CONOPS
Concept of Operations

Project: CTA Simulator

Authors: Douglas P. B. Renaux
Robson Ribeiro Linhares
Jean Marcelo Simão
Revision: Paulo Cézar Stadzisz

Version: 18-Jun-2014

Version	Date	Authors	Description
0.1	5-Feb-2014	Douglas	Proposed structure and contents
0.2	17-Feb-2014	Robson	Initial proposal for semaphore time adjustment policy (Section 5.1)
0.3	17-Feb-2014	Douglas	Concepts, Platform
0.4		Robson	Section 7
0.5	26-Fev-2014	Douglas	Physical scenario 1, document review
0.6	27-Fev-2014	Douglas	Physical scenario 2
0.7	06-Mar-2014	Simão	Document Review. Comments concerning Contents. Adjustment in Text Format.
0.8	?	Robson	?
0.9	09-Mar-2014	Douglas	Section 3 back to paragraph form; Section 4 separates definition from restrictions; Section 5.1 – correction to scenarios to avoid congestion; Section 6 platform selection; Section 7.2 – comments to traffic control strategy 2. Section 8 – comments to simulator implementation. Section 9 – added new section with log file estimations.
0.91	10-Mar-2014	Simão	Document Review. Comments concerning Contents. Adjustment in Text Format. First Chronogram Draft
0.92	11-Mar-2014	Douglas, Robson, Simão	Changes during final meeting on CONOPS.
0.95		Paulo	Document Review.
0.96	09-Apr-2014	Simão	Figure for System, References to NOP
0.97	22-Apr-2014	Simão	Class diagram updated File format changed to .docx
0.98	23-Apr-2014	All	Revised during meeting on 23-Apr-2014
0.99	06-Mai-2014	Simão	Baseline version Changes to Section 3 UML model as discussed in the meeting of 23-Apr.
1.00	18-Jun-2014	Simão	Changes to Section 3 UML model as discussed in the meeting of 28-Mai.

1 Introduction

1.1 Project Identification

Acronym: CTA Simulator
Title: Traffic Control System Simulator
Description: Development of a Traffic Simulator and Semaphore control strategy

1.2 Glossary

AoA	Area of Actuation – geographical area where the Traffic Light Controller operates
CONOPS	**Concept of Operations** – Document that describes the system´s operation from the user´s perspective, and possibly from the perspective of other stakeholders as well.
COO	Concurrent Object-Oriented Programming
NOP	Notification Oriented Paradigm

1.3 Project Objectives

The objectives of this project are:

1. To develop control strategies for the traffic lights of a given region of a city.
2. To develop a traffic simulator where these control strategies can be tested and evaluated.
3. To compare the performance of these control strategies.
4. To evaluate and compare the complexities of the software development processes when different programming paradigms are used to implement the above objectives; initially the comparison will be among NOP [4] (Notification Oriented Paradigm) and COO (Concurrent Object-Oriented Programming).
5. To develop a simulation player software that plays back a simulation log on a graphical interface.

1.4 Document Scope

The scope of the CONOPS document describes the traffic-system-simulator structure and operations, in a non-formal way, before the Requirements Analysis and Specification activities.

2 Applicable Documents

This section lists the documents that are referenced/consulted at the time the CONOPS document was elaborated.

[1] DoD, Concept of Operations Template (SWTM024). Oct, 2012.

[2] IEEE, IEEE Guide for Information Technology - System Definition - Concept of Operations. Dec, 2007

[3] Nordstrom, Robert - Heathrow Tests Personal Rapid Transit System – Airport Improvement, July 2009.

[4] J. M. Simão, R. F. Banaszewski, C. A. Tacla, P. C. Stadzisz, "Notification Oriented Paradigm (NOP) and Imperative Paradigm: A Comparative Study," Journal of Software Engineering and Applications (JSEA), p.402-416, v.5, n.6, 2012 http://www.scirp.org/journal/PaperInformation.aspx?paperID=19842#abstract – doi 10.4236/jsea.2012.56047.

3 Overview

A Traffic Control System is comprised of a set of traffic lights, vehicle sensors and a traffic lights controller.

Traffic lights are installed in street intersections. In this document, traffic lights refers to a set of three lights (green, yellow, and red) that control the traffic of a single street. Hence, each intersection requires at least two of such traffic lights.

Vehicle sensors are installed along the streets and monitor the amount of vehicles stopped along a given block. The sensors used in this simulator provide indications of 60% and 100% of the vehicle capacity of each block.

The Traffic Lights Controller monitors the sensors and controls the traffic lights in a manner to meet the safety objectives and the performance objectives.

