AQT-R 2002 (THETA 13)
2002 IEEE – TTTC International Conference on Automation, Quality and Testing, Robotics
May 23 – 25, 2002, Cluj Napoca, Romania

Object Street Traffic Model

T. S. Letia, A. Astilean, M. Hulea, C. Avram and H. Valean
Faculty of Automation and Computers
Technical Faculty of Cluj-Napoca
C. Daicoviciu St. 15, RO-3400, Cluj-Napoca, Romania
E-mail: Tiberiu.Letia@aut.utcluj.ro

Abstract: The paper presents a solution for real-time simulation of the urban vehicle traffic. Using an object model for intersection behaviour, the traffic system is modelled such that describe the delays introduced by the speed and distances.

Keywords: hybrid systems, object model, UML diagrams, real-time simulation, remote method invocation.

1. Urban traffic problems

Due the limitations of the current traffic control systems, the improving of the performances implies a new approach and new models [2]. The current research focuses on the developing of new models based on the software objects.

There are important issues that must be taken into account like: data – gathering techniques, data processing techniques, the development of more adequate description of traffic flow [3] and control techniques [1].

The activities involved by this research include the design of the information system, the traffic control system and the traffic surveillance system for a medium size town. This implies: the traffic surveillance, the congestion prediction, the control of the traffic lights, the dynamic route guidance, the estimation of the arrival times etc.[5.6].

To get these aims, the following have to be achieved: the models of intersections and the traffic system, the adaptation of the models to continuous system changes, the evaluation of the performances of the control algorithms to be able to choose the best algorithm and the best control algorithms according to different situations.

2. Models of intersections

The street traffic models are based on the following elements: intersections and streets. The core of these models is represented by intersections. There are many types of intersections like that represented in the figure 1.

The figure 1 represents an intersection of two streets each of them having four lanes for both directions. The cars can be driven on the three directions left, right or forward.

The traffic is controlled by traffic lights (S_{iL}, S_{iF}, S_{iR} for left, forward and right, where the index i takes values form 1 to 4).

The duration of the green colour of a traffic lights on a direction D_i and a specified lane is denoted by β_i. That expresses the maximum number of cars from the lane that cross the intersection during a cycle.

The sum of the green colour duration of all traffic lights of an intersection can be fixed or variable.

The figure 2 represents the buffers of the intersection. The symbols λ_{iL}, λ_{iF}, λ_{iR} denote

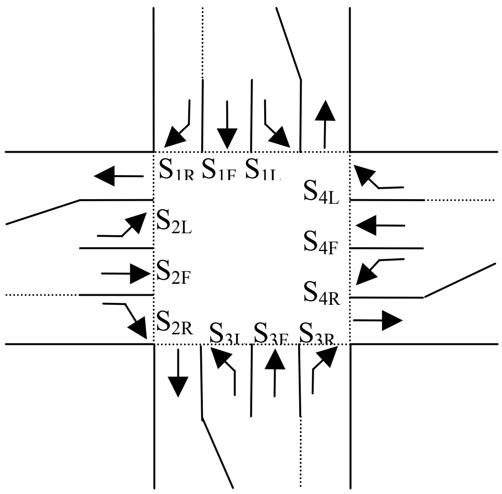

Figure 1. The intersection of two streets

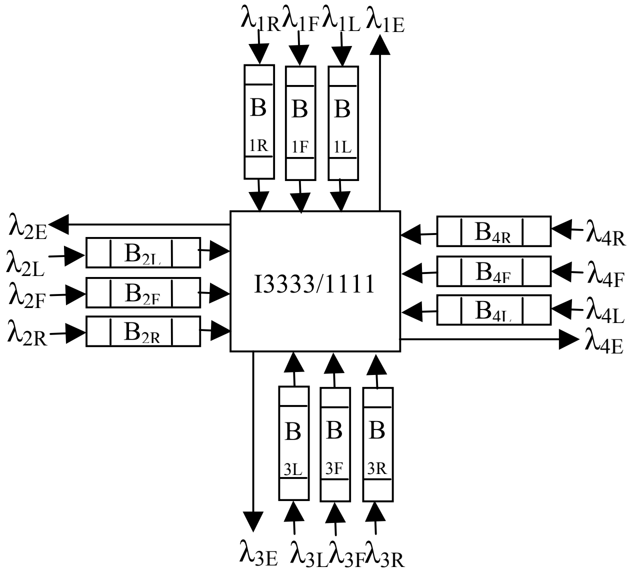

Figure 2. The model of an intersection 3333/1111

the input rates (numbers of cars entering on a buffer during a cycle). For a buffer, Markov chain can be constructed as that represented in figure 3.

