Title
PLANiTS: THE METHODS BASE, Model Selection and Model Integration

Permalink
https://escholarship.org/uc/item/5hk3t911

Authors
Picado, Rosella
Khattak, Asad
Lovell, David J.
et al.

Publication Date
1995-05-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

PLANiTS: THE METHODS BASE, Model Selection and Model Integration

Rosella Picado
Asad Khattak
David J. Love11
Adib Kanafani

California PATH Research Report
UCB-ITS-PRR-95-16

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department of Transportation, Federal Highway Administration.

May 1995

ISSN 10551425

Institute of Transportation Studies
University of California at Berkeley

PLANiTS: THE METHODS BASE
Model Selection and Model Integration

Rosella Picado

Asad Khattak

David J. Love11

Adib Kanafani

109 McLaughlin Hall
Berkeley, California 94720

Phone (510) 642-3585
Fax (5 10) 643-5456

Submitted to:

The California Department of Transportation

(MOU-52)

Final Report

Volume 4

First submitted for review in October 1994

December 1994

ACKNOWLEDGEMENT

This research was conducted at the Institute of Transportation Studies, University of California at Berkeley. It was funded by the California Department of Transportation, as a part of the PATH Program. PLANiTS team members include: Adib Kanafani, Asad Khattak, Randall Cayford, David Lovell, Thananjeyan Paramsothy, Rosella Picado, Joy Dahlgreen, Melanie Crotty, Nicholas Vlahos, and Marvin Manheim.

ABSTRACT

This paper presents some features of PLANiTS (Planning and Integration for Intelligent Transportation Systems). In particular, it discusses some functions of the Methods Base. These include representation of model chains, selection of models and data, and data transformations and aggregation. Sequences of models are represented using graph notation. Models and data are selected given the Planning Vector specification, although users may override system selections. The Methods Base features are illustrated with an example drawn from the PLANiTS prototype (version 1.0).

Keywords: Intelligent Transportation Systems, Transportation Planning, Decision Support System, Software Engineering, Prototype Development

EXECUTIVE SUMMARY

The objective of this paper is to present a basic structure for the Methods Base of PLANiTS (Planning and Analysis Integration for Intelligent Transportation Systems). The Methods Base performs the analysis and estimation functions required by PLANiTS. It assists users in selecting models that are appropriate for evaluating a Planning Vector. This paper emphasizes the direct application of existing models to solve problems within the PLANiTS environment. The paper deals specifically with two characteristics of the Methods Base: representation of analysis methods, and selection among complementary analysis techniques.

Urban transportation planning deals with a wide range of issues. The amount and variety of models and methodologies developed so far to address these issues is large and complex, and expected to become more so as new analysis techniques and legislation are developed. To support analysis and estimation in PLANiTS, we have developed a flexible framework that allows the integration of existing transportation planning models and generic methods of analysis.

We call an analysis process the collection of methods that specify sequences of models and transformations to be applied in the evaluation of a Planning Vector.. An analysis process is composed of all the different methods that could be used to estimate the performance measures associated with a particular action. Analysis process can be represented using two-dimensional graph notation. Different metrics may be used for evaluating and comparing possible analysis

paths. It should be kept in mind though, that no part of the PLANiTS philosophy is intended to circumvent the human decision making process, only to facilitate it via the provision of information and tools not necessarily available otherwise.

The Analysis Agent must not only choose an analysis method, it should also control the ordered execution of all models and transformations, and control the flow of data between the different data bases and models. A key issue for supporting model execution is the ability to aggregate and disaggregate data. Data aggregation is an important function of the Methods Base because if allows for the evaluation of Planning Vectors across multiple dimensions in space, time, and user characteristics. Aggregation may occur at two places in the analysis: when searching for data to feed as model inputs, and when analyzing model inputs and summarizing them as performance measures.

LIST OF CONTENTS

Page

ACKNOWLEDGMENT i

ABSTRACT ii

EXECUTIVE SUMMARY iii

LIST OF CONTENTS v

LIST OF FIGURES vi

LIST OF TABLES vii

1. INTRODUCTION 1

2. MODEL SELECTION

2.1 Functional specification of a model: inputs, outputs, and environment 2

2.2 Model chaining: introduction of the analysis process concept 6

2.3 Criteria and procedures for model selection 13

2.4 Data transformation and aggregation 18

3. MODEL EXECUTION

3.1 Data selection (model inputs) 24

3.2 Control routines 36

4. EXAMPLE: PLANITS PROTOTYPE VERSION 1 .O

4.1 HOV lane example 39

4.2 ATMIS analysis example 47

5. CLOSURE 48

REFERENCES 51

LIST OF FIGURE3

Page

1. Methods Base Structure 4

2a. Analysis Process Representation. Graph Tree 8

2b. Analysis Process Representation. Graph Matrix 9

3a. Node Selection Sequence 34

3b. Node Selection Sequence 35

4. Sample User Interface Window for a Model 38

5. Analysis Process Representation. Add HOV Lande Evaluation 40

6. Add HOV Lane Evaluation. Method 1: MINUTP Only 42

7. Add HOV Lane Evaluation. Method 2: FREQ Only 43

8. Add HOV Lane Evaluation. Method 3 : from MINUTP to FREQ 44

9. Add HOV Lane Evaluation. Method 4: from MINUTP to FREQ to MINUTP 45

10. ATMIS Evaluation Flowchart 49

LIST OF TABLES

Page

1. Criteria for Selecting Transportation Objects 31
2. Attribute Inheritance Structure 32

1. INTRODUCTION

To address urban transportation planning problems, we have developed a methodology called PLANiTS ((Planning and Analysis Integration for Intelligent Transportation Systems). A detailed description of PLANiTS appears in Kanafani, Khattak, Crotty, and Dahlgreen (1993); Kanafani, Khattak, and Dahlgreen (1994); and Vlahos, Khattak, Kanafani, and Manheim (1994). This new planning methodology integrates transportation planning and operations models with knowledge-based systems and electronic group decision support.

PLANiTS consists of four main elements:

- Database and Knowledge Base
- Strategy and Action Base
- Policy and Goals Base
- Methods Base

This report describes some features of the Methods Base. The main purpose of the Methods Base is to perform the analysis functions of the planning process. This base assists users to select models that are appropriate for evaluating a Planning Vector. To evaluate a Planning Vector generally means the calculation of measures of performance. This can be achieved either by using existing models, or by developing new models. Model development is supported with modeling techniques such as regression, simulation, etc. Here we focus on the use of existing transportation models commonly used in urban transportation planning and operational analysis. Emphasis is on the direct application of models to solve problems within the PLANiTS environment.

System performance estimation involves searching for an answer not only in the Methods Base, but also in the Knowledge Base. The search should be driven by the type of action to be evaluated and its parameters, by the measure of performance and its parameters, and by the environment dimensions. Ideally these should be sufficient to detect both the method and the data needed to evaluate it, and/or to search for an answer or advice in the Knowledge Base. Yet we recognize that these may not be enough: there may be more than one method (composed of one or more models) that could be applied to obtain an estimate, but not sufficient criteria in the planning vector for the Analysis Agent to choose one method over the others. The reason is that there are different methods and models that perform similar tasks.

This paper presents current developments in the structure of the Methods Base. The two specific problems tackled here are how to represent the analysis flow for a set of transportation projects and how to select among complementary analysis techniques.

2. MODEL SELECTION

2.1 Functional specification of a model: inputs, outputs, and environment

For the purposes of analysis, a model can be viewed as containing a fixed relationship among elements of a Planning Vector, such that when provided with inputs it predicts outputs. A model can be thought of as a black box that receives a set of inputs and produces a set of outputs. Yet all black boxes with similar inputs and outputs are not the same; they get activated depending on the characteristics of the environment in which the model is supposed to work. In that sense the internal structure of the model is important: aspects such as the model's underlying theory, its range of application, robustness, etc., can and should be used to decide which model is appropriate in a specific situation. By and large, though, this is not a decision that can be made by a software program; at the most it can suggest alternative courses of action. The selection of specific models usually requires an expert in the field of application, or for that matter an expert system that learns from repeated applications. Ultimately, this would be one of the functions of the Model-Based Reasoner in PLANiTS. For the time being, the focus will be on existing models applied on specific contexts. For the remainder of this discussion then, models will be treated as black boxes.

Although models are self-contained, they need to be connected to the other elements of PLANiTS. In particular, they are linked to the underlying data object structure in order to be able to receive and export data; more importantly, they are linked to the Planning Vector elements so that the Analysis Agent knows when to use a given model.

It is important that models communicate with each other. For this purpose, the main channel is the object-oriented structure. Models are not required to talk directly to each other, all communication is performed via the underlying data structure. This of course simplifies not just the initial implementation of the system, it also allows for the addition of new models as plug-in units. Figure 1 illustrates the concept. The Analysis Agent can communicate with each model through the Data Model and the Data Transformations. Potentially, a given model could receive input from any other model via the Data Model. For example, Model 1 could be a trip

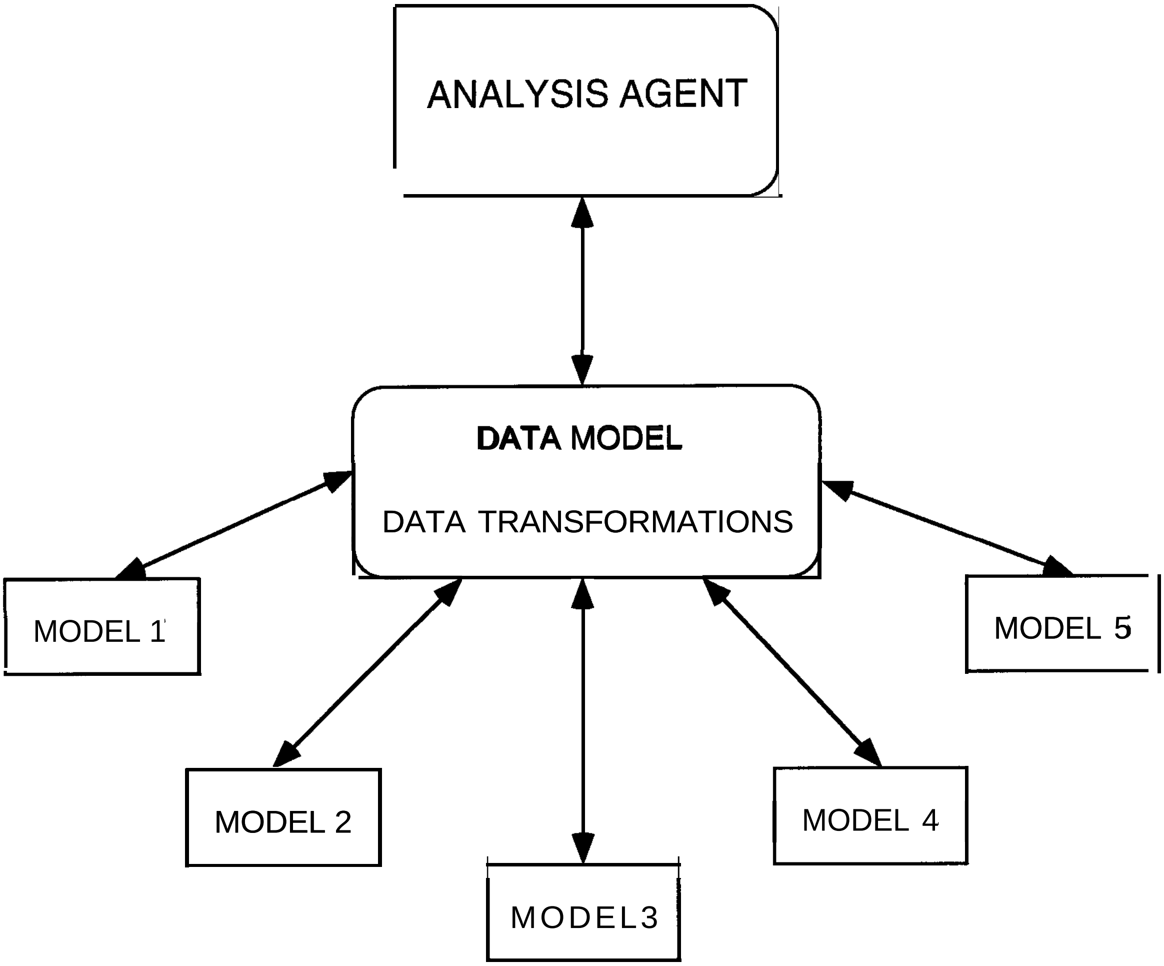


Figure 1

METHODS BASE STRUCTURE

assignment routine that produces vehicle flows, which can be used as inputs to other models (such as a simulation model), by expressing the flows on each link in a data format readable by these other models. Model 2 could be a traffic simulation model, and Model 3 an emissions inventory model. Instead of specific "translations" between Model 1 and Model 2, and Model 1 and Model 3, there would be routines that transform the data between each model and the Data Model.