The Traffic Control System has a geographical area of actuation, called Area of Actuation (AoA). Traffic lights cycle through the green, yellow, and red colors to control the movement of vehicles crossing an intersection. To improve traffic performance, the Traffic Lights Controller is able to synchronize the cycle of each traffic light to the cycles of the traffic lights on the intersections around it, if desired.

Safety objectives have higher priority and must guarantee that no two vehicles moving in different directions are allowed into an intersection. To further increase safety, a red-overlap policy is in place that requires that both traffic lights on an intersection are red for a short period of time before one of the traffic light changes to green.

The idea of developing a traffic system simulator was inspired by the ULTra [3] system developed at Heathrow – London, UK.

3.1 Model

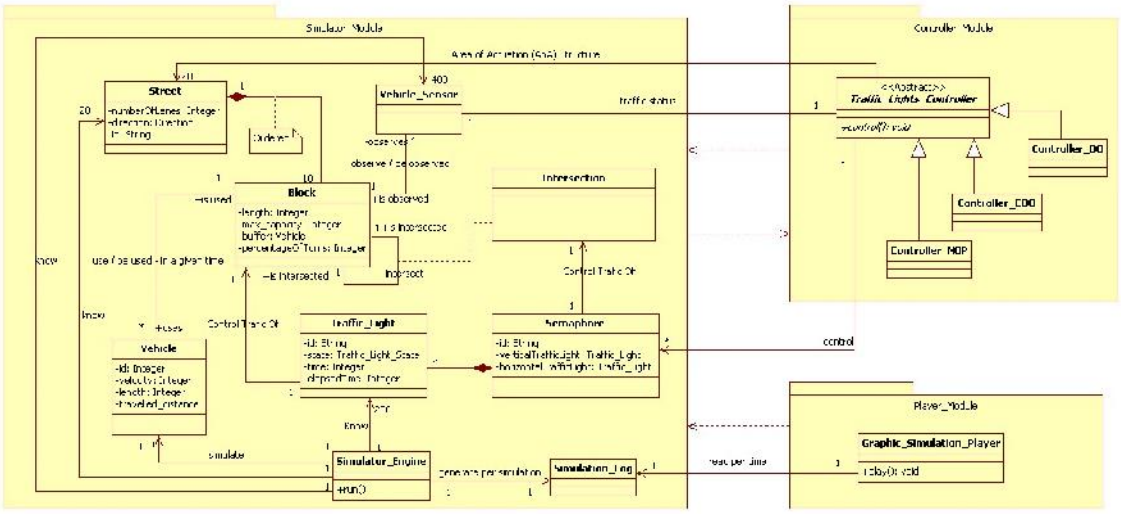
Based on this presented overview and in other parts of this current document, a general class diagram (here below) presents the main entities concerned in the envisaged Traffic Control System. Essentially, there are three packages related to the Player Module, Controller Module, and Simulator Module.

In the Player Module there is a class related to the Graphic_Simulation_Player whose instance must play the simulated system based on a simulation log from the Simulator_Module.

In turn, the Controller_Module presents a generic and abstract class called Traffic_Lights_Controller which define the interaction point with respect to Simulator_Module classes. Still, once it is abstract, the Traffic_Lights_Controller class must be specialized to some sort of specific control class based on a given orientation (OO, OO, NOP, etc). Supposedly, these derived classes would respect the same interface in order to interact with the Simulator_Module.

Finally, the Simulator Module presents the Simulater_Engine class, the model classes, and their relations. The Simulater_Engine instance should be related to instances of the Vehicle, Vehicle_Sensor, Traffic_Light, Street, and Simulation_Log classes. The Simulater_Engine instance generate a Simulation_Log instance per simulation, whereas know the instances of other mentioned classed in order to perform simulation.

Still, each one of the 20 Street instances aggregates 10 Block instances. Each Block instance is related to two Vehicle_Sensor instances (one of 60% occupation and other of 100%), is concerned to Traffic_Light instance, and when intersect other Block Instances constitutes an Intersection instance. Each Intersection instance concerns one Semaphore which, in turn, aggregates two Traffic_Lights. Actually, the each Traffic_Lights_Controller derived instance would control Semaphore instances based on data about Street and Vehicle Sensor Instances.



4 Concepts

Concerning the problem domain, this section lists the relevant concepts, definitions, restrictions, and rules of operations.

Vehicle		
	Definition:	A mobile machine that is used to transport passengers or cargo.
	Restrictions and Rules of Operation	To simplify the simulator, a vehicle is always an automobile. Automobiles are all of the same size. Automobiles move at constant speed.

