Chapter X

A General Modelling and Simulation System for Sustainability Impact Assessment in the Field of Traffic and Logistics

Lorenz M. Hilty
Swiss Federal Laboratories for Materials
Testing and Research, Switzerland

Ruth Meyer
University of Hamburg, Germany

Thomas F. Ruddy
Solothurn University of Applied Sciences Northwestern Switzerland

INTRODUCTION

Traffic comprises a large and persistently growing share of resource consumption and environmental stress in modern economies. Even on our way towards an Information and Knowledge Society, the demand for physical transport has not let up. Although the total physical mass transported is no longer increasing in modern economies, the distances and exchange frequencies still are, both in freight and in passenger traffic. That is making traffic with its effects on the environment into one of the most difficult problems that has to be solved if we want to attain sustainable development.

Advanced modelling and simulation techniques can, among other computer-based approaches, make an important contribution both to environmental management in the private sector (Hilty, 1995; 1996; 2000) and to sustainable urban management in the public sector (Hilty and Weiland, 1994; Weiland and Hilty, 1998). Modelling and simulation techniques are especially useful when the decision makers want to account for the dynamics of traffic development and management.

In this chapter, we describe a modelling and simulation system that models measures concerning traffic both in the public and in the private sectors, and supports assessing their ecological, economic and social consequences. The system makes it possible to integrate models on various scales (micro, meso, macro) and to represent the complex behaviour of the actors involved.

The system was developed in the three-year research project Mobile (Model Base for an Integrative view of Logistics and Environment), funded by the *Volkswagen-Stiftung*, Germany, and jointly carried out by the University of Hamburg and the Research Institute for Applied Knowledge Processing (FAW) in Ulm.

FROM ENVIRONMENTAL IMPACT ASSESSMENT (EIA) TO SUSTAINABILITY IMPACT ASSESSMENT

In the Nineties a wide spectrum of software was developed to support environmental impact assessment (EIA). There were, for instance, new database systems for the management of environmental data (e.g. EQUEL, Plank, 1994), specific applications of geographic information systems (e.g. SAMBA, Boman et al. 1994), or expert systems for ecological (e)valuation (e.g. EXCEPT, Weiland et al. 1994).

The Mobile System was also designed as an EIA tool, specialized to the traffic and logistics domain (Hilty and Meyer, 1996). During the course of the MOBILE project, though, it became apparent that such a politically controversial topic as traffic should be modelled under all three dimensions of sustainability (the ecological, economic and social dimensions). Thus, right during the development of the system, the goal was changed from environmental impact assessment to the more general *sustainability* impact assessment. Therefore, the main component of the system evolved to an object-oriented class library that allows the user to model large populations of market participants (such as customers, suppliers, all kinds of traffic participants) with their preferences, spatiotemporal action patterns, plans, etc., and thus incorporated economic and social dimensions of the phenomenon. This was done by implementing the approach of individual-based simulation which originally comes from biological modelling. In the following sections, we first describe the system in its function as a tool for environmental impact assessment, and then we go into its expansion for individual-based simulation, for which a case study is given.

THE MOBILE MODELLING AND SIMULATION SYSTEM

The MOBILE system supports the user in modelling and simulating a class of systems which are characterized by coordinated transformations of objects in space and time (logistical systems in the broadest sense of the term). In contrast to conventional traffic simulation approaches, in this case the systems analysis and modelling are focussed on the goal of reducing resource consumption and environmental pollution caused by the transportation processes right from the start. This approach is also called *eco-logistics*.

During the early Nineties, the first author developed several simulation models for *specific* problems in this field, e.g. for the environmental impact of just-in-time delivery and production strategies (Hilty et al., 1994). The aim of the MOBILE project was to develop a *general* tool which supports the user in modelling and simulating traffic and transport systems in the conceptual framework of eco-logistics.

In order to provide maximum flexibility, the MOBILE system is designed to be an object-oriented "construction set" for models. In contrast to commercial software, there is no "one and only" transport or environment model embedded in the program. Rather, the user is encouraged to build his own model by selecting and composing submodels as building blocks from the model base. These building blocks cover a broad range of aggregation levels, perspectives, and problems in the field of transport and environment. The system combines simulation techniques with geographic information system (GIS) technology.

One of the most important features of the MOBILE system is the fact that the model base includes *complementary* as well as *competitive* models. According to Zeigler's theory of modelling and simulation, models "are complementary when they embody the same hypotheses, but represent them in different ways." They are "competitive when they embody different, mutually exclusive hypotheses about how the real system works." (Zeigler, 1984, p. 13). The coexistence of *complementary* models is useful, for example, when the same phenomenon (e.g. highway traffic) can be modelled at different aggregation levels. The user then has the possibility to test systematically if – within the specific experimental framework of his study – the more detailed model can be replaced by the more aggregated one without losing aspects of the system's behaviour that are essential for the specific question the study aims to answer. To compare *competitive* models is especially important in a field like transport modelling, where there are many different theories and views of the same phenomena. For example, there are at least 18 basic types of private transport assignment models, depending on the assumptions made

- about the speed-flow relationships of network links and nodes (capacity restraint can be ignored or modelled on a macroscopic or microscopic level),
- about the route choice behaviour of the drivers (deterministic or stochastic behaviour model, travel time and/or other aspects included in the cost function),
- about the system being in a stationary or transient state (static or dynamic assignment).

Note, however, that traffic *assignment* is not even the most controversial aspect of transport modelling by far. Much greater variety can be found in the field of traffic *generation* and, of course, *environmental impacts*.

In order to support the evolution of theories and models in the field of transport and the environment, the model base can be extended whenever the user wants to include a new model.

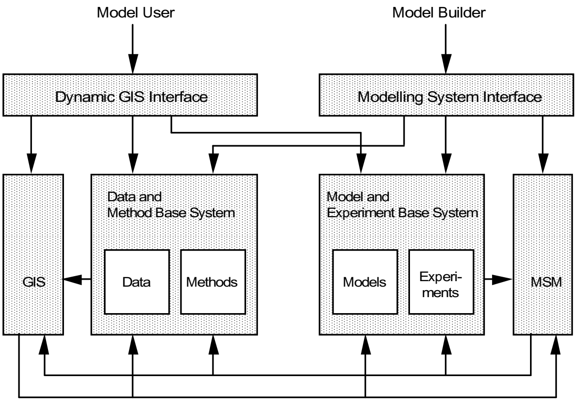
System Architecture

The architecture of the MOBILE system is sketched in Figure 1. This picture shows one possible configuration of the system. In fact the existing components of the MOBILE system are implemented as relatively independent services, which can be used and coupled with one another in different ways depending on the application.

