©1996 F. Cascio/The Stock Market

56

# Evaluating AUTOMATIC DEBITING $¥STEM$ by modelling and simulation of virtual sensors

*L. Dorst, A. Hoekstra, J.M. van den Akker, J. Breeman,*
*F.C.A. Groen, J. Lagerberg, A. Visser, H. Yakali, L.O. Hertzberger*

1053-5888/98/$10.00©1998IEEE

The Dutch government is considering placing Automatic Debiting Systems (ADS) for electronic fee collection (EFC) on the highways (in Dutch: 'Rekening Rijden'). These systems would interact via a transponder in each passing car, and subtract a fee from the driver's credit card. Nonpayers would be photographed and fined. The ultimate goal is to use these systems to influence road usage (see Box 2). There are very strict demands on performance, with error rates of the order of $10^{-5}$ to minimize the number of incorrect decisions, since those could make the systems socially unacceptable.

Specifications have been sent out [2], and several consortia are bidding with systems of various designs. The proposals of contractors have to be evaluated in their performance with respect to the (unfortunately) rich set of Dutch weather conditions, traffic flows, and vehicle types. The evaluation of the systems is to be done on a schedule permitting the Dutch government to plan the introduction, which in some cases is before the first prototype has been built. (Similar systems have been built and tested [1]; however, the present actual system requirements are so different that new designs have been made.) There are so many scenarios, the system's internal structure is so complicated, and the demands so severe, that an evaluation in an actual test setup is not practical, even if prototypes were available.

For all these reasons, it was decided to perform the evaluation to a large extent in *simulation models*, to be validated with available test results. It is important to determine the appropriate level of abstraction for the correct and quantitative evaluation of the proposed systems; to design a simulation kernel that permits contractors to implement models of their systems at this level of abstraction; to provide the proper statistical functionality to analyze the results of the simulation; and to have the simulation compute quickly enough to use the results (which in this case means about 2 million cars per day of simulation time).

In this paper, we show how the concept of *virtual sensors*, designed for goal-directed sensing in (robotic) autonomous systems [3], can be used in the design of the simulation. We also show how this forces the choice for a *discrete event simulation* [5], which in turn affects the implementation of the virtual sensor concept.

## A typical ADS

A typical ADS system, satisfying the specifications of the Dutch Ministry of Transport, Public Works & Water Management, consists of the following functional components.

### *Communication with OBU, and localization of OBU*

The vehicle has an OBU (On Board Unit) containing the driver's means of payment, with which the ADS communicates. The ADS can assign a temporary identifier to the vehicle, but information like the license plate is not permitted to be broadcast, for reasons of privacy. If communication fails because the user does not pay, the user needs to be registered. During communication, all contractors determine the approximate *location* of the OBU using the phase of the signal. Indeed, proposed solutions by the contractors for the communication module are all similar, and based on well-established radio-communication standards.

### *Sensing and tracking the car body*

One needs to be able to sense and track vehicles. These can be matched with successful communications, to know which OBU is related to which vehicle; but of course noncommunicators also need to be tracked, and registered. Especially in systems that have a considerable spatial extent (e.g. several gantries with equipment) this is a critical part of the total functionality. Proposed contractor solutions differ widely in their use of medium (visible or infrared light, natural or artificial light sources, etc.) and type of sensing device (1-D curtain, camera, etc); they may also differ considerably at the more abstract level of sensor interpretation.

### *Registration of license plate*

The system needs to take a photographic picture of the front and rear license plate of all offenders; this is called *registration*. If the point at which registration is done is different from the location of detection and/or communication, proper tracking is very important.

Thus there are many aspects in evaluating the user proposals, including not only the performance of physical sensors but also that of the software tying the components together into one functional ADS unit. One expects errors in sensor interpretation, in the capability of the sensors to separate the different vehicles under various traffic and weather conditions, and in keeping track of the correspondence between these various aspects.

## The simulator ADS-SIM

A sketch of the components of our simulator ADS-SIM [4], based on the components of the actual systems, is given in Figure 1. This figure shows that ADS-SIM provides a framework containing a traffic simulator and statistical data processing, and that into this framework modules are to be defined which model the actual ADS of each contractor. These modules are fed with generated simulated traffic; their behavior should be validated to provide the same functionality as the actual components, to a level of detail to be determined by the accuracy required in the evaluation.

