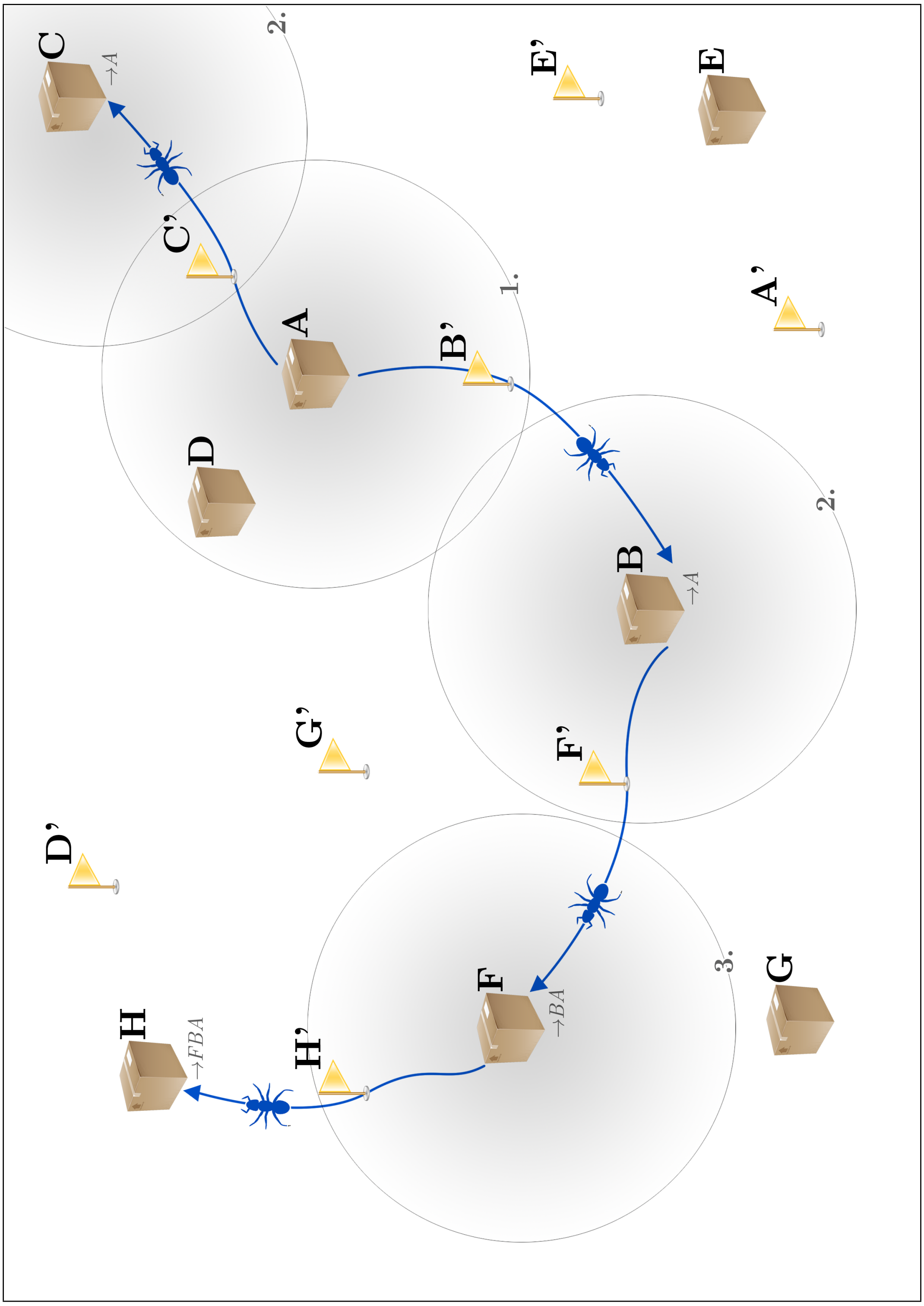
For the classic PDP problem, a max number of hops of 1 is fine. Package Agents only need to know which other Package Agents are near their delivery location. Therefore only one broadcast is needed. Nevertheless could some extensions of PDP benefit from multiple hops (obligatory agents etc ...).