The next step, once models are interconnected, is knowing when to use them. This is where the expert knowledge is required. Three aspects are particularly important: the model's data requirements, its outputs, and the conditions under which the model is applicable to a particular situation (context). They are important for different reasons. A model's data requirements are needed to search in the Data Base for the appropriate data, and either retrieve it if available, or indicate that is not available. A model's outputs are used to estimate measures of performance directly, or to generate predictions or forecasts used later for measure of performance evaluation. But just because there are data to evaluate a model, and the model's output can be related to the measures of performance, this does not mean that the model is applicable. The model itself is an abstraction of some environment, and it is used to evaluate certain impacts, both of which must be reflected in the Planning Vector. One of the most obvious components of this context is the action itself, though a full characterization of context would include other elements. The next sections refer in more detail to the process of data selection, model selection, model chaining, and model evaluation. The relationship between these and model input, output, and context, will then become evident.

2.2 Model chaining: introduction of the analysis process concept

In the context of transportation planning, there is not a single model that will produce the desired result from existing data. Sometimes, a chain of models has to be applied, where the output of one model is used as the input to the next. An example of this is the Urban Transportation Planning Process (UTPP), where socioeconomic forecast models, trip generation, distribution, mode choice and route choice models are chained to generate estimates of vehicle flows on a highway network. Another example is the forecasting of pollutant emissions due to mobile sources; in this case a battery of models are used to generate future emissions factors, VMT by travel speed, and total trips forecasts, which are subsequently used to produce an emissions inventory. A number of intermediate results have to be obtained, even if they are not of interest. The Analysis Agent has to know that, although there is no single model that will generate vehicle flows from socioeconomic forecasts, for example, a chain of models can be applied to achieve the desired result. One of the most important aspects of the "intelligence" of the Analysis Agent is being able to build and apply these chains.

The collection of methods that specify sequences of models and transformations to be applied in the evaluation of a planning vector will be called the analysis process. An analysis process is composed of all the different methods that could be used to estimate the measures of performance associated with a particular action. In that sense, analysis processes are action-specific; a particular action-measure of performance pair would have an associated unique sequence in that action's analysis process. The sequence is unique in the sense that it contains all the possible ways in which that measure of performance can be estimated. This is not the same as saying that there is only one way of obtaining that estimate; as mentioned before multiple procedures will be dealt with through deliberation.

Analysis processes must be encoded in PLANiTS. Each will be dependent on the models available and the measures of performance to be estimated. An analysis process must begin with a complete specification of the Planning Vector, and end with measure of performance estimates. The process is not a fixed sequence of steps though; at any point the user has the option of redefining the specification of the Planning Vector in order to reduce or expand his/her choice of methods and data. As coded in PLANiTS, a process will consist of all possible ways of estimating each measure of performance, as long as that measure of performance is relevant to the action in question.

An analysis process may be represented using the concept of two-dimensional graphs. One can think of models as nodes, where each node performs a different function. The links between nodes would be data flows. Links would exist connecting models that satisfy the condition that one model's output is another model's input. The graph would have one node of origin, corresponding to the Planning Vector, or more specifically, to the given action to be evaluated. Each branch of the graph would correspond to a complete method for estimating a particular measure of performance. Each method would consist of a chain of models.

A conceptual example of an analysis process graph is shown in figure 2. Notice how the graph has a tree structure, yet is not one due to the parallel branches. Figure 2(b) is a

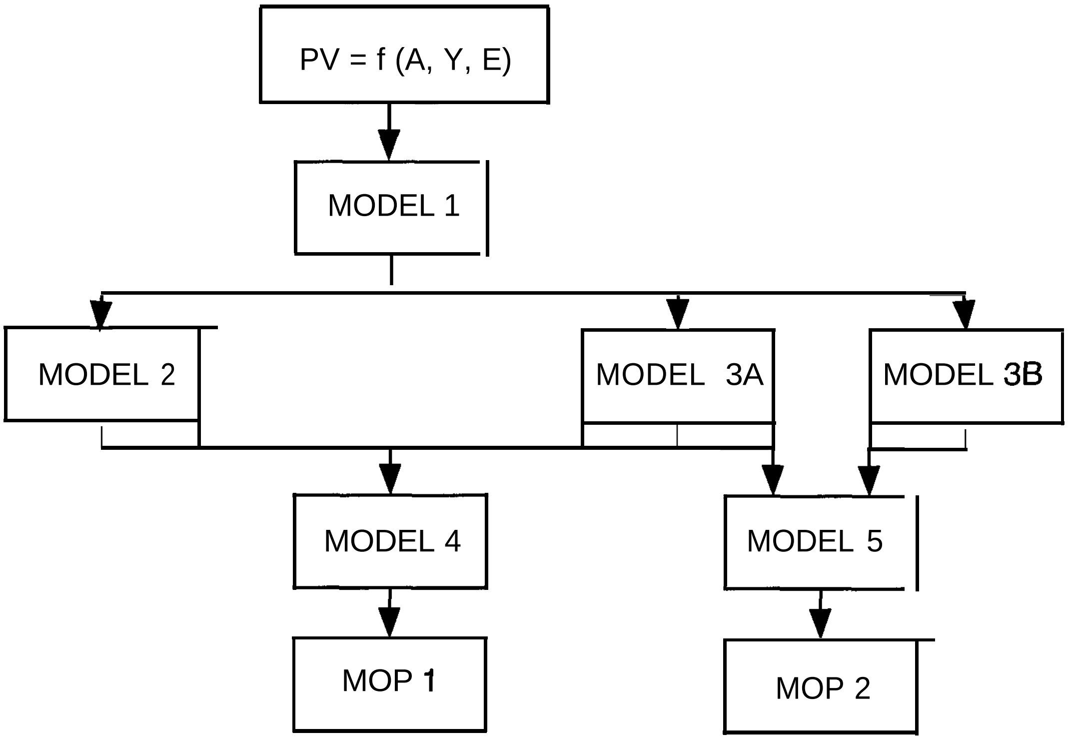


Figure 2 (a)

ANALYSIS PROCESS REPRESENTATION

GRAPH TREE

	A	M1	M2	M3A	M3B	M4	M5	MOP1	MOP2
A	0	1	0	0	0	0	0	0	0
M1	0	0	1	1	1	0	0	0	0
M2	0	0	0	0	0	1	0	0	0
M3A	0	0	0	0	0	1	1	0	0
M3B	0	0	0	0	0	0	1	0	0
M4	0	0	0	0	0	0	0	1	0
M5	0	0	0	0	0	0	0	0	1
MOP1	0	0	0	0	0	0	0	0	0
MOP2	0	0	0	0	0	0	0	0	0

Figure 2 (b)

ANALYSIS PROCESS REPRESENTATION

GRAPH MATRIX

representation of the analysis process depicted in figure 2(a). Nodes (models) in the graph are associated with the rows and columns of the graph matrix, while the links (data flows) are represented by coding a 1 in the cell corresponding to the pair of models connected by the flow. Model pairs which do not share data flows are identified by a code 0 in their corresponding cell. The row corresponding to each measure of performance contains 0 in all its cells. That is because each measure of performance represents the end of any given method in the analysis process.

To understand the function of the graph in explaining an analysis process, it is more convenient to begin at the end, that is, at the measures of performance. Suppose that an action has been specified for analysis, with a set of performance measures to be estimated. The system would select the analysis process corresponding to that particular action, and then look at the first measure of performance. This would locate us at the end of one of the branches. Traversing the tree backwards shows the sequence of models to follow in order to estimate that measure of performance. Whenever a junction is reached, the system decides which model to take (if there is enough information), or the user is prompted for a choice. In this way the system identifies models to use, and their order.

The exact same result would be obtained by using the matrix representation. In this case the starting point would be to look at the column corresponding to the measure of performance, and to search there for a cell coded "1". That cell’s row indicates the model to be applied in order to obtain the measure of performance. Now look at this model’s column, and search there for a cell coded with 1. If there are x "1" in the column, it means that there are x models that can be used to perform that step in the analysis process. That is the case, for example, of models 3A and 3B in figure 2b. Thus, a junction in the analysis process tree would be represented by multiple l's on any given matrix column. As we mentioned while discussing the tree structure, at any given junction either the system or the user has to make a choice. Repeating this process eventually would lead to the action row. Once a complete sequence of models is specified, the Analysis Agent can proceed to evaluate these in the appropriate order.

Notice then that the search is for a feasible path, as opposed to a minimum path, between the action and each measure of performance. Feasibility in this case may be defined in terms of the applicability of each model in the chain and availability of data to evaluate these, as well as satisfaction of other requirements placed upon the estimation process by users.

When a Planning Vector is specified, it is likely that some, or many, of the links in the process will be cancelled. Reasons for this are for example differences in the environment in which the analysis is to be conducted. A case in point is the analysis of the effects of an HOV lane: one could be interested in the effects on the corridor's average speed, in which case a macroscopic operational model would be appropriate, or one could be interested in regional changes in mode and route choice, in which case an urban planning model may be the tool to use. The next section will deal more specifically with model selection within an analysis process.

Due to the fact that there may be multiple ways to analyze a given Planning Vector, deliberation can be introduced explicitly in the analysis methodology. Issues such as modeling intensity, data sources, model reliability or sensitivity to the issues at hand, are opened to be discussed by the participants, in order to reach a agreement, not on the results, but on the methods. If the objective is to make the planning process transparent to everyone involved in it, then the ability to discuss model strengths and weaknesses is as important as building consensus on the results of the analysis. An analysis process is able to identify the alternatives available for evaluation of a given Planning Vector, and thus serves the purpose of deliberation during the estimation process.

Analysis processes may not exist between given action-measure of performance pairs for two reasons: either the action has no effect on the measure of performance, or no method has been devised to estimate effects. The answer that PLANiTS would give on each of these situations is different. In the first case, the system would indicate to the user that the action does not influence the measure of performance. In the second case, the system would indicate that possible effects can not be calculated. The user would always have the option of moving into the Tool Box and building a model that would satisfy his or her needs. In all these processes assistance from the Knowledge Base would be needed, both to identify the cause for the absence of an analysis process, and to suggest courses of action.