Lane		
	Definition:	Part of a street, usually identified by painted markings, wide enough for one vehicle to move.
	Restrictions and Rules of Operation	In this simulator all streets are one way, i.e., all lanes of a street have traffic in the same direction.

Block		
	Definition:	Section of a street between two intersections.

Intersection		
	Definition:	Crossing of two streets.
	Restrictions and Rules of Operation	In order to avoid collisions, intersections have either traffic lights or rules that impose the right way for vehicles crossing the intersection. All intersections have traffic lights

Traffic Lights		
	Definition:	A set of three lights (red, yellow, and green) used to control traffic.
	Restrictions and Rules of Operation	Vehicles must stop when the light is red Vehicles are allowed to move (when it is safe to do so) when the light is green. Yellow informs that the traffic light is about to change from green to red. The CTA simulator allows vehicles to cross the intersection during the first 2 seconds after the traffic lights turned yellow.

Semaphore		
	Definition:	The set of two traffic lights in an intersection.
	Restrictions and Rules of Operation	The two traffic lights must be always synchronized to comply to safety regulations. Hence, when one traffic light is not-red the other one must be red

5 Proposed Physical Architecture

The traffic light controller system will control the traffic lights on the intersection of the streets in a given area. The simulator (objective 2 in Section 1.3) consists of an area comprising 10 east-west streets and 10 north-south streets as depicted in **Error! Reference source not found.**. This comprises 100 intersections that form the AoA.