In the Rekening Rijden project, it is our task to provide clear guidelines for the choice of the level of abstraction, and to support that level in the utility functions provided in ADS-SIM so that the contractors may build their modules consistently and without too much effort. (In the following, the contractors will be called the *users* of the simulator.) To do so, we applied the concept of a *virtual* sensor (see, e.g., [3], where it is called 'logical sen-

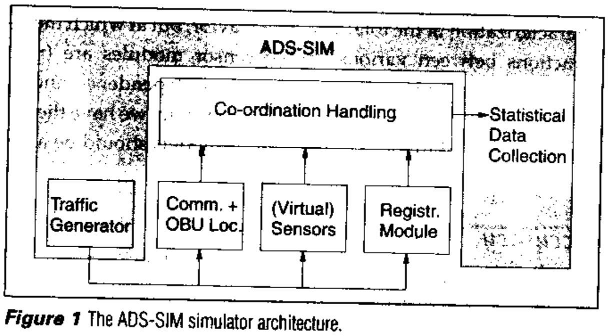


***Figure 1*** The ADS-SIM simulator architecture.

sor'), which is a (real or imagined) combination of a physical sensor with data processing and interpretation software, capable of measuring one of the relevant parameters characterizing the system. (We give a more precise definition in section 2.)

As a first step in our analysis, passages of vehicles through an ADS (whether actual or simulated) can be characterized by parameters of four types, differing in the rate of change across the various scenarios that are to be evaluated.

*Configuration parameters* describe the fixed layout of the ADS, constant during its operation (examples: the number of lanes, type of road surface, fixed angle of a sensor, etc.). This data follows from user designs.

*Scenario parameters* describe the circumstances of the experiment that is being simulated (examples: weather parameters, average speed of traffic, average distance between vehicles, etc.).

*Fixed vehicle parameters* are the unchangeable parameters for a vehicle whose passage under the ADS is simulated (example: the various shape parameters of a vehicle, the license plate type, color of the vehicle, etc.). Many of these parameters are known for Dutch (or European) traffic.

*Dynamic vehicle parameters* are the parameters of the vehicle that depend on its actual passage in the context of the rest of the traffic (example: speed, track, type and distance of other vehicles around it, etc.). These are the consequences of driving behavior. Data on dynamics models for Dutch drivers has been gathered.

This is the set of parameters that has to characterize the performance, independent of the proposed solutions of the various contractors. A correct comparison and evaluation is thus done if we perform the modelling consistently at this level (subject to the condition that each of the models can be validated at this level of abstraction!). In particular, the sensor modelling (both in conceptualization and in implementation) should be done *at the level of the fixed and dynamic vehicle parameters.* More precisely, the sensing system should be described as having only those parameters as output (or possibly some combinations of them), but *not* any of the lower level measurement data on which these measurements are actually based. We therefore describe the sensing system as a set of virtual sensors.

## The Virtual Sensor Model

In any goal-directed sensory system, a *virtual sensor* is a (conceptual) device whose output can be modelled in terms of the relevant characterizing parameters and the outputs of other virtual sensors. The virtual sensor modules should be chosen at the highest level of abstraction that enables a sufficiently accurate characterization of the total system behavior, but at which the interactions between various virtual sensor modules are (relatively) simple, both in their statistical (in)dependence and in their causal relationships. In a simulated system, we have the additional demand that the virtual sensor models should be amenable to being *validated.*

### *Accuracy*

A virtual sensor measuring a parameter $h$ will do so with a limited accuracy. This accuracy has various aspects, all reflected in the distribution of the measured $h$ under similar circumstances, in a great number of trials. There are two basic sources of inaccuracy: *sensor noise* of the physical sensors within the virtual sensor, and the fact that the available parameters of the virtual sensor can only characterize certain *ensembles* of inputs and input scenarios (for instance, the roof height of a vehicle may vary along its length due to roofracks, which would not be modelled in detail in the virtual sensor for height); this ensemble will exhibit an inherent spread, which manifests itself as an error distribution in the virtual sensor output in dependence of the characterizing parameters.

