Simulation of Urban Transportation Networks
With Multiple Vehicle Classes and Services:
Classifications, Functional Requirements and
General-Purpose Modeling Schemes

UCI-ITS-TS-WP-02-18

R. Jayakrishnan
Cristián E. Cortés
Riju Lavanya
Laia Pages

Department of Civil and Environmental Engineering and
Institute of Transportation Studies
University of California, Irvine
rjayakri@uci.edu, ccortes@translab.its.uci.edu
rlavanya@translab.its.uci.edu, lpages@translab.its.uci.edu

August 2002
Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697-3600, U.S.A.
http://www.its.uci.edu
Simulation of Urban Transportation Networks with Multiple Vehicle Classes and Services: Classifications, Functional Requirements and General-purpose Modeling Schemes

R. Jayakrishnan
Department of Civil Engineering and Institute of Transportation Studies
University of California, Irvine, CA. 92967, U.S.A.
Ph: (949) 824 2172, Fax: (949) 824 8385
Email: rjaya@uiuc.edu

Cristián E. Cortés
Department of Civil Engineering and Institute of Transportation Studies
University of California, Irvine, CA. 92967, U.S.A.
Ph: (949) 824 5623, Fax: (949) 824 8385
Email: ccortes@translab.its.uci.edu

Riju Lavanya
Department of Civil Engineering and Institute of Transportation Studies
University of California, Irvine, CA. 92967, U.S.A.
Ph: (949) 824 5623, Fax: (949) 824 8385
Email: rlavanya@translab.its.uci.edu

Laia Pagès
Department of Civil Engineering and Institute of Transportation Studies
University of California, Irvine, CA. 92967, U.S.A.
Ph: (949) 824 5623, Fax: (949) 824 8385
Email: lpages@translab.its.uci.edu

Submission date: August 1, 2002
Word count: 8,612 + 4*250 = 9,612
Simulation of Urban Transportation Networks with Multiple Vehicle Classes and Services: Classifications, Functional Requirements and General-purpose Modeling Schemes

R. Jayakrishnan
Cristian E. Cortés
Riju Lavanya
Laia Pagés

Department of Civil Engineering and Institute of Transportation Studies
University of California, Irvine, CA 92612


ABSTRACT

While traffic network simulation itself is a field with vast literature, majority of past research has focused on the simulation of auto traffic. There has not been sufficient research on the simulation of transit as well as emergency and commercial fleets in the network, with distinctly different route and service characteristics. These systems however may interact with the auto traffic, either at the supply level through shared right of way and network control schemes, or at the demand level in terms of mode splits. Operational planning of real-time schemes for emergency or commercial fleet use also requires modeling methods that capture the network supply dynamics properly and thus needs to be done in conjunction with auto traffic simulation. With improved computational capabilities and the availability of detailed software, some of them commercial, simulation for operational design and analysis of urban networks by practitioners is seemingly becoming a viable option. Further research into improving fundamental auto traffic simulation models is also being undertaken by agencies and researchers. How comprehensive urban network simulation environments will be, is however influenced also by their capabilities to handle multiple vehicle classes. This paper focuses on the issues that arise, classifies the simulation and contextural options, discusses the associated modeling requirements and suggests schemes to develop environments that can comprehensively model networks with a complete mix of set of various vehicle classes. The paper includes a discussion of the state of the art in such simulations and hopes to evoke a deeper examination of the complexities in using simulation as a comprehensive tool for urban network analysis.
INTRODUCTION

This paper evolves from the recent research by the authors in simulating the real-time routing schemes in the case of a hub concept in transit systems, namely high-coverage point to point transit, a design variation of demand-responsive transit, but with large number of taxi-like vehicles routed in real-time. The designs included certain hub-and-spoke designs for portions of the vehicle routes and transfer schemes across vehicles. The study of dispatching rules based on real-time stochastic control required modeling the network congestion dynamics. Needless to say, none of the existing simulation software allows simulation of such a system. In fact, even the simulation of something very basic as a streetcar system or a paratransit service is not an option to the existing simulation software. A cursory study itself would reveal that simulation of any vehicle class other than personal autos is always developed as an afterthought. In most cases the simulation developers have done only a superficial addition of transit simulation on top of detailed simulation of autos and control mechanisms on freeways and arterials.

However, there are considerable new developments taking place in urban transportation networks, where real-time routing is being increasingly used for several fleets, both transit and commercial, due to the advent of wireless and GPS technology. On the other hand, there are no modeling tools available for studying dispatching rules or routing policies/alternatives for real-time routing — both for new transit schemes such as ADART (1) and HCPT (2), or for the case of emergency services, which is receiving renewed attention after September 2001. Newer but more conventional ideas such as Bus Rapid Transit (BRT) are also receiving attention thanks to some large-scale initiatives from FTA. Again, methods to research their performance characteristics are scant, and some researchers have attempted to use simulation for these purposes.

There are reasons why simulation could become a useful (and perhaps the only) option to study many of such systems. First, the systems do interact with auto traffic and influence the network performance, or are at least influenced by it. On shared right of way with auto traffic, buses, streetcars or LRT can influence the supply characteristics at least in the heavier transit corridors. Control schemes such as transit preemption or signal preemption by emergency vehicles affect the network conditions. In all cases of shared right of way, the auto traffic (with the well known non-linear congestion characteristics they collectively produce) influences the movement of these other classes of vehicles. Such interactions are difficult to model with abstract mathematical models other than in simplistic academic contexts, and simulation may be the only option. On the performance side, even the traditional capacity analysis schemes may use simulation schemes in the future, if the discussions at a path-breaking recent joint conference of the TRB committees on Highway Capacity Analysis and on Traffic Flow Theory at Truckee, California, 2001, were any indication.

