Street-crossing Simulation in Real Environments - A Case Study

Viriato M. Marques[1], Hugo R. Felgar[2], Nelson O. Alves[3], Cristina C. Costa[4]
[1]PhD [2,3]Students [4]MsC
Computer Science Department
ISEC - Engineering Institute of Coimbra Polytechnic Institute
Quinta da Nora, 3000-199 Coimbra
PORTUGAL
1

Abstract: - The paper describes a discrete-event simulation model designed to study and optimize a very busy and problematic crossing located at Coimbra, a Portuguese town. The model takes into account a lot of details, namely arriving and crossing times, possible delays and traffic light control. Some common-sense simplifications were made. *BestFit* and *Extend* software were used. The model proved to fit reality constraints in every aspect. Some *what-if* situations have then been tested. However, the conclusion is, perhaps, surprising.

Key-Words: - Discrete-event simulation; optimization; traffic light.

1 Introduction

Discrete-event simulation allows the modeling of complex but very common situations. Once it's proved that the model meets reality, it's possible to look for an optimal solution, testing (new) *what-if* scenarios. This can be achieved by introducing modifications in the original model.

This paper deals with a street-crossing located at Coimbra, an ancient Portuguese town known by Aeminium among the Romans. It's been the capital of the kingdom with the first Portuguese king. Nowadays it has about 110.000 inhabitants, the third oldest University in Europe (about 700 years, more than 20.000 students), a Polytechnic Institute (more than 12.000 students), an hospital with 1200 beds just in the central unit, a 35.000 places football stadium and some big commercial areas, all of this located not far away from the downtown. It also has a lot of narrow streets and this means, of course, severe traffic problems.

The (primary) crossing is located in a very busy place, has no space left for future expansion or modifications and is influenced by another one (secondary) located near it. So, optimizing traffic light control is just *what can be done*.

Section 2 presents an overview of the model and the assumed simplifications. Section 3 describes data acquisition. Section 4 presents the system modules and their implementation, Section 5 describes some *what-if* scenarios and section 6 gives the conclusions.

2 Model Overview

As above-mentioned, the crossing under analysis is composed of two parts:

1. A main crossing located near the Coimbra Botanic Garden where the following streets meet: *Combatentes da Grande Guerra St*., towards the football stadium and a big shopping center (ST); *Dr. Júlio Henriques Av*. towards the Coimbra University (CU); *Marnoco e Sousa Av*. towards a sightseeing place (SS); *Vandelli St.* towards the Military Hospital (MH) (fig.1);
2. A secondary crossing is located nearby, at *Combatentes da Grande Guerra St*. where it meets *Ladeira do Seminário St.*This street arrives from a large area where schools, University campus II, ISEC and other commercial areas were built (CA) (fig.1).

Figure 1 – Street-crossing location (Coimbra, Portugal)

Fig.2 shows a schematic drawing of the above-mentioned crossing:

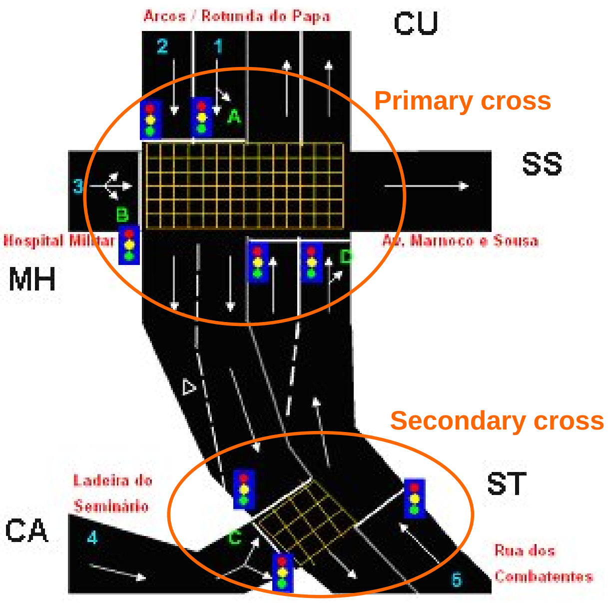

Figure 2 – Street-crossing schematic drawing

2.1 How it works

According to fig.2 the crossing has 5 inputs - numbered 1 to 5 - corresponding to the 5 possible directions of the cars that arrive at the cross from CA, ST, MH and CU (divided into two lanes). The crossing has 3 outputs corresponding to 3 possible directions of the cars that leave it: towards CU, SS and ST.