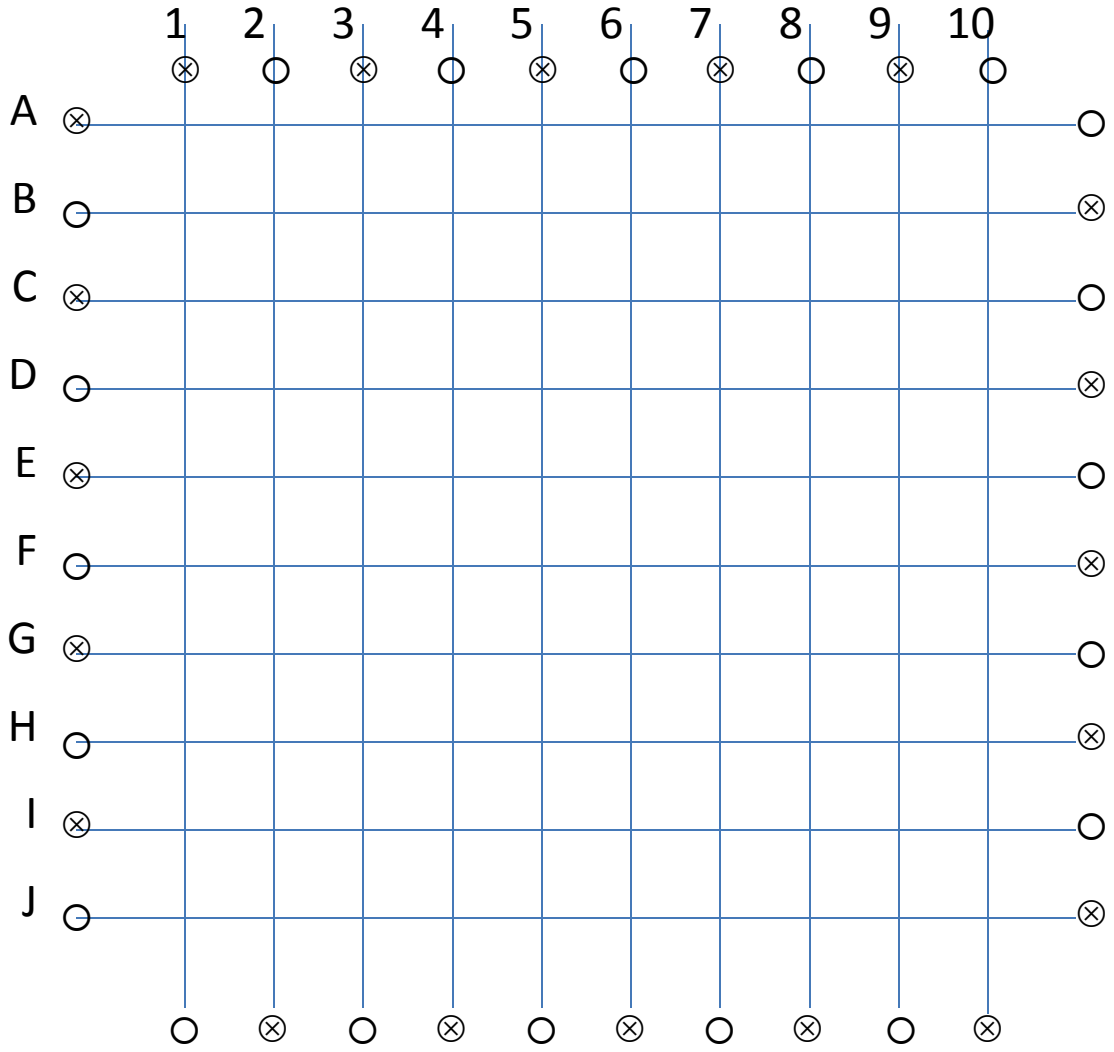


Figure 1- Street Layout and Identification

5.1 Characteristics of the Physical Architecture

The characteristics of the Physical Architecture are the following:

a. North-South streets are identified by 1 .. 10.

b. East-West streets are identified by A .. J.

c. All streets are one-way - attribute Direction of a street is either: N, S, E, W.

d. Streets may have 1, 2, 3 or 4 lanes, (attribute NumberOfLanes of a street).

e. Each block is 100 m in length.

f. In a block´s length there is space for 25 vehicles per lane.

g. A vehicle is either stopped (due to red signal or blocked by the vehicle in front) or moving at a speed of 20 m/s.

h. In each intersection there are 2 traffic lights, one for each street of the intersection. These two traffic lights must be always synchronized: when one is not red the other one is red. The two traffic lights are identified as A1 and 1A. A1 is the traffic light for street A and located in the intersection of streets A and 1.

i. For safety reasons, there is a red-overlap with duration of 1 s just before one of the traffic lights of an intersection turns green.

j. On every intersection, a number of vehicles turn. This is related to the attribute PercentageOfTurns of each traffic signal, where PercentageOfTurns varies from 5% to 35%. To obtains a fair comparison among implementations, each traffic light has a fixed PercentageOfTurns value, previously generated.

k. Vehicles are not allowed to stop in an intersection. If the next block reached its capacity for stopped vehicles then vehicles in the previous block cannot move forward.

l. Vehicles entering the AoA do so in one of the entry points (○) and exit in one of the exit points (⊗).

m. Each intersection has sensors before the traffic lights that monitor the number of stopped vehicles (from 0 to block´s capacity).

n. After a traffic light turns green, every second NL vehicles start moving.

o. The typical total time for the green light is 38s, the typical total time for the yellow light is 5s and the typical total time for the red light is 47s. Therefore the complete cycle time is 90s by taking into account the red-overlap policy.

The state transitions for the traffic lights T1 and T2 of a semaphore S are distributed in time according to

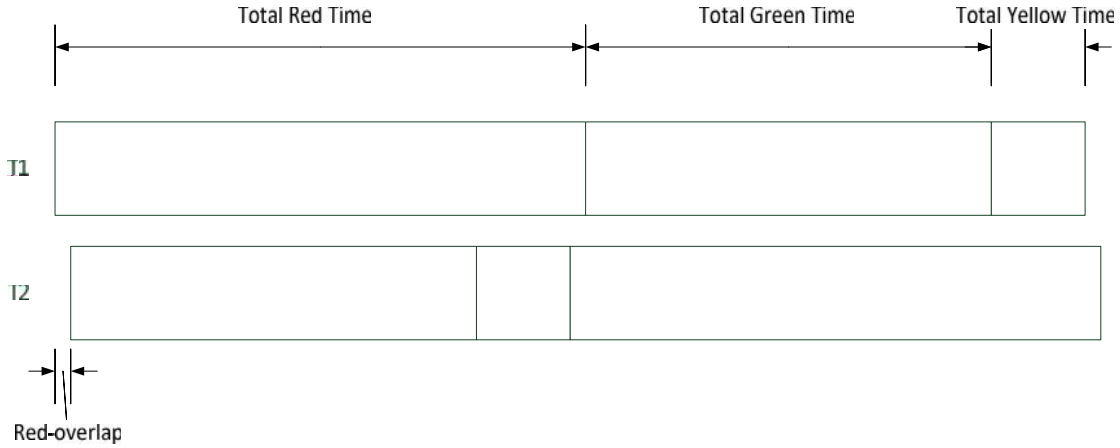


Figure 2 – Timing of the states of a Semaphore.

Physical Scenarios to be used in Simulation

Scenario 1

This is a simplistic scenario that is used to verify the simulation.

