MODELLING AND SIMULATION OF AUTOMATIC DEBITING SYSTEMS FOR ELECTRONIC TOLL COLLECTION ON MOTOR HIGHWAYS

A.G. HOEKSTRA, L. DORST, M. BERGMAN, J. LAGERBERG, A. VISSER, H. YAKALI, F. GROEN, L.O. HERTZBERGER

Proceedings IASTED International Conference Applied Modelling and Simulation, Ed M.H. Hamzam pp 104-108, 1997.

MODELLING AND SIMULATION OF AUTOMATIC DEBITING SYSTEMS FOR ELECTRONIC TOLL COLLECTION ON MOTOR HIGHWAYS

A.G. HOEKSTRA, L. DORST, M. BERGMAN, J. LAGERBERG, A. VISSER, H. YAKALI, F. GROEN, L.O. HERTZBERGER

Department of Computer Science, Faculty of Mathematics, Computer Science, Physics, and Astronomy, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, the Netherlands, tel. +31 20 5257463, fax. +31 20 5257490, email alfons@wins.uva.nl, URL http://www.wins.uva.nl/

ABSTRACT

We have developed a Modelling and Simulation platform for technical evaluation of Automatic Debiting Systems (ADS) for Electronic Toll Collection on Motor Highways. This platform is used in a project of the Dutch government to assess the technical feasibility of Toll Collection systems proposed by industry. The modelling approach and the simulation environment (ADS-SIM) will be presented. As a case study, an example of modelling and simulation of a fictitious ADS will be shown. Finally, the need for High Performance Computing to carry out the simulations is discussed.

KEYWORDS

Transportation, modelling and simulation, discrete event models, evaluation

1. INTRODUCTION

We are involved in a project from the Dutch Ministry of Transport, Public Works, and Water Management to evaluate the technical feasibility of Automatic Debiting Systems for Electronic Toll Collection on motor highways, as proposed by industry to be implemented on the Dutch road network. We have developed a modelling and simulation approach, and realised a software environment to perform these simulations. The environment, called ADS-SIM, is used both by the Ministry and industry for the technical evaluation studies. In a previous paper [1] we have embedded our modelling approach in a general framework of High Performance Discrete Event Simulations of Complex Industrial Systems.Here we describe our work in the evaluation project.

2. EVALUATION OF AUTOMATIC DEBITING SYSTEMS

2.1 Description of an Automatic Debiting System

Due to heavy road congestion in the Netherlands, especially in the western part around the major cities (Amsterdam, Rotterdam, Den Haag) the Dutch Government intends to implement a system of Electronic Toll Collection with the goal to reduce these congestion. It is foreseen that these systems will have to become operational in the year 2001. The toll collection should not interfere with normal traffic flow. Therefore, an Automatic Debiting System (ADS) is required.

An ADS will be constructed as follows (see Fig. 1). Vehicles will have an On Board Unit (OBU) which contains an electronic purse (a smart card) and μ-wave equipment to communicate with the Road Side System (RSS). The RSS consists of a communication system, a sensor system, a registration system and a coordination system.

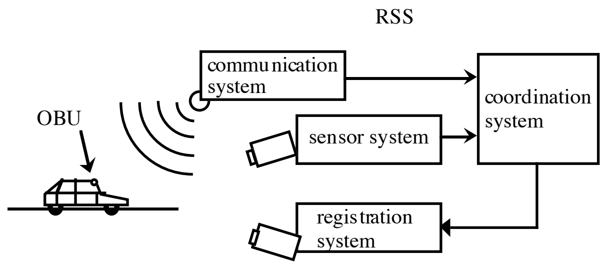

FIGURE 1: A SCHEMATIC DRAWING OF AN AUTOMATIC DEBITING SYSTEM. OBU STANDS FOR ON BOARD UNIT; RSS IS THE ROAD SIDE SYSTEM.

When a vehicle enters the tolling zone the communication system will start an exchange of messages with the OBU, resulting, under normal operation, in a debiting of the smart card in the OBU. The communication system keeps record of the debiting of each vehicle. Furthermore, the communication system will also assess the position of the OBU with which it is communicating. All this data is sent to the coordination system. At the same time the sensor system will be measuring the positions of vehicles in the tolling zone. This data is also transferred to the coordination system. The coordination system will correlate all incoming data, and in that way is able to verify if all passing vehicles have actually paid the tolling fee. Normally this is the case and the coordination system is able to correlate all vehicles seen by the sensor system with the data coming from the communication system.

