The Study of Human Cognitive Processing and Planning in the Solution of Geographic Problems using Artificial Intelligence Techniques

by
Joseph Georges Paul Leroux
B.Sc., Royal Roads Military College, 1980

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in the Department

of Electrical and Computer Engineering

We accept this thesis as conforming
to the required standard

[redacted]

Dr. Kin F. Li, Supervisor
(Department of Electrical and Computer Engineering)

[redacted]

Dr. Nikitas J. Dimopoulos
(Department of Electrical and Computer Engineering)

[redacted]

Dr. Maarten van Emden
(Department of Computer Science)

[redacted]

Dr. Zuomin Dong
(Department of Mechanical Engineering)

University of Victoria

ISBN 0-315-62390-X

Supervisor: Dr. Kin F. Li

ABSTRACT

The objectives of this research are to explore the mechanisms involved in geographic problem solving, the representations necessary to support such reasoning, and to build and test a model of human geographic planning using AI programming techniques. The blackboard paradigm is used to implement the artificial intelligence planner because of its opportunistic nature, one which closely resembles the characteristics of human planning.

The tourist information planning domain is a practical example of geographic planning. The system described in this research was designed to assist travel counsellors in a human-like fashion. A prototype was constructed to demonstrate the useful planning capabilities of this approach.

Examiners:

[redacted]

Dr. Kin F. Li, Supervisor
(Department of Electrical and Computer Engineering)

[redacted]

Dr. Nikitas J. Dimopoulos
(Department of Electrical and Computer Engineering)

[redacted]

Dr. Maarten van Emden
(Department of Computer Science)

[redacted]

Dr. Zuomin Dong
(Department of Mechanical Engineering)

Table of Contents

List of Tables

List of Figures

Acknowledgements

This thesis is a collaboration of the efforts of many people. The foremost is Dr. Kin Li, my supervisor, colleague, friend and patient mentor. His dedication and assistance are primarily responsible for the successful completion of this work. I thank him deeply.

To my fellow graduate students I also offer my thanks, for their assistance in everything from operating systems to word processors. I'd especially like to thank Ed Miska for his patient perusal of this text and Roger Kelly for his cooperation and patience with my questions, and for his wisdom to back up this system as my deadlines approached.

I owe a great deal of thanks to my family for their support during these past two years. My mother, the best cook in town, my father, the insightful expert travelling salesman on whom much of the original work in this thesis is based, and most importantly my devoted wife Lynn who patiently tolerated my excesses, and who herself worked long hours at the computer to assist with the testing of the system and the production of this document.

My final acknowledgement is to the Department of National Defence, Canada, and my sponsor, the Direcorate of Land Armaments Electronic Engineering and Maintenance, for the generous financial support I received during this research.

To my family

Chapter 1 Introduction and Research Theme

1.1 Introduction

There are many interesting areas of research in the field of Artificial Intelligence (AI). Human cognition is a fascinating subject to study, since researching human thought processes may lead to a more thorough understanding of human behaviour. AI researchers must select and focus on a specific aspect of this field to obtain productive results.

Planning was chosen as the activity to be explored in this research because it is common, complex and varied. Planning may take various forms. These include familiar plans such as daily work routines, weekly schedules, and vacation itineraries. Additionally, complex activities, such as major international sporting events, airline schedules, and parcel deliveries, push human and computer assisted planners to their limits. Plans may be required to coordinate the design or construction of physical items, to guide the management of many cooperating organizations in complex engineering projects, and to solve spatial problems based on geographic locations, routes and schedules.

Planning may be defined as the predetermination of a course of action designed to accomplish a goal [Haye 79a]. A goal is the requirement which must be satisfied by a plan. For example, the top level goal of a construction plan is to complete the building on time and within cost constraints. Intermediate construction milestones such as foundation pouring, elevator installation, etc., represent sub-goals in the overall plan. Planning is also problem solving since a plan represents a solution to a problem. In the construction analogy, the problem is to build the structure, the problem solution is the construction plan. This thesis will use the terms planning and problem solving interchangeably. Planning may also include the monitoring of plan execution. This work will concentrate on plan formulation, leaving plan monitoring for future research.

Chapter 1 describes the general planning problem. It defines the approach used to build a model of the problem solving process, and outlines the human planning research work of Frederick and Barbara Hayes-Roth. This chapter also describes one research field, geographic planning, which has been chosen as the practical focus for this thesis. An example geographic problem is outlined along with a general planning model.

1.2 Geographic Problem Solving

One aspect of human behaviour which is very interesting to study is our ability to learn about the environment and use this knowledge to move about and locate sites and objects. This knowledge consists of an understanding of the spatial relationships between objects and their locations, and between locations themselves [Davi 86]. This intelligent behaviour is referred to as geographic problem solving.

Geographic problem solving describes the mechanism used to solve a problem which is based on geographic information. This can also be described as achieving a geographic goal. It includes selecting sites at which to accomplish given sub-goals, and choosing routes to connect these sub-goals. The geographic goal is to construct a plan which solves the problem by satisfying all the sub-goals. A sub-goal can be satisfied by completing the task by visiting a site.

In everyday life humans frequently construct geographic plans. Almost any task of a spatial nature may be considered geographic planning. An example is route planning along roads within cities and towns to perform errands. Locating a book in a library, finding an item in a grocery store, planning a route to mow the lawn, or seeking the best site to fish a particular stream are also geographic problems.

Solving geographic problems requires considerable intelligence. A sketch or map must be interpreted, or a mental map recalled, in order to locate sites at which goals or sub-goals may be accomplished. Activities must be correctly sequenced to satisfy an overall objective. Constraints imposed by geographic and temporal issues must be satisfied. These constraints include the method of travel, the time available, and possible routes and their characteristics. Other more abstract constraints include a wish to avoid getting lost by following the most direct route, or a desire to remain close to the water on a scenic path.

This thesis explores the mechanisms involved in geographic problem solving, the representations necessary to support such reasoning, and builds a model of human geographic planning which has been implemented using AI programming techniques.

1.3 AI Schools of Thought

Every AI researcher must address one of the central issues in this field; should the system under construction be intended to 'think' or merely represent a model which is useful in the study of human cognition? John Searle [Sear 90] divides these schools of thought into 'strong AI' and 'weak AI'.

Strong AI researchers attempt to construct systems which will pass the Turing test [Turi

50]. Their objective is to simulate human cognition to the extent that an expert will not be able to distinguish the performance of the system from that of a human. Supporters of this school of thought would consider such a system intelligent.

Weak AI researchers approach this issue more pragmatically. Their systems are designed to be useful in the study of human cognition. They are not designed to 'think' in human terms, but to replicate human cognitive activity. A parallel may be made between AI cognitive research into computer simulations of human thinking and oceanographers consulting numerical models to understand current flow. The oceanographic model does not attempt to replicate the real ocean, and the AI computer model should not attempt to reproduce the human mind. This thesis will approach AI from a 'weak AI' perspective.

A software program may be viewed as a black box capable of certain transformations of input to output. A black box that replicates human-like outputs with software techniques which are not based on human cognition is not implementing human-like processing and will not add to the understanding of human thinking. The internal mechanisms of the software should also attempt to represent human cognition as accurately as possible [Shar 86]. The theme of this work is to model the computer system processes and data structures after human techniques.

A significant problem for all AI researchers arises from the attempt to model human cognitive activity. The goal is to define and understand how humans represent knowledge and solve problems. A model of human cognition is required before an AI implementation can be built.

1.4 Human Planning

Research by Hayes-Roth [Haye 79a] [Haye 80] describes one model of human planning. Their work characterizes human planning as opportunistic. At any point in the planning process, the human planner has many different decision choices to make in order to further plan development. The selection of which option to pursue is opportunistic; the planner pursues the option which appears to be the best choice given the current circumstances.

Humans encounter both routine and complex problems. Basic problems are solved using a successive refinement of steps, much like Sacerdoti's [Sace 74] hierarchical planner. More difficult problems require opportunistic decision making as the planner searches a less well understood and more extensive search space.

An opportunistic planner will focus attention on different levels of the plan as more details of the solution are exposed. A high level decision may illuminate a detailed activity to be incorporated into the plan. Alternatively, completing the details of an abstract plan may suggest other high level plans.

1.4.1 Hayes-Roth Blackboard

The general model described in [Haye 79a] to implement human planning is a collection of independent specialists, each of which are capable of making decisions which contribute to the solution of a problem. The specialists influence planning at various levels of abstraction. Each records its decisions on a blackboard, and all are able to review and retrieve decisions posted by other specialists.

The blackboard serves as a global data structure for the retention of problem solving information. A partial solution develops as each specialist adds information to the blackboard. The information may be hierarchical and specialists may operate on one or more levels of abstraction. One specialist must be responsible for halting problem solving activity when the solution is complete.

Human planning includes bottom up and top down planning. Bottom up planning consists of forward reasoning from an initial state to a conclusion using data driven decisions. Top down planning works backwards in a goal directed fashion. Specialists can be designed to incorporate either of these strategies.

The planning process consists of a series of cycles in which specialists read information from the blackboard, determine if they can contribute to the plan, and post the outcome of their actions to the blackboard. A measure of control is required in order to decide which specialist to invoke since more than one specialist may be able to act in a cycle.

The centralized executive decision maker must opportunistically select the most qualified specialist to invoke given the current context of the problem. Two examples of control measures which may be used are recency of invocation (the most recent specialist to propose a contribution is selected) and level of decision making (a higher level decision is preferred to a low level decision). The planner has no predefined sequence of specialists to invoke since decisions are posted as the solution develops.

The Hayes-Roth blackboard permits many possible decision sequences to be explored while creatively and opportunistically building a plan. Routine plans will tend to adhere to an iterative refinement and hierarchical planning process. Complex plans can capitalize on opportunism to develop on many levels, resulting in a more random decision making process.

1.4.2 Hayes-Roth Experiment

An experiment involving routing errand planning was conducted in an attempt to validate the Hayes-Roth model of human planning. The subjects were human planners who were each given the task of devising a plan to complete a set of errands in a fictitious town. The plan inputs included start time and location, end time and location, and additional constraints used to assist in prioritizing errands (such as mandatory errands and discretionary tasks arbitrarily specified by the experimenter). The problem usually allowed insufficient time for all errands to be completed. The planner was required to produce an output consisting of the errands to perform, the time to spend at each site, the order of site visits, and the routes between errands.

In [Haye 79a] and [Haye 80] thinking aloud protocols were recorded which provided significant substantiation for the opportunistic model of human planning. Additionally, it was found that early decisions influenced subsequent decisions at both lower and higher levels of plan abstraction.

Four levels of planning were identified in the experiment:

(1) <u>Outcome</u>: The planner determines an intent to perform the errand, but does not

decide details of where or when or which route to follow to complete this task.

(2) Design: At this level the planner divides errands into localized clusters and constructs a sequence of clusters to follow. No specific sequencing of visit sites within the clusters is undertaken. For example, the planner may decide to visit all sites in a mall before leaving that location.

(3) Procedure: The planner determines the detailed order of individual errands to perform within clusters.

(4) Operation: The planner specifies the routes to follow between individual errand sites.

The characteristics of the Hayes-Roth errand planner are:

(1) The input is a fully specified series of detailed tasks to perform and the period in which they must be planned.

(2) The output is a plan to accomplish the tasks the planner considers possible in the planning period. This includes how much time to spend at each errand.

(3) Planning is accomplished in a centralized opportunistic controlled manner.

(4) The best possible plan for a given problem is desired.

1.5 Tourist Information Geographic Problem

The work of a travel counsellor in a tourist information centre was selected as a practical example of geographic problem solving. Travel counsellors are frequently faced with difficult planning and re-planning tasks. They must discuss abstract personal desires with a tourist, retrieve information from a wide variety of sources, relate this to a map, and quickly suggest alternative plans. The problems they face are geographic in nature. The solutions they develop must be presented and related to the area of interest of the tourist.

Plans developed by a travel counsellor must be sound temporal and spatial descriptions of tourist activities. A tourist information planner should be more than an "Attractions of Interest" system. The output should reflect the input requests of the tourist and the considerations of the local expert. A plan will consider the impact of all these constraints and attempt to maximize the satisfaction of the tourist.

A recent survey of travel counsellor to tourist dialogues [Lero 89a] indicates that tourists ask a variety of questions. These include simple information retrieval queries, requests for routes to a single site, routes linking many sites, full daily plans, and longer term plans. This research address all of these questions except longer term plans.

The characteristics of a tourist information planner are:

(1) The input is a set of abstract tourist desires and a planning period.

(2) The output is a detailed visit schedule for a single day.

(3) The planning time is constrained, thus forcing the planner to seek an acceptable solution in a limited amount of time.

(4) The planner is static. It is not designed to monitor plan execution or to re-plan after partial plan completion.

(5) The routes are fully specified by detailed descriptions and are selected to require the fewest number of turns to prevent the tourist from becoming disoriented en route.

(6) Re-planning is initiated by the tourist and consists of planning a new daily plan after altering the starting conditions.

(7) All planning must be human-like in all specialists.

(8) The durations of stay at sites which are not appointments are based on system heuristic estimates of the normal stay at each class of site. These durations may be altered within limits to accommodate the temporal constraints of a problem.

The Hayes-Roth errand planning model would encounter difficulties if used to solve the tourist information planning problem because:

(1) The distinction between planning levels may need to be more flexible to permit the design and procedure levels to cooperate more closely. An alternative would be to combine these levels into one since their decisions are similar in nature and have significant effects on each other. For example, the design layer may decide to do all errands in a mall before leaving. This implies the procedure level should sequence all local sites immediately. However, the design level may also consider another site external to the mall to be a high priority visit and require that location to be scheduled early in the plan. This specialist conflict would require extensive coordination by the planning executive to resolve and may delay the solution for a considerable amount of time.

(2) The central executive control of the opportunistic planning process may result in long planning periods as the system bounces from one planning level to another. The tourist planner must quickly respond to tourist needs to assist the counsellors with their workload. As well, the tourist information system planner must implement the same policy human planners apply, which is to solve the problem simply and quickly to serve more tourists. The opportunism of the Hayes-Roth errand planner must be restricted in a tourist planning problem.

(3) The outcome planner responsibilities would alter significantly in a tourist information planner. This level would no longer be required to select a subset of tasks to perform. In a tourist planner this level must determine which tasks are best suited

to satisfy the desires of a tourist.

(4) The route planning techniques useful in an errand planner are different from those of a tourist planner. A tourist does not know the local geography and would quickly loose his or her way if directed along obscure local short cuts. The tourist route planner must find routes between sites which are easy for anyone to follow. Additionally, routes may be required which satisfy other desires such as scenic beauty or ceremonial paths.

(5) The Hayes-Roth model implements a clustering approach to site selection. This technique attempts to satisfy all errands in a local cluster before moving on to another location. This is one policy which may not be useful in short term tourist planning, but could apply to longer term and larger area plans. Time constraints are significant factors which must be considered before clustering and local site distribution are assessed. The Hayes-Roth approach does not seek to satisfy the time constraints for all sites since the planner is allowed to eliminate any errands which are too difficult to plan. This cannot be done in a tourist planner because the tourist specifies exactly which sites must be planned to satisfy his or her wishes.

The Hayes-Roth model is a good conceptual framework from which to begin to design a tourist information planning system. The problems outlined above can be solved by partially restructuring the levels of planning and altering the control strategy. As well, the detailed responsibilities of each level of planning will require adjustment. Each specialist in this model must adhere to human-like planning principles as outlined in Section 1.3.

Three planning levels were identified to better classify the hierarchy of planning spaces in the tourist information planner. These are:

(1) Task planning: The abstract intentions of the tourist are determined and converted into specific site selections. No details of when to visit a site or how to get there are considered.

(2) Concept planning: The detailed time and geographic precedence of sites is determined, beginning with high priority tasks and considering both temporal and spatial constraints.

(3) Route planning: An exact route between sites is found.

Task planning is quite different from Outcome planning. The process of locating sites which satisfy input requirements from a user is a feature of the tourist information system which is not required of an errand planner since the errand planner received explicit visit site specifications.

Concept planning combines both Design and Procedure levels to construct a schedule and sequence of site visits which satisfy the constraints typical of tourist planning. The errand planner conducts similar work, however the level of abstraction is higher. Errands are planned to be completed 'before' or 'after' other errands and clustering is used to prioritize tasks. In the tourist information system the planner must determine the exact time a visit should be scheduled. Clustering is not important for local short term plans since sites are within a reasonable travelling distance of each other. In addition, clustering may not be used to eliminate sites since all sites must be visited.

Route planning improves considerably upon the crude routing of the Hayes-Roth operational planning level. Tourist style routes must be described in great detail. These routes must be easy to follow and tailor to secondary wishes such as scenic beauty or historical interest.

The tourist information planning problem is not as arbitrarily opportunistic as errand planning since tourist counsellors routinely and repetitively plan in a very short time frame. The counsellor cannot afford to opportunistically juggle many possible planning considerations. The earliest useful solution must be found, not necessarily the most elegant plan. The routine nature of this planning can be implemented in a simple specialist executive controller which restricts the opportunism of the planner.

In summary, the Hayes-Roth errand planner and the tourist information geographic planner are similar in the following areas:

(1) They are both hierarchical planners.

(2) Both implement human-like opportunistic planning using a blackboard architecture.

The planners differ in the following respects:

(1) There are three levels of planning in the tourist system vs four in the errand

planner.

(2) The tourist planner significantly expands and alters the responsibilities of each planning specialist.

(3) The tourist planner is restricted in the amount of opportunism it may adopt whereas the errand planner incorporates a flexible, centralized controller.

(4) The inputs to the tourist system are at a much higher level of abstraction. The routes output by the tourist system must be more detailed than the errand planner since a tourist will not be familiar with the local area.

1.6 Tourist Information Planner Approach

The research conducted in [Haye 79a] and [Haye 80] substantiates their opportunistic model of human planning. Additionally, work in [Haye 79a] illustrates that the blackboard model is well suited to implement such a planning paradigm.

Each planning specialist is responsible for a different portion of the planning problem. The methods they employ to contribute to the solution must be tailored to the requirements of one level. Techniques which are well suited to route planning may not be applicable to task level planning. There has been considerable AI research directed at the mechanism of planning and the construction of planners for various problems. It is from this body of work that the approach to be used by each specialist was determined.

1.7 Other Geographic Systems

There are many Geographic Information Systems (GIS) which permit users to access and display information and spatial data [Kast 89, Kami 89, Smit 89, Mars 89, Lee 89]. The information is indexed to the spatial data so that queries may be answered which are based on geographic specifications. A GIS must be quite powerful and very comprehensive to respond to generic geographic queries.

A GIS can be used to construct an excellent "Attractions of Interest" package for tourist information. The system could quickly retrieve and display information regarding any attraction within the local map. The task level process of extracting sites to visit for specific purposes could be implemented as a simple rule-based interface.

The conceptual planning capabilities required of a tourist information planner are not normally present in a GIS. A GIS may contain information about sites, including their time constraints and functions, but the planning process of linking a series of sites and routes in a logical temporal plan is not a conventional GIS feature.

The route planning tasks of a tourist information planner are all within the capabilities of a GIS. A GIS may be used to find a route between two sites or routes which link a series of sites. Costing routes is easily accomplished by map coordinate calculations.

GeoRoute [Rous 89] is a geographic information system which is used to solve various delivery and public works problems. It is capable of routing a list of visit sites, planning a series of arcs to be covered by a vehicle and determining shortest path routes. This is primarily an interactive system with a comprehensive network editor. Users are required to conduct high level planning actions to establish site visit classes. The sites are extracted from the GIS and linked together by routes which are built with all traffic constraints satisfied.

GeoRoute performs route constraint verification tests to ensure temporal constraints are not violated by a schedule. The schedule is a product of the route planning algorithms. Temporal considerations are assessed after routes have been planned. A poor plan may be edited by the user to correct any timing difficulties. There is no capability to integrate temporal constraints with path planning.

The GeoRoute system is capable of limited task level planning, and comprehensive route planning. This includes the construction of routes to satisfy scenic or other considerations. GeoRoute is poorly designed to accomplish conceptual planning since scheduling and route planning are not integrated.

Conventional GIS data can be used as a source of this information and imported to GeoRoute through a conversion package. The GIS can be used to maintain the accuracy of the source data while GeoRoute responds to user requests and presents results in a simple graphical format.

Other route planning navigational systems integrate path planning with real time traffic information [Guzo 89, Hobe 89]. These systems are important low level tools for disaster relief planning and congested traffic control. The integration of these efficient route planners in a full planning package would require the addition of conceptual and task level planning modules.

In summary, other geographic planners contain many of the features required of a tourist information planner. There are none which integrate their planners in a structure similar to that proposed for a human-like tourist planner. Additionally, the planning approaches used in these systems are built to deal with large amounts of information. The programs do not attempt to model human cognitive processing. GIS and navigation systems represent an important source of geographic information and could be valuable support systems for a tourist information planner.

1.8 Organization of Thesis

This thesis consists of five chapters. This chapter describes the general model on which the tourist information system has been based. Chapter 2 contains a review of related AI research. This review was conducted to assess other planning paradigms and to extract techniques which could be used to solve the tourist planning problem. Issues regarding knowledge representation, search and path planning, re-planning, and scheduling were also investigated since they are important design issues in any geographic problem solving system.

Chapter 3 is a description of the system and software which constitutes the planning system. The methods used to implement the model are described.

Chapter 4 discusses system issues such as plan assessment, re-planning, software testing and lessons learned during development.

Chapter 5 draws conclusions from this work. Included are directions for future research. An Appendix contains hard copies of completed plans and software test results.

Chapter 2 Related Research

2.1 Introduction

Chapter 1 summarized the general model of a tourist information geographic planner. This chapter will review related AI research in order to identify planning techniques which may be used in the specialists (knowledge sources) of a blackboard based geographic planner.

Planning is not the only related AI field which must be considered. Other areas should be examined in order to determine the structure and mechanisms necessary to support a blackboard based planner. A knowledge representation suitable for geographic problem solving is necessary to effectively organize the large amount of information represented by a map. Efficient search techniques are required to locate a valid plan from among the many possibilities. Human planners commonly re-plan in the event of changed circumstances or plan failure. As a result, mechanisms to support re-planning must be explored. Related scheduling literature can be reviewed to provide techniques to build the schedule of site visits in a geographic plan.

2.2 Planning

Planning has been an important theme of AI from the very beginning of active research [Newe 56], [Newe 63]. Like a good game playing program, a computer-based planner will demonstrate significant intelligence if it can prepare plans and solve problems to which human observers can relate. Building good planners became an important early goal to define the objectives and capabilities of AI.

The different approaches to planning can be classified as linear, non-linear, hierarchical, constraint-based, meta-planning and script or skeleton-based planning. Knowledge rich planning also imposes interesting requirements and illustrates different techniques. Concepts may be extracted for use by the specialists of the tourist information system planner from the previous research undertaken in these areas. Each specialist may implement several different planning techniques to address its own unique responsibilities. These methods must all adhere to the principle of human-like planning.

2.2.1 Early Work

In Logic Theorist (LT) [Newe 56] Newell and Simon introduced the concept of heuristic search through intermediate problem states to reach a goal state. Their planner was goal directed, working towards the full solution by subdividing goals into sub-goals which could be satisfied by applying state operators. The General Problem Solver (GPS) [Newe 63] generalized this approach into "means end analysis". GPS conducted a search to locate the best operator to transform the current state closer to the goal state.

Robotic planning has offered much to the field of AI planning. STRIPS [Fike 71] is able to apply rule-like operators, triggered by pre-conditions, to change the problem state. It is based on predicate calculus techniques. A GPS-like means end analysis is used to select the relevant operator to apply.

STRIPS supports planning with a data structure to track goals and operators, and a database to represent the state of the planning space. The planner works towards satisfying the first goal in the data structure. When the operators required to do this are determined they are applied to the world state to change the database.

Planning in STRIPS terminates when there are no remaining goals to satisfy. At this stage the desired world state and the world state achieved by satisfying all the intermediate goals are compared. These will differ if the process of satisfying one goal reverses the satisfaction of another. This problem is addressed by putting the unsatisfied goal back into the data structure and resuming planning.

STRIPS is also a linear planner, operating on the assumption that a goal can be decomposed into independent sub-goals, each of which can be separately solved. The final plan consists of the complete satisfaction of one sub-goal followed by another in a linear sequence of activities. Although this reduces the search problem it can encounter difficulties when sub-goals interact. In this case the satisfaction of a subsequent sub-goal may undo the effect of earlier operations. In knowledge rich environments sub-goals will often interact [Suss 75] and linear planning may not be successful.

A non-linear planner intermixes the satisfaction of sub-goals. The sub-goals are treated as an unordered set of activities, not a sequence as in a linear planner. While this allows greater flexibility in planning, it also increases the size of the solution space due to the large number of possible permutations of planning steps. For non-trivial states this is a significant problem since any distinction between important and unimportant goals is lost in the set of competing sub-goals.

Sub-goals may interact in a geographic problem solver. Additionally, the planner will require the ability to distinguish important from unimportant sub-goals. As a result, non-linear planning techniques may be more useful than linear approaches. Since each specialist may conduct several different planning operations, linear planning may be appropriate for limited portions of the specialists work.

2.2.2 Interacting sub-goals

World state changes in the pursuit of a goal may result in the obstruction of a solution path through the state space. This occurs when an operator reverses the result of an earlier operation. Sussman has suggested that sub-goals will interact in all but the most simple of problems [Suss 75]. In HACKER [Suss 75], Sussman adopted a debugging approach to fix a plan after it had been constructed. He used plan critics to recognize errors and suggest corrections. In this way he dealt with sub-goal interactions as bugs, steps which had to be corrected to complete an acceptable plan.

Sacerdoti [Sace 77] suggested a least commitment strategy to address this problem. In his work, the plan was not ordered or completely formulated until absolutely necessary.

Interactions were avoided by delaying their sequencing. His approach reduced backtracking and plan fixing. NOAH [Sace 77] was a non-linear planner in which all the sub-goal orderings were considered possible until the planner was forced to commit to a particular sequence.

NOAH began by first satisfying sub-goals as if they were independent. It then determined any ordering of operators which would satisfy the constraints of the problem. This was performed by building a data structure to record any dependent ordering of operators. NOAH's final action was to activate the constructive plan critics which examined the plan and suggested orderings to resolve conflicts between sub-goals.

Humans may employ either of these two mechanisms to address the problem of interacting sub-goals. For relatively routine planning a process similar to NOAH could be successful. Each plan would begin with an abstract sequence of operations and as the planner progressed the details of sub-goal satisfaction would only be added at the last possible moment. This is similar to the simple daily planning people conduct [Haye 79a].

HACKER's approach may be more appropriate for more complex problems. Sub-goal interactions must be considered early in planning of complex interdependent solutions. This is similar to people planning longer term actions or solving difficult problems [Haye 79a].

2.2.3 Hierarchical Planning

The state spaces of a real world problems are quite large. Finding a solution in these spaces can overwhelm planners equipped with inefficient searching techniques. Newell and Simon

[Newe 63] suggested organizing the problem state into hierarchies and planning at different levels of abstraction. In ABSTRIPS [Sace 74], Sacerdoti implemented this idea by separating the state changing operators into hierarchies. His system relied on a priori designation of the levels of abstraction to which the operators would be applicable.

A hierarchical planner solves a problem by first satisfying the most critical goals in an abstract problem space. Subsequent refinement of the details at lower levels of abstraction will eventually result in a complete solution. A hierarchical planner will not be required to search extensive state spaces since each level represents successively less detail from the bottom up.

ABSTRIPS [Sace 74], begins by generating a high level problem solution. This plan is successively refined through the levels of abstraction until the primitive problem specific operations are known. Planning may be terminated at a high level of abstraction if details are unimportant. A special data structure is used to store all the individual actions as plans for different levels in the abstraction space. This data structure may also be used to guide plan execution, to refine an abstract plan in more detail, and to form a basis for re-planning should the problem state change during execution.

The reduced search space of a hierarchical planner offers significant advantages over a non-hierarchical planner. To ensure this, the hierarchy of these spaces must describe significantly different details of the problem space. Low level details should only be considered once a successful plan is completed in the higher space. The actual process of abstraction is domain independent but must be highly structured to build a useful hierarchy.

In ABSTRIPS, planning proceeds in a recursive manner beginning with the input of the level of planning, and a list of nodes from the search tree representing a skeleton plan. At the very beginning a new problem is at the highest level of abstraction and the skeleton plan consists of a dummy operator.

In the highest abstraction space the primary goal is solved by finding an applicable operator to reduce the distance between the current state and the goal state. The search for such an operator is constrained and rapid since there are relatively few applicable operators in the highest abstraction space. The planner does not leave the current abstraction space until a complete path to the goal is achieved. This policy is applied at each level of planning. Should a sub-problem, in a more detailed abstraction space, be impossible to solve, the planner returns to the next higher abstraction space. It then removes the operation which led to failure and resumes planning in the higher space to determine a new complete path to the goal.

ABSTRIPS incorporates automatic backtracking when goals are unattainable by the plan devised in the abstract space. The expense of backtracking and re-planning in higher spaces makes the selection of good plans in the abstract space essential to overall system performance.

Since geographic planning will be conducted in a large state space hierarchical planning is a likely technique to be used by the blackboard specialists. In the Hayes-Roth model this has been recognized and the blackboard organized in a hierarchy. Each specialist may also structure its planning in a hierarchical manner if the state space to be searched is extensive.

2.2.4 Meta-Planning

Complex planners are those which incorporate many different planning strategies. These planners must select and coordinate their modules in order to solve difficult problems. Meta-planning is thinking about planning. This technique was adopted in MOLGEN [Stef 81]. Stefik felt that a planner should be able to reason about its own problem solving methods and select an appropriate choice from the alternatives. Stefik also suggested that a flexible planner must be able to recognize when a chosen approach is succeeding or failing, and adjust the strategy accordingly.

A meta-planner must separate the control knowledge from the detailed planning mechanisms. This results in a hierarchy of meta-knowledge supervising the other planning levels. The meta-planner is responsible for selecting an appropriate strategy, supervising its execution, and determining when to alter the strategy if planning is not succeeding. The meta-planner must also recognize when a problem has been solved.

An agenda controller is used in a conventional expert system to select which rule to fire. The controller must select a rule from all triggered rules. This selection process is called conflict resolution. If a controller is rigidly structured and uses a single conflict resolution strategy, then it possess little meta-knowledge and even less meta-planning ability.

The solution to controlling the planner lies with the agenda. There are several possible options, ranging in complexity from a simple time stamped rule firing strategy, to a complete meta-planner controlling the agenda. MOLGEN implements the agenda controller as a complete planner which reasons about the nature of a problem, determines an appropriate planning strategy, and assesses the performance of this approach as planning progresses.

The tourist information planner is based on the blackboard planning paradigm. This permits the adoption of an opportunistic method of controlling the various planners in the system. The opportunism can be implemented as a variety of specialist invocation strategies such as most recent contributor or the specialist capable of contributing at the highest level of abstraction. Meta-planning could be incorporated in a complex specialist coordinating the opportunistic specialist invocations which could adjust the strategy as the solution develops.

The counsellors who perform tourist geographic planning are often working in a limited time frame. They do not have the freedom to solve problems in many different manners due to the high volume of requests they receive. They are constrained to a more routine problem solving approach. Incorporating complex meta-planning features to enable a computer model of tourist geographic planning to adopt many strategies, alter approaches, and re-plan to find better solutions, would not be consistent with the human problem solving method. The human planning process may not be the best approach to solving the problem, however it is the approach chosen to be modelled in this research.

2.2.5 Constraints

The relationships between interacting sub-goals can be expressed as constraints. For example, one sub-goal may require completion prior to another. This constraint can be expressed and used to ensure correct operator sequencing in the final plan. The constraints can also be used to limit the search for plans which resolve sub-goal conflicts by establishing partial orderings of operations. The planner must combine these partial plans to solve the problem while ensuring the constraints are not violated.

Constraints are formulated during the planning process as sub-goal interactions are discovered. As the planner proceeds, more constraints become apparent and the possible solution space is reduced. This leads the planner from the unknown to the known.

Constraints are an important search limiting technique. They also provide a mechanism to incorporate human requests into plans. Tourists may constrain a route to ensure that the most scenic path is followed, or to avoid hilly terrain on a bicycle. Tourists will also constrain the planner with temporal facts such as appointments, starting times, and plan completion times. The ability of a tourist information problem solver to provide good solutions is constrained by the number of possible sites to visit. All of these factors suggest that one or more specialist will incorporate constraint-based planning techniques.

2.2.6 Script or Skeleton Based Planning

A skeleton or script-based planner relies on another planning approach to generate the plans which are the basis for new problem solutions. For example, hierarchical planners produce plans describing actions which may be taken at various levels of abstraction in the problem space. An abstract hierarchical plan can be the outline of a plan which may be re-used by a skeleton planner to solve a new problem. A skeleton plan may be stored in different levels of detail, ranging from a fully specified solution to a simple outline.

Planning using skeleton plans begins with the location of a particular skeleton suitable to the given problem. The next step is to fill in the problem-specific information. The knowledge base must contain a considerable amount of domain specific information to support planning. A problem is solved once the details are added to the skeleton providing that a suitable skeleton was originally selected.

A critical issue for skeleton planners is the process of abstracting a more general representation of a plan. A plan may be generalized by replacing the specific objects in the sequence with more general representations. This process must be cautious since invalid plans may arise if indiscriminate substitutions are made. In Fikes' work, STRIPS is first used to generate a plan. A MACROP (MACro sequence of OPerators) is assembled from this plan to serve as the skeleton for subsequent solutions. An example is the actions a robot would take to pick up a block. This sequence can be stored in a skeleton and re-used.

Fikes describes an additional benefit to be derived from skeleton plans. If a data structure is used to retain the plan description and generalization it may also be used to monitor plan execution and to form a basis for re-planning. The generalized plan forms the guidelines for plan execution. If the plan begins to fail, re-planning may be undertaken to establish the actions necessary to return the deviant plan to the original sequence. In simple cases MACROPs may exist to repair the plan.

A script planner must relate the known to the unknown to select a skeleton to solve a new problem. In HACKER [Suss 75], Sussman described this as a process of classifying a given problem into a familiar class to which a known solution exists. A general problem solving strategy could be followed to construct a plan if the proposed solution could not be readily adapted to the new problem. Once an acceptable solution was found, it would be remembered and classified for future use. The planner would also learn from any mistakes made in the planning process.

Human geographic planning relies extensively on memory to assist planning. If repeated requests for the route to a popular location are made of a travel counsellor, the route will be retained and easily retrieved for new requests. This represents a pattern matching search task. General sequences of daily plans may be used to guide a travel counsellor. However, tourists come from a wide variety of backgrounds and exhibit many different preferences of sites to visit. Skeleton-based planning may not be of much use in these circumstances.

2.2.7 Opportunistic Planners

Many hierarchical and non-hierarchical planners were designed to operate in a restricted state space. In contrast to this, human reasoning is conducted in a space rich in knowledge and details. Humans are able to constrain search, reduce redundant details and situate a context during planning. The Hayes-Roth [Haye 79] cognitive model and blackboard paradigm were introduced in Chapter 1. In the model, human planning is characterized as opportunistic (see Section 1.4). However, opportunism is not without its problems.

In least-commitment strategies employed by hierarchical planners, such as NOAH, sub-goal interactions are tested and evaluated against constraints before details of their solution are developed. This prevents excessive backtracking. On the other hand, opportunistic planning is very likely to encounter sub-goal interactions. As a result, parts or all of a plan may need to be re-planned.

A human planner accepts the requirement to re-plan and exhibits a flexible response. Rather than blindly backtracking and re-planning, an opportunistic planner will seek to relax sub-goal constraints to develop a workable solution to the interaction. This is done 'on-the-fly' and is explicitly directed at correcting sub-goal interactions. The process may not involve backtracking, but may require constraints to be relaxed or imperfect plans to be accepted. Humans typically seek a quick solution at this stage. Backtracking and re-planning may generate a better plan, but the time and effort involved are extensive.

2.2.8 Summary

Each of the specialists in the tourist information planning system will be a small planner. This review of other planning concepts has illustrated some of the approaches which could be incorporated in the specialists. The techniques selected must all be human-like to be consistent with the theme that the tourist planner must be similar to human planning.

Sub-goals may interact in a geographic problem solver. Additionally, the planner is likely to require the ability to distinguish important from unimportant sub-goals. As a result, non-linear planning techniques may be more useful in the specialists than linear approaches.

However, each specialist may perform several planning operations. In this case linear planning may be appropriate for limited portions of a specialists work.

If a specialist encounters interacting sub-goals it may employ a least commitment strategy, as in NOAH, or a plan fixing approach like HACKER. Both are human-like solutions to this problem. Additionally, resolving sub-goal conflicts by relaxing constraints or accepting less than ideal solutions are alternatives human travel counsellors would implement. Fixing a nearly complete plan by small 'on-the-fly' changes may prevent excessive backtracking.

Specialists should restrict their planning to one level of the blackboard. In this way they can contribute decisions which apply to one level of the planning hierarchy. This will restrict the search space in which each specialist must work. If a specialist encounters complexity within its level of expertise, it may utilize a hierarchical planning process internally. Hierarchical planning is human-like and should be exploited.

A travel counsellor must satisfy a tourist request in a limited amount of time. This results in a restricted planning environment where the chance to employ limited opportunistic planning strategies exists. Similarly, there is little time for complex meta-planning control.

The wide variety of personal preferences tourists exhibit severely limits the suitability of skeleton planning. At a low level routes may be constructed from smaller route sequences. A small number of plans may be memorized and re-used if they are in frequent demand. This may require a more exhaustive survey of tourist queries. Constraint-based planning may be incorporated in a specialist. It is human-like and well suited to limiting search in knowledge rich planning environments. Constraints provide a useful mechanism to specify human likes and dislikes.

2.3 Knowledge Representation and Spatial Planning

2.3.1 Introduction

The internal knowledge representation used by the specialists in a geographic planning system must be analogous to human representations of spatial information in order to be consistent with human cognitive models. This applies to both the organization and contents of the information.

Several spatial cognitive models were examined to investigate the problem of knowledge representation for geographic planning, [Kapl 73], [Lee 74], [Pete 73], [Davi 86], [Kuip 78]. This led to a well supported concept of knowledge representation for cognitive maps.

Conventional research in knowledge base (KB) organization, spatial databases and geographic information systems was also reviewed to develop a model representation. This resulted in a knowledge base which is consistent with human cognitive maps and is well suited to deal with spatial data relationships.

2.3.2 Cognitive Maps

Human spatial or geographic planning is based on a model of the area of interest held in the mind (when such planning is not being conducted on a paper map). This cognitive model represents a large scale space which cannot be fully observed from one reference point. Humans attempt to solve geographic problems by extracting relevant information from their cognitive model during route finding and position calculations [Kapl 73]. Heuristics and logic are the tools used to solve these problems.

In [Pete 73] the use of cognitive maps by primitive man was investigated. He established that a hunter was required to formulate a simplified version of his hunting routes in order to find his way. This simplified mental map was organized around landmarks, regions, paths and intersections of paths. The paths formed the main organizational feature of the map.

A cognitive map can be characterized as a collection of many loosely related smaller maps [Davi 86]. A network of streets and intersections is used by people to represent the true geography of a large scale space. This network is often stylized to a rectangular grid in order to simplify geometric retention of information.

A cognitive map may contain a considerable amount of information. For example, a brief visit to a city will result in additions to the image of that city. This could consist of a very restricted and localized route network around a friend's house, or a more expanded but narrow route sequence followed to reach a desired location. This knowledge is constantly improving as the depth of detail increases. A person who knows a region intimately will have a well organized and readily accessible cognitive map of that area.

Route finding in a cognitive map consists of a search for a set of links from a source to a goal. Humans avoid brute force search since an unconstrained search is inefficient and time consuming. This implies the use of techniques which capitalize on constraints to limit search and the use of pattern matching to relate the known to the unknown. Pre-planned routes may be retained and incorporated into new paths. Route finding heuristics can be applied to seek out a new path. In this way, paths are learned and stored for future use [Davi 86].

2.3.3 Relationship of Cognitive Maps to AI Knowledge Bases

A cognitive map may also be referred to as a knowledge base [Davi 86]. This implies similar problems will be encountered when working with artificial cognitive maps as with any other knowledge base. In fact, since cognitive maps may themselves be very large, with partial and incomplete representations gathered from a lifetime of experience, their organization for efficient use is important.

Using conventional knowledge base terminology the problems encountered with artificial cognitive maps [Davi 86] are:

(1) Knowledge representation: the cognitive map must be encoded in some form of data structure which can represent many different types of geographic information, including spatial relationships and absolute locations.

(2) Information retrieval: a large and disjoint cognitive map in a human mind poses no problem for retrieval of pertinent information. An artificial cognitive map must overcome the problems of retrieving the necessary information to support geographic planning and support a shifting focus of attention.

(3) Learning: a cognitive map is a dynamic structure, constantly expanding and reorganizing as new information is obtained through experience. An artificial cognitive map should be able to learn and adapt with new experiences. Davis deals extensively with this subject.

(4) Knowledge base consistency: this is a part of learning, since new information should not be permitted to render previous, and still valid, information incorrect.

(5) Investigation: a planner must be able to use information from the cognitive map to solve new and unknown problems, the main subject of this research.

2.3.4 Knowledge Representation

An efficient knowledge representation is one which consumes a minimal amount of memory and can be easily accessed by a planner. This is a critical issue in the design of practical AI systems. The information represented in the knowledge base may be factual declarations of knowledge, heuristic rules representing the capabilities of domain experts, meta-knowledge or knowledge about how to use the knowledge to solve problems, and processing functions to support all of these features.

The more common methods of representing knowledge in AI systems are: logical formulae, semantic nets, production rules, frames [Geva 84] and analogous representations [Elli 86]. Logical knowledge representations use first order predicate logic to represent a description of the world. The knowledge base is a collection of formula which permit the construction of statements about the properties and relationships between objects. Logic formulated knowledge bases are difficult to manage when large amounts of knowledge are involved. They also lack organizational features to help structure the knowledge. Logical formulae use a simple notation which is precise and flexible, and derivations can be performed by automated theorem provers. In addition, facts need only be declared once.

Semantic nets are collections of nodes organized in a directional graph to represent the world in question. The nodes are objects with associated properties. Links define relationships between objects. Facts may be inferred from the network without searching the entire knowledge base and inheritance links can be used to support deductions. Semantic nets are difficult to use when representing temporal relationships.

A knowledge base organized as a production rule system consists of a collection of loosely coupled rules. Rules may be organized into independent modules to better structure their information. Each module is capable of reasoning about a limited domain of the entire subject. The rules rely on a global database to instantiate variables and confirm the problem state. A controller implements a conflict resolution strategy which selects the appropriate rule to fire from the triggered or activated rules.

Rule based systems are easily expanded and can be used to represent heuristic and imprecise knowledge. They provide a mechanism to link procedural and declarative knowledge representations. Rule based systems are a natural means of capturing and expressing expert knowledge, and have been responsible for the large number of "Expert Systems" currently under development or in use. Rule based systems suffer performance degradation as the size of the knowledge base increases. They experience rule consistency problems with ever expanding rule sets. In addition, they represent knowledge about the structure of information only implicitly, making such information difficult to deduce.

Frames are structures which provide a uniform method of organizing information in a complex database. A frame is a structure, with slots for values. Each frame resides in a hierarchy from which values for the slots may be inherited. Slots may also hold relations and other frames, thus permitting the mixture of both declarative and procedural knowledge in one database. A frame based knowledge representation is a natural method of organizing real world data which supports efficient retrieval and inference techniques.

An analogous knowledge representation [Elli 86] uses the natural knowledge structure and content to define an internal configuration. Knowledge is retained in a format as similar to its real organization as the computer permits. For example, an image can be retained as a matrix of brightness values of its pixels. Analogous representations are useful in spatial planning and navigation.

An analogous knowledge base permits the critical properties of the knowledge to be readily retrieved and processed. There is a trade-off between the computational retrieval effort and knowledge storage requirements. Very efficient search techniques are required to locate information in these large knowledge bases. An analogous knowledge base is easy to maintain because the information is simple to see and understand.

