# DEVELOPMENT OF INTELLIGENT TRAFFIC LIGHTS USING MULTI-AGENT SYSTEMS

*Allan Saldanha Vital, Senac Universitary Center, allan.vital@gmail.com*

*Vitor Navarro, Senac Universitary Center ,vitor.navarro87@gmail.com*

*Fabrício J. Barth, Senac Universitary Center, fabricio.barth@gmail.com*

*Orlando Rodrigues Jr, Senac Universitary Center, rodrijr@gmail.com*

## ABSTRACT

The goal of this paper is to develop an intelligent traffic light controller to optimize the flow of a given region roads. The behavior of each traffic light controller is implemented by an intelligent agent able to act autonomously and communicate with other agents aimed at collaborative decisions and actions into a region's traffic. The concepts of intelligent agents and multi-agent systems have been applied, partitioning the control of roads and regions across types of agents. In addition, the solution was created based on three parts: the simulator, the multi-agent systems and the agent framework. The results of the simulations with the prototype showed their strengths and weaknesses, and despite his inefficiency, it showed too the development potential of the proposed solution.

*Keywords: Intelligent Traffic Light Control, Multi-Agent Systems, Traffic Engineering*

## 1. INTRODUCTION

The expansion and diversity of technology in today's world have caused, among other effects, the reduction of the production cost of many devices which made them and their sub-products accessible to social layers of low-income and more attractive to social layers of middle and higher incomes. One product that is part of this group is the automobile. The automobile manufacturing had suffered great automation since the industrial revolution and its plants are among the most robotic of the world, therefore the availability and price coupled with the easy credit led to an accelerated increase in sales (FRANCE and GHORBANI 2003; WIERING et all 2004).

Therefore the city roads can no longer contain this rampant growth, so traffic jams became more constant in the large cities causing an increase in travel time of all vehicles, resulting in greater pollution, greater dissatisfaction of users and major logistical problems for companies both public and private.

The solutions presented today do not solve this problem anymore, and therefore need new approaches. These may propose changes in various components of a traffic system, including: vehicle size, road size, performance of public transport and traffic control. One way to reduce the high flow of vehicles is to modify the form of traffic control, in other words, the performance of traffic routes.

The traffic light is an artifact of traffic control placed on a track, usually at intersections. He is responsible for release or detain the flow of vehicles traveling over it. The way one or a set of traffic lights controlling the flow of vehicles, change the characteristics of flow, such as travel time and number of vehicles on the roads.

Currently, most traffic signals installed in real environments are using timers and are coordinated by central control. This type of implementation takes into account the time that the lights should remain open or closed, and only carries a few settings for times and situations predetermined high traffic, low or normal. Furthermore, the lights do not have sensors to enable real-time changes in accordance with the traffic situation. An example would be cameras count vehicles located at the traffic lights.

Yet there is the problem of lack of coordination between the traffic lights, since the synchronization between them is only temporal. This problem is nothing more than the fact of a traffic light cannot act in accordance with the requirements of all other traffic lights, failing to get the best possible solution for the track in question.

The most successful approaches to improve this flow, also acting on the traffic light, uses as the basis of their solutions intelligent agents combined with other knowledge, as historical data (BALAN and LUKE 2006), reinforcement learning (WIERING et all 2004), estimation of value (SCHEPPERLE, BÖHM & FORSTER 2007) by inserting calculations and planned, complex and dynamic actions.

Following the research of the studies cited, it is believed that the solutions closer to the best results are the fruits of a similar combination. Therefore, the traffic lights will be modified using techniques of artificial intelligence and multi-agent systems as a solution to the problem presented above.

## 1.1 Goal

This paper goal is to develop an autonomous traffic light control, where the behavior of each one is implemented by an intelligent agent able to perceive the environment through sensors and act on the environment autonomously. In addition, each of the lights should be able to communicate with others due to a collaborative decision on the actions that must be performed in a region of traffic.

It is believed that with the traffic light control system based on the concepts and technology, described above, you can improve the flow of vehicles in the region where it's employed.

Simulations will be carried out to validate this hypothesis, they will compare this paper proposition with other solutions already employed in traffic light control systems.

### 1.2 Paper organization

This paper is organized as follows. Chapter 2 presents a description of the problem through related paper. Chapter 3 describes the concepts of intelligent agents, multi-agent systems, communication between agents via FIPA specifications and the proposal of this paper. Chapter 4 describes the development of solution. Chapter 5 shows the experiments and its results. The final considerations are given in Chapter 6 and future works are shown in Chapter 7.

## 2. RELATED PAPERS

The use of intelligent agents is evidenced in several projects that aims to solve some of the problems related to land transport.
Roozemond (1999) describes the use of intelligent agents for control of traffic lights and intersections where each agent receives traffic data about its region and works autonomously on it, and also can interact with other agents to optimize the system solution as a whole. It also provides an idea of using leader agents for a group of them to achieve this optimization in an organized manner.
This solution is deepened and broadened in the similar paper of France (2003) which in order to achieve this optimization uses the same concept as Roozemond creating a multi-agent system and implementing the traffic coordinator.
In this paper the goal of the intersection agents is to keep the best possible performance, constant flow of vehicles, at their designed location. The coordinator agent has the goal of maintaining the best possible set of intersections, limiting the autonomy of decision from the intersection agents, so this achieves the optimization of the system as a group.
The work of Wiering (2004) presents a set of solutions that have been used previously by other research and proposes the use of multi-agent systems with reinforcement learning (reinforcement learning) for control of traffic lights.
The reinforcement learning is combined with results of functions given by road users, being used as a parameter to determine the best decision in each traffic light. The difference of the proposed solution is into modeling the system with co-learning, where the traffic lights and cars learn to improve both its flow and its path with every decision they make. According to the article that allows greater flexibility, however a part of the system need to be incorporated into vehicles for the effective use, which is likely to cause problems of price for invasion of privacy.
There are many more interesting researches about the theme in question but this paper shown the most important and connected to its solution.

## 3. THEORETICAL FOUNDATION AND PROPOSAL

To adapt the concepts of the structure of the control system for traffic on the area of artificial intelligence is necessary to define a series of concepts, its equivalents and responsibilities.

## 3.1 Intelligent Agents

The term intelligent agent has several definition differences. Each fits the needs of application subject.
One of the most broad definitions and that begins to give shape to who are the agents within the traffic control system proposed in this work, is showed by Russel and Norvig (2002, page 32).

> "An agent is anything that perceives its environment through sensors and act on it through actuators"

Building on that thought it is created a relationship between traffic lights and intelligent agents. As the traffic light is a mechanism used to organize the streets flow trought commands that indicate that vehicles should stop or go, it is stated that the traffic light is an agent since it is also a device that senses its environment, i.e. traffic flow, and acts on it trough commands like open or close.
In addition to the basic definition showed before, is added Wooldridge (2002, page 3) vision "An agent is a computer system that is located in an environment and which is capable of autonomous action in this environment to achieve their goals".
Summing all these definitions resulted in the creation of the following basic characteristics which the implementation of intelligent agents should follow: ability to act autonomously; ability to perceive their environment, i.e., to capture information about the current state of the road system; ability to act indirectly in their environment.
Agents has been developed considering these concepts, they incorporate these and other features to meet the traffic control system goals.

## 3.2 The system agents: RCA and LCA

The traffic light is the smallest control unit in the traffic system. They control the traffic flow in only one route. Thus your control is termed as local, he would be represented by a Local Controller Agent (LCA), that means this intelligent agent is responsible for just one route.
The ACL will receive information about traffic flow in real time, carry out the calculations and then take the decision to open or close in order to optimize the flow on the road under their control.
Therefore, the actions taken by the LCA are:

- Request information about the current state of the road under its control;
- Request information about the current state of other LCA's;
- Send information about its current state to another requesting agent;
- Perform calculations to determine whether to open or close;
- Ordering green sign;
- Send the result of its calculation.