An even more interesting scenario is the use of multiple vehicle-class simulation analysis for planning purposes in urban networks when one needs to model the demand-side dynamics in terms of modal split between say a transit system (or many transit modes with sufficiently different characteristics transit systems). In this case, the auto system and the network performance are certainly influenced by the transit systems and the resultant mode splits, and thus simulation is even more important because simpler performance models are very difficult to find. In fact, some researchers already use simulation-based assignment for planning. There are now indications that practitioners may indeed consider simulation as an option even for planning purposes in urban networks.

There are also expectations that microsimulations would be the way of the future, which is perhaps more due to their flexibility and detail-oriented design. This makes it easier for developers to at least claim to have incorporated capabilities (albeit often with insufficient insights or care and without well-tested fundamental models) to simulate various aspects of urban transportation networks that practitioners and researchers would like to study. Nevertheless, some of the packages with their attractive graphical displays and "bullet point" modeling capabilities have captured the attention of the transportation modeling community. In our view this is not necessarily a bad development in the long run - it is high time that the transportation area started using models with modern tools and wider scope similar to those in use in various other areas (weather models, biotech models, etc) for a decade or two. It
also provides the filip for public, private and research agencies to develop the underlying models more carefully, such as in the newly proposed FHWA initiative for private-public-research participation in enhancing new core models for simulation, called NGSIM (Next generation simulation models).

The above discussion points to the need for a careful look at modeling of urban networks in completeness, properly accounting for various vehicle classes and modal details. It is true that there cannot be any model that can simulate everything in an urban network, however the state-of-the art needs to improve significantly. This paper attempts to make some contributions in this direction. The primary purposes of the paper are,

- To show the needs and requirements of comprehensive multiple-class simulations
- To point out the complexities in the modeling
- To provide insights into attempting such modeling
- To suggest possible schemes that may find application in simulation
- To motivate discussion on the associated issues.

The paper does not intend to provide elaborate details on multiple-class vehicle and service simulation. It also does not pretend to provide indisputable claims on how to develop simulation environments for transit and fleet systems for all the possible kinds of real-time routing plans or right-of-way characteristics, demand and network contexts or vehicle-movement characteristics.

The paper begins with a background section that focuses on broad classifications and some of the existing simulation approaches, and proceeds to suggest possible modeling schemes. The final section focuses on such frameworks applied to developing simulation environments for a few selected vehicle classes operating with auto traffic. The discussions are perhaps influenced by the authors’ experience with some of the existing software packages, but are hoped to be largely applicable to most of the simulation packages and contexts the researchers and practitioners would encounter.

CLASSIFICATION OF SERVICE VEHICLES CATEGORIES OTHER THAN PERSONAL AUTOS

In this section a classification of the existing classes of vehicles and services is presented. Perhaps the most logical way to distinguish between several kinds of transit systems is by its functionality. It is possible to enumerate three different systems: transit, paratransit and fleet systems.

1) Transit: Such services are normally composed of fixed route systems with routes and schedules planned in advance. Trains, light rail vehicles, metros, buses normally continue traveling in a circuit or back-and-forth route to pick up or drop off passengers at the stop locations. The service characteristics of interest in simulation include items such as whether these services share the right of way among each other as well as personal autos.

2) Paratransit: Taxis, limousines, shuttles, and demand-responsive systems are examples of this. This class of services can be further subdivided according to their routing or/and scheduling characteristics. Using this feature, we have the following services:

- Dial-a-ride: According to [3], it is a service, which is more flexible than conventional transit services. It can be demand-responsive in the routing schemes (vehicles go exactly where the passenger wants - door to door service) or/and the scheduling (vehicles arrive when the passenger wants). Taxis would be effectively a special case of such services, where the passengers do not share rides.
• Jitney: According to (3), a jitney is a cross between taxi service and regular bus route. Jitneys operate along a fixed route, and they may have fixed stops. But there is no fixed schedule and passengers share the vehicle.
• Ride Sharing: In this case travelers form groups to share vehicles that operate when and where they want. From a simulation standpoint these would be similar to personal autos, but with potentially more stops for passengers to join the ride.
• Vehicle Sharing: These are systems where the rides are not shared, but different passengers or passenger groups share the same vehicle for multiple rides. The primary difference from a dial-a-ride or taxi service would be that there are no drivers for these vehicles and as such the vehicles are not "dispatched" (though there may be methods to control the passenger selection of vehicle in locations such as transit centers, as in the case of the proposed station-car schemes).
• Mixed systems: These are systems which may be considered similar to regular transit systems rather than paratransit, in that the vehicles may have fixed routes and transfer points in certain parts of the network but with non-fixed, demand-responsive or real-time routed portions in other parts of the network. An example is the recently proposed High Coverage Point to Point Transit (HCPPT) (2)

3) Fleet services: Commercial services and Emergency vehicles form the largest subdivision in this class. Regarding commercial vehicles, it is possible to see this system as a pick up and delivery problem with prior planning. At the beginning of the day or season drivers know where their demand is located for that period. Finally, emergency fleets include ambulances, fire and paramedical vehicles, and police vehicles. These may often be real-time routed services with a fixed drop-off location.