Development of this system is still at a preliminary stage. It needs to be tested and refined. One problem that is not clear at this point is how to represent iterative execution between two or more models. Nevertheless, this representation seems promising because it captures the essence of the analysis process, and is relatively simple to translate into computer code.

The process between Planning Vector specification and performance measure estimation can be subdivided in three major tasks. The first task is the selection of an analysis process based on the characteristics of the Planning Vector. The second task is the selection of a method within this analysis process, this time based not only on the Planning Vector, but possibly also on user input. A method could be a single model or a chain of models. The third task is the selection of specific data-items used as model inputs. The tasks are sequential, broadly speaking; they represent an ordered way of searching the Model and Knowledge Bases with the objective of estimating measures of performance. Task execution is not completely linear because whenever the data for a given model are missing, the model would be flagged when the second task is performed. Thus the user may be required to take some action at that point. More details are provided in the following sections.

2.3 Criteria and procedures for model selection

Analysis processes can be defined as action-specific. Selecting an analysis process based on the Planning Vector elements involves examining each of the actions in the Vector, and selecting its corresponding Process. Actions will be classified in groups. For example, an action called “Add HOV Lane” belongs in the group “Roadway Projects”. From within the analysis process for the latter action, the system would extract those sequences corresponding to an HOV lane evaluation. In other words, the type of action selects the first subset of methods in the analysis process.

This subset is further specialized by measure of performance. Only the methods that culminate in the evaluation of the measure of performance at hand are examined by the Analysis Agent. The Action-MOP tree is built from the analysis process graph, and all possible paths leading to that measure of performance from that action are recorded. It is likely that at this point the “tree” would have parallel branches.

Next a series of rules have to be applied to eliminate some of these parallel branches. The criteria for this come from the Planning Vector. Some of these criteria are contained in the Environment specification, while other are specified by the measure of performance dimensions. For example, whether the measure of performance is to be evaluated on the short, medium, or long term basis, determines how far up the chain of UTP models to go (assignment, mode choice, distribution, etc.). Another example: both measure of performance and environment dimensions specify whether the analysis is to be conducted at a corridor, or at a regional level. Some models are appropriate for corridor analysis, while others are used at regional network level. If the environment is defined as a region, but the measure of performance is to be estimated over a corridor, then a sequence of UTP and corridor simulation models should be used. If both environment and measure of performance are defined over a corridor, then perhaps only the simulation models are appropriate.

There will be cases in which PLANiTS cannot choose between models. In those cases the user will be prompted for a decision. Based on that decision, the system would select the upper parts of the tree until another decision point is reached. Again PLANiTS would apply

Planning Vector rules to select a branch. If it fails, then user intervention is requested again.

It is important to note that throughout this process no checks are made for data availability. These checks are done later, when a method of analysis has been selected. The idea behind treating the method and the data issues separately is precisely that they are different issues. The user can always go back and forth between model selection and data selection, until a satisfactory method of analysis is chosen.

We do not want to leave the impression that the only choices for this process are for the computer to construct a unique process by chance, or for the user to make all of the choices whenever parallel paths are present. By a simple modification of the matrix representation discussed above, we can develop a series of "metrics" for evaluating possible analysis paths. Suppose that whenever it were possible to reach one model from another, that instead of putting a 1 in the cell of the matrix that represents this connection, a numerical value were entered that represented, in some sense, the "cost" of traversing that link. Then, if a number of parallel routes were discovered (it would be a trivial task for the computer to compile all possible paths between any two nodes) each could be assigned an overall cost, and selection could then be made on the basis of which were least "expensive". If it were more appropriate in some cases to think of benefits rather than costs, then the matrix could be constructed appropriately, and a search for the maximum path would result. In general, what this requires is that a number of matrices be constructed, each of which composed of the same column and row headings, but each representing a different metric. Some global data would need to be stored which would represent some basic information about each metric; i.e. whether it should be minimized or maximized, what the units are, etc.

There are a number of criteria that are known to be applied in practice when choosing amongst methods or models. In fact, it is reasonable to suppose that more often than not, a combination of criteria might be required, and an aggregate cost constructed to reflect the overall merits of each path. We propose that the user be presented with a list of possible metrics, be allowed to choose as many of these as necessary, and be allowed to define the functional form of an aggregate measure of merit. This functional form should include linear operators (addition and scalar multiplication), exponents, simple monotonic transformations (i.e. logarithms), inverses, and the like.

It should be illustrative to discuss some specific examples of metrics that could be used. For users concerned with time resources, a cost could be an estimate of the computer time required to move from one node to the next in an analysis process. For those most concerned with error propagation, estimates of the errors associated with traversing a link could be used, and these could be aggregated in an additive, multiplicative, or some other, fashion. If data resources were a concern, then the volume of data required to complete a link could be represented as a cost in some numeric fashion. Using the computer's ability to learn from past activities, one could store the number of times a particular link between two nodes was actually chosen for analysis purposes. This way, users could develop a sense of what the most popular methods for a particular type of analysis were. Of course, this type of data would be very precarious at first, until a sufficient number of runs were made to suggest stability. Any number of similar scoring methods could be devised. Another example of the flexibility of the PLANiTS architecture is that these metrics need not be carved in stone when PLANiTS is implemented; rather they can be added “on the fly” and used for evaluating any subsequent analysis choices.

The most complicated facet of this tool would be the collection of the data used to represent scores for any particular metric. Some seem relatively straightforward, for example the computer time estimates or the volume of data. While some flexibility should be built into the computer time estimates to account for different hardware and processor load conditions, there should be little argument as to the final results. Some of the other examples, however, such as the estimates of compounding error, would be much more difficult. It may not always be possible to label a specific method with an estimate of its error, as this error may vary tremendously depending on the conditions under which the method is applied. The inherent variability in a metric such as this should be taken into account when selecting how to employ it as a criterion, either by itself or as part of a compound decision process.

It should be recognized that although considerable efforts could be made to develop robust matrices such as these, that ultimately the decision for which methodology to employ should come exogenously. No part of the PLANiTS philosophy is intended to circumvent the human decision-making process, only to facilitate it via the provision of information and tools not necessarily available otherwise. Thus, it would be necessary to ensure that no “default” selection of analysis paths be made, even under the influence of an expert system. Any information compiled by the computer about the frequency of previous choices of possible paths should be made available to the user on request, but should not be allowed to supercede the availability of all possible metrics.

2.4 Data transformation and aggregation

For the purposes of PLANiTS a model could be any operation that takes a data-input and transforms it into a data-output. Yet we want to make a distinction between a model and what we have called so far data transformations. Functionally, a transformation serves the same purpose as a model and can be treated as one. One can think of transformations as operations performed across object dimensions. Transformations are used for a number of reasons: to change the units of a given data-item, to extrapolate the results from one time period to another, to aggregate results from disaggregate populations, to distribute an aggregate result into population segments, etc.

Transformations can be classified into two broad groups: exact and approximate. An exact transformation for example would be the one used to change Travel Distance from units of veh-mi to veh-km. There are no assumptions involved in going from one unit system to another. Exact transformations are not very interesting for the PLANiTS user, and they will be incorporated as functions of the Data Base, more than elements of the Methods Base. Other types of operations that may fall in this category are those now commonly performed by Geographical Information Systems.

Approximate transformations on the other hand are not so trivial. An approximate transformation would be, for example, to calculate Total Person Distance Traveled instead of Total Vehicle Miles Traveled by multiplying the latter by the average vehicle occupancy, as opposed to computing VMT for each vehicle occupancy category. There is an assumption involved in this calculation, namely, that any vehicle, regardless of its occupancy level, travels on average the same distance per time period. The assumption would be violated if for example, carpools were more likely to be formed when the commute trip was long. Another case of an approximate transformation would be to estimate daily VMT from a peak period estimate. Again, knowing that a larger share of work trips occur during the peak periods, and that these trips are expected to follow patterns different from those of non-work trips, one cannot expect peak period estimates to be representative of daily conditions, even if weighted by their hourly distribution.

Nevertheless, while it is recognized that these calculations are not exact, they are often done for any number of reasons. It may be that the data are not available to evaluate the exact model, or that an exact model itself has not been formulated/estimated; also, sometimes the error in the estimation from a “exact” model is large enough to make its estimates not significantly different from an approximate solution. Often, the cost of an exact solution overrides the gains from increased efficiency.

A specific type of data transformations are aggregation rules. Aggregation rules are important in PLANiTS because, with a limited set of models, they allow for the evaluation of Planning Vectors across multiple dimensions. Data aggregation, in and of itself, is a big issue in transportation. For example, in urban travel demand modeling, methods for aggregation across individuals is still a contended issue.

Aggregation may occur at two places in the analysis: since data are stored at the most disaggregate level, it will usually be necessary to aggregate across one or more dimensions to reach the level used by the model. On the other hand, instead of aggregating input data, the system may be required to aggregate measure of performance estimates. This will happen whenever there are no models that can estimate measures of performance at the desired level of aggregation.

Aggregation of data items for model input is primarily a function of the Data Base, and its rules are imbedded in the Object Oriented Data Structure. The treatment of measures of performance and their aggregation rules lies in a region where the Model Base and the Data Base intersect. At the most elementary level the measures of performance themselves are functions of data-items, and thus subject to the same rules and transformations as the whole data structure. Conceptually , since the measures of performance are the result of analysis estimation, the rules for aggregation should be explicitly linked to the dimensionality of the Planning Vector.

In order to illustrate the ideas above, an example is used. The measure of performance to be estimated is Total Distance Traveled. One of its dimensions is user unit, namely, total distance traveled by vehicles or total distance traveled by people. For the time being the other dimensions will be ignored. Define TDT_{ij} as total distance traveled on link (street) i (i = 1, 2, . . . , L), expressed in user unit j ($j = 1,2 / \{j$: 1 = vehicle, 2 = person}). Suppose that the user requests as the measure of performance TDT_{*2} (read total traveled distance summed across all links, in person units), which is defined as:

$$TDT_{*2} = \sum_{i=1}^{L} \gamma_i \, TDT_{i1} \tag{1}$$

where γ_i is the average vehicle occupancy in link i. Suppose further that the only model in the Methods Base is a regional planning model which can estimate vehicle flows, but not average vehicle occupancies or person flows per link. The Analysis Agent would look for the γ_i in the Data Base. Any of the following scenarios are possible:

(a) The γ_i are currently in the Data Base. They are retrieved and used to evaluate the measure of performance. No transformations or aggregation rules are needed.

(b) Instead Of γ_i, the Data Base contains γ_{ik}, a frequency distribution (percentage) of vehicle occupancies (indexed by k) per link. The Analysis Agent invokes a data structure rule that transforms a discrete distribution into an average value. This rule is a relationship of the following type:

$$\gamma_i = \sum_{k=1}^{K} \gamma_{ik} \, k \tag{2}$$

that is, the average value is obtained by weighting the distribution by the vehicle occupancy. This would typically be an application of a general rule imbedded in the Data Base structure.

(c) The Data Base does not contain vehicle occupancy information at the link level, but it has an average vehicle occupancy at the regional level, γ. In the absence of more information, one may approximate the average vehicle occupancy in link i, γ_i, by the average vehicle occupancy at the regional level, γ. The aggregation rule would be: If γ but not γ_i, then instead of Equation 1 use Equation 3 :

$$TDT_{*2} = \gamma \sum_{i=1}^{L} TDT_{i1} \tag{3}$$

Rules that involve distributional assumptions, such as (3), can be optional. Users should have flexibility in evaluating whether the rule is applicable under specific circumstances.

(d) The Data Base does not contain an average vehicle distribution by link, but it has a distribution by region, γ_k. In this case the Analysis Agent would invoke the rule contained in Equation 2 (dropping the link index), and then apply the aggregation rule of Equation 3.