The central part of the system as shown in Figure 1 is the *model and experiment base system*. The *model base* contains simulation models which can be combined into more complex models, the latter tailored to the particular problem the user wants to study.

The *experiment base* stores and organizes experiment classes as well as explicit specifications of the simulation experiments carried out previously. Experiment classes are used as templates enabling systematic variability of experiments. Experiment instances are persistent objects assuring reproducibility.

Figure 1: *The* Mobile *system architecture.*



The Mobile System treats simulation models as independent processes (simulators). Complex models as well as simulation experiments are implemented as hierarchically structured process groups which can be distributed all over the system.

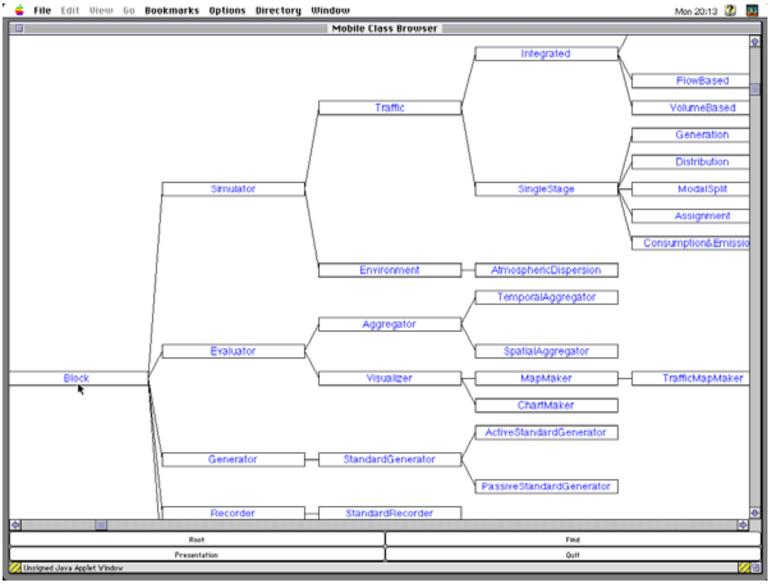
To increase the reusability of existing simulation models, the underlying source code *implementing* simulators is not limited to any one particular programming language. On the level of *specification*, however, models and experiments (as well as methods and the access to data) are described uniformly by means of the *Mobile Script Language (MSL)*. Such scripts are interpreted by the *Mobile Script Machine (MSM)*. The execution of an experiment corresponds to the interpretation of the according MSL specification. Hence the MSM plays a crucial role in the Mobile system.

The *model and experiment base system*, the *MSM*, and the *data and method base system* form the model builder's view of the Mobile system. This is emphasized by the *modelling system interface* which provides the modeler with graphical and textual specification editors, a class base browser (see Figure 2), which gives access to models, experiments, data and methods in order to combine existing models, develop new building blocks and validate them.

For users who are domain experts, but have little experience in modelling and simulation, the system provides the *model user's interface*. To the model user the Mobile system appears as a *Dynamic Geographic Information System (DGIS)*. Hence he is provided with a common GIS, enhanced by features for simulation and dynamic data processing. Thus an expandable open GIS also forms an important component of the Mobile system.

Mobile is designed as an open distributed system so that it can easily be expanded, e.g. by an additional GIS component or by multiple distributed model bases.

Figure 2: Screenshot of the M*OBILE class browser.*



The MOBILE Script Language

The *M*OBILE *Script Language* (MSL) is an object-oriented, high-level specification language for composing simulation models and experiments from existing building blocks (Mügge et al., 1997). It is not supposed to be just another simulation language. It is, in particular, not suitable for implementing the behaviour of a single model. Unlike general purpose or simulation languages, it only covers issues such as structure representation of complex models and experiments as well as interface definitions in order to facilitate component coupling. Therefore MSL can be characterized as a channel-based Module Interconnection Language (Rice and Seidman, 1994) focussing on simulation modules.

MSL forms the basis of model and experiment specification in the MOBILE system. It enables the model user to deal with specifications on a highly semantic level. He does not need to know about implementational details in order to choose the right model or to couple models. It does not even matter which programming language a particular model is written in. The only form of representations he has to handle is MSL specifications.

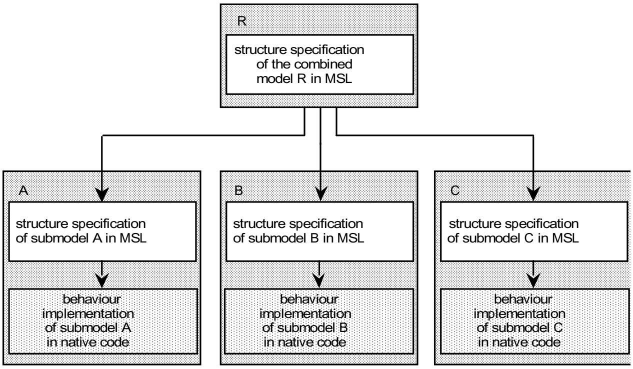
This is achieved by a strict separation of model specifications into structure and behaviour defining parts. MSL is above all a structure specification language. Hence model structure is completely specified in MSL. The behaviour of each individual building block (atomic model) is specified in another language, e.g. C^{++} or Java, which we will denote as the *native* language further on.

To join these specification parts together, the interface of each building block, which is at least implicitly specified in the native language, is explicitly given in MSL.

Figure 3 illustrates the connection between MSL and native language parts in an abstract manner. It depicts a model (*R*) consisting of three submodels (*A*, *B* and *C*). The corresponding representation in the model base consists of four MSL parts and three parts of native code. The MSL parts represent *interface and structure* of the combined model and the three submodels, respectively. The three native code parts form the behaviour specifications of the individual submodels.

The double interface specification for atomic simulators (the leaves of the model tree) does not only technically enable the system to deal with the model behaviour implementations. It also bridges the gap between the *computer-oriented implementation level* of data structures in native code and the *problem-oriented semantic level* of MSL specifications.