Classification of simulation types

Simulation models can be categorized according to the nature of the system that they are trying to represent, and of these concepts have been well-known in the general systems simulation area for decades. A discrete system is one in which the state of the system changes at discrete points in time. We can think of car arrivals at a certain point, occurring at distinct points of time, as events. Between two consecutive items nothing happens; that is, the state of the system remains unchanged. When the number of these events is finite, the simulation is known as discrete event simulation. A continuous simulation is one where the state of the system changes continuously over time. Water level in a dam may be thought of as a system where it changes continuously over time according to some known differential equations describing its state. Note that a discrete event simulation could be used as an approximation to continuous time simulation in many cases. Most systems in the real world are both discrete and continuous but usually one predominates over the other (4). Traditional traffic flow descriptions have been based on continuous speed and distance variables. As far as the personal auto traffic is concerned, continuous simulation is the only possibility, as the system performance (in terms of speeds, flow, and density) is the result of continuous interactions between vehicles. In other words, a continuous simulation for personal auto traffic becomes a requirement, as it is a "collective" system with continuous response. On the other hand, the simulation of control hardware operations (signals) as well as some of the fixed schedule transit systems have been done using event-based simulation.

In recent years, object-oriented or agent based simulations have been proposed to be useful in depicting traffic movement. Traffic can be viewed as a complex system composed of various entities interacting with each other. Through agent-based simulations, relatively complex global phenomena can be expressed as a sum of small, localized interactions among the agents in the system. The main entities in the traffic network are road segments, vehicles, traffic signals etc., which can be modeled as agents. These agents have the ability to perceive the changes in their environment. Based upon this perception, an agent can modify its behavior to achieve a certain goal. For instance, a vehicle on the road can sense its neighboring vehicles and can change its speed or acceleration, which is analogous to "behavior". Thus, this car-agent behaves as a real-life driver who wants to reach his destination while attaining certain
goals. An example of one such goal might be to reach the destination in the shortest time possible without violating any speed limits. An advantage of such an environment is that it lends itself naturally to distributed computing. Since each agent is in full control of itself, the whole simulation can be divided into individual pieces in an intuitive fashion, which can then be simulated by different processors. It must be noted that some of the recent literature on "agent based simulations" in traffic is only referring to the object-oriented design of the software, and the fundamental interactions simulated are not exactly true to the definition of agent interactions. Admittedly, the definitions in the literature are not rigid, however.

**Classification of simulation based on modelling the personal autos:**

Simulation models can be categorized according to the level of detail at which automobile traffic is represented. This is essentially a classification based on the modeling of one of the classes of vehicles of interest to us; however it is a significant classification because it is the dominant class of vehicles in most urban contexts. Furthermore, this classification is one that will conceivably influence the development of more comprehensive simulation environments in the future. The classifications are well known and the literature on auto-traffic simulation models is vast. We provide a brief overview for completeness.

Microscopic simulation models deal with a group of vehicles rather than treating vehicles at the individual level. General flow relationships, applicable to fluids, can be applied in these models to arrive at the condition of traffic at any given time. This level of aggregation can be usually found in static planning models of typically large areas. Being static, these models do not respond to changes in traffic conditions over a short period of time and hence, are fairly limited in their application. Many of the well known microscopic applications were developed in the late 1960's or the early 1970's. Examples of such models include TRANSYT, FREQ, FREFLO, META, SIMAUT and several special-purpose research simulation programs.

A slightly higher level of representation is provided in mesoscopic simulation models. These models are able to handle small changes in the traffic patterns over a short period of time, which can be of the order of a few seconds. The level of detail in these simulations may change over time depending on traffic conditions. DYNASMART, which uses a scheme based on macroscopic traffic flow relations but with individual vehicle tracking, is sometimes added in this category. Other examples include SIMNET, SATURN, PREDICT, CONTRAM and PACTSIM.

The highest level of traffic detail is provided by microscopic simulation models which simulate the time-space trajectory of each individual vehicle by applying models of car-following and lane-changing. They are more accurate than macroscopic models in estimating delays, queue lengths and other associated traffic characteristics, but often suffer from the deficiencies of the underlying microscopic models which may not have been well-calibrated. Representative examples in this category include AIMSUN2, PARAMICS, CORSIM, TRANSIM, MITSIM and VISSIM.

**Emergence of microscopic simulation as a viable practitioner option?**

The details in microscopic models yield the flexibility to add many more modeling contexts and options than macro and mesoscopic models, as well as show much more detailed graphical and animated displays. This makes it easier for them to be "sold" to the practitioners, despite the limitations of the fundamental equations therein, many of which may be rudimentary at this time (but could conceivably improve in the future if the models become popular). With some of the microscopic models having been developed commercially and marketed more aggressively than the models of the past, it is our opinion that microscopic simulation is indeed here to stay.