The number of inputs and outputs is imposed by external conditions such as the general traffic plan of the city and the dimension of the streets. So it's assumed that nothing can be changed about these facts.

Four decision points where cars choose what direction to take next are identified by A, B, C and D:

A. At point A, lane 1, cars may turn left towards SS - thus leaving the system - or just go straightforward to the secondary crossing and towards ST. Turning left may require waiting until there are no cars coming from entrance 5, since these ones have priority. Cars at lane 2 may go straightforward only. These are the restrictions that compelled the subdivision of input A into distinct lanes 1 and 2.
B. A car at point B may choose one of three directions: turn left towards CU; go straightforward to SS - leaving the system; or turn right to the secondary crossing and towards ST.
C. At point C, located at the secondary crossing, a car may turn left to the primary crossing point D or just turn right towards ST, leaving the system.
D. At point D a car may continue straightforward towards CU - leaving the system - or may turn right towards SS, leaving the system too.
E. At any other point a car may go straightforward only.

As fig.2 shows, the crossing inputs are all controlled by traffic lights organized into two groups: access to the main crossing and access to the secondary one. The traffic lights of each group are synchronized. Ideally a car may enter the crossing if and only if there are no cars already there, as no vehicle may stand on the zones marked by yellow squares.

2.2 Model simplifications

As usually in simulation studies, some common-sense simplifications have been made in order to obtain a reasonable model, detailed enough to treat all the important aspects of the main and secondary crossings as well as the mutual influences between them, but not unnecessary complicated or exceeding the domain limits. Some simplifications respect to very unusual occurrences for which a distribution couldn't even be found for a reasonable sampling period. A list of the assumed simplifications follows:

- A single lane is considered between the main and secondary crossings when going towards ST: in fact there are initially two lanes (in front of the ones identified by 1 and 2, after the yellow squares) but just for a few meters. So, after the A point a single lane is considered that carries a number of cars equal to the sum of those proceeding from lane 2 and those proceeding from lane 1 that didn't turn left.
- A single lane is considered towards SS as parking places don't leave free space enough for more.
- The yellow traffic light is considered a green extension as, according to the observations taken in the site, drivers ignore it or even accelerate to avoid the red light.
- Nearby the crossing there are two bus stops that have been ignored in the model since their impact is noticeable only if stopped buses are there. However, such a situation is rare (public transportation is a real problem in Coimbra although somewhat better in the

past few years); and the number of buses is very low compared with the number of light vehicles.

- All the vehicles are assumed as light vehicles. Heavy load vehicles and motorcycles are negligible: the former due to the fact that the crossing is located near the city centre and so their frequency is very low; the last because motorcycles almost don't affect the behavior of the queues.
- The time needed for cars to start moving after the green light appears is considered negligible as, once observed, it's very short when compared to the crossing time, almost not affecting the period of permanence in the system.
- Data sampling has taken place at rush hour only: this assumption implies that any improvement valid for this condition is supposed to work as well for any other time of the day.
- Day to day experience shows that weather conditions influence the behavior of the crossing. Due to time constraints data acquisition has taken place during winter only. However, this means exactly a bad scenario.
- Traffic accidents have been ignored due to their (happily) very low frequency. Stopped vehicles or other occasional obstacles haven't been considered. Drivers are assumed to be law compliant: once located in a lane it's assumed that they don't change it and that they don't stop where forbidden (yellow squares, for instance).
- The crossing period of each car is an average computed according to the acquired data.