Street	Number of Lanes		Street	Number of Lanes
1	1		A	1
2	1		B	1
3	1		C	1
4	1		D	1
5	1		E	1
6	1		F	1
7	1		G	1
8	1		H	1
9	1		I	1
10	1		J	1

All intersections have T = 10% (i.e. 10% of the cars turn when arriving at the intersection).

Possible simulation scenarios to be used with this Physical Scenario

Total simulation time: 2000s (this is the time of the simulated real-time clock; not to be misinterpreted as the processor execution time taken to perform this simulation).

Traffic intensity:

- Constant: during the simulation, each entry point receives 0.1, 0.2, 0.3, 0.4 and 0.5 cars per second. Total of five simulation scenarios.
- Linear growth: start with 0.1 car / entry point / second and increase by 0.1 car / entry point / second every 400 s
- Poisson distribution: requires the elaboration of a table with 20 entry points x 2000 s. Each cell of the table represents the number of vehicles entering one entry point of the AoA at that time. This table is an input to each simulation run so that simulations performed on different platforms can have their performances compared.

Scenario 2

This is a more realistic scenario than the previous.

Street	Number of Lanes		Street	Number of Lanes
1	1		A	2
2	1		B	1
3	1		C	1
4	4		D	1
5	4		E	4
6	1		F	2
7	1		G	2
8	2		H	4
9	2		I	1
10	1		J	2

Value of PercentageOfTurns for traffic on streets 1 .. 10 turning onto streets A..J.

	A	B	C	D	E	F	G	H	I	J
1	10%	35%	10%	10%	10%	10%	15%	10%	10%	35%
2	10%	10%	10%	10%	10%	10%	15%	35%	10%	10%
3	15%	10%	10%	10%	10%	10%	35%	15%	10%	10%
4	20%	20%	20%	10%	10%	10%	10%	10%	10%	20%
5	15%	10%	10%	10%	10%	10%	15%	15%	10%	10%
6	15%	35%	20%	20%	35%	15%	15%	15%	20%	35%
7	15%	15%	10%	10%	10%	10%	10%	35%	10%	15%
8	10%	10%	10%	10%	10%	15%	35%	15%	10%	10%
9	15%	10%	10%	10%	10%	10%	15%	15%	10%	10%
10	20%	15%	35%	20%	20%	10%	10%	10%	20%	35%

Value of PercentageOfTurns for traffic on streets A .. J turning onto streets 1..10.

	1	2	3	4	5	6	7	8	9	10
A	10%	15%	10%	10%	10%	10%	15%	10%	10%	10%
B	10%	15%	35%	10%	10%	10%	15%	10%	10%	10%
C	10%	35%	15%	10%	10%	10%	35%	10%	10%	10%
D	10%	10%	10%	10%	10%	10%	10%	20%	10%	10%
E	10%	15%	15%	10%	10%	10%	15%	10%	10%	10%
F	15%	15%	15%	20%	35%	15%	15%	20%	20%	35%
G	10%	10%	35%	10%	10%	10%	10%	10%	10%	10%
H	15%	35%	15%	10%	10%	15%	35%	10%	10%	10%
I	10%	15%	15%	10%	10%	10%	15%	10%	10%	10%
J	10%	10%	10%	20%	20%	10%	10%	35%	20%	20%

Possible simulation scenarios to be used with this Physical Scenario

Total simulation time: 2000s.