There is indeed a perceptible change in the practitioners' and researchers' view of microscopic simulation in recent years, brought about partly by the vastly improved computational capabilities and the development of a few elaborate commercial microscopic simulation software along with that. Microscopic simulation is now being considered as a potentially viable option for analyzing traffic networks in the near future. In addition to the analysis of real-time operational policies in urban
transportation systems, microscopic simulation is considered a possibility even for planning purposes where static (assignment-based) models have been the primary method for decades. This has resulted in a recent proposal from the Federal Highway Administration to develop the Next Generation Simulation Models (NGSSIM), which will be largely based on microscopic models, but with the research community working with the simulation software "vendors" to further develop the core models in them. A motivation behind this paper is the need to look beyond the simulation of personal autos for such simulation to become useful in the future.

Capabilities of current microscopic simulators in modeling non-auto traffic

The differences between microsimulation models existing in the market have been broadly studied in several research projects, such as (5), (6) and (7) and their applications have been tested in many studies. However, work involving these microsimulation packages has always been related to general traffic; very few studies in the literature deal with the simulation of transit and other special vehicle fleets. In addition, the objective of introducing the simulation of transit in a network has always been to evaluate the automobile performances taking into account the private user (car-owner) standpoint, thus focusing on the effect of transit on auto traffic. In very few cases has the simulation of transit been a tool to study and analyze the performance of the transit system itself from the point of view of the transit operator. Not many studies have been performed where the objective has been to simulate transit as an end in itself. In the following discussion, we use the word "traffic" to refer to the auto traffic.

Some of the traffic simulation software packages in the market nowadays are able to simulate fixed route transit systems quite accurately. In order to model such a fixed-route vehicle class, users have to follow the following steps, largely independent of the software they use. The routes for the system have to be predetermined, then locations of the stops need to be fixed and, finally, the frequencies of the service for all the routes have to be input. The packages such as PARAMICS and VISSIM have many vehicle types, allowing users to choose a different type for transit vehicles. The packages do not necessarily include sufficient details to model the operational characteristics of any type of vehicle, such as a people mover system or a streetcar system that shares the right of way with auto traffic.

Simulation packages vary in their ability to simulate transit. For instance, most available simulators have a fixed time for stoppage. However, in VISSIM, the duration of wait at a given stop for a transit vehicle can also be specified as a function of the demand at that place, namely the number of people waiting to take the bus in this location. Only VISSIM allows vehicles to stop/stage on the left-hand side of the road. Signal preemption schemes for transit vehicles are very difficult to model properly in the existing simulators primarily. Signal preemption for emergency vehicles is generally impossible to model in the packages because there is no simulation of a special class of non-auto vehicles without fixed routes. The available simulation packages are not flexible enough to simulate a real-time routed transit systems without any external subroutines like Application Programming Interfaces (API's).

Special Techniques to simulate Transit Systems

Some researchers have invented approximate techniques to overcome the deficiencies of commercial packages in simulating rail transit systems, in effect "tricking" the package to do transit simulation.

For the simulation of LRT (Light Rail Transit) in (6) using NETSIM, the network had to be modified in order to have fewer nodes and different configurations regarding the preferences in the intersections. For simulating a BRT (Bus Rapid Transit) in (9) using VISSIM, the corridor had to be divided into shorter sections in order to model the center-running guided busway on an arterial street. A different vehicle type was coded for the buses as well. Another example of a BRT simulation using VISSIM is in (10). The network was coded similar to the preceding case, and additional signals were coded to hold vehicles at some pre-specified locations in the network and maintain a constant headway. The control held a bus in a location if its headway to the bus ahead of it was less than the minimum time desired. Another interesting example is a simulation of the streetcar system using PRAMICS for the city
of Toronto (11). Tracks were coded on top of the existing network, stops were coded as additional nodes with virtual traffic lights affecting only streetcars. The traffic was trapped behind the streetcars during the red phase of the traffic light. In this case, the duration of the stop of the streetcar was a function of the demand. Note that in all the above cases the location of the stops has to be pre-specified. It is not clear how to simulate a system with uncertain demand occurring at any random point in the network. The struggle by the researchers and practitioners to attempt to simulate anything other than an auto traffic system is cause for a careful look into the requirements for comprehensive microscopic simulation environments to be developed for the future.

NETWORK HIERARCHY AND NETWORK SIZE ISSUES

The primary difficulty with many microscopic simulation models is their inability to handle path dynamics in large networks. For example, PARAMICS allows vehicle routing according to routing tables and feedback capturing information supply, but does not allow storage of sufficient path trees and storage of individual vehicle's routes, which are essential requirements for the simulation of route choice. The difficulty arises from the detailed network descriptions used in such microscopic simulation models. The node and link representations for microscopic simulations are often such that any point on a physical link with a change in geometry or other characteristics results in an extra node in the representation.

The need to properly model the street and lane infrastructure in microscopic models results in an order of magnitude more nodes and links, than needed to model the path dynamics, which requires only the network comprising the true decision nodes. These are the nodes that are of significance in for example, the transit driver's route decision or the route decision taken by a central dispatcher in order to optimize the operation of a service fleet. Microscopic models such as PARAMICS have the scalability to permit vehicle simulation of large networks, but if detailed service response modeling and path processing are to be incorporated, such models can only be used to simulate small to medium-sized urban areas. This is because many network path processing algorithms show nonlinear increase in storage and computational requirements as network size increases, as opposed to the auto traffic simulation algorithms that can be intuitively seen to operate on local variables and thus show linearly increasing computational requirements.