The actions number two and three exists so the system could be refined when is necessary, with the goal of creating an interaction between LCA's, making the solution in each road intersection more coordinated and optimized.
In addition, to the LCA make its decisions and realize one or more actions some data was been determined to be captured from the environment through its relevance. These data are present in the equation that define how they will be used.

$$\mathrm{ACL_{value}}\,(\alpha,\varphi,\rho,\omega)=\begin{cases}\alpha*\varphi*(\rho/\omega)\ \text{if}\ \rho\geq\omega\\ \alpha*\varphi\ \text{if}\ \rho<\omega\end{cases}$$

Where:

- Number of cars in the road of some traffic light, represented by α;
- Amount of time (in cycles) that the traffic light is closed, represented by ρ;
- How many times the LCA requested green sign and it was denied, represented by φ, and.
- The maximum time limit (in cycles) which the traffic light could remain closed, represented by ω.

Currently, most traffic lights perceives its environment, this means taking the above information, with the aid of one operations control center, which can be controlled by people, automated, or a combination of both. Moreover, their role is important in all of the current system because the orders for the operation of traffic lights originate in each one of these control centers, which leads to two other decisions: these control centers will also be represented by agents, and the physical mechanism that is used as a traffic light should contain means of gathering data from their routes without the need of direct help from the central, with the aim of decentralizing part of the decisions.
These central have an area of operations designated under a series of political and operational rules set by the technical areas responsible for traffic engineering, for example the São Paulo Traffic Engineering Company (CET1).
They are represented by the Regional Controller Agent (RCA) and the delimitation of its actuation area is physical and computational, that is, it is limited to how much information can render in real time according to the size of the region and the number of LCA's it has communication and infrastructure.
The goals of each RCA are: controlling the synchronization between the LCA's, to organize the flow of your area, and, optimize the flow of your area;
Next are described the possible actions that an RCA can execute to achieve its goals:

- Request information about the current state of ACL's that are in your region;
- Request information about the current state of an RTA from another region;

[1] The São Paulo Traffic Engineering Company (CET) is responsible for traffic managing and engineering in São Paulo state.

- Send information about its current state to another ACR requesting;
- Perform calculations to optimize control of the region, and;
- Send answers on applications for green light, taking into account their calculations.

The calculation and redistribution of ACR values can be displayed in the snippet below.

```
int k ← 0
while k < cache.size()
        /*...code...*/
         if cache.position(k).getAclValue() > 0 then
                formula ← ((carNumber * requisitionOcurrence) - openedTime /
10) * cache.position(k).getAclValue()
        end if
        else then
                formula ←((carNumber * requisitionOcurrence) - openedTime /
10)
        response.add(cache.position(k).getI()+","+
cache.position(k).getJ()+":"+formula+";")
        end else
        k ← k+1
end while
```

Figure 1: Pseudo-code with RCA formula

The calculation is performed for each ACL and uses the result of your formula, represented by cache.position(k).getAclValue(), sent to the ACR during messages exchange.
With both agent types defined is known that the lights are represented by the equivalent amount of LCA's and the centrals by RCA's. Thus, the traffic system is controlled by multiple agents, who must act together to achieve a common goal, characterizing the solution as a multi-agent system (MAS).

### 3.3 Multi-Agent Systems and Layered Hierarchy

One MAS is composed by a group of intelligent agents that have common goals and cooperate in an organized way to achieve those goals (BORDINI, Hübner & WOOLDRIDGE 2007). Figure 3 have a sample of this description.

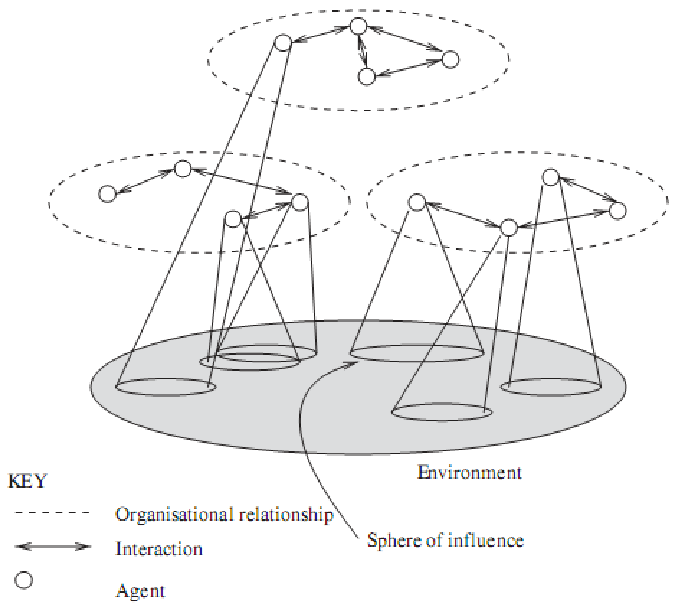


Figure 2: MAS environment sample (BORDINI, HÜBNER & WOOLDRIDGE 2007)

In an environment of multiple agents, each one has its own work space and can be multiple sets of agents with different sub-goals who act in an organized manner to achieve a common goal on the environment, and there is a point of communication between each agent or between the sets of agents, this means agents can communicate with each other at any time or the group can exchange messages block, forming an MAs. In this aspect there is room for communication between multiple MAS's, but they must meet the definition of cooperation and common goals.

The LCA grouping is one of the sets from MAS and the function of each member was been extended from perceive their environment and act on it to achieve its sub-goal to do this through communication with other agents. As an aid to decision-making and control of information, the other set is represented by an RCA, which communicates with the LCA's to reach the sub-goal of optimizing a region which in the case of the proposed MAS is the common goal of all intelligent agents.

Layers hierarchy between agents has been defined for better organization and a more coherent architecture. The motives for this decision were the need for local control (LCAs) and another control region (RCA), the first child to second, and also to make full use of the discussion among agents for decision making.

Thus the hierarchy of the proposed MAS is initially composed of three levels, and the third is optional and was not required for the implementation.

Level 1 or Local Control: composed by traffic lights agents, called local controller agent (LCA) they are responsible for capturing information about the route or crossing where they are located and pass their decisions for the region controller agent (RCA);

Level 2 or Regional Control: composed of one or more RCA, where it is responsible for coordinating the flow optimization of a region, it receives the information passed by the LCA's, states who has the decision accepted and who must make another decision and returns for each LCA the matching resolution;

Level 3 or Global Control: composed of one or more global controller agent (GCA). The GCA has similar functions to RCA, but it optimizes the flow between regions (RCA's). This means he is an agent that talks to the RCA's of his domain and passes them data about flow control at the points of connection between these regions.

The third level should be used when a more accurate control is needed and/or there is too much regions of level 2 self-controlling and is desired to have some level of organization between them. The number of levels can increase or decrease as often as necessary according to the size of each region and the desired granularity and control for these regions. Given the size and goal of the experiments only levels 1 and 2 were necessary.

### 3.4 Agents Communication

By definition MAS must include communication between its agents. Even if the agents composing the system are independent or different, it is imperative that they have the ability to exchange information to achieve goals.

To achieve communication capabilities is necessary to implement or use a framework with agent communication language, which describes how they will exchange information and its shape. The most know possibilities are: Knowledge Query and Manipulation Language (KQML) (WOOLDRIGE 2002, page 170) and Foundation for Intelligent Physical Agents-Agent Communication Language (FIPA-LCA) (WOOLDRIGE 2002, page 175).

KQML is a language based on messages, which are similar to the objects of object-oriented programming. According to Wooldridge (2002) Knowledge Sharing Effort (KSE) has defined it as follows: KQML is an 'outer' language for agent communication. It defines an 'envelope' format for messages, using which an agent can explicitly state the intended illocutionary force of a message. KQML is not concerned with the content part of messages"

Messages are formed by one perfomative and parameters (key/value pair), next is the KQML message sample:

```
(ask-one
        :content  (traffic  region_sp  ?traffic)
        :receiver  ACR0
        :language  java
        :ontology  traffic_com
)
```

FIPA also defines an 'outside' language for messages and fixes to 20 the number of performatives, assisting the interpretation of these messages. Another plus it the fact it does not require specific language for the message content. The visible difference between KQML and FIPA are the performatives it sets.

Like KQML the message defined by FIPA is formed the same way:

```
(Inform
        : sender LCA11
        : receiver LCA12
        : content (intense traffic)
        : language java
        : ontology traffic_com
)
```

From the above options the chosen language was FIPA. The reason for the choice, besides its potential, is that Java Agent Development Framework (JADE) already implements FIPA LCA for agent communication and there is was need to lose time creating anything to use FIPA LCA, which made this work proposition less harder to implement.