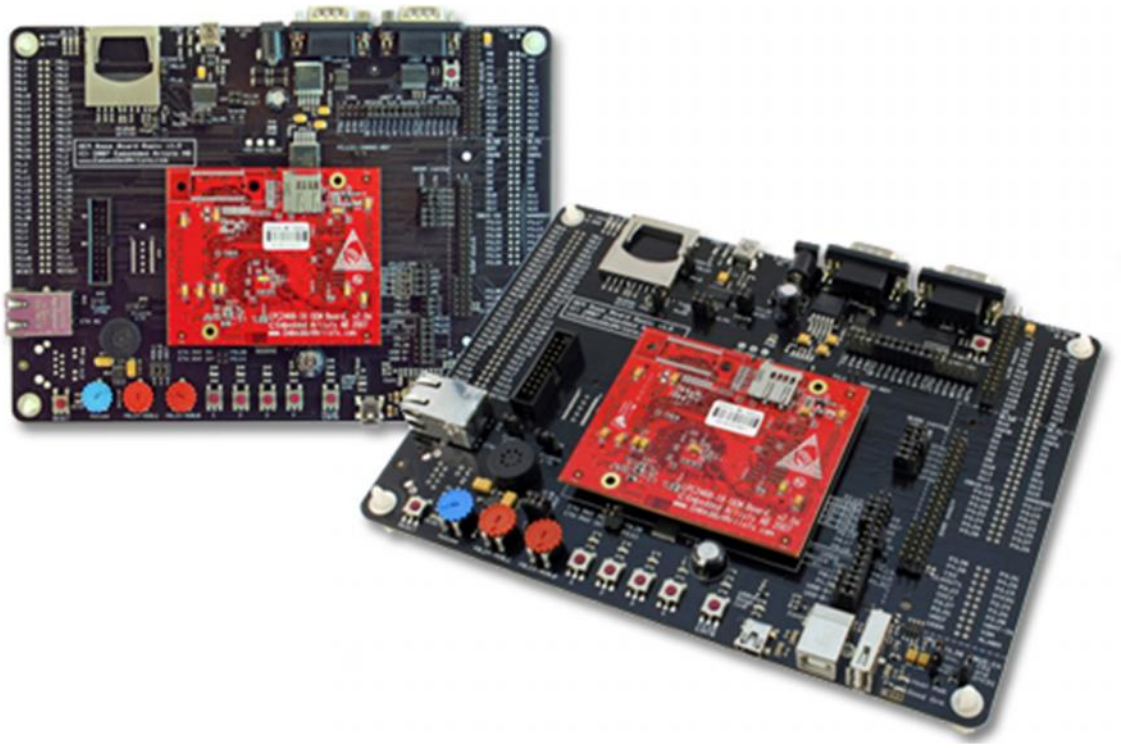
Traffic intensity:

- Constant: during the simulation, each entry point lane receives 0.1, 0.2, 0.3, 0.4 and 0.5 cars per second. Total of five simulation scenarios .
- Linear growth: start with 0.1 car / lane / entry point / second and increase by 0.1 car / lane / entry point / second every 400 s
- Poisson distribution: requires the elaboration of a table with 20 entry points x 2000 s. Each cell of the table represents the number of vehicles entering one entry point of the AoA at that time. This table is an input to each simulation run so that simulations performed on different platforms can have their performances compared.

6 Simulation Platform

In this section, the term **hardware** (or **hardware platform**) refers to the physical platform consisting of all integrated circuits on a board, including microprocessor, memory and communication interfaces. The term **platform** refers to the hardware, the operating system (RTOS), the language run-time system (when required), and libraries (communication stacks, device drivers, …).

The hardware that will be used to run the simulations is an LPC2468 processor (with an ARM7TDMI) architecture.



Characteristics of this processor and board (EA 2468) include:

- 512 KB of internal Flash
- 96 KB of internal RAM
- 32 MB of SDRAM
- 4 MB of NOR Flash
- USB host and device
- Serial ports – one on DB9 and on over USB.

There is a port of X Real-Time Kernel for this board; hence, this RTOS may be used to implement the control and simulation software.

Three architectural solutions were considered:

1. Dual platform: PC and embedded. The PC platform consists of a PC running Windows 7. The embedded platform consists of the EA 2468 board running X Real-Time Kernel version 2.1 (device drives and USB stack are available if

needed). The simulator runs on the PC while the Traffic Lights Controller runs on the EA 2468 board. The PC and the EA 2468 board are connected by an RS-232 serial link. Simulation can be shown graphically in real-time.

2. Single embedded platform: the simulator and the controller run on the embedded processor.
3. Single PC platform: The simulator and Traffic Lights Controller run ona PC (Windows or Linux).

The first approach more processing power and a graphical interface, however, the serial link is likely to be a severe bottleneck for system´s performance. In the second approach, the simulation results may be made available via log files.

Considering the serial-link bottleneck, the selected approach was the second one, i.e. the CTA Simulator and the Traffic Light Controller will run concurrently on the EA 2468 board. Each simulation will generate a log file that will be later visualized on a PC, hence, a play back tool must be developed.

The third alternative was not selected due to the interference of OS services on the precise measurement of execution times of the simulator/Traffic Lights Controller.

7 Traffic Control System Simulator Objectives

The objectives of the Traffic Control System Simulator are:

- To allow the evaluation of system behavior and performance according to the following metrics:
    - Total CPU execution time for a given scenario: cars arriving at the entry points during a 2000 s simulation.
    - Average vehicle speed during the simulation time.
    - The functional correctness of the traffic scenario after processing a given amount of events.
- To allow the comparison of different execution platforms (HWs, RTOSs, Programming Languages, and Run-Time Systems) using the aforementioned metrics.
- To allow the comparison of different Traffic Control strategies, such as:
    - independent – each semaphore is controlled independently of other semaphores or sensor readings;
    - control of semaphores cycle times based solely on the number of vehicles waiting for the semaphore to open;
    - control based on a synchronization strategy with other semaphores in the neighborhood or in the same street (called "green wave");

The traffic control strategies to be implemented are detailed in next Section.