Thus, if we consider large networks where a microscopic simulation data sets would include several tens of thousand nodes, the storage of individual paths of each transit vehicle with such networks require prohibitive random access memory (RAM). This is true even with modern computers and thus microscopic models developed for such purposes will justifiably have limitations in path processing. It is logical to see that traffic flow modeling requires only local information and can be very scalable. That is, larger networks can be modeled with carefully developed distributed processing schemes that allocate the modeling of portions of the network to different processors. On the other hand, modeling changes in path-related characteristics such as travel times, and path-related decisions by transit managers, requires information from possibly all parts, i.e. paths going across sub-areas. Thus microscopic models which were developed with initial focus on auto traffic simulation in relatively smaller regions would need to be augmented with schemes to handle them at a different level of network abstraction, as discussed further below.

A previous work by (12) developed a hybrid simulation approach, integrating the PARAMICS microscopic simulation with the routing and behavior response simulation schemes as in DYNAMSMART (13), so that the integrated simulator could evaluate information/routing schemes with route choice behavior models. This approach was based on integrating networks of two different levels of abstraction and communication of vehicle positions between the detailed network (as in PARAMICS) to the more abstract network (as in DYNAMSMART). The vehicle route decisions processed at abstract network are then transmitted to the detailed network simulation that controls vehicle movement at the microscopic level. The integrated simulation program allowed them realistic evaluation of a variety of technologies in advanced traffic management and information systems (ATMIS).
The scheme proposed in the context of this paper is based on some ideas developed in (12), but oriented to a different kind of approach for communication, integration and routing. It is possible to study the detailed operation of a general transit or commercial fleet system, where all the path-based decisions, treatment of passengers and routing of transit vehicles are made at an abstract level, while all traffic operations are controlled by the microscopic simulator. Thus, the vehicle route decisions processed at abstract level are then transmitted to the detailed network simulation that controls vehicle movement at the microscopic level.

Hence, one of the key aspects of our approach is the data communication from the micro-level to the abstract level and vice versa. The idea is to create a simplified network to be used at the abstract level, but consistent with the original network coded for running the microscopic model. In addition, this process may require the construction of a lookup table (“communication interface”) for passing information between the two networks, such as level of service and detailed information of individual vehicle positions, speeds, route decisions, etc.

The equivalent abstract network (henceforth ABSNET) is made taking into account the following possible simplification. In terms of link characteristics, a relatively simple program can aggregate across those microscopic links, whose end nodes represent only a change in geometry or capacity. Note that this includes all nodes except for the real decision nodes such as intersections or interchanges. Calculation of link cost in the abstract network is consistent with the microscopic model link cost calculation, which includes link costs themselves, and turn movement costs. Look-up tables identify the original links that correspond to the abstract network links and to aggregate travel times on them.

In the case of simulating transit systems running on a subset of the whole area network (such as a fixed route system), the ABSSNET will be a representation of just that sub-net (the rest is not needed at the abstract level). Moreover, for fixed route transit and fleet systems, where the route remains fixed, it is not necessary to transfer information related to link and turn movement costs, since there are no decisions taken according to the level of congestion of the network at any time. This is in contrast to simulating taxi or “dial-a-ride” systems where the driver or the central dispatcher could choose the shortest path to get to his next stop, depending upon network traffic conditions. In this case, however, it is important to communicate vehicle-related information between both levels of abstraction, say the vehicle positions and the stop-time required by every vehicle when it reaches its predefined bus stops.

In the next section we provide a detailed description of a hybrid scheme to simulate a general transit or service system using detailed traffic simulation from a last generation microscopic simulation model. The general approach requires the microscopic model to have some modern software capabilities in terms of the development of functional interface or application programming interface (API), allowing additional functionality by adding more external modeling routines. Many of the existing simulation software do allow such APIs, which may also be called plug-ins, as the developers are aware of the need for it for anything but the simplistic auto traffic simulation in realistic urban networks.

A GENERAL PURPOSE SCHEME FOR SIMULATION OF MULTIPLE CLASSES OF VEHICLES AND SERVICES

The modeling scheme we suggest is for simulating any kind of flexible real-time routed service. In this sense, it will be possible to model optimal routing algorithms which may be based on the individual vehicle’s position, passenger calls, and real-time traffic conditions, in case of studying “re routable systems”, such as typical paratransit or “dial-a-ride” designs. In addition, the most general “pick-up and delivery” commercial fleet contexts could be modeled under real-time traffic conditions, incorporating any kind of decision rule to assign vehicles to serve customers optimally. Note that fixed route systems are essentially a restricted subset of flexible-route systems or real-time routed systems.

The premise is that once the routing modules are separately coded in the simulation environment through the API, they can be easily modified to simulate various designs of fixed route systems, feeder...
short-haul services, etc. Customer demand generation and performance measures are also embedded into the simulation framework.

The new capabilities need to include the detailed modeling of vehicle operations at stop or passenger pick-up and delivery locations, real-time traffic network conditions impacting vehicle travel times and user waiting times at pick-up locations.

In Figure 1 we present a scheme to represent the proposed framework for simulating/evaluating a general commercial fleet service. General in the sense that all modules composing the integrated system can be adapted to simulate most of transit and commercial services available nowadays (see the next section for examples of application of such a scheme to the modeling of various multiple-class vehicle/service systems).