# 4. IMPLEMENTATION

The proposed system was implemented as a prototype, whose goal is to compare the proposed solution with other solutions presented in this work.
This prototype consists of 3 separate components that all together create the MAS traffic system. The components are: Green Light District - GLD, JADE and the agents (LCA and RCA) and link between them and the other two environments.
The architecture and environments are discussed below.

## 4.1 Architecture

To create the prototype it was determined the use of object-oriented programming (OOP) with Java™ programming language, due to high capacity for abstraction, modularity, ease of maintenance and production capacity of OOP and interoperability of Java™. Besides this combination provides easy connection to GLD and JADE, since both are java-base programs.
Another motivation for the choice of Java ™ is its wide use in projects and work related to AI in academic and commercial areas, for example WIERING (2004) and FRANCE & Ghorbani (2003) in academic area and JACK® in commercial area.
The prototype requires an environment for agents to act and for testing its efficiency, this environment is a traffic simulator. This simulator needs to represent as closely as possible streets and crossings, produce quantitative results and provide traffic information to the prototype.
An external application was determined to assume this role, since its implementation would cost too much time. This application is the Green Light District (GLD), which despite some limitations meets the requirements listed above in addition to easy integration with the rest of the components because it is an open source project written in Java™. It is noteworthy that despite the tight integration with the prototype, he can be easily disengaged from the GLD changing the connection points between them and adding another environment and/or simulator.
GLD create and feed the environment of the agents, but is JADE that supports its existence and implementation. JADE is a framework that aim to simplify the creation of MAS's through a middleware, and that it is compliant with FIPA-LCA.
This is why the JADE is used, it provides base classes for implementation and communication between agents within the FIPA specifications. In addition he is a host for the agents of the system, registering and managing the life cycle, helping in MAS control.
The macro view of the system is shown in Figure 3, which presents only the classes more relevant in terms of architecture.

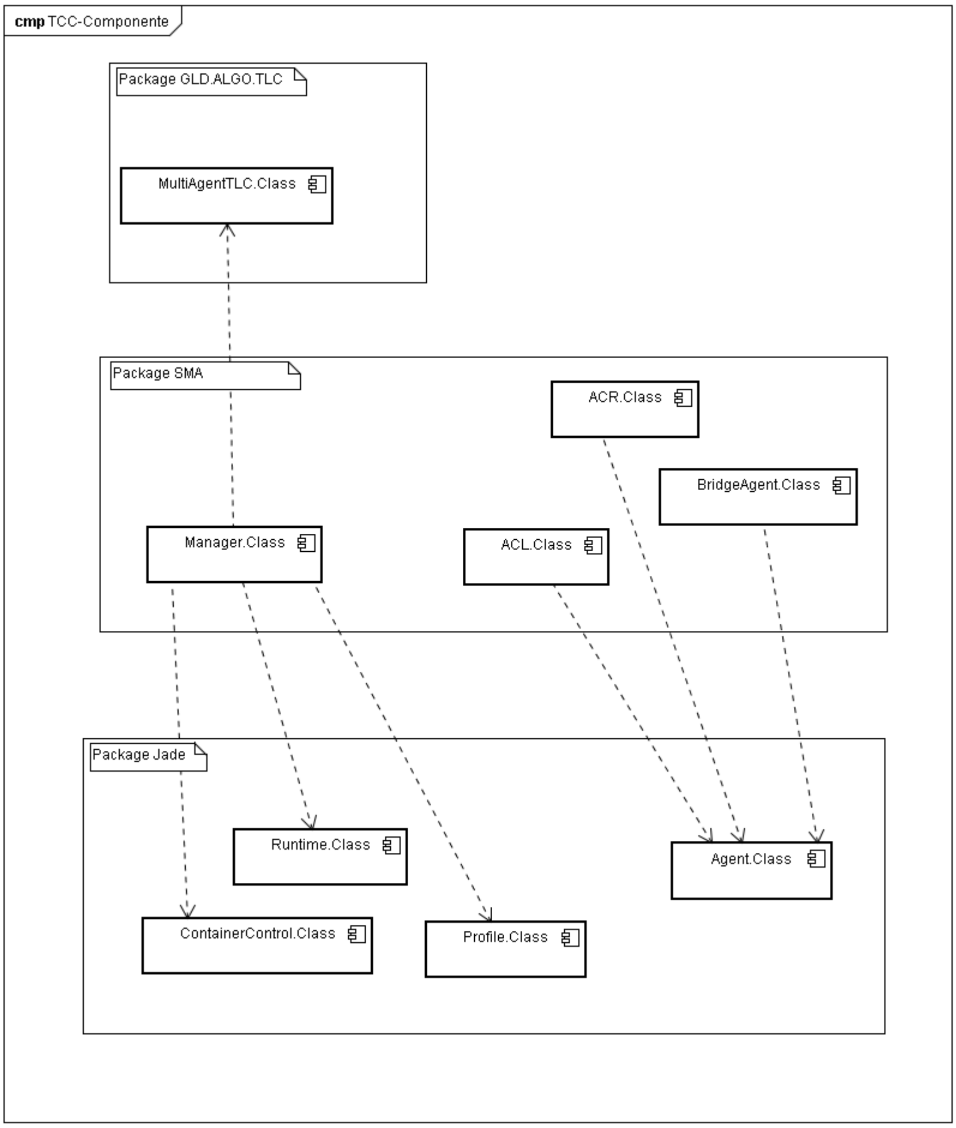


Figure 3: Prototype macro view

These three environments are placed in a single Java ™ project created within the development IDE Netbeans ®. The description and explanation of the structure are divided into the following three sections, and are represented by the three components of the system.

## 4.2 Green Light District - GLD

Composed by modeling and representation classes of traffic infrastructure, graphical interface and statistical simulations, is responsible for providing the environment and the base to other traffic related classes.

Internal structure of the GLD provides a class that acts on the entire set of traffic lights by changing its status, thus avoiding the need to iterate over each light to change their signs.

This class is the TLController and is inherited by MultiAgentTLC from the prototype, thus fulfilling the connection point between the simulator and the system.

For general control over traffic lights of the structure, the TLController has a matrix composed of TLDecision objects, where there are two connected values: traffic light and the float value Q. The traffic light is a class of internal infrastructure and it is only logical. The Q value is a float that is used to determine whether the traffic light will enter the green state, this value is analyzed by GLD and the only relevant detail is that the larger it is greater is the possibility to open the traffic light.

Then MultiAgentTLC receive the original matrix, give it to MAS, which changes it accordingly to agents decisions and gave it back to MultiAgentTLC that returns it to GLD, which in turn will make arrangements to view the values and determine the next state of the environment. Note the relationship with the class diagram in Figure 4.

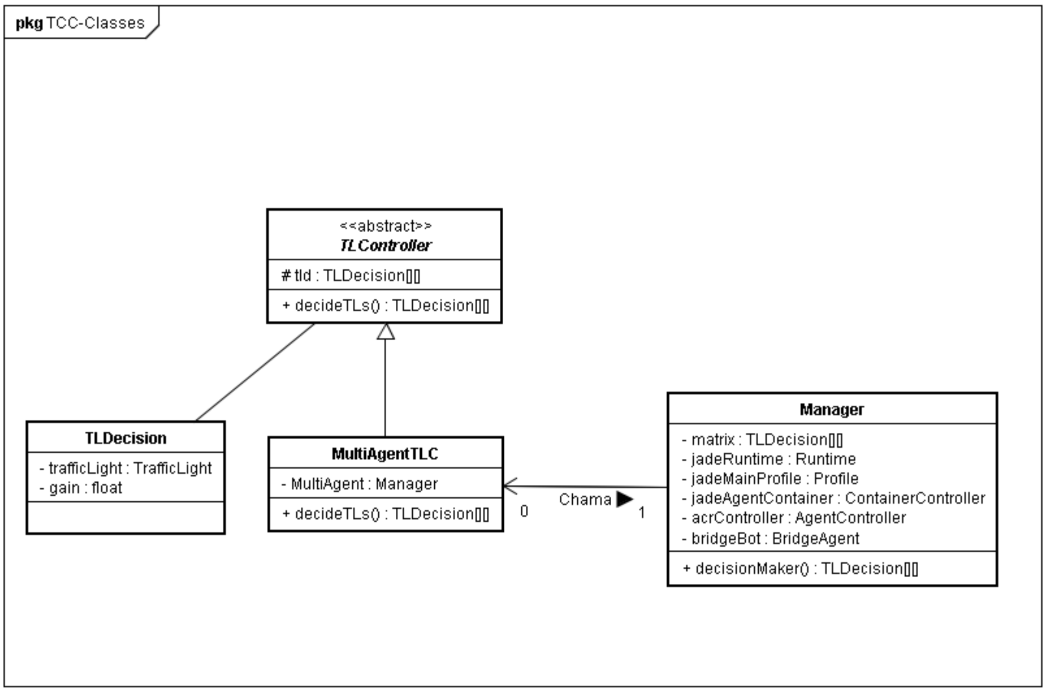


Figura 4: MAS connection to GLD through TLController

It not only help in the implementation, it is also used to perform tests in order to obtain results for comparison with other algorithms of traffic light control. This comparison is carried out by the amount of users who reached your destination and the average journey time, which is recorded within a time interval of light cycles, a cycle is a change of state of the traffic light. These values are obtained during the system execution through graphics and the statistics windows in GLD.

