The virtual sensor concept applied to simulation: the evaluation of Automatic Debiting Systems

L. Dorst, F.C.A. Groen (groen@wins.uva.nl),
J.M. van den Akker[†], J. Breeman[†], A. Hoekstra,
J. Lagerberg, A. Visser, H. Yakali, L.O. Hertzberger

Rekening Rijden Project,
Faculty of Mathematics, Computer Science, Physics and Astronomy,
University of Amsterdam, Kruislaan 403,
1098 SJ Amsterdam, The Netherlands
[†] National Aerospace Laboratory NLR, The Netherlands

Abstract

An ADS (Automatic Debiting System) is a toll-collecting system on a freeway, which interacts with a transponder in each car, and subtracts a fee from a credit card. Non-payers are to be photographed and fined. In requirements on such systems in The Netherlands, privacy laws demand separation between the financial transaction and the registration; the system should not be usable for tracking citizens. There are also very strict demands on performance, with error rates of the order of 10^{-5} to minimize the number of incorrect enforcements. Both demands combined require complex ADS systems.

This paper describes a simulation of the proposed ADS's of bidders, to be used for their quantitative evaluation. It is highly important to chose the proper level for the abstraction of the simulated system, both to keep the simulation within reasonable cost and time, and also to establish that the simulated system indeeds represents the actual proposal. In this paper, we show how a consistent application of the concept of *virtual sensors* can be applied to clarify the choice of the proper level of abstraction for the ADS proposals, and its implementation. It leads to a simulation that is faster, but no less specific in its quantitative results than a full implementation of the ADS.

I. Evaluation of Automatic Debiting Systems (ADS)

The Dutch government is considering to place Automatic Debiting Systems (ADS) on the freeways (in Dutch: 'Rekening Rijden'). These systems would interact with a transponder in each passing car, and subtract a fee from the driver's credit card. Non-payers would be photographed and fined. The ultimate goal is to use these systems to influence road usage. 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 would make the systems socially unacceptable.

Specifications have been sent out [2], and several consortia are bidding with systems of various designs. The proposals of users bidding for the placement of such systems have to be evaluated before they are considered, in their performance under the (unfortunately) rich set of Dutch weather conditions, traffic flows, and vehicle types. The evaluation of the systems is to be done on a time-scale permitting the Dutch government to plan the introduction, which in some cases is before the first prototype can have been built. (Similar systems have been built and tested [1]; however, the present system requirements are sufficiently 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 practicable, even if prototypes were available.

For all these reasons, it was decided to perform the evaluation to a large extent in *simulation*, validated with available test results. It is important to determine the appropriate level of abstraction for the fair and quantitative evaluation of the proposed systems; to design a simulation kernel that permits the bidders 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 sufficiently fast (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] is used in the simulation design. We also show how this forces the choice for a *discrete event simulation* [6], which in turn affects the implementation of the virtual sensor concept.

A. A typical ADS

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

- *communication with OBU, and localization of OBU:*
  The vehicle has an OBU (On Board Unit with the driver's credit card), 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 want to pay, the user needs to

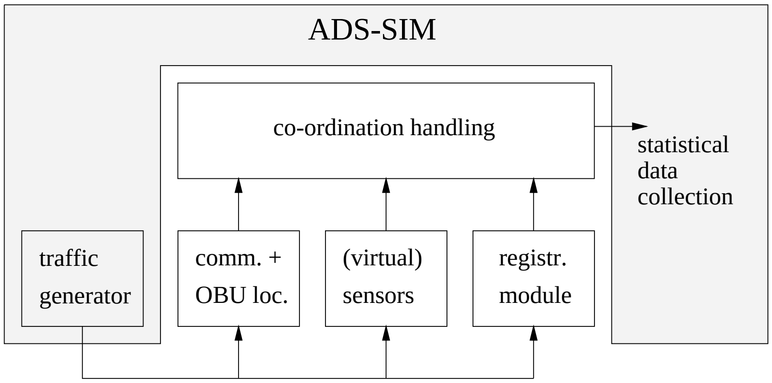

Fig. 1. *The ADS-SIM simulator architecture.*

be registered. During communication, all bidders determine the approximate *location* of the OBU using the phase of the signal. Indeed, proposed solutions by the bidders 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 what OBU is related to what vehicle; but of course also non-communicators need to be tracked, and registered. Especially in systems that have a considerable spatial extent (e.g. several gantries with equipment), this is an important, and critical, part of the total functioning. Proposed solutions of bidders differ widely in use of medium (visible or infrared light, natural or artificial light sources, etc.) and type of sensing device (1-D curtain, camera, etc); also, at a more abstract level of sensor interpretation, they may differ considerably.
- *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.* When the point at which registration is done is different from the location of detection and/or communication, proper tracking is very important.