In practical AI systems many of these knowledge representation schemes are combined to obtain maximum benefit for a particular application. Single representation systems are rarely constructed. Each scheme has advantages and disadvantages which must be considered when designing an appropriate representation for a particular problem or class of problems.

2.3.5 Spatial Databases

A geographic knowledge base contains spatial information. Techniques used to organize spatial information will be applicable to this type of planning knowledge base. There are two principle models of spatial database organization: the relational model and the object-oriented model [Moha 88].

A relational database is built of domains and relations and contains no links between relations. This organization is ideal for linear tabular data. However, spatial or 2-dimensional data must be stored in multiple interlinked tables. This results in lengthy access paths and redundant information storage. Additionally, in order to represent hierarchical information, different tables for each level of the hierarchy must be constructed along with a table relating the hierarchies.

When a relational database is used to store spatial information, incorporating multiple levels and tables, the structure of the database is very complex. In addition, users must be familiar with the structure in order to build queries and obtain the desired information. Their queries must be well formed and precise in order to obtain appropriate answers. Inexperienced users may be unable to build such queries.

Object-oriented databases for spatial information are consistent with the analogous approach to knowledge representation [Elli 86]. In object-oriented databases an object hierarchy can directly represent the hierarchical nature of spatial information. Collecting objects in classes allows properties to be inherited, and may reduce information storage requirements. Objects may reside in several hierarchies and inherit properties from many parents. This results in a lattice type of structure which closely resembles the real world data organization.

Objects are entities which include both procedural (methods) and declarative (attribute values) information. Operations are performed by message passing between objects. Each of the objects can be directly referenced to obtain or edit the local information it contains. An object-oriented database closely resembles a frame or schema based hierarchical knowledge representation. The principle advantage with an object-oriented representation is the addition of features and capabilities provided by executable methods and message passing.

Object-oriented databases are more capable of representing spatial information than relational databases. However, they are unable to capture the heuristics and logic used by humans to work with the information they contain. Objects must be augmented by logical assertions and a deductive reasoning or inference mechanism to permit processing of information during problem solving.

2.3.6 Summary

The preceding sections have examined contributions from the work in cognitive psychology (cognitive maps), knowledge representations (logic, rules, analogous representations and frames), and database models (spatial, object-oriented). This leads to a knowledge base model for the tourist planner which represents a composite of the characteristics of all these fields.

An object-oriented paradigm has been selected as the data model of the map information to capture the spatial nature of the geographic information. This information is organized hierarchically, the root being a map object which has sub-classes consisting of links, regions and buildings. Figure 2-1 shows the organization of the geographic information. The map information has been augmented with rules which are used to reason about actions and plans, to assess outcome of planning stages, and to incorporate some of the heuristics humans use for geographic planning. The rules represent a logical reasoning component of the knowledge base and are retained in separate rule hierarchies. An inference engine is used to implement both forward and backward chaining over these rules. Rules are invoked by Lisp function calls.

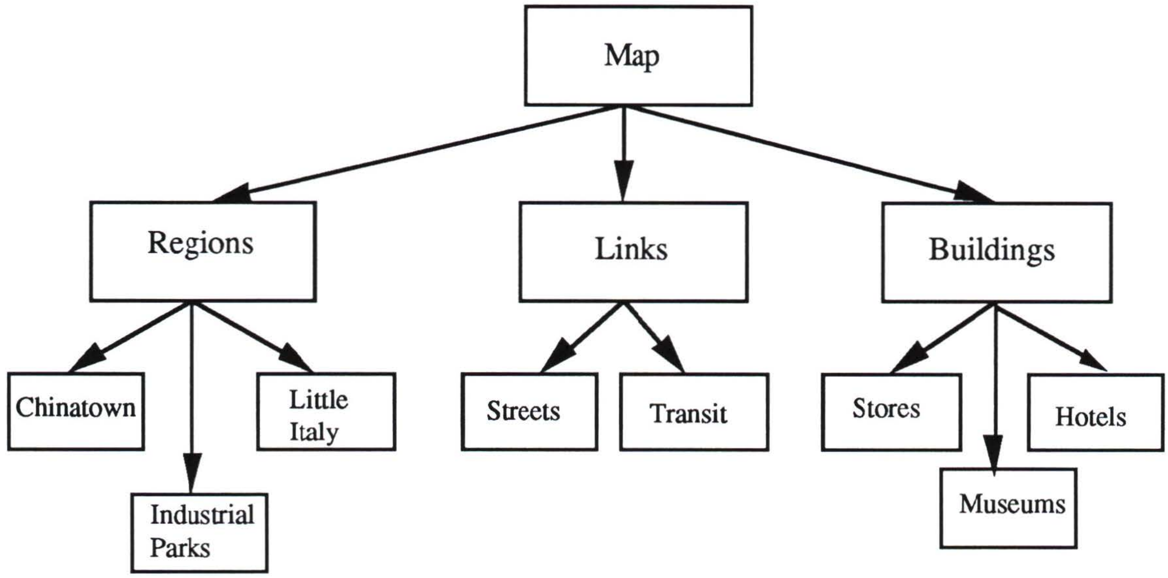

Figure 2-1 Map Hierarchy

The detailed site instances of the map hierarchy must be designed to contain information useful for planning. Object-oriented features of the knowledge base permit the generalization of slots in a generic class object, and the specialization of site specific information at the instance level. For the purposes of tourist planning all sites must contain certain basic information:

(1) Opening Time: time site opens.

(2) Closing Time: time site closes.

(3) Appointment Time: any specified appointment time.

(4) Duration of Stay: the usual duration of stay at that sub-class of site. For example, a bank is 45 minutes, a restaurant is 90 minutes.

(5) Latest Arrival Time: the last possible time a tourist could arrive at a site and still be productive. A bank latest arrival time is the closing time less one minute while a museum's is the closing time less 60 minutes.

(6) Scheduled Arrival Time: the time the planner suggests arriving at that site.

(7) Site Before: the site before the current site in the final plan.

(8) Path: the selected path from the previous site to the current site.

(9) Paths: all the possible paths the route finding specialist has determined are available.

(10) Intersection Near: the intersection near the site which is used in path planning.

Some of the values instantiated in actual site object slots are characteristics of a class of sites. These are only entered once as values in the slots of the parent of the sites. Other slot values are local and must be entered for each site object. Items 6 through 9 are plan specific information inherited by an object only when it is incorporated into the planning hierarchy for a specific plan.

An additional feature of an object-oriented knowledge representation is the ability to use an object-based blackboard for planning. Specialists may access the blackboard by sending and receiving messages. The decisions posted by specialists may be recorded in slots of the blackboard objects, or as separate objects which themselves spawn more specialist activity. The goal tree may be constructed of objects which use methods and message passing to control the planning activity.

Many geographic planning operations are not merely information retrieval functions or reasoning tasks. Distance calculations, routing algorithms and system utilities are required. Object-oriented techniques permit this code to be embedded in objects thus allowing inheritance through the object hierarchy, and activation by message passing.

2.4 Search and Path Planning

2.4.1 Introduction

There are many methods available to plan a path through a restricted representation of a geographic region (a map). Selecting the appropriate method depends to a considerable extent on the manner in which the map is stored, the amount of information available to the route planning specialist, and trade-offs between computation and search. This section reviews searching techniques frequently used in AI systems, concentrating on topological problem representations such as networks of nodes and links.

2.4.2 Network Representation

The internal representation of a network must be considered before discussing a path finding search strategy for geographic planning. Humans think of streets as objects in a spatial context, with associated descriptive features such as length, adjacent objects and intersecting streets. This suggests that one of the descriptive components of a street should be a list of intersecting street names. These would be used by a human to describe the street for routing purposes. For example, a person describing Fort street (Figure 2-2) could refer to it as the street which crossed Wharf, Government, Douglas, Blanshard, and Cook streets.

To implement this approach a street must be described by its own name and a list of streets with which it intersects. A consistent pattern of building the descriptive list of streets can be used to add information to the representation in an implied fashion. For example, always building the list from North to South or East to West. This information can be used to assist route finding specialists during planning.

Figure 2-2 illustrates two highlighted streets: Quadra and Simcoe. Using this representation Quadra is described by the list (Bay North.Park Balmoral Pandora Johnson Yates View

Fort Broughton Courtney Burdett Humboldt Superior Dallas) and Simcoe by the list (Dallas Oswego Menzies Government Douglas).

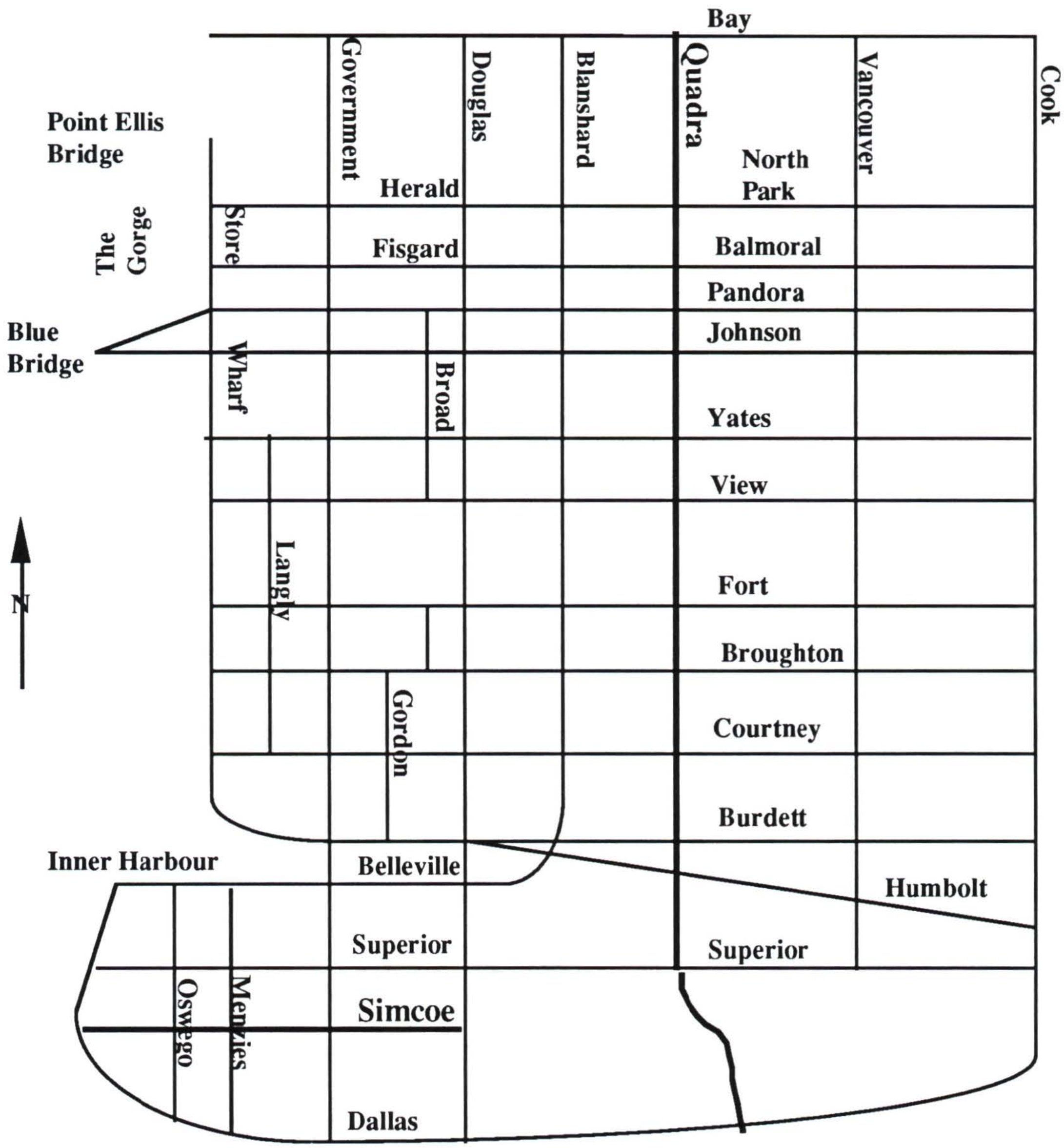

Figure 2-2 Sample Street Representations

The nodes and arcs which would be required to represent a map in a graph structure can be replaced by concise and descriptive street lists. This makes translation of any network of paths into a computer representation quick and simple to implement. Additionally, this representation is consistent with the structure of human cognitive maps (see Section 2.3). People retain streets as lists of names and visualize routes as street traversals. A typical path description may be:

> *"Take Johnson from Wharf past Government and Douglas until you reach Blashard, turn right, and continue to Fort. Turn left off of Blanshard to Fort and the bank is on your right."*

The human planner has used street names in order to guide the traveller along this path.

2.4.3 Path Planning

Searching for a path through a network is a well documented AI research field. The techniques used to locate any path, and to locate the shortest path, must be examined to determine which may be used in a network represented as a human cognitive map.

2.4.3.1 Finding Any Path

There are several approaches to searching for a path to a goal [Tani 87], [Wils 84]. Basic depth first and breadth first searches simply require the ability to determine if a node is the goal node, and a means of storing the path to the goal.

A depth first search traverses a network through the immediate descendants of the current node until the goal is reached or no further exploration below that node is possible. If the goal is not found the procedure backtracks to the start node and examines another successor. Depth first searching can waste much effort if the network is very deep and the goal node is not found in one of the first deep branches explored.

Breadth first searches concentrate at one level of the network before exploring deeper. This technique is more suited to searching for goals in networks which are very deep and the branching factor at any one level is not excessively large.

Hill climbing improves depth first search but requires a natural measure of remaining distance to the goal to succeed. As a result, more information must be available to the searcher and additional computation is required at each decision point. Hill climbers use the estimated remaining distance to the goal to select the most promising alternative to explore first. The distance measure may be parameterized to consider many factors when assessing remaining costs.

Hill climbing search performance degrades in difficult conditions. The procedure may wander aimlessly when all possible alternatives appear to cost the same because it will be unable to differentiate improved positions. If nodes which are not the goal have an attractive cost measure (foothill problem) [Wins 84], they may draw the search away from the true goal. Parameterizing the estimate exacerbates these problems.

A beam search explores a reduced number of alternatives at each level. The searcher uses cost factors to order the options before pruning. This is a form of breadth first search which relies on an estimate of remaining distance to the goal to order the exploration possibilities at a branch point. Only a sub-set of these options will be explored at each level. The search is limited by selectively controlling the branching factor. Beam search also suffers when the alternatives at a branch point are equally promising.

A best first search reduces backtracking and is more likely to produce a path with fewer nodes expanded. In this technique the node chosen for expansion is the closest node to the goal in the entire network. The node is selected by comparing candidates for expansion over the entire network, not just within a local branch.

2.4.3.2 Shortest Path

There are several techniques which may be used to determine the shortest path between two points. The British Museum procedure finds all possible paths and chooses the least expensive route to follow. This approach does not require costing information during path generation, but relies on brute force to generate all paths. This is a form of generate-and-test exploration which is very inefficient in large search spaces.

Branch and Bound is another procedure which does not rely on additional information in order to build paths. This technique follows the shortest path generated to date in its exploration of the next step. This is an improvement over the British Museum approach but remains inefficient. Branch and bound can be greatly improved if estimates of remaining cost to the goal are added. However, this generates a requirement for additional network information.

Dynamic programming techniques can be used to improve path generation. A dynamic programing searcher will compare all paths which reach a common intermediate node and will select only the path of minimum length for further searching. If a branch and bound search reaches the same node from two separate paths, only the shortest path to that node will be explored further. This reduces searching effort by eliminating redundant paths.

The best features of these search techniques can be combined into a branch and bound search with an estimate of remaining distance to the goal, using dynamic programming to eliminate redundant paths. This is the classical path planning algorithm A* [Nils 68]. A* assigns priority to paths with nodes which are on the shortest path from the start of the search, and are estimated to be the closest to the goal.

A* is an admissible search algorithm since it has been proven to find the lowest cost path to a goal, given that the estimate of remaining cost to the goal from any node is never larger than the actual remaining cost.

The A* search algorithm is [Wins 84]:

Begin

Form queue of partial paths.

Until the queue is empty or the goal has been reached,

Determine if the first path in the queue reaches the goal.

If this path reaches the goal, do nothing.

Form new paths by expanding one step.

Add new paths to the queue.

Sort the queue by the sum of the cost accumulated to date and a lower bound estimate of the cost remaining.

If more than one path reaches a common node, discard all but the shortest.

If goal found, terminate successfully, else signal failure.

End

A* relies on the estimated remaining distance to the goal from a node n, denoted h'(n). It determines which path to follow by computing:

$$f'(n) = g'(n) + h'(n) \tag{2-1}$$

where f'(n) is the estimated total distance of a path from the source to the goal through node n, and g'(n) is an estimate of the cost of the path from the source to the node n. The lower f'(n) value is deemed superior and that path expanded towards the goal. In order to ensure admissibility of A*,

$$h'(n) \leq h(n) \qquad (2\text{-}2)$$

must be true for all nodes, where h(n) is the true distance remaining to the goal from node n.

The A* algorithm is a preferred technique for finding the shortest path through a network. However, A* relies on two key factors. The first is the availability of sufficient information to compute the estimated cost from a node to the goal. The second is the requirement that the estimated remaining cost to the goal not exceed the actual cost for any node. A weak heuristic could be devised to estimate h' for this network and A* implemented to find routes. However, A* is not a human-like path planning algorithm and the heuristic does not relate to any human concept.

Alternatives to A* must be considered for use with this network representation. Depth first or breadth first strategies in a British Museum procedure, or the branch and bound approach are two alternatives. Since the British Museum procedure is very inefficient, branch and bound becomes a likely option.

2.4.4 Summary

Searching for paths between points in a geographic space is a challenging task. Several algorithms exist which are capable of finding the shortest path to a goal. The amount of information required by each algorithm differs. The most optimal, A*, requires the ability to estimate remaining cost to the goal from a node. If this information is not available then alternative algorithms must be considered.

A human cognitive map is organized with very little costing information. This prevents human planners from employing optimal search algorithms such as A*. A computer based path planner may be able to employ more brute force searches, like the British Museum procedure, but this is not an option if the system is to mimic human cognitive processes. The approach used by human planners must be studied to determine an appropriate path searching mechanism for the tourist information planning system.

2.5 Re-Planning

2.5.1 Introduction

Re-planning is the formulation of an intended course of action in response to changes to the factors which were used to conceive the original plan. The amount of effort required to re-plan depends directly on the number and severity of the alterations to the first plan. If only minor changes are made, re-planning may simply involve the addition of several operations to the original plan. Should the entire problem be revised, re-planning may require considerable work in searching for appropriate actions to take. In either case the issue of whether to re-plan or begin the plan from the beginning must be addressed.Which of these re-planning concepts to apply is dependent on the efficiency of the planner and the amount of disruption a change has made to the plan.

Re-planning is an important capability of complex problem solvers. The ability to slightly alter an intended course of action in response to some changes in circumstances is a highly intelligent behaviour. In many situations re-planning represents a crisis management activity. The effect of the problem state changes may not have been considered when solving the first problem, but now must immediately be accommodated in a new plan.

Re-planning may be stimulated by monitoring plan execution. An execution monitor can ensure the application of the specified operators will lead to the intended goal [Geor 88]. Re-planning can begin as soon as the monitor determines that the original plan will no longer solve the original problem.

2.5.2 Re-Planning Requirements

Re-planning requires a system to possess some knowledge of the original plan and the changes which stimulated the requirement to re-plan. The state changes may be explicitly declared or derived from a new state observed during the monitoring of plan execution.

In the Hayes-Roth Planners' Workbench [Haye 81] the authors add to these requirements the need for a plan rationale. This is a knowledge of the issues which went into the original plan. The rationale should include data, assumptions, considerations, and decisions which were made while solving the problem. Re-planning without a plan rationale resembles planning, and is often as time consuming. This is unacceptable in crisis situations.

A complete plan rationale must contain considerable information. Simple plans may hide complex reasoning and detailed search in elegant solutions. A plan rationale can also be used to provide proof that significant factors were not overlooked during the planning process. In addition, a plan rationale may provide the means of explaining decisions by detailing the alternatives considered and discarded, and supporting the individual actions taken. A plan rationale must be a well organized data structure to support the storage of decisions and the retrieval of re-planning information.

In O-Plan [Curr 85], Currie uses a Goal Structure Table (GOST) to record the pre-condition(s) of any action in the plan, and the point in the plan where these must be met. The relationship between individual actions in the plan is recorded along with the goal or sub-goal each action is attempting to satisfy. The monitor watches for low level faults during execution and plans corrective action which will not affect the high level plan. More drastic re-planning must be undertaken at a higher level should this fail. Re-planning begins with the world state as it was when the plan began to fail and works to satisfy the goal.

Croft [Crof 85] records all decisions made during planning in a tree structure. While this structure is not used to monitor plan execution, it does provide a starting point for re-planning if the intended goal should change. The user specifies the point at which the plan should diverge towards a new goal and the planner begins from there to construct a new decision tree which reaches the new goal.

2.5.3 Knowledge Base Support

The knowledge base must provide a mechanism to satisfy the requirement for a structure to assist re-planning. An object-oriented hierarchy can be used to represent the plan rationale. This structure would be accessible to all the planning specialists. A hierarchical record of planning decisions will permit each specialist to quickly review decisions which it considers relevant, and constrain any search through the planning details.

An object-oriented rationale also offers other advantages. Both the plan and the rationale behind it are object-oriented hierarchies located in the same knowledge base. As a result, redundant information need only be instantiated once and linked to both hierarchies. In this way, both space and accessing time may be saved. Additionally, information in the rationale may be more relevant when presented to the re-planner in conjunction with the final plan. This is especially true for decisions taken early in the planning process which could be meaningless if viewed in isolation, but provide substantiation for the plan when seen from a broader perspective.

An additional advantage of designing the re-planning structure as an object-oriented hierarchy in the knowledge base is the ability of the specialists to directly post their planning considerations at the same time as they alter the state of the solution tree. This results in incremental expansion of the re-planning structure concurrent with the plan as well as permitting specialists to backtrack from unpromising solutions by retracting operators.

2.5.4 Blackboard Considerations

Human planning can be based on deferred or highly structured decisions with complex interdependencies considered during the planning process. The latter case can be satisfied by building specialists which formulate a complete plan which considers all interrelating dependencies prior to execution. If all planning decisions are stored in a rationale, re-planning can be initiated by adjusting interdependencies and allowing the specialists to correct the plan from that point on.

The deferred decision approach requires a planner to simulate the commitment of actions while following an abstract high level plan. The planner must complete the low level planning tasks during the simulation to fill in the details. In this case, planning specialists can be triggered to solve local problems as they arise. For example, a lunch planning specialist can be activated as the simulation approaches noon hour. The specialist will consider the constraint that the customer is at work, the weather, what was eaten previously, likes, dislikes, etc., to develop the lunch plan for that day. This specialist is also constrained by the high level plan, but is free within its own field to explore many possibilities.

An added advantage of deferred decision making is the possible satisfaction of collateral tasks. If the lunch planner decides to leave the office and go to a restaurant, the world state changes during execution. If the user had deferred the planning of a postal activity until later that day, this change in world state could be used to trigger the postal activity planner to fit its task into the plan.

The blackboard architecture supports simulation of a deferred decision planner by constructing specialists which are capable of executing plans. The planned actions can be used to change a model of the world state stored on the blackboard and trigger low level specialist activities. All specialists are able to observe and contribute to plan formulation in the simulated world state since this state resides on the blackboard. The lessons learned during the simulation may be fed back into the planner to improve the plan.

2.5.4 Summary

An object-oriented plan rationale is ideally suited to support re-planning. It provides a location for retention of decisions and considerations made during planning. In addition, it allows this information to be linked directly to the plan in an efficient manner.

Re-planning is quite similar to planning. The specialists responsible for the original plan may be used to re-plan given a new problem state. The re-planning approach may be to return a deviant plan to an original course of action, or to re-plan from a new problem state to the first goal. Re-planning specialists can be constructed and integrated into the blackboard to implement either of these policies. Re-planning capabilities may also be integrated in all planning specialists.

2.6 Scheduling

2.6.1 Introduction

The geographic scheduling problem consists of assigning arrival times to sites. The process of selecting which sites to visit is a planning task. The scheduling problem requires that the time constraints of the tourist and the sites be satisfied. A valid schedule consists of a time-ordered sequence of the sites the tourist will visit. The problem of developing a schedule is primarily a search through the large state space consisting of all possible valid schedules. Many techniques may be used to limit or constrain this search. Without constraints, scheduling more than a few sites would overwhelm a problem solver.

A schedule must satisfy both temporal and spatial constraints. Simply meeting all timings in a schedule will not satisfy the spatial requirements. These requirements consist of global concerns such as minimizing overall path length, and local issues such as stopping off at one site on the way to another to eliminate backtracking whenever possible. The planner must integrate spatial considerations in all scheduling actions.

2.6.2 Related Work

Scheduling in a factory environment can be defined as the process of selecting an ordered sequence of operations to accomplish a task. Each operation is assigned a start time, end time, and the resources it requires.

Scheduling may be separated from the process of planning the operations required to produce an object [Fox 84]. This is extremely difficult to do in job-shop scheduling since an operation can only be scheduled when the entire sequence is considered. Each operation in a sequence is only feasible when the resources it requires are available at the scheduled time. The sequencing of two simple operations must be done with complete knowledge of the effect this action would have on an overall plan.

Production scheduling is difficult because of the combinatorial complexity of assigning many jobs to many machines with varied requirements, and because the execution of a predicted schedule is often uncertain in a factory environment. In addition, scheduling is hard because of the diverse and conflicting constraints which have to be considered. These include due dates, cost limits, production rates, machine utilization, resource availability, and labour requirements.

The combinatorial complexity of production scheduling can be approximated for a factory with *m* machines and *n* jobs [Stok 89].

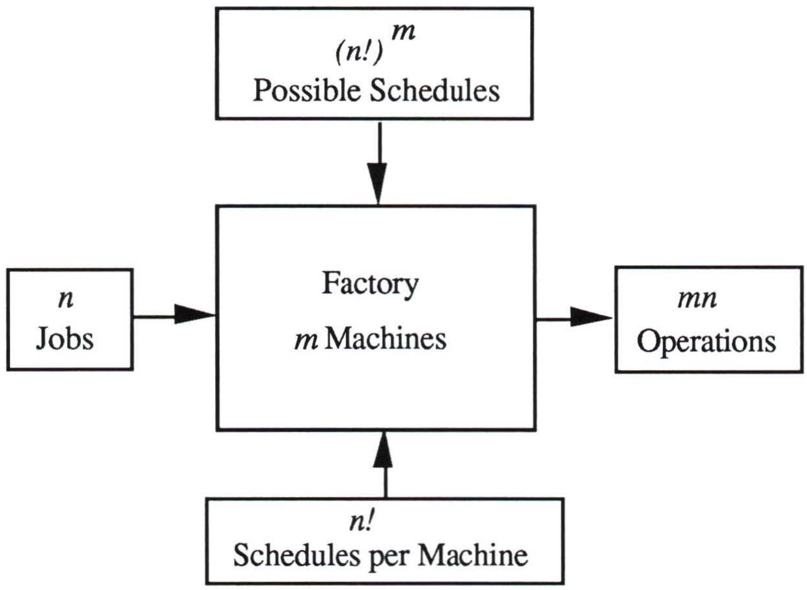

Figure 2-3 Factory Scheduling

There are *n* jobs to be performed by a factory which contains *m* machines. Each job must go through every machine, but they do not all follow the same sequence of machine operations. A schedule is defined as the start time of the *mn* operations. The schedule is usually designed to minimize some criteria such as work in progress or processing time. With $n!$ schedules possible per machine, and *m* machines to do the work, there are $(n!)^m$ possible schedules.

2.6.2.1 ISIS

Fox [Fox 84] identified five constraint categories which are used in scheduling:

(1) organizational: due dates, costs, machine utilization

(2) physical: limits of machines, storage limits

(3) causal: precedence relationships

(4) availability: resources such as parts and labour

(5) preferences: preferred machine for a job

Fox also established two objectives of factory scheduling: to meet due dates and to satisfy the constraints generated by other parts of the plant. Humans have difficulty solving the scheduling problem because it is very dependent on other decisions made throughout the plant, not just at a central planning office.

ISIS uses hierarchical constraint-directed reasoning to conduct scheduling. Knowledge is viewed as constraints which guide a heuristic search for possible schedules. Search is conducted at four levels of planning. Each level consists of three phases: problem construction, search, and solution assessment. Constraints are used to bound, guide and analyze the search.

The search space is composed of the partial schedules generated to date. New states are created by applying an operator to the current partially completed states. Then constraints are used to score states before conducting a beam search for the best schedule to pursue.

Since there may be more than one alternative schedule, a post search analysis phase is undertaken to select any one which scores higher than a threshold value. If no schedule scores sufficiently high, the source of the scheduling error is determined and fed back into the problem construction phase to select new operators and re-schedule the same operations.

To summarize, ISIS uses constraints to score the alternative schedules incrementally constructed in a hierarchy of planning levels. It uses beam search to limit the number of possible alternatives expanded during planning, and scores the outcomes prior to selecting an acceptable resulting schedule. ISIS re-schedules by feeding back the source of scheduling failures to the problem definition phase to improve operator sequence selection.

2.6.2.2 SCORE

High volume manufacturing presents a slightly different scheduling problem [Chio 84]. This differs from job-shop scheduling in the volume of work and the rates of production. High volume manufacturing consists of a reduced combinatorial scheduling problem since the components are already manufactured and the task is assembly-line oriented.

Assembly line scheduling is defined as the process of coordinating the flow of components through the line so that the output requirements are met at all times. This is constrained by the issues of minimal storage space requirements and a minimum amount of work-in-progress (Just-In-Time manufacturing).

The predictive scheduling task consists of translating the master production plans into detailed flow schedules of work centres. In SCORE [Chio 84] this is accomplished in four phases using a top down methodology. The phases are:

(1) Phase 1: High level constraints focus the demands of the master production plan. The process of binding quantities and process time intervals to work centres begins. Assembly lines are selected and reservation hypothesis built.

(2) Phase 2: Resource allocation conflicts are resolved and more precise time bindings are specified.

(3) Phase 3: Fill model process reservations are built around the fixed activities of the earlier phases.

(4) Phase 4: Lot sizes are optimized in the fill models.

Since significant down stream problems may arise from even minor delays at work centres, reactive scheduling is critically important in assembly line operations. Most re-scheduling is caused by component shortages due to high rejection rates, late deliveries, or work centre equipment malfunctions.

Reactive scheduling is a bottom-up operation which responds to delays encountered at work centres. It is structured as a search of the space of possible valid alternative schedules for an acceptable solution. A valid schedule is one in which the problem has been eliminated or postponed by some change. Constraints are used to focus the search on the most critical issues. The A* search algorithm is implemented to apply these constraints to score possible schedules and to select the best partial state to expand.

SCORE is very similar to ISIS in its approach to scheduling. Both use constraints to limit search through the possible space of admissible schedules when building or repairing a plan. The constraints are applied to score prospective states generated by the application of state space operators. The selection of operators is focused by higher level constraints to address the most significant problem with the previous schedule.

2.6.2.3 Steel Manufacturing

Steel manufacturing requires frequent scheduling due to the increased user demand for small volume, high variety parts [Numa 84]. The schedules required are daily, short range plans which must be generated in a reasonable amount of time. Manufacturing constraints to be considered include limits on waiting time of molten steel, continuous use of rolling machines to maximize yield, and repair periods for high use machines. A good schedule will be one which satisfies the conflicting requirements of minimizing waiting time and maximizing production.

In [Numa 84] the authors describe their approach as cooperative scheduling. The system consists of procedures and rules which are used to construct the schedule. This is combined with an efficient user interface to permit user controlled schedule evaluation. The goal is not to find an optimal schedule, but to find a feasible schedule in a reasonable amount of time and then to allow the user to review and improve it interactively. This approach integrates the current expert with the new computer based product to improve user acceptance.

The architecture consists of three elements: a scheduling engine, a rule base, and the user interface. The scheduling engine deals with machine conflict resolution. The input is a schedule in conflict and alternative machines, and the output is a schedule which does not violate the constraints. The engine preserves the global structure of the input schedule by changing starting times and machine utilization to repair a schedule.

The rule base is used to shift and join units of schedules and to employ expert heuristics about efficiency and quality of schedules. The interface allows the user to input his decisions at any point in the scheduling process and see the effect they may have on the final schedule. An expert can construct an acceptable schedule by iteratively refining his decisions. Evaluation is left completely to the expert. This means complex objective functions need not be used after schedules are constructed. In addition, only one schedule is produced because all constraints are applied and satisfied during the scheduling process.

2.6.3 Relationship To Current Research

The geographic scheduling problem differs from job-shop or assembly line scheduling because resource requirements are not a factor. This reduces the combinatorial complexity to a $m!$ problem. The user requires a sequence to visit m tasks, each of which is available between specified time periods for specific durations. A tourist planner need only search through the space of *m*! possible sequences to find a solution. Although this significantly reduces the search space, it remains extensive due to the continuous nature of time where any sequence may have an infinite number of possible time assignments.

Humans normally consider time a discrete entity. When planning daily activities people rarely concern themselves with fractions of a second, or even seconds [Lero 89a, Haye 79a]. The smallest unit is usually minutes, even hours when planning at a high level of abstraction. This serves to limit the number of possible schedules for each of the *m*! sequences.

Constraints may be useful in both job-shop and geographic scheduling. Constraints are applicable at various points of the scheduling process and serve to limit the search space. Defining these constraints and determining when they should be applied is required before geographic scheduling can succeed.

The cooperative scheduler is the most human-like of the schedulers examined above. Both SCORE and ISIS concentrate on multiple schedule generation, constrained search, and evaluation functions to select an appropriate schedule. These AI techniques are not necessarily the tools used by humans when constructing geographic schedules. The cooperative scheduler uses similar constrained searching, but incorporates user specified constraints and user evaluation to construct only one final schedule through a process of iterative refinement.

2.6.4 Summary

Geographic based scheduling is a hard problem to solve. It will require a system which is capable of dealing with many conflicting constraints and a large search space. The system must not use an elaborate search tree constructing paradigm as in ISIS or SCORE since this is not human-like scheduling. Although cooperative scheduling permits more user involvement in the scheduling process, its search techniques are designed for computer implementation as in ISIS and SCORE.

A system of human-like geographic scheduling will be required to incorporate the rules and procedures used by people when they solve this problem. A specialist similar to that described by [Numa 84] would permit this.

From the related research it is clear that scheduling consists mainly of constraint directed planning, where constraints are used to limit the search space of possible solutions. Since the problem is geographic, both temporal and spatial constraints must be considered in planning. In attempting to implement human-like scheduling the use of extensive search tree construction and multiple solution generation as in [Fox 84] and [Chio 84] should be avoided.

2.7 Chapter Summary

The previous sections of this chapter have examined the issues of planning, knowledge representations, search and path planning, re-planning and scheduling. The review of related planning research in Section 2.2 identified human-like planning techniques which could be incorporated in the specialists of a blackboard planner. These included linear planning in simple problem cases, and non-linear planning to solve problems where sub-goals interacted. A non-linear planning specialist may adopt a least commitment approach, or it may choose to eliminate sub-goal interactions through plan fixing. This could include relaxing problem constraints as the planner progresses towards a solution.

Individual planners in the tourist information system may be hierarchically organized. This may be necessary to restrict the search space of the specialists at each level. In addition, the specialists may employ hierarchical planning to develop their local contributions to the overall plan within the blackboard model.

The flexibility of planning strategy is limited in a tourist information planner. Queries must be promptly answered to avoid backlogs in the information centre. This suggests the opportunism of a tourist information planner should be constrained to speed plan development.

After reviewing the requirements of a geographic knowledge representation it has been determined that the spatial nature of geographic planning data can best be organized in an object-oriented hierarchical database. Constructing the entire knowledge base in an object-oriented fashion permits the problem goal tree to be organized as a hierarchy of planning sub-goals. Additionally, the methods embedded in sub-goal objects may be used to control the planning process. The object-oriented paradigm also supports modular, re-useable code to perform the routine processing and retrieval tasks required of any knowledge based system.

The map data must be augmented with rules in order to implement the heuristics humans employ during geographic planning. The rules represent the logical component of the planner. They can be exploited by forward and backward inferencing techniques since humans plan in both directions during opportunistic problem solving.

The restricted amount of information retained in a human cognitive map makes path planning difficult. Conventional search algorithms rely on distance measures between objects to improve their performance. A geographic planner which mimics the cognitive planning process of humans must find routes through a very sparse network representation based on lists of street intersections. This requires a path finder which differs from the more common search algorithms, while at the same time simulates the methods humans would use in similar circumstances.

Re-planning requires the retention of decisions and factors which affected the original plan formulation. This can be implemented in a plan rationale constructed as an object-based hierarchy of decisions in the knowledge base. The planning specialists all contribute to the plan rationale during planning. The rationale may also be linked to the problem sub-goal hierarchy in the knowledge base to eliminate redundant information storage.

Re-planning can be implemented in a blackboard architecture by constructing re-planning specialists. Some specialists could be responsible for returning a plan to the original sequence of actions by constructing a small re-planning sequence. Others could plan a complete new path from an altered world state to the goal. A re-planning policy specialist could be used to determine which strategy to adopt in each new case. A re-planning specialist may simply be a planner with additional re-planning capabilities.

Scheduling problems have been solved in many job-shop manufacturing systems. The usual approach is to direct a search for valid schedules with constraints. More than one valid schedule often results. Constraints may be used to score and select the preferred option.

Human planners do not construct several solutions to a scheduling problem since this requires extensive memory storage and recall actions. Human scheduling is very similar to other planning problems and may employ any, or all, of the techniques described in the planning section. The most likely approach is a form of constrained search with plan fixing if sub-goals interact.

This review of related research has revealed many techniques which can be incorporated in a tourist information planner. There remain problems which are specific to the tourist planner which have not been solved by previous researchers. The following chapter will describe the details of a working tourist information planner including the manner in which problems such as route planning in restricted map representations and scheduling geographic visits to satisfy temporal and spatial constraints have been solved.

Chapter 3 System Description

3.1 Introduction

This chapter begins with a theoretical description of the tourist information planning model to illustrate how the knowledge sources are constructed. Then, system issues such as software organization, object-oriented techniques, knowledge source details and the implementation of blackboard-based opportunism are discussed. The remainder of the chapter describes the implementation of the tourist information planning specialists.

3.1.1 System Description

The tourist information planner is a blackboard-based human-like planning system. It consists of a hierarchy of planning levels: task planning, concept planning and route planning. This hierarchy is complimented by a plan assessor module which assembles planning decisions in a plan rationale in order to provide an explanation facility and support re-planning.

The task planner is a rule-based planning module. This specialist converts a tourist's abstract desires into categories of visit sites which could be satisfied in the planning region. The planner implements simple if-then rules to select the appropriate sub-class of sites from the knowledge base and to present the visit choices to the tourist. The tourist interactively selects the actual sites to visit.

The concept planner is a complex planning module. It incorporates hierarchical skeleton planning, constraint-based planning, non-linear planning and human-like constraint relaxation procedures. This planner must satisfy both temporal and spatial constraints established by the tourist.

Concept planning begins with skeleton planning implemented as a priority scheduling system. The planner schedules the highest priority sites at the beginning of planning. Priority sites are established by the tourist as appointments, and by the system which uses a skeleton priority list to ensure the most important sites are included in the schedule first. The priority scheduling system is supported by a rule-based module which is used to determine the most appropriate hour to schedule a food site.

Actual site scheduling consists of constraint-based planning where the site is scheduled at a time which satisfies all site time and overall time constraints. The overall time constraints are propagated through all site scheduling actions to ensure the plan is valid. This prevents one site from consuming most of the available time at the expense of other visits. Spatial constraints are considered when each site is scheduled.

Human-like constraint relaxation techniques are applied in the event of a scheduling failure during conceptual planning. This approach eliminates excessive backtracking and re-scheduling activities by relaxing global and local constraints 'on-the-fly' as scheduling problems arise. This is a human approach to the problem of interacting sub-goals in a non-linear planner.

The route planner is a series of processing functions designed to support the other planners. It implements a human-like route finding algorithm designed to overcome the lack of information in a cognitive map which precludes the use of more conventional search techniques.

The plan assessor is a rule-based module which interprets the decisions taken by other planners during the problem solving process. The interpretations are extracted from the 'then' portion of the rules and assembled for presentation to the tourist as an explanation feature. Other rules are used to initiate re-planning by restoring the state of the blackboard (resetting control variables and clearing the plan rationale and plan assessment of stale data) and invoking the appropriate planner if a tourist requests an alternative plan.

The tourist information planning model is illustrated in Figure 3-1.

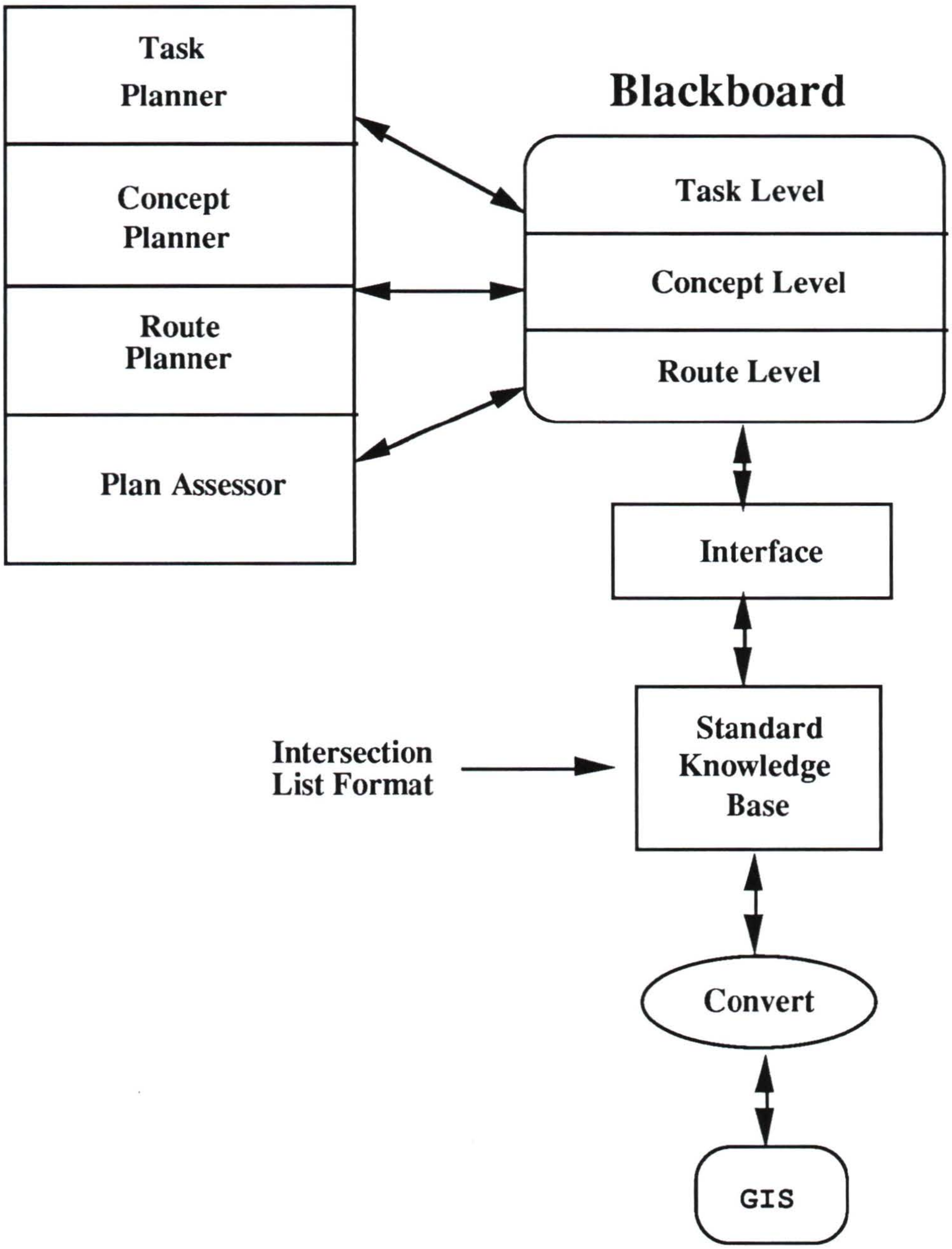

Figure 3-1 Tourist Information Planning Model

3.1.2 Software Development System

The tourist information planning system has been developed with the IntelliCorp KEE™ Software Development System. KEE is an advanced AI shell which is run on a Sun™ 3/60 Unix™ based workstation. KEE was selected as the AI shell to implement this project for several reasons. KEE supports forward and backward chaining rule-based inferencing with a wide variety of conflict resolution strategies ranging from simple agenda controllers to user defined techniques. This permits rules to be used to represent the logical component of human planning. KEE is also an object-oriented development environment which supports the hierarchical representation of spatial knowledge, the object-oriented representation of blackboard goal trees, and the integration of graphical windowing features to display plans. In addition, KEE is LISP based and allows the definition of LISP functions to support planning with symbolic processing.