### *4.2.1 MAS Enviroment - The road infrastructure model*

Prototype environment is the set of roads in a region and is represented virtually trough GLD. Each intersection is represented by node and the roads are the connections between those nodes. Each node can contain up to 16 traffic lights and the double of lanes in each.

There are two types of intersection nodes: the entry and/or exit node, which injects cars into the system and receive cars from it; the intersection node itself, which connects roads to another intersection node or entry/exit node and may contain traffic lights. This model is present internally in GLD, as exemplified in Figure 5.

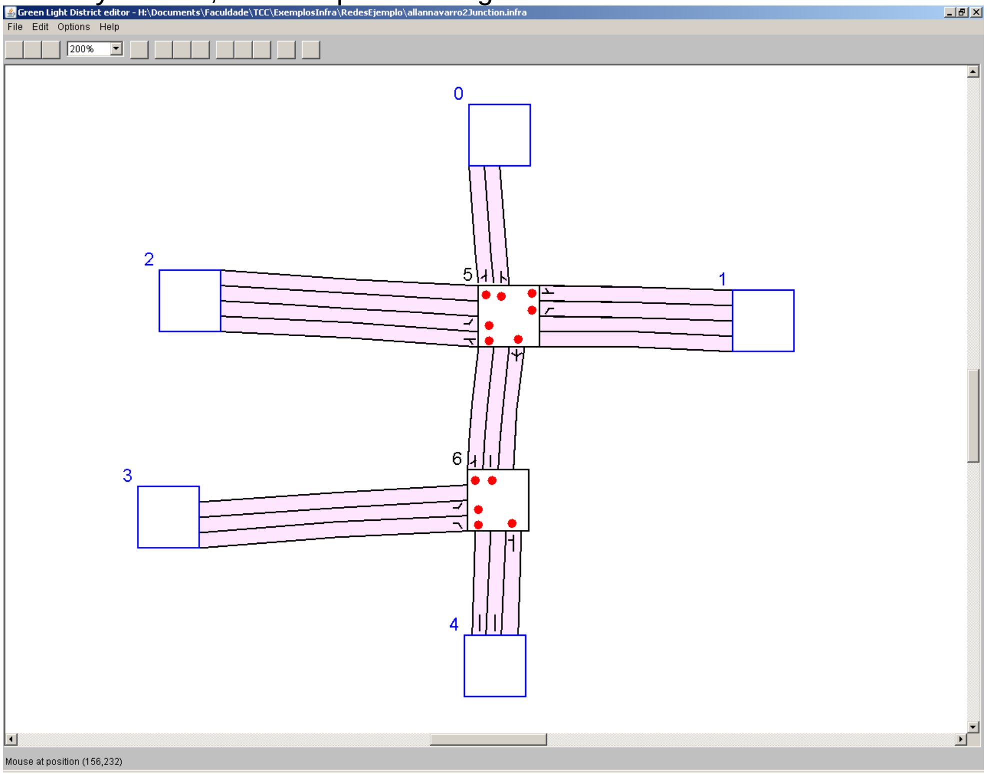


Figure 5: The blue nodes are entry and/or nodes and the black ones are intersection nodes, the red dots are traffic lights

So, some structure that is willing to represent a complete environment for the prototype, is composed of at least one entry node and one exit node and one intersection node between them.

In this environment each traffic light position, red dots, is equivalent to an LCA and there is an RCA coordinating the optimization of the whole environment, and as said before they are connected to this environment trough MultiAgentTLC and the BridgeAgent classes.

### 4.3 JADE Framework

The action and communication interfaces between intelligent agents, as well it registration and life cycle are provided by JADE. The host is an environment that initiates the agents cyclic routine and keep them running while they are active or they do not order to stop.

Instead of using the graphical interface or run framework completely, your library is imported in the project and the classes that activate the host: Runtime, Profile and ContainerController, are instantiated by the prototype class Manager. So the management would be done with the JADE GUI is done through the Manager, which will be detailed in next section.

An agent register is used since no agent can directly access the other. This registry is called yellow pages, it stores the identification of all the agents were added to the host to be accessed through communication, the FIPA messages.
This identifier is represented by the AID class, which keeps the name and the address in the format <name>@<plataform-name>, this could be used to access agents from another computer in the network.
In agents creation the use of Agent class was necessary for three reasons: it makes possible the insertion of agents into the host and into yellow pages; enables the implementation of actions (behaviors); and enables communication via messages.
These actions, called behaviour, are methods implemented according to what you want the agent to make, and it can be executed in different ways. Among the various types offered by JADE, the following were used: Ticker Behaviour, Cyclic Behaviour and Parallel Behaviour.
Ticker is a behavior that occurs from time to time, where this time is set by the agent in its initialization. It is used for actions that happen all the time, but which have a spacing of time between executions, for example the case of an agent that stores statistics in the detection of accidents is done sporadically.
Cyclic behavior is similar to Ticker, but it occurs at every step of the program, in this case every traffic light cycle. One example is the LCA itself which needs to look at the traffic information every cycle to make a decision for the next cycle.
These behaviors are executed on their own threads, but in the effort for acquiring better performance all of them are grouped into Parallel behaviors.
It is not enough just to create behaviors for an agent if they are unable to make any exchange of information, this being one of the details that makes a truly intelligent agent in the MAS, the ability to communicate with other agents in their environment.
When performing the inheritance of Agent class MAS agents have access to sending and receiving messages within the host. This functionality is implemented by the class LCAMessage which is responsible for sending, receiving and its content.

## 4.4 Multi-Agent System

MAS domain starts in its connection with GLD trough MultiAgentTLC and Manager. The Manager class is responsible for: instantiating and starting the host, LCA's and RCA's; and receive and respond to requests from MultiAgentTLC and from agents. See the class structure in figure 6.