The error distribution of the inaccuracy is affected by specific parameters, especially by the scenario parameters and the characterizing parameters of the input objects (which in the ADS can be the vehicle parameters). The modelling of this dependence, using only a relevant subset of this suite of parameters, is what constitutes the *statistical aspects* of the virtual sensor model. Actually establishing these statistical aspects involves a combination of engineering insights (as to the relevance of the parameters), parameter estimation techniques, and nonlinear modelling. Constructing them is a responsibility of the user. These virtual sensor models must be validated before they can be used in the evaluation.

### *Timing*

Virtual sensors also have a *timing*: *when* is the parameter measured indeed available; and at what temporal rate does it change significantly. *A higher level of abstraction implies a lower rate of change.*

For virtual sensors defined for parameters that are defined on the input of individual objects, it only makes sense to talk about the parameters of their distributions when there is actually such an object in the system (in an ADS, for example, only when there is a vehicle in the ADS does it make sense to consider the parameter 'length of a vehicle'). Thus the parameters measured by a virtual sensor only occur in a certain time interval, characterized by initial and final time. In the desired abstraction, we model the variation within such a time interval statistically (if necessary subdividing the interval when intermediate occurrences are important to the statistical behavior), and we characterize the time lots discretely, as specific *events* on the time axis. It is not necessary to sample the time axis equidistantly. The events and the times at which they occur are implicitly determined by the virtual sensors considered. We thus obtain arbitrary discrete events that need to be handled within the simulator in their proper order.

Our simulator ADS-SIM is thus naturally implemented as a *discrete event simulation* (see [4], [5], and Box 2). Such a system processes events at discrete times, which spawn actions which lead to other discrete events, as determined by the *causal* structure of the simulated system. A discrete event simulator handles these events in order of occurrence on the simulated time axis. There is one important property which affects the embedding of virtual sensors into such systems: *events once scheduled on this axis cannot be unscheduled.* This complicates the implementation of causally, not fully independent virtual sensors, as we will see below.

Figure 2 shows schematically how the virtual sensor models in ADS-SIM generate events (called "VehicleMeasured") with a *time stamp*, and parameters characterizing their statistical distribution.

## Box 1: Modelling and Simulation

Using computers to "experiment" with natural or man-made systems, by building a suitable *model* and running *simulations*, has become a well-accepted alternative way of doing science and engineering (alongside developing theory and performing "real" experiments). Although this "third way" has strong roots and a certain tradition within a number of disciplines in science (e.g. computational physics) and engineering (e.g. computational electromagnetics or computational fluid dynamics), it has only recently received more interest. It has matured into a new discipline: Computational Science and Engineering. This development can be partly attributed to the availability of relatively cheap, but very powerful, high-performance computing and networking environments. Recent developments in this field are reported in the journal *IEEE Computational Science & Engineering.*

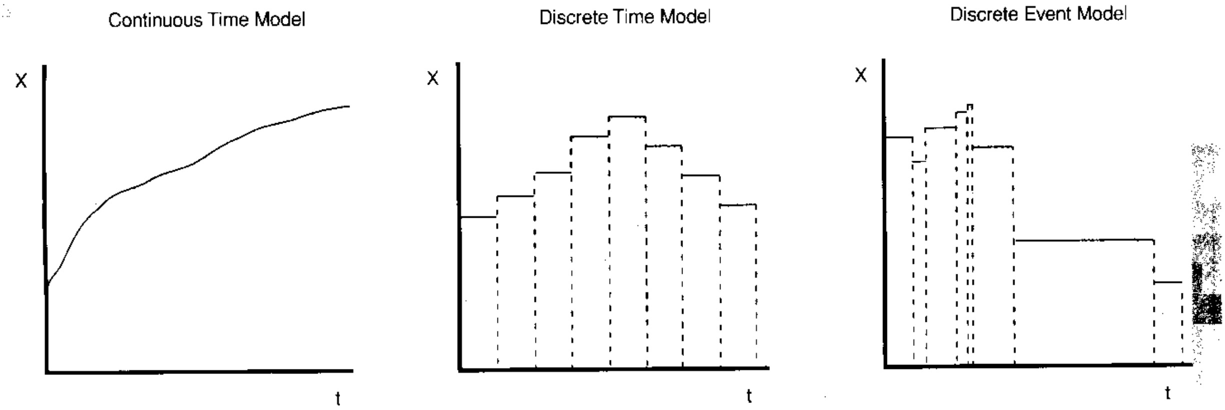
Arguments to perform simulations can be the following: the actual physical system is not yet available (applies to the case of this paper); the experiment may be dangerous; the cost of the experiment is too high (also plays a role in the current case); the time constants of the system are not compatible with those of the experimenter (in our case, a real field test would require many months or even years to get enough statistics, which is not acceptable); or control variables may be inaccessible. In many realistic cases, especially for engineering, it turns out that modelling and simulation is highly effective for rapid prototyping or to allow for rapid design for experimentation. The key issue is always to develop models of the real system that have to be validated to a certain degree of accuracy (depending on the exact application).

