High performance discrete event simulations to evaluate complex industrial systems, the case of automatic

Hoekstra, A.G.; Dorst, L.; Bergman, M.; Lagerberg, J.; Visser, A.; Yakali, H.; Groen, F.; Hertzberger, L.O.

Published in:
Lecture Notes in Computer Science

DOI:
10.1007/BFb0031578

Link to publication

High Performance Discrete Event Simulations to evaluate Complex Industrial Systems

The case 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 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. Motivated by this work we introduce the concept of High Performance Discrete Event Simulations to evaluate the performance of Complex Industrial Systems. We assume that the elements have well-known behaviour, and demand that the interaction between the elements is well-known and their state changes at pre-definable time intervals. As a result we can model the *total* complex system as a discrete event model. We present the case of Automatic Debiting Systems for Electronic Toll Collection and introduce the modelling approach, the simulation environment (ADS-SIM), and discuss the need for HPC to carry out such simulations.

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. As a result of this specific work we are inspired to propose a general framework for evaluation studies of industrial installations, based on the concept of complex systems. This paper describes both our work in the evaluation project and our ideas towards a generalising framework.

The concept of complexity and complex systems as a general paradigm to study a wide variety of natural phenomena is getting more and more attention. A good illustration of this are e.g. the series “Lectures in Complex Systems” published by the Santa Fe Institute, or the research program BioComplexity, which is currently being set up in the Netherlands. Fox et al. define a complex system as a large collection of, in general, disparate elements which share a set of mutual dynamic connections [1,2]. Sloot et al. add to this definition the property that the elements have non-linear interactions, resulting in emerging macroscopic behaviour which, in general, cannot be predicted from the individual elements and their interaction [3].

As was noted by Fox et al. [2] and by Sloot [3,4] one can view a system, a model of that system, a computer simulation of that model, and the computer on which that simulation is executed as a complex system. The process of (parallel) computation can therefore be viewed as a formal mapping of complex systems on each other. By studying the space-time structure of complex systems, Fox et al. have defined general classes of applications and general classes of parallel computations (or computers), and in doing so, were able to draw conclusions with respect to the ability of parallel computing to solve realistic problem classes [2]. Of course, these general notions were based on a large amount of actual parallel computations solving real problems.

In this paper we follow the ideas of Fox et al. and Sloot et al., and define the class of *Complex Industrial Systems*. We map this class on the class of Discrete Event Models. The evaluation of such models requires *High Performance Discrete Event Simulations*.

2. Complex Industrial Systems

Industrial installations are, in many cases, constructed by combining a (large) number of sub-systems into the total system such that it will operate according to its specifications. Typical examples are e.g. chemical plants, surveillance and control systems (for e.g. air traffic), autonomous mobile robots, parallel or distributed computers, and so on. Each of these installations can be modelled as a complex system. The elements are the sub-systems which make up the total system. The connections between the elements are dictated by physical or logical links between the sub-systems. The emerging macroscopic behaviour in this case is the overall operation of the system according to the specifications and possibly unexpected behaviour of the total system, resulting in a faulty operation.

One can argue that the elements themselves are also complex systems. In the example of a distributed computer, an element of the total complex system would be a single computer which by itself is also a complex system. Fox et al. refer to such complex systems as compound complex systems [2]. This definition has an element of hierarchy in it. The compound complex system representation depends on the granularity (resolution) with which the system is viewed. In other words, what is defined as the element of the complex system depends on the resolution of the model.

In our definition a Complex Industrial System is a compound complex system representation of an industrial installation, where the elements are formed by sub-systems which compose the total installation. The elements are defined by the desired level of detail of the representation, i.e. the resolution with which we want to model the overall industrial installation. The connection between the elements are in general asynchronous.

In our evaluation study of Electronic Tolling we demanded that the elements of the Complex Industrial System all have well-defined state vectors which change, due to external inputs or due to the connections with other members, on pre-definable time stamps. This is an important requirement because in this way the modelling concentrates on the total system design and on coarse grained modelling. We believe however that this property can characterise a wide class of Complex Industrial Systems.

This property allows to map the Complex Industrial System on discrete event models. Simulation of the behaviour of a Complex Industrial System is now possible by developing a discrete event model and implementing it in the form of a discrete event simulation.

The discrete event model contains models for each element of the overall Complex Industrial System. This model is typically a response function which depends on external input parameters and inputs via connections with other elements. Usually, the behaviour of the sub-systems, i.e. the elements of the Complex Industrial System, are well-known, even under different input conditions. Therefore, the necessary models for the elements themselves are available or can be extracted from existing experimental and theoretical data, or from more detailed simulations on the separate sub-systems.