KEE was selected as a practical development choice because it was readily available on a less frequently used workstation within the department. This permitted uninterrupted development during critical design periods. As well, KEE provides an easy to follow tutorial package, video tape instructional assistance and comprehensive documentation. This documentation is further enhanced by KEE's adherence to Sun Common™ LISP standards, windows and conventions.

3.2 Software Organization

3.2.1 General Issues

The software which constitutes this system is made up of modules which are organized as facts, rules, functions, methods, and active values. Facts are the spatial knowledge in the hierarchical object-oriented knowledge base. They include all the information present in the planning map. Links, buildings, regions and parks are all in this hierarchy. Each sub-class refines the details until actual visit sites, streets, parks, and regions are described. Figure 3-2 is a partial representation of the hierarchy implemented for Victoria, BC, Canada.

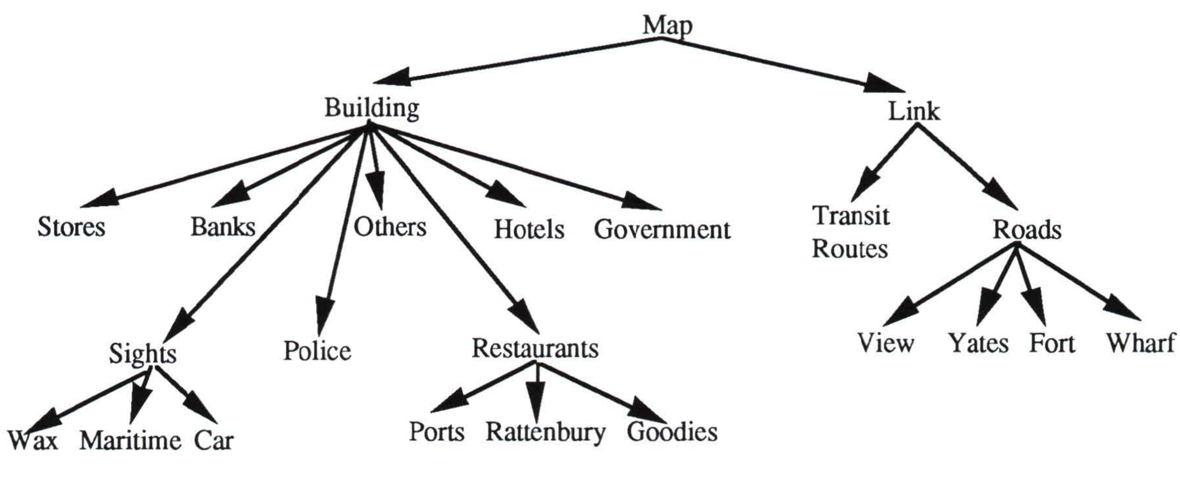

Figure 3-2 Partial Map Hierarchy of Victoria

The leaf objects in the knowledge base represent locations which can be visited by a tourist to satisfy a sub-task in an overall plan. These objects contain site specific information organized as slot-value pairs. Some slots contain object-oriented executable code while others retain static information. Table 3-1 summarizes the slots and values of a sample leaf object.

SLOT	SLOT TYPE	SAMPLE VALUE
Hard_Arrival_Time	Number	NIL or value
Build_Scheduler_List	Method	function named after slot
Closed	Number	1700
Coords	List	(190 300)
Filter.Path	Active Value	Cost.By.Turns.Av
Intersection.Near	List	(Yates Wharf)
Latest_Arrival_Time	Number	1530
Open	Number	1200
Output.Path	List	((Fort Douglas) (Broughton Douglas) (Broughton Government) (Fort Government))
Scheduled_Arrival_ Time	Number	1200
Scheduler_Working_ List	List	(Eatons 1200 1700 120 1530 1200 NIL)
Site.Before	Active Value	Router.Av
Stay	Number	120
Turn.Costed.Path	Active Value	Output.Route.Av
Turn.Instructions	Text	(Right Right)

Table 3-1 Leaf Object in Hierarchy

Rules are organized in separate modules, each of which is a rule hierarchy in the knowledge base. Every module is fully capable of independent inferencing. Rule sets are either forward or backward chaining, although the syntax of all sets is the same. Rules are invoked by LISP calls from other rules, LISP functions, methods, or active values.

Methods are LISP functions embedded in objects. These functions are activated by messages sent to the object with the parameters required of the code. Methods are characteristics of an object and may be inherited by descendants of the object. This permits code to be re-used in other objects. An active value is similar to a method in structure but different in activation. A method is triggered by an explicit message arriving at an object, whereas an active value is triggered by the addition or deletion of information in the object. Active values are inheritable properties of objects.

3.2.2 Knowledge Sources

A knowledge source is an entity responsible for planning (a specialist). It may consist of rules, LISP functions, methods and/or active values. At the outset of this research most of a knowledge source was coded in rule modules. However, it was quickly found that the KEE inferencing engine was much too slow for such an approach to be useful. As a result, many portions of the knowledge sources were removed and re-coded in LISP, either as functions, methods or active values. LISP COND clauses are used to implement the more routine choices in simple decision making cases. Complex decision making operations are coded in rules to capitalize on the features of rule-based systems, at the expense of increased processing time.

Knowledge sources which perform relatively routine symbolic processing tasks are coded in LISP. Examples are the task planner and some specialists in the concept planner. Knowledge sources which are required to infer more complicated decisions, such as the route planner and the food time selection expert, are rule modules. Although some knowledge sources incorporate both functions and rules, they remain a single module in terms of system organization and functionality. This permits modifications and maintenance tasks to be undertaken with little difficulty.

3.2.3 Blackboard Paradigm and Opportunism

The blackboard paradigm was developed to support opportunistic reasoning. Opportunism is the ability to select a path to pursue at decision points without a priori knowledge of a sequence of actions to take. There are two techniques used to implement opportunistic knowledge source control: centralized and decentralized.

A centralized knowledge source controller must be able to determine the appropriate specialist to activate when given a planning state and proposed contributions from all activated knowledge sources. All specialists bid for activation through this controller. A planning cycle consists of bid-select-activate, with selection and activation under central control.

In the tourist information planner executive decision making is decentralized and control is implemented with active values, methods and knowledge source calls. Each knowledge source is able to decide whether it should continue planning or invoke the assistance of another specialist.

Methods and active values provide the mechanism to implement decentralized opportunism. For example, the route planner has a LISP path finding module. The module is triggered whenever any planner requires a path between two intersections. At the concept planning level this occurs when a function is called to cost the path to several intersections. At the route planning level this occurs when the final order of sites is known and the planner must construct a route to link the sites together.

In the tourist information planner a knowledge source is not permitted to activate and alter the blackboard until its level of planning is required. All planners are always aware of the current planning state. A decentralized control mechanism is sufficient for the tourist information problem because each knowledge source is capable of solving one portion of the problem. The restricted opportunism of the human travel counsellor in a busy travel information centre is mimicked by the decentralized control of the tourist information planner. The following sections contain detailed descriptions of the specialists in the tourist information planning system.

3.3 Task Planner

3.3.1 Process Description

Task planning is the highest level of cognitive activity undertaken by the planner. It mimics the process a tourist follows when deciding what to do for one day. The planner is designed to plan activities in one 24 hour period. It begins with a tourist locating goal sites to visit which will satisfy some important personal desires. For example, an ornithologist may be seeking sites most suitable to bird watching or a horticulturist may wish to visit all local display gardens. In addition to abstract personal wishes, task planning frequently incorporates the inclusion of the obligatory visits for that day. Examples include banking, booking into a hotel for the night, visiting a friend's house, or finding an appropriate restaurant for dining.

Task planning is a process in which the human does not wish to be burdened by the details of routes or visit timings. The task planner is a goal directed specialist which attempts to locate visit sites most suitable for each activity. The system should enable the tourist to examine the planning domain (the map) and express his or her goals, locate goal visit sites, and select any or all of these possible locations for inclusion in a plan.

High level constraints such as starting location and method of travel must also be determined so the other planners may focus their search. The process of task planning terminates when all the background information required by the planner has been determined and the tourist is satisfied that all the high level visit locations are appropriate to his or her desires for that day.

3.3.2 Task Planner Software Operation

The task planner consists of a backward chaining rule module which implements the goal directed search for visiting sites, LISP windowing and selection functions, and methods embedded in the root goal object of the blackboard. At the start of planning the task planner backward chains to trigger the tourist interface code to present the tourist with a map display and request a starting intersection.

LISP functions are used to present the tourist with a task level activity menu (Figure 3-3). A task level object is instantiated in the goal hierarchy of the blackboard when an activity is selected. The task planning rule system sends a message which triggers a method in the new object to consult the knowledge base and locate any sites which may be used to satisfy the selected task.

Select Desired Activity
Shelter/Accomodations
Restaurants/Snack Bars
Sights/Museums/Attractions
Banks/Financial Establishments
Government Provincial/Municipal
Police
Shopping
Other Sites
*** Selection Complete ***

Figure 3-3 Task Level Activity Menu

The possible visit sites are graphically presented to the tourist for selection. Each one chosen is linked from the static knowledge base hierarchy into the dynamic planning hierarchy under construction in the task planner. Multiple inheritance permits every selected site to add to its own attributes the slots and values of an ancestor site object in the goal hierarchy (Figure 3-4).

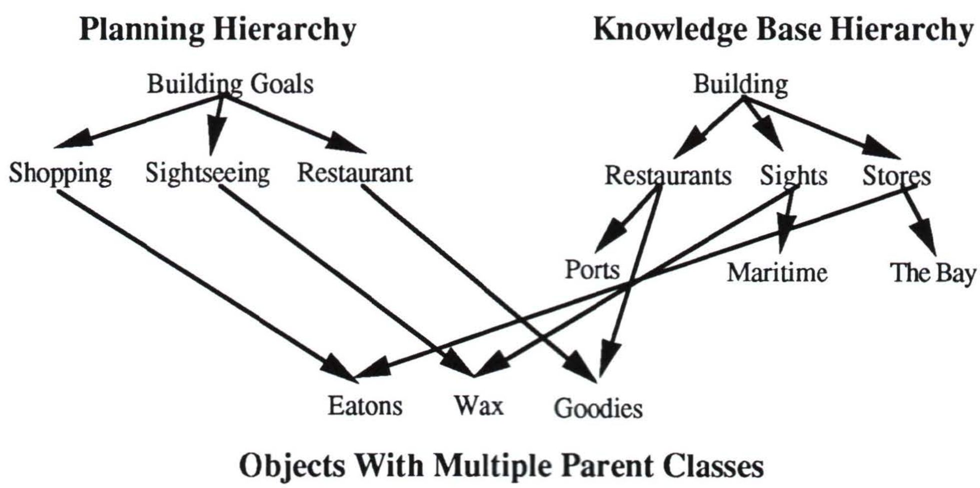

Figure 3-4 Example Planning Hierarchy

The tasks selected are assembled in a sequence which is sent to the root concept planning object in the knowledge base. This action triggers an active value to create a descendant concept planning object with the sequence as the sub-tasks to plan. The concept planner is activated by the existence of the descendant concept planning object on the blackboard.

3.4 Concept Planner

3.4.1 Process Description

The second level of cognitive activity in the tourist information system is conceptual planning. This begins once a tourist has determined the desired activities to be performed. It is the most complex of the levels of planning since it makes most of the key decisions and searches the largest problem space. It is assumed the travelling time between sites can be distributed equally to all stay periods for the geographical domain of the prototype. This is valid because planning is restricted to a small area. For larger problems travelling time would become a separate temporal constraint to be considered by the conceptual planner.

Conceptual planning refers to the process of determining the visit time for each selected site. The ordered sequence of these visits is the concept planner schedule. The list of sites will have been determined by the task planner prior to the commencement of conceptual planning. The problem begins with this list of tasks and some global constraints such as starting time, ending time and method of travel. Every site imposes additional local constraints such as opening time, closing time, duration of stay, latest arrival time, and appointment time (if any). These details are extracted from the site object in the knowledge base (see Figure 3-2).

The scheduling process is primarily a search problem. The search space is the multitude of possible schedules which could satisfy the local and global constraints. The goal of this work is to develop and implement a system of conceptual planning which closely resembles the process a travel counsellor follows when searching this space. During this research several approaches were attempted with varying degrees of success. The final implementation features a blackboard architecture for conceptual planning which operates within the overall system blackboard. This internal blackboard is local, in terms of functionality, to the concept planner, but it is also accessible to the site sequencing module of the route planner for geographic site ordering. The local blackboard implements a human-like conceptual planning process in an architecture ideally suited to opportunistic planning.

3.4.2 Simple Approach

A simple approach to the conceptual planning problem consisted of ordering all the sites based on their distance from the starting location and scheduling the visits in that order. Prior to site ordering, simple global time rules were applied to ensure the tourist had specified sufficient time for all visits. Site durations of stay could be reduced to fit all stays in a specified time frame during scheduling (Figure 3-5).

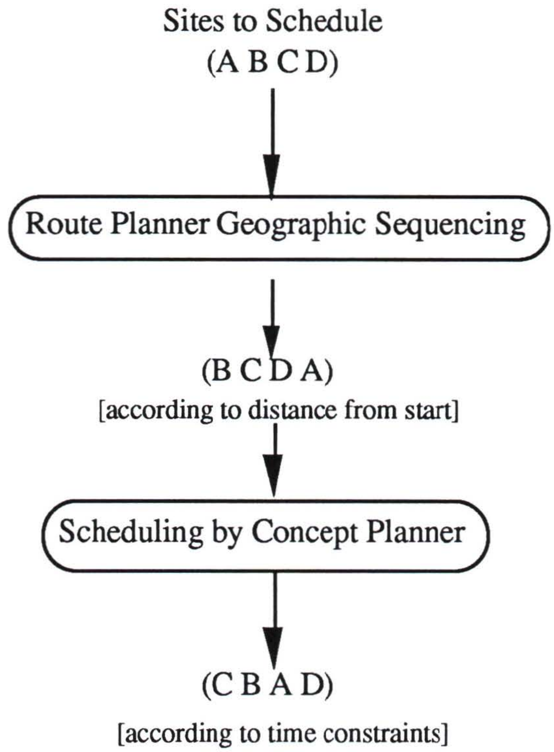

Figure 3-5 Simple Conceptual Planning

This approach worked fine for some problems, but when the scheduler was unable to achieve a workable schedule due to time constraints such as arriving too late or too early at a site, the planner simply re-ordered the schedule with no consideration of the geographic impact this may have. The resulting schedule often proved to be valid, even if stay times had to be reduced in the process, but frequently made little geographic sense. For example, the tourist would occasionally be required to travel across the city to the next site because of time constraints when a short wait could permit a closer site to be visited first.

This approach was abandoned for two reasons. The first was the poor plans produced because geographic considerations were often very important but could not be considered after the original sequencing. The second was because humans do not plan this way [Haye 79a]. Conceptual planning must include considerations of both time and space concurrently during the decision making process.

3.4.3 Cooperative Approach

A cooperative concept planner was implemented which relied heavily on the top level blackboard architecture to support the consideration of both temporal and spatial constraints. A module in the concept planner was assigned the task of monitoring and propagating time constraints. These constraints are propagated by ensuring previous scheduling decisions which affect the planning time are never reversed. For example, the act of scheduling an appointment removes a period of time at a specific hour from the planning period. This constraint is applied at the outset of planning and propagated throughout the plan to ensure it is never violated.

Another module in the route planner was made responsible for determining the ordered geographic sequencing of sites. These modules were required to use the blackboard as a common repository of their knowledge since neither was capable of doing the work of the other. The modules must cooperatively solve the conceptual planning problem.

The first cooperative approach attempted was based on the assumption that humans are most concerned with building a schedule incrementally, while at the same time ensuring global constraints can be met. This meant that one site would be scheduled at a time, but the impact this would have on the overall schedule would also be considered.

Sequences were first geographically ordered from the starting location by the route planner module. After this, the scheduling module in the concept planner was invoked. The scheduler is capable of relaxing temporal constraints and geographic site ordering to arrive at a valid schedule. A valid schedule is one in which all sites have been allotted a scheduled arrival time and the durations of stay at the sites do not overlap. No schedule is valid if any site is unscheduled or if two sites overlap in time due to their duration of stays.

The scheduler consists of a series of rules which are used to construct a valid schedule. The rules are ordered to ensure the scheduler will return a sequence which does not deviate from the suggested geographic sequence if time constraints permit. Otherwise, additional rules attempt different strategies to construct a schedule. All the valid schedules are assembled in a list and returned by the expert. Figure 3-6 illustrates the scheduling module. The rules are prioritized in the module to ensure the best possible schedule is returned at the front of the list.

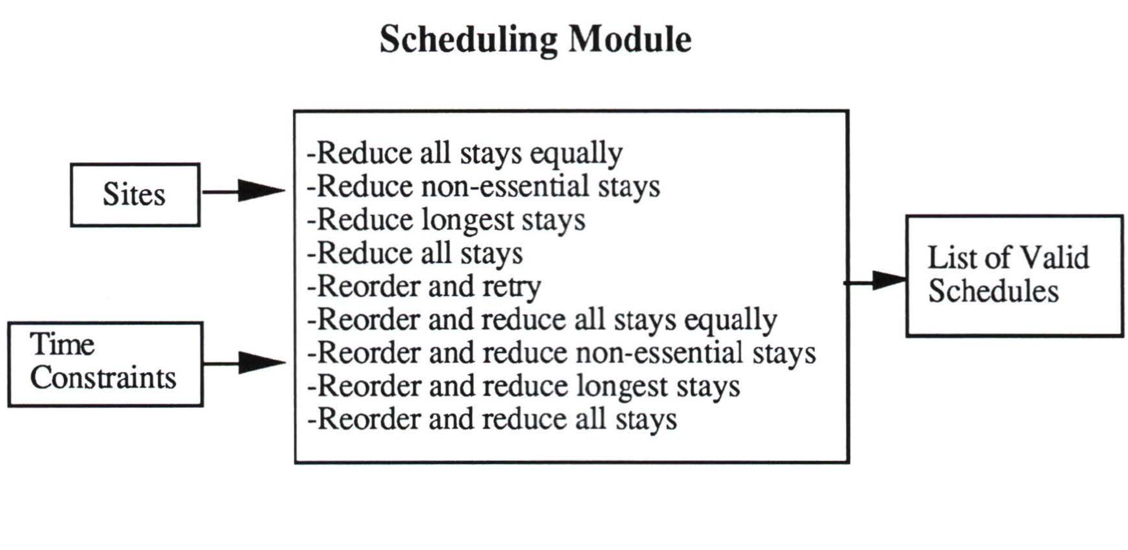

Figure 3-6 Cooperative Scheduling Module

The first cooperative planner implemented merely selected the first schedule in the list of valid schedules as an acceptable solution. After each schedule selection the spatial constraints were reconsidered. This was done by setting the current location to the first site in the first valid schedule of the set, reducing the planning period by the time this site required, and reordering the remaining sites in the route planning module based on cost from the new current location. This process continued until all sites were scheduled (Figure 3-7).

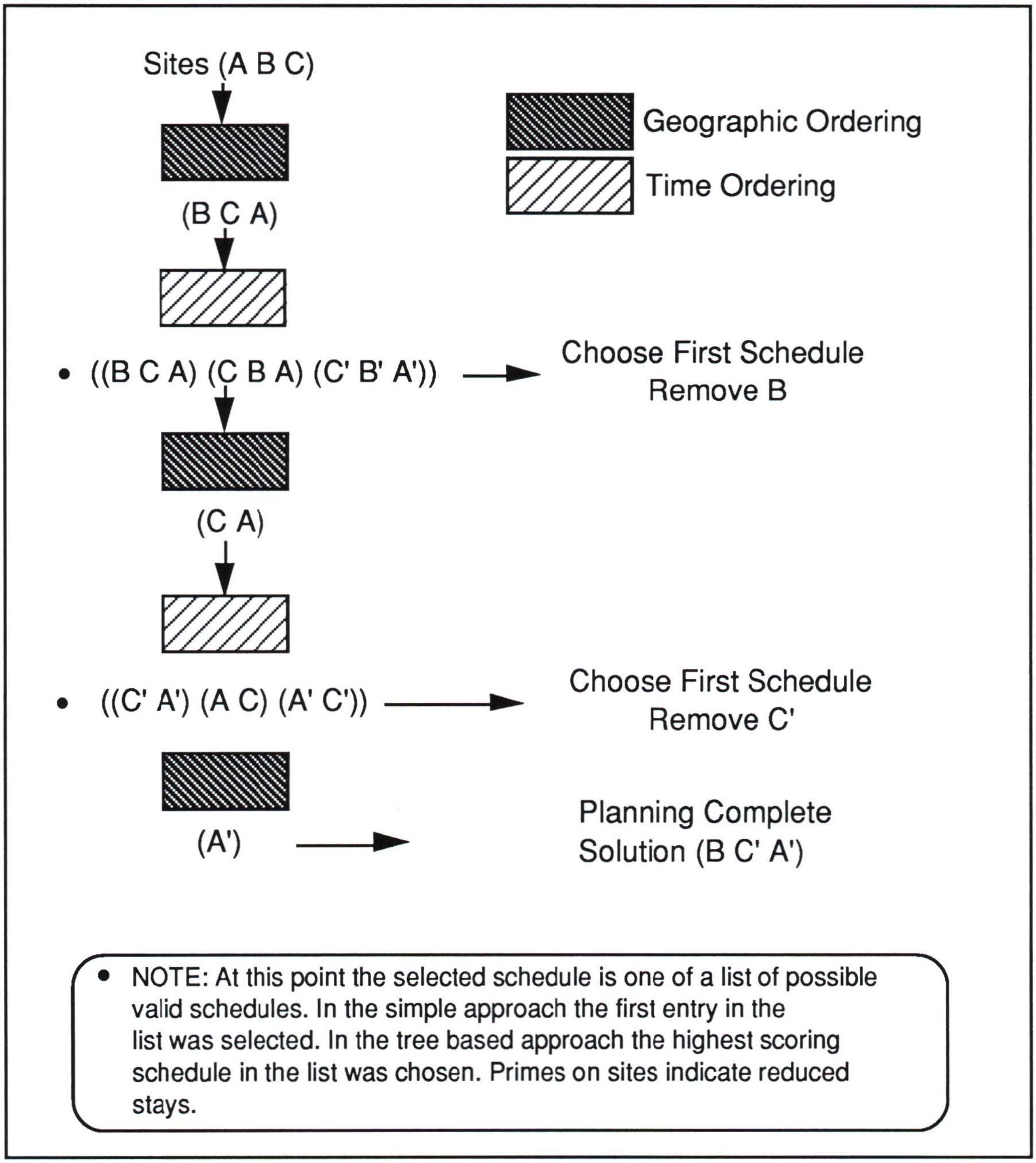

Figure 3-7 Combined Conceptual Planning

The advantage of the cooperative approach was the ability to consider both time and space from each site as the plan was incrementally constructed. The planner was guaranteed to be working with a visit that was part of a valid and complete schedule at each stage of the process. This prevented sites from being scheduled which would render the remaining visits impossible. The disadvantage of this iterative cooperative scheduler was the method of building and selecting the schedule at each stage. The conceptual planner built a list of several alternative valid schedules at each stage. Although one schedule may have been better than another, the planner simply selected the first one in the list to pursue. This approach was unsatisfactory because it occasionally resulted in plans with visits which were very short. As a result, a tree based schedule selection strategy was attempted.

3.4.4 Tree Based Selection of Alternative Plans

A tree based schedule selection strategy was implemented to improve the tourist information system after the basic cooperative scheduler was found to be inadequate. This permitted heuristic search algorithms to be used to select a schedule to pursue. It also enabled the planner to retain a visible record of decisions and supported re-planning in the event of plan failure. This approach was easily implemented but relied extensively on a scoring mechanism to determine which of the possible schedules to pursue.

The planner scored valid schedules which adhered to the geographic site ordering constraint higher than those which did not. A valid schedule is a product of scheduling at intermediate stages in the cooperative planning process. For example, in Figure 3-7 the list of valid schedules after the first scheduling attempt is ((B C A) (C B A) (C' B' A')). The preferred geographic sequence is (B C A). Thus, the second and third entries in the possible schedule list would score lower for this parameter than the first which adheres to the recommended geographic order.

Scores were also assigned for the length of stay recommended by a schedule. The reduction techniques applied by the scheduling module occasionally resulted in stays of very short duration. The list of schedules were scored on the basis of how much of the original duration of stay they retained. A schedule with short stays would score very low for this parameter.

This solution dramatically improved the schedule selection mechanism. However, it could not be retained because generating multiple alternative schedules in a search tree is not human-like. The approach taken in this work is to use human-like representations and processing to explore human cognition. Scheduling routines could be developed to establish an 'optimal' solution based on some arbitrary schedule rating criteria, however, the techniques and approaches used would not add to the understanding of human scheduling. In addition, the criteria for schedule rating may not consider the individual tendencies of each tourist.

During interviews with a local travelling salesman [Lero 89b] it was established that the outcome of his conceptual planning was only one schedule. In addition, the thinking aloud protocols in the errand planning experiments of [Haye 79a] concluded the product of planning was one valid schedule. The tourist information planner required a human-like solution to this problem, therefore the system should produce one valid schedule, not a list of acceptable alternatives.

3.4.5 Human-Like Conceptual planning

Several subjects were interviewed to develop a human-like model of conceptual planning. Their solutions supported the cooperative temporal and spatial nature of conceptual planning. During subsequent interviews a more detailed human-like model of conceptual planning emerged.

The subjects cooperatively blend time and space issues when selecting the next site to visit. They begin conceptual planning with an assessment of durations of stay of all sites to ensure that there will be a good chance that all sites can be scheduled. The subjects then focus their attention on high priority visits before proceeding with the remainder of the sites. Once priority visit times are determined, the subjects opportunistically schedule other sites before each priority site, starting with the first (earliest) priority site. When no sites can be fit before the first priority site, the site is scheduled and the impact this has on the geographic constraints is considered. This process continues with the other priority sites until all priority sites are scheduled. Any additional sites which remain are scheduled by the other specialists in the concept planning blackboard. The result of this approach is a plan which is constructed hierarchically, with temporal and spatial constraints satisfied by every new conceptual planning act.

The stay time assessment undertaken at the outset of conceptual planning relies on priorities to determine which sites can be visited for a reduced period of time. This list begins with appointments, and includes other high priority tasks such as meals and accommodations, and concludes with shopping, touring, and general interest sites as the lowest priority visits. The priority list ensures the duration of stay of an appointment is never reduced. Only non-appointment sites are eligible for reduction in stay time constraints.

After a priority site is scheduled the planner must consider the impact this has on the geographic sequence of the visits. If an appointment is a long distance from the current location but must be scheduled first due to time constraints, the planner must re-order the remaining sites to visit from the appointment location after it is scheduled. On the other hand, if an appointment is the closest site and is the first to be scheduled, the planner does not re-order the remaining sites to visit. A similar rule applies for food site scheduling.

The initial stay time assessment is designed to ensure that sufficient time exists in the planning period to permit all sites to be visited. This may not be an effective strategy when problems include appointments and rigid time constraints (Figure 3-8). This is primarily due to the placement of appointments and the resultant fragmenting of the time line. To overcome this limitation the subjects implement an additional step, constraint relaxation 'on-the-fly'.

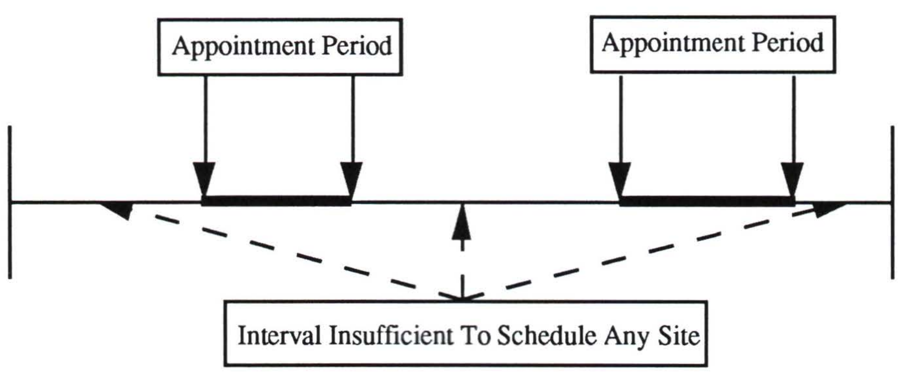

Figure 3-8 Fragmented Time Line

'On-the-fly' approaches begin with the reduction of the duration of stay of low priority tasks. For example, tourist attractions and shopping sites are the first site classes to be reduced to fit the sites in the available time frame. High priority sites such as accommodations and food sites are not considered at this stage. In addition, no appointment site can ever have its duration of stay reduced because the tourist has specified this time requirement at the outset of planning.

Human subjects can implement this approach opportunistically by examining the low priority task nearest to the current planning position. The local constraints of all low priority sites are reviewed until a valid schedule is developed. Reducing stay times is permitted only until stays reach a minimum allowable limit.

If the scheduling problem is difficult, as in the case of a badly fragmented time line, the local stay time reduction approach will not succeed. To overcome this, the subjects implement other 'on-the-fly' solutions which include: adding time to the planning period, matching sites to planning windows, permitting food sites to extend beyond the end of the planning period, or dropping low interest sites.

The result of this human-like constraint relaxation planning is a single, valid schedule where all sites which remain are allotted an arrival time. The outcome is not scored, selected or fixed since all factors of interest to the person making the plan have been considered during planning. The approach is illustrated in Figure 3-9.

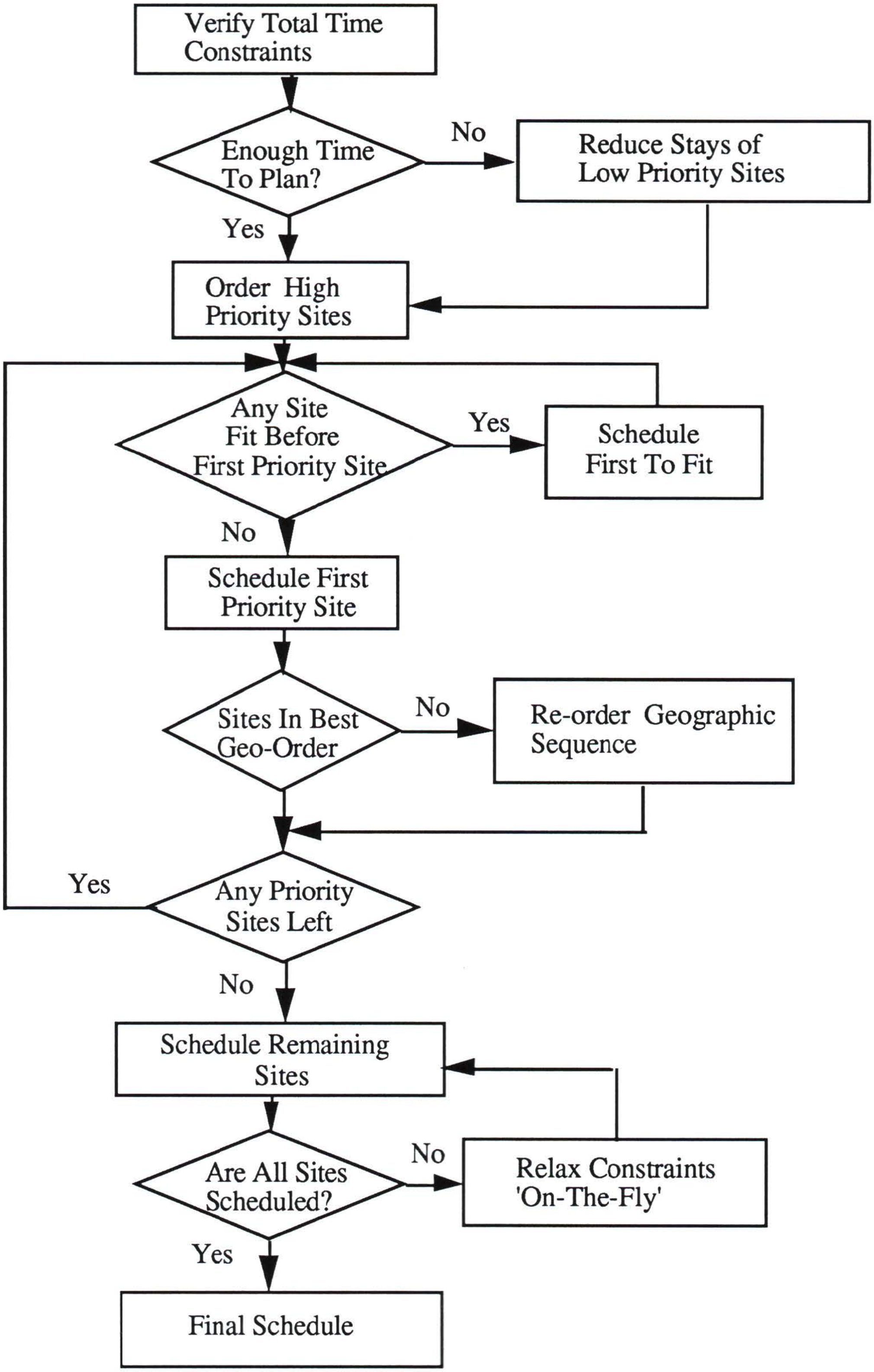

Figure 3-9 Human-Like Conceptual Planning

This human-like conceptual planning has been implemented as a blackboard within the concept planning expert. The specialists are designed to perform the various functions required to build a schedule in the approach illustrated in Figure 3-9. The details of this blackboard are discussed in the next section.

3.4.6 Concept Planner Software Operation

3.4.6.1 General

The concept planner blackboard incorporates many KEE features. It begins with an active value in the concept object which was created at the end of task planning. This code geographically sequences the sites to visit. This active value alters the state of the top level planning blackboard so that the route planner can be activated. The route planner recognizes the requirement for an ordered list of sites and activates a site ordering module to sequence them. A more detailed description of the code which accomplishes site ordering can be found in Section 3.5.

The global time constraints are checked prior to the activation of the concept planning blackboard. The global constraint verification function will prompt the tourist to alter the problem specifications if a site never opens in the defined planning period, if an appointment is desired when a site is not open, or if appointment times conflict. This is accomplished interactively by changing the time frame, appointment times, or visit list.

Global constraint verification includes the process of ensuring sufficient time is available in the time frame to accommodate all visits. All non-appointment sites will have their duration of stay reduced proportionally until all sites fit the time frame. This ensures longer activities will be allocated a greater portion of the planning period than sites which require little time.

The usual duration of stay of a site is a property of the visit object, and is heuristically defined by visit class. For example, a shopping site usually requires 120 minutes, a banking site 45 minutes and a restaurant 90 minutes. These times were established by estimating the average duration of stay of a tourist for each site class. Further research would be required to experimentally substantiate these values.

A simple delay rule set is invoked at the start of planning. These rules examine the available planning period and appointment times as well as the start time and the opening time of the closest site to the starting location. The rules determine whether it is possible to delay the start of planning until the closest site is open and still accomplish all sub-tasks without reducing their usual duration of stay. If a delay is warranted the planning period is altered and the plan assessor informed of the change. This rule set improves geographic plans since retracing a route to the closest site is not required if the start is delayed until that site opens.

One additional global constraint check is performed prior to activating the conceptual blackboard. A tourist decision is required if a police site is included in the planned visit list. The tourist is requested to accept the rule that police sites be visited at the very beginning of the planning period, regardless of geographic constraints. The tourist may reject the rule if the visit is not an emergency, in which case the police site will be incorporated into the conceptual plan where it best fits the temporal and spatial constraints of the problem.

The police site processing module is designed to deal with all emergency tasks present in a full sized knowledge base. If a tourist requests a visit to any site in the emergency visit class, for example, hospital, fire hall, doctor or dentist, then the system will prompt the tourist to accept the rule that this class of site must be visited first. If several visits in the emergency class are indicated, the tourist will be prompted to prioritize these as well.

3.4.6.2 Conceptual Planning Method

The conceptual planning blackboard is activated by a method entitled GEO-SCHED which is located in the concept plan object on the top level system blackboard. This method coordinates the cooperative human-like priority-based conceptual planning described in Section 3.4.5 and illustrated in Figure 3-9. The method consists of a conditional test which examines the conceptual planning problem. It determines which conceptual planning specialist to invoke depending on the state of the plan. The conditional test represents the control strategy of this blackboard.

The conceptual planning problem is resolved by the cooperation of specialists. Each specialist deals with a specific planning situation: all appointments, food sites and some appointments, food sites and no appointments, some appointments but no food sites, and no appointments or food sites. Each case is handled by a specialist implemented as a LISP function. Cooperating with these specialists is the site sequencing module of the route planning specialist. The architecture is illustrated in Figure 3-10.

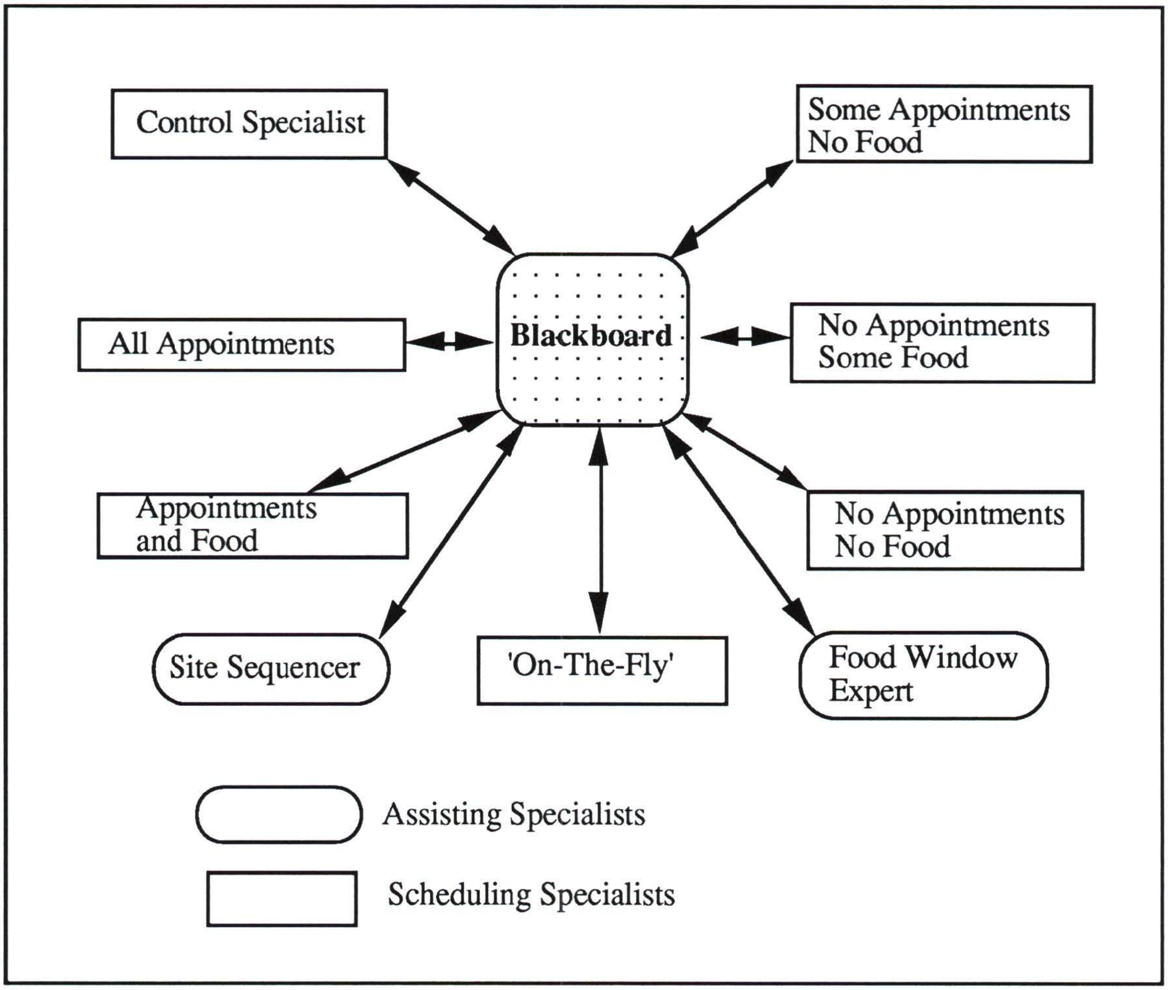

Figure 3-10 Conceptual Planning Blackboard and Specialists

The blackboard holds a simple list data structure in which all the sites to be scheduled are stored. Each site is represented by a structure which includes information extracted from the site object in the knowledge base. One element of this information is the scheduled arrival time of the site. The control specialist continues invoking specialists as long as there remains at least one unscheduled site on the blackboard. The details of each specialist are given in the following sections.

3.4.6.2.1 All Appointments

The planning task is quite simple when only appointments are to be considered. The specialist has no alternative but to schedule each site at the indicated starting time for the specified duration of stay. This expert quickly places a final schedule on the concept planner blackboard. The control specialist retrieves the final schedule and posts it to the top level planning blackboard to activate the route planner.

3.4.6.2.2 Food Sites and Appointments

A mixture of appointments and food sites to visit is a complex conceptual problem. Food sites are tasks which have a great impact on a conceptual plan. Intelligently selecting when to branch out of a series of site visits to eat is an important feature of a conceptual planner. The food and appointment specialist must perform this task well to generate acceptable plans.

The priority sites (food and appointments) are first ordered by their time requirements. Food sites which are not appointments require a time window which is best suited to accommodate the visit. This is determined by a forward chaining rule-based expert (a separate functional entity in the conceptual blackboard architecture) which examines the available time and the desired meal activity to find a suitable meal window. For example, if lunch is to be scheduled and the available planning time is between 0900 and 1100 and from 1230 to 1700, the expert will return (1230 1400) as a suitable lunch time window.

Prior to scheduling the food or appointments the specialist must determine if any other site could be scheduled before the earliest of these priority tasks. The time available before the first (earliest) priority site is compared with the time required by each of the remaining sites, along with opening and closing constraints, to locate a site which may fit. The remaining sites are in a geographic order suggested by the site sequencing module of the route planner. A site which fits in the period before the priority visit is scheduled, removed from the visit list, the remaining planning time reduced, and the search process repeated. The first priority site is scheduled when no other site can be planned prior to it.

After a priority site is scheduled, the specialist considers the impact this has on the geographic site sequence. If a site is moved ahead in the plan due to time constraints, such as an appointment at a distant location scheduled early in the day, the recommended geographic order of visits must be revised. This revision is performed by the site sequencing module of the route planner which is available as a specialist in the concept planning blackboard (Figure 3-10). For example, the act of scheduling a food reservation for dinner at the end of the day should not cause any remaining sites to be reordered. However, a food site scheduled at the start of the planning day should be considered the new current location of the tourist and the remaining sites reordered in a geographic sequence beginning at the restaurant. These rules are implemented in a simple conditional clause.