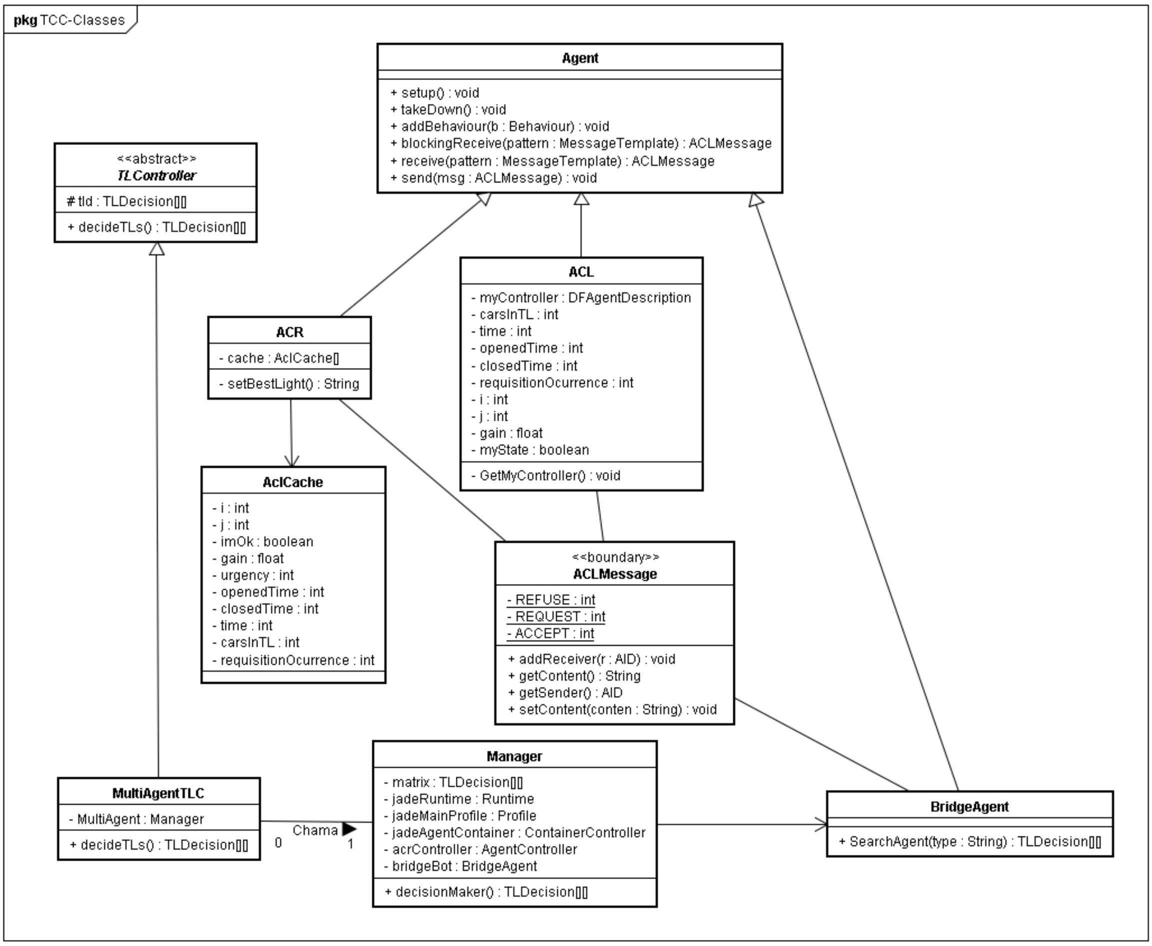


Figura 6: MAS and GLD integration classes diagram

The amount of agents that is registered on the host is the same number of traffic lights in GLD infrastructure model, and that value is given according to the columns of each row of the matrix TLDecision.

With the host and the agents created Manager's task is to respond to requests from MultiAgentTLC about the decision matrix and requests from agents regarding traffic information.

This information is stored by GLD and could be accessed through MultiAgentTLC, so to pass it from GLD to the agents it was needed another agent, the BridgeAgent class.

By the following facts BridgeAgent is necessary, instantiated and stored in the Manager:

- A class, agent or not, cannot communicate to other agent without messages;
- No other class can use the messaging feature unless it inherits from *Agent*, JADE interface for agents;
- The class must be registered in the host, and.
- There is no other way to pass information directly to agent unless it is kept a reference to it.

He doesn't belong to MAS structure, because it's not an intelligent agent, is needed to pass messages from outside the host into it without destroying MAS concepts.

Using Bridge, Manager is capable of communicate with LCA's and RCA's to send and receive messages regarding all the needed information from GLD.

As mentioned before, to represent an agent and have access to the features offered by JADE, LCA class inherits from Agent class. All of them implement the initialization method called setup, in which they register into yellow pages, create whatever is necessary for its existence and add behaviors to its list.

Four behaviors were modeled to perform in accordance with agents decision: sending collected traffic information; GLD requested information; green and red time counting; request for state (green and red) exchange. The calculations are performed on fourth behavior through mathematical formula described in chapter 3.3.

The RCA also inherits from the Agent and has 3 behaviors to be used in accordance with its decision: sending traffic overall results of its region; control of incoming requests to open a traffic light; filling LCACache with LCA's variables and information.

In respect of behaviors responsibilities, the 1st is to make the global calculations and return it to the Bridge, the 2nd is to receive requests from the LCA and answer them properly and the 3rd is to ask each of LCA's your information at any given moment and populate them in the cache. You can view the inheritance and behavior in the class diagram of figure 7.

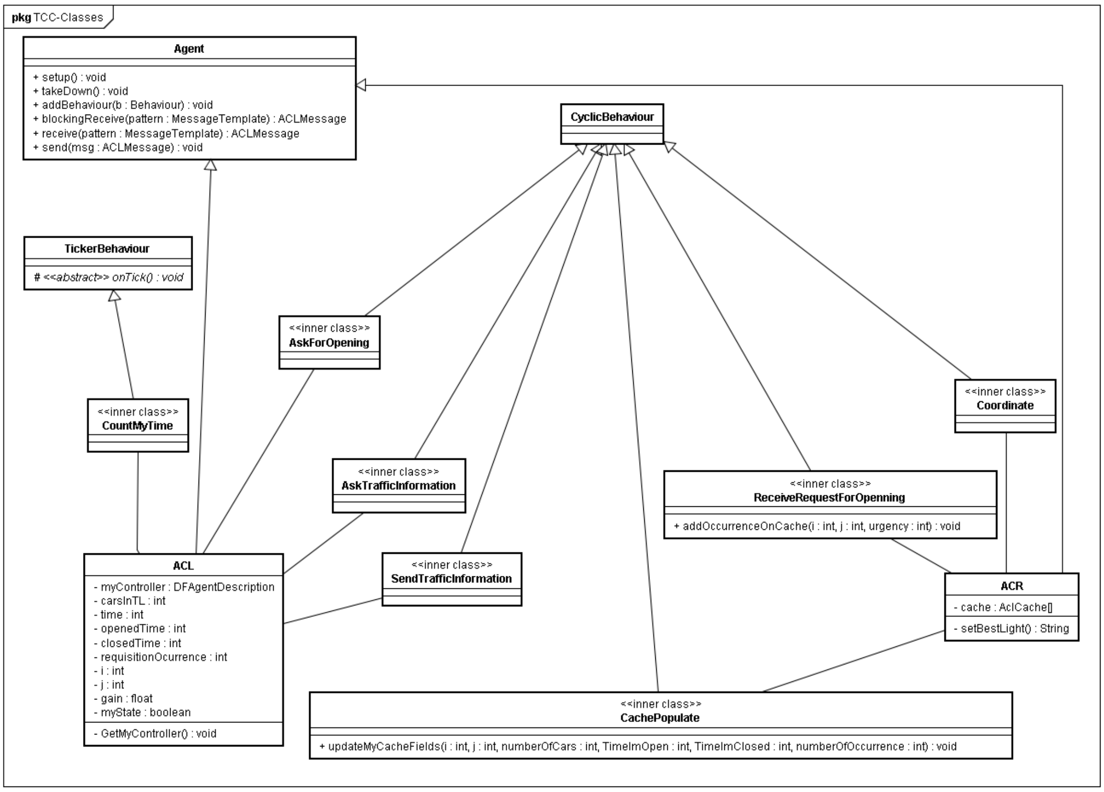


Figure 7: Class Diagram

To avoid the delay caused by direct and constants request for information to the LCA, it was necessary to create a class that keeps this information centralized, in this case the already mentioned cache. Cache is implemented by LCACache, which maintains the latest information of all LCA's and is maintained and updated by the RCA, in addition it prevents situations of lack of information for delay or failure to reply from one of LCA's.

# 5. RESULTS

Aiming to test the prototype and its efficiency, and to compare it to different solutions proposal, some simulations were performed. It was two tests, A and B with 10 runs of 10,000 cycles each, where the output nodes generate 0.25 cars and 0.15 bus per cycle for first one and 0.35 cars and 0.15 bus per cycle for test B. Test B has a traffic set more intense than test A.

These tests were applied to two infrastructure model, both were borrowed from Wiering (2004). The choice is deliberate, since both had comparative analysis and through them showed good ability to test different solutions. The difference between the tests presented here and those of Wiering (2004) is in the setup and sampling.

Mesh one (1) and mesh two (2) are the names from the two mentioned infrastructures and are depicted in Figures 8 and 9 respectively. Both meshes were created in GLD own editor.