A good illustration of this point comes from an application in the field of simulations of parallel computer architecture [6,7], in particular the evaluation of novel networks. In this case the behaviour of the processors is modelled by a very limited set of parameters. These parameters however were obtained from more detailed model studies on the processor level.

It is vital to this 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].

3. Evaluation of Automatic Debiting Systems

3.1 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.

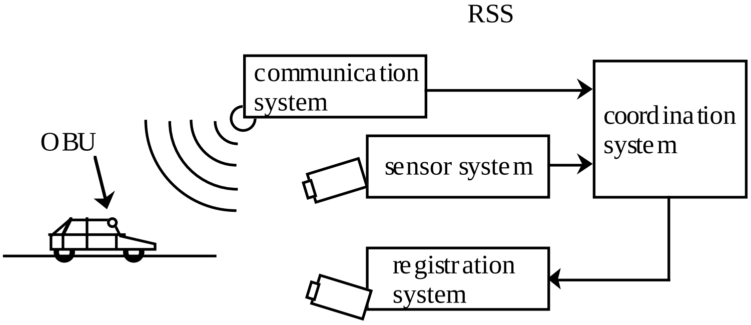

Figure 1: A schematic drawing of an Automatic Debiting System. OBU stands for On Board Unit; RSS is the Road Side System.

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 was 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 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.

3.2 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 subsystems. 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 as a Complex Industrial System. This means that we hierarchically decompose an ADS into sub-systems which by themselves can be complex systems. This hierarchical decomposition is a first important step for modelling an ADS. The models are designed to fit in the discrete event paradigm.

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 model of an ADS.

As an example 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.

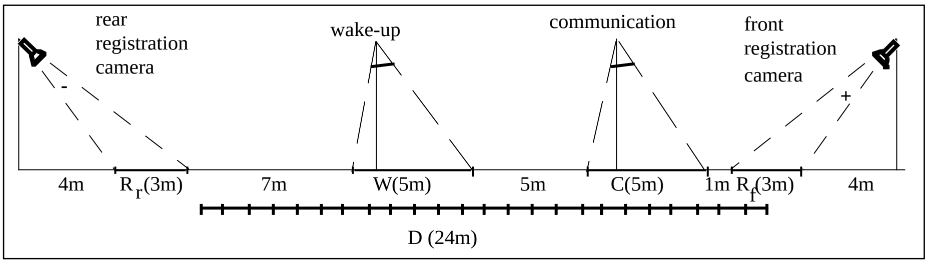

Figure 2: The fictitious ADS

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 3.3.

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 be should pointed out here that, especially in modelling of the sensor system, it is possible that after the decomposition of the ADS the resulting elements of the Complex Industrial System 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 [9] and later Weller, Groen and Hertzberger refined it to virtual sensors [10]. 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

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

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.

3.3 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 function of time) to a file, which is read by the Framework. The Framework is based on the evaluator tool developed by CMG [11]. 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.

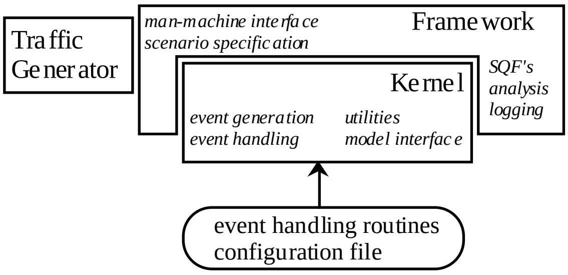

Figure 3: ADS-SIM design

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

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

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.

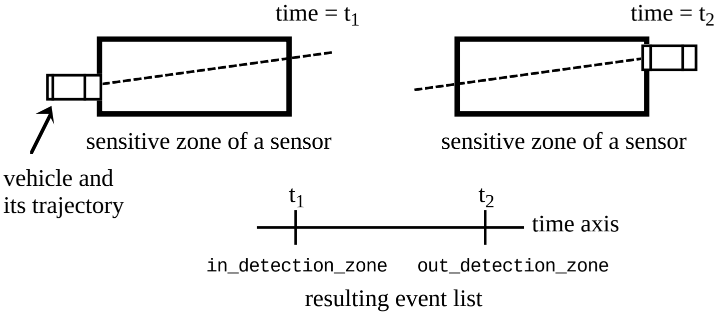

Figure 4: Automatic scheduling, by ADS-SIM, of the basic in_ and out_ events.

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.

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. For further information about ADS-SIM, please contact the authors.