7.1 Traffic control strategies

Strategy #1 – Independent Control Strategy (IC)

The Strategy #1 – Independent Control Strategy – has the following characteristics:

- the semaphores have fixed cycle times for all the intermediate states.
- the semaphores are not synchronized to each other.

Strategy #2 – Control based on congestion level (CBCL)

The Strategy #2 – Control based on congestion level (CBCL) – has the following characteristics:

- every Block is equipped with a sensor, which is mounted in such a way that it is able to detect that a certain amount of N or more cars is stopped (waiting for the green light) at each of the lanes of the corresponding Block. This amount N of cars is the "congestion level" and is set to 15 (60% of a lane total capacity) by default.
- for the sake of simplicity, one can consider that all the lanes are uniformly filled, i. e., if the sensor detects that certain lane L1 has just been filled with N cars, thus all

the other lanes pertaining to the same block of the same street are also filled with the same amount N of cars.

- if the sensor detects that N cars or more are stopped at each lane of the Block, the traffic light T1 that controls the traffic in that block is red, and the current time for red light of T1 is 24s or less, the total time for the red light of T1 is adjusted to 30s.
- if the sensor detects that N cars or more are stopped at each lane of the Block, the traffic light T1 that controls the traffic in that block is red, and the current time for red light of T1 is between 25s and 39s, the opposite light T2 of this semaphore is immediately turned to yellow and the remaining time for the red light of T1 is adjusted to 6s.
- if the sensor detects that N cars or more are stopped at each lane of the Block, the traffic light T1 that controls the traffic in that block is red, and the current time for red light of T1 is more than 39s, no change is performed.
- if the sensor detects that N cars or more are stopped at each lane of the Block, and the traffic light T1 that controls the traffic in that block is green or yellow, no change is performed. This can occur in a situation when the traffic light T1 is green but the cars are not allowed to go ahead because of congestions at the next Blocks.

Strategy #3 – Control based on Traffic Facilitation (CBTF)

The Strategy #3 – Control based on Traffic Facilitation (CBTF) – has the following characteristics:

- For every semaphore, one of its traffic lights has a flag indicating that it belongs to a street that has its traffic facilitated (also called “green wave”). This light is referred to as Traffic Facilitating Light (TFL) from this point on.
- Every TFL is aware of the state of the previous TFL that belongs to the same street.
- Every TFL defines a “block current time”, that is the time, from the previous TFL´s turn-to-green event, after which this TFL itself should turn to green in order to keep the “green wave” synchronized. This time is set to 5s by default, which is the time spent by a car to drive through a block at maximum speed.
- The control strategy consists of adjusting the delay between the cycle of one TFL and the previous one. If the block congestion level is below 60% the delay is set to 5s. If it is between 60% and 99% then the delay is set to 0s. If it is 100% the delay is set to -5s (i.e. the TFL turns green five seconds before the previous TFL).

Strategy #4 – Alternative Control based on Congestion Level

This strategy is an alternative to Strategy #2 and is possibly simpler to implement. It has the following characteristics:

Each semaphore is modeled by a state machine with six states. The timing of each state is presented in the table below for the two possible scenarios: equal timing and preference to street X.

Equal timing				preference to street X		
Time	street X	street Y		Time	street X	street Y
2s	Red	Red		2s	Red	Red
38s	Green	Red		46s	Green	Red
5s	Yellow	Red		5s	Yellow	Red
2s	Red	Red		2s	Red	Red
38s	Red	Green		30s	Red	Green
5s	Red	Yellow		5s	Red	Yellow
90s				90s		

At the end of every 90s cycle each semaphore evaluates its congestion sensors and decides on the timing for the next cycle. If both streets (X and Y) are either not congested or both are congested then the Equal Timing strategy is used. Otherwise, the other strategy is used (preference to …) with longer Green time to the congested street.