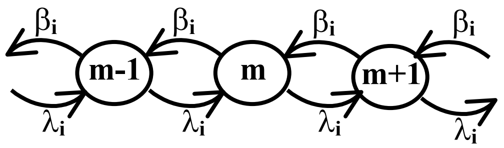

Figure 3. Markov chain

The corresponding relation is given by :

$$(\lambda_i + \beta_i) * P_i(m) = \beta_i * P_i(m+1) + P_i(m+1) * \lambda_i \quad ; \qquad (1)$$

where $P_i(m)$ expresses the probability that in the buffer are m cars.

For results:

For $m = 0$ *results* $\lambda_i * P_i(0) = \beta_i P_i(1)$; (2)

For $m = 1$ *results* $P_i(1) = \frac{\lambda_i}{\beta_i} * P_i(0) = q_i * P_i(0)$; (3)

denoting $q_i = \frac{\lambda_i}{\beta_i}$; *general relation is*

$$P_i(m+1) = (1 + q_i) * P_i(m) - q_i * P_i(m-1); \qquad (4)$$

From (4) the following relations are obtained:

$$P_i(m+1) = q_i^m * P_i(0); \qquad (5)$$

from (5) taking in to acount that $\sum_{m=0}^{\infty} P_i(m) = 1$ result $P_i(0) \sum_{m=0}^{\infty} q_i^m = 1$;

The condition to avoid the accumulation of the cars is $q_i<1$ or $\lambda_i < \beta_i$.

If $q_i > \beta_i$; during a period the accumulating occurs which could lead to overflow.

From (5) result:

$$P(0)\sum_{m=0}^{\infty} q^m = \frac{1}{1-q} \quad \text{result } P(0)\sum_{m=0}^{\infty} q^m = 1; \quad (6)$$

$$P(0) * \frac{1}{1-q} = 1 \quad \text{result } P(0) = 1 - q;$$

Such model can be used to determine the maximum green colour duration without overflow the next intersection capacities.

Input \ state \ output model

Singh and Title [7] use for normal flow rates the following model of an intersection:

$x(k+1) = A\cdot x(k) + B_0\cdot u(k) + B_1\cdot u(k-1) + B_\delta\cdot u(k-\delta)$

$y(k+1) = C\cdot x(k)$

Where:

- u is the input vector representing the input rates
- x is the state vector representing the waiting queues
- y is the output vector representing the output rates of different lanes
- δ is the delay introduced by the distance between intersections.

The matrices A, B and C have the appropriate dimensions. This model works only around a steady point. The control concerns the reducing (eliminating) of the deviation of the state vector (i.e. the waiting queues).

Such linear model has a disadvantage on solving the problem only for normal traffic.

The nonlinear model of a buffer

It is proposed a model taking into account the non-linearity of the traffic around of an intersection. It is considered a waiting queue having $x_i(k)$ cars the β_i the transfer capacity (number of cars) for the current cycle. At the end of the cycle, the waiting queue contains $x_i(k+1)$ cars and $y_i(k)$ cars went out. They are given by the following relations:

$$x_i(k+1) = \begin{cases} 0 \text{ if } & x_i(k) + u_i(k) \le \beta_i(k) \\ x_i(k) + u_i(k) - \beta_i(k) & \text{other wise} \end{cases};$$

$$y_i(k) = \begin{cases} x_i(k) + u_i(k) & \text{if } \quad x_i(k) + u_i(k) \le \beta_i \\ \beta_i(k) & \text{other wise} \end{cases};$$

β_i represents the transfer capacity of the lane for the specified duration (i.e. cycle) that is given to controller.

This model is extended to all the lanes of the intersection. The duration (periods) of the color cycles should be the same for all the intersections; otherwise the sequential usage of this model could lead to an unacceptable deviation from the real situation. Because that is not a realistic constraint, another more reasonable one is proposed. All the intersections have the period of the cycle a multiple of a fixed duration τ called *sub-cycle*. The green color of each lane is given (planned) related to these durations. In this case, it is accepted that a lane is opened during a cycle in more than one sub-cycle. The system is modeled with the time step given by τ time units. For the usage of the model an object model of the traffic control system is developed. The previous model is enhanced with a simple controller that applies the specified opening duration.