One can distinguish three types of mathematical models (see figure): *continuous time models* in which state variables change continuosly with time; *discrete time models* where the state variables change their value at regular discrete time intervals; *discrete event models* where the state variables change on well-defined time stamps.

Because of their nature, discrete time and discrete event models are suited for implementation on a digital computer. How to arrive at a suitable model is far from trivial. Usually, a system is modelled by a set of partial differential equations (i.e. continuous time) that need to be discretized to arrive at a discrete time model.

In our case of an Automatic Debiting System we have taken a high-level modelling approach. We model an ADS in a top-down fashion. This means that we hierarchically decompose an ADS into a small number of subsystems. The subsystems themselves are modelled using (known) parametrizations. We assume that the state of a subsystem changes on predefinable time-stamps, due to external inputs and changes in other submodules. This assumption results in a discrete event model of an ADS.

In many cases modelling and simulation is very succesful. However, its generality and ease of use is both its strength and weakness. Quoting F. Cellier from his book *Continuous System Modelling* (Springer-Verlag, 1991): "All too often, simulation is a love story with an unhappy ending. We create a model of a system, and then fall in love with it. Since love is usually blind, we immediately forget all about the experimental frame, we forget that this is not the real world, but that it represents the world only under a very limited set of experimental conditions (we become model addicts)." Taking these words of warning into account, modeling and simulation becomes an important tool in engineering, especially in cases where we have to deal with highly complex systems like an Automatic Debiting System.



### *Validation*

Our simulation should give a realistic representation of reality, at the desired level of evaluation accuracy for each scenario processed. This implies that all elements in the simulation should be validated as indeed representing the real system at the abstraction level chosen. This is fairly straightforward if the virtual levels are also present in the actual system (which is often the case since they coincide with the sound design engineering choices of 'intermediate results'). But in some cases, the virtual sensor models need to be validated on the basis of a combination of actual data from physical sensors, mathematical analysis and software verification. In practice, this may influence the most convenient level at which to define the virtual sensors; it makes little sense to define an abstract model that cannot be validated.

### *(In)dependence*

Although not strictly necessary, it is desirable to choose the level of abstraction such that the parameters at that level are fairly independent, so that their interactions can be modelled simply. This is important for the validations of the models (since one does not need to take into account their interdependence) and also for the number of cases that have to be evaluated separately in the simulator. Virtual sensors will thus preferably be chosen in a way that makes their interdependencies, *both in statistics and in timing*, simple and measurable (all the more so in a discrete event simulator, for which dependencies in the timing of events generated by virtual sensors are awkward to handle).

## Implementing Virtual Sensors in ADS-SIM

Applying these issues in the virtual sensor concept to the ADS simulation is not straightforward. The concept of 'virtual sensor for a relevant parameter' may be a level of abstraction to treat statistics, timing, and validation in a unified manner—and to guarantee consistency and completeness of treatment—but it does *not* uniquely specify the implementation of these models as software modules.

### *The choice of virtual sensors*

We have already indicated that a natural level of virtual sensors for the ADS is that of the vehicle parameters, defined in the first section above. In the Rekening Rijden project, this was *not* the level at which the users thought about their system (rather, that tended to be determined by the physical sensors and their data flows). As a consequence, higher level data processing functions in the simulator and the actual ADS could differ, complicating the validation of the model. In practice, several users found the 'virtual sensor' way of considering their system advantageous, and they adapted the data processing in the *actual* system to the enforced structure of the simulator model!