Suppose now that the regional planning model could distinguish between vehicles by vehicle occupancy; that is, instead of producing total vehicle flow it produces vehicle flow by vehicle occupancy. Then the rules to estimate total travel distance in person units would have to be changed slightly. A new dimension must be used to identify the vehicle occupancy corresponding to each flow estimate. The definition of the measure of performance would now be:

$$TDT_{**2} = \sum_{i=1}^{L} \sum_{k=1}^{K} k \, TDT_{ik1} \tag{4}$$

where it becomes evident that, given the TDT_{ik1}, no aggregation or data rules are needed to estimate TDT_{**2}.

The definition of an approximate transformation can be summarized into two characteristics: they transform model outputs into the exact measures of performance, and they can be substituted whenever exact models become available. Given these two characteristics, it is clear that the right place for these operations is in the Model Base. A transformation can be treated as a model. They can be incorporated initially as part of different analysis processes, and later replaced whenever appropriate by other transformations or by new models.

In the context of these transformations, the transparency of the system must be emphasized. Some applications of approximate procedures may be considered more dangerous than others. In particular, given the aggregate nature of many transportation models, it is sometimes required to use empirical distributions to disaggregate a given result into a range of time periods or users. While these methods may be appropriate in some contexts, they can be the source of misleading results in others. Thus, the user has to be aware of both the limitations of existing models, and of the available methods commonly used to overcome these limitations. Presumably, this awareness will lead to the development of new models whenever the analysis so requires. By letting the user know the assumptions involved in any given method of analysis, PLANiTS gives him/her the choice of either going ahead with the analysis, going back and changing the specification of the Planning Vector, or making new models available to the system.

3. MODEL EXECUTION

Four tasks are involved in model execution:

- retrieving data from databases (including remotely accessible data and results from previous model runs),
- transforming data into the appropriate format and making it available for the model,
- calling the model's executable module, and
- manipulating and storing the model output in order to enrich the database, generate input for a subsequent model, and calculate measures of performance.

The second and fourth tasks will be performed primarily through the object-oriented structure, and through a set of rules that specify the formats and characteristics of each data file for each model in the Model Base. This is by no means an easy task; it depends on the flexibility of the underlying data structure. The focus here is on the first and third tasks mentioned above.

3.1 Data selection (model inputs)

Each model needs a set of inputs, i.e., data-items. The system must know which inputs a given model needs. This means that every single data-item that is accessible to PLANiTS has to be identified as a member of a family: a highway network, a trip table, traffic counts, vehicle occupancy distributions, etc. All model inputs have to be defined in a data dictionary. Some of these are higher level concepts in the data structure. A highway network, for example, is in principle formed by a set of nodes and links, though not any set will work as a highway network.

Given that the concept of highway network can be identified by PLANiTS, then the question is, for a given model, which is the appropriate network to use? The answer to this question can be obtained from the Planning Vector parameters. We use examples to explain how this will work.

To understand how the Analysis Agent searches for data-items, we need to characterize the data-items needed. But first, let us summarize the steps involved in analyzing any given measure of performance.

(a) The user is given the option of evaluating the Planning Vector via modeling, the Case Base Reasoner, or the Expert Base. For the purposes of this discussion, assume that the modeling alternative is selected.

(b) The Analysis Agent examines the Planning Vector (A, Y, E) to be evaluated, and selects an analysis process.

(c) From the analysis process, the sequence of steps to evaluate the measure of performance is reconstructed by working through the graph matrix. Whenever more than one option are available, a choice is made by the system in consultation with the user.

(d) The Analysis Agent uses information about actions, performance measures and the environment, all defined in terms of spatial, temporal, and user dimensions, to select models.

(e) Starting with the first model in the sequence, the Agent looks for the data needed to run that model. It examines the model inputs and searches for the fully identified data in the Data Base. If found, the model is applied and its results are stored (most likely temporarily) to be used in the next model. If not found, the model looks for aggregate (or disaggregate) data that would allow it to estimate the data needed. If none are available, the user is prompted to provide a data source, or to go back to step (c) and redefine the chain of models.

(f) Step (e) is repeated until the measure of performance is evaluated.

(g) Measure of performance aggregation across dimensions is done if necessary.

The following discussion is based on the analysis of an HOV lane operation using regional planning models. Only short term effects (route choice) are discussed. The traffic assignment routine for planning models requires two inputs: a highway network and a trip table. In particular the discussion will be based on MINUTP, which contains an assignment routine modeled after the UROAD program of the UTPS.

3.1. I Model requirements

MINUTP is composed of many modules. The traffic assignment module requires binary network and trip table files. An additional module can generate these files from text records, in which case additional information is required. This information is usually coded in a control file (see section 4.3). The description will include both types of information, since otherwise the example would not be clear.

(a) Highway network: A highway network is composed of two types of records, links and nodes. In addition, it contains a header where the following information is defined: number of links, number of nodes, number of zones, a speed table, a capacity table, and an identification record. Links represent streets; nodes represent link intersections or zone centroids. A link must contain the following information: begin (A) node, end (B) node, length (field may be empty), speed or time, speed/time indicator, speed class indicator, capacity class indicator, number of lanes, two-way indicator, direction indicator, and observed traffic count (field may be empty). The node records are optional. Each consists of a node number and a (x,y) pair of coordinates that locate the node in a plane. These records are used to display the network graphically, and to calculate link lengths. It is likely that PLANiTS will have its own graphics display utility. Link lengths will be obtained from the Data Base. Thus node records will probably not be needed. There are certain rules which must be followed when coding the network. These can be found in the MINUTP Technical User's Manual (199 1).

(b) Trip tables: The assignment module may load many tables simultaneously. Each table is composed of trip records. A trip record is defined by a zone of origin, a zone of destination, a table indicator, and the number of trips between the origin and the destination. There must be a one-to-one correspondence between the zones in the trip table and the zone centroids in the highway network. Not all (O,D) pairs need have trips. Records are stored into tables according to the table indicator number.

3. 1. 2 Input data as objects in the dictionary

Both network and trip table are keywords that should be contained in the object dictionary.

(a) Highway network: A network is composed of multiple objects. These inherit attributes from three classes: streets, intersections, and zones. Each of these classes has a set of attributes. In principle, by inheriting from the street class, for example, all links in the network will have the same attributes as the class. We know from above that ultimately the model needs only a few of these attributes. Some will go directly into link characteristics, while others are needed to select links, and to set other model parameters. Besides the characteristics inherited from these object classes, a network has other attributes. These are mode, time unit (AM peak, PM peak, etc.), year of analysis.

(b) Trip Table: a trip table is a collection of trips. Trip is also an object in the dictionary. Both trips and trip tables have mode, purpose, time unit, and year of analysis attributes.

3.1.3 Data selection conditions

The data selection procedure consists of extracting from the Data Base streets, intersections, zones, and trips, as well as their attributes. These in turn are used to build the two input files described above. This basically involves answering the questions:

- which streets (intersections, zones) to extract?
- which street (intersection, zone) attributes to keep?

The information needed to answer the questions above resides in the Planning Vector. Specifically, the Environment dimensions (geographical area, time elements, user characteristics) have an almost one-to-one correspondence with the network and trip table attributes. Tables 1 and 2 summarize the correspondence between the model requirements, the object structure, and the Planning Vector dimensions. One can look at the process of selecting data as a two step process: first to apply a series of filters on the Data Base until the exact set of elements desired is left, and then to choose a subset of attributes for these elements. Table 1 describes the first step, while table 2 describes the second. Other information needed to actually build a highway network file will be given in section 3.2.

Suppose that PLANiTS is connected to an extensive Data Base, which contains disaggregate detailed information of the transportation system and peoples' activities. The first filter is to define the boundaries of the region to be analyzed. Once these boundaries are defined, then through inheritance, all the elements that lie inside these boundaries could potentially be selected as part of the analysis. Subsequent filters would have to specialize the objects selected. In the case of zones, the second filter defines the zoning system. In the case of links, two more filters are needed: one to identify the link elements selected as streets, i.e., part of an highway network; the other to define which streets will be selected based on their functionality. For that purpose, streets have an attribute that identifies them as part of a freeway, an arterial, a neighborhood street, etc.

In table 1 no selection criteria are given for nodes. Nodes represent either intersections, changes of direction, or zone centroids. Thus, once the set of zones and links to be used is completely defined, a set of nodes can be derived from these. Consider figure 3, which represents a detailed street network, and suppose that the Planning Vector Environment specification were such that links represented by dashed lines will not be selected. First a node is located at each zone's centroid (figure 3b). Then all nodes which do not belong on links of the types selected before are dropped (figure 3c). Finally, for nodes that connect only two links, those that do not represent changes of direction, or whose link attributes are equal, are also dropped. The remaining nodes (figure 3d) are the ones that will go into the highway network.

There are a number of other rules that have to be observed. Some of these are general, while others are specific to the traffic assignment software. An obvious one is that zone centroids have to be joined by an imaginary link to the network. Others include a series of checks to ensure that the network does not have loose ends. Throughout this process, because of the nature of the data structure used, one can always trace back each zone, link, and node, to its corresponding region, street(s), or intersections. This feature will be used next to derive the characteristics of each of these elements.

Table 2 depicts the relationship between MINUTP's and PLANiTS' data structure. In particular, some of the attributes have a direct correspondence among the data structures, while others must be derived through data transformations and rules. Furthermore, the transformations can be either a function of other objects' attributes, as in the case of capacity; a function of the network structure itself, as in the case of the reverse indicator; or a completely arbitrary user function, as in the case of the speed and capacity classes.

The final step in data selection corresponds to actually going into the Data Base and searching for data to fill up the data structure. Transportation data has not only spatial dimensions, but also temporal and user ones. Different years and different time units have to be

Table 1

Criteria to use for selecting transportation objects

Object	**Environment Dimension**	**Selection result**
Zones	Geographical Area	Defines the size of the region of analysis. Possible options include a set of any of the following: TAZs, blocks, census tracts, cities, counties, states.
	Area unit of analysis	Defines the geographical unit of analysis. Possible options are: user defined TAZ, block, tract.
Links	Geographical Area	Defines the size of the region of analysis.
	Network mode	Defines the type of elements that will make up the links. For a highway network, the network mode is auto, which will point to street objects.
	Link type	Defines a subset of streets in the region which will be processed. Streets are selected according to their function (freeway, major arterial, minor arterial, etc).
Nodes		Node selection is a function of zone and link selection.

Table 2

Attribute inheritance structure

MINUTP data structure	**PLANiTS data structure**	**PLANiTS data transformations**
Link	Street	
A node	A node	
B node	B node	
Length	Length	
Number of lanes	Lanes	Add number of lanes on street
Speed class	Free flow speed	Speed class number is equivalent to free flow speed in mph, or User defined
Capacity class	Lane capacity	Add capacity of all lanes on street Capacity class number is equivalent to capacity/50 in vph, or User defined
Speed/Time indicator		If link is a street, then 0. If link is a centroid connector, then 1.
Speed/Time	Free flow speed	If link is a street, then ff speed. If link is a centroid connector, then 0.