However, the ADS must also be able to handle violators. If a vehicle does not have an OBU, if it is turned off or does not contain a smart-card, or if the smart card does not contain enough money, the vehicle is considered to be a violator. In the first two cases the vehicle will not communicate with the communication system. However, the sensor system will still detect the car, and the coordination system is in that case not able to match the sensor data with communication data. In the other two cases the communication system will communicate with the OBU, but is not able to charge (enough) money from the smart card. The communication system will send this information to the coordination system, where it is correlated with data coming from the sensor system. In all four cases the coordination system concludes that the vehicle is a violator, and triggers the registration system, which takes a photograph of the license plate of the vehicle. This photograph is sent to a back office system for further processing.

So far the desired, correct operation of an ADS is described. However, in many situations errors can occur. For instance, if a vehicle is not a violator, but for some reason an error occurs in the communication system. In that situation the coordination system can only conclude that the vehicle *is* a violator and trigger the registration system. In the most severe case the driver will have no proof that he or she was willing to pay and the driver is unjustly forced to pay a fine. These so-called Non Charging Events have to be reduced to an absolute minimum.

Another case occurs when the debiting has been done correctly, but through an error in the sensor system, or through a wrong correlation of data in the coordination system, the coordination system decides to trigger the registration system. Again, the driver is forced to pay a fine, but now a proof of toll fee payment is available by presenting the log file which is maintained on the smart card. This mistake is less severe than the previous example, but should still be very rare.

A last example is that of a free ride, where a violator, through an error in the sensor system or coordination system, is not identified as such and therefore is not registered. Many other possible errors in the operation of an ADS can be identified.

The requirements of an ADS, as laid down by the Ministry, result in a set of System Quality Factors (SQF) which describe the chance that a particular error in the system occurs. An example is the already mentioned Non Charging Event, which, in the Dutch setting, is required to be smaller than one in a million. The key issue is to proof that an ADS can operate correctly under such strict requirements. The approach taken by the Dutch government is to simulate the behaviour of an ADS with the goal to find possible faulty operation. In such a way the technical feasibility of all proposed ADS systems will be evaluated.

We have developed a modelling approach to set up discrete event models of an ADS, and we have realised a simulation environment, called ADS-SIM, to implement the models and run the simulations. ADS-SIM is used by the Dutch government to assess the technical feasibility of ADS designs proposed by industry.

2.2 Modelling and Simulation of an ADS

The technical requirements for an ADS are of such a stringent nature that highly sophisticated methods are needed to assess to what extent the proposed solutions can be compliant with those requirements.

For this we need to model a complete ADS, apply realistic traffic input to the model, and use as much as possible knowledge which is available on ADS sub-systems. The model should be able to deal with stochastic parameters such as the probability of a non functioning OBU or dirty license plates. As the response of an ADS depends on detailed microscopic traffic configurations it is natural to use a discrete traffic model as input to the ADS simulator. Finally, the ADS model should be able to take environmental parameters into account, such as the influence of weather conditions on sensor behaviour and on traffic, or e.g. rare specular reflections of sunlight directly into a registration camera.

We model an ADS in a top-down appproach. This means that we hierarchically decompose an ADS into a small number of sub-systems. The sub-systems themselves are modelled using (known) parametrisations. We assume that the state of a sub-system changes on pre-definable time-stamps, due to external inputs and/or due to changes in other sub-modules. This assumption results in a discrete event model of an ADS.

The ADS model is set up as follows. First, we apply a small number of hierarchical decompositions of the ADS and use (known) parametrisations of the remaining modules to describe their behaviour. The connections between the modules and the response of the modules to the presence of vehicles are specified, resulting in a discrete event model of an ADS.

To illustrate the concept we introduce the case of a fictitious ADS (fADS) and show how this hypothetical system can be modelled. In Fig. 2 a schematic drawing of fADS is shown. It consists of four gantries, put on the road and containing the rear-registration camera, the wake up antenna (to put the OBU from sleep in idle mode), the communication antenna, and the front registration camera. Furthermore, fADS has a sensor which measures position and length of each vehicle, on 25 positions on the road.