There are thus many aspects in evaluating the user proposals: the performance of physical sensors, but also of the software tying the components together into one working ADS. 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.

B. 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 mostly containing a traffic simulator and statistical data processing; and that into this framework modules are to be defined which model the actual ADS. These modules are fed with generated simulated traffic; and 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 the task of th researchers at the University of Amsterdam to provide clear guidelines for the choice of the level of abstraction, and support that level in the utility functions provided in ADS-SIM, so that the bidders may build their modules consistently and without too much effort. (In the following, the bidders 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 sensor', and [5]), 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 II).

As a first step in our analysis, passages of vehicles through an ADS (whether actual or simulated) can be characterized by parameters of 4 different types, with different dynamics of changes in the various scenarios that are to be evaluated:

- *configuration parameters:* These describe the fixed layout of the ADS, fixed during its operation (examples: the number of lanes, type of road surface, fixed angle of a sensor, etc.). This data follows from the user's design.
- *scenario parameters:* These describe the circumstances of the experiment that is being simulated (examples: weather parameters, average speed of traffic, average distance between vehicles, etc.) Data about Dutch scenarios is being gathered.
- *fixed vehicle parameters:* These 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.)

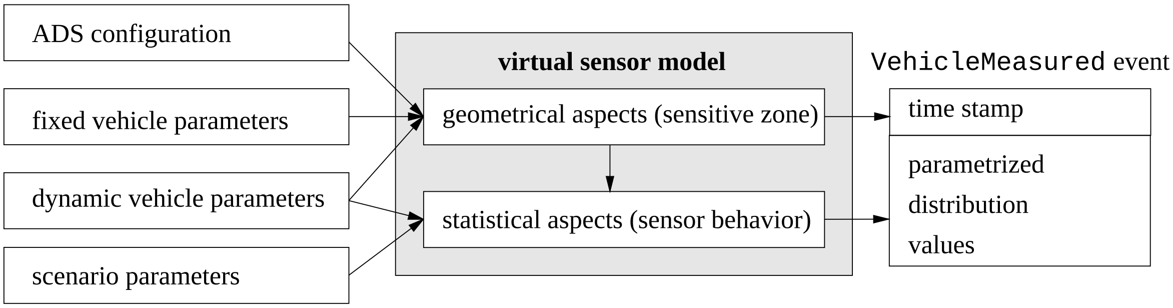

Fig. 2. *The virtual sensor model.*

Many of these parameters are known for Dutch (or European) traffic.

- *dynamic vehicle parameters:* These 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 consequence of driving behavior. Data on dynamics models for Dutch drivers has been gathered.

Since this is the set of parameters that has to characterize the performance, it makes sense to perform the modelling consistently at this level. 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 in which these measurements are actually based. In other words, the sensing system should described as a set of virtual sensors.

II. 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 the causal relationships. In a simulated system, we have the additional demand that the virtual sensor models should be amenable to being *validated.*

A. 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; 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 non-linear modelling. Constructing them is a responsibility of the user. These virtual sensor models must be validated before they can be used in a realistic evaluation.

B. 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: 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 occurs 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 if 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 equi-distantly; 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 dealt with in the simulator in their proper order.

Our simulator ADS-SIM is thus naturally implemented as a *discrete event simulation* [6][4]. 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.

C. 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 and 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.

D. (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. This all the more in a discrete event simulator, in which dependencies in the timing of events generated by virtual sensors are awkward to handle.

III. 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, 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 section I-A. In the Rekening Rijden project, this was *not* the level at which the users thought about their system; that tended to be determined by the physical sensors and their data flows.
- *Basic events*
  When a vehicle passes under an ADS system, at the level of vehicle parameters very few events are salient. Most are indicated in the table of Figure 3, with a brief description. It is obvious that the communication-related (virtual) sensors are like the other sensors in their event structure; we will treat them as such in what follows. Basically, a particular virtual sensor involves the functionality of one or several event handlers, see Figure 4:
  - 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 the geometrical information on its sensitive zone,
  - it then schedules salient events in acquiring information about the parameter it models (these are called `VehicleMeasured` events),
  - and 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; we will discuss these actions as *co-ordination handling.*
- *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 can not always be foreseen when the statistical model is evaluated,

event	description
`InDetectionZone`	specified vehicle entered specific detection zone
`OutDetectionZone`	specified vehicle left specific detection zone
`VehicleMeasured`	a specific vehicle parameter has been measured
`DetectionCompleted`	no more data are to be expected for this vehicle
`InCommunicationZone`	equals `InDetectionZone` for communication
`OutCommunicationZone`	equals `OutDetectionZone` for communication
`OBUMeasured`	equals `VehicleMeasured` for OBU location
`TransactionCompleted`	equals `DetectionCompleted` for communication
`ClassificationCompleted`	classification has been done
`CoordinationDecision`	start co-ordination (at specified time)