Direction		User function
Reverse indicator		If B-A link exist and: - all fields above equal, then 2 - any fields above different, then 1 If B-A doesn't exist, then 1
Centroid (node)	Zone	
Zone number	Zone ID	Consecutive numbering from 1 to no. of zones
X coordinate		Zone centroid's X coordinate
Y coordinate		Zone centroid's Y coordinate
Node	Intersection	
Node number	Node ID	Lowest node number must be higher than total number of zones
X coordinate	X coordinate	Min is 1, Max is 32767
Y coordinate	Y coordinate	Min is 1, Max is 32767
Trip record	Inter/Intra Zonal Activity	
Origin zone	Origin zone	
Destination zone	Destination zone	
Number of trips		Add all trips with similar Origin and Destination zones
Trip type	Activity mode/purpose	Assign a trip type for each mode/purpose pair

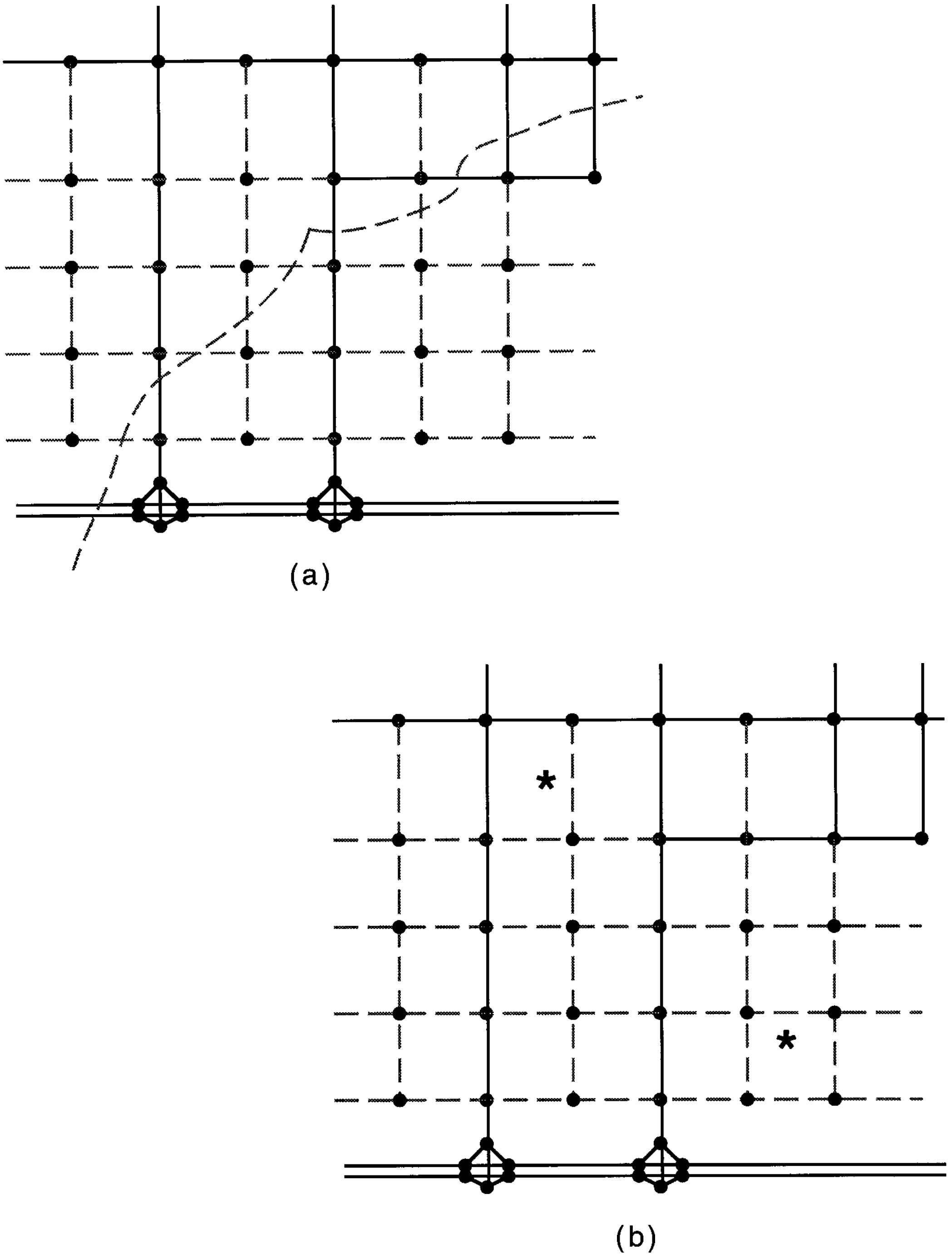


Figure 3 (a)
NODE SELECTION SEQUENCE

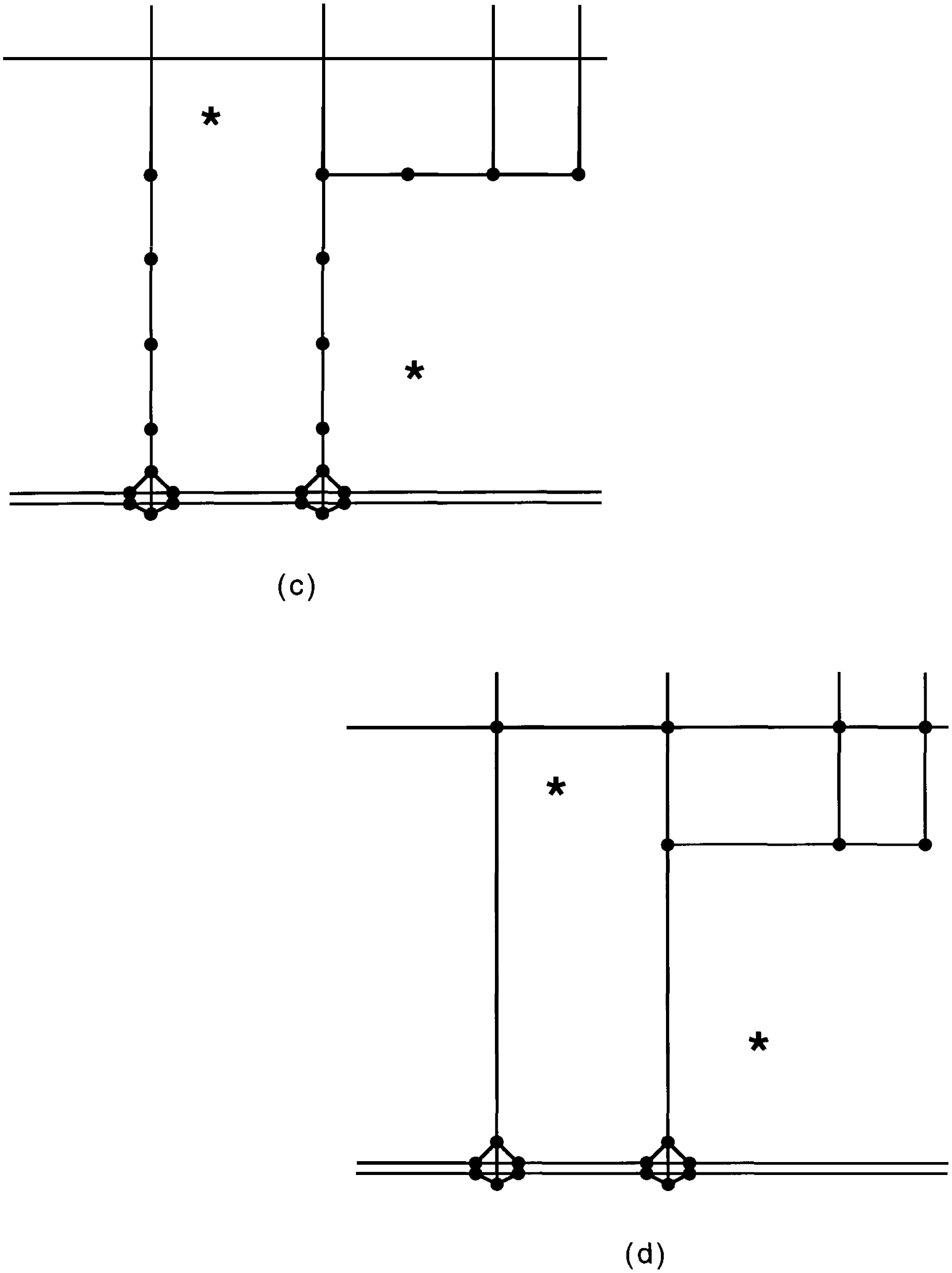


Figure 3 (b)
NODE SELECTION SEQUENCE

selected based again on the Planning Vector's environment specification. Of all the data described above, only the trips, speed, and time attributes have temporal dimensions.

The Data Base may not contain data at the required level of disaggregation. In such cases the system has the option of aggregating data from a more disaggregate level (if available), or disaggregating from a more aggregate level (if applicable). If no data are available, the user is prompted for a data file, or some other source of information. The problem of disaggregating data is not trivial. While the system may suggest ways of doing it, it would be primarily the user's responsibility to ensure that the resulting distributions are not biased. This is a situation in which expert knowledge, as well as consensus among the participants, could help in overcoming the data shortcomings.

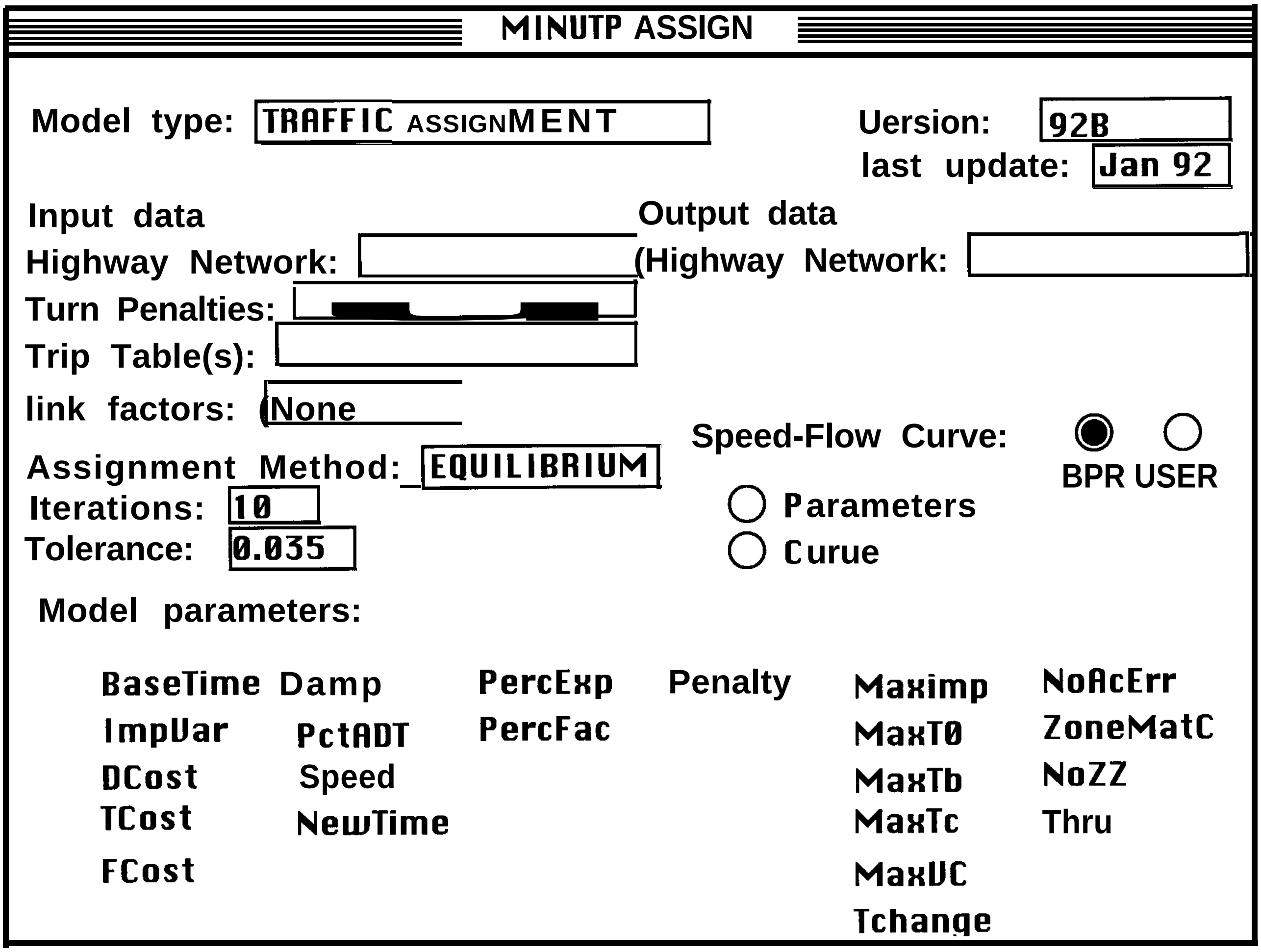
3.2 Control routines

Executing a model usually involves more than just calling an executable file. Most transportation models perform a series of complex operations, where the user is often required to specify exactly which operations are to be performed, and how. An example is used to illustrate this point.