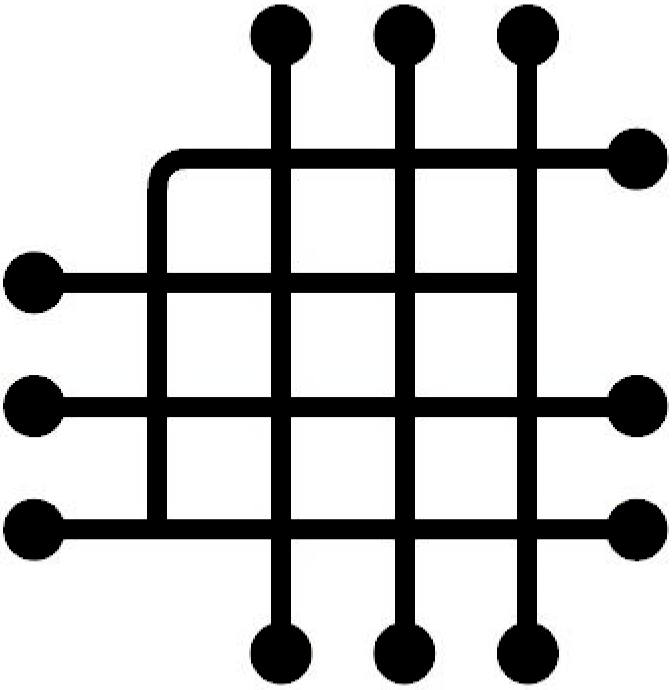

Figure 8: Mesh 1 (WIERING 2004)

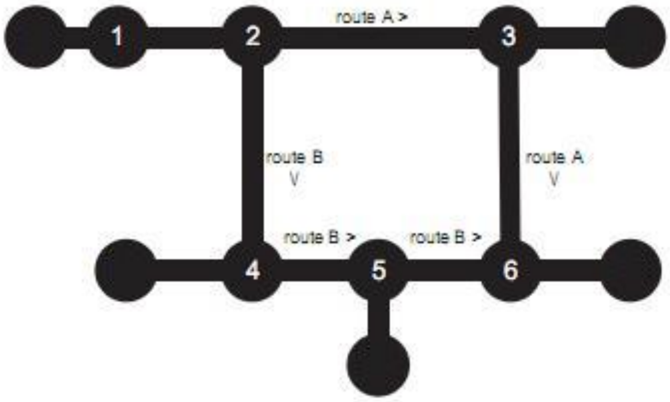


Figure 9: Mesh 2 (WIERING 2004)

Besides the infrastructure is necessary define the other traffic light controllers, among those available in GLD, with the purpose of comparing these solutions and this proposal. From the possibilities the following were chosen: TC1, Relative Longuest Queue, Best First and ACGJ3. Their solutions are explained in Appendix A.

## 5.1 Tests and results

With meshes and traffic light controllers defined it is possible to begin simulations. At each simulation's end, the information provided by GLD were separated and stored.
From this group of information the ones used for comparison are: ATWT, which is the average trip waiting time; users arrived, which is the total number of users that reached destination. Following are two tables, 1 and 2, which contains the results of the controllers at test A.

Table 1: Mesh 1, test A

| Mesh 1 Test A | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | TC1 | | ACGJ3 | | Best First | | Relative Longuest Queue | | SMA | |
| | atwt | users arrived | atwt | users arrived | Atwt | users arrived | atwt | users arrived | atwt | users arrived |
| | 4,71 | 36567 | 3,91 | 36816 | 4,07 | 35908 | 4,82 | 35952 | 150,10 | 4236 |
| | 4,80 | 42773 | 3,76 | 35838 | 4,14 | 39529 | 5,01 | 36188 | 160,60 | 4543 |
| | 4,90 | 38907 | 3,76 | 35822 | 4,19 | 36155 | 4,85 | 35668 | 94,80 | 5589 |
| | 4,83 | 37910 | 3,78 | 40936 | 4,08 | 35753 | 4,82 | 35700 | 87,32 | 6949 |
| | 5,88 | 35643 | 3,78 | 38304 | 4,06 | 35374 | 5,10 | 36176 | 106,54 | 5936 |
| | 4,70 | 35294 | 3,80 | 35795 | 4,30 | 35593 | 4,90 | 35705 | 127,93 | 8004 |
| | 4,91 | 35766 | 3,80 | 35906 | 4,10 | 35774 | 4,90 | 35901 | 147,49 | 4059 |
| | 4,81 | 36110 | 3,80 | 35769 | 4,10 | 35669 | 4,91 | 35656 | 108,25 | 6074 |
| | 4,74 | 35707 | 3,72 | 35706 | 4,00 | 35629 | 4,90 | 35896 | 88,84 | 4400 |
| | 5,24 | 35511 | 3,80 | 35897 | 4,15 | 36069 | 4,83 | 35658 | 137,54 | 5753 |
| Average | 4,95 | 37019 | 3,79 | 36679 | 4,12 | 36145 | 4,90 | 35850 | 120,94 | 5554 |
| Deviation | 0,36 | 2331,71 | 0,05 | 1699,30 | 0,08 | 1210,91 | 0,09 | 207,31 | 27,21 | 1279,39 |

Table 2: Mesh 2, test A

| | Mesh 2 Test A | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | TC1 | | ACGJ3 | | Best First | | Relative Longuest Queue | | SMA | |
| | atwt | users arrived | atwt | users arrived | atwt | users arrived | atwt | users arrived | atwt | users arrived |
| | 1,3 | 15013 | 1,20 | 14968 | 1,30 | 14936 | 1,46 | 14861 | 2,63 | 15085 |
| | 1,31 | 14987 | 1,13 | 15152 | 1,31 | 15009 | 1,49 | 16810 | 2,47 | 17482 |
| | 1,81 | 15141 | 1,21 | 14585 | 1,34 | 16168 | 1,43 | 15002 | 2,51 | 14926 |
| | 1,25 | 16056 | 1,17 | 15050 | 1,26 | 14899 | 1,42 | 15434 | 2,59 | 15015 |
| | 1,3 | 16761 | 1,18 | 14976 | 1,34 | 15160 | 1,45 | 14995 | 2,4 | 15326 |
| | 1,2 | 14904 | 1,20 | 15028 | 1,35 | 14920 | 1,44 | 14931 | 2,6 | 15005 |
| | 1,20 | 14869 | 1,20 | 15078 | 1,30 | 14640 | 1,40 | 14918 | 2,5 | 14775 |
| | 1,21 | 14956 | 1,15 | 14899 | 1,34 | 14914 | 1,40 | 14796 | 2,5 | 14966 |
| | 1,23 | 14846 | 1,20 | 14809 | 1,31 | 14864 | 1,43 | 14941 | 2,8 | 14825 |
| | 1,3 | 14937 | 1,20 | 14982 | 1,30 | 14941 | 1,44 | 14738 | 2,6 | 14682 |
| Average | 1,31 | 15247 | 1,18 | 14953 | 1,32 | 15045 | 1,44 | 15143 | 2,56 | 15209 |
| Deviation | 0,18 | 639,65 | 0,03 | 160,18 | 0,03 | 414,89 | 0,03 | 615,15 | 0,11 | 818,37 |

The first data presented by the prototype demonstrated MAS inefficiency for the normal traffic intensity (test A).

In tests of the mesh 1, MAS has an average score of 120.94 on ATWT, while the highest average among the other drivers was that of TC1 of 4.95. These values began to indicate the inability to handle large meshes by the current implementation.

In mesh 2 tests, MAS has an average score of 2.56 on ATWT, and this case the value is only 1 cycle greater than all other controllers. This is a hint that, although getting worse results than the other solutions, the prototype seems to get better results on smaller infrastructure in relation with mesh 1.

Figure 10 demonstrates the closeness and lower deviation of values from controllers in the mesh 2.