Fig. 3. *Some basic events in ADS-SIM.*

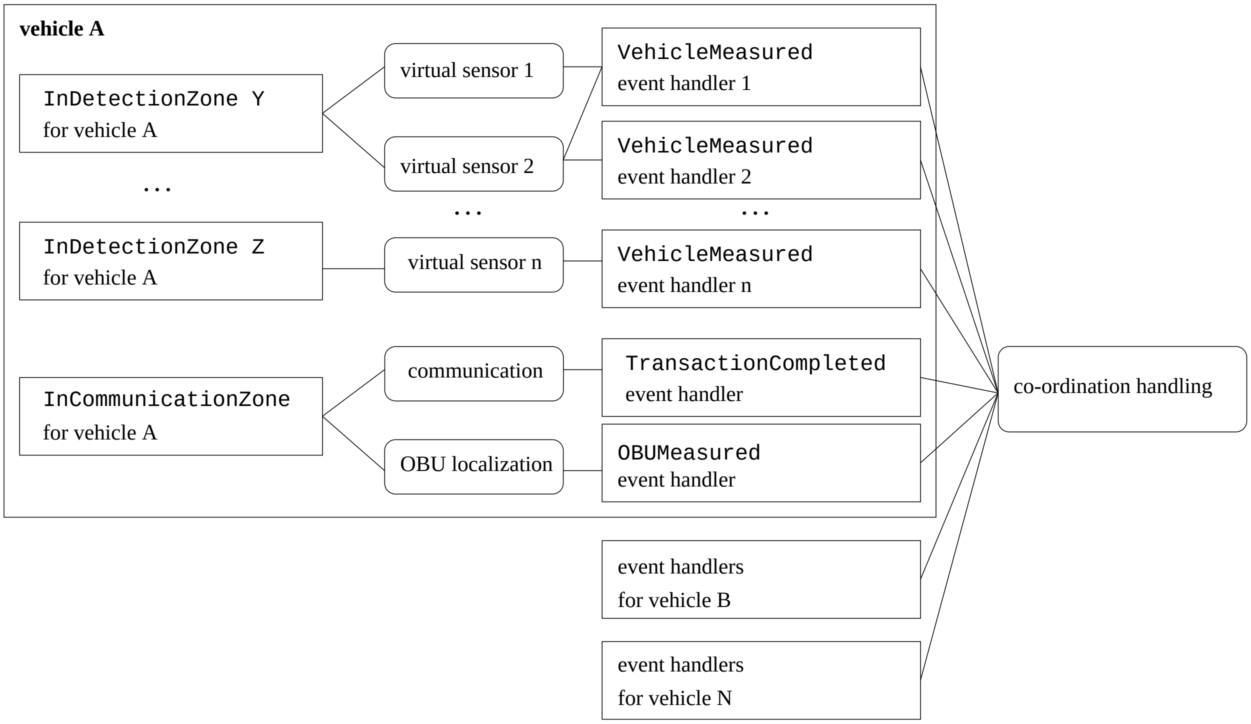

Fig. 4. *Event handlers associated with virtual sensors.*

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 co-ordination handling, and the other the way the virtual sensor model is implemented:

– The simulator only starts the co-ordination computations when the `CoordinationDecision` events have been scheduled for the parameters that need to be co-ordinated.

– 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.

- *Co-ordination 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 co-ordination. 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.