### *Basic events*

When a vehicle passes under an ADS system, at the level of vehicle parameters very few events are salient. Basically, a particular virtual sensor involves the functionality of one or several event handlers:

- The virtual sensor becomes active when a vehicle enters an appropriate detection zone, as marked by the event "InDetectionZone," generated automatically by the kernel of ADS-SIM for each (possibly composite) physical sensor, based on *geometrical* information on its sensitive zone (ADS-SIM contains a Geometry Library to generate such events).
- It then schedules salient events in acquiring information about the parameter it models (these are called "VehicleMeasured" events).
- The set of virtual sensors is designed under certain assumptions of independence of parameters and vehicles; the checking (and, if necessary, mending) of those assumptions is performed by the "VehicleMeasured" or "DetectionCompleted" event handlers. These actions are artifacts of the discrete event simulation, not present in the actual system; we will discuss these actions as *coordination handling*, below.

### *Timing*

The timing of a VehicleMeasured event depends on the geometry of the ADS for the physical sensors on which the measurement of that virtual parameter is based. Effectively, it may be viewed as the specification of a *sensitive zone* of the virtual sensor for a particular parameter and vehicle, from the actual sensitive zone of the physical sensor(s), the salient points of the data processing, and the vehicle motion from the traffic generator.

One would like to have the correct statistical data for the event from the virtual sensor model, evaluated at the appropriate time. But the appropriate time cannot always be foreseen when the statistical model is evaluated, which (in the simulator) is at the moment that the InDetectionZone event is handled. This is especially true when there is an interdependence of the statistics for different virtual sensors. There are basically two possibilities to ensure proper evaluation, one involving the coor-

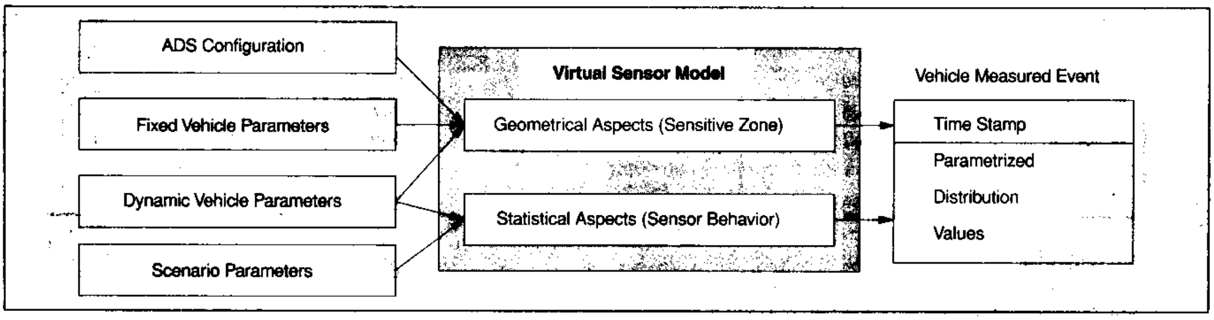


***Figure 2*** The virtual sensor model for a VehicleMeasured event.

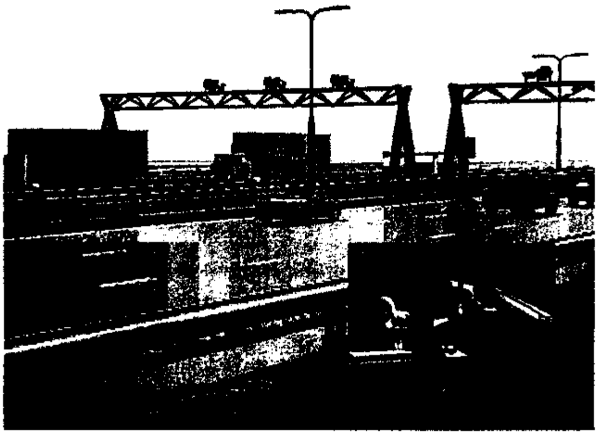
## Box 2: Electronic Toll Fee Collection to fight traffic jams