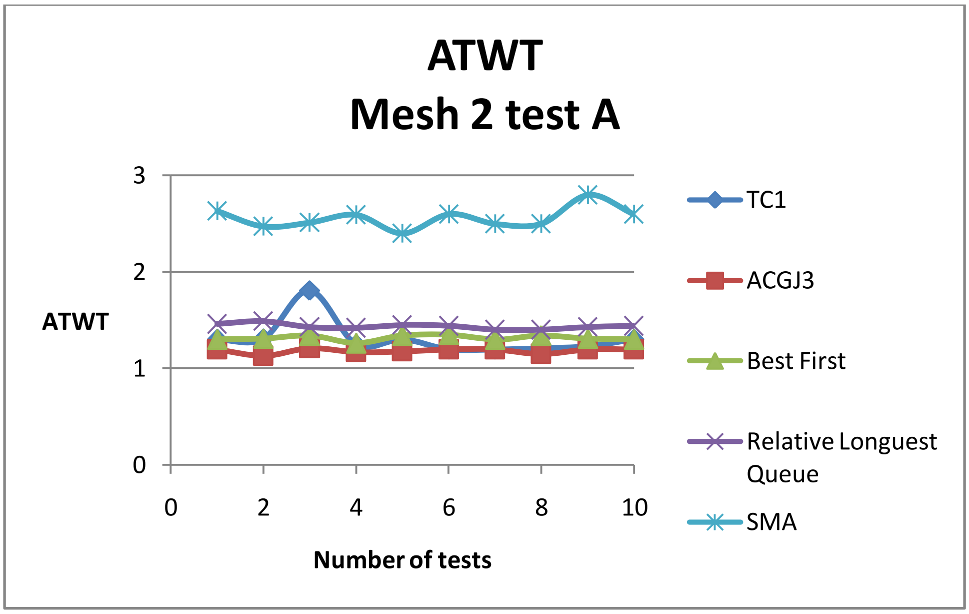


Figure 10: ATWT - Mesh 2 - Test A

The following are the tables 3 and 4, which contains the results of controllers in test B.

Table 3: Mesh 1, test B

| Mesh 1 Test B | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | TC1 | | ACGJ3 | | Best First | | Relative Longuest Queue | | SMA | |
| | atwt | users arrived | atwt | users arrived | atwt | users arrived | atwt | users arrived | atwt | users arrived |
| | 73,20 | 5093 | 2384,13 | 2802 | 62,53 | 4215 | 27,74 | 1525 | 188,20 | 2748 |
| | 67,03 | 4599 | 1709,15 | 3399 | 73,58 | 6217 | 41,10 | 2165 | 144,13 | 2642 |
| | 94,66 | 4520 | 963,28 | 3744 | 47,11 | 3347 | 23,23 | 1334 | 105,24 | 2689 |
| | 56,47 | 4490 | 765,70 | 2782 | 55,95 | 4002 | 40,83 | 1720 | 134,00 | 2712 |
| | 87,23 | 3090 | 503,79 | 2708 | 72,49 | 5483 | 79,10 | 2185 | 42,46 | 2656 |
| | 88,50 | 3760 | 488,46 | 3460 | 60,12 | 4941 | 67,50 | 1881 | 36,54 | 1821 |
| | 74,69 | 5925 | 547,30 | 4104 | 54,44 | 2777 | 72,97 | 2583 | 97,74 | 3472 |
| | 38,53 | 2604 | 318,34 | 2956 | 48,04 | 3481 | 69,02 | 2094 | 100,03 | 3096 |
| | 92,06 | 5471 | 966,36 | 3878 | 65,50 | 1981 | 48,20 | 1518 | 177,15 | 2529 |
| | 58,90 | 5171 | 600,99 | 3451 | 72,97 | 2593 | 38,69 | 1646 | 54,26 | 3546 |
| Average | 73,13 | 4472 | 924,75 | 3328 | 61,27 | 3904 | 50,84 | 1865 | 107,98 | 2791 |
| Deviation | 18,18 | 1049,26 | 646,03 | 495,86 | 9,92 | 1341,68 | 19,84 | 386,95 | 53,40 | 493,88 |

Table 4: Mesh 2, test B

| Mesh 2 Test B | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | TC1 | | ACGJ3 | | Best First | | Relative Longuest Queue | | SMA | |
| | atwt | users arrived | atwt | users arrived | atwt | users arrived | atwt | users arrived | atwt | users arrived |
| | 8 | 7685 | 603,69 | 16059 | 31,23 | 26457 | 8,27 | 6338 | 338,3 | 21585 |
| | 241,62 | 22215 | 584,78 | 17021 | 160,95 | 24334 | 7,35 | 8630 | 374,16 | 20434 |
| | 465,14 | 19171 | 350,34 | 18928 | 129,52 | 24664 | 11,60 | 1585 | 334,1 | 23830 |
| | 364,777 | 18551 | 468,08 | 9438 | 33,47 | 24916 | 8,58 | 3098 | 314,71 | 22038 |
| | 178,9 | 23522 | 694,19 | 15976 | 160,50 | 21533 | 21,05 | 1686 | 328,1 | 25027 |
| | 332,38 | 21387 | 111,71 | 19390 | 107,15 | 18733 | 8,26 | 24736 | 423 | 18000 |
| | 333,82 | 21615 | 85,80 | 6621 | 23,73 | 24748 | 10,46 | 9795 | 402,11 | 21715 |
| | 308,4 | 20904 | 35,54 | 7161 | 44,52 | 24044 | 7,10 | 13458 | 342,76 | 17357 |
| | 227,20 | 22628 | 174,90 | 22489 | 106,42 | 24157 | 8,97 | 4586 | 375,78 | 27964 |
| | 393,44 | 18515 | 1346,19 | 5801 | 254,07 | 19340 | 9,44 | 21713 | 359,69 | 21926 |
| Average | 285,37 | 19619 | 445,52 | 13888 | 105,16 | 23293 | 10,11 | 9563 | 359,27 | 21988 |
| Deviation | 128,65 | 4530,98 | 395,69 | 6071,82 | 74,43 | 2550,67 | 4,08 | 8141,80 | 34,47 | 3133,15 |

In test B, where the traffic is considered intense, none of the controllers obtained low ATWT values. Furthermore the Relative Loguest Queue was invalidated for that test because it was the only one who failed to complete a single simulation without causing traffic congestion overall, where no vehicle was moving, or completely halt the execution of the simulator.

The prototype showed a better control on mesh 2, what is confirmed by it low oscillation in ATWT values at both tests. The ATWT variation in test B is better understood by observing the figures 11 and 12.