Figure 3: Specification of a combined model with MSL and native code.



Below we present the main characteristics of MSL in order to give a general idea of the language. First we focus on its object orientation, which shows up clearly in the following features:

- An MSL specification of a model or experiment always defines a *class,* and not an *instance*. Only experiments can be instantiated explicitly in order to specify an executable simulation experiment.
- All components (the building blocks used) of an MSL specification are (at least virtually) *independent processes* communicating via *message passing*, so that they can easily be distributed.
- Communication between components of an MSL specification strictly follows the *client/server paradigm* and each component can appear as both a server and a client relative to its different neighbours in the data flow of one experiment.

The components are *encapsulated* in the sense that they communicate only according to their interface and topology definition.

Inheritance is used for model specifications in order to enhance reusability and to facilitate navigation in the model base.

Another important characteristic of MSL is its high abstraction level, as indicated below:

- the MSL type concept is based on *mathematical constructs*, e.g. sets and functions;
- direct use of data structures on the implementational level is avoided, since MSL represents data on a *semantic level*. The MOBILE Script Machine (MSM) transforms MSL data types into native language data structures;
- MSL enables the modeler to represent *metadata* about data objects, models and experiments in a structured way. Examples for model metadata are physical dimensions and units or validity restrictions. This metadata can be used by the MOBILE system to automatize unit and data format conversions as well as plausibility checks for model coupling.

Two versions of MSL are defined, a textual and a graphical version. In this section we focus on textual MSL. An example of the graphical representation is given in the next section.

The following experiment specification in MSL serves as an example. A simulation experiment is regarded as a data flow network. A simple experiment consists of one or more simulation models (called *simulators* because of their local simulation control) fed with input data from active data sources (called *generators*), followed by the evaluation of the simulation results with the help of analysis or visualization methods (called *evaluators*). Finally, the data flow ends in report generators (called *recorder*s) that store and/or present the results in a readable form. (This corresponds to the four subclasses of blocks as can be seen in Figure 2.) The structure of this data flow is defined in a *MSL experiment class* specification. In an executable *MSL experiment instance* specification, this class is provided with instantiation parameters.

In Figure 4 we present a condensed example of an experiment class RoadTrafficEmission in MSL source code. The specification of an MSL experiment class is divided into three parts: in the parameters part, the experiment interface is given in terms of the names and types of its parameters; the following contains part declares the components of the experiment and a concluding connects part defines the topology of its data flow network.

Figure 4: Example of a MSL experiment class specification.

```
experiment class RoadTrafficEmission specializes Experiment
    parameters
        trafficDemand : DayProfile;
        roadNetwork : RoadNetworkTopology;
        geometry : RoadNetworkGeometry;
        /* further parameters omitted */ end;
    contains
        generators
            TrafficDemand of class EquidistantTimeSeries:
                (dataObject, deltaTime; ) -> (data)
                {    dataObject:=    this.trafficDemand;
                     deltaTime      :=          1 [h]; };    end;
        simulators
            AssignmentModel of class TrafficAssignment :
                (roadNetwork; demand) -> (loads, velocities)
                {    roadNetwork    := this.roadNetwork; };
            EmissionModel of class VehicleEmission :
                ( ; trafficFlow, velocities) -> (emissionRates)
                /* body omitted */    end;
        evaluators
            EmissionVisualization of class Visualization :
                (geometry; rates) -> (map)
                /* body and further evaluators omitted */ end;
        recorders
            EmissionMaps; /* further recorders omitted */ end;
    end;
    connects
        TrafficDemand.data to AssignmentModel.demand;
        AssignmentModel.loads to EmissionModel.trafficFlow;
        AssignmentModel.velocities to EmissionModel.velocities;
        EmissionModel.emissionRates to EmissionVisualization.rates;
        EmissionVisualization.map to EmissionMaps;
        /* further connections omitted */ end;
end.
```

The contains part is subdivided in sections depicting the different types of components, such as data sources (generators), models (simulators), transformation methods (evaluators), sub-experiments (experiments) and data sinks (recorders). In addition to the instance and class names, the interface of each component (parameters, inputs and outputs) is declared and its parameters are connected to experiment parameters or provided with data objects (values, e.g. deltaTime := 1 [h]). The interface declaration would be superfluous for reports, since their interface is completely determined by the preceding objects in the data flow.

The class *RoadTrafficEmission* defines a simulation experiment which assigns traffic demand to the given road network, calculates the emissions and visualizes them in terms of maps. The traffic demand (parameter *trafficDemand*) is passed on to the source *TrafficDemand* which provides the assignment model with one data item for each hour of simulation time. The assignment model then computes traffic flow and speed from the input *demand*, passing this data to the emission model. The emission model provides emission rates which are dependent on the composition of the traffic and the average speed of the current flow. Finally, the results are transformed into thematic maps by the visualization method *EmissionVisualization*.

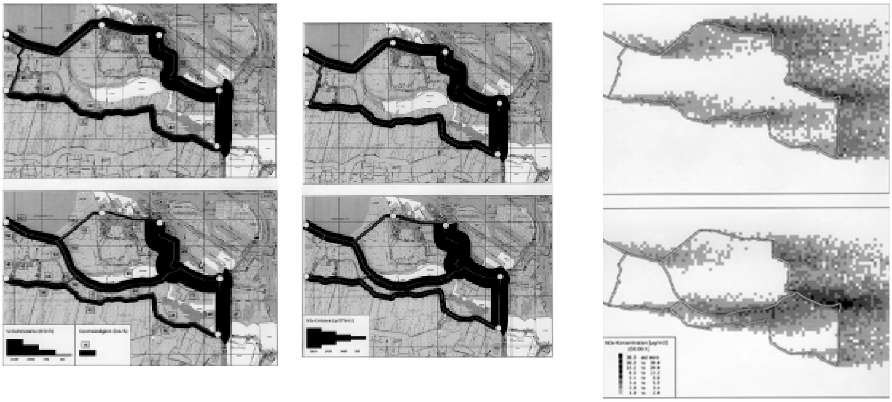
MSL is capable of specifying much more complex experiments than we can present in this chapter. For instance it offers the definition of higher order experiments which are comprised of subexperiments. This enables the modeler to compare different scenarios or competitive models within one experiment specification. This is also the case for models which can be built up of any kind of submodel, so that arbitrary hierarchical networks of models and data processing methods can be specified.