<Insert FIGURE 1 here>

First of all, in the figure it is possible to visualize the two levels of aggregation. On the right side of the chart, we have all the procedures (modules) corresponding to the aggregated level, including the implementation of any sort of routing/scheduling rule and the customer behavioral models. Note that we use the term "aggregated level" only under an assumption that the network on which the transit or fleet system's routing and other details are simulated would be a smaller version of the detailed microscopic auto traffic simulation network, generally with only "decision nodes", as mentioned above. It would be entirely appropriate to call this network "transit/fleet routing network".

Next we describe three important issues regarding the proposed scheme: the objects (fundamental data structures), the important events that trigger some action at the aggregated level, and the routing/scheduling rules and interrelation modules within the API.

Fundamental Data Structures

The basic API frame is composed by the fundamental data structures that can be either objects or agents, depending upon the kind of system to be simulated. These are the "fleet", the "customer (user)" and the "network" data structures.

The "network" data structure keeps the information contained into the ABSNET (simplified network) and the way in which the information is kept within it, will depend on the kind of algorithm to be run in the "Routing and Scheduling" module, requiring any attribute of the network at any time (i.e. link and turn movement cost). For example, we could need a different data structure for running efficiently a shortest path algorithm, or a TSP-based routing for vehicle dispatching. The "network" data structure read information from the "network conditions update" routine, which is fed directly off the detailed microscopic network conditions via the "communication interface" module that translates every network attribute from one level of aggregation to the other. The interface includes the necessary Callback functions as defined in the microscopic simulation software's features). Normally, the network conditions used in the context of our scheme are the link travel time and turn movement cost (intersection delay).

The "customer" data structure can keep all the information obtained from a demand table, which can be either known in advance or generated in simulation time. It is important to store the attributes, special requirements and general features of the customers of the system. All information obtained from the simulation concerning the performance measures associated with the customer (say average ride and waiting time, which can be at the pick-up spot or in a bus stop, etc.) need to be stored, as these would be needed in any customer decision process modeling.

Finally, the "fleet" data structure becomes the central object of the system, keeping the details and behavior of all transit vehicles at the aggregated level. For each vehicle in the transit or fleet system that we keep track of in the ABSNET, we have a corresponding actual vehicle moving around in the microscopic simulation model. The correspondence between vehicles needs to be handled through a look up table (or a hash-table, depending on the microscopic simulation model's vehicle naming/numbering...
method). In this data structure, we will store the vehicle paths and stops at the aggregated level in order to control the movement of such vehicles at the microscopic level. These paths can be either fixed or variable, depending on the system, routing rules, etc. In addition, this data structure keeps all transit vehicles’ features and they are connected to a module that stores the statistics of the performance of the system from a vehicle (manager) standpoint. In all figures hereafter, dotted lines pointing to data structure boxes represent all those procedures that update the information contained in the respective data structure.

Simulation events in the API

In the background section we pointed out the difference between continuous and discrete simulation. The way in which microscopic traffic movement is simulated is by discretizing the continuous operation of vehicles on the network over time, using a fixed time step $\Delta t$. On the other hand, the nature of commercial fleet operations makes the use of a discrete event simulation approach more attractive. The key factors that trigger a change on the evolution of the system at certain moment are basically discrete events (see for example, the approach used in [2], for simulating spatially a high-coverage point to point transit system under simplified conditions). In this case, we will use the simulation tools provided by the microscopic model itself (defining both, the simulation time-step $\Delta t$ and the update or feedback time-step $\Delta f$) along with the embedded nature of a discrete event simulation for calling the API’s procedures.

In the scheme of figure 1 we find two discrete events generating an action performed by some of our external routines: “Service request” and when a “Transit vehicle reaches a stop”.

- **Service request**: Every time a customer asks for service, the central dispatcher (or whoever is in charge of the assignment of passenger-vehicle) has to take a decision of routing and scheduling, changing the conditions of the system. Basically, the general “Routing and Scheduling Rules” are called in order to decide which transit vehicle has to serve the new customer, and in which position of the specific vehicle’s route, among all previously scheduled stops. Once this decision is made, the vehicle’s path is modified in order to insert the new request into the original vehicle’s route, changing the predefined vehicle’s path at the aggregated level. This event happens at absolute time $s$, measured from the beginning of the simulation time $t_0$. The new vehicle’s route (or vehicles’ routes in case we modify more than one vehicle path) is communicated to the microscopic model via the “communication interface” after the execution of the time-step at which $s$ belongs to. Then, the new route (or routes) is (are) introduced into the microscopic level.

- **Transit vehicle reaches a stop**: Every time a vehicle reaches a stop location a transfer operation happens (whether passengers boarding to or alighting from boxes, or a certain load is picked up or dispatched at a certain location, etc.). At this moment, say time $t$, the physical interchange takes place (“pax-vehicle transfer” box), modifying both the “customer” and “fleet” databases. The former is needed as there are changes in the status of the customer at $t$. The latter is needed because when a vehicle reaches a stop location, its load and status change. The details of the stop event operation depend on the type of operation, number of entities transferred, conditions of the transfer, physical place of the stop, etc. After considering all these stop features, the stop time and exact location of it are then transferred to the detailed microscopic network in the same way as before. At the microscopic level, some Control function written in the API software have to be modified in order to make the stop as realistic as possible, most of them associated to lane changing and car following procedures.