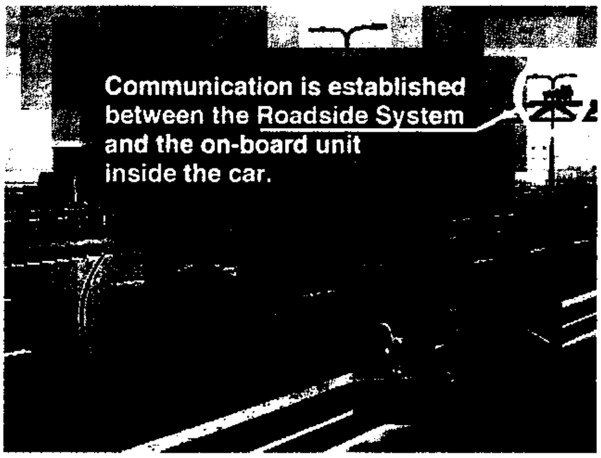
Driving in the western part of The Netherlands during rush hours is an exercise in patience; all major highways are completely blocked by traffic jams. If the Dutch government does not take any action is expected that The Netherlands would soon come to a complete standstill. Already today, economic losses due to traffic jams are great. For instance, transportation of goods from the Rotterdam harbor is severely hindered, and getting to and from Amsterdam Schiphol airport by car takes too much time.

The Dutch government is taking many initiatives to fight the road congestion. One such initiative is to implement a system of Electronic Fee Collection (EFC), which has to be operational in the year 2001. By installing a large number of ADS's on the Dutch road network, and by letting drivers pay a certain amount of money for each passage through an Automatic Debiting System (ADS), the government hopes to reduce the amount of traffic during rush hours. Only a relatively small decline in traffic would drastically decrease the number of traffic jams. The toll fee should move drivers to either use an alternative for their car (public transport) or to share a car with a number of people (i.e., car pooling, over 80% of the cars during rush hour contain only one person).

The toll collection function should not interfere with normal traffic flow. Therefore, a free flow ADS is required. The idea of EFC is that each vehicle has an On Board Unit (OBU) on its front screen. The OBU consists of an electronic purse (a chip card) and a transponder which can communicate with the Road Side System (RSS). The RSS is a gantry on the road, containing all the equipment of the ADS. An impression of an ADS is drawn below.

Typically one should introduce a suitable chip-card with enough money on it in the OBU when starting to drive. On approaching an ADS it will start to communicate with the OBU and charge an amount of money from the chip-card. A possible way of operating the system would be that if a driver does not have an OBU, the ADS will take a picture of the licence plate; after a while the driver will receive a bill that will be larger than electronic payment would have been.





dination handling, and the other the way the virtual sensor model is implemented:

- The simulator only starts the coordination computations when CoordinationDecision events have been scheduled for the parameters that need to be coordinated.
- Virtual sensor models are implemented so that they can be evaluated at arbitrary times, i.e., as *validated sensor functions* with time-dependent distribution parameters.

### *Coordination handling*

The InDetectionZone event handler leads to the scheduling of VehicleMeasured and DetectionCompleted events for the particular virtual sensor under consideration, plus CoordinationDecision events for those sensors that play a role in the coordination. These events have event handlers whose task is twofold:

- Reality checking: occlusion management

In the actual ADS system, vehicles may occlude each other. This makes certain measurements, or the registration, impossible. In the simulation, we have defined the measurements by setting up virtual sensors which are mostly independent in their statistics and scheduling—independent for various parameters of a vehicle and, more seriously, *independent across vehicles.* Such virtual sensors cannot take occlusion into account when scheduling their events.

As a consequence, the user will need to ensure explicitly that actually occluded measurements are not processed, or generate the more complicated events that may occur when the system gets mixed up and generates nonsensical measurements. The proper place for this is in DetectionCompleted and/or CoordinationDecision handlers. Note that this software has no counterpart in the actual system, where Nature itself does the occlusion — it is an artifact of the simulator!

- Actual coordination: sensor fusion and classification

Once the occluded data has been removed, the coordination of multi-sensor and multi-vehicle events in the simulator corresponds closely to the real coordination module in the user's actual ADS system: it is mostly the 'logic' of that module, separated into each of the virtual sensors involved. In many cases, for these tasks one may use the actual code for the real ADS in ADS-SIM.