Application Example

In order to illustrate the system's features introduced so far, we want to show an example of the system's application. The objective of this study was to assess whether or not a planned bypass would reduce the traffic load and pollution in Finkenwerder, a region in the south of Hamburg which is heavily affected by growing commuter traffic. Part of this region is a nature preserve.

Figure 5 shows the results for two scenarios, the status quo (top) and the situation after the bypass would be built *and* the road in the north (presently used as a thoroughfare) would be closed (bottom). The two maps in the left column show the traffic flow, differentiated by the two driving directions. The comparison shows that

Figure 5: Some simulation results visualized as thematic maps. For explanation see text.



the situation in the northeast would become even worse. This can be explained by the fact that some inhabitants of Finkenwerder would then be now forced to go a roundabout way. The middle column of Figure 5 shows how the different traffic flow and traffic conditions affect the emission of nitrogen oxides (NO_x) by the vehicles. Finally, we show in the right column the NO_x concentrations calculated by an atmospheric dispersion model using the Lagrange particle simulation approach, taking a typical wind field and the emissions of the middle column as input.

All these results are shown here for illustration only. For a reliable interpretation, it would be necessary to include and discuss many more simulation data and information about the models in detail, their basic assumptions and the actual parameter values chosen.

Furthermore, note that the maps in Figure 5 only show a snapshot at 8:00 a.m. when commuter traffic flow is just past its maximum. However, we simulated the temporal behaviour of traffic flow and pollution over the course of a whole workday, because only that way is it possible to adequately represent traffic peaks, the resulting jams and the emission levels influenced by the changed driving pattern. Models based on daily mean values of traffic loads are seldom useful for environmental investigations. In order to illustrate the dynamics, we show in Figure 6 the simulated traffic flows in hourly intervals (from 6 a.m. until 11 p.m.).

The simulation experiment executed to produce these maps for one scenario can be specified in graphical MSL. The user can create simulation experiments by using the graphical MSL editor of the MOBILE system. Figure 7 gives an impression of this part of the user interface, which was implemented on Apple Macintosh and generates textual MSL code. The latter is interpreted by the MSM, usually on a Unix platform. The MSM is implemented in Java.

The experiment we used as an application example in this section is a relatively simple case and has been discussed here only for the purpose of illustration. In fact, the Finkenwerder case study involved more complex simulation experiments with two competitive assignment models (which turned out to produce the same results in this case) and emission models for different pollutants (see Hilty, Meyer, Page and Deecke, 1997, for more detail).

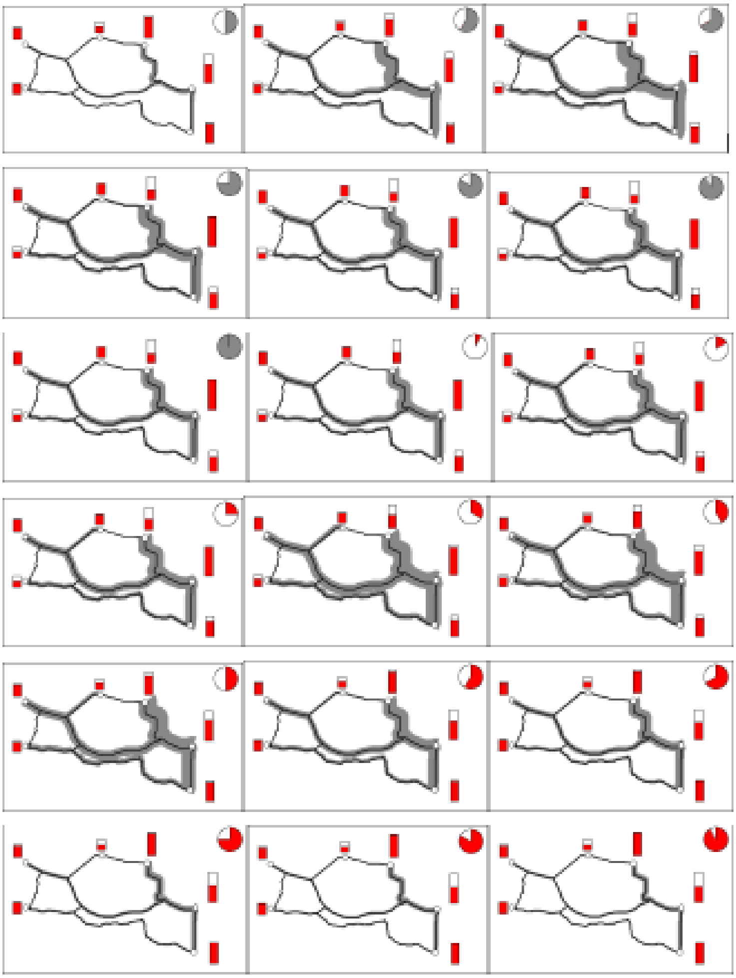
The MOBILE Class Library for Individual-Based Traffic and Logistics Modelling

The class library developed for the individual-based modelling of traffic and logistics was divided into different interrelated packages. Figure 8 shows an overview of the individual packages and their interdependence. We use the UML notation (Fowler 1998).

The packages Core, Geometry and Container contain the implementation of the data types that are needed in this area. They are independent from the modeled real system and the concrete problem. Each data type is implemented as a Java class; one instance of such a class represents exactly one value of the respective type.

The package Pollution comprises classes that can be used to model the environmental impact of traffic in particular. Data types often needed here are concentrations, emission rates and emission factors. Further classes serve to represent point, line or surface-shaped sources of emissions, for example.