The conceptual blackboard control specialist also invokes reordering of sites from the new current location after any of the other scheduling specialists has returned a partially completed schedule to the blackboard. This policy ensures the specialists are always working with the best geographic sequence at all times.

The following example illustrates the conceptual thought processes:

Problem state:

Appointment at City Hall at 1100 for 30 minutes

Remaining sites (Bank Museum)

Current Time 1000

End Time 1400

Current Geographic Sequence to minimize travel (Bank Museum City Hall)

Thinking process:

"Appointment is at 1100, ok, do any sites fit before this? Yes, the bank site can be visited before the appointment, and the bank is closer to me than the museum. I'll schedule the bank site now."

After Scheduling the bank the problem state is:

Appointment at City Hall at 1100 for 30 minutes

Remaining sites (Museum)

Current Time 1045 (banking usually takes 45 minutes)

End Time 1400

Current Geographic Sequence to minimize travel (Museum City Hall)

Thinking process:

"Now, the Museum site does not fit in the time before my appointment at City Hall. I have to schedule the appointment next."

After scheduling the appointment the problem state is:

No appointments

Remaining sites (Museum)

Current Time 1130

End Time 1400

Current Geographic Sequence (Museum)

Note: if other sites had remained the geographic sequence would have been reordered based on distance from the appointment since it was moved ahead in the scheduling process.

The specialist continues this process until no appointments and food sites remain to be scheduled. It may conclude with all sites scheduled, or sites may remain to be scheduled by another specialist, for example the no appointments, no food specialist, as directed by the controlling knowledge source.

3.4.6.2.3 Some Appointments, No Food Sites

This specialist addresses a degenerative case of the some appointments and some food sites situation described in Section 3.4.6.2.2. It incorporates the same priority based scheduling scheme and the search for sites to fit in before appointments prior to scheduling the appointments. It also uses many of the same LISP functions. This specialist does not require the assistance of the food window selection expert system.

3.4.6.2.4 Food Sites and No Appointments

The work of the specialist responsible for food sites when no appointments exist is easier than the mixture of food and appointments. The specialist fits food sites into the windows selected by the rule-based food expert and incorporates other sites where the time and space factors are best satisfied. This process is very similar to the work of the specialist described in Section 3.4.6.2.2 and utilizes many of the same LISP functions.

3.4.6.2.5 No Appointments, No Food Sites

The specialist responsible for scheduling when no appointments or food sites are present uses the scheduling functions described in Section 3.4.7. This specialist returns a completed schedule to the blackboard if the scheduling functions are successful. If the functions fail, the control specialist invokes the 'On-The-Fly' knowledge source.

3.4.6.2.6 Relaxing Constraints 'On-The-Fly'

Relaxing constraints 'On-The-Fly' is the final action of the conceptual planning blackboard specialists. It is undertaken only when the remaining specialists have been unable to construct a successful schedule. The measures used by this specialist may result in a less than ideal schedule due to the difficult scheduling conditions the problem presents.

The first on-the-fly relaxation technique is to attempt to fit the remaining tasks to the available time. This function matches planning time windows with the remaining tasks. It may not preserve the geographic sequence of the sites because the time constraints of a fractured time line may force sites to be scheduled earlier or later than geographically suggested. A rule-based expert attempts to divide the remaining time equitably so the tasks to be scheduled will all fit. This approach may not succeed if the time line is badly fragmented.

Another constraint relaxed on-the-fly is the duration of stay of the remaining tasks. This is arbitrarily reduced to 30 minutes, since it is assumed that almost any visit can usually be accomplished in this time period. After this, scheduling functions are called to attempt to schedule these altered sites. If the scheduling functions fail again, the duration of stay of the remaining sites is reduced to the minimum allowable by site class. For example, a banking task could be completed in 10 minutes, but a restaurant task may not be reduced below 30 minutes. The scheduling functions are then called to schedule the reduced sites.

A failure at this stage usually indicates a very difficult scheduling problem. The specialist realizes this and attempts more drastic constraint relaxation measures. One is to add an hour to the planning time frame. The most drastic measure is to drop a low priority site. Site priorities are heuristically established in ascending order of importance as: shopping, sightseeing, and government sites. The planner continues to drop low priority sites until the scheduling functions eventually succeed, or it reports that the problem is very difficult and terminates. Some sites are never considered for removal from the visit list because they are essential. These include food, banking and shelter activities.

The 'On-The-Fly' specialist is the last resort in the conceptual planning process. If it is unsuccessful in building a valid plan the tourist has created a problem beyond the scope of the tourist information planner. The control specialist realizes this situation, aborts planning and advises the tourist of the difficulty it has encountered. The tourist may then initiate another planning attempt with altered problem conditions.

3.4.7 Scheduling Functions

The tourist information planner uses low level LISP functions and procedures (functions with side effects) to accomplish scheduling. These routines are called with a series of tasks and a time frame. They return these tasks scheduled at appropriate times within the time frame.

The functions which implement scheduling were originally written for the most simple conceptual planning approach described in Section 3.4.2. In order to support this approach the functions include the ability to schedule a combination of appointment and non-appointment visits. The more human-like cooperative conceptual blackboard planner isolates food and appointment planning separate from the scheduling of other sites. As a result, some features of the scheduling functions are not required by the human conceptual planner. The features do not detract from the performance of the planner and have been included in this description for completeness.

The tasks represent site visits. Their structure contains all of the information necessary to schedule a visit to a site. This information is stored in a fixed list format:

```
(name opening_time closing_time stay_time latest_arrival_time
scheduled_arrival_time hard_arrival_time)
```

Reading and writing functions are used to access and alter the values of any of these variables. A formal LISP structure was originally implemented for this information, however a simple list was found to be easier and more efficient (fewer function calls) to implement.

The information contained in a task list is obtained from the task object on the blackboard. This object is in the goal tree of the current problem and can be accessed as a leaf node descendant of the overall task goal (Figure 3-11). Each of the entries in the task list is obtained from the Scheduler_Working_List slot in the task object (see Table 3-1). The complete list of all tasks which constitute a problem is a list of all these structures. For example, this list corresponds to the unordered task list obtained from the hierarchy represented in Figure 3-11.

((The_Bay 0900 2100 120 2030 NIL NIL) (Wax 1030 1800 120 1700 NIL NIL) (Ports 1200 2300 90 2200 NIL NIL))

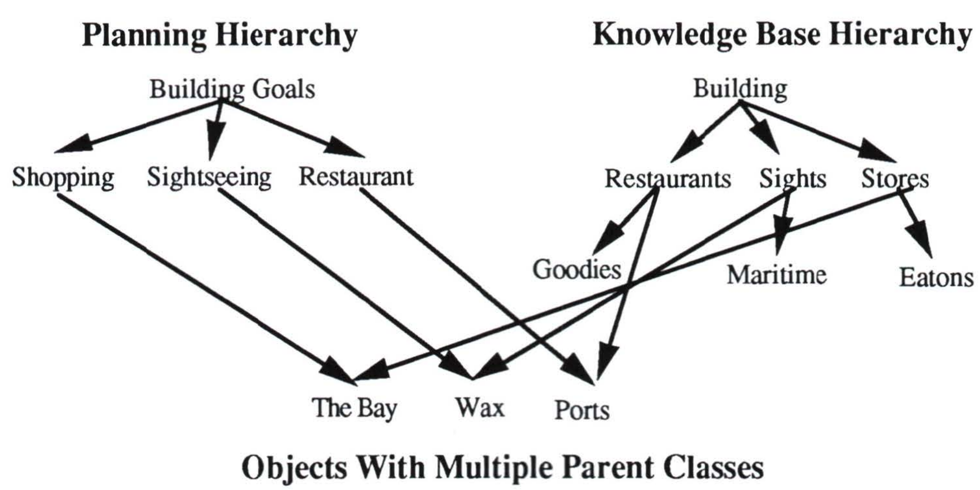

Figure 3-11 Example Task Tree

The top level scheduling function is TASK-SCHEDULE. This function is called with a geographically sequenced list of tasks to schedule and a time line in which they are to be scheduled. This function is recursive and stops when no tasks remain to be scheduled. It begins by finding any task which has been identified by the tourist as an appointment task (a non-NIL entry in the hard_arrival_time position). Appointments are scheduled by changing their scheduled_arrival_time entry to the value found in the hard_arrival_time position. Once a task has been scheduled the time it will take is removed from the available scheduling time line. The scheduled task is then consed onto a recursive call to TASK-SCHEDULE with the remaining tasks and reduced time line.

A time window in which to schedule the first of the remaining tasks is required if there are no appointments (or they have already been scheduled). This is accomplished by the function BUILD-ANY-TIME-WINDOW which is called with one task and the remaining time line. The time line is represented as a list of time frames. For example, ((0900 1000) (1130 1500)), where the period between the first and second entries in a time frame represents the time available for scheduling tasks. In this example the planner may schedule a task between 0900 and 1000 or between 1130 and 1500.

BUILD-ANY-TIME-WINDOW considers five possible scheduling conditions. It recursively examines each time frame in the time line until it finds a condition which is satisfied.

In Condition 1 the task site opens during the time frame, closes before the end of the time frame, and the opening time plus the duration of stay is within the time frame. In this case the time window returned is from the opening time of the site to the opening time plus the duration of stay.

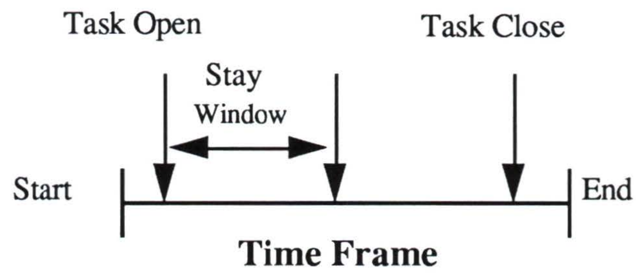

Figure 3-12 Condition 1

Condition 2 occurs when the task site opens after the start of the time frame, closes after the end of the time frame, and the opening time plus the duration of stay is within the time frame. The window returned is the same as Condition 1, from the opening time to the opening time plus duration of stay.

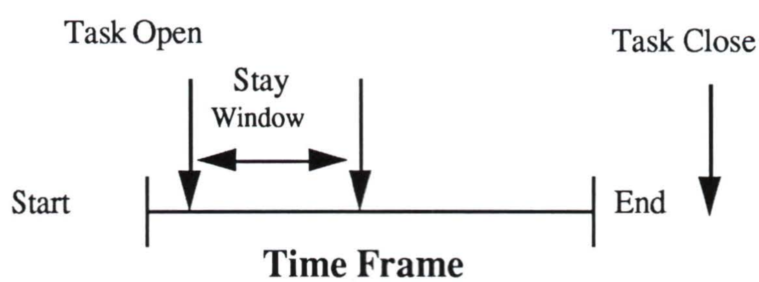

Figure 3-13 Condition 2

In Condition 3 the site opens before the time frame begins, and the start of the time frame plus the duration of stay is less than the end of the time frame. The task fits the time frame and the window returned is from the start of the time frame to the start of the time frame plus the duration of stay.

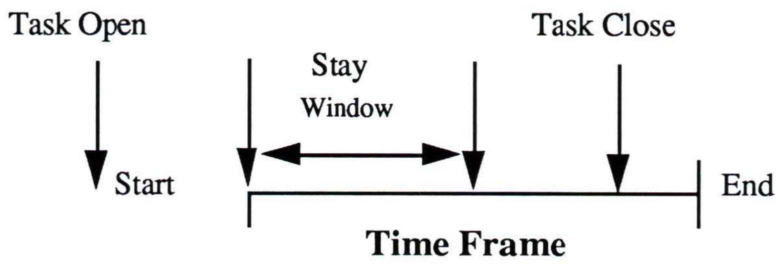

Figure 3-14 Condition 3

Condition 4 is valid when the ending time of the time frame minus the duration of stay is greater than the opening time of the task. The time frame must also end before the site closes and the end time of the time frame minus the duration of stay must be greater than the start of the time frame. The time window returned is from the end of the time frame minus the duration of stay to the end of the time frame.

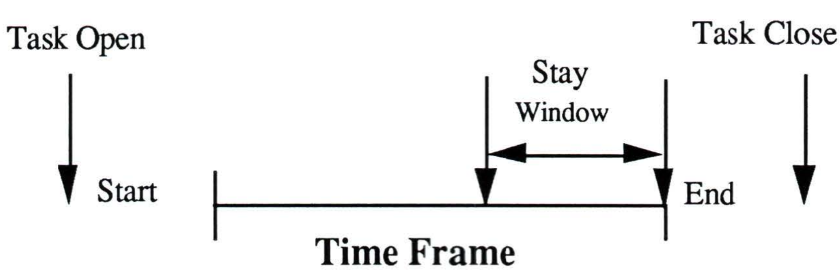

Figure 3-15 Condition 4

Condition 5 considers the problem where it is impossible to schedule an activity so that the required duration of stay may be accommodated. In this situation the function capitalizes on the knowledge that there is a latest_arrival_time associated with every site to be scheduled. This information states the absolute latest time that a site may be scheduled. For example, a banking site may usually require 45 minutes to be accomplished. If it is impossible to arrive at the bank with this much time left in the time window, it may be possible to arrive with less and still accomplish the task. For a banking task, arriving just prior to closing is still acceptable. If a bank closes at 1500, the latest_arrival_time is 1459. Condition 5 builds a window from the latest_arrival time to the closing time.

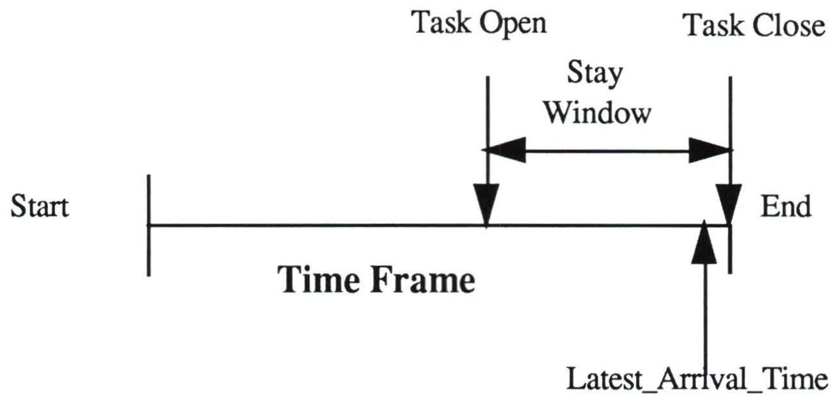

Figure 3-16 Condition 5

BUILD-ANY-TIME-WINDOW consists of a COND statement with these five conditions in numerical order. The first condition satisfied will be implemented, even if more than one is true. By ordering the conditions in this manner the total effect of the function is to achieve a first fit schedule, preferring to schedule tasks as early in an available time frame as is possible, despite the fact that each individual condition implements a separate scheduling strategy of its own. A last fit strategy could be achieved by rearranging these cases with Condition 4 or Condition 5 first.

The scheduled_arrival_time of a task is set to the start of the time frame found by BUILD-ANY-TIME-WINDOW. The time line for the entire scheduling problem is reduced by the function SCHEDULE-ANY-TASK which cuts the time window required by the newly scheduled task from the problem time line. This results in a fragmented time line as more tasks are scheduled.

The scheduled_arrival_time of a task is set to NIL if there was insufficient time to schedule the task in the selected window. TASK-SCHEDULE then recurses with the remaining tasks and the time line as it was before the unsuccessful scheduling attempt. If the task was scheduled TASK-SCHEDULE recurses with the remaining tasks and the reduced time line.

TASK-SCHEDULE returns the list of tasks it was originally passed with the scheduled_arrival_times as specified during planning. An unscheduled task will have NIL in its scheduled_arrival_time slot. If the planning list contains sites with NIL scheduled_arrival_time values after all scheduling cases in the loop have been addressed, the 'On-the-fly' relaxation process attempts to creatively address the problem.

The functional dependencies of the scheduling functions are illustrated below:

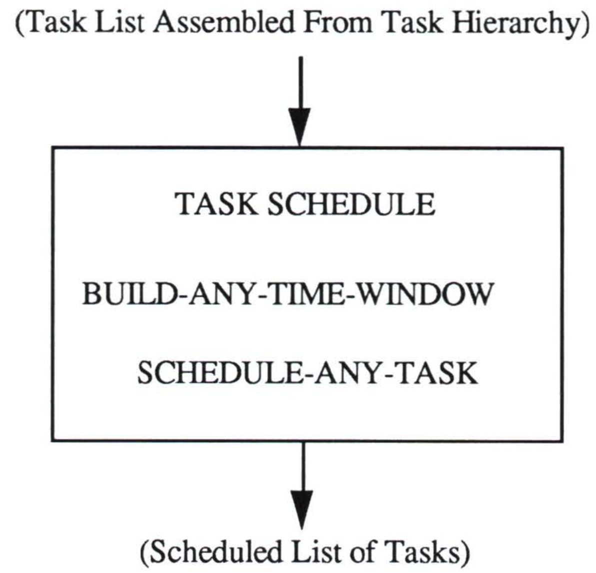

Figure 3-17 Scheduling Functions

3.5 Route Planner

3.5.1 Process Description

Route planning is the final geographic planning operation. The planner must determine the 'best' route to follow to link a series of sites in the sequence defined by the concept planner. People plan routes by looking at a paper map or by consulting their cognitive map. In either case, precise distance measuring is rarely undertaken. The sites are linked together by routes which satisfy simple costing desires such as what looks like the simplest route to follow or the shortest one to take.

The concept planner has constructed a site visit sequence which satisfies all the time and space constraints of the problem. The route planner must locate the best route to link these sites together.

3.5.2 Route Planning in a Restricted Domain

The network of streets in a human cognitive map contains very little detail. As a result, a search algorithm which is able to work with this limited information in a human-like fashion is required. The algorithm must be able to find paths without cardinal information or Cartesian cost estimates. It should be able to cost paths by counting intersections traversed or turns taken.

The branch and bound algorithm can be used if no estimate of distance from a point to a goal is known. If this approach were to be applied to the problem of moving from a source located at Fort and Wharf to a goal at Superior and Oswego, many difficulties would arise. From the source it must generate all successors, namely (View Wharf), (Fort Langly), and (Broughton Wharf) [Items 1,2,3 respectively in Figure 3-18] and then cost each generated path. The least cost path to pursue towards the goal must now be selected. In this case choosing a better one to follow becomes impossible since they all cost the same. Without a discrimination capability the algorithm becomes a British Museum approach which is very inefficient and not human-like.

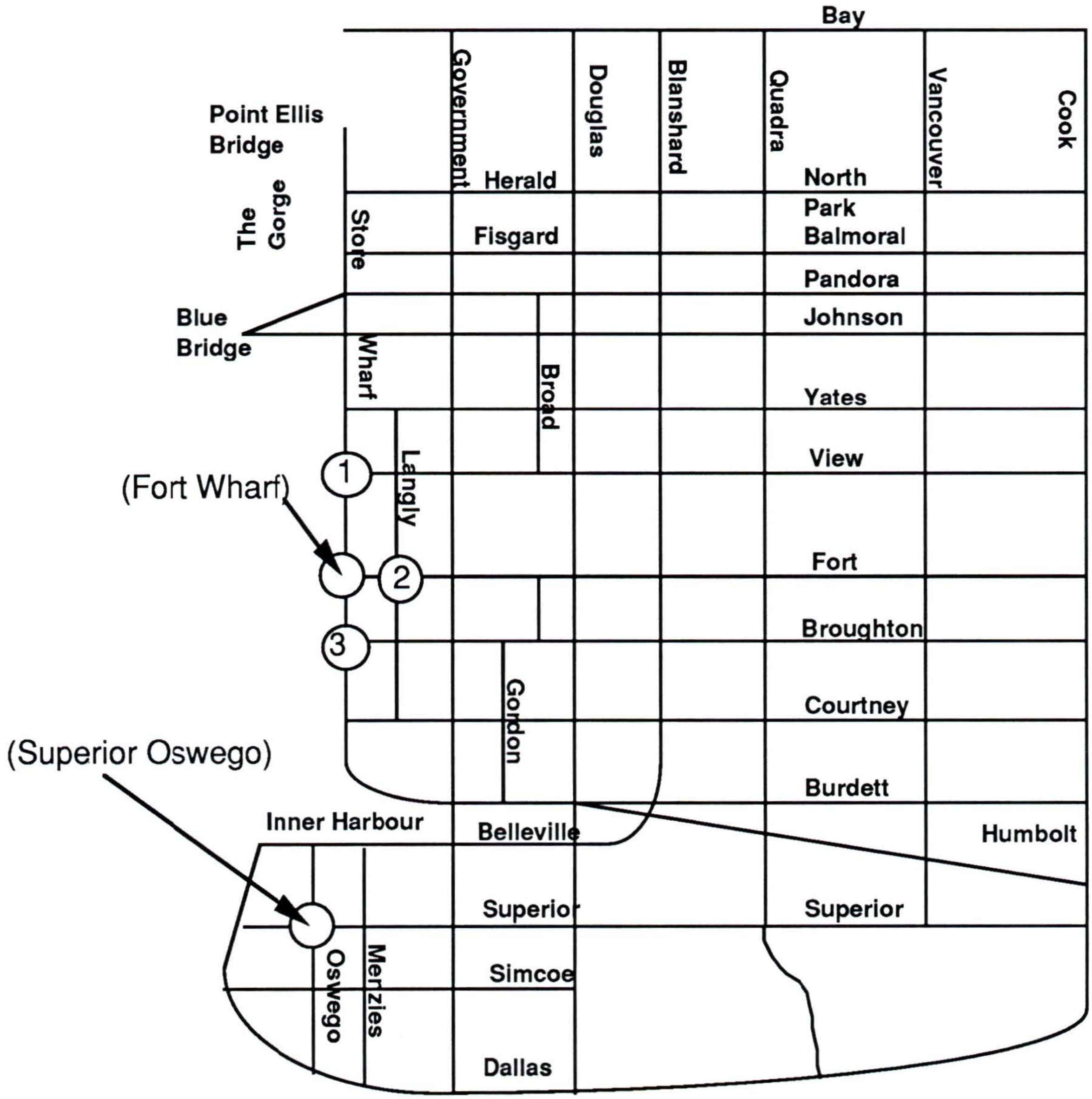

Figure 3-18 Successors of (Fort Wharf)

A different technique results if the same problem is approached from a human perspective [Lero 89b]. The first step is to find a common point between the source and goal and move to it since neither the start nor goal are on streets which intersect. Following Fort to Government is a reasonable move since Government also intersects with Superior, a street on which the goal is located. The remainder of the path is a move down Government to Superior, then along Superior to the goal.

This technique of finding a path relies on the generation of sub-goals from original goals until a sub-goal is found which represents an intersection between common streets. At that stage a path can be found to the goal. If two points share a common intersection then a path between them exists which runs through the common point.

In order to reinforce this point, consider the problem of a source located at (Johnson Government) and a goal at (Yates Vancouver) (Figure 3-16). These two points share two common junctions: (Johnson Vancouver) and (Yates Government). This approach would result in the generation of two paths, one through (Johnson Vancouver) and the other through (Yates Government). Two possible paths exist between source and goal, both of which are valid. The route planner must select one of these for the tourist to follow. The options are costed (see Section 3.5.5) and the least cost path returned.

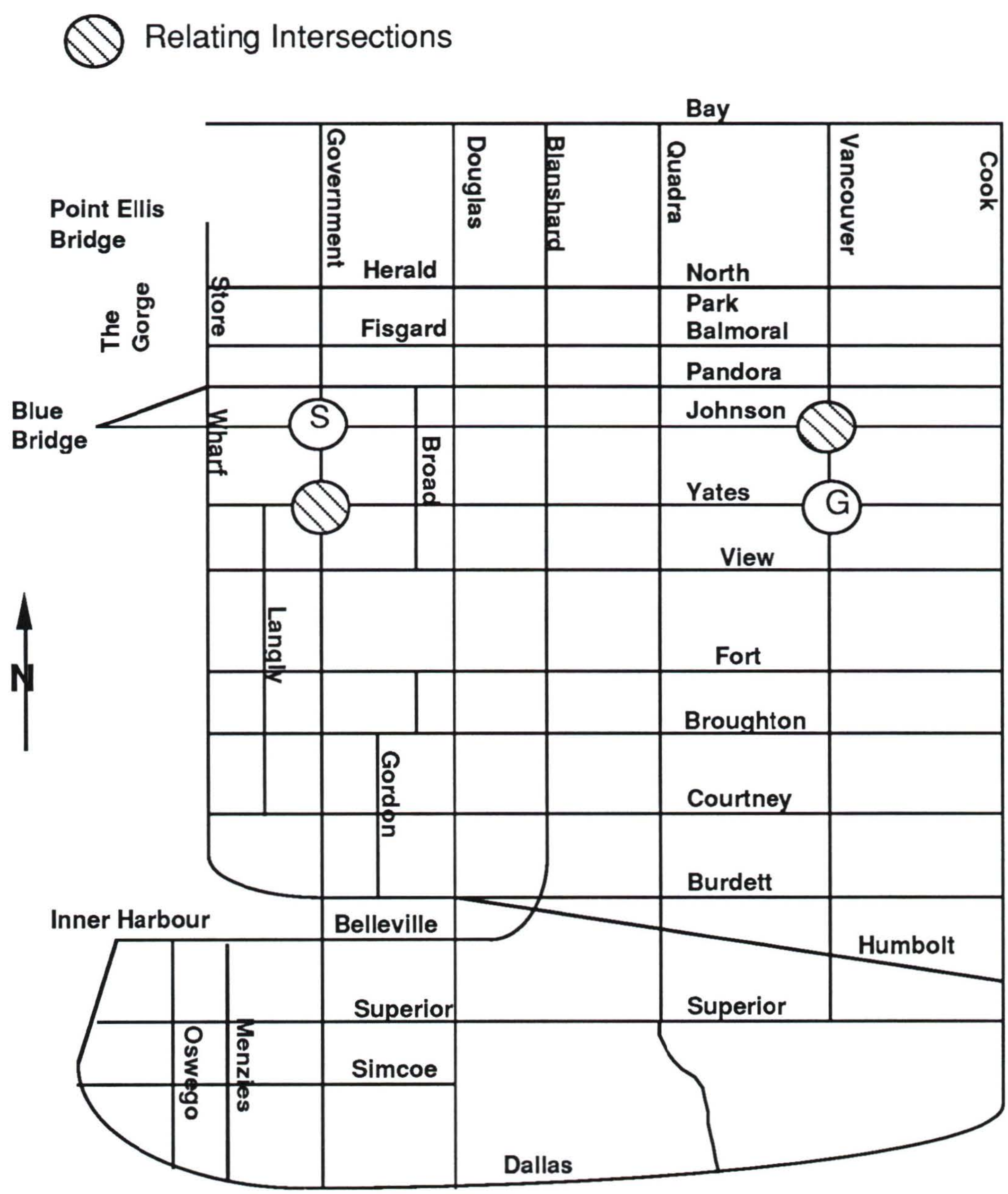

Figure 3-19 Relating Intersections

3.5.3 Street Representations

In Section 2.4 the road network of a human cognitive map was described as a collection of street objects in a spatial context, with associated descriptive features such as length, adjacent objects and intersecting streets. The street name description was established as the principle manner in which people describe routes for tourists to follow.

The human path planning algorithm requires the streets to be represented as list structures. The lists are formed by examining the street on the map and determining whether it is a horizontal street, a vertical street, or a complex street (a winding road). Horizontal streets are entered as intersecting street names beginning at the most Western end of the street. Vertical streets are formed as North to South lists of intersecting streets. Complex streets are decomposed into sub-streets components which are vertical or horizontal. After conceiving this representation it was found that GeoRoute [Rous 89] adopts a similar approach. The GeoRoute technique is to represent streets as links in a segment encoding scheme, although not for the same purposes as this research.

The human cognitive map is a pragmatic simplification of the true street network. The network used in the tourist information planner is the same. Simple streets are tagged either horizontal or vertical when they are entered. Complex street components are similarly tagged and named with an extension.

In Figure 3-19 Dallas Road is a complex street. It is entered as two simple sub-streets Dallas.1 and Dallas.2. Dallas.1 is the vertical component from Belleville to Simcoe. Dallas.2 is the horizontal street from Simcoe to Cook. This simple decomposition of complex streets greatly simplifies the process of entering a map network in the system. Any network may be similarly decomposed and quickly input for rapid path planning using the human path planner.

The tagging of streets as horizontal or vertical is used to provide a script description of a path to the tourist. The street construction convention and the directional tag are sufficient for the system to establish whether a turn in a path is left or right. This information is presented in a text window upon completion of planning. For example, the route:

> ((Fort Cook) (Fort Vancouver) (Fort Quadra) (Fort Blanshard) (View Blanshard) (Yates Blanshard) (Yates Douglas))

must be described to the tourist in a more simple manner. A turn analysis function examines this route and determines by street name changes that the route includes two turns. These are from (Fort Blanshard) to (View Blanshard) and from (Yates Blanshard) to (Yates Douglas).

The direction of turn function examines the first turn and notes that the tourist is travelling West on Fort street and then North on Blanshard. It knows a traveller changing direction from West bound to North bound must make a right turn. Similarly, the traveller now proceeds from North bound on Blanshard to West bound on Yates, making a left turn. The script which results from this analysis would read as follows:

"Take Fort street until you reach Blanshard. Turn right on Blanshard and proceed to Yates. At Yates and Blanshard turn left and carry on to Yates and Douglas, your destination"

The similarity between the computer generated script and a description a human might generate for the same route makes the system appear even more personable to the tourist.

The actual implementation of the router in software is a simplification of the methods described in this section. No vertical or horizontal tags are incorporated in the current version because a convention of listing intersections as the horizontal street followed by the vertical street was adopted. This is an artificially imposed system constraint which can be eliminated by adding the vertical or horizontal tags to the street objects in the knowledge base.

The process of determining whether a street is vertical or horizontal is very simple for the example city. In more complex networks the angle a street makes with a northing or easting can be used to classify the street. Streets closer to an easting than 45 degrees are vertical, those closer than 45 degrees to a northing are horizontal. Streets exactly 45 degrees can be arbitrarily designated vertical or horizontal, as long as the convention is consistent.

3.5.4 Reducing Possible Paths

The router is capable of locating many possible paths between a source and goal. In order to locate at least one path the router may be required to expand a large number of successors in complex city layouts with dead ends and many one way streets. Conversely, well laid out block structured cities will not require an extensive search for routes.

The search algorithm can be augmented by a heuristic which stops successor path generation once a certain number of distinct routes between a source and goal have been found. Three routes were originally chosen as the limit because human subjects rarely consider more than this number when conceiving paths, and because at least three were required to enable other layers of the planner to consider options during problem solving.

After experimenting with three routes it was discovered that paths returned within this limit occasionally linked the source to the goal from only one side of the source. This could cause problems when complex maps include many one way streets. It was then decided that the search process should examine all possible directions around the source to the goal. The search continues until at least one, and up to four, paths are generated. This wider search also enables the path planner to operate in many varied city configurations. It is analogous to a human examining all the possible directions away from an intersection before deciding the best way to proceed.

3.5.5 Cost Estimation

A simple costing function was devised since no distance information was available to the router. The cost between any two points was estimated to be equivalent to the number of intersections traversed in the route. This cost function is representative of true conditions within a city since most intersections include traffic control features, such as lights and stop signs, which contribute to the time required to reach a destination.

Another costing function was devised based on a desire to provide tourists with the most easily followed route to a goal. Tourists are easily disoriented when making many turns in a strange city. The fewer turns in a route, the less likely the tourist would be to loose his or her way. The function simply counts the number of turns in a path, and can be used to select a minimum cost path on this basis.

3.5.6 Routing Algorithm

The formal routing algorithm described above is:

Begin

If source and goal are the same,

then return a one-intersection route.

else

DO (exit when the number of paths generated is four or all interrelationships have been examined)

Begin DO

If no relationship between source and goal exists, then generate successors until relationship is found between the source or its successors and the goal or its successors.

Generate path through common point from source to goal, store path in path list.

End DO

End

A hierarchical representation can be used to improve the performance of the router in cases where the search between two points would involve traversing a very dense network of streets. The router can quickly determine how to move from one cluster to another by planning in an abstracted network of major roads and freeways. Afterwards, the same router could be used in the more detailed representation to finish the route planning process. This has not been implemented in the current system because of the small scale of the planning area.

3.5.7 Relationship Between Two Points

Two points in a map may relate to each other in several ways:

Case 1: Both points are on the same vertical or horizontal street.
Ex: (Johnson Government) and (Johnson Vancouver)

Case 2: Points share a common intersection.
Ex: (Johnson Government) and (Yates Vancouver)

Case 3: Points are on same street but the street name changes.
Ex: (Yates Wharf) and (Pandora Store)

The path planner works with these basic cases. If no case is satisfied by the source and goal, then the successors to the source are examined to see if they relate to the goal through any case. A successor to any intersection is defined as an intersection which is adjacent to the original and can be reached in one move from that intersection. If a successor to the source relates to the goal through one of the cases above, then a path exists from the source, through the successor, through the inter-relating point, and on to the goal. Similarly the source may relate to a successor of the goal, and a path could be built from the source, through the inter-relating point, through the successor to the goal, to the goal itself.

This approach first explores inter-relationships between the source and the goal. Then the successors of the source are related to the goal. After this, the source is checked against the successors of the goal. This cycle repeats, relating successors of the goal with successors of the source, gradually expanding the search space, until sufficient paths between source and goal have been generated.

This router is an efficient and effective path finder. In cases where both source and goal are on the same street, the path is immediately apparent to a human, and to the router as well. It quickly builds simple and efficient routes between sites which are related through the cases, and intuitively mimics human path planning. This router is distinct from other search algorithms because it closely resembles human path planning.

Constraints on the route to follow to satisfy the planning objectives of a tourist may be checked at many levels of planning. This route planner examines only one constraint during path generation, that of one way street traversal. If a route is suggested which traverses a one way street incorrectly, and the tourist is in a vehicle, then the router will discard that route and seek another. This highlights the importance of incorporating search through successors in order to by-pass one way streets and circumnavigate obstacles. Other constraints can be incorporated into the router in order to enable it to make even more intelligent routing decisions. They may also be left to higher level planners.

3.5.8 Routing Functions

The top level route planning function is called HELP-BUILD-PARTIAL-PATHS. This function is called with a source and goal intersection and returns a list of paths from the source to the goal. Each path consists of a series of intersections leading from source to goal.

HELP-BUILD-PARTIAL-PATHS calls the procedure BUILD-PARTIAL-PATHS which has the side effect of setting a global variable TEMPSEQ to the list of paths from source to goal. This list is checked to ensure that each path within it leads from the source to the goal by the function ENSURE-CORRECT-PATH. At the end of HELP-BUILD-PARTIAL-PATHS the verified list of paths is returned to the caller.

BUILD-PARTIAL-PATHS is a procedure which builds up the paths between source and goal by implementing the human-like routing algorithm. Each source and goal combination is examined to see if one of the interacting cases is satisfied. A path is produced for each case that is valid. If no interacting intersection is discovered, the successors to the source and goal are investigated in an expanding search space until at least one, and at most four, possible paths are found. In order to complete a search in all directions around the source for paths to the goal, BUILD-PARTIAL-PATHS is controlled by the path limiting heuristic which directs it to try and find four paths between the points. This ensures the first four successors to the source are examined, thus implementing a circumnavigation of the source in the search for paths.

An example of the successors of the intersection (FORT QUADRA) is:

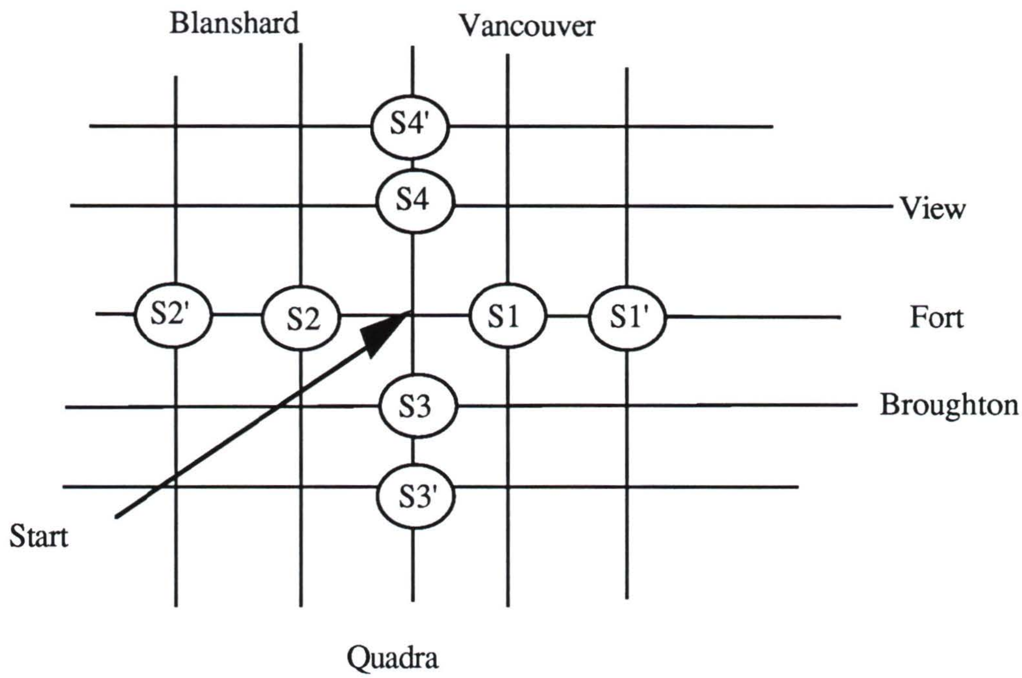

Figure 3-20 Successors of an Intersection

Where S1 is (FORT VANCOUVER), S2 is (FORT BLANSHARD), S3 is (BROUGHTON QUADRA), and S4 is (VIEW QUADRA).

Successor lists are defined for both source and goal:

(source S1(source) S2(source) S3(source) S4(source) S1(S1(source))
S2(S2(source)) S3(S3(source)) S4(S4(source) S1(S1(S1(source)))
S2(S2(S2(source))) S3(S3(S3(source))) S4(S4(S4(source))))

For example, in Figure 3-20 the intersection S1' is represented by S1(S1(source)) in the source successor list, and S2' by S2(S2(source)). The goal successor list is similar to the source list:

(goal S1(goal) S2(goal) S3(goal) S4(goal) S1(S1(goal)) S2(S2(goal)) S3(S3(goal)) S4(S4(goal) S1(S1(S1(goal))) S2(S2(S2(goal))) S3(S3(S3(goal))) S4(S4(S4(goal))))

These successor lists represent a search pattern for street relationships. The pattern expands out from the source or goal along both vertical and horizontal streets, as illustrated by the numbers in the following figure:

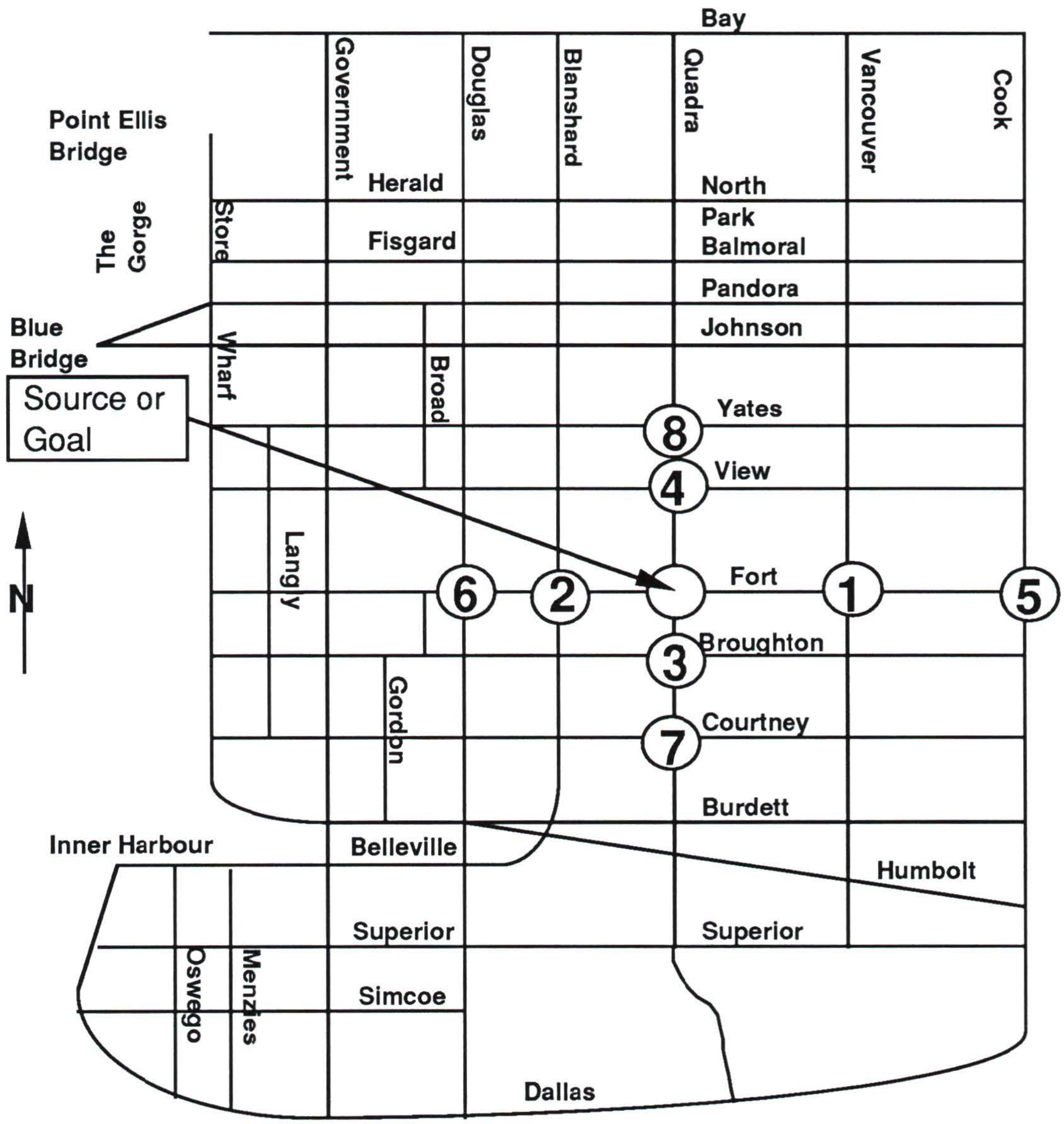

Figure 3-21 Successors of a Node

BUILD-PARTIAL-PATHS begins by examining the successors of the source (as per the source successor list defined above). It seeks a relationship between the intersections in this list and the goal intersection. The function which does the actual work is COMPLEX-BUILDUP which is called with the two intersections. If an intersection in the successor list (which includes the source or goal, depending on the case being examined) is found to relate to the source or goal, then a complete path from the source, through the successor of the source, through the common intersection, and to the goal, is returned.

COMPLEX-BUILDUP implements the actual intersection relationship processing tests. This function terminates when it finds at most three paths between the two intersections it is investigating, or when it has exhausted all test cases. A limit of three paths is applied at this low level of planning because the function BUILD-PARTIAL-PATHS is responsible to search all around the source and goal, not COMPLEX-BUILDUP. COMPLEX-BUILDUP implements the human-like process of finding up to three routes to follow in one direction away from a source. Only the shortest of the paths returned by COMPLEX-BUILDUP to BUILD-PARTIAL-PATHS is extended to connect the source and the goal. Figure 3-23 illustrates the combination search limits of both COMPLEX-BUILDUP and BUILD-PARTIAL-PATHS.