3 Data and statistical distributions

Data acquisition took place during winter and between 6.15 pm and 7.15 pm. A list of the acquired data follows:

- Vehicle arrival time at each one of the system inputs, considering the arrival time as the moment at which a vehicle reaches its queue.
- Red and green+yellow time periods for each traffic light.
- Transition periods between traffic lights, for both groups. This study had as target the understanding of the synchronization mechanisms. For instance, periods like time transition between red and green of distinct traffic lights has been taken.
- Time period for each section of the crossing (for instance, between primary and secondary crossings). As already mentioned, an average has then been computed.
- Number of vehicles that have room enough to stay in each section of the crossing and between primary and secondary crossings.
- Number of cars that change direction and which direction, at each one of the points A, B, C and D (fig.2)

The statistical analysis of part of the data above listed was carried out with *BestFit* [1] from Palisade. The software easily allows the identification of the more convenient statistical distribution that fits the data under analysis. Fig.3 shows the Weinbull distribution [2,3] that best fitted the inter-arrival times of the vehicles at some inputs. Exponential distribution [2,3] has also been identified and chosen for other ones. *BestFit* also computes the associated parameters of each distribution: for Weinbull the location, scale and shape parameters, for exponential the mean and standard deviation.

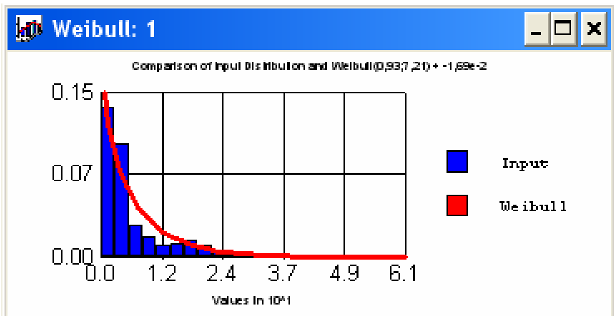

Figure 3 –Weinbull distribution for inter-arrival times

For each decision point A, B, C and D (fig.2) the probability of going forward or taking the left or right direction as also been computed.

4 The model

The simulation model has been developed with *Extend* [4] software from ImagineThat!. The model is composed of distinct modules that accurately simulate, each one, the behavior of a crossing entrance or a small group of traffic lights. These modules are then synchronized and the whole set implements the complete model of the primary and secondary crossings.

4.1 Entrances and traffic lights

According to the previous sections, the complete model is composed of 5 inputs, 3 outputs, the traffic lights and the way between the primary and the secondary crossings.

4.1.1 Entering and leaving the crossings

Each input has different statistical distributions or distinct parameters for the same distribution. So, inter-arrival times are implemented by the Extend *generator* block configured with the respective distribution and parameters.

A crossing comprises various possible ways to go. In the model, each way is represented by the Extend block *multiple activity* configured to accept the maximum number of vehicles the crossing can hold in that direction and the average time that each vehicle takes to cross. These values are defined according to the acquired data and mean time computations. Each *multiple activity* block holds (and counts) the cars that enter the crossing in a given direction and holds them as long as specified by the mean time of permanence in that direction.

An example of a complete crossing entrance is shown in fig.4. This module implements the input MH at decision point 3 (fig.2):