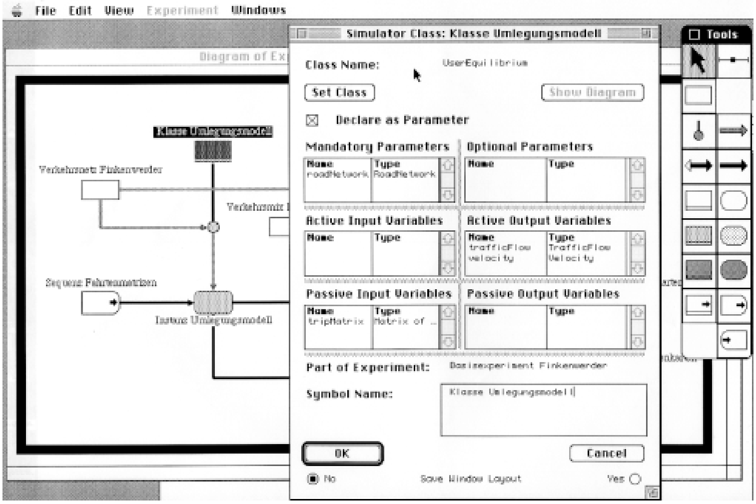
Figure 6: Simulated traffic flow from 6 a.m. until 11 p.m.



The package TrafficSimulation contains classes specially tailored for traffic and logistics simulation. The starting point for modelling individuals is the class DestinedObject, which serves to represent moveable objects with places of destination (targets). One instance of this class represents an individual that knows its current position and the point in time at which it is at this place. It also has a list of targets towards which the future movement of the individual is aimed. The class DestinedObject is consciously kept general: The manner of movement — independent or directed and using whichever means of transport— is explicitly not predetermined. This class can therefore be used as a basis for a wide spectrum of moveable objects, ranging from living individuals (e.g. couriers) to inanimate objects (e.g. parcels). By specializing the class DestinedObject, individuals with arbitrarily complex behavior can be created.

Classes for means of transport and traffic networks are available to model the traffic infrastructure. The class TrafficNetwork is a specialization of Geometry.Graph

Figure 7: Specifying a simulation experiment using the graphical MSL editor.



Class Name: UserEquilibrium

☒ Declare as Parameter

Mandatory Parameters

Name	Type
roadNetwork	RoadNetwork

Optional Parameters

Name	Type

Active Input Variables

Name	Type

Active Output Variables

Name	Type
trafficFlow	TrafficFlow
velocity	Velocity

Passive Input Variables

Name	Type
tripMatrix	Matrix of ...

Passive Output Variables

Name	Type

Part of Experiment: Basisexperiment Finkenwerder

Symbol Name: Klasse Umlegungsmodell

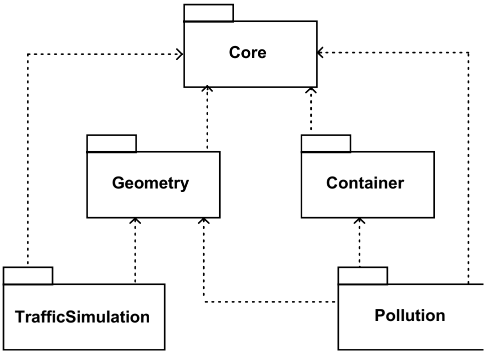
● No Save Window Layout Yes ○

(a directed graph) with more specific nodes and edges. It offers the functionality of determining the closest traffic node to any geographical point and finding the shortest or fastest path between two nodes. A further specialization to the class RoadNetwork offers additional possibilities to represent daytime-dependent background traffic, capacity functions and traffic resistance parameters.

INDIVIDUAL-BASED SIMULATION OF A LARGE CITY COURIER SERVICE: A CASE STUDY IN SUSTAINABILITY IMPACT ASSESSMENT

Courier and express services form a market sector of urban freight traffic which is growing rapidly. It is typical in cities such as Hamburg to have several thousand bicycle and car couriers in operation, covering more than 100,000 kilometers every day. Using the individual-based and object-oriented modelling approach of the MOBILE system, we developed a model which simulates a courier service in detail. This includes accepting the orders, passing them on to the couriers by radio, as well as the movement of each courier in the road network. It was worked out in cooperation with

Figure 8: The packages of the class library in UML notation. Dashed arrows symbolize dependencies.



a large courier service in Hamburg. The purpose of the simulation study is to assess the effects of possible changes to the courier service system (scenarios) and to evaluate them using economic, social, and ecological criteria.

Within the scope of this chapter, we only can present a simple type of scenario, namely changing the size of the fleet, that is, the number of couriers available for deployment at one time. This relatively simple step already shows, however, that an individual-based model creates results an aggregated model could not deliver, for instance, the distribution of the couriers' workload.

Modelling the City Courier Service

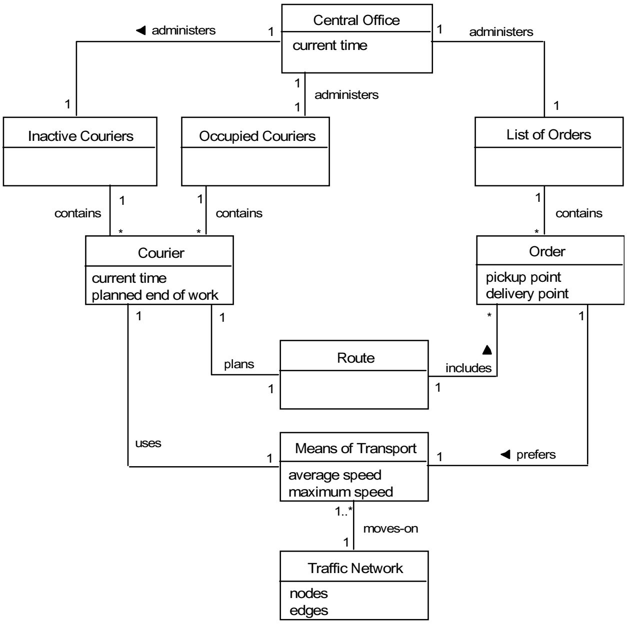
The objective of this case study was to estimate the effects of different measures on the following three targets:

- social goal: fair distribution of orders, high level of income for couriers;
- economic goal: competitiveness through high quality service (reliability, speed);
- ecological goal: less motorized mileage (higher percentage of bicycles, optimal routes).

The structure of a city courier service was formalized by means of Larman's object-oriented analysis (Larman, 1998). The most important classes and associations of the conceptual model are shown in Figure 9.

A city courier service consists of the central office in a fixed location and the courier fleet moving in the city area. It is the job of the central office to accept the orders (often several per minute) and to pass them on to the couriers who are in continuous radio contact with the office. A courier who has taken on an order collects the consignment from the sender, also collecting the carriage charges there, and then quickly delivers the consignment to the consignee.