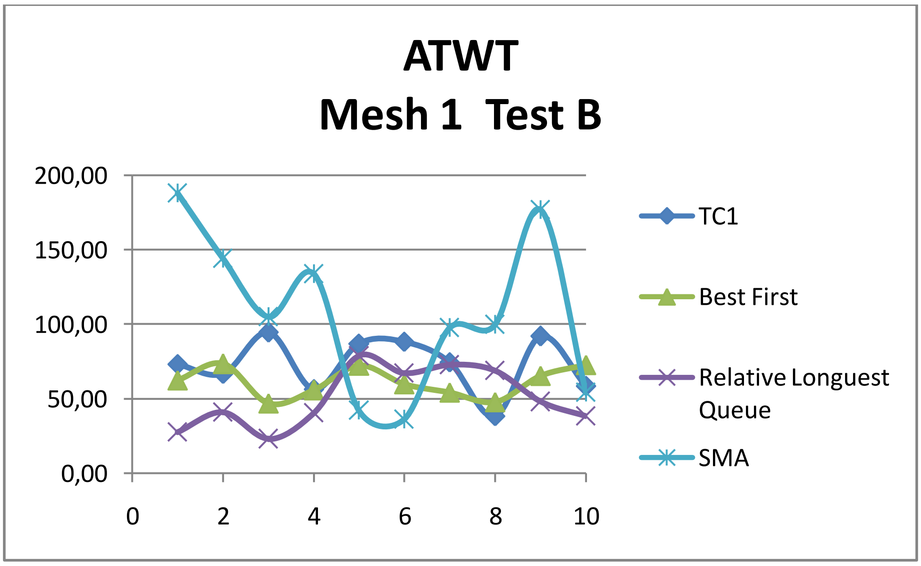


Figure 11: ATWT - Mesh 1 - Test B

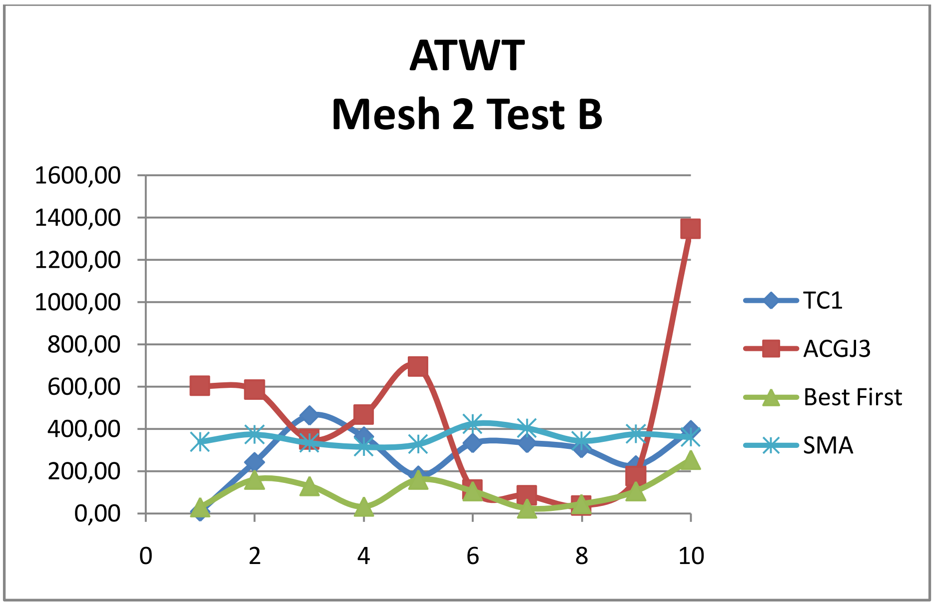


Figure 12: ATWT - Mesh 2 - Test B

Through these results was identified a number of future studies from the presented solution which are discussed in Chapter 7.

## 6. CONCLUSIONS

This work showed the possibility of creating intelligent traffic lights using artificial intelligence. We then defined the concepts necessary to create a prototype that utilizes the interaction and collaboration of multiple agents in order to organize and improve traffic flow in general. This prototype then used the concept of multi-agent systems, where each agent has been appointed as an intelligent traffic light (LCA) and operations control center were appointed as region control agents (RCA). Where the scope of LCA's is the street where the represented traffic light is positioned and RCA is the set of traffic lights that have been defined as a region (e.g. city or neighborhood).

From this the focus was to develop the prototype, which presented the components, their structures and connections. Important to note that the prototype is composed of three parts (GLD, JADE and MAS) and that the process of connection and interaction between them is of utmost importance for the operation and opening of the proposal for future studies.

For the generation of results GLD simulation capability was used and two test were described, A and B, which settings and the controllers were determined to be compared. The simulations showed that the way the prototype was developed there was not some considerable improvements in respect to the algorithms presented and that in the scenario with better efficiency (mesh 2, test B) prototype showed only an increase of 1 cycle in relation to others.

The scenario mentioned above leads to the conclusion that in smaller meshes and with the right settings and modifications the prototype can achieve certain improvements in the ATWT values with MAS.

It is noteworthy that compared controls are not implemented in real systems and it is important to create controllers similar to current use to obtain efficiency results closer to reality and define if it's better than actual traffic control systems.

Although it not achieved the expected performance, in general the prototype demonstrates the feasibility and space to create intelligent traffic control capable of working in some social way to achieve a common goal. This is because of its interoperability and architecture.

# 7. FURTHER RESEARCH

From the results of this paper is possible to indicate some possible studies following the same proposal.

One of the works is the study that is needed to verify the optimum size of mesh in which MAS should act and what the best configuration of hierarchy levels for each size of mesh.

Furthermore the implementation of agents and calculations to optimize the prototype also shows important work that can be continued, since there is huge potential for inclusion of different calculations given the ease of changing them.

Continuing along the line of change in the calculation, it is possible to study the use of priority weight for vehicles and roads and the addition of other information in choosing the best action from each agent. This weight could be used by the RCA to provide the best configuration of green and red states in the region of its control.

The compared controls are algorithms that are not distributed and deployed as a standard in many cities, this can generate poor results with respect to algorithms that in certain situations seems be perfect, so the job of creating a controller that resembles those used today is of great importance not only for this but for all other similar works.

Finally, there remains the possibility of creating intelligent agents for vehicular devices that communicate with the prototype providing more information to the calculations and allowing predictions of traffic situation in real time.

# REFERENCES

Balan, G. & Luke, S. History-based traffic control, in 'AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems', ACM, New York, NY, USA, p. 616-621, 2006.

Balmer, M.; Cetin, N.; Nagel, K. & Raney, B. Towards Truly Agent-Based Traffic and Mobility Simulations, in 'AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems', IEEE Computer Society, Washington, DC, USA, p. 60-67, 2004.

Bordini, R. H.; Hübner, J. F. & Wooldridge, M. Programming multi-agent systems in AgentSpeak using Jason, Wiley, 2007.

France, J. & Ghorbani, A. A. A Multiagent System for Optimizing Urban Traffic, in 'IAT '03: Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology', IEEE Computer Society, Washington, DC, USA, p. 411, 2003.

Jin, X.; Itmi, M. & Abdulrab, H. A cooperative multi-agent system simulation model for urban traffic intelligent control, in 'SCSC: Proceedings of the 2007 summer computer simulation conference', Society for Computer Simulation International, San Diego, CA, USA, p. 953-958, 2007.

Roozemond, D. A. Using intelligent agents for urban traffic control control systems, in 'In Proceedings of the International Conference on Artificial Intelligence in Transportation Systems and Science', 1999.

Roozemond, D. A. & Rogier, J. L. H. Agent controlled traffic lights, in 'Proc. ESIT2000', 2000.
Russel, S. J. & Norvig, P.International, P. E., ed. Artificial Intelligence, A modern Approach - 2nd Edition, Prentice Hall, 2002.
Wahle, J. & Schreckenberg, M. A multi-agent system for on-line simulations based on real-world data, in 'In Proc. of the Haiwai International Conference on System Science (HICSS). IEEE Computer Society', 2001.
Wiering, M.; Veenen, J. V.; Vreeken, J. & Koopman, A. 'Intelligent traffic light control', Technical report, ERCIM News, European Research Consortium for Informatics and Mathematics 53, 2004.
Winikoff, M. & Padgham, L. Developing Intelligent Agent Systems: A Practical Guide, Halsted Press, New York, NY, USA, 2004
Wooldridge, M. An Introduction to Multiagent Systems, Wiley, Chichester, England, 2002.
Schepperle, Heiko; Böhm, Klemens & Forster, Simone Towards valuation-aware agent-base traffic control, IN AAMAS '07: Proceedings of the 6th internacional joint conference on Autonomous agents and multiagent systems, ACM, New York, NY, USA, P 1-3, 2007.
GLD, THE Green Light District Project. Last access was in Monday, June 20, 2009. http://www.students.cs.uu.nl/swp/2001/isg/
WHAT IS ARTIFICIAL INTELLIGENCE?, site de John McCarty. Revised in November 12, 2007
Last access was in Sunday, November 1, 2009.
http://www-formal.stanford.edu/jmc/whatisai/whatisai.html

# APPENDIX A

Here are presented the descriptions of traffic light controllers, contained in the GLD, which were used to compare to this paper solution prototype.

## TC1

It is the controller created by Wiering (2004). It uses algorithms based on reinforcement learning2 and through them is able to compute the gain values (Q) setting two traffic lights to green state and choosing the configuration with the highest of these values.
To generate the gain value it compares the desired waiting time while traveling in your state red and green for the users of the road in question, so it is also necessary to have the travel information of each user.

## Relative Longuest Queue

He chooses the longest line on a set of tracks in an intersection. The largest relative queue is the queue with the maximum ratio between the number of users on hold and queue size. As

[2] "... in computer science, reinforcement learning is a sub-area of machine learning concerned with how an agent ought to take actions in an environment so as to maximize some notion of long-term reward..." http://en.wikipedia.org/wiki/Reinforcement_learning

roads filled completely take priority over others where there are many people waiting, but are not completely full, that controller possibly avoid traffic congestion.

## Best First

This controller chooses a configuration option of traffic lights, green and red states, from an intersection where the largest number of vehicles waiting can continue their journey. What the algorithm does is tell how much vehicles can move according to each combination of traffic lights in green and red state at some intersection and choose one in which this amount is as large as possible.

## ACGJ3

The gain value (Q) is calculated by summing the multiplication of the weight of each user on hold by a factor of length, where the weight of each user is given, for example, by the number of passengers or a fixed value. Also, road users who arrive first are valued when the factor is less than 1 and the size of the queues are valued exponentially when the factor is greater than 1.