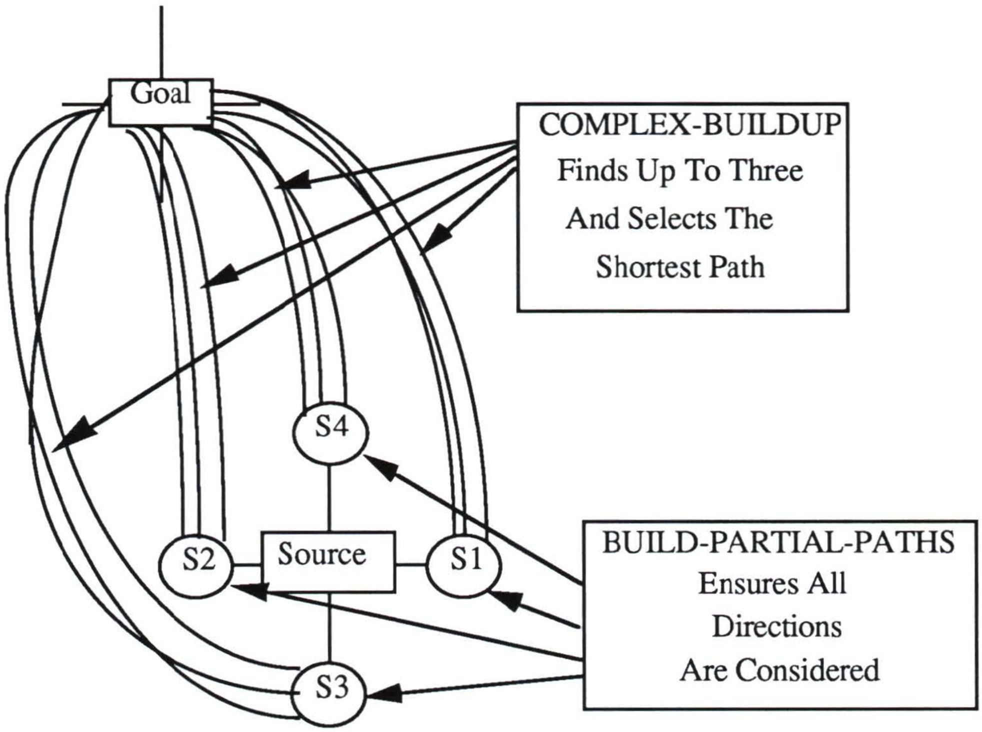

Figure 3-22 Effect of COMPLEX-BUILDUP and BUILD-PARTIAL-PATHS

The relationship testing cases are: (source refers to the starting intersection, goal to the end intersection for the call to COMPLEX-BUILDUP, not necessarily the source and goal intersections of the original problem)

Case 1: The source is the same as the goal so the planner returns a one intersection path.

Case 2: The source is on the same vertical street as the goal, hence it returns a path vertically along this street from source to goal.

Case 3: The source and goal are on the same horizontal street, it returns a path along the horizontal street from the source to goal.

Case 4: The vertical street in the goal list is a member of the horizontal street in the source. The router follows a path horizontally from the source to the common intersection of the source and goal, then vertically to the goal.

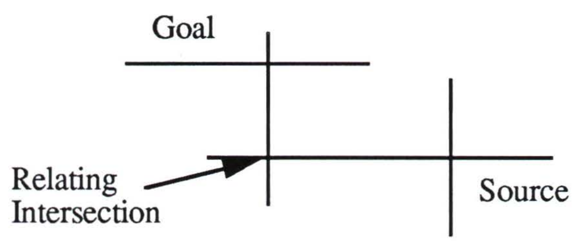

Figure 3-23 Case 4

Case 5: The vertical street in the source is a member of the horizontal street of the goal. The path is from the source vertically to the common point and horizontally to the goal.

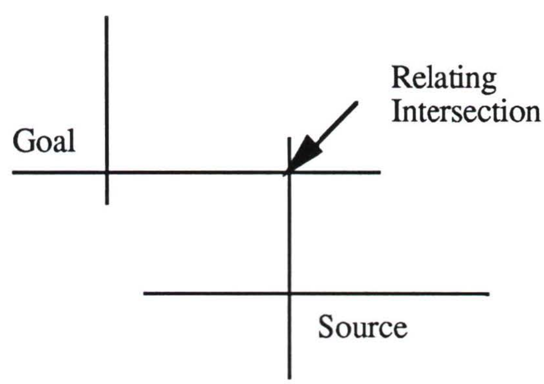

Figure 3-24 Case 5

Case 6: The street on which the source and goal lie changes name going vertically. The route is to move to the end of the first street, then along the second street to the goal. This process is based on the assumption that street name changes always occur at intersections. Artificial intersections can be created to accommodate situations where streets mysteriously change names in other locations.

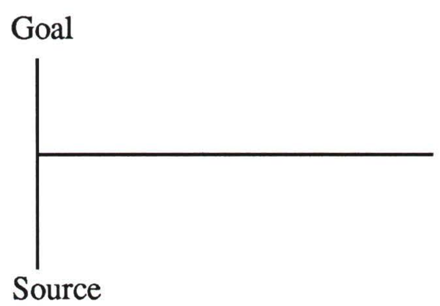

Figure 3-25 Case 6

Case 7: Same as case 6, but horizontally.

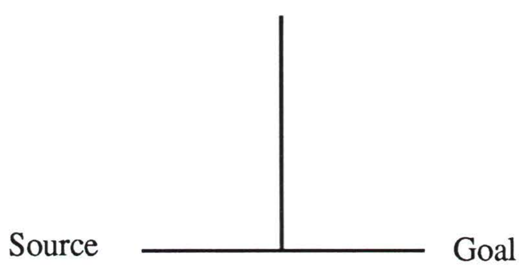

Figure 3-26 Case 7

Case 8: The first successor of the source relates to the goal as per case 4. The path is from the source, through the first successor, through the common intersection and on to the goal.

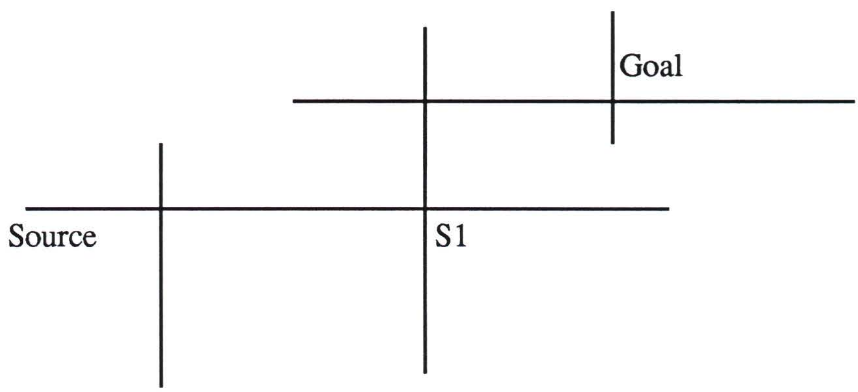

Figure 3-27 Case 8

Case 9: Same as Case 8 but for second successor of the source

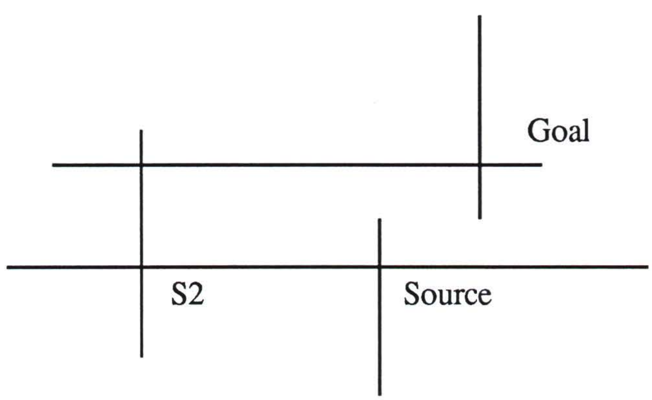

Figure 3-28 Case 9

Case 10: The third successor of the source relates as in Case 5. The path is from the source, through the third successor, through an interrelating intersection and on to the goal.

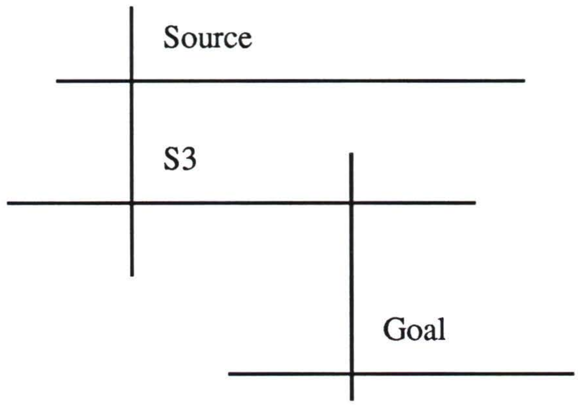

Figure 3-29 Case 10

Case 11: Same as Case 11 for fourth successor of source.

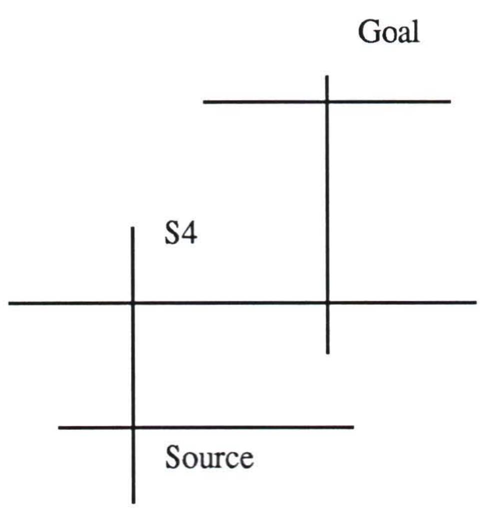

Figure 3-30 Case 11

Case 12: This case begins the process of searching for a relationship between the successors of the goal and the source. Case 12 is similar to Case 8.

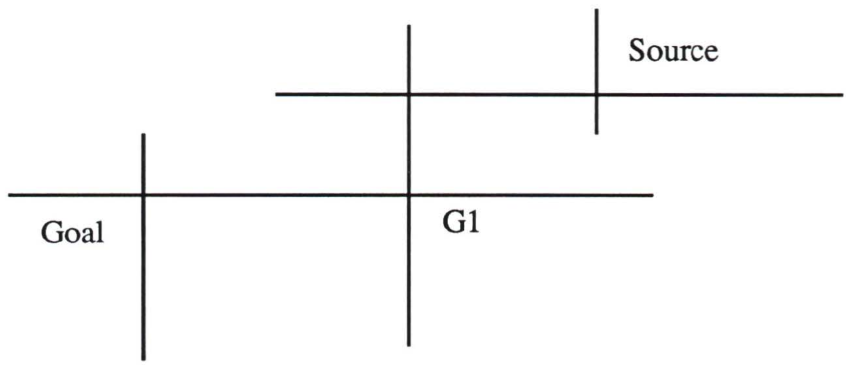

Figure 3-31 Case 12

Case 13: Same as Case 12 for second successor of the goal

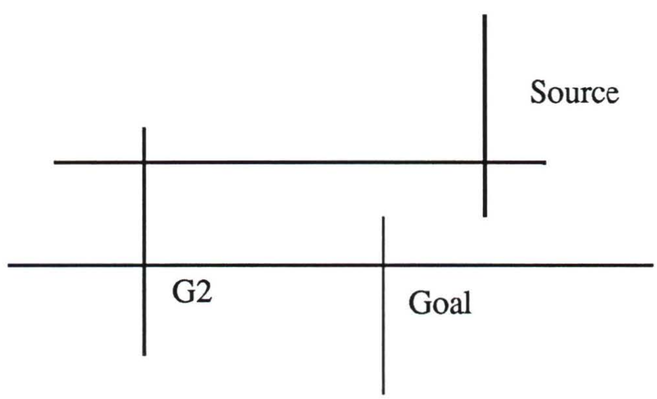

Figure 3-32 Case 13

Case 14: Same as Case 10 for third successor of goal

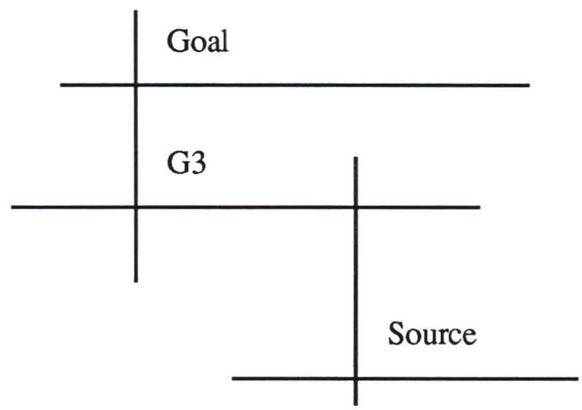

Figure 3-33 Case 14

Case 15: Same as Case 11 for fourth successor of goal

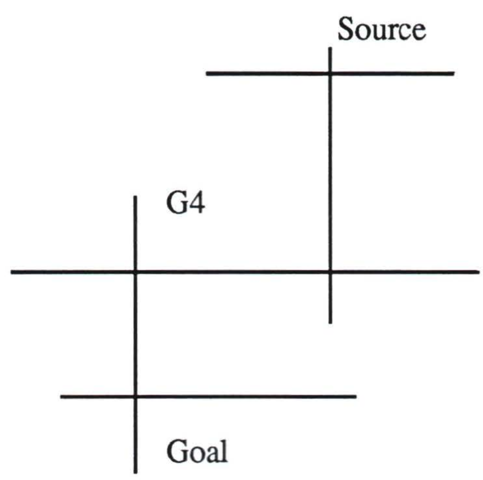

Figure 3-34 Case 15

The result of combining the search within COMPLEX-BUILDUP in these cases, and the work done by BUILD-PARTIAL-PATHS, is to methodically explore a search space out from the source and goal until at least four paths are found, or until the limits of the search space are encountered. Figure 3-35 illustrates the pattern of this search:

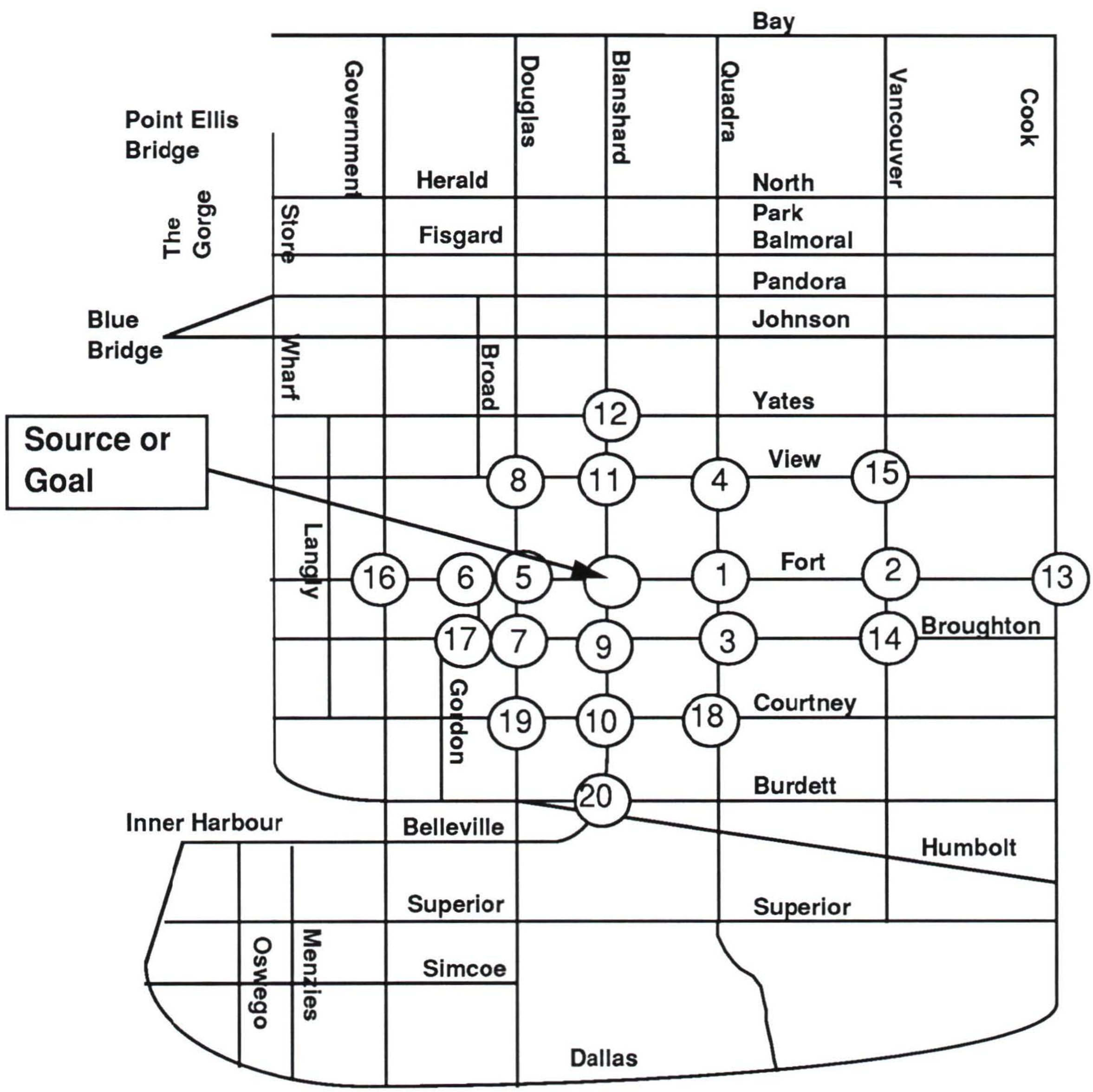

Figure 3-35 Combination Search Pattern

In the pattern described in Figure 3-35 site 18 is not located above site 6 because there is no vertical fourth successor to site 6 in the map.

The functional dependencies of the route planning functions are illustrated in Figure 3-35.

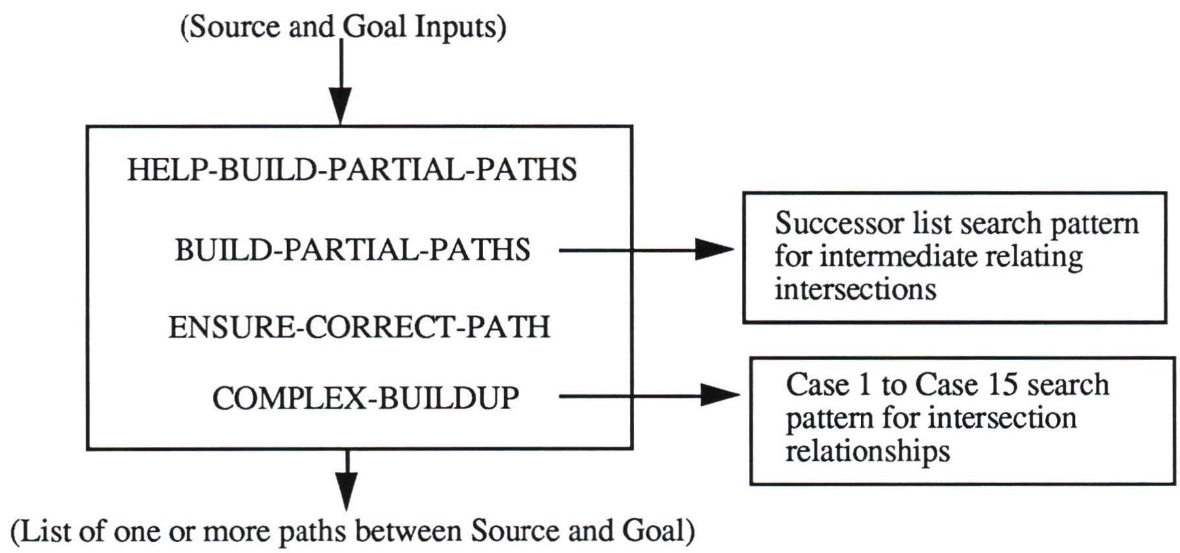

Figure 3-36 Route Planning Functions

3.5.9 Site Sequencing Module

The route planner is used to determine routes between sites. It is also responsible for ordering a series of site visits into a logical geographic sequence. The conceptual planner requires this ordered sequence of sites to balance time and space constraints when formulating a conceptual plan.

The code which implements sequencing of sites is a combination of rules and LISP functions. The rules begin the process by examining the state of the blackboard and determining what is required of the route planner. There are two options, either costing by closest site or costing by shortest site.

Closest site sequencing is implemented to enable the route planner to quickly establish a rough geographic ordering of visits. This method simply costs the path to each site in the list from the starting location. It uses a sorting function to order the list of sites from lowest to highest cost. Closest site planning is available should it be required but it is not currently used by any knowledge source. It may be of use when sequencing a large number of sites for clustering purposes during long range planning since it avoids the combinatorial explosion encountered in the shortest site sequencing functions.

Shortest site sequencing is performed by ordering the sites based on cost from the starting location, selecting the one which is closest to the start, putting that site first in the visit list, and removing it from the sites under consideration. This site is then designated the current location and the process repeats until all sites are sequenced on the ordered visiting list.

Shortest site costing sequences the sites by which is nearest to the current site each time. This begins with the starting location and proceeds until all sites are sequenced. The list produced is a better geographic plan than that which results from closest site costing, but the processing time is longer. The conceptual planner relies of the shortest site sequence as the primary geographic constraint since a path which follows this sequence will be close to an optimal geographic plan. Either of these geographic sequencers is sufficient for human-like planning. It is unlikely that a human would be interested in constructing a more optimal site sequence at this stage of planning since time constraints have not yet been addressed.

3.6 Chapter Summary

The task planner is a rule-based knowledge source which implements human-like goal directed backward chaining to locate visit classes and specific sites to visit from the knowledge base. Task planning attempts to mimic the high level human process of establishing a set of daily tasks.

Conceptual planning is a complex process which is implemented in a blackboard architecture within the top level blackboard. The scheduling specialists cooperatively and opportunistically apply a human-like scheduling process patterned after the techniques used by several subjects, including a local travelling salesman, and supported by the research in [Haye 79a].

The route planning algorithm is designed to operate with a restricted network which is modelled after the human cognitive map. The network is a simplification of an actual city map, the downtown core of Victoria, British Columbia, Canada. The development of this algorithm enhanced the human-like behaviour of the route planner.

This chapter has discussed the detailed implementation of a tourist information planner. The components specified in this description must interact to develop acceptable tourist plans. Global system issues along with system verification and validation are examined in the next chapter.

Chapter 4 Global Issues

4.1 Introduction

Each individual module of the tourist information planning system must be integrated into a complete package. There are some components which can be described from a system's perspective since they contribute to the overall effectiveness of the planner. These include the plan assessment module which summarizes the most important contributions from the planners. Re-planning is also a system level activity which is initiated by the tourist upon completion of a plan. The user interface provides the means for the system to extract information from the user and vice versa. Additionally, test protocols exercise individual system characteristics within the integrated package. After examining each of these system issues this chapter concludes with a discussion of points of interest which arose during system development.

4.2 Plan Assessor

The plan assessment module adds an explanation facility to the tourist planning system. The specialists make decisions during planning which are based on dynamic problem conditions. The plan assessor assembles these decisions and the associated conditions and explains the plan to the tourist.

4.2.1 Process Description

The plan assessor reviews the product of the planning process and provides the tourist with a high level description of decisions taken during planning. The assessor is activated by the completion of planning and interprets information compiled during the problem solving process. The comments are presented in a text window for the tourist to review.

The assessor requires information gathered at critical decision making points to provide valuable high level comments. This information is assembled in a separate assessor object on the blackboard. This object is designed like a plan rationale [Haye 81]. This has the advantage of centralizing all critical decision information for review by the assessor.

4.2.2 Plan Assessor Software

The plan assessor interprets planning information from the assessor object in the following areas: plan timings, appointment sites, food sites, geographic sequencing and scheduling. Each assessment is conducted by a separate LISP function which examines the contents of the assessor object and writes relevant interpretations to a text window. The assessment rules are conditional cases in the LISP function.

Plan timings are specified by the tourist in the initial interaction dialogue. The planner has the ability to adjust the start and end times of the planning window depending on the plan status. For example, the planner can delay a plan until the closest site to the start is open. This policy can prevent excessive route retracing during plan execution. However, the start of planning could not be delayed if this action would adversely affect the duration of stay of subsequent sites in the plan. The delayed departure rules are invoked by a LISP function at the start of planning.

The timing assessment process examines the specified time frame, the planned time frame and the comments collected from the delay function. High level comments regarding plan timings are deduced and printed in a text window. One example is illustrated in Figure 4-1 and the Appendix contains examples from the test protocols.

EXPLANATION OF THINKING PROCESS

THE USER SPECIFIED TIME FRAME FOR THIS PLAN GOES FROM 1100 TO 1900
THE PLANNED TIME FRAME STARTS AT: 1100
THE CLOSEST SITE WAS OPEN AT THE STARTING TIME
THE SPECIFIED TIME PERIOD WAS USED FOR PLANNING

THE FINAL SCHEDULE IS:
THE SCHEDULED ARRIVAL TIME OF COMMERCE IS 1100
THE SCHEDULED ARRIVAL TIME OF EATONS IS 1145
THE SCHEDULED ARRIVAL TIME OF HARBOUR_SQUARE IS 1345

STAY TIME REPORT:
AT SITE COMMERCE 45 MINUTES
AT SITE EATONS 120 MINUTES
AT SITE HARBOUR_SQUARE 315 MINUTES

SCHEDULING REPORT:
WHEN SHCEDULING SITE COMMERCE AT 1100 THESE SITES WERE OPEN:
HARBOUR _SQUARE EATONS
WHEN SCHEDULING SITE EATONS AT 1145 THESE SITES WERE OPEN
COMMERCE HARBOUR_SQUARE
WHEN SCHEDULING SITE HARBOUR_SQUARE AT 1345 THESE SITES WERE OPEN
COMMERCE EATONS

REPORT ON SEQUENCE OF SITE VISITS:
THIS PLAN SATISFIES YOUR TIME REQUIREMENTS AND IS
ALSO THE BEST GEOGRAPHIC ORDERING OF SITES POSSIBLE

Figure 4-1 Example Plan Assessment Window

Appointments have a significant affect on a plan. Appointment constraints may cause a plan to bounce all around the map. A full list of the selected appointments and timings is provided in the assessment window. This ensures the tourist is aware of the effect appointments have on the plan. This also permits plans to be understood by tourists who were not involved in the original problem specification.

Food sites are the most interesting of free scheduling sites (sites which are not appointments). Scheduling a food site to best suit the temporal and spatial requirements of a problem is very difficult. A list of food sites and the tourist specified meal activity is presented in the text window. This information explains why a meal site was scheduled at a particular time.

Deciding which site to schedule next during planning is affected by the opening and closing times of sites. The closest site cannot be scheduled if it is not open at the time in question. Similarly, sites farther away from the current location must be advanced in the schedule and planned earlier if they would close in the time period that a closer site may use. Scheduling comments describe all these issues by telling the tourist which sites were open when a particular site was scheduled, and which sites were advanced in the scheduling sequence because they would have closed before they were reached.

The final sequence of a plan may not be the ideal geographic order of visits. Time constraints often necessitate the reordering of sites. As a result, the tourist may not understand why a particular visit order was suggested. The geographic assessment function compares the sequence of site visits, determined by the site ordering module of the route planner at the start of planning, with the final site sequence. The original site ordering represents a route which requires little or no retracing of steps to visit all the sites. Any variation on this route will likely require the tourist to travel farther to visit the same sites. This assessment informs the tourist of the impact the temporal constraints of the problem have had on the original route. The result is reported in the plan assessment window.

4.3 Re-Planning

Re-planning may require the construction of a new plan from the current problem state to the original goal. It may also involve the planning of a sequence of operations to return a deviant plan to an intended state, after which the original plan may be followed to the goal. Re-planning can be activated by a plan monitor or by re-specifying the problem conditions.

The focus of the research has been plan formulation, not plan execution. As a result, the tourist planner is not equipped to react to dynamic changes which may be encountered during plan execution. Re-planning has been implemented as a reactive process which is initiated by the tourist through a menu (Figure 4-2). Re-planning actions begin at the task planning level. Once the tourist has altered the task level decisions, the system proceeds to develop a solution in the same manner as for planning.

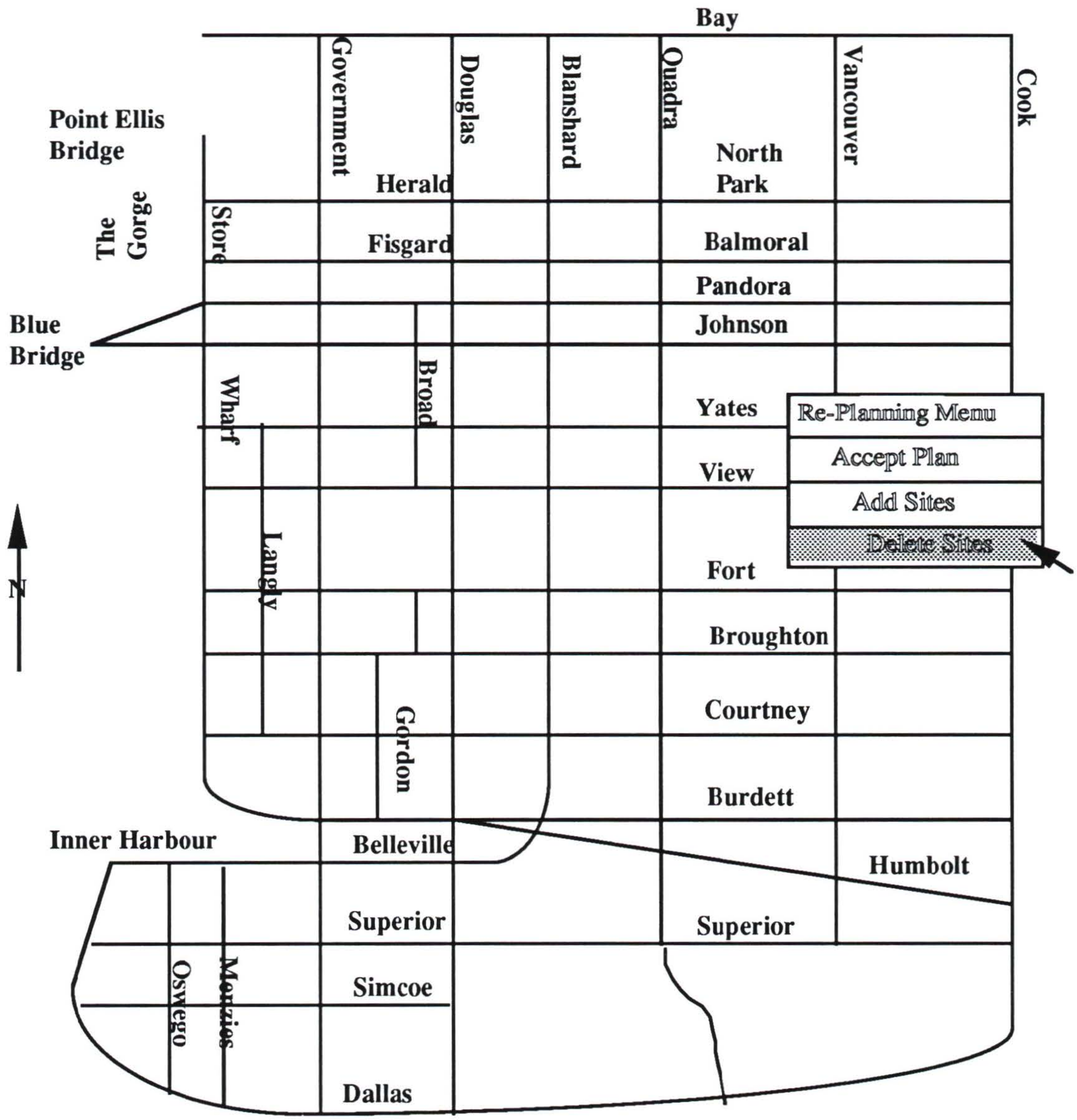

Figure 4-2 Re-Planning Options

The re-planning functions reset the state variables of the planner, adjust the status of the blackboard, and activate the task planner. Information used to reset the blackboard is contained in a plan rationale object located on the blackboard. This object holds data obtained from the tourist and information derived by the system during planning.

Re-planning uses much of the planning information obtained from the tourist during the initial task planning phase. The dynamic planning hierarchy is retained and tourist decisions are stored in the plan rationale. This approach could also be used to implement dynamic re-planning since much of the original plan is retained and could be modified by a re-planning expert.

4.4 User Interface

One of the principle objectives of this research was to construct a prototype tourist information planner. The issue of an effective interface to such a system is itself a substantial research problem. The prototype interface is a pragmatic and efficient solution to this difficult problem.

KEE provides a rich object-oriented programming environment along with support for proprietary windowing techniques and Sun Common Windows™. Common Windows has begun to emerge as a de facto LISP windowing standard. As a result, most of the user interface is built to adhere to this standard in an effort to improve future code portability.

The primary method of tourist interaction is a mouse activated menuing system. A map display is used to assist tourists with decisions which depend on local information. KEE's object-oriented features allow graphic characteristics to be assigned to each map object. The entire map and all sub-classes are easily displayed on the screen by sending a message to the root of the map object hierarchy. The same code which draws the map is used to highlight specific features. Figures 4-3 to 4-9 illustrate a typical tourist interaction session.

SELECT ONE OF THE FOLLOWING:
(A) TRAVELLING BY CAR
(B) TRAVELLING ON FOOT
A

Figure 4-3 Mode of Travel Selection

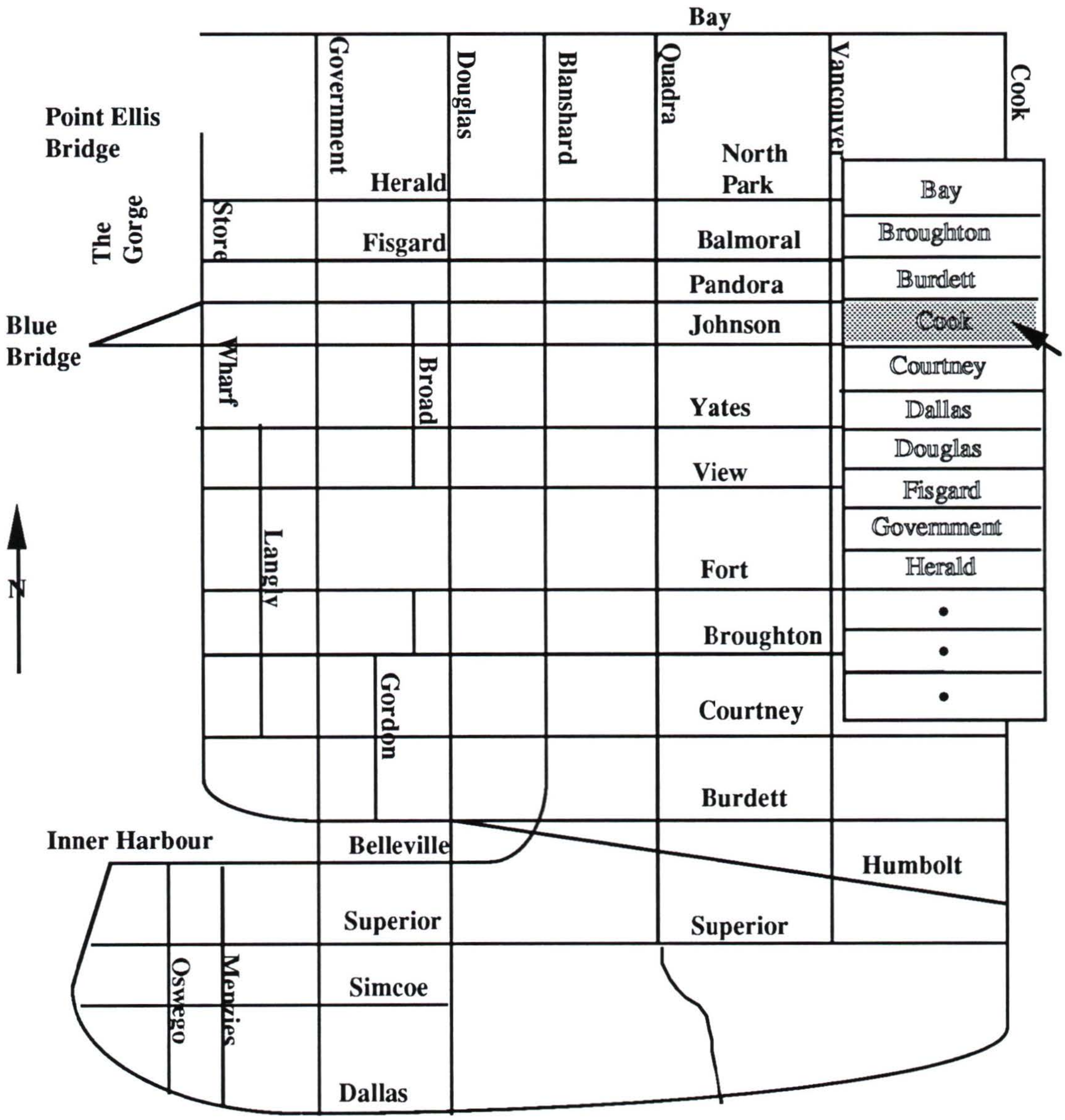

Figure 4-4 Starting Intersection Selection Screen

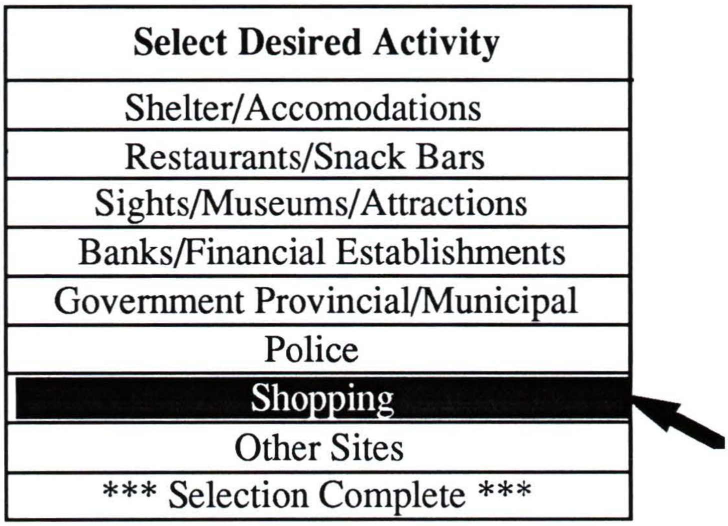

Figure 4-5 Task Level Selection Menu

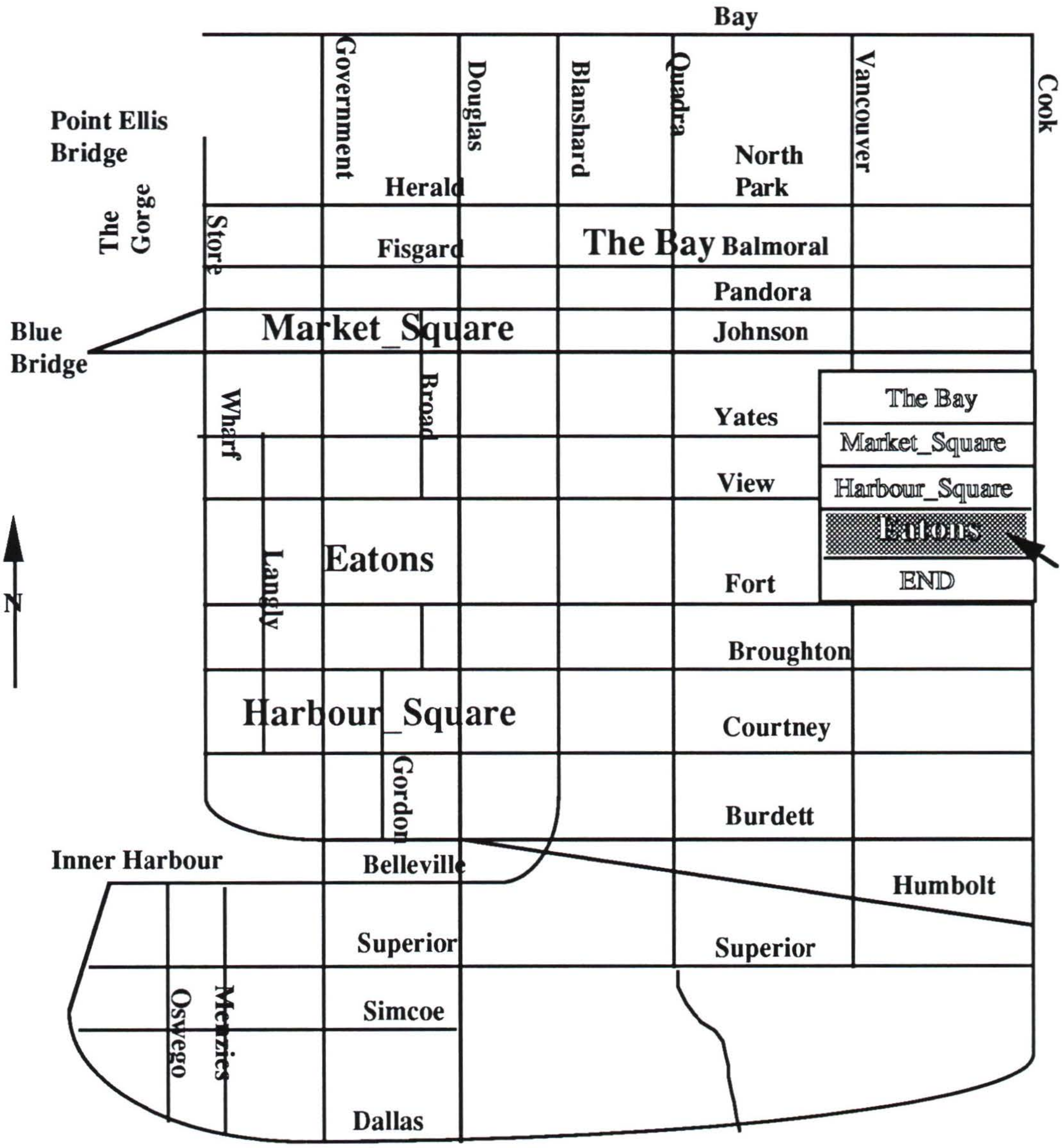

Figure 4-6 Site Selection Screen

```
I AM CONSIDERING SITE EATONS
DO YOU HAVE AN APPOINTMENT AT THIS SITE? (Y/N)
Y
PLEASE ENTER TIME OF APPOINTMENT
CAUTION: THE SYSTEM USES THE 24 HOUR CLOCK
1330
WHAT IS THE AMOUNT OF TIME OF THIS APPOINTMENT (IN MINUTES)?
60
```

Figure 4-7 Appointment Interaction

```
PLEASE ENTER STARTING TIME:
1100
AND ENDING TIME
1900
THANK YOU, WORKING....
```

Figure 4-8 Time Inputs

Theoretically there should be no limit on the number of sites to be visited, but in practice there may not be sufficient time to visit more than a reasonable number of sites. The limit depends on the amount of planning time allotted by the tourist and the class of sites to be visited. For example, it takes more time to visit a museum than a bank. A small rule set can be used to implement a set of heuristics to advise a tourist when the number of sites selected may be too large to reasonably expect to accomplish all the tasks. This feature has not been implemented in the prototype system.

The output from the planner consists of a window containing the map with a highlighted route to follow. This includes site order, paths and suggested visit times. The accompanying text window contains a verbal description of the plan and a summary of the decisions made by the tourist while interacting with the system. Another window displays the plan assessment. A fourth hidden window displays the decisions made during conceptual planning. This window is used primarily for debugging and explanation purposes. All information is available in hard copy as in Figure 4-9. The test protocol results are in the Appendix.