8 Traffic Control Simulator Implementation

To achieve a fair comparison among platforms, all implementations must follow these implementation rules:

- Each vehicle in the system must have a unique identity and its position in the AoA must be known during all the time it participates of the simulation.
- Each traffic light has an associated vehicle buffer with a capacity of 25 * NumberOfLanes. Hence, each intersection has two such buffers and the AoA has 200 buffers in total. Each buffer must keep track of the ids of the vehicles.
- If a buffer if full, no vehicles can be inserted, even those that see a green traffic light.
- The turns are implemented as:

T	Implementation
10%	After 9 vehicles go forward the 10th turns
15%	After 6 vehicles go forward the 7th turns
20%	After 4 vehicles go forward the 5th turns
35%	After 2 vehicles go forward the 3rd turns

- The position of a vehicle is recorded as the buffer where it is currently located.
- Provided the next buffer has space, a vehicle leaves a buffer when the light is green (at most NumberOfLanes vehicles leave each second) and enters the next buffer.
- A vehicle must stay in a given buffer for at least 5 seconds (corresponding to the amount of time to move the distance of one block).
- The simulator must keep track of the average speed of each vehicle individually. Average speed is given by: total distance traveled inside AoA / total amount of time inside AoA. The total distance is given by the number of blocks traveled * 100m. The total number of blocks traveled is given by the number of buffers the vehicle visited while in the AoA.
- At the end of a simulation the system´s average speed is calculated, as the average speed of all vehicles participating in the simulation.
- The entry and exit points have infinite sized buffers. The entry points receive vehicles generated by the simulation scenario and the exit point buffers store the vehicle that left the AoA.
- When the simulation stops (at time 2000) there will be vehicles in the AoA that did not arrive at one of the exit buffers. These vehicles must be considered for the systems´ average speed calculation.

- Vehicles leaving an entry buffer must stay for 5 seconds in the next buffer. (I.e. entry buffer is one block before the first intersection in the AoA).
- Vehicles leaving a buffer at the outbound edge of the AoA immediately enter an exit buffer.
- To have a fair comparison among implementations, the evaluation of semaphores during simulation will be performed in a fixed order. In this way, eliminating the interference to the simulation results due to the order of evaluation of semaphores.

9 Log File Size Estimation

As a feasibility check, two possible implementations of the simulation log are evaluated.

Worst case estimates for the simulator:

- Total number of lanes in Physical Scenario 2: 18 N-S lanes and 20 E-W lanes, total of 38 lanes.
- Vehicle capacity of AoA: considering that each lane in each block has a capacity of 25 cars, 38 lanes, each one being 10 blocks long, have a combined capacity of 38 x 10 x 25 = 9500 cars. Round up this value to 10.000 vehicles.
- Total number of vehicles participating in a given simulation: during simulation, the worst case is to have 0.5 new vehicles per lane entering the AoA on each entry point every second. Hence, 0.5 vehicles, 19 entry lanes, 2000 seconds results in 19k vehicles. Considering that 10k vehicles were already in the AoA at the start of the simulation, there would be 29k vehicles, at most. Hence, a vehicle identification should use 16 bits.
- In a single buffer / block implementation, there would be 200 buffers representing blocks of the AoA, plus 20 entry buffers and 20 exit buffers. Total of 2400 buffers. Hence, 8 bits are sufficient for buffer ids.
- In a two buffer / block implementation (one buffer for cars stopped on the red sign and one buffer for cars in transit, i.e. moving) then the total would be 420 buffers requiring 9-bit ids.

9.1 Log File Implementation based on Snapshots

The implementation approach is to record a snapshot of the AoA every second. The snapshot lists the identifications of vehicles in each one of the buffers (blocks) in the AoA.

Log size:

One snapshot has: 10.000 vehicle ids each with a buffer id – 10.000 x 4-bytes = 40K

All 2000 snapshots of a simulation take, at most: 40K x 2K = 8 MBytes.

To reduce space at the cost of memory alignment, 3-byte entries may be used, reducing the size to 6 MBytes.

9.2 Log File Implementation based on Events

The implementation approach is to record events that occur every second. The four possible events are listed below:

Event id	Event description	Max events /s
1	Vehicle_id entered transit_buffer_id.	38 (*)
2	Vehicle_id stopped at semaphore_buffer_id	38
3	Vehicle_id entered entry_buffer_id	9.5 (**)
4	Vehicle_id exited AoA via exit_buffer_id	9.5
	Total	95

(*) – 38 lanes, on each lane one vehicle moves / s when the light is green

(**) – 19 entry lanes where at most 0.5 vehicles enter the AoA per second on each lane.

Each event records: event id, vehicle id, buffer id. Hence, four bytes are sufficient to record one event.

Total log size: 95 ev/2 x 4 bytes x 2000 s = 760 KBytes.

This approach requires about 10 times less space for the log file.

The selected alternative is the Log based on Events.