Scenario II: Sending of Exploration Ants

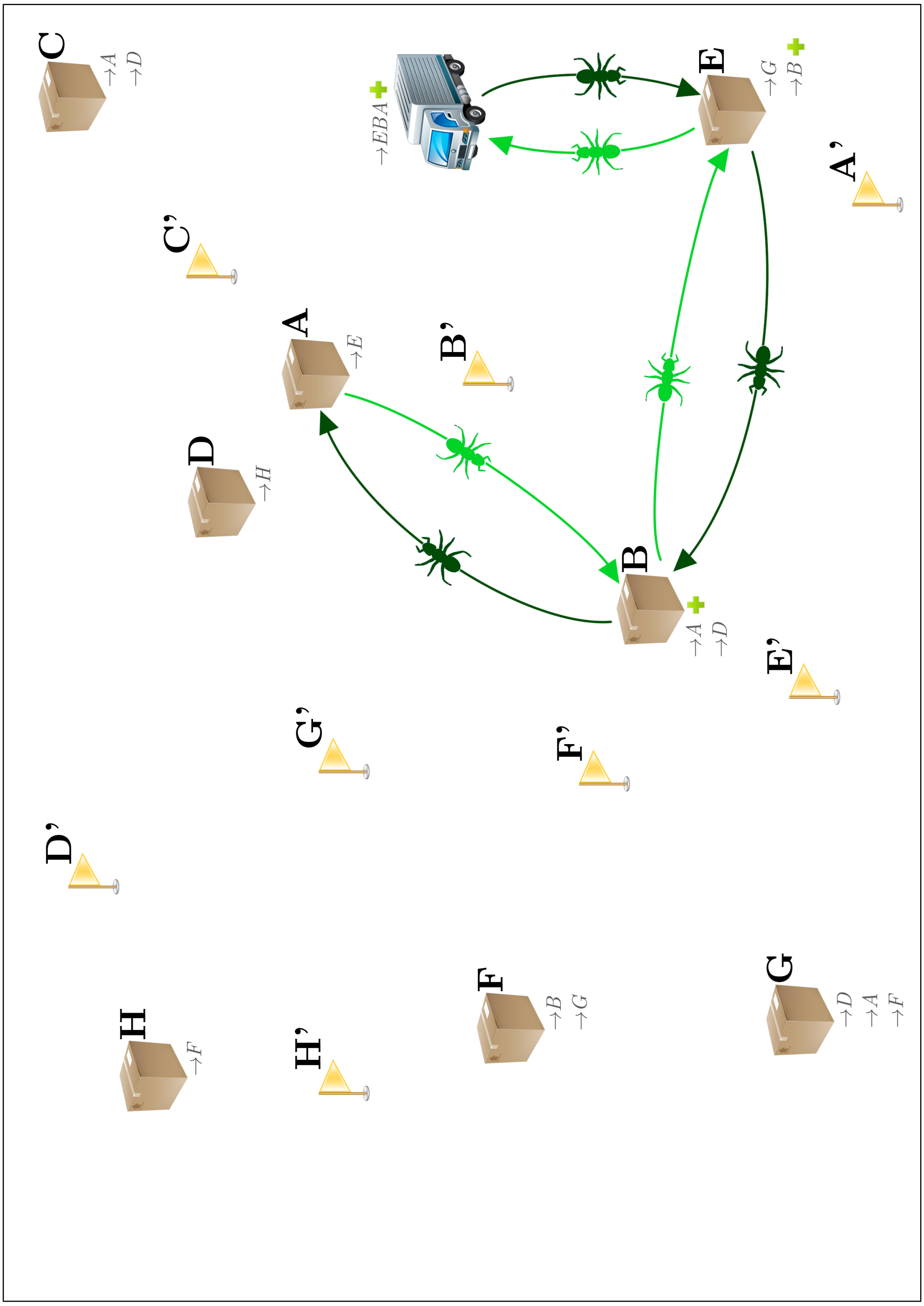
Scenario II is illustrated in Figure 2.

In this scenario, the Truck Agent is moving to Package Agent E to pick up the package. It sends an ExplorationAnt to Package Agent E to explore wich other packages he is able to pick up after he delivers package E. Package Agent E receives the ExplorationAnt and sees in his pheromone table that packages G and B are near the dellivery location of E. Scenario I (2) showed how these entries came into the pheromone table through Feasibility Ants. Package Agent E choses to send the Exploration Ant further to either $\rightarrow G$ or $\rightarrow B$. The higher the pheromone value for a path the higher the chance that the path will be chosen. The pheromone value *ëvaporates"* over time but returning Exploration Ants from that path can increase the pheromone value. The amount to increase depends on a heuristic function that takes several properties of the path into account, like the priority of the packages and the distance to the pickup location.