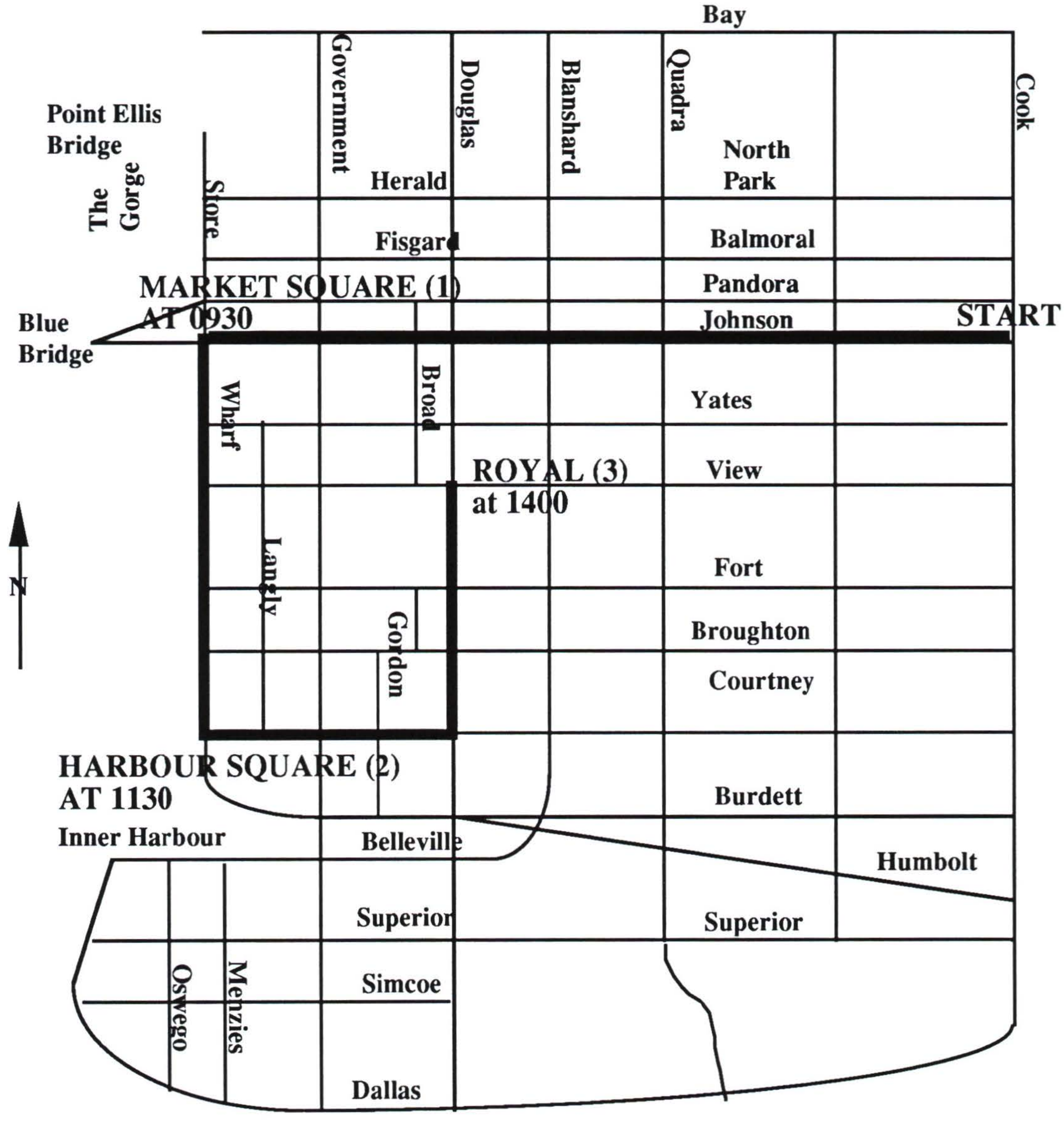

Figure 4-9 Example Plan Output of Route to Follow (Graphic)

```
A RECOMMENDED PATH TO FOLLOW IS

TO GET TO MARKET_SQUARE FOLLOW THIS PATH
GO FROM (JOHNSON COOK) TO (JOHNSON WHARF)
CONTINUE TO (JOHNSON WHARF)

TO GET TO HARBOUR_SQURE FOLLOW THIS PATH
GO FROM (JOHNSON WHARF) TO (COURTNEY WHARF)
CONTINUE TO (COURTNEY WHARF)

TO GET TO ROYAL FOLLOW THIS PATH
GO FROM (COURTNEY WHARF) TO (COURTNEY DOUGLAS)
TURN LEFT
GO FROM (COURTNEY DOUGLAS) TO (VIEW DOUGLAS)
CONTINUE TO (VIEW DOUGLAS)

YOU ARE TRAVELLING ON FOOT FOR THIS PLAN
```

Figure 4-10 Example Plan Output of Text Describing Route

The text in Figure 4-10 can be augmented with detailed descriptions of the final path from the destination intersection to the actual visit site. This information would be available from the object in the knowledge base. For example, a site like the Empress Hotel is reached from the intersection (Belleville Government) by following this tourist oriented description:

"Now that you have reached the intersection of Belleville and Government the Empress Hotel may be found by travelling up Government in the direction of the centre of the city, keeping the Inner Harbour on your left, until you reach the main entrance driveway. Then, turn right and proceed up to the front door of the hotel."

4.5 Test Protocols

The tourist information planner is a modular software system. Each module has been individually tested during development to ensure it functions as specified. Complete test protocols are required to ensure these modules function correctly in an integrated package. Test protocols must be based on the key features of a system. For the tourist planning system, protocols are designed to highlight:

(1) spatial ordering and scheduler sequence retention

(2) global time constraint relaxation

(3) priority site scheduling (police sites)

(4) appointment scheduling

(5) food site scheduling

(6) 'on-the-fly' constraint relaxation.

There are three suites of testing protocols in total. The first suite includes protocols which verify the system features (Protocols 1 to 3). The second suite consists of protocols which examine several of these characteristics as they interact during complex planning problems (Protocols 4 to 6). The final suite represents random problems generated by testers unfamiliar with the system (Protocol 7).

Each protocol includes several tests. These are described by input characteristics which would be determined interactively with a tourist. No attempt has been made to predict the outcome of a test since this would be biased in favour of individual preferences. Timing data are included to provide an estimate of the capabilities of the prototype and to identify potential areas for improvement.

The tests in Protocol 7 are divided into three categories: easy, medium and hard. An easy test is one which contains very few conflicting requirements. Some constraints may conflict in a medium test, but the tester does not consider the conflicts excessive. A hard test has many difficult requirements to satisfy. The plans which result from the tests in Protocol 7 are graded unacceptable, acceptable or excellent by the tester. An unacceptable plan is one which does not satisfy the problem constraints or suggests a route or schedule the tester does not consider adequate enough to form a basis for plan execution. The tester grades a plan acceptable if it can be used to satisfy the specified tasks. An excellent plan is one the tester feels suggests a plan which describes a good route between sites and a schedule which easily satisfies all constraints. This is a very subjective measure of performance. All tests are listed in the Appendix.

4.5.1 Protocol 1

Spatial ordering and scheduler sequence retention are desirable characteristics of a geographic planner. The concept planner must build efficient plans which minimize path length while accommodating temporal constraints. If no time constraints conflict the planner is required to preserve the order of site visits established by the geographic site ordering module at the commencement of planning. This geographic sequence represents a reasonable route through all of the sites to visit. The quality of the sequence is limited by the capabilities of the site sequencing module in the route planner. This module depends on the restricted list representation of the city map to sequence the sites.

This protocol also tests the ability of the route planner to locate a good path between two sites. The planner uses the turn costing function to select the most suitable of the possible paths between two sites. The selection of foot or car travel causes the route finding module to consider one way streets during path processing. Changing the mode of travel may occasionally result in a different selected order of sites, depending on the influence of one way streets in the local map.

All the tests in this protocol result in paths which are linked in a route which is a good circuit through the visit sites. No temporal constraints were imposed which would have required the rearrangement of the sites. The planner did not alter the sequence of visits determined by the shortest site sequencing module of the route planner. In addition, all paths between two sites followed the route which had the fewest turns.

4.5.2 Protocol 2

Global time constraints play an important role in scheduling. One global constraint is the recommended duration of stay of each visit site. This must be reduced by the planner prior to scheduling if there is insufficient time to accommodate all planned visits. The stays are reduced in proportion to the amount of time they would require if sufficient time was available. Appointment sites are not reduced by any length of time because their stay is defined by the tourist in the task planning stage.

Another global constraint relaxation feature verifies tourist input conditions. If the tourist specifies a time frame in which one or more of the intended visit sites is not open, he or she will be requested to alter the visit time frame or drop the offending site. This also applies to appointment tasks, and appointments which are scheduled concurrently. A time constraint relaxation mechanism is used in situations where the planner identifies the possibility of improving a plan by delaying the proposed start time. This is an important capability since small delays can result in significantly improved plans, both from a temporal and spatial perspective.

These test results illustrate the application of the time constraint relaxation mechanisms. The duration of stays of non-appointment sites are reduced proportionally to fit the available planning period when the time required exceeds the time permitted. The planner uses the relative length of durations of stay to calculate these reduced stay times. In addition, the planner implements the delayed departure strategy when this action would not reduce the duration of stay of the sites in the plan, and the closest site to the start is not open at the starting time. The input test functions successfully prevent the tourist from specifying visits which are impossible due to time constraints.

The current planner does not explicitly deal with travelling time when calculating stay times. All the sites in this problem are sufficiently close together to permit the assumption that travel time can be distributed equally among the stay times of all sites. This assumption would not hold in a larger area network. In such a case the route planner would be required to estimate travel times based on the number of intersections traversed and the method of travel. This time would then be factored into the scheduling process. The route planner could formulate an accurate travel time to each site if given additional GIS information, such as traffic densities at different times of day and accurate travel distances.

4.5.3 Protocol 3

The tourist planning system is designed to deal with police sites as high priority visits. These sites are special since tourists specifying the need for a police visit may require urgent assistance. Additionally, the mechanism which is used to ensure priority scheduling of police sites may be used in future versions of the system to facilitate additional prioritization of tasks such as hospitals, medical offices, clinics, etc. The test results in this protocol illustrate the effect of a tourist directing the planner to visit the police site as an urgent priority task or as a routine visit.

4.5.4 Protocol 4

One of the essential features of the scheduler is its ability to schedule appointments. Appointments are tourist specified visits to sites at specific times for a fixed period of time. The ability integration of appointments in an efficient spatial and temporal plan is critical because appointments represent a common occurrence in every day planning. In cases where all sites to be visited are appointments, the planner links together the visits with the best route between each site for a tourist to follow. Appointment times may result in longer paths and inefficient sequences of visits, however time is the most important constraint and these plans must be accepted.

The test results illustrate the effect appointments have on the geographic visit sequence. If an appointment is not the closest site, but it is the next site which must be scheduled due to time constraints, the nearest site to the current location will not be visited next. This breaks the best geographic sequence of visits and results in longer routes to follow. The appointment report in the plan assessment window displays the specified appointments to enable the tourist to understand why a longer route may have been suggested.

4.5.5 Protocol 5

Food sites are one of the most interesting scheduling problems encountered in a geographic planner. It is very difficult to integrate a food site in a daily plan with a minimum of spatial plan disruption, while still satisfying global time constraints, appointment constraints and visit requirements. If the food scheduling specialist fails then no plan involving food sites will be useful.

Test results illustrate the capability of the tourist information planner to integrate visits to food sites in a daily plan. The food scheduling expert system determines the most appropriate time for the meal sub-task to be accomplished. For example, if the lunch meal can be scheduled at any time between 1100 and 1400, and there is an appointment at 1130 for 60 minutes, the expert system will respond by suggesting lunch at 1230. The food scheduling expert forward chains over the rules to locate a planning window within the key time frames for the meal in question.

The key time frames for all meal activities are adjustable parameters in the expert system. For North American tourists, meals are established as follows:

(1) Breakfast: 0600-1000

(2) Lunch: 1100-1400

(3) Dinner: 1700-2000

These parameters guide the expert to an appropriate window in which to schedule a meal. The range of meal hours may be adjusted for tourists from other regions of the world.

4.5.6 Protocol 6

Relaxing constraints on-the-fly is what humans do when they encounter problems near the completion of planning. People do not build elaborate decision trees to backtrack and re-plan. They address a difficulty late in the planning process by relaxing local constraints. This may include fitting tasks into the best time window, reducing site visit durations, reducing site stays to an absolute minimum value, extending the scheduling period, or eliminating low priority site visits.

The on-the-fly specialist in the concept planner local blackboard system conducts this planning. The tests in this protocol show the effect this process has on a plan. The planned geographic sequence may not be the best possible route because the on-the-fly planner is only concerned with accommodating temporal constraints. Problems which require the actions of this specialist are tightly constrained by temporal parameters. The strategy adopted is printed at the bottom of the plan assessment text window.

4.5.7 Protocol 7

The tests in protocol 7 were randomly generated by independent system evaluators. The resulting plans were given to the individual for scoring. No plan was evaluated worse than acceptable, and a small portion of the easy and medium problem solutions were graded excellent. This is primarily due to the flexibility of both the planning system and the human who executes the plan. The subjective assessment of the usefulness of a plan was favourable in all planning tests.

The system tests in the protocols were all reviewed by the travelling salesman on whom the conceptual planning and route finding experts were based. He concluded they all form a valid basis from which to begin plan execution.

4.5.8 Test Results

Software testing seeks two objectives: verification and validation [Fox 90]. Verification concerns whether the specification of the system has been correctly implemented. Validation examines the system performance and determines if it satisfies the requirements of the design.

The successful tests in Protocols 1 through 7 verify the key features of the tourist information planning system. The top level blackboard is controlled in a restricted opportunism with methods, active values and function calls. The performance of this blackboard is verified by interactive statements to the tourist at the task level, and by trace statements output to the screen during concept and route planning. The tourist information system is valid for the tests presented because the results have been subjectively judged acceptable by the local travelling salesman. A more comprehensive suite of tests would be required to validate a production version of this system.

Test timings were gathered for assessment from all protocols. Although the current system is not a production tourist planner, these figures can suggest the capabilities of the model and areas which could be improved. Tables 4-1 and 4-2 contain the results. All times are given in minutes and seconds. SD refers to the sample standard deviation. The times were taken from the end of tourist interaction until the entire plan was displayed on the screen. This includes conceptual planning, route planning, plan assessment and graphical presentation of the result. It does not include task planning because task planning is an interactive process.

Protocol	Average Time to Plan	SD of Planning Time
1	1:22	0:15
2	2:43	1:06
3	2:26	2:24
4	3:08	1:34
5	1:30	0:44
6	2:32	0:48
7	2:16	0:54
All tests	2:01	1:09

Table 4-1 Overall Performance Figures

Number of Sites	Average Time to Plan	SD of Planning Time
2	0:38	0:07
3	0:56	0:21
4	1:50	0:41
5 or more	3:10	0:51

Table 4-2 Performance Related to Number of Sites to Plan

These performance figures do not include the time taken by KEE for stop-and-copy garbage collection. Garbage collection usually occurred at least once during planning tasks of more than five sites. Garbage collection would also increase in frequency the longer the system ran as the available free memory decreased. The inefficiency of the collection process was responsible for this reduction in free memory. The system was reset once garbage collections exceeded two passes per plan to prevent free memory deficiencies from reducing performance.

The average planning time for all tests in Table 4-1 is reasonable for a research system. The large standard deviation for some protocols is a result of grouping problems of many visits with those visiting only a few sites. For example, Protocol 3 contains tests with two sites and tests with five sites. The most time consuming portions of the prototype system are site sequencing and route planning. The addition of more complete geographic information system would permit the sequencing process to be improved and reduce planning times.

The qualitative and quantitative measurements of system effectiveness, though determined from a small sample, suggest the current prototype is a successful implementation of a tourist information planning system. The next section reviews some items of discussion which were encountered during the construction of the prototype.

4.6 Discussion

There were several valuable design and implementation lessons learned during this research. Most stem from the AI approach guiding this work, an approach which attempts to build AI systems which model human cognitive processes and human knowledge organizations. These lessons are discussed in the following sections.

4.6.1 Spatial Knowledge Representation

A human geographic planner is faced with the difficult task of storing a large amount of spatial data. Cognitive mapping research [Kapl 73, Lee 74, Pete 73, Davi 86, Kuip 78] suggested that humans represent spatial information in a street-based format with map objects related to the streets on which they are located. Much of the map information is condensed and reduced to prominent points, landmarks and stylized networks of streets. This simplified cognitive map led to a human-like spatial knowledge representation. The approach is to represent streets as lists, not as nodes and links. All streets in the downtown core of Victoria are represented in the tourist planning system in this list manner.

4.6.2 Path Planning

The simplified spatial data representation creates difficult path planning problems. There is no accurate distance information, no reliable cardinal directions, and no means to implement conventional AI searching algorithms [Nils 68, Tani 86, Wils 84]. As a result, a novel route finding approach was developed. This has been modelled after a technique described by a local travelling salesman [Lero 89b]. It consists of a human-like search for connecting paths between the source and goal intersections. The planner locates common intersections along the route and constructs a path through these points from the source to the goal.

The human-like path finder must be designed to locate a number of paths between sites. This information is required by other levels of the system to support high level decision making. For example, the scheduler may wish to know if an alternative path would permit the tourist to accomplish a task on the way to another. This requires more than one route between sites.

A controlling heuristic is required to determine when sufficient paths had been found for the other planners. This is accomplished by limiting the route finder to a maximum of seven routes. This limit was chosen because it would ensure the node expansion process outlined in Section 3.5 would explore paths leading in all directions from the source and goal. This is analogous to humans checking in all directions before heading off.

The path counting problem highlights one of the dilemmas which face AI practitioners attempting to model human cognitive processes. The speed of computer computation is very difficult to resist. The path planner could have been permitted to search for many more routes to improve path planning. This would have compromised model realism for computational performance. AI practitioners who do not attempt to model human cognition and representations of information can capitalize on the increasing computational performance modern computers offer. However, this approach may not increase their understanding of human cognition.

4.6.3 Costing Paths

The paths generated by the route planner consist of a list of intersections to follow from one site to another. These lists contain no costing information. This generates a requirement for heuristic costing estimates based on common sense. One method of costing a path is to count the number of distinct intersections traversed from source to goal. A path crossing fewer intersections is deemed to cost less than one with more. This ignores the issue of distance between intersections but solves the problem. In human cognitive maps a similar problem arises because only a rough estimate of true distance is possible. In addition, a key cost factor in a city path is the number of intersections crossed, since traffic control measures slow or impede progress at these points.

Another costing function counts the number of turns in a path. This is a tourist oriented measure since a path with fewer turns is much easier to follow in a new city. This feature is used to recommend the most simple path to follow.

The system is capable of employing either of these costing functions to select the best route between two sites. The intersection crossing costing function is used when sorting sites based on their cost from an original location. Once the concept planner has satisfied all spatial and temporal constraints, the route planner picks the easiest route to follow for the final plan using the turn costing function.

4.6.4 System Design

Large and complex software systems must be designed to be modular, extensible and maintainable. The key feature which assisted the development of the tourist planning system was the modularity of the blackboard model. Knowledge sources were added and deleted with little difficulty. In addition, the system was rapidly prototyped by constructing shells of the specialists. Later research led to the details of each knowledge source.

The system is enhanced by object-oriented programming techniques. These are used to control the limited opportunism of the tourist planner. A hierarchical data organization is implemented to organize and speed retrieval of spatial information. A further enhancement made possible by object-oriented programming is the construction of the user interface. Graphic attributes are assigned to each object and programmatically controlled to display the map. A final plan is drawn by sending messages to the components in the planning hierarchy.

4.6.5 Programming Common Sense

There are many features of the tourist planning system which can be considered common sense. During development it was found that much of what one person considers common sense may not appear all that common to someone else. The issue of common sense is an interesting research field of its own.

One common sense issue concerns police sites. In general, a tourist who specifies a desire to visit the police station probably requires some sort of assistance, perhaps in locating a lost family member or to report the theft of a wallet. Based on this consideration, the system was first programmed to consider it imperative that if a police site was selected it would be the first site scheduled. Other testers suggested that some people may wish to make police sites routine visits. To accommodate this the system was modified to permit the tourist to override the primary planning rule that police sites be visited first.

Scheduling food site visits also offers an opportunity to incorporate more common sense. An expert must decide when it is best to schedule a food site within an overall plan. For example, if a tourist selects a breakfast activity at a food site, but asks for planning to begin at 10 AM, it makes sense to schedule the food site early in the plan, because the usual breakfast time period is ending. Similarly, if a lunch site is selected and the plan is to end around 11 AM, then the meal activity is best placed at the end of the time period. The food expert embodies rules to select the most appropriate time window for dining.

Food sites can also be scheduled in another common sense manner. If the tourist were to select the class of food and the meal desired, a food scheduling specialist could find the closest restaurant meeting these requirements at the most suitable planning opportunity. The current expert responsible for finding a food window would still be needed to advise the food scheduler. The food scheduler would also require the assistance of the task planner to find a site suitable to the task. The route planner may be required to determine which site was closer should more than one location be acceptable. This alternative food scheduling strategy could be implemented by adding another food scheduling expert capable of accessing the local concept planning blackboard.

Common sense is required in the global constraint relaxation process at the onset of scheduling. Some method of reducing these stays must be implemented if insufficient time exists within the planning period to schedule all sites for their usual duration of stay. Various reduction schemes were investigated, such as reducing only the longest stay by enough time to permit all visits, reducing all stays proportionally, and reducing stays by priority.

It has been assumed that tourists intend to visit all the sites they select during task planning. This suggests the best strategy is an equal reduction of stay times to ensure all sites fit the specified time frame. However, an additional problem arises when appointments are specified. These are high priority visits which the tourist has specified and they may not be reduced in the constraint relaxation process.

4.7 Chapter Summary

The performance of the tourist information planner has been deemed acceptable as an integrated software package by the developer, the expert on whom the concept and route planners were based, and independent evaluators. The simple user interface did not prevent untrained users from quickly understanding inputs and outputs. Their queries ranged in complexity from simple planning tasks to difficult schedules integrating appointments, food sites and many visit locations.

The quantitative performance of the system summarized in Tables 4-1 and 4-2 is reasonable for a prototype developed in LISP. The planner generated plans in a short period of time which were readily accepted by evaluators as good starting points for a day's activities. The overall satisfaction of a tourist executing one of these plans cannot be determined because no plan monitoring features are included.

Chapter 5 Conclusions and Direction of Future Research

5.1 Summary of Project Objectives

The objectives of this research were to explore the mechanisms involved in geographic problem solving, the representations necessary to support such reasoning, and to build and test a model of human geographic planning using AI programming techniques. The implementation of the model is intended to improve our understanding of planning. A seminal theme of this work was to model the internal representations, processes and outputs of planning on human techniques.

A tourist information planning system was conceived as a practical example of geographic planning. The system was designed to assist travel counsellors in a human-like fashion. A prototype was constructed to demonstrate the useful planning capabilities of this approach. A generalization of the model used to solve the tourist problem can be extracted for use in different geographic problem areas.

5.2 General Model Specification

The specialists which collectively conduct human-like geographic planning in a blackboard architecture have been constructed for the tourist planning domain. This section will abstract the details from these planners to define a general model suitable for a broader range of geographic planning problems. The abstraction is restricted to geographic domains since the planning model is not designed to address more general problems.

5.2.1 Task Planner

The task planner conducts high level problem decomposition in order to constrain the search for visit sites which may satisfy a given problem. It interprets input requirements and uses these to search a hierarchical database of visit locations. The task planner may be a simple function which equates separate classes of visit sites to particular problems, or it may consist of an expert system which reasons about the input conditions to select the most appropriate site visit classes.

Task level planning establishes the initial visit search constraints. This stage is necessary to limit the specific site searching process because a map represents a large search space with many possible visit locations. The search constraints are used to locate sites which may be visited to solve the stated geographic problem.

The information retrieval task of a geographic problem solving system may be satisfied by task level planning alone. Details of possible sites, numbers of sites, or descriptions of sites are all determined at this level. As a result, data may be quickly and easily extracted from the knowledge base and presented to the user to satisfy a basic information request.

5.2.2 Concept Planner

Conceptual planning begins when the task planner has determined the sites which must be visited to solve the geographic problem. The sites must be sequenced and scheduled within specified temporal constraints. The concept planner applies a human-like approach to this problem which satisfies both time and space constraints while building the conceptual plan. A small, local blackboard is an excellent method of implementing this process.

A general geographic planner requires some means of establishing a priority of sites to be visited. The conceptual planning problem is less complex in the case when all sites are equally important; it merely determines a good spatial sequence which satisfies all time constraints. Prioritized site sequences require the conceptual planner to ensure the highest priority sites are visited, with lower priority sites incorporated in a manner which produces a good solution. The concept planner must balance time and space in this prioritized planning scheme.

The model of conceptual planning may also apply to other geographic problems. These could include rolling stock management in railway planning systems, local delivery schedules for courier services, pick up plans for rural farm products, or urban waste collection schedules.

5.2.3 Route Planner

The route planner is responsible for establishing routes between two sites, and routes which link together a series of sites to visit. The latter feature is used by the concept planner during site sequencing operations to establish visit sequences. The former is used by the concept planner to narrow visit site locations by geographic constraints, such as distance or time, and also by the route planner to establish the exact route between sites.

A general route planner may contain several routing algorithms. Selecting which to use will depend on the amount of information available to the planner in the knowledge base. Altering the router to capitalize on map information in the problem domain is not a violation of the principle of human-like processing and knowledge representation. This approach merely implements a more structured and efficient routing mechanism. This is similar to a human deliberately planning a route with a map, pencil, calculator and ruler.

The various routing mechanisms incorporated in a route planner may be selectively applied to different geographical problems. The human-like router described in Section 3.5 is quite suitable in the case of the tourist information problem. A flexible and well equipped route planner may be used for a wide range of geographic problems. These include water main re-routing and repair planning for city services, high power transmission line routing in disaster planning, transit route planning, interconnection planning of communication networks, and robotic path planning in factory environments.

5.2.4 Knowledge Representation

Knowledge representations of most AI systems are heterogeneous collections of rules, facts and processing functions. It is difficult to specify which portions of the knowledge base may be problem independent, even within the restricted domain of geographic planners.

Rules are used to implement some of the features of each knowledge source. Those which determine the processing sequence and control of a specialist are domain independent. Rules which make specific domain decisions cannot be applied to other problems. Some domain specific rule sets will be required for each new problem. The modular nature of the blackboard permits rapid construction and implementation of these replacement experts.

Facts are domain specific. Their organization is standardized as an object-oriented hierarchy in this model. The hierarchy is designed to permit rapid selection of sites according to the task level requirements of a tourist information system. The hierarchy must be reorganized for a different geographic problem. This can be accomplished by examining the problem specific task queries. A conversion module may be used to parse these facts into the requisite object-oriented hierarchy along user defined organizational lines [Rous 89] if a conventional Geographic Information System is used as a database.

The most domain specific component of any geographic planner is the knowledge base. The general model of a geographic planning knowledge base is an object-oriented hierarchy organized to satisfy problem specific queries. Object-oriented programming features such as multiple inheritance and message passing may be used to enable more varied queries to be satisfied by one knowledge base.

Processing functions are used to conduct much of the routine work of a geographic planner. Many are utilities which can be effective for a range of problems. Others are domain specific functions which will need to be revised for each new problem. In some cases the functions may not be required at all.

5.3 Conclusions

A prototype human-like geographic planning system has been successfully implemented. The construction of this planner has led to a more thorough understanding of human planning. In addition, this research has explored the methods used by people to represent and reason with geographic information.

Human planning is hierarchical and opportunistic. People plan by selecting the best level of decision making at the appropriate time based on the status of the problem state. A human-like geographic planner can be used to solve tourist information geographic problems in a three level blackboard problem solving architecture.

Blackboard specialists can be designed to implement a wide variety of conventional AI planning techniques. The mechanisms employed by the specialists can be extracted from human-like portions of planners developed by previous researchers. The restricted opportunism of the tourist counsellor can be controlled using object-oriented programming techniques in a blackboard planner. Individual specialists may also employ a blackboard to conduct local planning. An added advantage of the blackboard paradigm is the ability to support modular system development and testing. This allows rapid prototyping of individual components and simplifies the integration of modified specialists as enhancements are made.

The 'weak AI' systems design approach requires a developer to pattern representation and processing after human techniques. This restriction does not prevent a geographical problem solver from successfully planning effective solutions. In addition, the design of knowledge representations and planning specialists is simplified by modelling them on human techniques.

The algorithm developed in this thesis to locate a route between two sites is based on human path planning. It is capable of locating routes in a network representation based on a human cognitive map. This algorithm more closely resembles human route planning than conventional AI searching algorithms. In addition, the conventional AI search algorithms encounter difficulties when working with the restricted amount of information present in a cognitive map.

The general model of geographic problem solving used to implement the tourist information system can be extended to other geographical problems. The hierarchically organized opportunistic planner could be tailored to address the different requirements of other types of geographic problems. The object-oriented knowledge base which supports the planner may require reorganization for other problem domains. A conversion module to a conventional GIS could be used to simplify this process.

Common sense elements are required in any system attempting to conduct human-like planning. These will represent the common sense of the system designer and expert assistants. Their common sense may not appear that common to others wishing to use the system, hence the system must be able to respond with a flexible range of capabilities when making common sense decisions.

5.4 Future Research

There are several research area which can be investigated to improve the general model of a geographic planning system. One is the development and testing of alterative knowledge bases for different problem domains. The organization of the knowledge base suggested by this work can be abstracted and written into a specification of a general purpose geographic knowledge base. A conversion module to import data from a GIS would permit additional geographic information to be easily accessed. Guidelines to be used in developing the object-oriented hierarchy for alternative problems should be developed, along with methods to automate the re-structuring process. Real-time knowledge base updates should be explored.

Another area of future research is the development of a general route planner. This planner should incorporate many path finding algorithms along with meta-knowledge to determine which to use for each problem. These algorithms could be implemented in languages more efficient than LISP. The human-like routing algorithm developed in this work should be assessed in more complex networks of streets and its performance compared to conventional search algorithms. The impact of machine learning of routes and the use of pattern recognition to locate 'known' paths should be investigated.

The human-like path planner requires only a simple list network description. The application of this planner to other routing problems which do not require a full geographic information system level of complexity should be explored. The quick and easy manner in which networks may be input and routes planned may make the human-like path planner a useful rapid prototyping tool for other routing problems.

Re-planning could be investigated to develop re-planning policies. A geographic planning system could incorporate a re-planning specialist to determine the most appropriate means of changing a given plan in a new problem state. Re-planning could also be linked to plan execution monitoring in a responsive, real-time system which could assess the impact of changes observed during plan execution and re-plan to solve new problems. The re-planner would require a dynamic and up to date knowledge base. Re-planning may also be researched by simulating plan execution.

The modular nature of a blackboard planner supports distribution of experts in a cooperative problem solving system. A distributed geographic planner may perform better than a uniprocessor system. The problems of computation vs communication overhead, distributed or centralized control, and knowledge base access, distribution and consistency must all be addressed.

The plan assessment module could be enhanced to improve the explanation facilities of a geographic problem solver. This module could trace rule invocations and assemble more comprehensive decision traces. The plan assessment could also be improved by enlarging the rule-based inferencing module currently implemented.

The problem of delivery platform for a practical tourist information planning system must be examined. A less expensive user machine may require some recoding of the current product. The implementation language may change but the model should remain unaltered. This requires an object-oriented delivery environment. In addition, the user interface must be enhanced to make it more friendly and complete. The delivery platform must also address the performance problems encountered during garbage collection, especially if the system employs a stop-and-copy mechanism.

Common sense is difficult to classify and organize. The process of extracting common sense rules and policies from humans, along with an appropriate classification scheme, could be of much use to future human-like planning systems.

Bibliography

[Chio 84] Chiodini, V., "SCORE: An integrated system for dynamic Scheduling and control of high volume manufacturing", IEEE Computer, Vol 17, # 9, September 1984,, pp 272-278

[Crof 85] Croft, F., "Choice Making in Planning Systems", Expert Systems 85, Cambridge University Press, New York, 1985, pp 125-140.

[Curr 85] Currie,K. and Tate, A., "O-Plan: Control in the Open Planning Architecture", Expert Systems 85, Cambridge University Press, New York, 1985, pp 225-240.

[Dade 84] Dade, C.J., An Introduction to Database Systems, Vol II, Addison-Wesley Publishing Company, Don Mills, Ontario, 1984, pp 181-231.

[Davi 86] Davis, E., Representing and Acquiring Geographic Knowledge, Copyright Morgan Kaufmann Publishers, Los Altos, Ca, USA, 1986.

[Elli 86] Elliot, L.B., "Analogical Problem Solving and Expert Systems", IEEE Expert, Summer 1986, pp 17-28.

[Fike 71] Fikes, R.E., and Nilsson N.J., "Strips: A New Approach to the Application of Theorem Proving to Problem Solving", Artificial Intelligence, Vol 2, 1971, pp 189-208.

[Fox 84] Fox, M.S. and Bourne D., "Autonomous Manufacturing: Automating the Job-Shop", IEEE Computer, Vol 17, # 9, September 1984, pp 76-86.

[Fox 90] Fox, M.S., "AI and Expert System Myths, Legends, and Facts", IEEE Expert, February 1990, pp .8-20.

[Geor 87] Georgeff, M.R., "Planning", Annual Review of Computer Science, Vol 2, 1987, Annual Reviews Inc., pp 359-400.

[Geva 84] Gevarter, W.B., AI and Robotics: Five Overviews, Manufacturing Productivity Centre, Chicago, Illinois, 1984.

[Guzo 89] Guzolek, J. and Koch, E., "Real-Time Route Planning in Road Networks", IEEE First Vehicle Navigation and Information Systems Conference, Toronto, September 1989, pp 165-169.

[Haye 79a] Hayes-Roth, B.F. and Hayes-Roth, F., "A Cognitive Model of Planning", Cognitive Science 3, 1979,pp 275-310.

[Haye 79b] Hayes-Roth B.F, Rosenschier F., Cammarata S., "Modelling Plans as an Incremental Opportunistic Process", Proc of 6th IJC on AI, Los Altos, California, William Kaufman, Inc, 1979, pp 375-383

[Haye 80] Hayes-Roth, B.F. and Thorndyke, P.W., Decisionmaking During the Planning Process",Rand Research Note N-1213-ONR, Rand Corporation,Santa Monica, California, October 1980.

[Haye 81] Hayes-Roth et al, Planners' Workbench: A computer aid to re-planning, Rand Research Note P-6688, The Rand Corporation, Santa Monica, California, October, 1981.

[Hobe 89] Hobeika, A.G., "A Route Guidance System under Modified Network Conditions", IEEE First Vehicle Navigation and Information Systems Conference, Toronto, September 1989, pp 170-175.

[Kami 89] Kamijo, S., Okumura K., and Kitamura A., "Digital Road Map Data Base for Vehicle Navigation and Road Information Systems", IEEE First Vehicle Navigation and Information Systems Conference, Toronto, September 1989, pp 319-323.

[Kapl 73] Kaplan, S., "Cognitive Maps, Human Needs and the Designed Environment", Environmental Design Research, Vol 1, WFE Presser, Ed, Copyright Dowden, Hutchinson&Ross, Inc, Stroudsburg, Penn, USA, 1973, pp 275-283.

[Kast 89] Kasturi, K., Fernandez, R., Amlani, M.L., and Feng W., "Map Data Processing in Geographic Information Systems", IEEE Computer, December 1989, pp 10-21.

[Kemp 84] Kempf K.G.,"Manufacturing Planning and Scheduling: Where We Are and Where We Need to Be", Proceedings of the Fifth IEEE Conference on AI Applications, IEEE Computer Society Press, 1989, pp 14-19.

[Kuip 78] Kuipers, B., "Modelling Spatial Knowledge", Cognitive Science, Vol 7, 1978, pp 129-153.

[Lee 74] Lee, S.A., "Cognitive Mapping Research", Responding to Social Change, B. Honikman, Ed, John Wiley and Sons, Stroudsburg, Penn, USA, 1974, pp 172-188.

[Lee 89] Lee, N.S.T., Karimi H.A., and Krakiwsky, E.J., "Road Information Systems: Impact of Geographic Information Systems Technology to Automatic Vehicle Navigation and Guidance", IEEE First Vehicle Navigation and Information Systems Conference, Toronto, September 1989, pp 347-352.

[Lero 89a] Leroux, G.P., "Summary of Tourist Information Booth Survey", UVic Internal Report, December,1989.

[Lero 89b] Leroux, D.L.P., remarks on path planning during casual conversation, Victoria, June, 1989.

[Lero 89c] Leroux, G.P. and Li, K.F., "An AI-Assisted Geographical Problem Solver", Canadian Conference on Electrical and Computer Engineering, Montreal, PQ, September 17-20, 1989, pp 134-137.

[Lero 89d] Leroux, G.P., "A AI-Assisted Geographical Problem Solver", Proceedings of IEEE Section Conference, Victoria, BC, December 2, 1989.

[Lero 90a] Leroux, G.P. and Li, K.F., "An AI-Assisted Geographical Planner", to be presented at the IEE First International Conference on Expert Planning Systems EPS 90, London, UK, June, 1990.

[Lero 90b] Leroux, G. P. and Li, K.F., "Geographical Problem Solving: A Practical AI Implementation Experience", to be presented at ICCI 90, Niagra Falls, Ontario, Canada, 23-26 May 1990.

[Mars 89] Marsh, D.C., "Database Design, Development and Access: Considerations for Automotive Navigation", IEEE First Vehicle Navigation and Information Systems Conference, Toronto, September 1989, pp 337-340.

[Moha 88] Mohan, L. and Kashyap R.L., "An Object-Oriented Knowledge Representation for Spatial Information", IEEE Transactions on Software Engineering, May 1988, Vol 14, No 5, pp 675-681.

[Newe 56] Newell A., and Simon H.A., "The Logic Theory Machine: A Complex Information Processing System", IRE Transactions on Information Theory, Vol IT-2, No. 3, 1956, pp 61-79.

[Newe 63] Newell A., and Simon H.A., "GPS: A Program that SImulates Human Thought", Computers and Thought, E. Feigenbaum and J. Feldman (Eds.), New Yourk: McGraw-Hill Co, 1963

[Nii 86a] Nii, H.P., "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures", Part 1, AI Magazine, Summer 1986, pp 38-53.

[Nii 86b] Nii, H.P., "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures", Part 2, AI Magazine, August 1986, pp 82-106.

[Nils 68] Nilsson, N.J., Hart, P.E., and Raphael, B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", IEEE Trans of Systems Science and Cybernetics, Vol SSC-4, No 2, July 1968, pp 100-107.

[Numa 84] Numao, M. and Morishita, S., "A scheduling environment for steel making processes", IEEE Computer, Vol 17, # 9, September 1984,, pp 279-286

[Pete 73] Peters, R., "Cognitive Maps in Wolves and Men", Environmental Design Research, Vol 2, WFE Presser, Ed, Copyright Dowden, Hutchinson&Ross, Inc, Stroudsburg, Penn, USA, 1973, pp 274-253.

[Rous 89] Rousseau, J.M. and Roy, S., "GeoRoute: An interactive Graphics System for Routing and Scheduling over Street Networks", IEEE First Vehicle Navigation and Information Systems Conference, Toronto, September 1989, pp 161-163.

[Sace 74] Sacerdoti, E.D., "Planning in a Hierarchy of Abstraction Spaces", Artificial Intelligence, Vol 5, No 2, 1974

[Sace 77] Sacerdoti, E.D., A Structure for Plans and Behavior, Elsevier North-Holland, Inc, New York, New York, 1977.

[Sear 90] Searle, J.R., "Is the Brain's Mind a Computer Program?", Scientific American, January, 1990, pp 26-31.

[Shar 86] Sharkey, N.E. and Brown, G.D.A., "Why Artificial Intelligence needs and Emperical Foundation", Artificial Intelligence: Principles and Applications, M. Yazdani ed, Chapman and Hall Ltd, New York, NY, 1986, pp 267-293.

[Stok 89] Stokey, R.J., "AI Factory Scheduling: Multiple Problem Formulations", SIGART Newsletter, October, 1989, pp 27-30.

[Smit 89] Smith, A.B., "Prototyping a Navigation Database of Road Network Attributes (PANDORA)", IEEE First Vehicle Navigation and Information Systems Conference, Toronto, September 1989, pp 331-336.

[Stef 81] Stefik M., "Planning and Meta-Planning (MOLGEN: Part 2)", Artificial Intelligence, Vol 16, 1981, pp 141-170.

[Suss 75] Sussman, G., A Computational Model of Skill Acquisition, New York: American Elsevier, 1975.

[Tani 87] Tanimoto, S.L., The Elements of Artificial Intelligence, Computer Science Press, 1987, pp 139-178.

[Turi 50] Turing, A., "Computing Machinery and Intelligence", Mind, Vol 59, pp 433-460.

[Wils 84] Wilson, P.H., Artificial Intelligence, Second Edition, Addison-Wesley Publishing Company, Don Mills, Ontario, 1984, pp 87-131.

Appendix

This appendix contains the test protocols used to validate the tourist planning system. This includes test descriptions and hard copy screen outputs from the planner.