To analyze the effects of adding capacity to a regional network (in the form of an HOV lane, for example), traffic assignment is used to "load" the trips in the network and derive traffic flows and speeds on all network links. Traffic assignment can be performed in a number of different ways. Options exist on the travel impedance to be used as travel cost, on the speed-flow relationship used to represent the effect of traffic on average vehicle speed, *on* the algorithm used to build and select minimum paths and assign trips, among many others. Most of these options are not a function of the Planning Vector. In fact, they represent some of the differences in method that individual planners could deliberate about. In order to make an informed decision, knowledge about the model itself and about the conditions in which it is being applied are necessary. Figure 4 shows a sample user interface window for the assignment routine of a regional planning model.

Besides the options related to the model itself, sometimes data characteristics need also be specified in the control file. Furthermore, some of the model options may be a function of the data on which the model will be executed. Examples of this can also be drawn from the traffic assignment routine:

(a) Vehicle capacities coded in the network usually represent maximum flows in vehicles per hour. If the trip tables represent trips over a day, and the analysis desired is for a peak hour period, then there are two options for scaling the trips. One is to directly apply a transformation to the trip table, and change its time unit dimension from daily to peak hour. The other is to specify in the control file the peaking factor to be used in the analysis.


MINUTP ASSIGN

Model type: TRAFFIC ASSIGNMENT

Uersion: 92B

last update: Jan 92

Input data

Highway Network:

Turn Penalties:

Trip Table(s):

link factors: (None

Assignment Method: EQUILIBRIUM

Iterations: 10

Tolerance: 0.035

Output data

(Highway Network:

Speed-Flow Curve: ● BPR ○ USER

○ Parameters

○ Curue

Model parameters:

BaseTime
ImpUar
DCost
TCost
FCost
Damp
PctADT
Speed
NewTime
PercExp
PercFac
Penalty
Maximp
MaxT0
MaxTb
MaxTc
MaxUC
Tchange
NoAcErr
ZoneMatC
NoZZ
Thru

Figure 4

SAMPLE USER INTERFACE WINDOW FOR A MODEL

(b) Trip tables are usually expressed in person units, but for traffic assignment, they must be converted to vehicle units. Again, this could be done via a data transformation on the trip table itself, or via the control file, by specifying the average vehicle occupancy for each table.

4. EXAMPLE: THE PLANiTS PROTOTYPE (VERSION 1)

The prototype supports the analysis of two types of transportation projects on urban freeways: high occupancy vehicle (HOV) lanes and advanced transportation management and information systems (ATMIS). The elements of the Model Base described are illustrated using both the HOV lane analysis and the ATMIS analysis.

4.1 HOV Lane example

The analysis consists of evaluating the effects of adding an HOV lane on a congested urban freeway. The measure of performance to be evaluated is Travel Delay. Currently, the Model Base contains two models: an urban transportation planning model (MINUTP), and a freeway traffic simulation model (FREQ). We ignore any changes in trip distribution or mode shift that are likely to occur due to the HOV lane implementation.

Figure 5 shows an analysis process for estimating travel delays. Figures 6 to 9 show the individual methods contained in this analysis process, together with a description of the data flows between models.

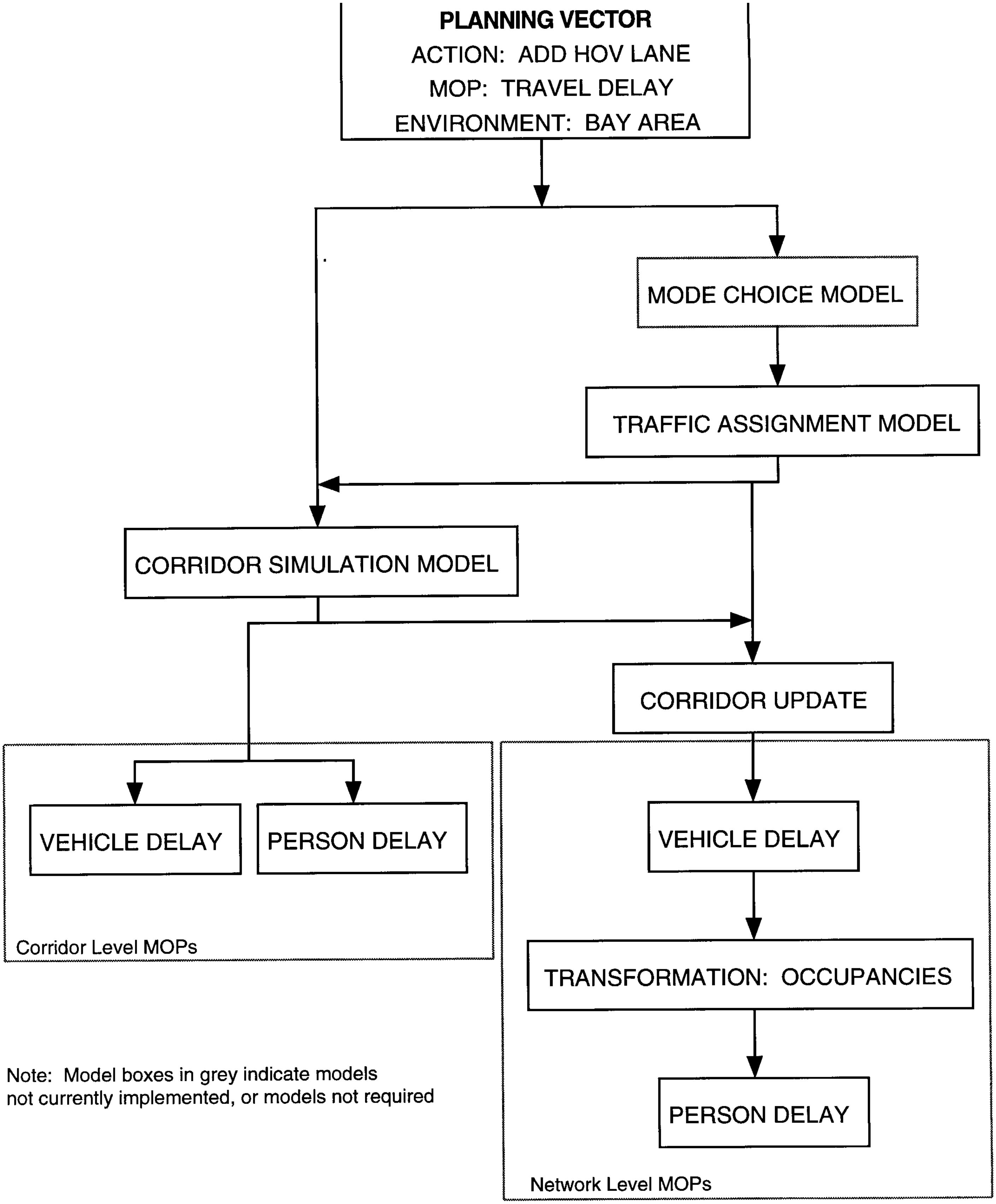


Figure 5

ANALYSIS PROCESS REPRESENTATION
ADD HOV LANE EVALUATION

There are four possible methods for getting the desired result, and each of them uses one or both of the models available. The first method (figure 6) models the HOV lane in the regional network, using MINUTP's multi-user assignment feature. Method 2 (figure 7) uses the freeway simulation model to obtain estimates of travel speed, flow, etc., both in the priority and non-priority lanes. Method 3 (figure 8) also uses the freeway simulation model to analyze the corridor, but instead of reading flow data from the database, it gets estimates of Origin/Destination flows from the regional network tile. Method 4 (figure 9) performs an additional step, by using the corridor speeds from the simulation model to update speeds estimated with the regional highway traffic assignment model. These methods are not exhaustive.

The Analysis Agent’s task in this case is to choose one of these methods to estimate travel delays according to the Planning Vector specification. Simple rules can be used to exclude one or two methods. For example, if travel delays are to be estimated over the whole region, then methods 2 and 3 cannot be used because they only provide information about the corridor. Suppose now that travel delays are to be estimated over the corridor only. It seems most appropriate to use the simulation model, but the regional model could also be used. Under the latter scenario, none of the methods can be ruled out, but if PLANiTS were asked by the user to choose a default, it would choose the simulation model over the regional one.

In most cases though, the user will be prompted by the system to choose a method for modeling the corridor. Either one of methods 1 and 4 can be used when the travel delays are