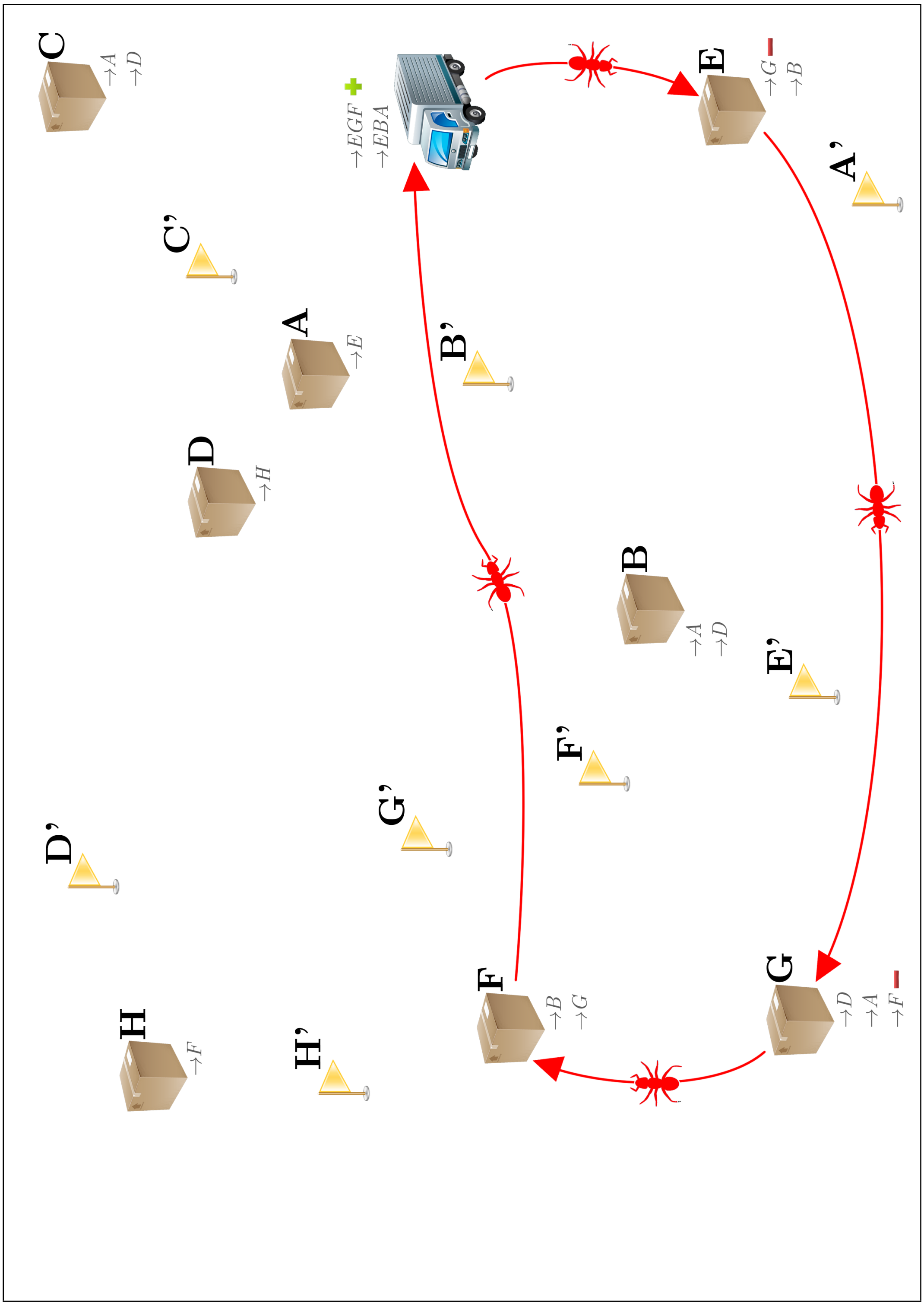
PackageAgent E choses to send the Exploration Ant further to PackageAgent B, who in turn choses $\rightarrow A$ (using his pheromone table) to send it further to PackageAgent A. The Exploration Ant has reached its max number of hops and PackageAgent A send the ant back to B. In B, the ExplorationAnt will evaluates the path $\rightarrow A$ and increase its pheromone value before going back to E. In E, the ExplorationAnt will evaluates the path $\rightarrow B$ and increase its pheromone value before going back to Truck. At the truck the exploration ant will put the whole path $\rightarrow EBA$ in the pheromone table (if it was not already in there) and increase the pheromone value with the evauluated value for the whole path $\rightarrow EBA$.