When the distance between intersections leads to significant delays of the car arrivals to the next intersection, this can be modelled by introducing fictive intersections (without crossroad) that are not open during a sub cycle (of duration τ) but having the maximum transfer capacity on the next sub cycle. The model precision of temporal approximation of the traffic system is given by τ.

The traffic structure specifications

The traffic system model includes intersections that have inputs and outputs denoted by the following rules: Before slash every character represents the number of buffers on the input of that direction. After the slash every number represents the number of the outputs of that direction.

For example, 2133/1331 means the following:

- There is an intersection of 4 streets.
- The first street has 2 input buffers and one output.
- The second has 1 input buffer and 3 outputs.
- The third and the fourth have 3 input buffers and 1output.

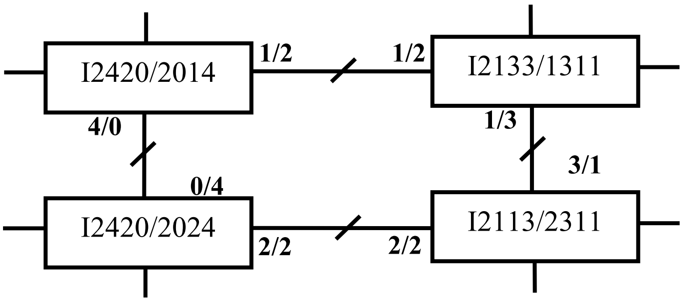

Figure 4.Connection between intersections

Using this notation the traffic system can be represented as in figure 4 making the links between intersections.

3. Traffic system structure

The architecture of the traffic control system is based on a supervisor and a numbers of controllers [4].

The supervisor goals are:

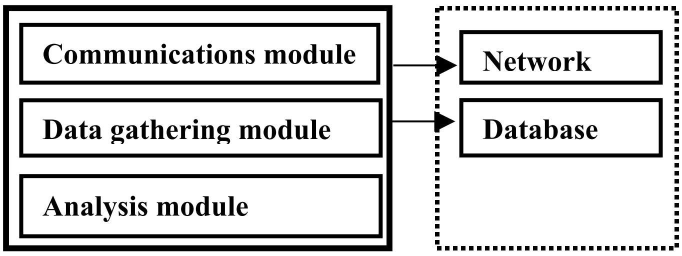

Figure 5. Supervisor modules

- Collect data from the controllers. From the point of view of the client \ server protocol the supervisor it is seen as a client because it ask for data to the controllers.
- The data that are collected from the controllers are used to update the state of the system – the model of the system.
- The model of the system it is used by the supervisor for generating new controllers parameters.
- The parameters are transmitted to the controllers. These parameters represent the cycle length and the green states for every traffic light.

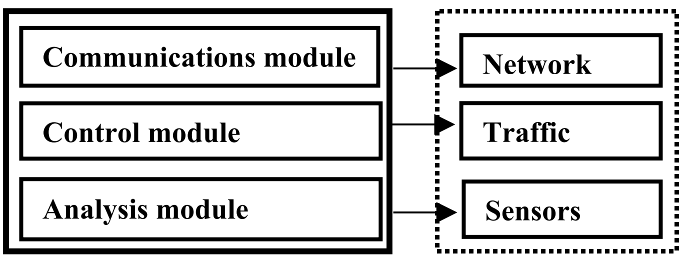

Figure 6.The controller modules

The controller goals are:

- Send data about the state of the intersection when the coordinating asks for them. From the viewpoint of the client \ server protocol the controller it is seen as a server because it offer services – like sending dates and changing the control parameters at request.

- Change the control parameters when receives them from the supervisor.
- The controller receives the commands from the supervisor and acts as the requirements.

4. The system implementation

For the simulations it is used the Java language. The main advantage taken into account was the simplicity of the implementation and the portability of the java programs.

Since Java is an object orientated language we could easily make a flexible system. So can be simulated the behaviour of different types of intersections.

The road traffic control system – a distributed architecture

Distributed computing involves the design and implementation of applications as a set of cooperating software entities that are distributed across a network of machines

Difficulties in developing distributed computing systems are latency, synchronization and partial failure.

Dealing with all of the details of network communication can be tedious. Java language provides simple mechanisms for creating powerful client – server applications. That makes simple to write client – server applications.