### *Validation*

The virtual sensor models need to be validated. This is an important test of the viability of the virtual sensors concept. It involves the design and estimation of quantitative models for the statistics, mathematical analysis of the error propagation through deterministic steps, and line-by-line verification of the modules in the simulated ADS that are virtually identical to the actual modules. This is the most time-consuming and controversial step in the modelling, mainly because the architectures of the simulated model and the actual system may be so different that the 'corresponding' measurements are hard to define.

## Conclusions

In our approach, the design of the simulation model is only partly based on the actual physical characteristics of the proposed ADS system; the overriding influence on the model is the need for well-defined discrete events. We found that consistent application of the abstract concept of 'virtual sensor' streamlines the design process:

- It clarifies the modelling required for simulation considerably, since it relates the various event handlers as different aspects of the same natural concept: the virtual sensor of a specific vehicle property.
- It minimizes the time required to get a realistic simulation running, since only the essential accuracies of the relevant events and parameters need to be validated and estimated. No detailed models of physical sensors will have to be built in the Rekening Rijden project.
- Of all components, only the relevant details for the application are present, and modelled explicitly as statistical models, with (causal) interdependencies. As a consequence, the simulation runs faster than one that contains full, but ultimately irrelevant, detail (such as a full simulation of the physics of the communication process).
- The description of many physical sensors in the same virtual spatio-temporal model permits us to provide standard utility functions for the necessary spatio-temporal calculations, for all users, independent of their physical sensors (we provide a Geometry Library in ADS-SIM).
- If an ADS system design should not live up to its specifications, the virtual sensor architecture makes the tracking of the design errors causing this more effective. In that way, ADS-SIM is not only useful as a simulator of a completed design, *but also as a design tool*. Several users were quick to appreciate this!

Users have currently (end of 1997), implemented their ADS models at the level of virtual sensors according to our guidelines, and validated the statistical dependence on the various parameters.

Most found this approach enlightening. Individual contractors found that in *simulation design* of their system, they needed to descend one more level to get data and events more closely corresponding to the physical proposal. They use ADS-SIM to generate and process these more detailed events, and the resulting designs are then again modelled at the virtual sensor level for their evaluation and comparison. Thus the virtual sensor concept has played an essential and useful role in structuring the RekeningRijden project.

## References

[1] W. Asmuth et al., *Feldversuch Autobahntechnologien A555: Ergebnisse und Vorschläge*, TÜV Rheinland, Proj. no. FP 9.93906F1, 1995.

[2] Dutch Ministry of Transport, Public Works & Water Management, *Requirements specification for the critical processes of the Rekening Rijden system on behalf of the evaluation phase*, DPTR/221295/14.021/II, 1995.

[3] T. Henderson, E. Shilcrat, "Logical sensor systems," *Journal of Robotic Systems*, vol. 1, no. 2, pp.169-193, 1984.

[4] A. Hoekstra, et al., "High performance discrete event simulations to evaluate complex industrial systems, the case of automatic debiting systems for electronic toll collection on motor highways," *High Performance Computing & Networking Europe 1997*, Springer Lecture Notes on Computer Science 1225, pp. 41-50, Vienna: Springer-Verlag, 1997.

[5] S. M. Ross, *A course in simulation*, MacMillan Publishing Company, 1990.

***Leo Dorst** has M.Sc.(1982) and Ph.D.(1986) degrees from the Applied Physics Department at Delft University of Technology, The Netherlands. His Ph.D. thesis was on accurate geometrical measurements in discretized images. From 1986 to 1992 he worked as senior research scientist at Philips Laboratories, Briarcliff Manor, NY, USA, focusing on robot path planning and task abstraction in goal-directed systems. Since 1992, he has been an assistant professor at the University of Amsterdam, The Netherlands, where his continued interest is on planning and representation in autonomous systems, with a recent emphasis on reasoning with uncertainty, and the use of Clifford algebra for geometric representation and computation.*