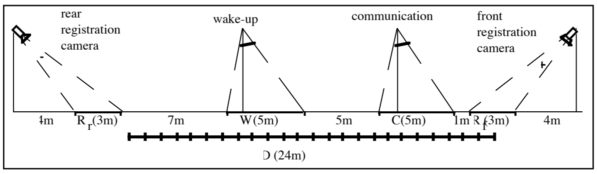

FIGURE 2: THE FICTITIOUS ADS

If a vehicle enters fADS, the sensor will first detect the vehicle, and because it measures the length and position of the vehicle, is able to detect when the back of the vehicle is in field of view of the rear-registration camera. In that case the camera is triggered and a picture of the rear license plate is stored. As the vehicle moves on, the sensor keeps measuring the position of the car, and with these measured points a track of the vehicle is built up. When the vehicle moves under the wake-up antenna, the OBU is put in idle mode, and under the communication antenna the debiting is performed, and the position of the OBU is measured. This OBU position is matched to the closest available track. The vehicle moves on, and when it is in field of view of the front registration camera, the coordination system checks if the track of that car was correlated with an OBU. If so, the vehicle performed a correct debiting and registration is not necessary. The picture of the rear license plate is removed from the system. However, if the track of the vehicle does not contain an OBU, the coordination system must conclude that the vehicle is a violator, and will therefore trigger the front registration camera, and send the pictures to the back office for further processing.

The first step in building the model of fADS is to specify the sensitive volumes of all sensors. A sensitive volume is defined as the volume where a specific sensor (be it an antenna, or a camera) is able to detect a vehicle. In this example we assume a two dimensional model and we will speak of sensitive zones. The real modelling of the ADS systems as proposed by the industry will of course in practice be in three dimensions. The sensitive zones are needed to trigger basic events in the simulator, as will be described in section 2.3.

Next we perform the hierarchical decomposition. The rear and front registration system are modelled as one abstract module. We assume that the OBU wake up always succeeds, and therefore the wake up system is not included in the model. The communication system is also modelled as one abstract module, which performs the debiting and measures the position of an OBU. Finally, we assume that the sensor system cannot be modelled as one single module. In real models this depends to what extend it is possible to define validated models of e.g. such track sensor. In the fADS case the sensor system is decomposed further to 25 sensors which measure the length and position of the vehicle.

It should be pointed out here that, especially in modelling of the sensor system, it is possible that after the decomposition of the ADS the resulting sub-modules need no longer be physical sub-systems, but might as well be logical units which perform a certain measurement. We could say that the sensor system of fADS is modelled as 25 virtual sensors which measure a length and position. The concept of logical sensors was first introduced by Henderson and Silcrat [2] and later Weller, Groen and Hertzberger refined it to virtual sensors [3]. A discussion of the virtual sensor modelling that we have developed for ADS simulations is beyond the scope of this paper and will be published elsewhere.

Finally we need to find a model for the remaining elements. In the fADS example the models are very simple, but they do reveal the general ideas. The virtual length and position sensors are characterised through a parametrisation of the final measurement errors made by the virtual sensors, which are Δl and Δx respectively. The errors will normally depend on a number of other parameters, but in the example we assume that the errors are constant. In the simulation we use these errors to generate a measurement, by

$$x_m = x + \mathrm{RAN} \, , \qquad [1]$$

where RAN is a random number drawn from a normal distribution with zero mean and with a standard deviation Δx.

The communication module is modelled by a probability p that the communication fails, by a transaction time τ needed for the communication, and by an error Δx for the OBU localisation.

Modelling a real ADS will result in more elements in the model and more complicated parametrisations of the elements. However the models will be comparable to the highly simplified fADS example.