4. High Performance Computing

Recently, Sloot coined the term High Performance Simulation, where large scale simulations of complex systems are performed on state-of-the-art HPC platforms, such as monolithic MPP's or heterogeneous distributed systems. [8] The approach towards High Performance Simulation should be to "take a holistic view incorporating all different levels in the mapping of application architecture to machine architecture" [8]. We will discuss to what extend the High Performance Simulation approach can help in setting up simulations of Complex Industrial Systems, resulting in the concept of High Performance Discrete Event Simulation.

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 the example of ADS simulations we have to track a few millions vehicles through the simulator. Given the execution times as reported in section 3.3, this shows that real production runs need a powerful HPC environment. Large scale simulations of Complex Industrial Systems 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 workstation 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.

Fox et al. [2] pointed out that discrete event models fall in the class of asynchronous models which are very hard to parallelise. Highly sophisticated techniques like parallel time warp mechanisms are required [2,12]. Although such techniques offer interesting possibilities for massive parallelism in ADS simulations (e.g. parallelising over vehicle tracks, over lanes in a multi lane road, or over sensitive zones in the ADS) we currently do not see the need for such massive parallelism in ADS simulations. However, in the general context of Complex Industrial Systems these techniques can have great value.

5. Discussion & Conclusions

Within the framework of formalising computing in terms of mapping of complex systems, we have defined the class of Complex Industrial Systems as being a model of an industrial installation. Furthermore, we introduced a discrete event property, which immediately places the Complex Industrial System in the class of discrete event models. As was suggested by our work on modelling and simulation of ADS designs, the discrete event property is valid as long as the number of hierarchical decompostions remains small. In other words, the modelling is done on a high level and the granularity of the models must remain relatively coarse. Using these general insights 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 be able to assess the overall strength of this simulation approach.

An interesting question is if the discrete event property is too strict, or if some elements of the Complex Industrial Systems should be represented by discrete time models, resulting in a hybrid discrete-time-discrete-event model of the total industrial installation. Our current experience in modelling ADS designs suggests that the discrete event model is suitable, although it raises subtle issues in e.g. defining and modelling virtual sensors (data not shown). In future work we plan to investigate this question in more detail.

We have studied one example of a Complex Industrial System and have thus identified a number of connections between on the one hand modelling industrial installations and on the other hand HPCN. In the future we will study other cases and investigate to what extend the general concept of Complex Industrial Systems is of value for evaluation studies of industrial installations using High Performance Discrete Event simulation techniques.

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. G. Fox et al., *Solving Problems on Concurrent Processors, volume I*, Prentince-Hall International, 1988.
2. G.C. Fox, R.D. Williams, P.C. Messina, *Parallel Computing Works !*, Morgan Kaufmann Publishers, 1994.
3. P.M.A. Sloot and J.A. Kaandorp and A. Schoneveld, *Dynamic Complex Systems (DCS): A New Approach to Parallel Computing in Computational Physics*, Technical Report, University of Amsterdam, TR-CS-95-08, 1995.
4. P.M.A. Sloot, “Modelling for parallel simulation: Possibilities and pitfalls, invited lecture”, in F. Breitenecker and I. Husinsky Eds., Eurosim '95, Simulation Congress, Elsevier Science Publishers, Amsterdam, the Netherlands, 29-44, 1995
5. P.L. Knepell and D.C. Arangno, *Simulation Validation, A Confidence Assessment Methodology*, IEEE Computer Society Press, 1993.
6. A.D. Pimentel, J. van Brummen, T. Papathanassiadis, P.M.A. Sloot and L.O. Hertzberger, “Mermaid: Modelling and Evaluation Research in Mimd ArchItecture Design," in L.O. Hertzberger, G. Serazzi (eds.), Proceedings of the High Performance Computing and Networking conference, p. 335-340, 1995.
7. A.D. Pimentel and L.O. Hertzberger, *The Mermaid Architecture-workbench for Multicomputers*, Technical Report, University of Amsterdam, TR-CS-96-04, 1996
8. P.M.A. Sloot, “High performance simulation,” EUROSIM - Simulation News Europe, vol. 1, p. 15-18, 1995.
9. T.C. Henderson E.Silcrat, “Logical sensor systems”, Journal of Robotics Systems 1, p. 169-193, 1984.
10. 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, Springer Verlag, p. 351-364, 1990.
11. 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, October 14-18, 1996.
12. B.J. Overeinder and P.M.A. Sloot, “Application of time warp to parallel simulations with asynchronous cellular automata”, in *European Simulation Symposium 1993,*(A. Verbraeck and E.J.H. Kerckhoffs, eds., Delft, The Netherlands, , Society for Computer Simulation International, p. 397-402, 1993.