Figure 9: Basic structure of the conceptual model of the city courier service in UML notation.



Because the couriers work on their own accounts, they compete for orders. A decisive element for organizing courier services is therefore a procedure to allocate the orders as fairly as possible to the couriers who are ready for deployment. Our system analysis of a large courier service showed that this procedure is based on unwritten rules that are accepted and adhered to by all parties concerned. They involve, for example, the preferential treatment of drivers who have registered availability (inactive at present) or the allocation of unpopular (not lucrative) orders by the office. This social goal of fairness should be harmonized as well as possible with the economic goal of holding one's own as a courier service in a highly competitive market. Above all, the quality of service (reliability and speed) is crucial.

The potential conflict between social and economic goals is finely balanced by the set of rules described above which has arisen from years of practice. With increasing competitive pressure, however, courier services are considering changes to their logistical organization. In order to estimate the effects of possible changes we have developed a simulation model, in cooperation with the largest courier service in Hamburg. This model adopts the existing procedure for allocating and fulfilling orders in great detail. The simulated scenarios were processed using real order files from the courier service. In addition to social and economic aspects, we also considered the ecological goal of reducing motorized transport. This is possible with courier services, as part of the courier fleet consists of bicycle couriers and logistical concepts are imaginable which utilize the specific advantages of bicycles in city traffic.

Figure 10: The Hamburg traffic network with approx. 45,000 edges that was used in the simulation study (vector map).

Each courier is modeled as an individual who possesses a fixed means of transport (car or bicycle) and who has access to the functionality of the traffic network for planning routes. For this purpose, a subclass of DestinedObject specific for the application was derived in which the complex behavior of couriers is implemented:

- One courier can work on several orders in parallel.
- Couriers try to plan their route optimally between the pickup and delivery points of all orders they take on.
- The acceptance of an order depends on an assessment function that combines the current need for orders and the attractiveness of the order.

The couriers move within the Hamburg traffic network which is represented as a directed graph with around 45,000 edges for motorists and 47,000 edges for cyclists (see Figure 10). Not only have cyclists more connections available, they are also practically uninfluenced in their speed by the traffic flow on the routes. The addresses of the pickup and delivery points in the order data of around 10,000 orders were geocoded and assigned to the respective closest nodes in the traffic network.

As an example of the application of the model we will show how the target Figures change as a function of the number of couriers authorized at one time. A special aspect of the courier service we investigated is that the central office limits the size of the fleet. Other courier service offices are usually interested in employing as many couriers as possible because they pay a lump sum monthly to the central office. In our case, the couriers themselves are partners in the office and can therefore take part in determining the size of the fleet. It is obvious that the quality of the service is best when there is a very large number of couriers, but the income of the couriers then shrinks because the volume of orders is distributed among more drivers. The number of couriers also influences the way in which the mentioned rules of behavior to ensure fair distribution of orders take effect in concrete terms.

Figure 11 shows how the gross hourly wage of the couriers, the motorized mileage and the accumulated delay time change when the number of couriers as regards the status quo (94) is raised or lowered in steps of 10 %. As there are normally no promised deadlines in the courier business (everything is dealt with ”as quickly as possible”), the *ideal processing time* was chosen as reference quantity for the delay time. This is the processing time which would result if a courier had already been at the pickup point when the order was taken and had then transported the consignment as quickly as possible to the point of delivery. The delay time should therefore not be understood here as the delay perceived by the client, but as the time which is needed for the journey to the pickup point and also for accepted detours when processing several orders in parallel.

As was to be expected, delays increase more and more with a reduction of the fleet. The delay time also initially increases with a rise in the number of couriers. This

Figure 11: Effects of varying the number of couriers on the most important target figures (simulation result).

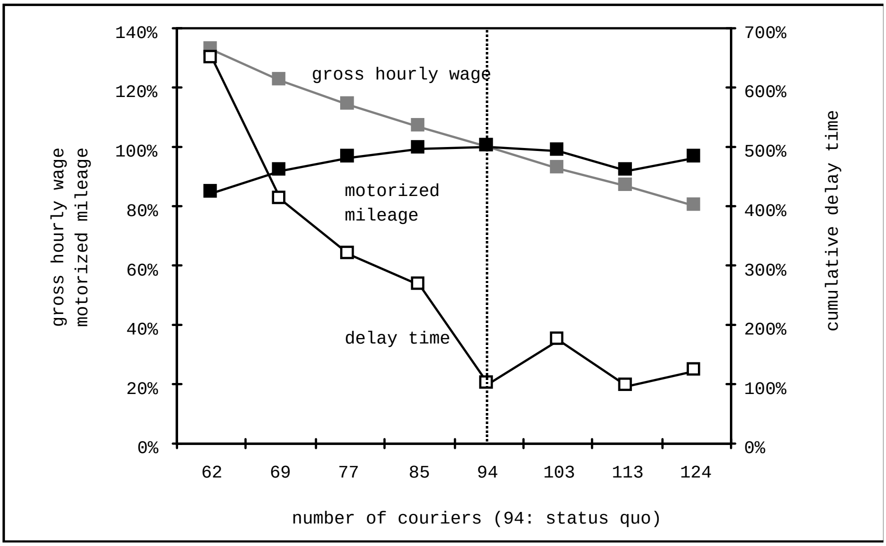
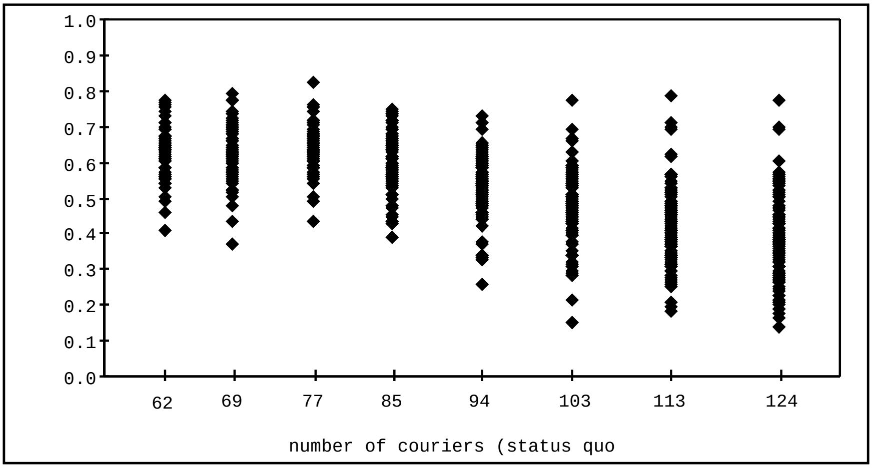