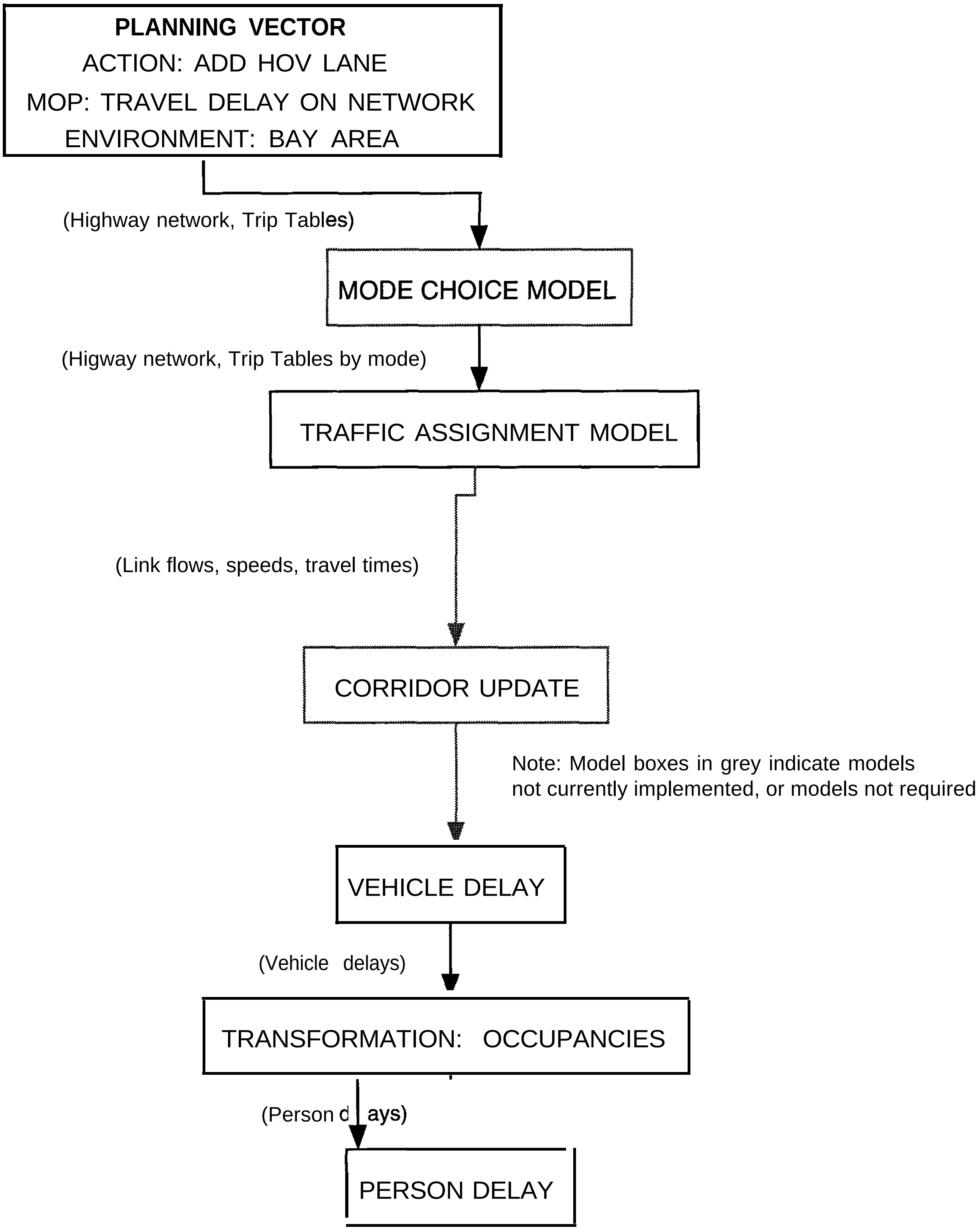


Figure 6

ADD HOV LANE EVALUATION
METHOD 1: MINUTP ONLY

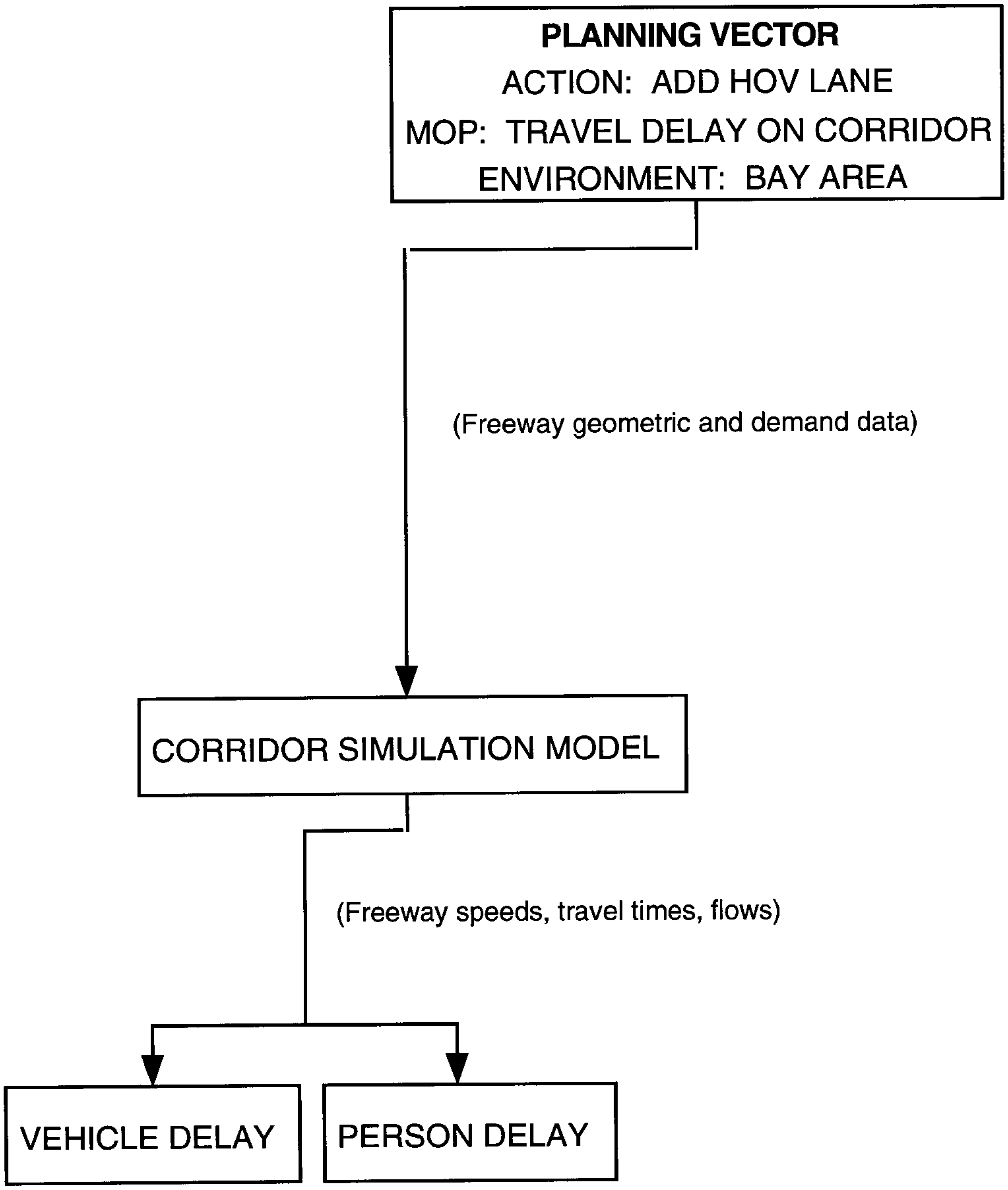
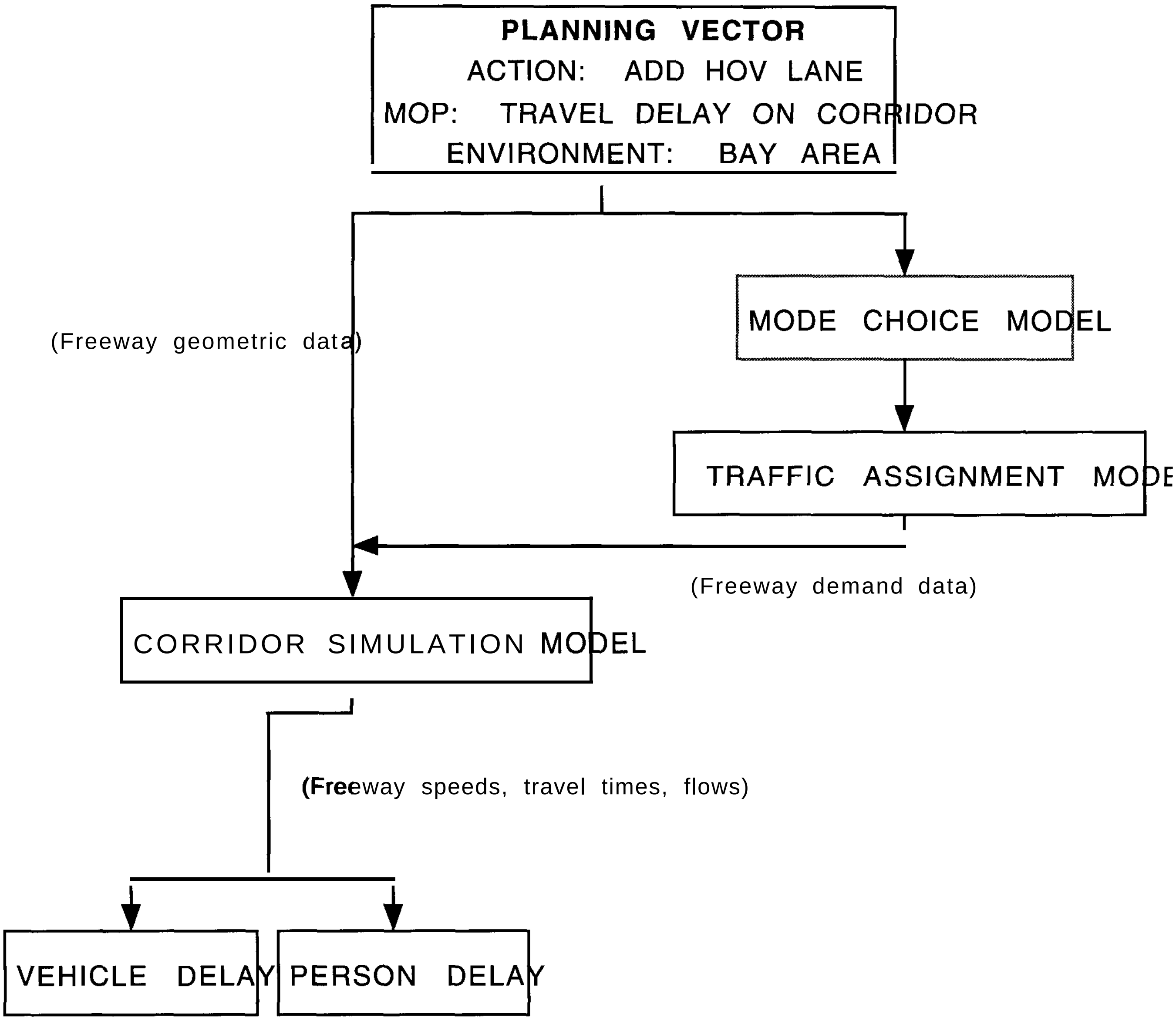


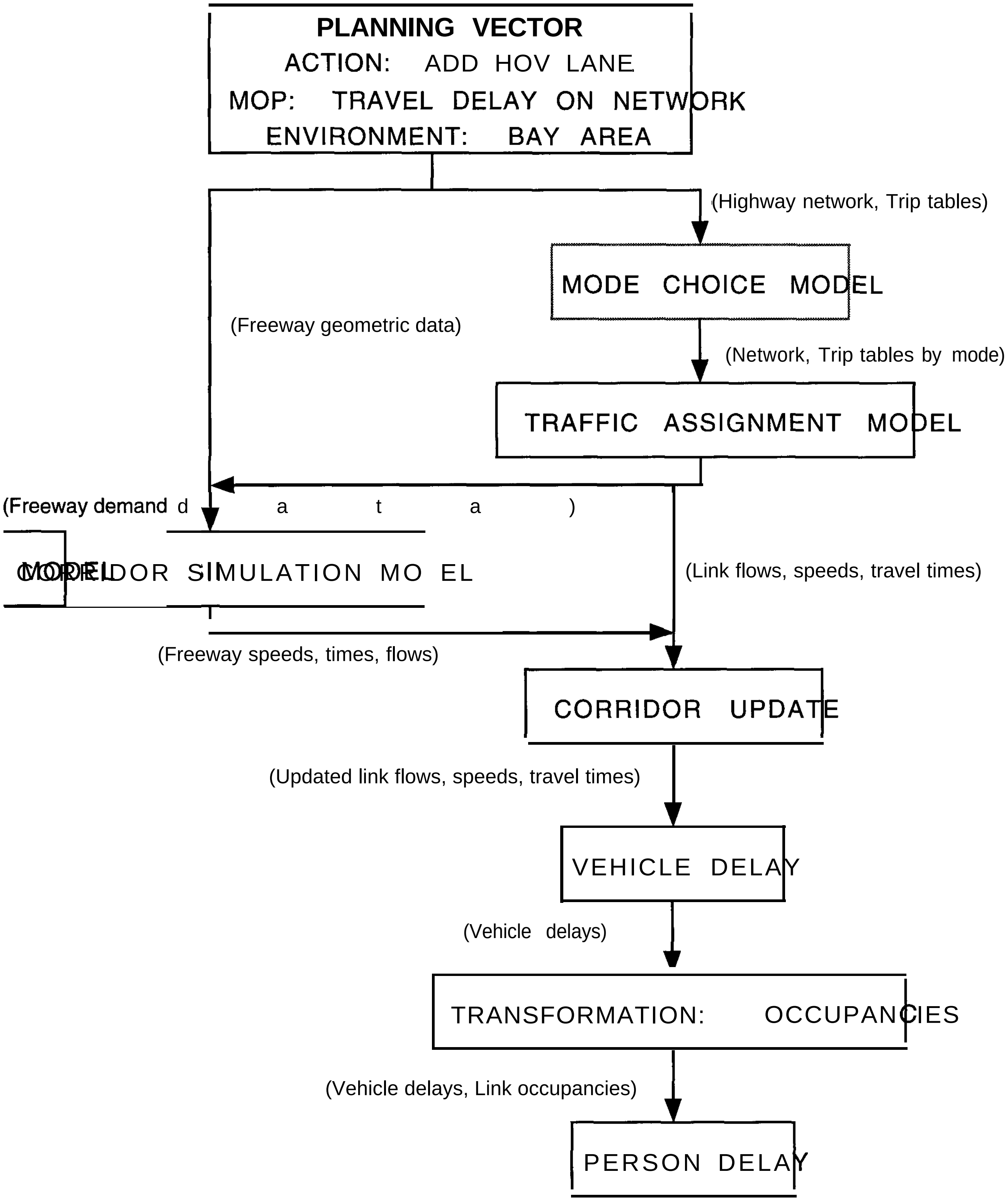
Figure 7
ADD HOV LANE EVALUATION
METHOD 2: FREQ10 ONLY