A-1 Testing Protocols

A-1-1 Protocol 1

Test 1

Time Span: 1100-1900

Sites: Harbour Square, Eatons, Commerce Bank

Mode of Travel: foot

Starting Intersection: (Bay Cook)

Appointments: Nil

Processing Time: 0:42

Test 2

Time Span: 1100-1900

Sites: Harbour Square, Eatons, Commerce Bank

Mode of Travel: car

Starting Intersection: (Bay Cook)

Appointments: Nil

Processing Time: 0:48

Test 3

Time Span: 1000-1800

Sites: Empress Hotel, Wax Museum, Maritime Museum

Mode of Travel: foot

Starting Intersection: (Dallas Government)

Appointments: Nil

Processing Time: 0:49

Test 4

Time Span: 1000-1800

Sites: Empress Hotel, Wax Museum, Maritime Museum

Mode of Travel: car

Starting Intersection: (Dallas Government)

Appointments: Nil

Processing Time: 0:36

Test 5

Time Span: 1200-2200

Sites:Commerce Bank,Harbour Square,City Hall, Provincial legislature

Mode of Travel: foot

Starting Intersection: (Johnson Wharf)

Appointments: Nil

Processing Time: 0:52

Test 6

Time Span: 1200-2200

Sites:Commerce Bank,Harbour Square,City Hall, Provincial legislature

Mode of Travel: car

Starting Intersection: (Johnson Wharf)

Appointments: Nil

Processing Time: 1:00

Test 7

Time Span: 1100-1900

Sites: Royal Bank, Provincial legislature, Laurel Point Inn

Mode of Travel: foot

Starting Intersection: (Superior Cook)

Appointments: Nil

Processing Time: 0:47

Test 8

Time Span: 1330-1900

Sites: Wax Museum, Maritime Museum

Mode of Travel: car

Starting Intersection: (Bay Government)