Figuur 1: Scenario I (2)

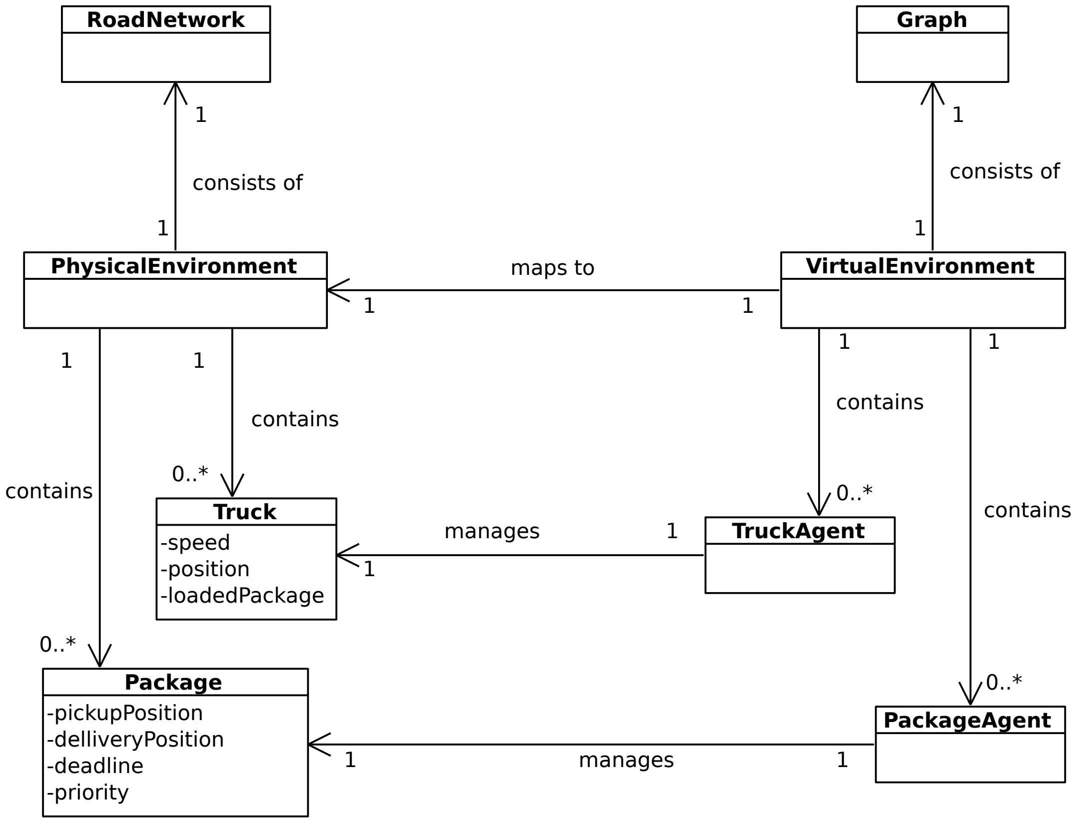


Figuur 2: Scenario II (2)