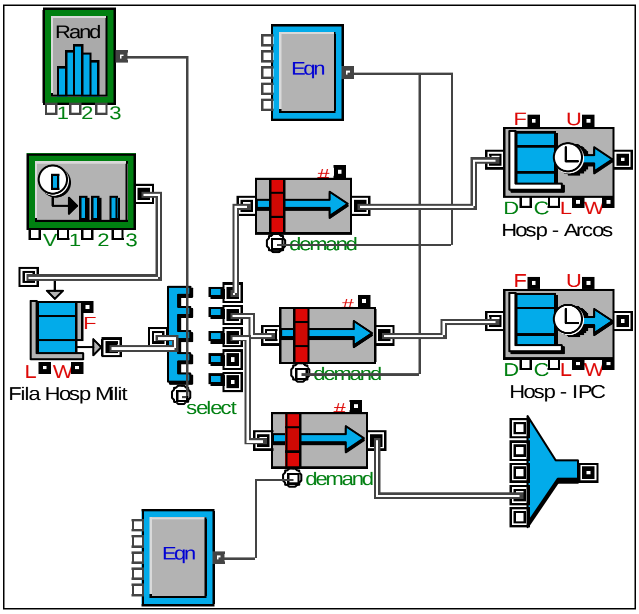

Figure 4 –Simulation module for the MH entrance

A *generator* inputs items (cars) to a FIFO queue according to the distribution previously identified for this input. As vehicles may chose one of three directions, a *select* block is used, controlled by a *random generator* configured to follow the previously computed probabilities of turning left, right or going forward.

However, for a car to start moving it's mandatory that the traffic light is green and the crossing is free of traffic. This is achieved by sensing the busy state of the *multiple activity* blocks - mentioned at the start of this section - that implement other ways of the crossing, thus allowing to know if the vehicles that did enter the crossing have already leaved it or not. This information - number of cars specified by *Var2*, *Var3* and *Var 4* in eq.1 - is used as input for *equation* blocks that compute functions such as

$$Result = Var1 \text{ and not } ((Var2>0) \text{ or } (Var3>0) \text{ or } Var4) \quad [1]$$

The *equation* blocks also sense the state of the traffic lights of the input under consideration, looking for a green. In eq.1 this state is represented by *Var1*. This way it's possible to control *activity service* blocks that pull in cars only when the traffic light is green and the way in front of them is free.

The *Multiple activity* blocks of fig.4 represent, as in any other module, the time taken to cross towards CU (Hosp-Arcos in the figure), SS (Hosp-IPC in the figure) and ST (see fig.2). *Combine* blocks are then used for directions ST and CU (the last not visible in fig.4) as they must receive traffic from other entrances of the crossing too.

4.1.2 Traffic light control and synchronization

Fig.5 shows how traffic lights of the main crossing are implemented.

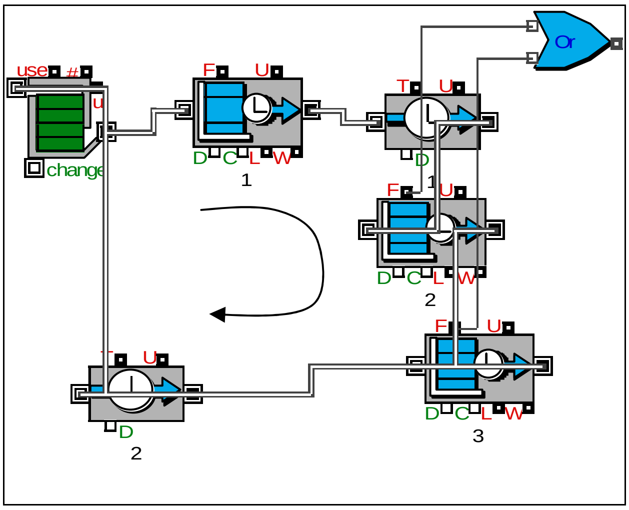

Figure 5 – Blocks of the main traffic light controller

As shown in fig.2 the main crossing comprises three small groups of traffic lights located near decision points A, C and D. The implementation "trick" is using a *resource* block that generates an item that infinitely moves between *activities multiple* and *activities delay* for generating red and green states for the traffic lights: when the item resides in one of the *activities multiple*, this means a green state for that traffic light and a red state for all the others. This is achieved by sensing the output F of the block, which outputs a "1" if the block is busy. When the item resides in an *activity delay* block, this means a red state for all the traffic lights of the crossing as in this situation no *multiple activity* block holds the item. According to this, the following operation cycle results:

a. The traffic light at point B starts with a green state that lasts for 29s. All the others remain in the red state;
b. Then the item moves to the following *activity delay* block and all the traffic lights enter the red state for 3s;
c. Reaching the next *activity multiple*, the item generates a green state for traffic lights located at points A and D that lasts for 57s;
d. When the item enters the next *activity multiple*, traffic lights at point D become red and those at point A stay in the green state. Notice that there's no need for a global red state between situations *c* and *d* and so no activity delay is used here;
e. Then all the traffic lights become red when the item enters the last *activity delay* block;
f. Finally the item reenters the *resource* block that pulls it out again restarting this cycle;
g. The OR gate is used for the traffic light at point A to remain in the green state at situation *d*.

This approach may be used as a basis for the implementation of state machines. This solution drastically reduced the number of needed blocks and simplified the synchronization process that naturally emerged from it; and the implementation of some *what-if* scenarios, described in the following sections, became much easier.

4.2 Model validation

The model has been validated by comparing the results of simulation with the data previously acquired. Table I shows, for instance, the results for number of cars arriving at locations 1 to 5 (fig.2):

	Zone 1	Zone 2	Zone 3	Zone 4	Zone 5
Real data	**497**	**428**	**71**	**313**	**619**
Sim 1	477	394	79	281	631
Sim 2	441	447	74	327	592
Sim 3	494	408	65	329	571
Sim 4	522	448	60	313	634
Sim 5	446	427	71	305	645
Mean	477	427	71	313	631
Diference	20	1	0	0	12
Diference (%)	**4,02%**	**0,23%**	**0,00%**	**0,00%**	**1,94%**

Table 1 – Cars arriving at zones 1 to 5 in reality and simulating

As can be seen, the differences are negligible. For other tests such as number of vehicles choosing a given direction, the results were similar.

5 What-if scenarios

The results obtained by simple modifications of the original model didn't show significant improvements. In order to test a more complex solution a previous simplification has been made.

5.1.1 A simplified model

This simplification consisted of considering a single lane between secondary and primary crossings when moving towards point D. This means a single queue for cars arriving at this point and considering a volume of traffic (A in table 1) equal to the sum of the two original and distinct values B and C of table 2. This is a common-sense assumption since the original street has a single way that bifurcates just before the primary crossing. In fact the values obtained for this simplified model are in complete agreement with the original ones. Some results follow:

	Queue length average	Waiting time average
Lane A = B+C	**5,22102**	**21,46734**

Table 2 – Some values for the simplified model

	Queue length average	Waiting time average
Lane B	**2,396942**	**21,118444**
Lane C	**3,525964**	**20,698204**

Table 3 – Corresponding values for the original model

Fig.5 shows the module used to implement this single way between the secondary and the primary crossings. The probability of cars turning right at decision point D has also been adjusted in the *random generator* block since initially it was applied to the cars in the right lane only.

5.1.2 An enhanced simplified model

Looking for results that could show clear improvements in the crossing behavior or even suggest an optimal solution, the placement of traffic sensors has been considered for both crossings.

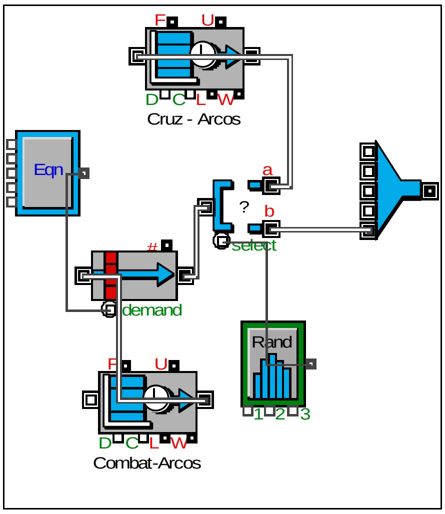

Figure 2 – Simplified model: the single way between secondary and primary crossings

Fig.6 shows the module that implements the primary crossing traffic light controller for one of these scenarios (see also fig.5).