The communication subsystem between the supervisor and the controllers it is implemented using a distributed object technology – Remote Method Invocation. Using RMI, the requests of the client to the server look like a local procedure call.

The RMI technology allows a Java method to obtain a reference to a remote object and invoke methods as easy as if the remote object existed locally.

Analysing the goals of the controllers and supervisor results the next interface, which will be implemented and will be accessible through RMI mechanism:

```
public interface InterfaceCOM extends java.rmi.Remote{
        public Statistics getStatistic();
        public void addAlgorithm(Algorithm alg);
        public void setAlgorithm(Parameters prm, int alg);
}
```

The method getStatistic() it is used by the supervisor to obtain information about a specific intersection containing the medium size of queue and throughput for a given period of time.

The controller can work with different algorithms. These algorithms can be added into the controller engine by the supervisor using the addAlgorithm() method.

After the supervisor analyses data for entire network decide the control strategy for every intersection in the network. The method setAlgorithm() it is used to set the parameters and algorithm used by the controller until the next set.

Class Diagram

The program has a flexible structure because it is possible to add and remove intersections from the structure. Also the process of intersection descriptions it is made easy by using configuration files which describe the structure of intersections in the system. The class diagram is presented in the figure 7.

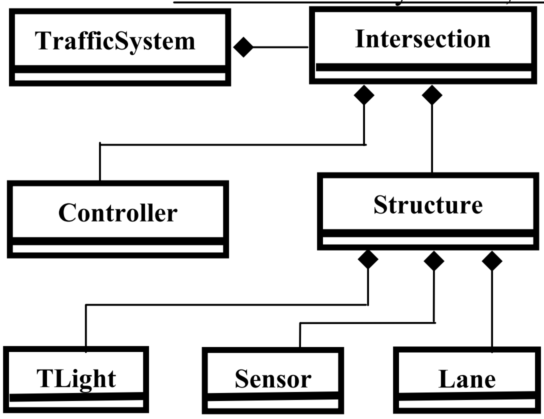

Figure 7. Diagram class

Class Lane – models the input \ output lanes from intersection. The principal parameters are capacity, input frequency and output frequency.
Class Sensor – models the sensor placed on the road level.
Class TLight –models the traffic lights. These correspond to the in buffers and could be commanded by the controller.
Class Structure – models the different types of intersections that contain a specified numbers of sensors, lanes and traffic lights. The Structure objects are used for building different types of intersections. Every Structure object will contain instances of type Sensor, TLight and Lane.
Class Controller – It is used to implements controllers that command the traffic lights, based on an algorithm set up by the supervisor.
Class Intersection – Is build using the aggregation from an object having the Structure type who models the intersection, and a controller object who contains the algorithms for controlling the traffic lights with the set parameters.
Class TrafficSystem – An intersections network is built using the TrafficSystem class. This class will contain instances of the Intersection type.

8. Summary and Conclusions

An object model was introduced with the aim to be used for traffic system model. It cannot be used for mathematical demonstrations but suits to computer approach solving the problem of finding the parameters of control algorithms and the appreciation of the algorithm behaviour. The models can be adapted to real-time simulations as well. The hybrid genetic algorithm needs a significant time to find a solution, so it can be used only off line. It can be used to improve the control algorithms and generally the control system behaviour.

References

[1] Cassandras, C.G. and Wei-Bo Gong (1990). *On-line Control of Discrete Event Systems: Some Extensions of Perturbation Analysis*. Preprints of the IFAC World Congress Automatic Control, Vol. 10, p.259-263, Talin, Estonia.
[2] Cremer, M. and X. Zhang (1993). *System Architectures for Complex Road Traffic Information and Control Systems.* In the Proc. of IFAC 12th World Congress, Sydney, Vol. 9, p.229-232.
[3] Gerken Jeff (2000). *A Practical Approach to Managing Intersection Traffic Data for Large Scale Studies*, Mid –Continent Transportation Symposium.
[4] Gessing, R. and Z. Duda (1990). *Two-level Optimal Resource Allocation in a Large Scale System.* Preprints of the IFAC World Congress Automatic Control, Vol. 10, p.74-78, Talin, Estonia.
[5] Husch, D. (2000) *Intersection Capacity Utilisation*, www.trafficware.com.
[6] Husch, D. *Specification for Traffic Data Format*, www.trafficware.com.
[7] Singh, M.G. and A. Titli (1978). *Systems: Decomposition, Optimisation and Control*. Pergamon Press, Oxford.