In the next subsection, we describe the role played by the “Scheduling and Routing Rules” in the context of the design of such an API, the importance of this module under different schemes and its interrelation with other modules and data structures.
Routines for Routing and Scheduling

The box containing the Scheduling and Routing Rules cannot be explicitly defined for the general case. Every particular system is commanded by a different set of rules, however, there are some common characteristics regarding this issue that can be broadly discussed here. A more detailed definition of this topic is presented in the next section for each particular example. The discussion is largely about transit systems, but replacing the word "passengers" with "packages" would yield insights into a package pick-up-and-delivery system as well.

As we show in Figure 1, this procedure is called every time a new service request enters the system. The main objective of this procedure is to decide the customer-vehicle assignment following any general objective function to be optimized (minimize user cost, or to minimize a combined measure of user and operator cost, to maximize productivity, to maximize profit, etc.).

In some cases, as in service of fixed route, the rules are oriented only towards the assignment of passengers to the right vehicles, once the vehicle arrives to a terminal or bus stop, depending on the distribution of the demand, and customer features. In case of "dial-a-ride" services, on the other hand, where vehicles can change routes dynamically, the objective concerns both passengers and vehicles. And, for mixed services, combining long-haul corridors fed off "retrieval" services operating on surface street areas, routing rules will apply to some vehicles but not for others under fixed route operation. The interchange of passengers occurring at terminal or hub locations will define the limit between one kind of operation and the other.

The other important issue to be considered in these procedures is more related to the nature of the demand: whether it is a system with uncertain demand generated in real-time or it is a system where the demand is known in advance. In the first case, depending the scheduling rule used, the vehicle-passerger assignment does not necessarily have to end up as a transfer. The reason is that the optimization is made in real time and the assignment decisions could change over time before the transfer itself. That is why the "passenger-vehicle assignment" routine does not mean an update in the "customers" data structure attributes. In case of demand known in advance, the optimization would most probably take place at the beginning, and the resulting routes will remain invariant over the whole simulation period.

With regard to the initialization of the system, we need to set the initial vehicle routes (which will not change in case of fixed-route services), read the vehicle lookup table, the ABSNET and the demand table as well. All this information has to be input as a part of the API-setup function.

In the next section, we present some specific examples of how to apply our scheme for specific transit and commercial fleet systems as described in a previous section. The corresponding routing rules, procedures and differences in coding/implementation of every studied system are highlighted and discussed in detail.

SOME APPLICATIONS OF THE PROPOSED SCHEME

The above-described hybrid scheme is flexible enough for the modeling and analysis of a wide variety of commercial fleet and transit systems. In this section we show the possible applications of the general framework to three transit systems with different characteristics.

Common cases

First at all, let us develop the scheme for the analysis of para-transit transit systems as defined in (3), extendible to a variety of non-fixed route systems such as a typical demand responsive transit or a taxi service. In addition, let us suppose that the demand is uncertain, and that vehicle's routes can change over time depending on the new calls entering the system. This is what we call a real-time routed system. A representation of such a scheme is drawn in Figure 2.
The most relevant difference with respect to the general case is the detailed representation of the "Routing and scheduling" routine. In fact, this procedure is now split into two pieces: the "Cost comparison" and the PPSP (point-to-point shortest path) modules. Thus, once a request enters the system, the pick-up and drop-off locations are automatically specified. Then, the central dispatcher has to decide which vehicle to send in order to serve that request ("Cost comparison" module). To do that, we have to compute the best service option, based on a set of rules to be updated in real-time using historical system information through the "Network condition update" module.

Notice that since vehicles are moving on a real network, the travel time between any two points of the network has to be measured through some path, which will be the shortest path between these two points under the traffic conditions known by the system (provided by the microscopic simulation model via the "Network condition update" module). In addition, notice that the "Cost Comparison" most likely utilizes an insertion algorithm that compares the best insertion of the stops associated with the new request between any feasible pair of stops already scheduled to any candidate vehicle, and among all such vehicles. Therefore, the implementation of the PPSP algorithm needs to be very efficient since it will be called many times for every assignment decision, and these decisions have to be taken in real time. Some other important features to be considered for enhancing the "Routing and scheduling" module are, of course, the efficiency of the network data structure as well as the efficiency of the insertion heuristics into the "Cost comparison" module.

Regarding the network attributes required at the abstract level only the "link cost" and "turn movement cost" are necessary for running the PPSP routines. In addition, the PPSP routine can accept any generalized measure of cost that might be different from travel distance or travel time. This allows for higher levels of flexibility in depicting system costs and also avoids the problem of hard-coding costs into the simulation itself. For instance, in the traveling repairman problem (in pick-up/delivery logistic fleet systems), the cost incurred in servicing a customer may not depend only on the travel time; there might be fixed costs involved just with the decision to serve that particular customer.

A second case, quite different from the paratransit service, is the modeling of a fixed-route service.

In this case, as we show in figure 3, the "Routing and Scheduling rules" module is not relevant in setting vehicle routes, since they have been previously determined by the planner at the beginning of the simulation. However, there is still a module that replaces the routing rules. This module handles the proper distribution of customers at stop locations, the simulation of passenger queues at stops.

Once the boarding and alighting has happened at the bus-stop or terminal, the API sets the duration of the stop (depending upon number and type of passengers, the characteristics of vehicles, etc.). Since the location is known in advance all this information is then conveyed to the microscopic model via the "communication interface" in order to call the set of control functions associated with the stop operation at the microscopic level.