Figure 12: Effects of varying the number of couriers on the distribution of workload (simulation result).



can be explained by the fact that couriers then are more often unoccupied and register their availability, which in turn leads to preferential treatment, preventing a logistically optimal distribution of orders. This result suggests that the fleet size of about 94 couriers established in practice is well chosen. The gross hourly wage falls as expected with rising fleet size. Due to the given volume of orders, the couriers are on average utilized worse (and have a correspondingly lower income) the more numerous they are. Considering aspects of fairness, it is also interesting to see how intensively the workload is spread with different fleet sizes. Figure 6 shows the distribution of workloads for each fleet size. The workload is the proportion of travelling time to the couriers' working time. A courier would have a workload of 1.0 if (s)he would be on the move for the whole working period without waiting time.

To begin with, it can be stated that, even with a quite small number of couriers, hardly any of them have a workload over 0.8. By accepting a much worse quality of service, the workload theoretically could be raised further, but this case has no practical relevance. The effect seen in Figure 12 is interesting, where the drop in workload with growing fleet size affects, above all, the lower margin. In other words, there are more and more couriers who are utilized less than 30%, while fewer individuals still have above 70% deployment.

For the simulation of a working day (about 1000 orders) on a workstation (Sun Ultra Enterprise 450 with two 250 MHz processors) the model needs about 20 minutes computing time.

The case study showed that the individual-based modelling of a complex logistical system is very costly on the one hand. However, it brings results which are easier to interpret for the decision-maker than aggregated coefficients. For instance, the concrete distribution of the workload as shown in the example is much more demonstrative than stating mean value and variance of the workload – quantities that at best could be calculated with an aggregated model. Nor would a true-to-life representation of the order distribution process with its numerous special rules – which play an important role in practice – be possible at a higher level of aggregation.

FUTURE TRENDS

We are in the process of developing the approach presented here further by integrating concepts from the area of multiagent systems. The intelligent agents approach has the following advantages for complex modelling tasks within a framework for sustainability impact assessment:

- Intelligent agents allow for a rich representation of behaviour in space *and* time, which is difficult to deal with in traditional simulation methodology. However, the MOBILE system with its class library for individual-based simulation is already a step in this direction.
- Intelligent agents are a natural way to model market mechanisms. That provides a good basis for modelling the demand for goods or services, the satisfaction of which can then be investigated under sustainability aspects. The instruments of global environmental and development policy, to the extent that any such exist, increasingly rely today on market mechanisms (such as tradeable emission permits, e.g.).
- Agents also allow for the simulation of political processes, e.g. negotiations or the effects of new legal measures. Hence, multiagent systems could also become relevant as a general methodology of impact assessment for measures and instruments of global environmental and development policy.

This work is part of a more general framework in which the authors develop computer-based methods and tools that can contribute to a sustainable development. The whole spectrum of ideas is outlined in a recent article (Hilty and Ruddy, 2000).

It is typical for environmental informatics that new impulses for the development of methods and tools come from political trends in the application domain. We expect that the domain of *traffic and logistics* will be deeply affected by new requirements stemming from international efforts to curb global warming under the United Nations Framework Convention on Climate Change (UNFCCC) signed in Rio de Janeiro in 1992 and its subsequent Kyoto Protocol in 1997, and that this will stimulate new developments in environmental informatics.

These binding treaties of international law call for the reduction of greenhouse gas emissions, primarily carbon dioxide and methane, in all countries that are parties to the treaty. The European Union, for instance, has committed to reduce its emissions by an average of 8 percent between 2008 and 2012.

However, that will only be a tentative first step in the direction of the more definitive action that will be needed once the scientific basis can be more firmly established in the eyes of global warming skeptics, such as many in the American oil industry. In coming years, new forms of international cooperation must be found in order to overcome the policy paralysis described in gaming theory as “the prisoners’ dilemma”. Such theory explains the disincentive to any individual country daring to go ahead before others do with legislation involving the preservation of global public goods, such as an atmosphere with an inhabitable CO_2 concentration (Ruddy, 2000).

To be effective, the commitments of national governments to international protocols must, of course, be followed by domestic implementation measures such as ecological taxes. Such plans for an ecological tax on the CO_2 content of fuels have existed in the European Union since the early Nineties (Ruddy, 1991a). Here again, the phenomenon of policy paralysis described above set in, preventing individual parties from proceeding in the face of a lack of international cooperation. In this case, the EU did not dare go ahead with its plans without seeing comparable taxes passed the US and Japan, lest it jeopardize its global competitiveness as an industrial location. Germany has recently embarked on a course of making the burning of fossil fuels less attractive. Great Britain has also passed a new climate change levy, and even France has announced plans to have a climate related ecological tax soon.

As *road traffic* is a major contributor to CO_2 emissions, implementing the goals of international climate policy will thus apply additional pressure to the operators of cars and trucks to optimize trips or to avoid them altogether. For this reason, there is a trend to introduce a new generation of route planning software in combination with GPS and communications technologies. Switzerland, for instance, has introduced a road tax on truck kilometrage which makes it worthwhile now for truck operators to use software to plan their vehicles' routes dynamically. Dynamic route planning is more effective than static route planning because orders coming in can be matched continuously to the traffic situation. (Our case study about courier services is one example of dynamic route planning). A market study of a Swiss software company predicts that a new market for route planning software in Switzerland and neighbouring countries alone will grow to a few 100 million Euros.

When the price signals of ecological taxes are reflected in consumer behaviour, new technologies are expected to come into greater demand. Even hydrogen, once prohibitively expensive as compared with petrol and only applied for testing purposes (e.g. by Hamburg buses and Berlin vans with internal combustion engines (Ruddy, 1991b), is expected to be used more widely soon in fuel cells in conjunction with electric drive systems.