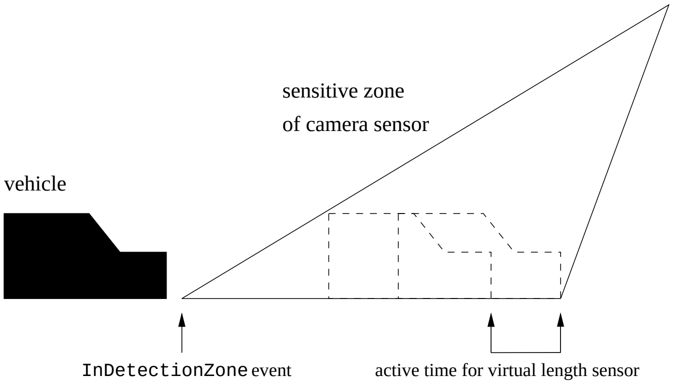

Fig. 5. *Measuring the length of a vehicle using a camera.*

- *actual co-ordination: sensor fusion and classification*
  Once the occluded data has been removed, the co-ordination of multi-sensor and multi-vehicle events in the simulator corresponds closely to the real co-ordination 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. We hope to report on the validation issues in Rekening Rijden in the future.

IV. An example of scheduling virtual sensor events

We describe the sequence of actions by means of a simple example: modelling the measuring of the length of a vehicle using a camera, see Figure 5. This parameter 'length' is a necessary virtual sensor level for a system whose actions depend on it, for instance since it uses the length in a classification of the vehicle to determine whether it has paid the proper rate.

This length measurement can only be made when both front and the rear of the vehicle have been seen (it is not necessary for both to be visible at the same time if the system is also capable of measuring the speed of the vehicle). But not necessarily will it be used in the ADS at the first opportunity, since waiting a little may give more accurate results. Here is how this measurement of the parameter 'length' might be modelled:

1. *When* the vehicle is likely to begin to be visible can be derived from the `InDetectionZone` event for the physical camera sensor; that event is scheduled by the kernel of ADS-SIM automatically, using the configuration and the traffic. This just establishes the presence of the vehicle within the sensitive zone; it is not particularly specific on which point of the vehicle enters the zone.
2. Therefore the event handling routine of the `InDetectionZone` event is used to compute the times a specific point of the vehicle enters and leaves the detection zone (in this case bottom front and top back), by combining data from the ADS configuration and the simulated traffic flow. In general, the answers will depend on the fixed parameters, such as its height and length, and on the dynamic vehicle parameters such as its speed.[1]
3. The answers of those two computations for the front and the back of the vehicle are sufficient to determine the time at which the ADS could know the length of the vehicle, and therefore a `VehicleMeasured` for this property can be scheduled.
4. The user has to provide a *validated module* which mimics the actual statistics of her (virtual) sensors; this is the module that computes the values to be included in the `VehicleMeasured` event. Usually, these values are based on elements from both traffic and from the scenario parameters. The result is that either sensor data or a sensor function that permits that data to be computed later is added to the `VehicleMeasured` event by the `InDetectionZone` event handler.
5. When no more information can be gathered for this virtual sensor, the user generates a `DetectionCompleted` event (or a `CoordinationDecision` event to enforce co-ordination handling).[2]

[1] This is a generic, basically geometrical, functionality and we have built a *Geometry Library* of utility functions (delivered with ADS-SIM) which should provide the necessary functionality to all users.

[2] Again, she may use the utility functions of the Geometry Library to compute the timing of these events.

6. Similar actions are taken for other virtual sensors whose scheduling depends on the same `InDetectionZone` event. They generate their own `VehicleMeasured` events, `DetectionCompleted` events and `CoordinationDecision` events.
7. When all data has been gathered from a particular physical sensor, the kernel generates an `OutDetectionZone` event.
8. Virtual sensors that depend on other sensitive zones are treated in other `InDetectionZone` event handlers.
9. Meanwhile, the kernel of ADS-SIM has started to perform co-ordination tasks by the functions inside the various event handlers, as their discrete time comes up on the simulated time-line.

This flow of events, for the whole simulated traffic flow, produces the data which is used in the statistical analysis module to compute the quality factors that are the goal of the evaluation.

V. Conclusions

Abstraction using the virtual sensor concept has the following advantages:

- 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, in our case as a Geometry Library within the ADS-SIM package.
- If an ADS system design should not live up to its specifications, the virtual sensor architecture makes the tracking of the designs 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.

VI. Acknowledgement

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.

References

[1] W. Asmuth et al., *Feldversuch Autobahntechnologien A555: Ergebnisse und Vorschläge*, TÜV Rheinland, Proj. nr. 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, 1(2), 1984, pp.169–193.

[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*, accepted for High Performance Computing & Networking Europe 1997, Vienna. Will be published in Springer Lecture Notes on Computer Science.

[5] J. Lagerberg (ed.), *Organization and design of autonomous systems*, Faculty of Mathematics, Computer Science, Physics and Astronomy (WINS), University of Amsterdam, August 1996.

[6] S. M. Ross, *A course in Simulation*, MacMillan Publishing Company, 1990.