Notice that in this case the only event that creates an action is the "vehicle reaches the stop" event. The "service request" event only accumulates passengers in queues at terminals or bus stops, but it does not create an impact on the microscopic simulation state. That is why the network conditions are not important at the aggregated level, and therefore, the "Network condition update" module can be easily removed from the API.

With regard to the ABSNET, as we mentioned in a previous section, the topology corresponding to the transit route system is the only portion of the microscopic network that needs to be translated in order to keep track of transit vehicle positions at the aggregated level.
A Special Case

Finally, let us discuss the simulation of a combined (mixed) service, such as that described (2), which is a real-time routed transit service but based on a novel design, loosely called High Coverage Point-to-Point Transit System (HCPP). Transit vehicles are assigned to certain "hub areas", and each of such vehicles has a "reroutable" portion of their trip within its assigned area (as if it were a typical "dial-a-ride" service) and a "non-reroutable" portion on a trunk line to a given neighboring hub (as if it were a "fixed route" service). Therefore, ultimately that system turns out to be a combination of the both previous schemes illustrated in figure 2 and figure 3.

Figure 4, though seemingly more complicated, is still a representation of the general scheme illustrated in figure 1.

We keep the same data structures for all objects, however the interrelation among entities is more complex due to the addition of two vehicle states corresponding to its presence on either the fixed portion or the reroutable portion of the trip.

In terms of events, we incorporate one additional event forcing an action at the abstract level, which is when vehicle enters the trunk portion of its route. At this time, the modeler sets the path as fixed, and sends that information to the microscopic simulator. The procedure "vehicle reaches stop" is further subdivided into two different cases: whether the stop is a customer location or a hub (terminal). In the first case, the next set of actions is similar to that of the dial-a-ride scenario. On the other hand, when the stop is a hub the vehicle performs deliveries and pick-ups in the same way as in the fixed-route scheme, but in this case the passengers that are delivered there may not necessarily leave the system because may not be their final destination. In general, they are transferred to a different vehicle depending upon factors such as the size of the queue, the frequency of service, etc. That is why the "distribution of customers at hubs forming queues" procedure is depicted by a closed-loop waiting line connecting deliveries to pick-ups.

This procedure is extendible to any multimodal transit system.

We also add a new function within the "routing and scheduling rules" routine, in order to compute the initial TSP route of a vehicle entering the reroutable portion of its journey for distributing all passengers picked up at the hub stop. The resulting route is conveyed to the microscopic simulation model through the "communication interface" module. All other procedures remain unchanged.

The above examples illustrate the issues that come up in modeling different types of transit systems, and how the basic scheme we introduce in Figure 1 can be used with necessary modifications for different types of multiple class vehicle/service systems. Myriad other issues will inevitably arise when one attempts to simulate transit and other systems operating with auto traffic in urban networks. Depending on the nature of API functionality provided by the given microscopic simulation software, the basic scheme would need changes. We hope that the discussions in this section have touched on sufficient number of key issues and have provide enough insights for the interested researchers to attempt further research and development work for comprehensive simulations of urban transportation networks.

CONCLUSIONS

This paper discussed the need for developing more comprehensive urban transportation network simulation environments that go beyond the conventional microscopic simulation models that have largely been auto-centric. The paper discusses the recent interest among practitioners towards the use of micro-simulation and the renewed interest among researchers in considering such simulations as a potentially viable option in the future, especially with commercial software vendors entering the market place in a bigger way than in the past. That more fundamental research is still needed into auto traffic
microscopic simulation is relatively well-known, but how far we need to go to attempt complete multiple class vehicle simulations is less discussed. We hope that this paper has made an initial attempt at this.

The discussions in the paper are often slanted toward the issues we have encountered in our research, but several more issues would invariably arise in future research. It does appear however that there may be no easy way around the complexities involved in multiple vehicle/service class simulations, and the solutions would quite possibly be complex too. The paper provided a candidate framework to develop such simulations and presented the application of such a framework for certain simulation contexts.

Much further work remains to be done. One such area concerns the general-purpose vehicle movement simulations, especially for "unusual" vehicles, such as articulated coaches, tractor-trailer trucks, and even some of the vehicle types proposed in BRT and LRT systems. Much more work would be needed before any simulation environment as discussed in this paper would become useful for real-time evaluation of operational plans (as opposed to off-line evaluation). This would require methods to incorporate such systems into even more complicated dynamic traffic assignment models. Suffice to say that the directions of use of simulation would drive much of future work on this topic. To expect that everything in an urban transportation network can be simulated in one place is quite unrealistic; however it is not incorrect to say that the state of the art needs to improve. We hope that this paper provides insights to attempt some of the improvements.

REFERENCES


List of Figures

FIGURE 1 Simulation Framework for a General Transit System
FIGURE 2 Simulation Framework for a Dial-a-ride System
FIGURE 3 Simulation Framework for a Fixed Route System
FIGURE 4 Simulation Framework for a Real-time Routed System
FIGURE 2 SIMULATION FRAMEWORK FOR A DIAL-A-RIDE SYSTEM
FIGURE 3 SIMULATION FRAMEWORK FOR A FIXED ROUTE SYSTEM
**FIGURE 4 SIMULATION FRAMEWORK FOR A REAL TIME ROUTED SYSTEM**