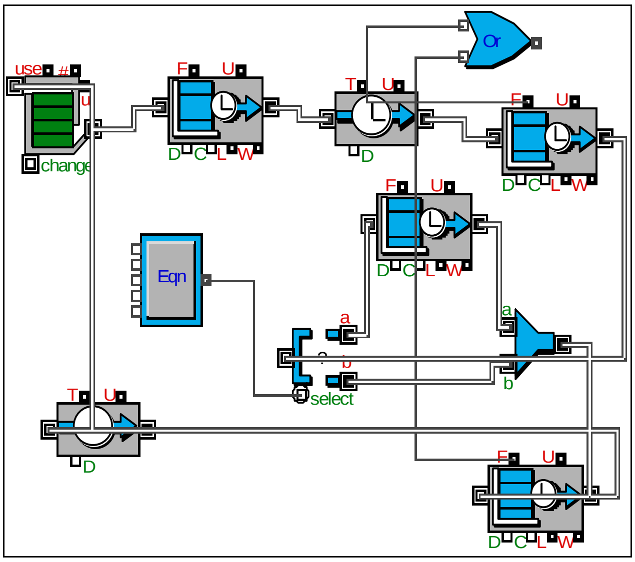

Figure 3 – Blocks of the main traffic light controller with traffic sensors

In this case sensors were considered to be placed somewhere before points A and D. For cars coming from the CU direction a queue length of at least 20 cars or a full queue before point D determines an extra green time period for the traffic lights near these points. In the model this is accomplished by taking into account the number of items contained in *activity multiple* blocks and deciding what to do in the *equation* block that computes, for instance, eq.2:

$$Result = (Var1>20)\ or\ (Var2>20)\ or\ Var3 \quad [2]$$

If *Result* is *true* then the *select* block introduces an extra *activity multiple* block configured for 10s. For point D a similar situation has also been considered. The following tables compare some results obtained with the simplified and the enhanced models:

Simpified Model	Mean Length	Mean waiting time	Max Length	Max waiting time
Zone 1	0,867±0,02	6,44±0,144	10,1±0,451	34,8±0,071
Zone 2	0,729±0,02	6,38±0,145	9,02±0,391	34,8±0,102
Zone 3	0,457±0,01	25,4±0,921	4,22±0,277	85,3±9,39
Zone 4	2,17±0,050	25,9±0,373	11,8±0,453	67,8±0,084
Zone 5	1,40±0,030	8,68±0,154	13,5±0,46	40,9±0,041

Table 4 – Simplified model (averages, 50 simulations)

Enhanced Model	Mean Length	Mean waiting time	Max Length	Max waiting time
Zone 1	0,783±0,02	5,89±0,13	9,88±0,366	34,9±0,113
Zone 2	0,697±0,01	5,97±0,142	8,76±0,37	34,8±0,108
Zone 3	0,531±0,03	29,7±1,10	4,68±0,356	105±13,4
Zone 4	2,11±0,061	25,6±0,397	11,9±0,456	67,6±0,136
Zone 5	1,44±0,030	8,77±0,173	14,5±0,665	40,9±0,050

Table 5 – Enhanced model (averages, 50 simulations)

The values of tables 4 and 5 are very similar except for zones 1 and 2 that have a high traffic load and were object of the described enhancements.

6 Conclusion

This paper just summarizes some tests and results of the whole study. In fact, a lot of other *what-if* scenarios have been tried, always looking for an appealing, optimized solution. However, the best results are those of table 5, obtained as described in section 5.1.2.

So, using a very detailed model in perfect accordance with reality showed that the present solution hardly could be significantly improved. Congratulations for the designers and ... well ... we'll keep in queue -:) (reality is hard to swallow; it is no walk in the park, this thing called life).

References:

1. Palisade Corp., BestFit Guide, 2006
2. Banks, Jerry et. al, Discrete-event system simulation, Prentice Hall, 2000
3. Law,A., Kelton,W., Simulation modelling and analysis, McGraw-Hill, 2000
4. ImagineThat Corp., Extend Guide, 2006