***Alfons Hoekstra** has an M.Sc. (1987) from the Applied Physics Department at Twente University of Technology, The Netherlands, and his Ph.D. (1994) degree from the Comuter Science Department at the University of Amsterdam, The Netherlands. His Ph.D. thesis was on high performance computing and applications in the area of computational electromagnetics. Since 1994 Hoekstra has worked at the computer science department at the University of Amsterdam, since 1996 as assistant professor. His current research is in the field of computational science, with an emphasis on modeling and simulation of natural phenomena using particle based methods.*

***Marjan van den Akker** has M.Sc. (1990) and Ph.D. (1994) degrees in Mathematics from Eindhoven University of Technology, the Netherlands. Her Ph.D. thesis is in the area of combinatorial optimization and has as its subject 'LP-based algorithms for single-machine scheduling problems.' In 1995, she held a post-doc position at the*

*Catholic University of Louvain, Belgium. Since then, she has been a research engineer in the Informatics Division of the National Aerospace Laboratory NLR, the Netherlands. She works on modelling, optimization, and simulation with applications in Air Traffic Management and Road Pricing (Rekeningrijden).*

***Jan Breeman** has a M. Sc. degree in Physics from Delft University of Technology, The Netherlands. Since 1971, he has been a research engineer at the National Aerospace Laboratory NLR, The Netherlands, working in the area of flight instrumentation. His work of the field of flight testing includes the development of a Flight Data Processing Facility. He was also involved in research on the theory and application of navigation systems and in flight simulation. Recently, he started working the Informatics Division, where he mainly works on Road Pricing (Rekeningrijden).*

***Franciscus C.A. Groen** received his PhD in 1977, from the Delft University of Technology. He was a staff member of the Pattern Recognition Group of the Department of Applied Physics of the Delft University of Technology and full professor in Physics Applied Computer Science at the Free University of Amsterdam (part-time). His current appointment is full professor in the special applications of Computer Science at the University of Amsterdam. In 1984 he was a Fulbright Research Scientist at the Robotics Institute of Carnegie-Mellon University in Pittsburgh and in 1996 as visiting professor at the University of Utah. His research interest is in the field of intelligent autonomous systems and sensor data processing. He is Editor-in-Chief of the journal Robotics and Intelligent Autonomous Systems.*

***José Lagerberg** studied mathematics at the University of Leiden, The Netherlands. After working on mathematical software development in a number of industrial companies, she began at the Computer Science Department of the University of Amsterdam in 1988. She has worked on diverse topics in the field of robotics, such as Kalman filtering for position estimation and sensor based obstacle avoidance.*

***Arnoud Visser** studied physics at the Univeristy of Leiden, The Netherlands, where he performed investigations in the field of nonlinear optics. At the University of Amsterdam since 1991, he participated in several international projects (ESPRIT, ESTEC) for the Computer Science Department. Currently he is working on his dissertation on decision support for autonomous robots.*

***Hüseyin Hakan Yakali** has M.Sc.(1989) and Ph.D.(1994) degrees from the Department of Electrical Engineering at Florida Atlantic University in Boca Raton, Florida. His Ph.D. thesis was on extraction of 2-D visual cues for autonomous vehicle control. In 1994, he started working as a Post Doctorate at the University of Amsterdam, The Netherlands, where he has been working as a researcher since 1996. His areas of interest are in robotics, controls and autonomous systems.*

***Louis O. Hertzberger** received his Master's degree in experimental physics in 1969 and his PhD in 1975, both from the University of Amsterdam. From 1969 till 1983 he was a staff member in the High Energy Physics group, later the NIKHEF-H (Dutch Institute for Nuclear and High Energy Physics). As an instrumental physicist he was involved in a number of experiments, mostly at CERN, Geneva. In 1983 he became a full professor in computer science at the University of Amsterdam. His current research is in the field of High Performance Computer Systems as well as complex- and autonomous systems research. He has participated in a large number of parallel computer projects with industrial companies such as Philips.*

*The authors can be contacted in care of the Rekening Rijden Project, Department of Computer Science, University of Amsterdam, The Netherlands National Aerospace Laboratory NLR, The Netherlands.*

*This research was performed with the financial backing of the Dutch Ministry of Transport, Public Works & Water Management, for the 'Rekening Rijden' project. M. Bergman (University of Amsterdam) and T. van der Post (CMG) were instrumental in the programming of the ADS-SIM code.*