Note: Model boxes in grey indicate models
not currently implemented, or models not required

Figure 8

ADD HOV LANE EVALUATION
METHOD 3: FROM MINUTP TO FREQ



Note: Model boxes in grey indicate models
not currently implemented, or models not required

Figure 9

ADD HOV LANE EVALUATION

METHOD 4: FROM MINUTP TO FREQ TO MINUTP

estimated over the region. Any of the methods can be used to estimate corridor-specific travel delays. The choice of method should not just be based on whether the estimation can actually be performed. The simulation model uses queuing and weaving analysis to account for congestion, while the regional model does not.

Finally, if additional information were available about the statistical precision of the models, or their time requirements, etc., then this could be used to further assist the user in making a choice. For this example, because only a few options exist and concrete reasons for choosing or not choosing them can be seen clearly, this additional facility seems to be of little use. However, the need for this type of tool would grow as the number and types of analysis methods increase.

For this application trip matrices by mode and purpose were available at the daily level only. Representative matrices for the morning peak hour were estimated using highway peaking factors. Also, the trips were converted from person trips to vehicle trips using average vehicle occupancies for drive alone, carp001 2, and carp001 3+. These transformations were performed before executing the model, instead of using the control file options.

Other transformations are done to aggregate measure of performance dimensions. An example of one of these is to calculate person travel delay given the traffic flows on the regional network. The flows are given in vehicles per hour; there is no way to tell from the program output the proportion of the link flow that corresponds to a given vehicle occupancy. Obviously, then, average occupancies per link, link group, region, or the like have to be used to convert vehicle counts into person counts. The prototype uses a regional average vehicle occupancy for this purpose. An assignment routine that keeps track of the different types of vehicles loaded in the network would not need this kind of transformation. Even then, results would have to be aggregated at some occupancy level, which is currently at three or more persons per vehicle.

One could extend this analysis beyond these two models. Consider for example the case of medium to long term effects. The simulation model estimates modal shifts and travel time savings. A full regional model could have a behavioral mode choice model, as well as a traffic assignment routine that takes into account a much larger choice set. Under these conditions, even if the corridor analysis is done via simulation, OD traffic demands are likely to be generated with the regional model.

4.2 ATMIS analysis example

The analysis consists of how to evaluate the effects of implementing a driver information system that has the capability of detecting accidents or other incidents generating non-recurring congestion. Those drivers that were far enough upstream as to still have an alternate route choice would be warned of the prevailing traffic conditions on both the primary and alternate route, and would then have a choice of taking an alternate route. The measure of performance to be evaluated is Travel Delay. It was convenient for the purposes of the prototype to choose two alternatives (HOV and ATMIS) that could be compared across an identical measure of performance, to allow direct comparisons of the actions and facilitate deliberation. The Model

Base currently contains a model called COMBEHQ, developed by Khattak, Al-Deek, and Thananjeyan (1994), which simulates diversion for a specific incident type and duration.

Figure 10 shows a flowchart for the analysis process for the ATMIS action. The information required from the Data Base consists of physical information, traffic information, and incident information. The physical information is road geometry, and consists of segment length, number of lanes, and the presence or absence of shoulders. The traffic information consists of flows, and free-flow speeds, both on the main route and the parallel arterial. The incident information consists of an accident rate, and a Z-factor used to convert an accident rate to an incident rate (Epps, Cheng, and May; 1994).

Using incident trees developed by Epps, Cheng, and May (1994), this information can be used to generate a distribution of accident types, severities, and durations. These incidents are simulated, one at a time, by COMBEHQ, and the predicted delays are summed to generate an aggregate measure of delay savings. This can be used to determine the Travel Delay after implementation. The same transformations can be done to aggregate measure of performance dimensions as discussed in the HOV lane example. This example is made much simpler because at present only one methodology is included in the prototype to address this specific question.

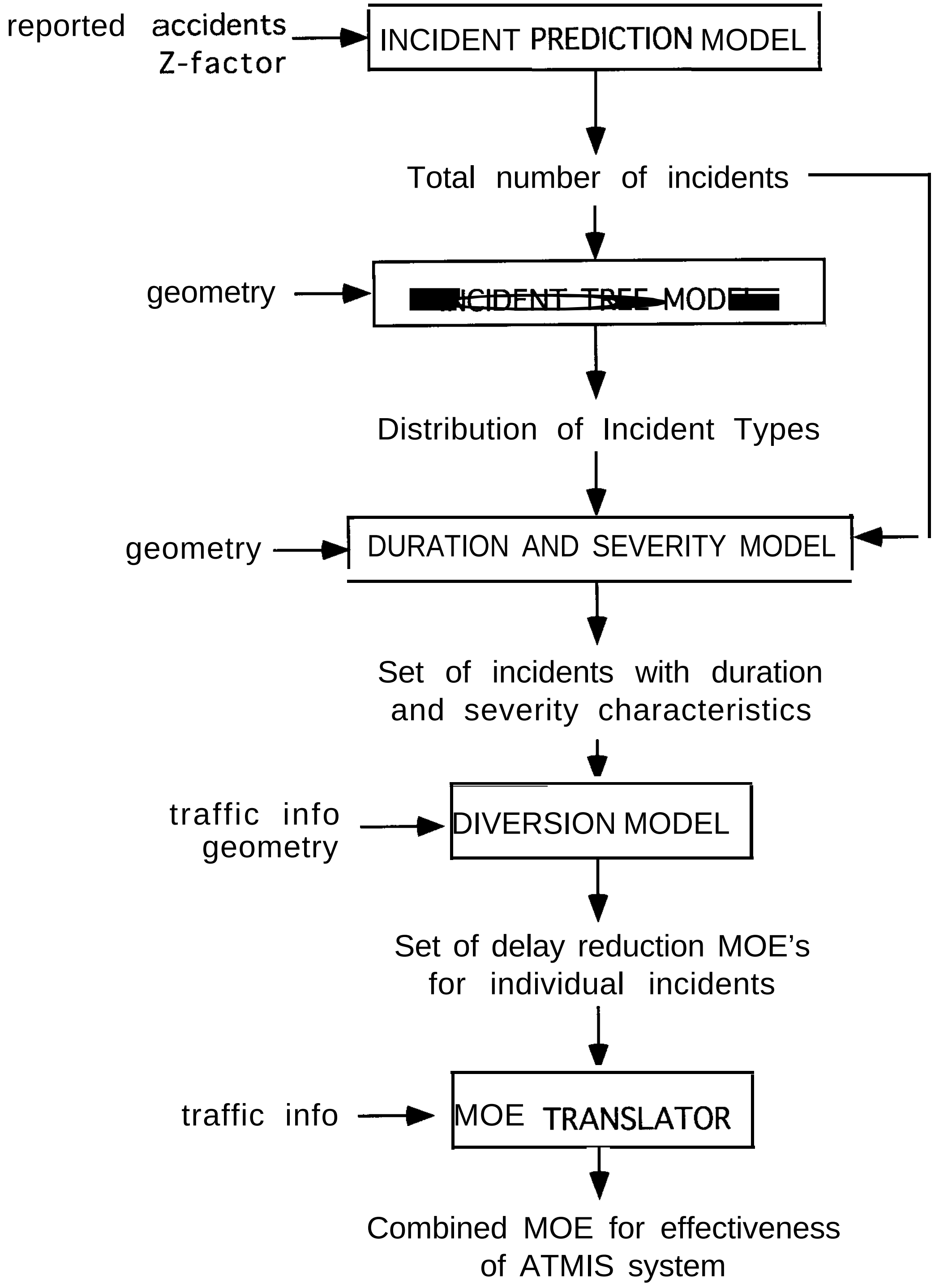


Figure 10

ATMIS EVALUATION FLOWCHART

5. CLOSURE

This paper has layed-out some important features of the Methods Base. Presently the main challenge is to incorporate these features into a working version of PLANiTS. To do so, it is imperative that the underlying data structure and database structure be developed. We believe that a few models, such as the ones explored here, are sufficient to, not just illustrate the concepts, but build a PLANiTS application able to solve real transportation planning problems. Emphasis should be given to identifying model sequences and links, and to developing aggregation rules that support a wide range of spatial, temporal, and user dimensions. To that effect, the procedures and models developed in the last 30 years around the UTPS seem ideal as a starting point for the Methods Base, given the range of applications and the compatibility of models.

REFERENCES

Cheng, J.C., and A.D. May. 1993. *Developing a Methodology for Quantifying Non-Recurring Freeway Congestion Delay*. University of California at Berkeley, California: Institute of Transportation Studies. Working Paper UCB-ITS-WP-93-15.

Epps, A., J.C. Cheng, and A.D. May. 1994. *Developing Methodologies for Quantifying Freeway Congestion Delay*. University of California at Berkeley, California: Institute of Transportation Studies. Research Report UCB-ITS-RR-94-2.

Kanafani, A., A. Khattak, M. Crotty, and J. Dahlgreen. 1993. *A Planning Methodology for Intelligent Urban Transportation Systems.* University of California at Berkeley, California: Partners for Advanced Transit and Highways (PATH), Institute of Transportation Studies. Research Report UCB-ITS-PRR-93- 14.

Kanafani, A., A. Khattak, and J. Dahlgreen. 1994. A Planning Methodology for Intelligent Urban Transportation Systems. *Transportation Research C*, Vol. 2, No.4.

Khattak, A., H. Al-Deek, and P. Thananjeyan. 1993. *A Combined Traveler Behavior and System Performance Model with ATIS.*

Vlahos, N., A. Khattak, A. Kanafani, and M. Manheim. 1994. The role of Teamwork in a Planning Methodology for Intelligent Transportation Systems. *Transportation Research C,* Vol. *2, No. 4.*

Minutp Technical User's Manual. 199 1. Maryland: Comsis Corporation.