These forthcoming changes call for advanced methodologies and tools to support the decision makers in the public and private sectors who will be affected. We hope that environmental informatics can – by providing methods and tools for sustainability impact assessment – make a substantial contribution to implementing the vision of a sustainable information society.

CONCLUSION

Simulation modelling is an important instrument with which to face the challenge of planning in the domain of traffic and logistics, especially in a world that must approach sustainable development. It can be used as a basic approach to environmental impact assessment or – even more challenging – to sustainability impact assessment.

MOBILE is a flexible modelling and simulation system for both model builders and model users. It makes intensive use of object orientation for specification and management of complex models and experiments. The class library for individual-based simulation implemented in Java is a first step into the direction of an agent-based system. It supports the development of models accounting for all three dimensions of sustainability (ecological, economic and social).

The MOBILE system is designed for applications in traffic and logistics. In this domain, the international efforts to limit the emission of greenhouse gases are expected to create a market for sophisticated methods and tools, such as on-line dynamic route planning systems.

We hope that MOBILE contributes to a better understanding of complex traffic systems and supports decision making, in particular with respect to sustainability issues.

REFERENCES

Boman, J., Olsson, M. and Sundholm, M. (1994). SAMBA – A Way to Deal with GIS. In Guariso, G., Page, B. (eds.). *CSEIA 93, Proceedings of the IFIP TC5/WG5.11 Working Conference on Computer Support for Environmental Impact Assessment,* 1993, Como, Italy, IFIP Transactions B-16, pages 1-11. North-Holland, Amsterdam.

Fowler, M. (1998): *UML Distilled – Applying the Standard Object Modeling Language.* Addison-Wesley, Reading, Mass.

Hilty, L.M., Page, B. and Martinssen, D. (1994). Designing a Simulation Tool for the Environmental Assessment of Logistical Systems and Strategies. In Guariso, G., Page, B. (eds.). *CSEIA 93, Proceedings of the IFIP TC5/WG5.11 Working Conference on Computer Support for Environmental Impact Assessment,* 1993, Como, Italy, IFIP Transactions B-16, pages 187-198. North-Holland, Amsterdam.

Hilty, L. M. and Weiland, U. (1994): Sustainable Cities – Opportunities and Risks of Information Technology. In Brunnstein, K.; Raubold, E. (eds.). *Proceedings of the 13th World Computer Congress (IFIP World Congress).* Volume II: Applications and Impacts, pages 613-618. North-Holland, Amsterdam.

Hilty, L. M. (1995). Information Systems for Industrial Environmental Management. In Avouris, N., Page, B. (eds.). *Environmental Informatics – Methodology and Applications of Environmental Information Processing,* pages 371-384. Kluwer Academic Publishers, Dordrecht.

Hilty, L. M. (1996). Computer-Supported Environmental Management – Why Simulation Methods are Useful. In Denzer, R.; Schimak, G.; Russell, D. (eds.): *International Symposium on Environmental Software Systems (ISESS),* Malvern PA *(USA),* June 1995, pages 155-164. Chapman and Hall, New York.

Hilty, L.M. and Meyer, R. (1996): A flexible modelling and simulation system for environmental impact analysis in traffic planning. In *Proceedings of the 2nd International Conference on Urban Transport and the Environment for the 21st Century (UT 1996),* 2-4 October 1996, Barcelona, pages 221-230. Computational Mechanics Publications, Southhampton (UK).

Hilty, L. M.; Meyer, R.; Page, B.; Deecke, H. (1997): *Anwendung konkurrierender Verkehrsumlegungsmodelle zur umweltbezogenen Evaluation einer Umgehungsstraße.* Universität Hamburg (Berichte des Fachbereichs Informatik; FBI-HH-B-201)

Hilty, L. M. et al. (2000). *Strategische und betriebsübergreifende Anwendungen betrieblicher Umweltinformationssysteme.* Marburg: Metropolis.

Hilty, L. M. and Ruddy, T. F (2000). Towards a Sustainable Information Society. *INFORMATIK/INFORMATIQUE,* 4/2000: 2-7.

Larman, C. (1998). *Applying UML and Patterns – an Introduction to Object-Oriented Analysis and Design.* Upper Saddle River.

Mügge, H., Meyer, R., Hilty, L.M. and Page, B. (1997). Object-Oriented Specification of Models and Experiments in Traffic Simulation. In Swayne, D.; Denzer, R.; Schimak, G. (ed.): *2nd International Symposium on Environmental Software Systems (ISESS), Whistler (Canada),* pages 355-342. Chapman and Hall, New York.

Plank, T. (1994). EQUEL – A Database System for the Management of Environmental Data. In Guariso, G., Page, B. (ed.). *CSEIA 93, Proceedings of the IFIP TC5/WG5.11 Working*

Conference on Computer Support for Environmental Impact Assessment, 1993, Como, Italy, IFIP Transactions B-16, pages 61-73. North-Holland, Amsterdam.

Rice, M. D. and Seidman, S. B. (1994). A Formal Model for Module Interconnection Languages. *IEEE Transactions on Software Engineering,* 20, 88-101

Ruddy, T. F. (1991a). Euro-options: A review of some current efforts to encourage European sustainable transport. In: *Environment Strategy Europe*, p.127-130. Campden Publishing, London.

Ruddy, T.F. (1991b). Hydropower Technology Applied in Canada as a Source of Hydrogen. *International Journal of Hydrogen Energy*, Vol.16: 579-580.

Ruddy, T. F. (2000). Sharing the Sky. In Bohlin, E. (ed.). *Proceedings ASIS Conference,* February 21-22.2000.

Weiland, U. and Hilty, L. M. (1998). Sustainable Urban Management: Opportunities and Risks of Information Technology. In Hamm, B.; Muttagi, P. K. (ed.). *Sustainable Development and the Future of Cities,* pages 197-202. Oxford and IBH Publishing Co., New Delhi, Calcutta.

Weiland, U., Pietsch, J. and Hübner, M. (1994). The EXCEPT-Project: Expert System for Computer-Aided Environmental Planning Tasks – Methodology of Computer-Aided Evaluation. In Guariso, G., Page, B. (ed.). *CSEIA 93, Proceedings of the IFIP TC5/ WG5.11 Working Conference on Computer Support for Environmental Impact Assessment,* 1993, Como, Italy, IFIP Transactions B-16, pages 358-367. North-Holland, Amsterdam.