2.3 ADS-SIM

We have developed a discrete event simulation environment for ADS simulations. This tool, called ADS-SIM, is developed using Modsim[1], which is a generic discrete event system. ADS-SIM executes ADS models. The dynamics of vehicles in the ADS is governed by the statistics of traffic flow on highways for a number of selected scenarios. The vehicles, i.e. cars, motor drivers or trucks, are traced through the ADS topology, and events are scheduled which represent all activities of the ADS. These events represent e.g. a start-up of communication, OBU activity, a sensor activity or a registration activity. For each vehicle all ADS activity is logged. Next, after the vehicle leaves the ADS, it enters an analysis module, which generates estimates of the desired System Quality Factors. The analysis module carries out the statistical analysis of the generated data (e.g. variance estimation), and decides upon termination of the simulation.

ADS-SIM, which is a joint development of CMG and the University of Amsterdam, consists of three main modules (see Fig. 3). A traffic generator, developed by the RWTH Aachen, Germany, simulates traffic moving over a segment of the road. The traffic is validated against real (Dutch) traffic. Currently the traffic generator writes trajectories of each vehicle (i.e. position of vehicles as

1 Modsim is a trademark of CACI, La Jolla, CA, USA.

function of time) to a file, which is read by the Framework. The Framework is based on the evaluator tool developed by CMG [4]. It contains the man-machine interface of ADS-SIM, and takes care of analysis and logging of the simulation results. The Kernel is used to implement the actual ADS model and to run the simulation. The Kernel contains a number of predefined events which are used to map the discrete event model of an ADS to ADS-SIM.

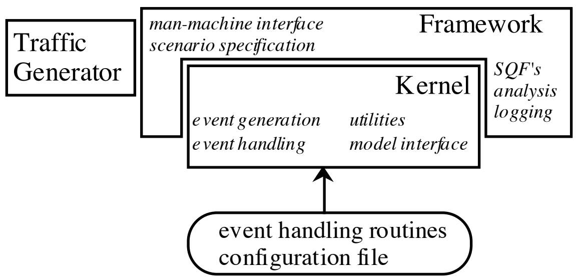

FIGURE 3: ADS-SIM DESIGN

In the evaluation project it is necessary to be able to implement several, different ADS models. We provided a mechanism for this. The user of ADS-SIM provides a configuration file, containing the geometry of the ADS, and the event handling routines. These routines, which are written in C, are linked with the Kernel through the model interface. They contain the parametrisations of the sub-systems and all the logic to connect sub-systems, mapped to events, with each other.

The simulation proceeds as follows. The framework reads trajectories of each vehicle and passes them to the kernel. At the correct time the kernel launches the vehicles in the simulator. Next, the kernel calculates at which time the vehicle enters and leaves all sensitive zones which are defined in the configuration file, and on each of these times schedules an `in_zone` or `out_zone` event (see Fig. 4). These events, which are automatically scheduled by ADS-SIM, form the basis to implement the ADS model. The event handling routines for the `in_` and `out_` zone events, which have to be provided by the user of ADS-SIM, will typically schedule other events, available in ADS-SIM, which are used to implement the ADS model. In the example of Fig. 4, if the next event is the `in_detection_zone` event on time t_1, ADS-SIM will execute the event handling function which was provided by the user.

Using a β-release of ADS-SIM we implemented the fADS model and performed a number of tests. Running on a Sun UltraSparc workstation ADS-SIM handles in the order of 2 million vehicles per day. Under normal traffic conditions the simulations show that fADS is a feasible design which operates correctly. However, if an obvious design error is introduced in fADS (such as a very high transaction time for communication) this immediately shows up in the simulation results. More interesting, under dense traffic conditions an unexpected and very subtle error showed up in the simulations, which has to do with a combination of measurement errors in the OBU localisation and faulty correlation of OBU's with measured vehicle tracks. We never intended to have this subtle error in the fADS system. Only after the simulations we realised that under dense traffic conditions fADS is prone to such errors.

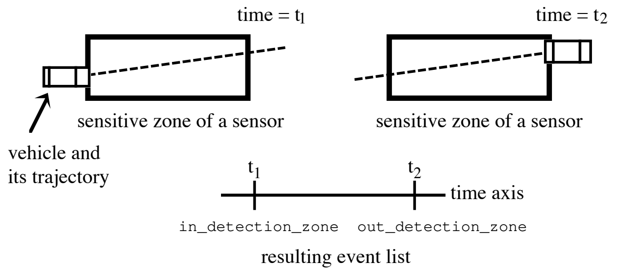

FIGURE 4: AUTOMATIC SCHEDULING, BY ADS-SIM, OF THE BASIC `IN_` AND `OUT_` EVENTS.