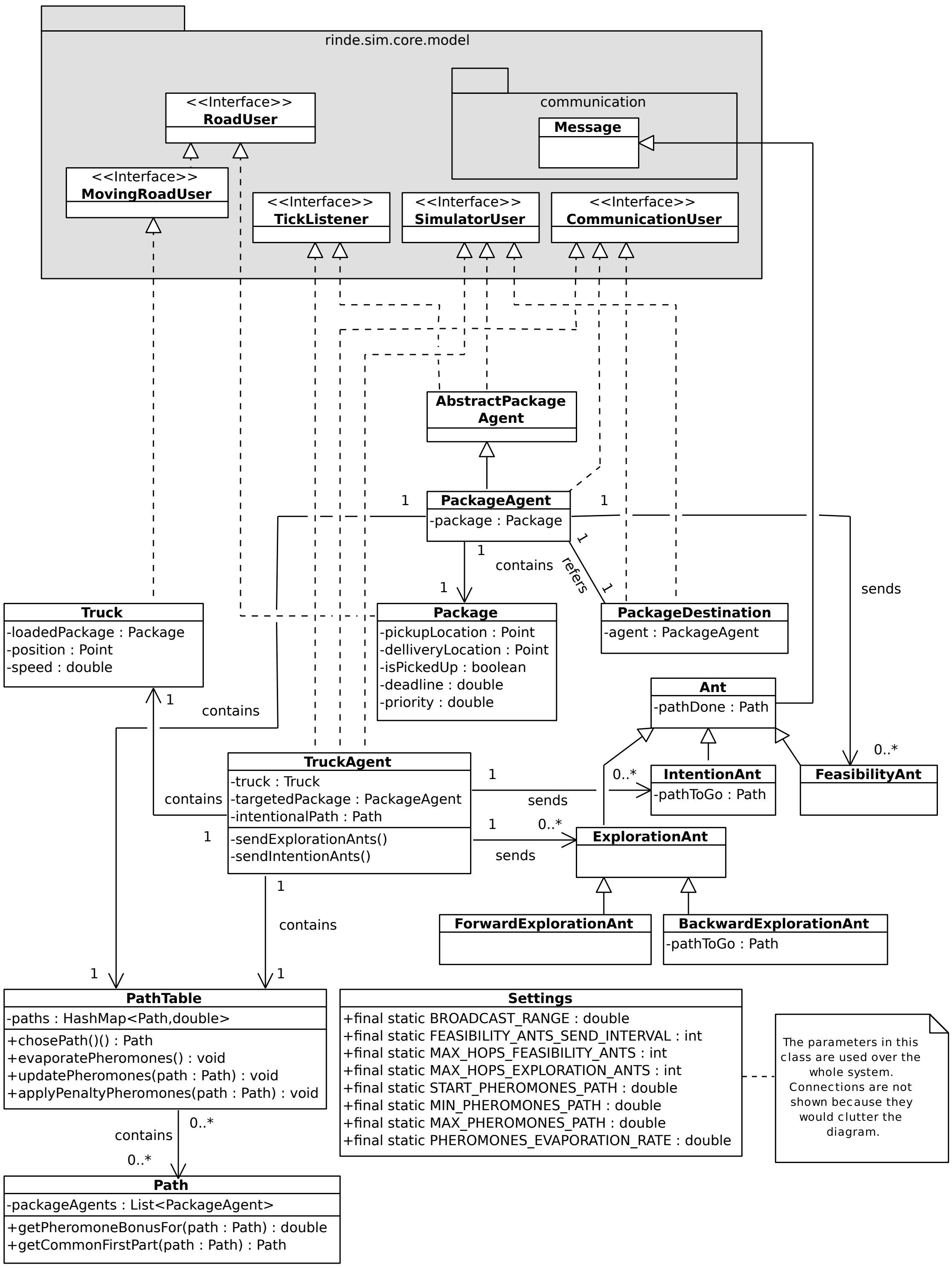


Figuur 3: Intention Ants (example scenario 3)

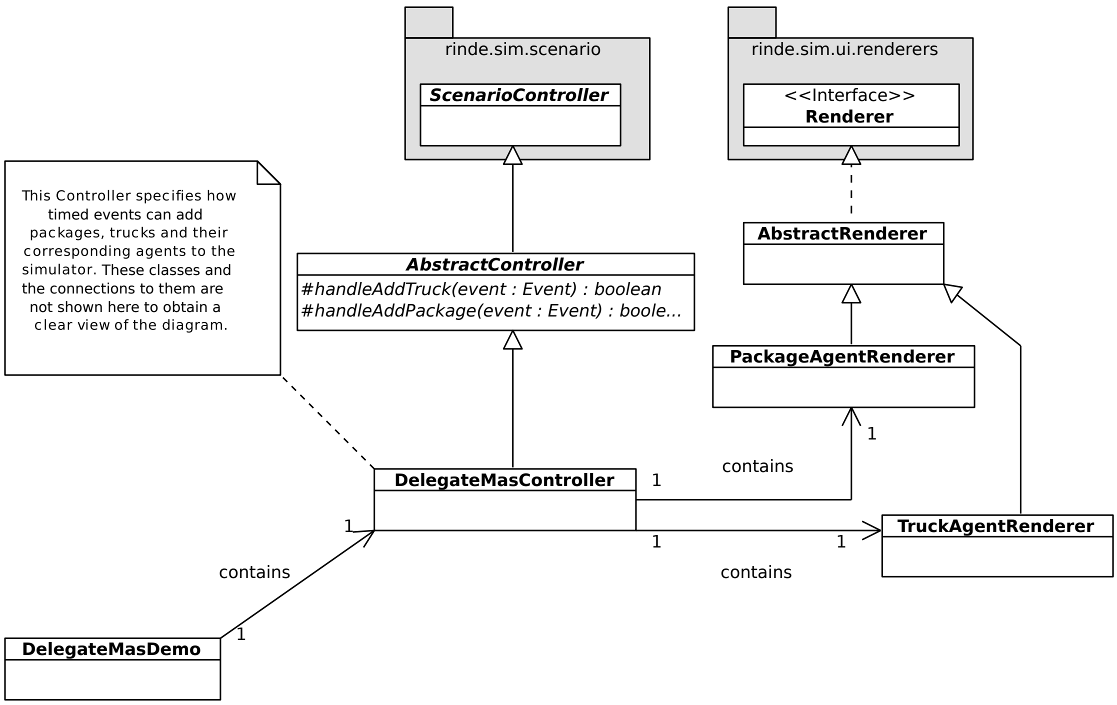
3 Development



Figuur 4: Class Diagram (part I)



Figuur 5: Class Diagram (part I)



Figuur 6: Class Diagram (part II)

4 Evaluation

In order to evaluate the DMAS solution, it is compared to two reference solutions: Gradient Field and Contract Net. We will describe them in a nutshell.

Gradient Fields are based on electrical fields: packages and trucks have opposite charges and will thus attract each other. However, trucks will repel other trucks because they are able to move and have equal charges. This will make sure that the work is divided among all trucks.

In a Contract Net, package agents will broadcast the position of their packages. (Some) Truck agents will receive the broadcasted message and repond with an offer to pick up the package. The package agent will then evaluate all the offers it received and inform the senders whether they won the offer or not. When a truck receives an accepted proposal, it can still decide that there is a better alternative. In this case, the package agent is informed of a failure. The contract is now broken and the package will again broadcast the position of its package.

4.1 Experiments

As explained in **??**, the DMAS solution will try to plan an optimal route. As a consequence, we are particularly interested in the performance of this system. We test this in a number of ways.

First, the overall performance is measured by counting the number of delivered packages in a fixed time window.

Second, we track the time that passes between the creation of a package and the pickup. This can serve as an indication for the responsiveness to dynamism.

Third, all packages should be delivered within a given time window. Therefore, the lateness for every delivery is logged.

Finally, to evaluate the efficiency of the chosen routes, the total distance travelled by trucks is also measured.

All this information is gathered during a fixed-time scenario in which a number of trucks start with an initial amount of packages. Every time a package is delivered, a new one is added. This will prevent the trucks from running out of packages and will not flood the system with too much packages either.

Because of the optimal path planning, we expect to see a high thoughput (delivered packages) and a low value for total distance travelled.

4.2 Comparison

4.3 Critical Reflection

First of all, the system manages to get the task done. A huge advantage to this system is that it can plan ahead, whereas the Contract Net and the Gradient Field solution in particular do not/cannot plan ahead. Using heuristics, the delegate MAS solution can find an optimal path throughout the system.

One of the disadvantages about this system is that it will not always find packages at a given moment. Sometimes, it is required to navigate randomly until packages nearbycan "hear" the truck agent. One way to solve this problem would be to periodically increase the broadcast radius of truck agents (up to a maximal range) when no packages are found in a certain time window.

Another way to solve this problem could be to combine multiple solutions into one hybrid solution. When no packages are found, a Gradient Field could be used to spread the agents accross the physical environment. This will increase the coverage of the map and hence the chance for packages to be heard by trucks.

5 Conclusion