We must keep in mind that the fADS example is constructed to serve as an example. Validation of the model was of no concern in that case. In setting up models or a real ADS it is vital to our modelling approach, in terms of connecting models of sub-systems together in a discrete event model, that the sub-system models are validated and the range of validity of the models is known exactly [5].

The ADS-SIM tool is currently being finalised, and is used by the Dutch Ministry of Transport, Public Works and Water Management for evaluation studies of real ADS designs. ADS-SIM is currently designed as an evaluation tool. However, the modelling techniques are also very useful for the design and optimisation of an ADS. ADS-SIM will therefore be extended to be used as a design tool.

3. HIGH PERFORMANCE COMPUTING

In real ADS-SIM production runs we need to be able to extract SQF's which are very small, in the order of 10^{-6}. This number is typical for many safety - or reliability criteria of industrial installations. In order to extract such SQF's in a statistically correct way from the discrete event simulation means that we have to run extensive simulations. In our case of ADS simulations we have to track a few millions vehicles through the simulator. Given the execution times as reported in section 2.3, this shows that real production runs need a powerful HPC environment. Such large scale simulations require High Performance Discrete Event Simulations.

The most promising approach is to see to what extend ADS-SIM can use the limited parallelism offered by an HPC platform consisting of a small number of general purpose workstations in a network. The mean reason is that typical users of ADS-SIM have such HPC platforms readily available. The most straightforward way is to implement farming parallelism, where the traffic input files are split up and each processor in the network processes a part of the input. The current modular design of ADS-SIM easily allows for such parallelism, by executing identical copies of the Kernel (see Fig. 3) on each processor. The Framework will then coordinate this farming parallelism. Another approach would be to create a small functional pipeline, where for instance the traffic generator executes on one processor, and the Kernel is running on another. Obviously, hybrids of both methods are also possible. In the future we will further investigate such approaches towards parallelism.

4. CONCLUSIONS & FUTURE WORK

We have developed a modelling and simulation approach for ADS designs. The tool ADS-SIM is a generic discrete event simulation tool specifically tailored to handle ADS models. Currently we are in the process of modelling and simulating real ADS designs, and in the near future we will assess the overall strength of this simulation approach.

In future work we plan to investigate the validity of the discrete event assumption. The possibility exists that some sub-modules have to be represented by discrete time models, resulting in a hybrid discrete-time-discrete-event model of the ADS. Our current experience in modelling ADS designs suggests that the discrete event model is suitable. However, it raises subtle issues in e.g. defining and modelling virtual sensors.

ACKNOWLEDGEMENTS

This work was carried out under a contract with the Dutch Ministry of Transport, Public Works and Water Management. We thank D. van Wijk and H. Hadderingh of the Ministry for the many stimulating discussions. We acknowledge close cooperation with T. van der Post, G.J. van Dijk, and G. Meyer from CMG Den Haag B.V., division Advanced Technology, in the realisation of the ADS-SIM environment.

REFERENCES

[1] A.G. Hoekstra et al., High Performance Discrete Event Simulations to evaluate Complex Industrial Systems, the case of Automatic Debitting Systems for Electronic Toll Collection on Motor Highways, Accepted for *High Performance Computing & Networking Europe 1997*, will be published in *Lecture Notes in Computer Science*.

2. T.C. Henderson E.Silcrat, Logical sensor systems, *Journal of Robotics Systems*, 1, 1984, 169-193.
3. G.A. Weller, F.C.A. Groen, L.O. Hertzberger, A sensor processing model Incorporating error detection and recovery, in *Traditional and Non-traditional Robotic Sensors*, T.C. Henderson ed., NATO ASI series Vol.63, 1990, Springer Verlag, 351-364.
4. G-J. van Dijk, R. Poestkoke, G.R. Meijer, A methodology for performance evaluation of electronic toll collection systems, in *Proceedings of the Third Annual World Congress on Intelligent Transport Systems*, Orlando, USA, 1996, 14-18.
5. P.L. Knepell and D.C. Arangno, *Simulation Validation, A Confidence Assessment Methodology* (IEEE Computer Society Press, 1993).