Appointments: Nil

Processing Time: 0:36

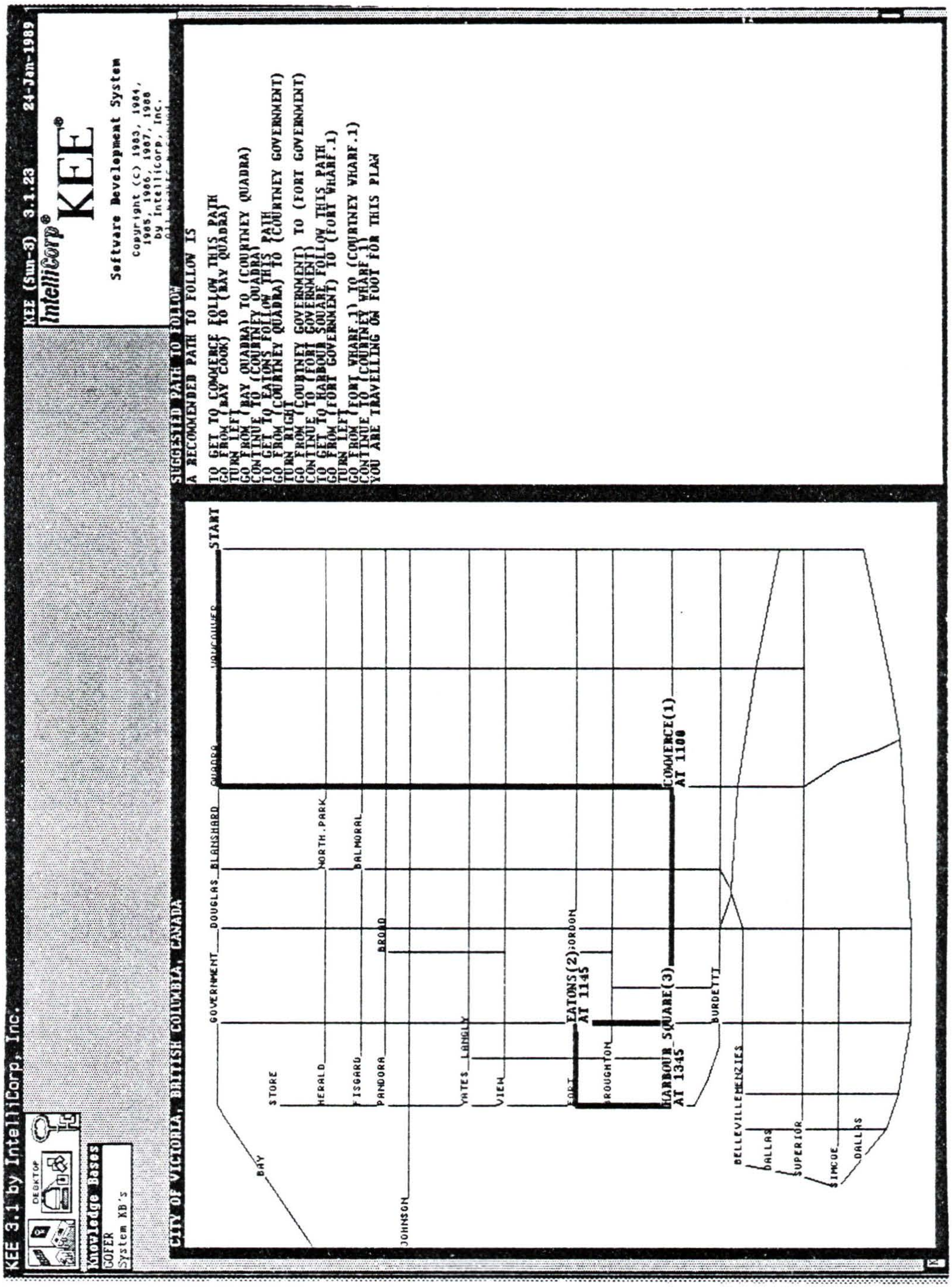

Figure A-1 Protocol 1 Test 1

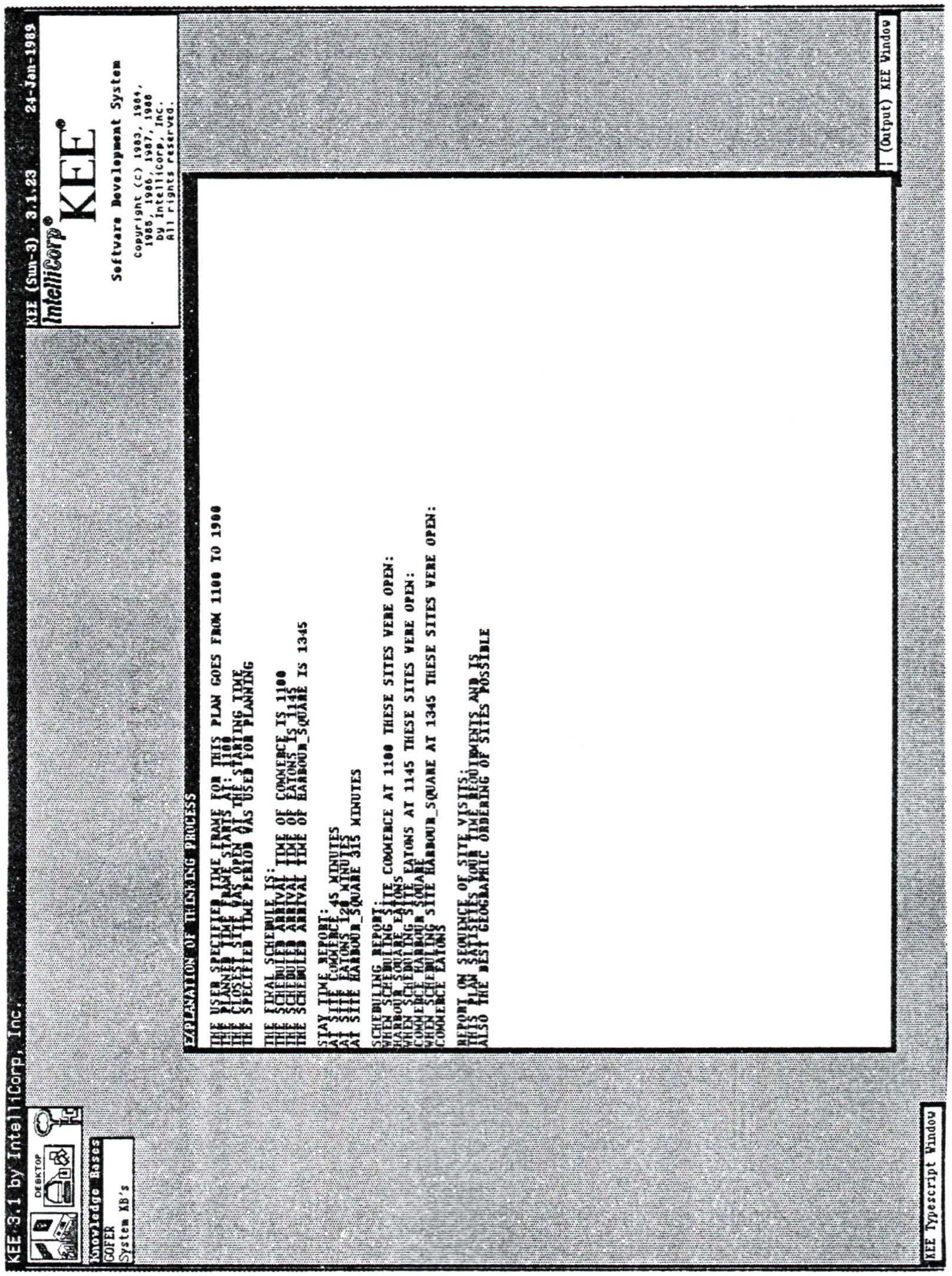

Figure A-2 Protocol 1 Test 1

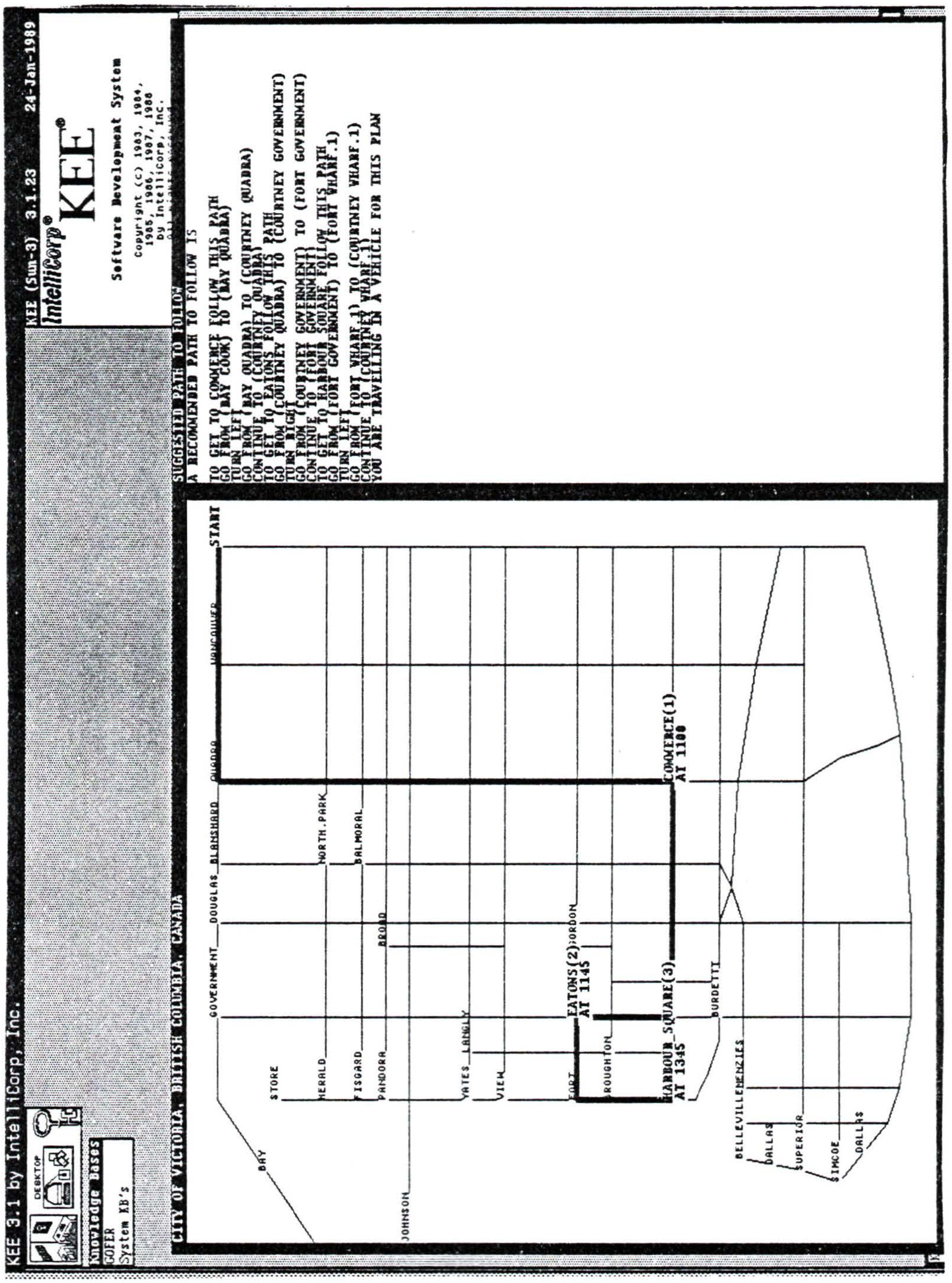

Figure A-3 Protocol 1 Test 2

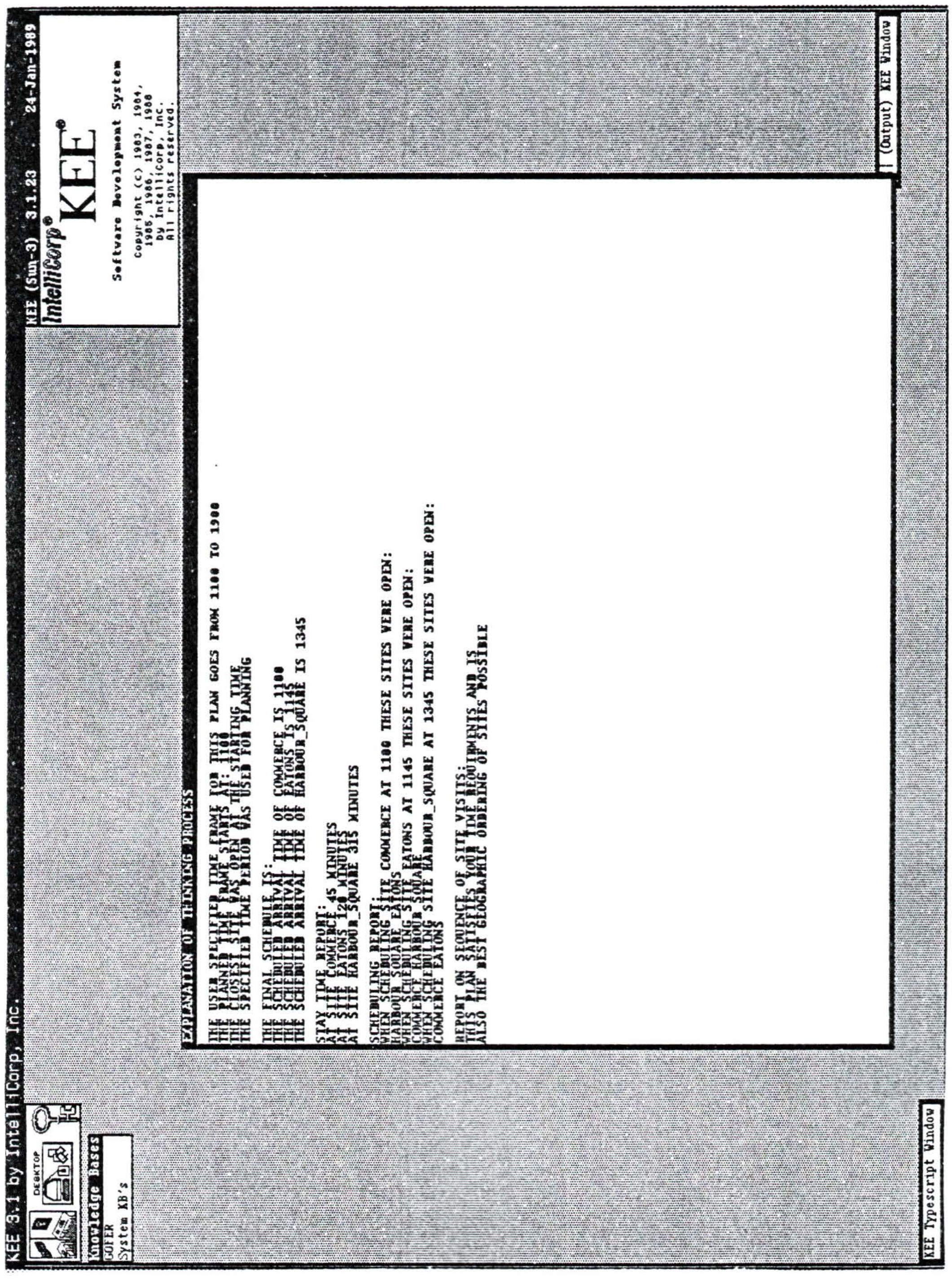

Figure A-4 Protocol 1 Test 2

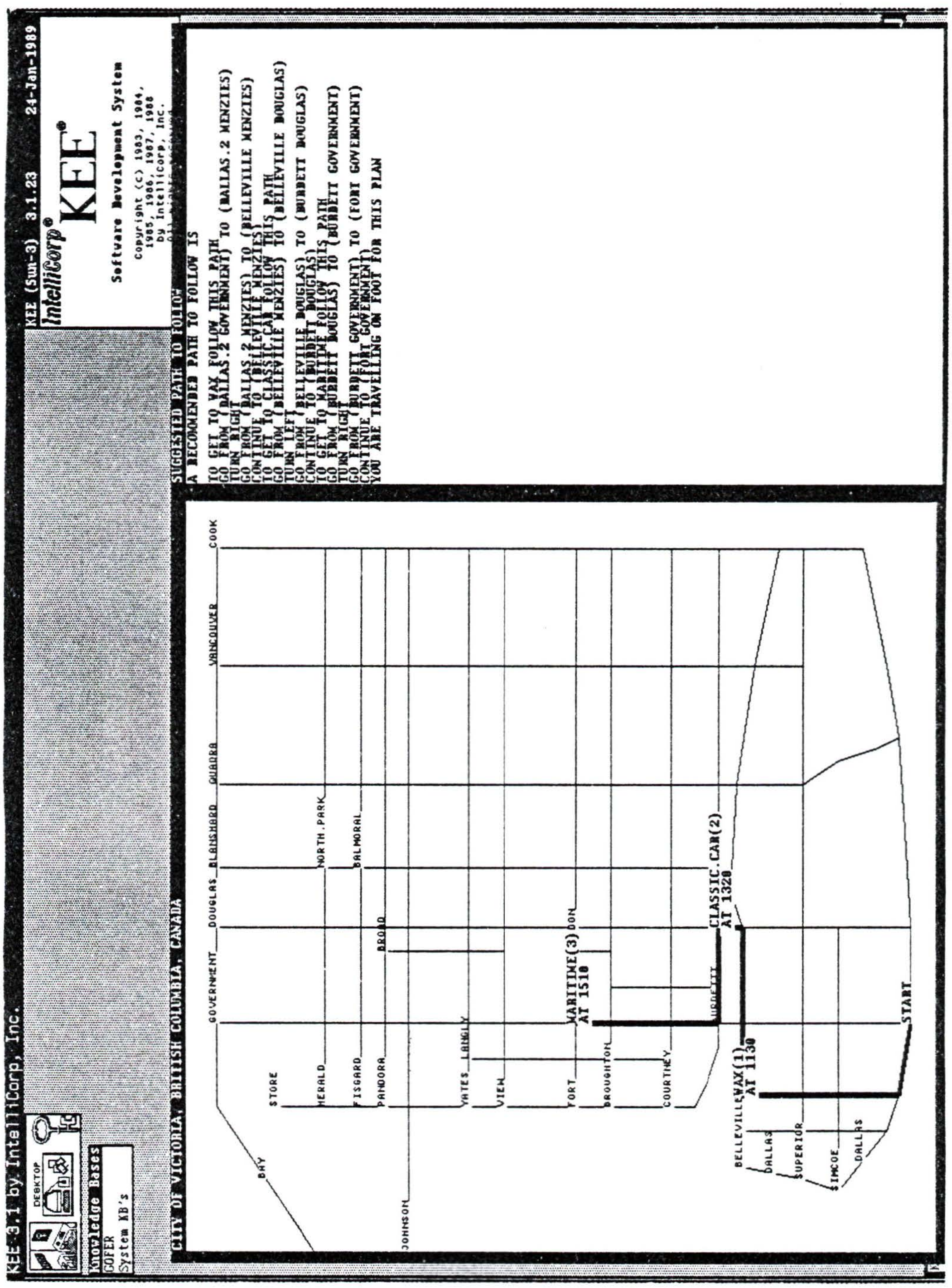

Figure A-5 Protocol 1 Test 3

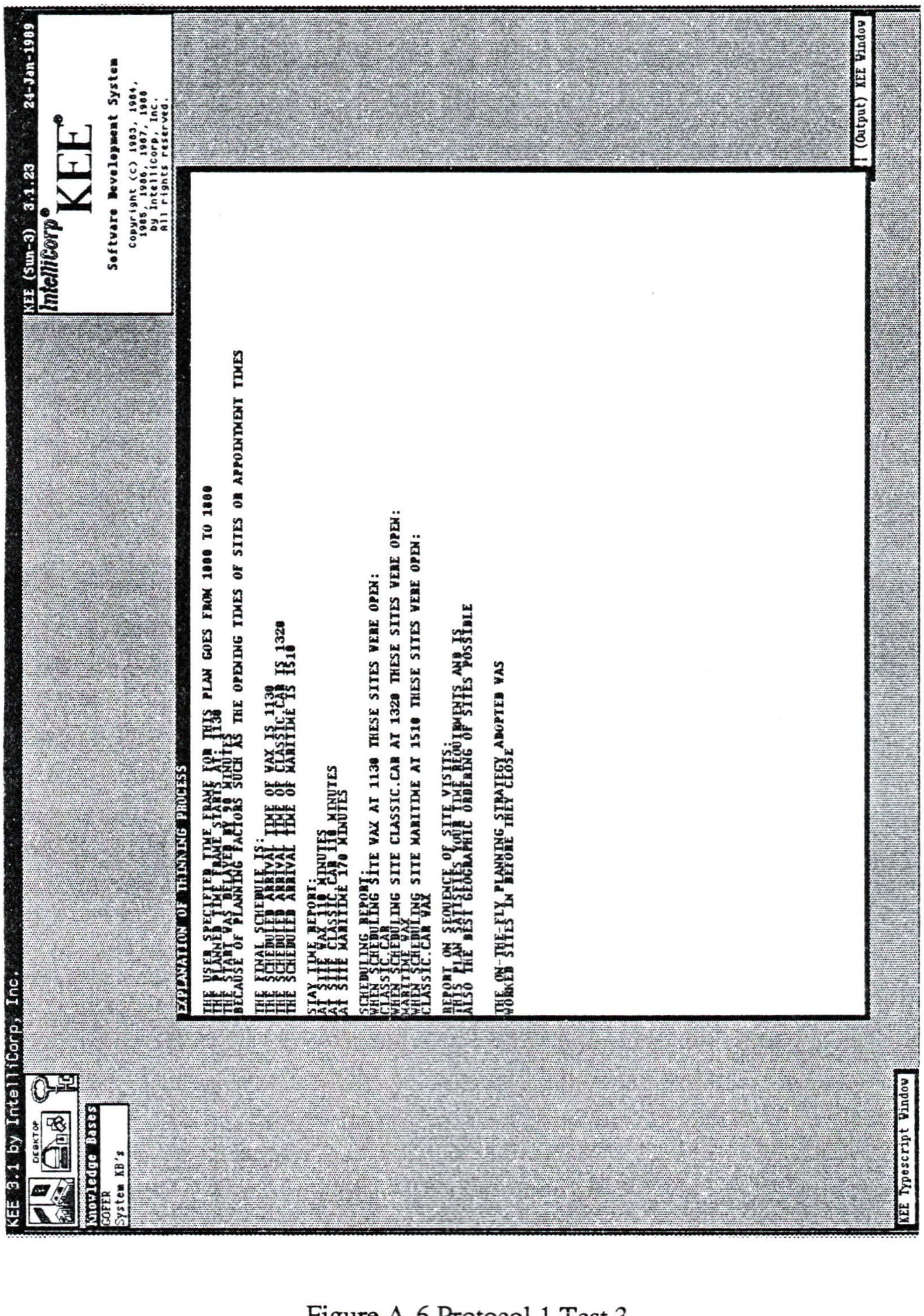

Figure A-6 Protocol 1 Test 3

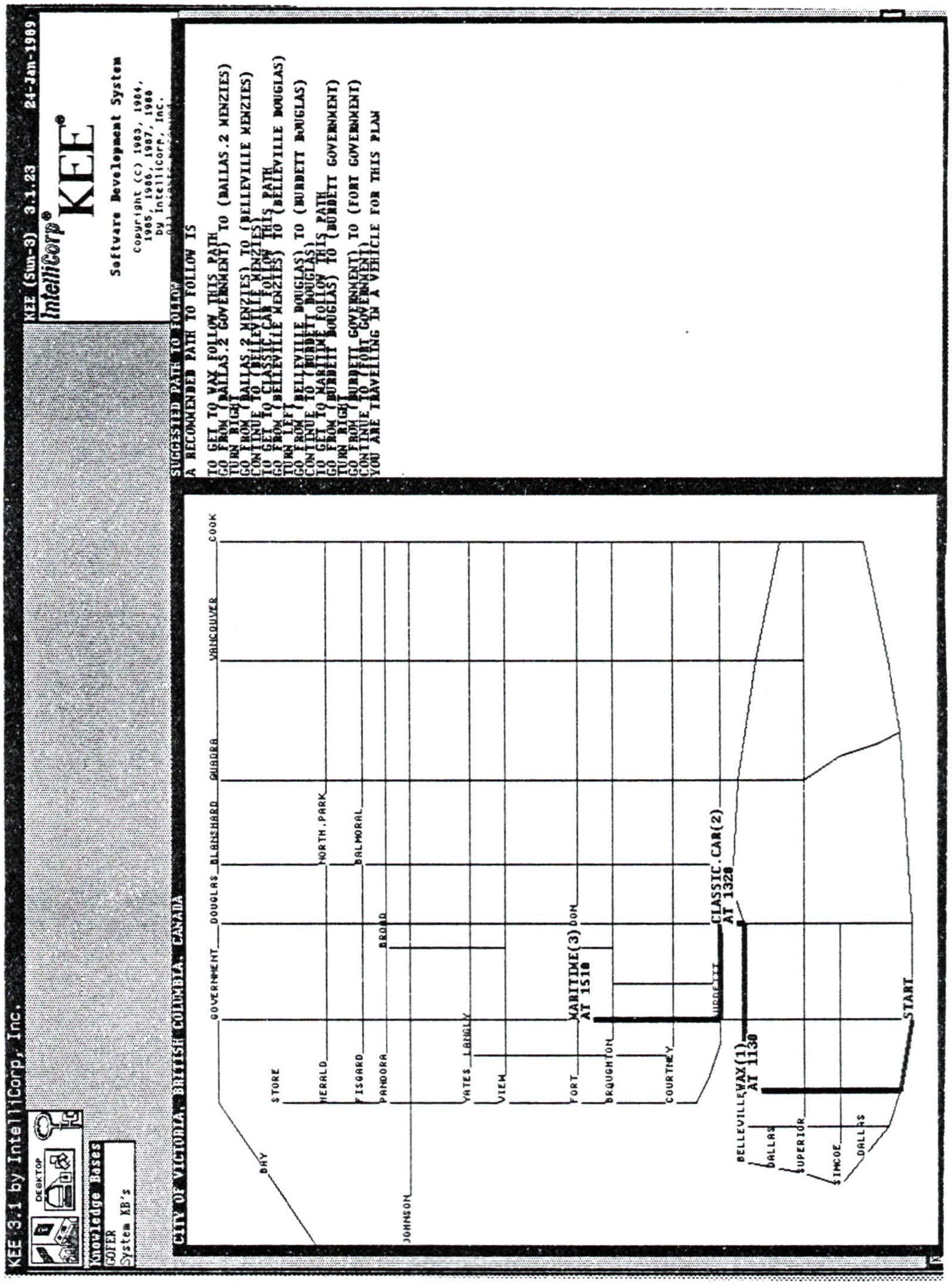

Figure A-7 Protocol 1 Test 4

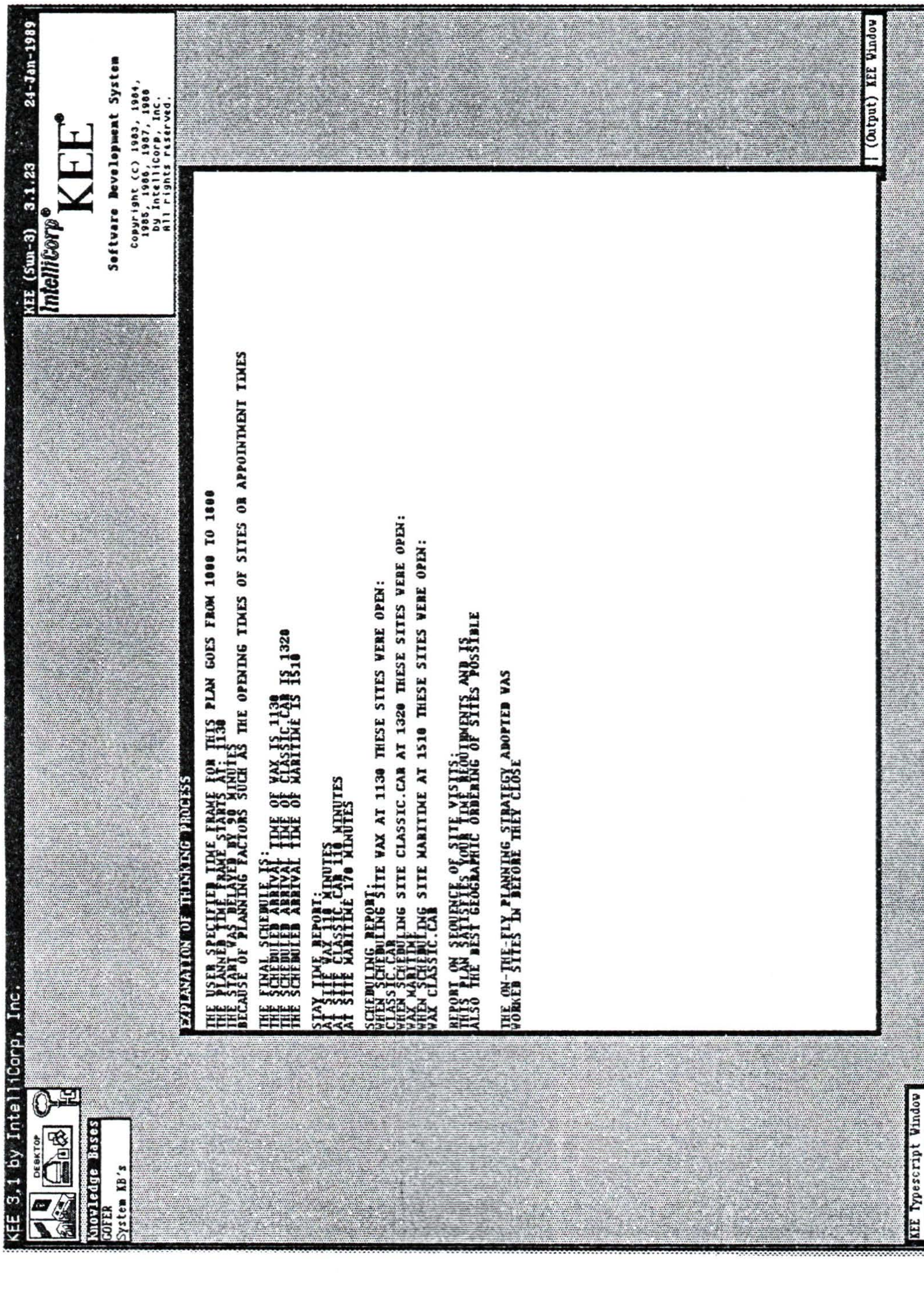

Figure A-8 Protocol 1 Test 4

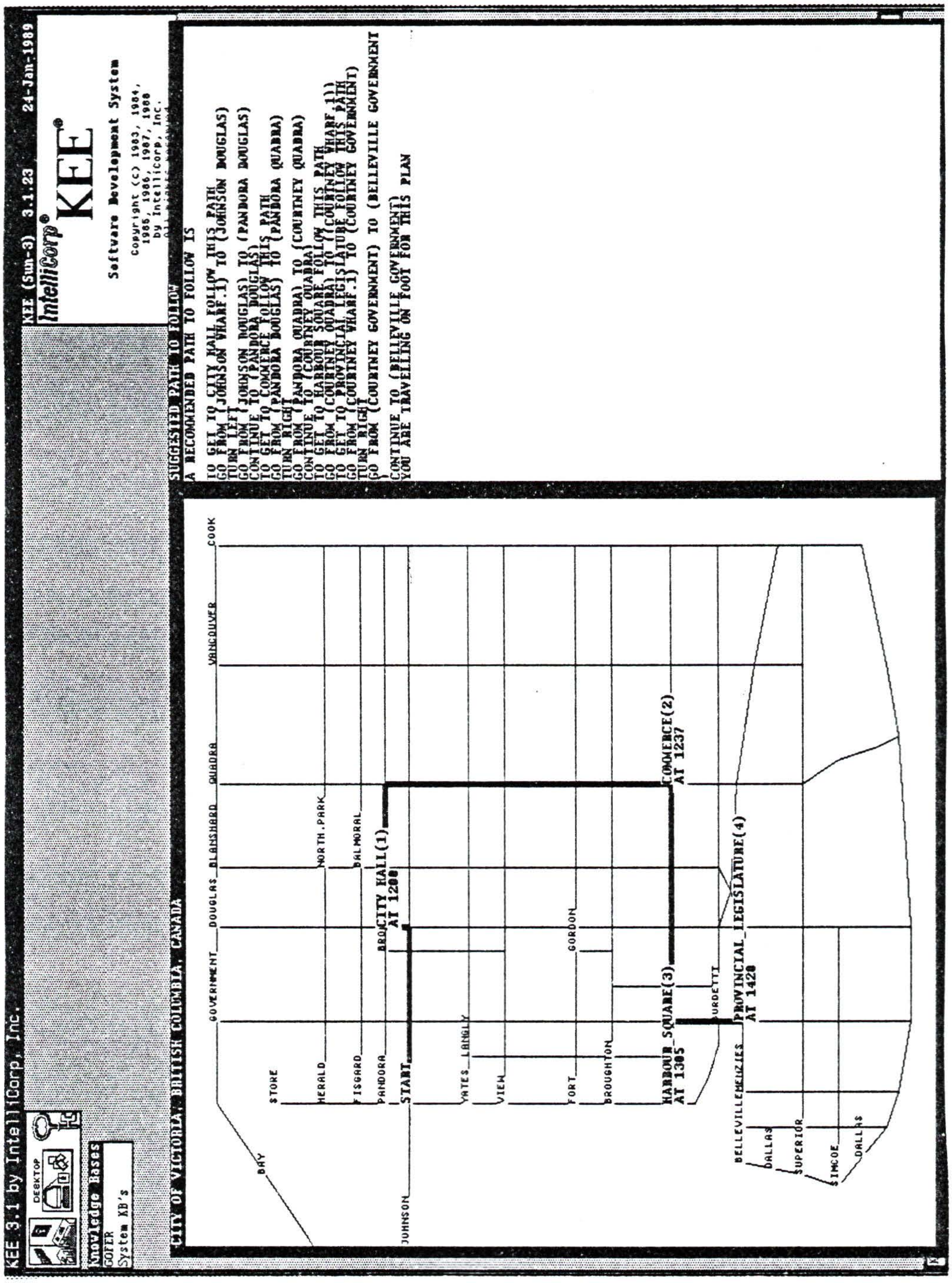

Figure A-9 Protocol 1 Test 5

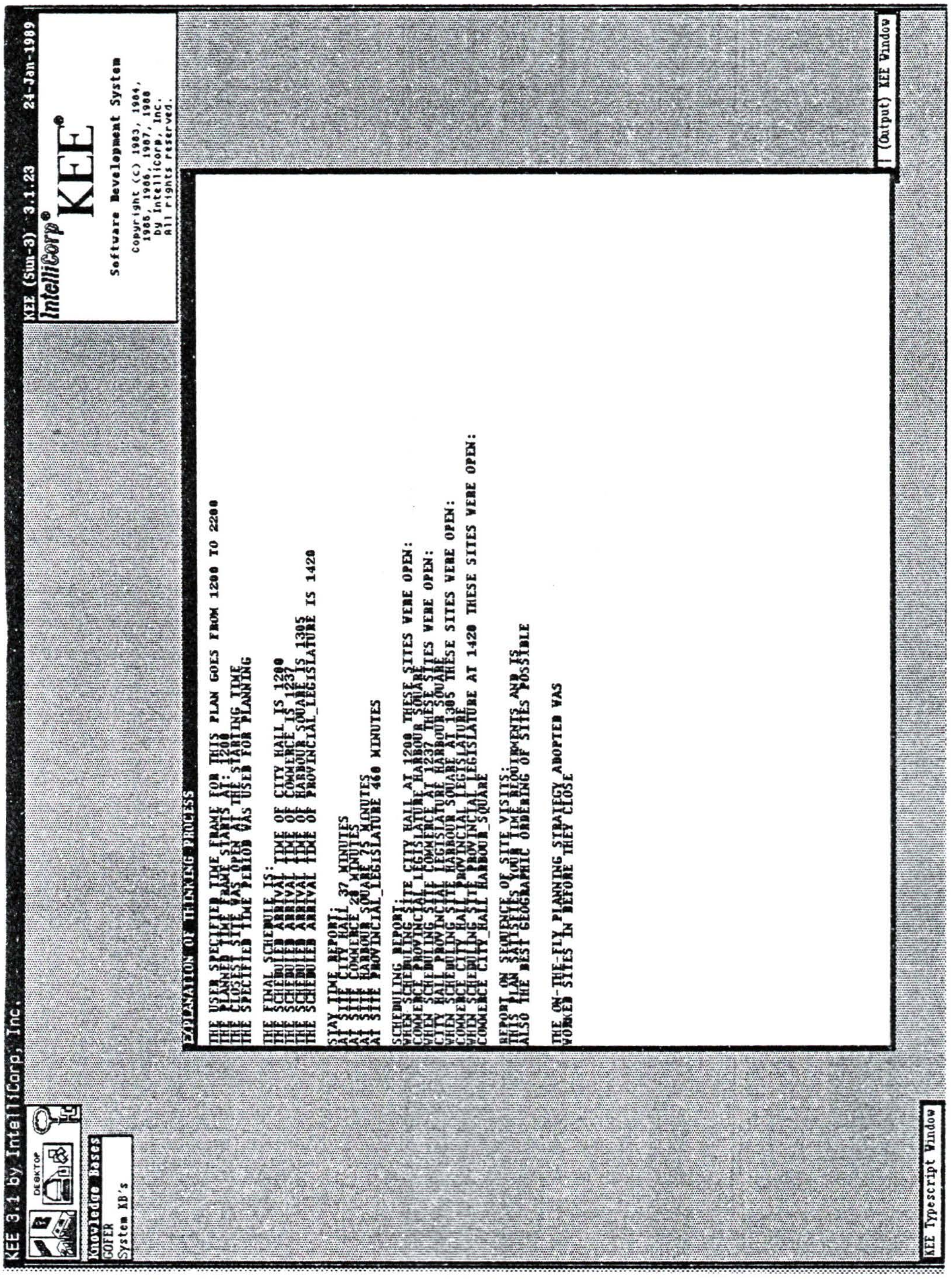

Figure A-10 Protocol 1 Test 5

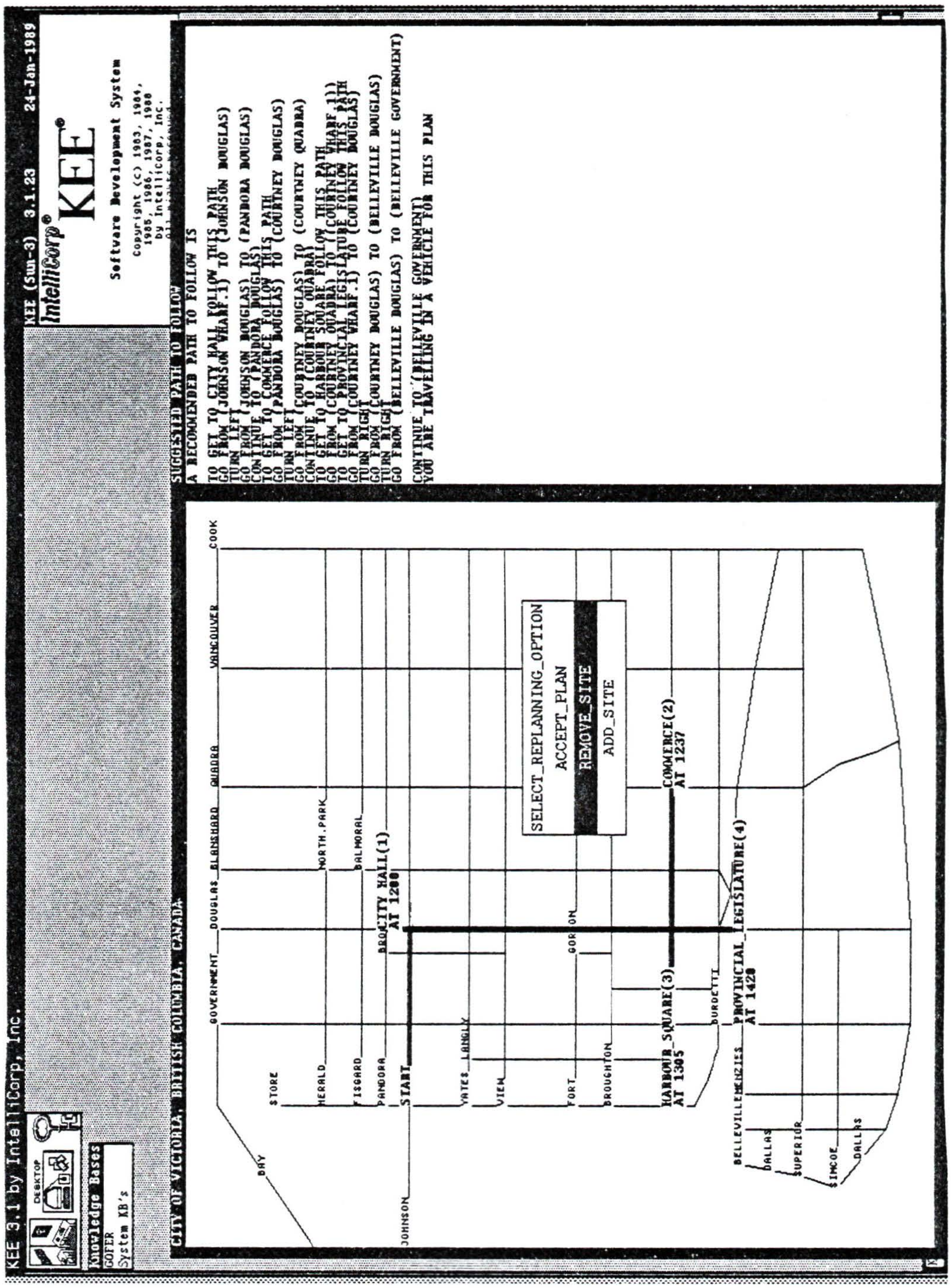

Figure A-11 Protocol 1 Test 6

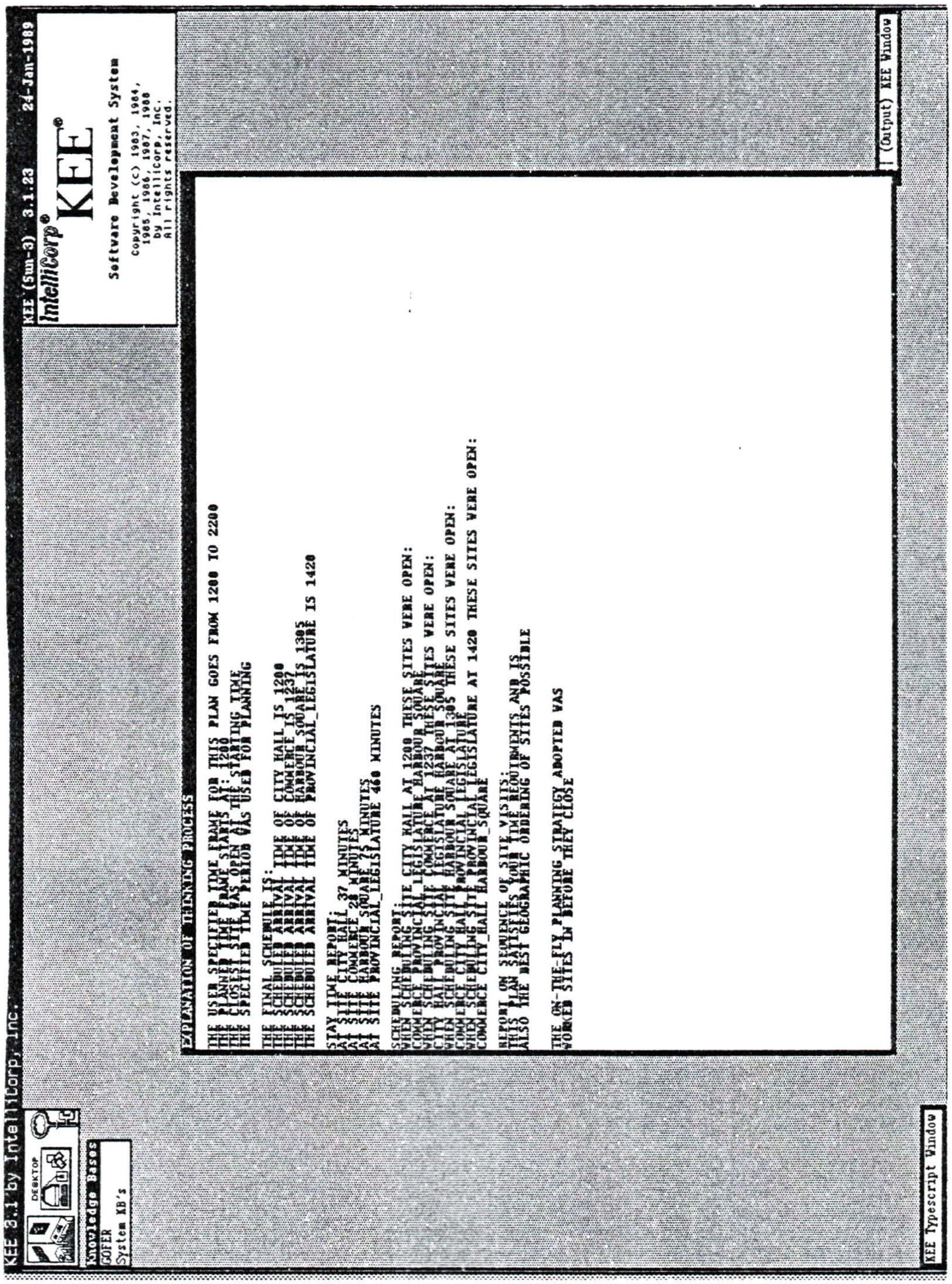

Figure A-12 Protocol 1 Test 6

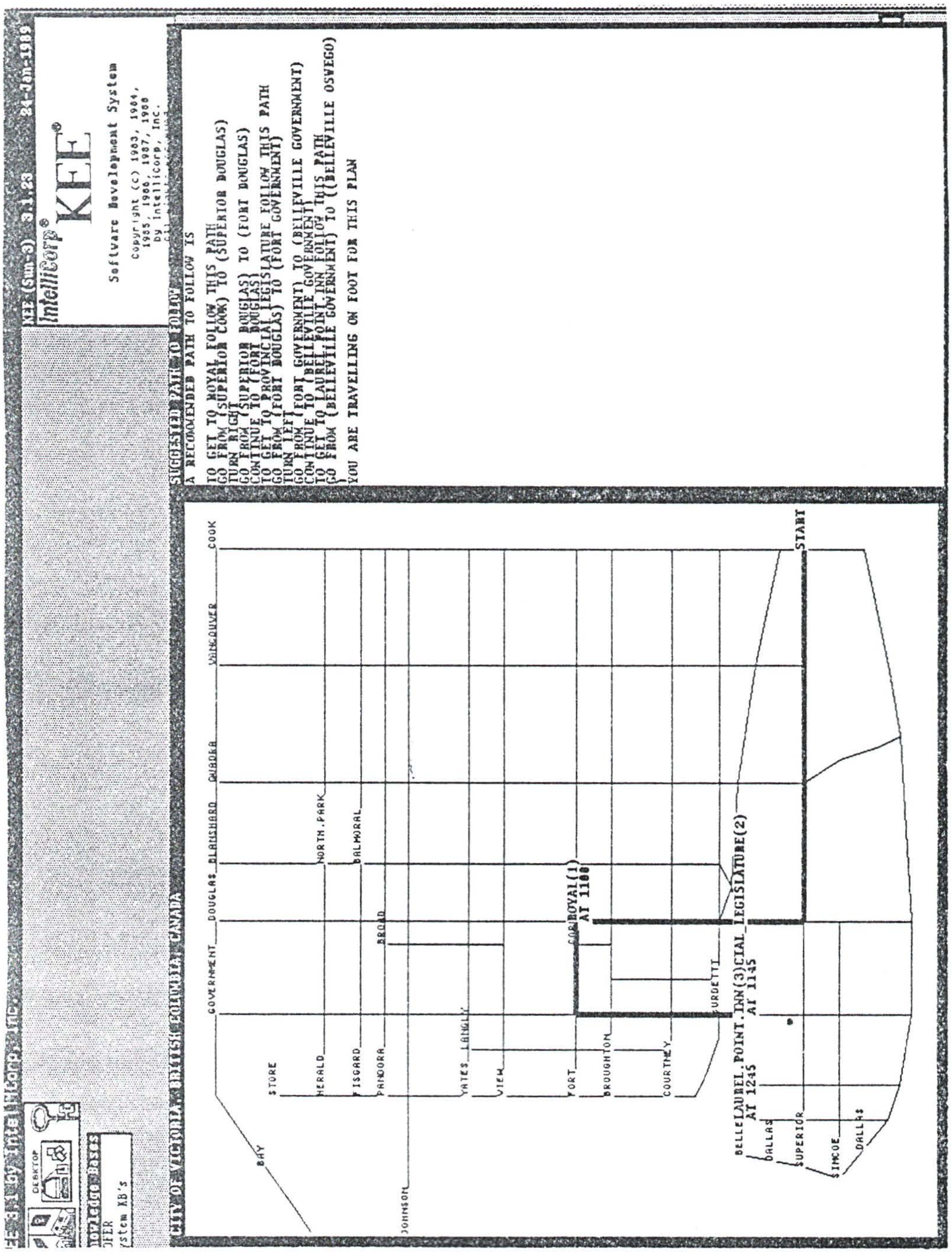

Figure A-13 Protocol 1 Test 7

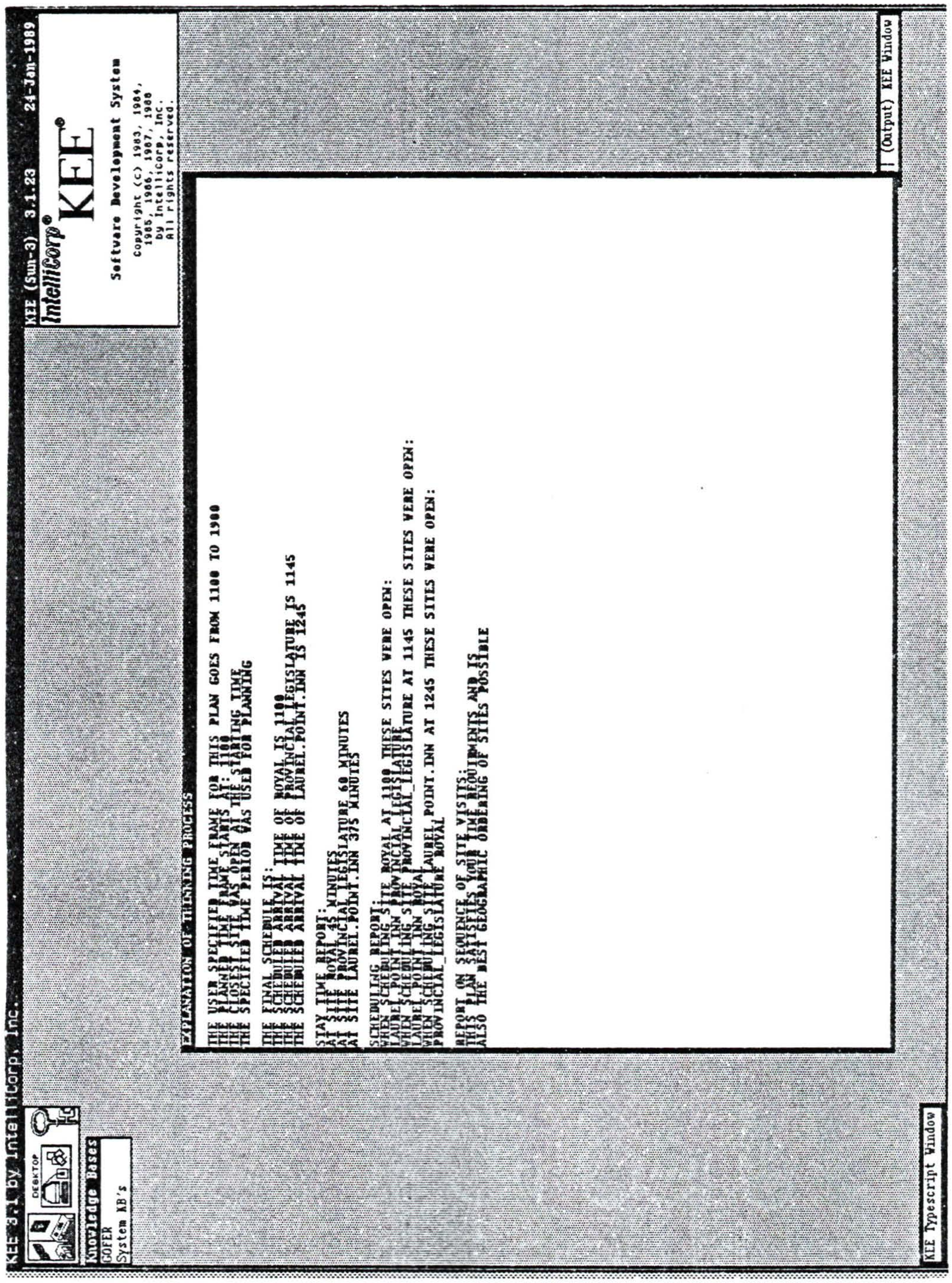

Figure A-14 Protocol 1 Test 7

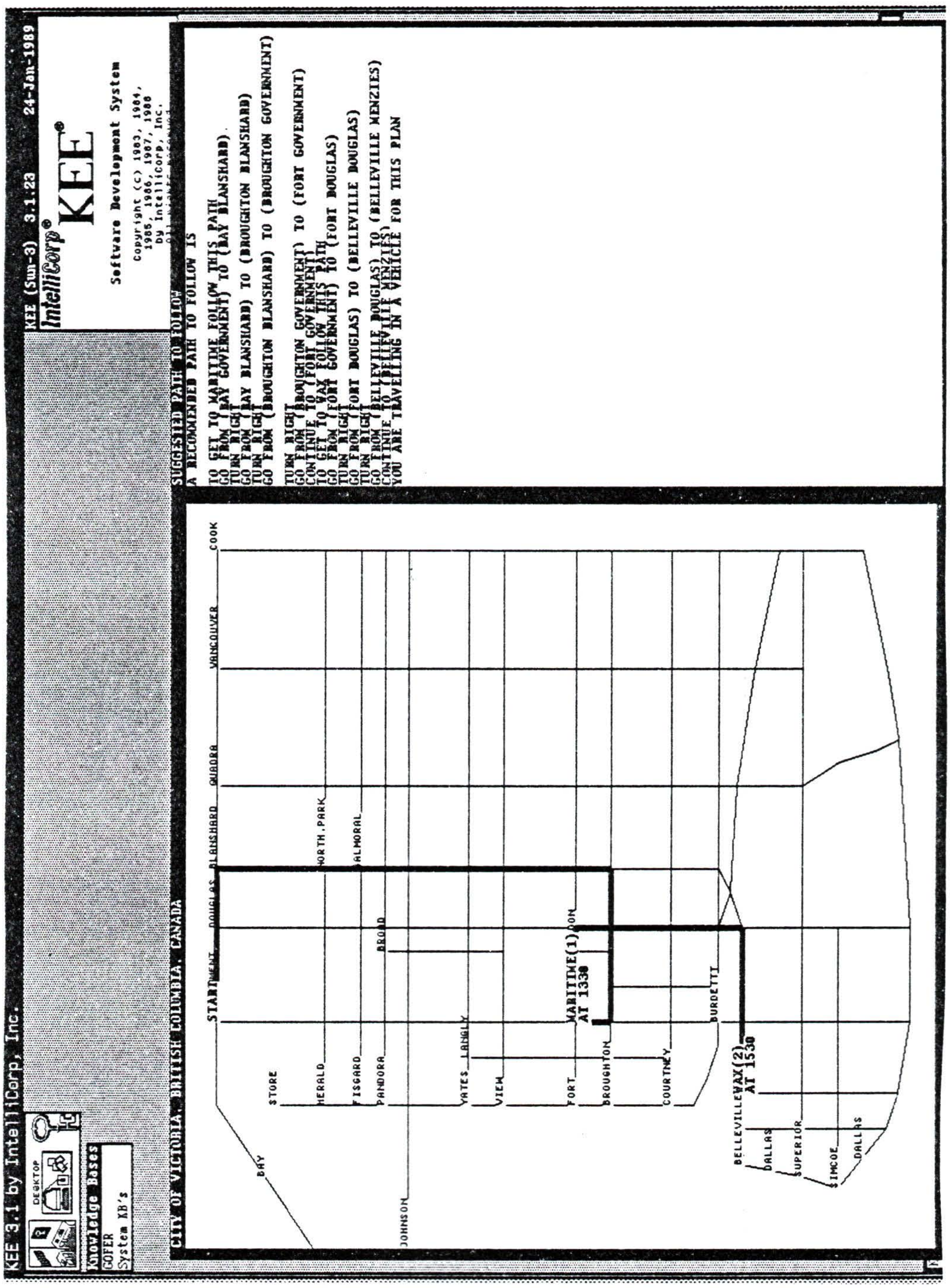

Figure A-15 Protocol 1 Test 8

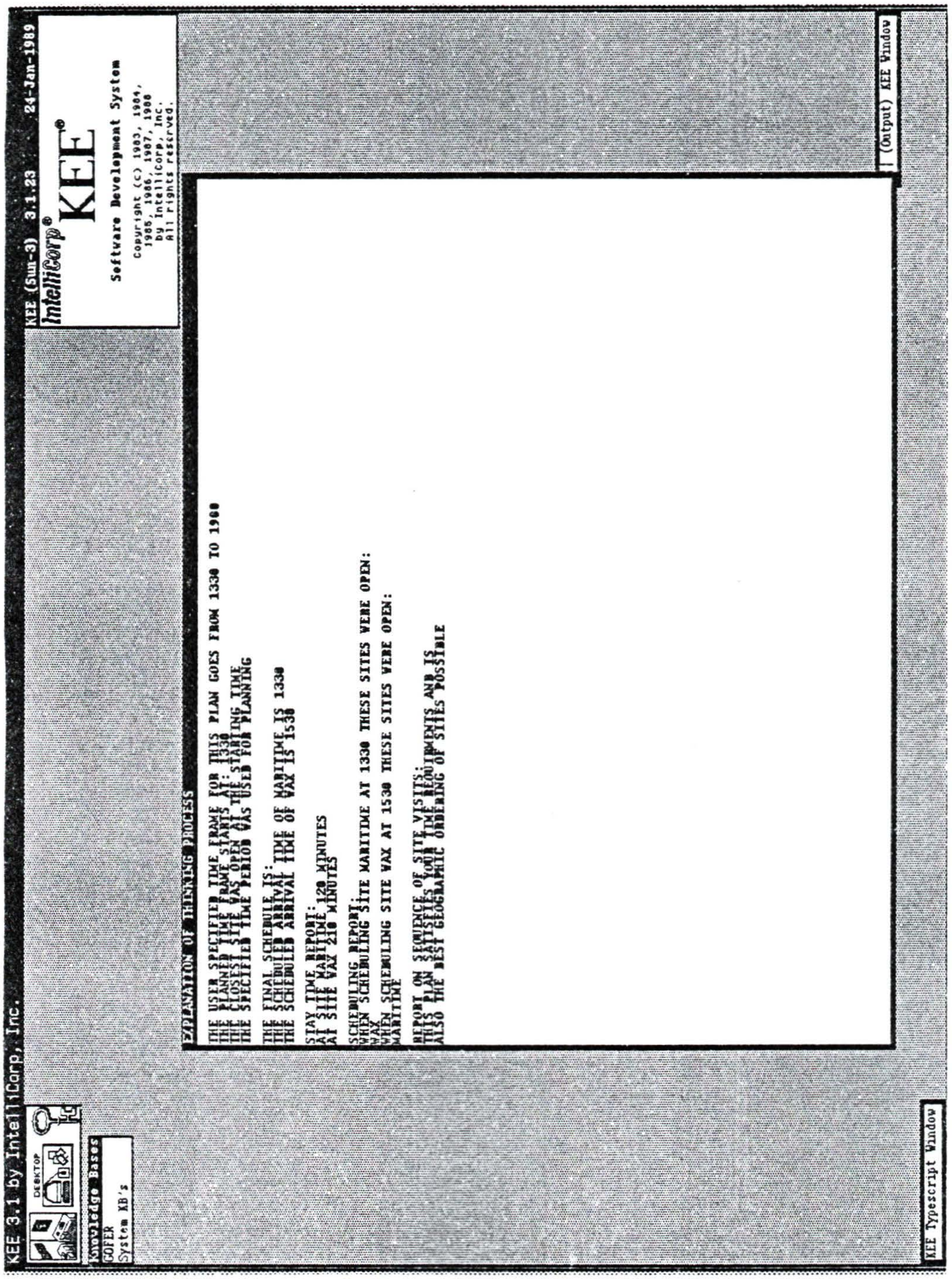

Figure A-16 Protocol 1 Test 8

A-1-2 Protocol 2

Test 1

Time Span: 1000-1400

Sites: Eatons, Harbour Square, City Hall, Commerce Bank

Mode of Travel: foot

Starting Intersection: (View Cook)

Appointments: Nil

Processing Time: 0:53

Test 2

Time Span: 1000-1400

Sites: Eatons, Harbour Square, City Hall, Commerce Bank

Mode of Travel: foot

Starting Intersection: (View Cook)

Appointments: Commerce Bank at 1130 for 60 minutes

Processing Time: 2:33

Test 3

Time Span: 1000-1500

Sites: Wax Museum, Empress Hotel, Maritime Museum,

Commerce Bank, Royal Bank

Mode of Travel: car

Starting Intersection: (Johnson Cook)

Appointments: Commerce Bank 1030 for 45 minutes,

Royal Bank 1400 for 30 minutes

Processing Time: 3:30

Test 4

Time Span: 1600-2000

Sites: Commerce Bank, City Hall

Mode of Travel: car

Starting Intersection: (Superior Dallas)

Appointments: Commerce Bank at 1500 for 30 minutes

Notes: this should prompt tourist correction of input conditions since the time span starts after the appointment is planned, then time changed to 1500 to 2000.

Processing Time: 0:50

Test 5

Time Span: 0900-1600

Sites: Harbour Square, Market Square, Eatons, Royal Bank

Mode of Travel: foot

Starting Intersection: (Johnson Government)

Appointments: Royal Bank at 1100 for 30 minutes

Notes: this should test the delayed starting rules since all sites will still be possible is the start is delayed until 1000 when the nearest site to the start is open (Market Square).

Processing Time: 2:45

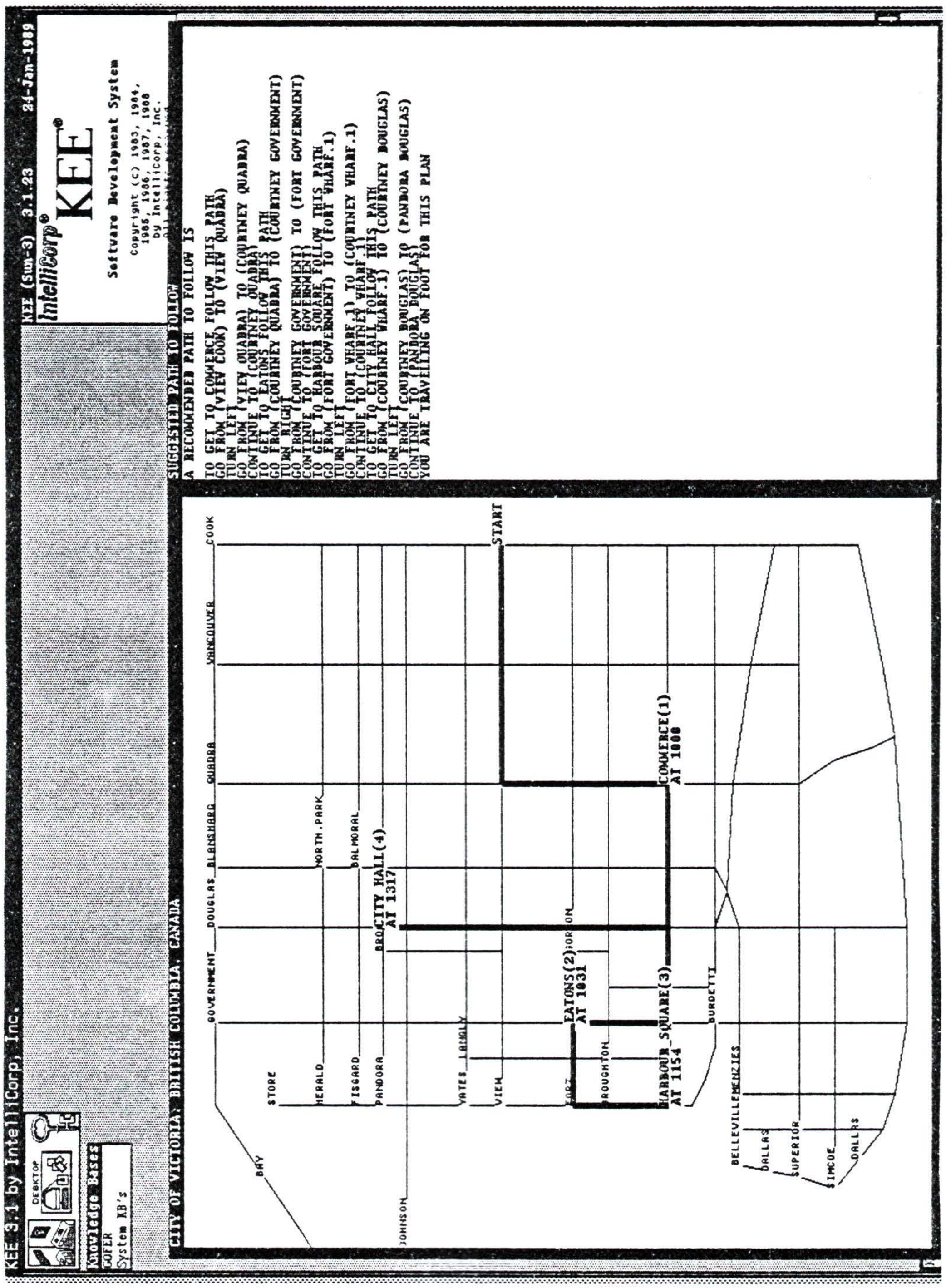

Figure A-17 Protocol 2 Test 1

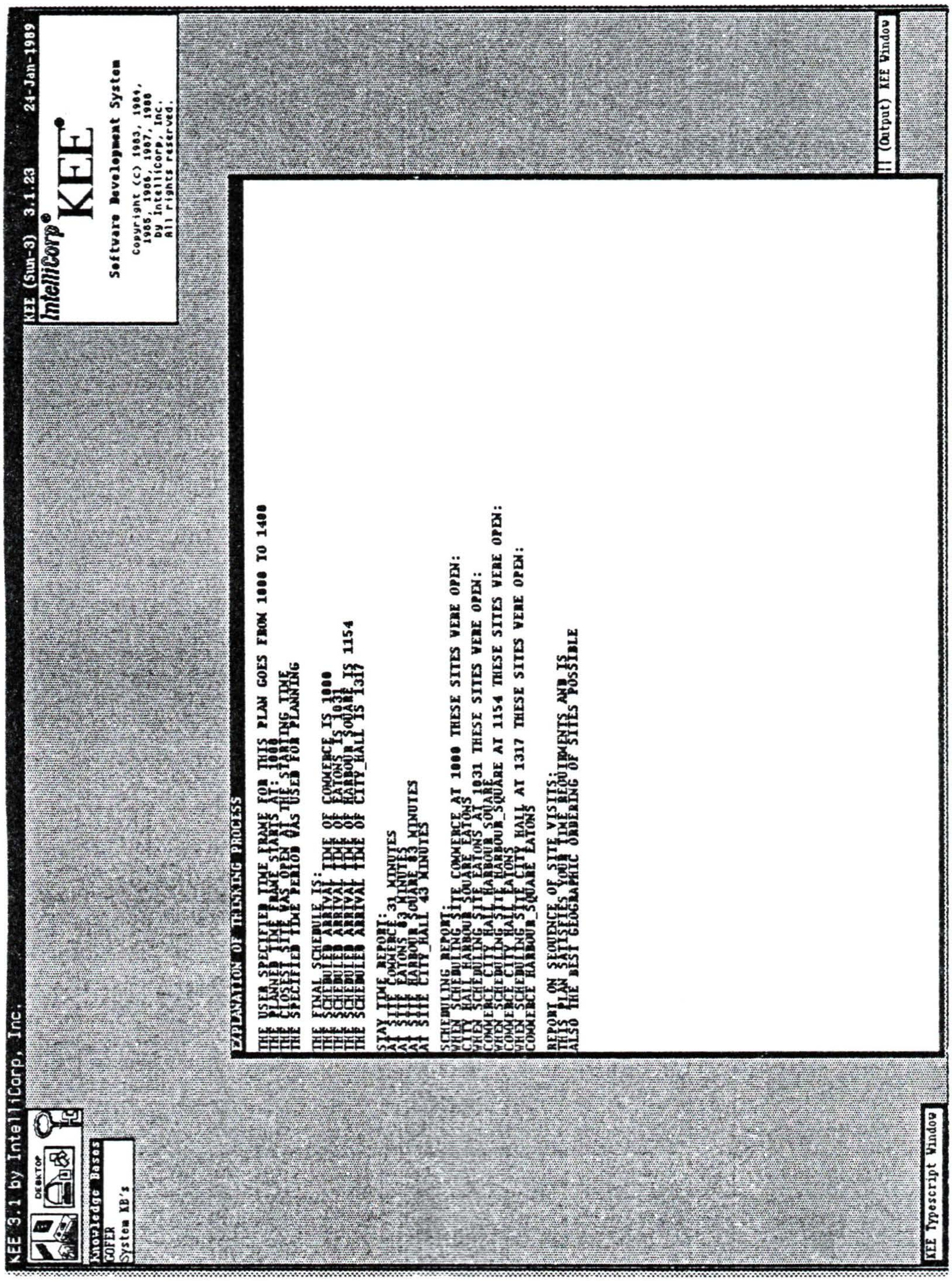

Figure A-18 Protocol 2 Test 1

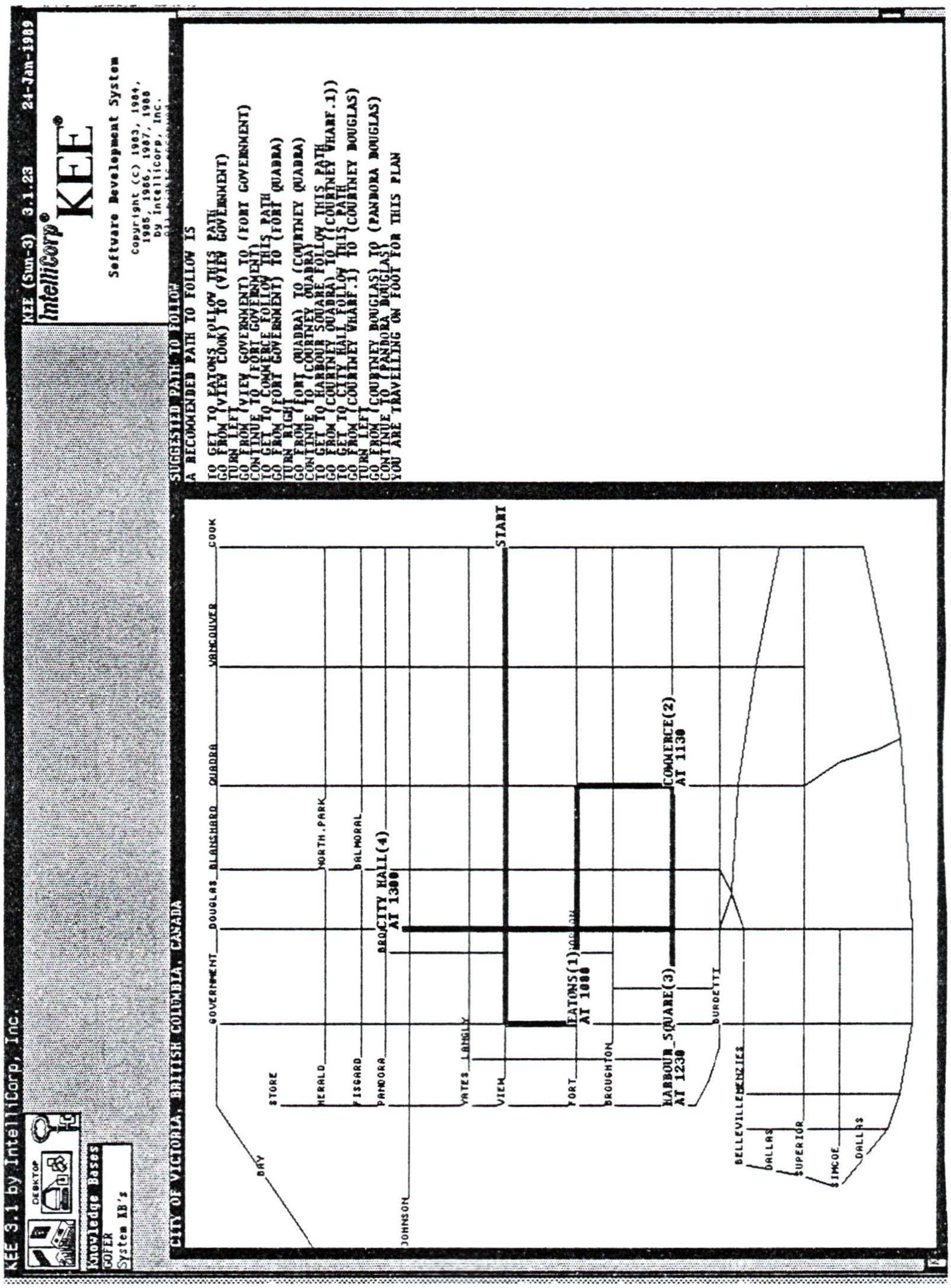

Figure A-19 Protocol 2 Test 2

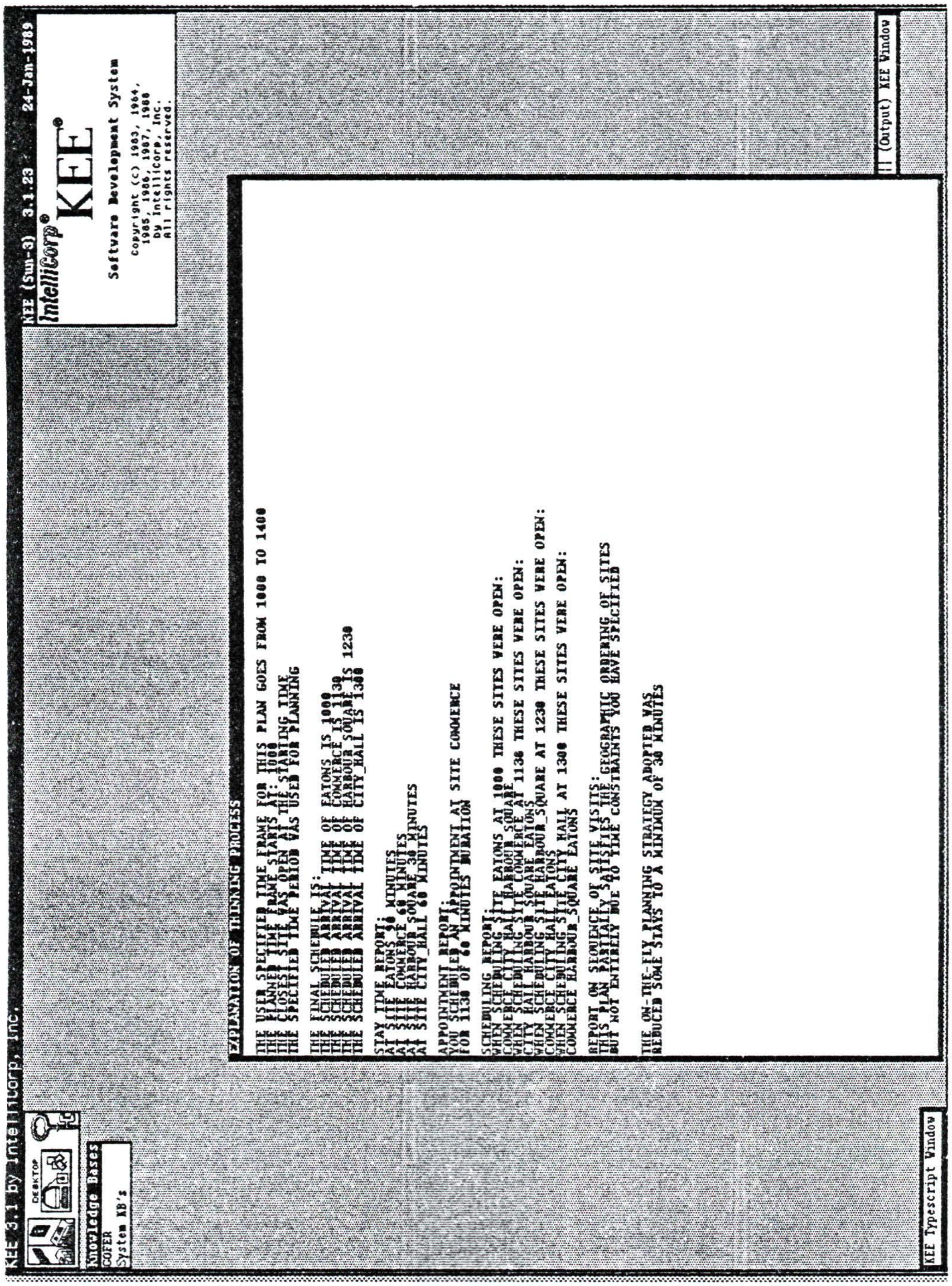

Figure A-20 Protocol 2 Test 2

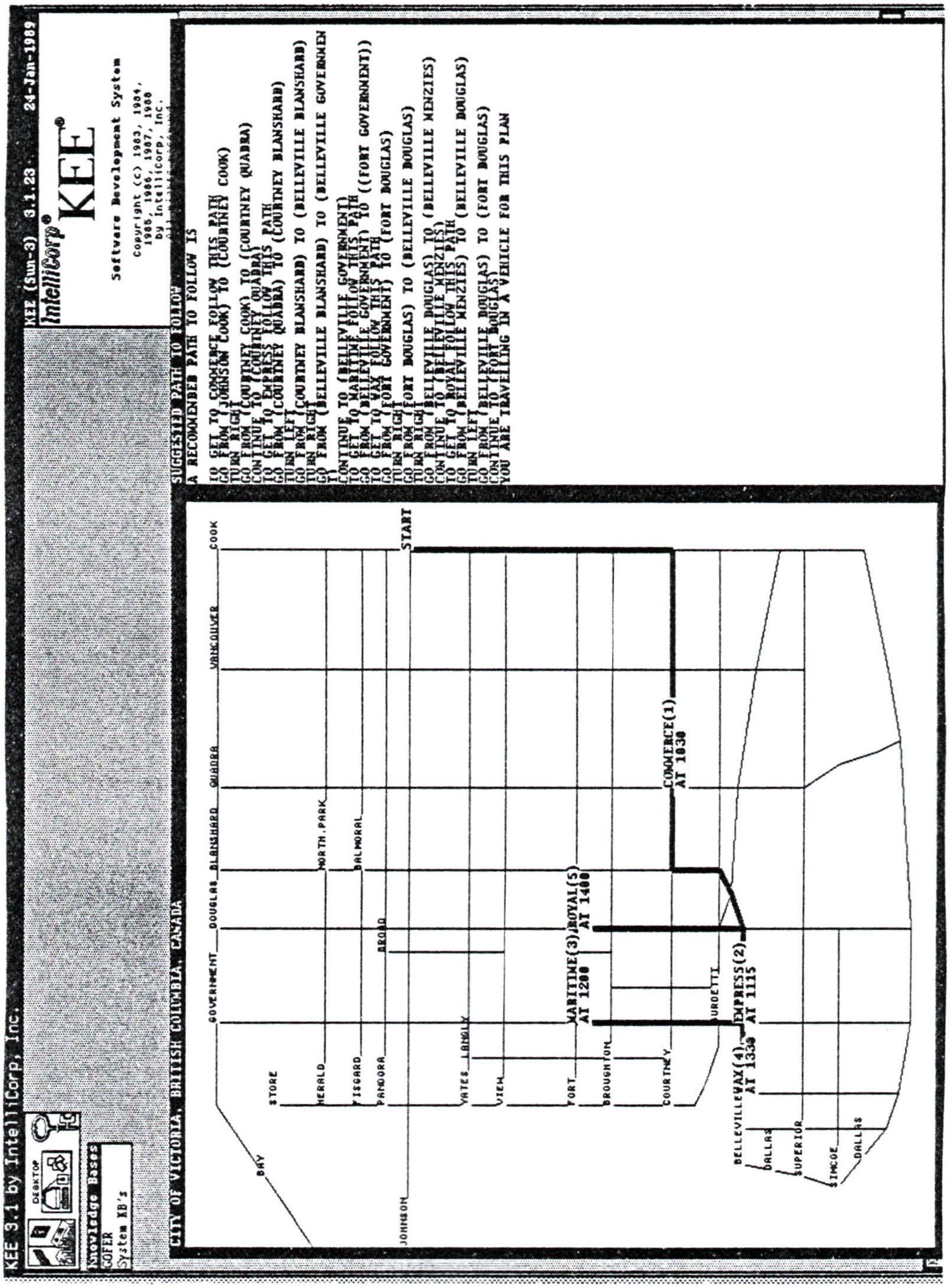

Figure A-21 Protocol 2 Test 3

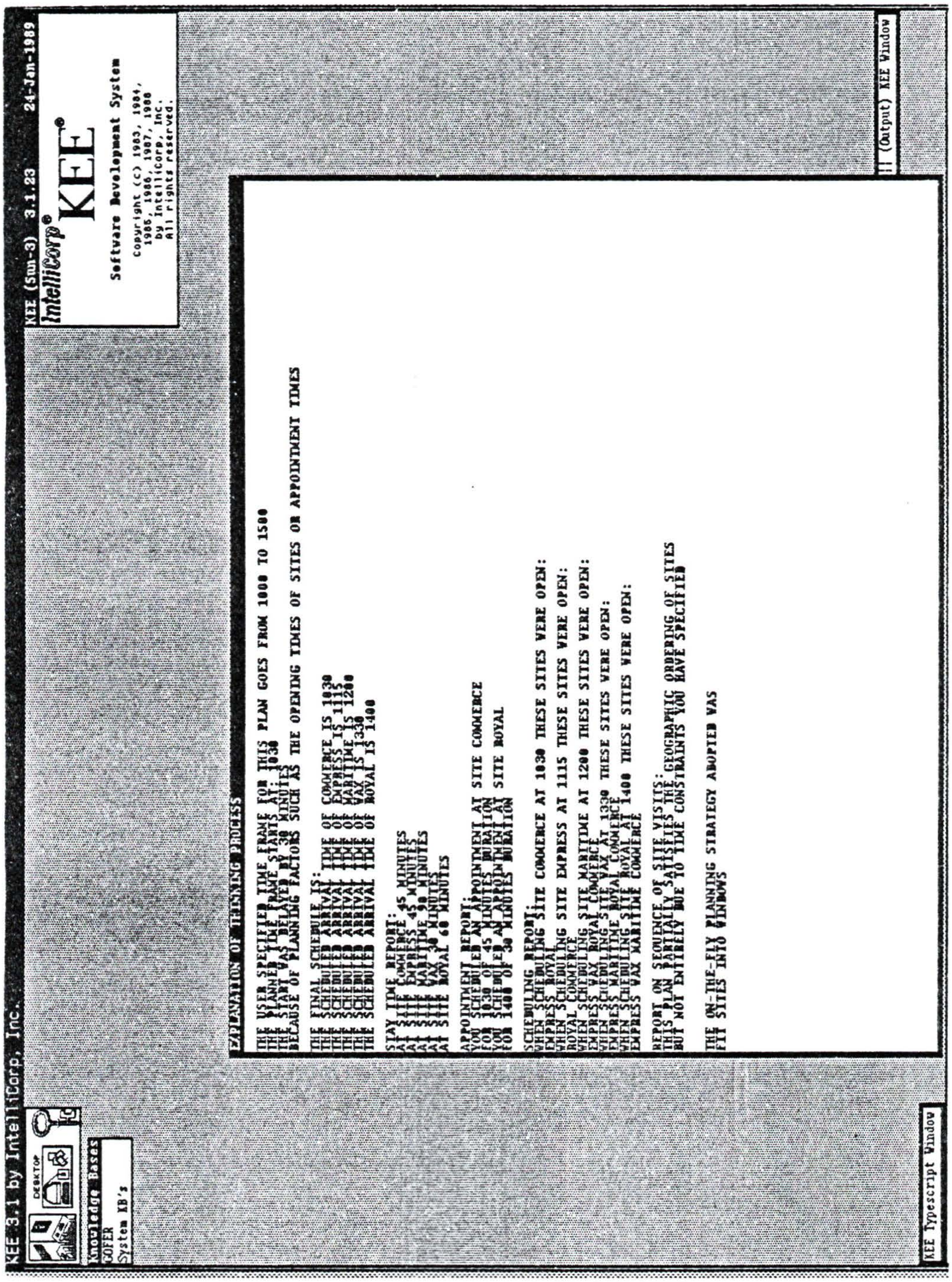

Figure A-22 Protocol 2 Test 3

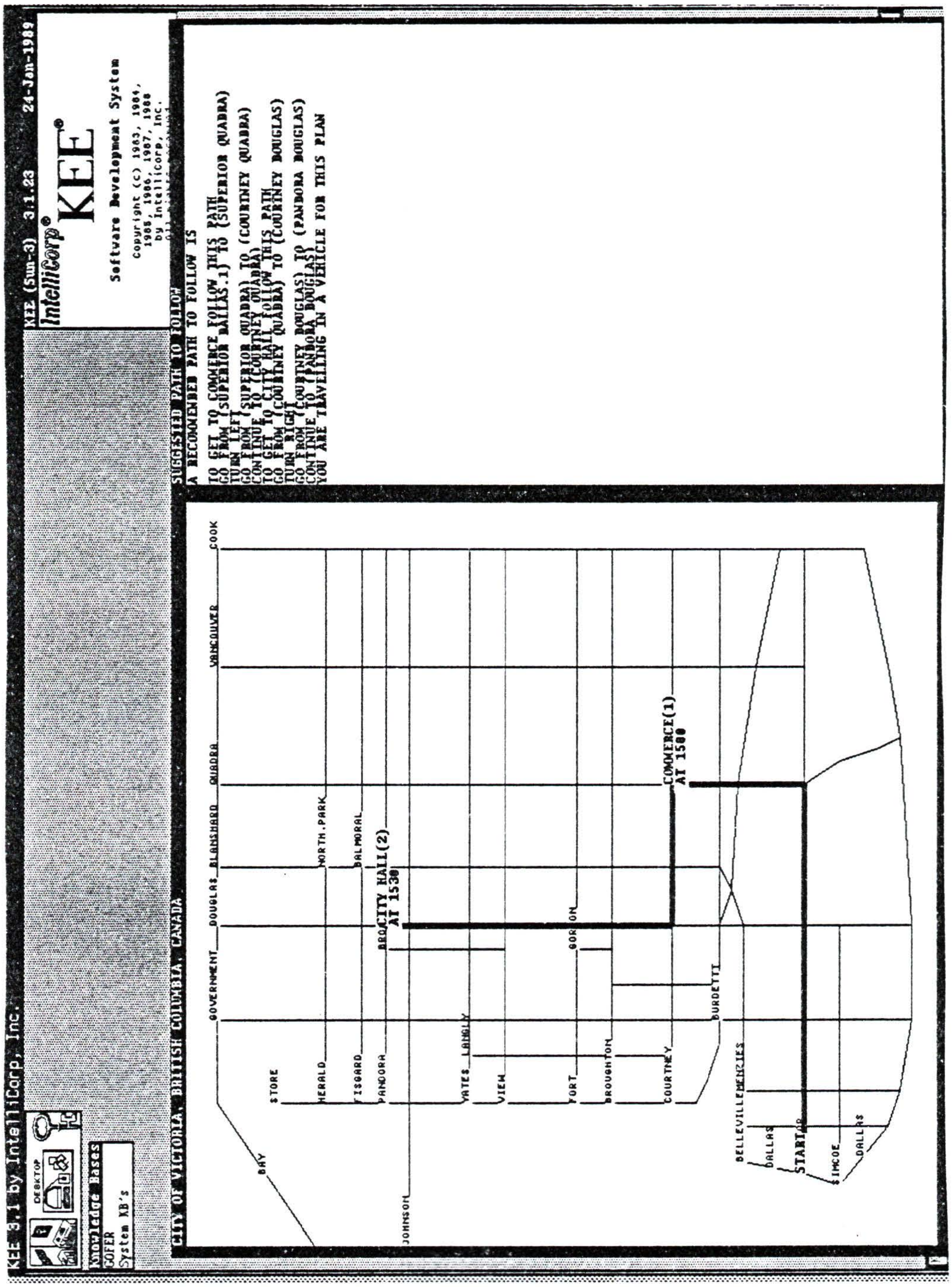

Figure A-23 Protocol 2 Test 4

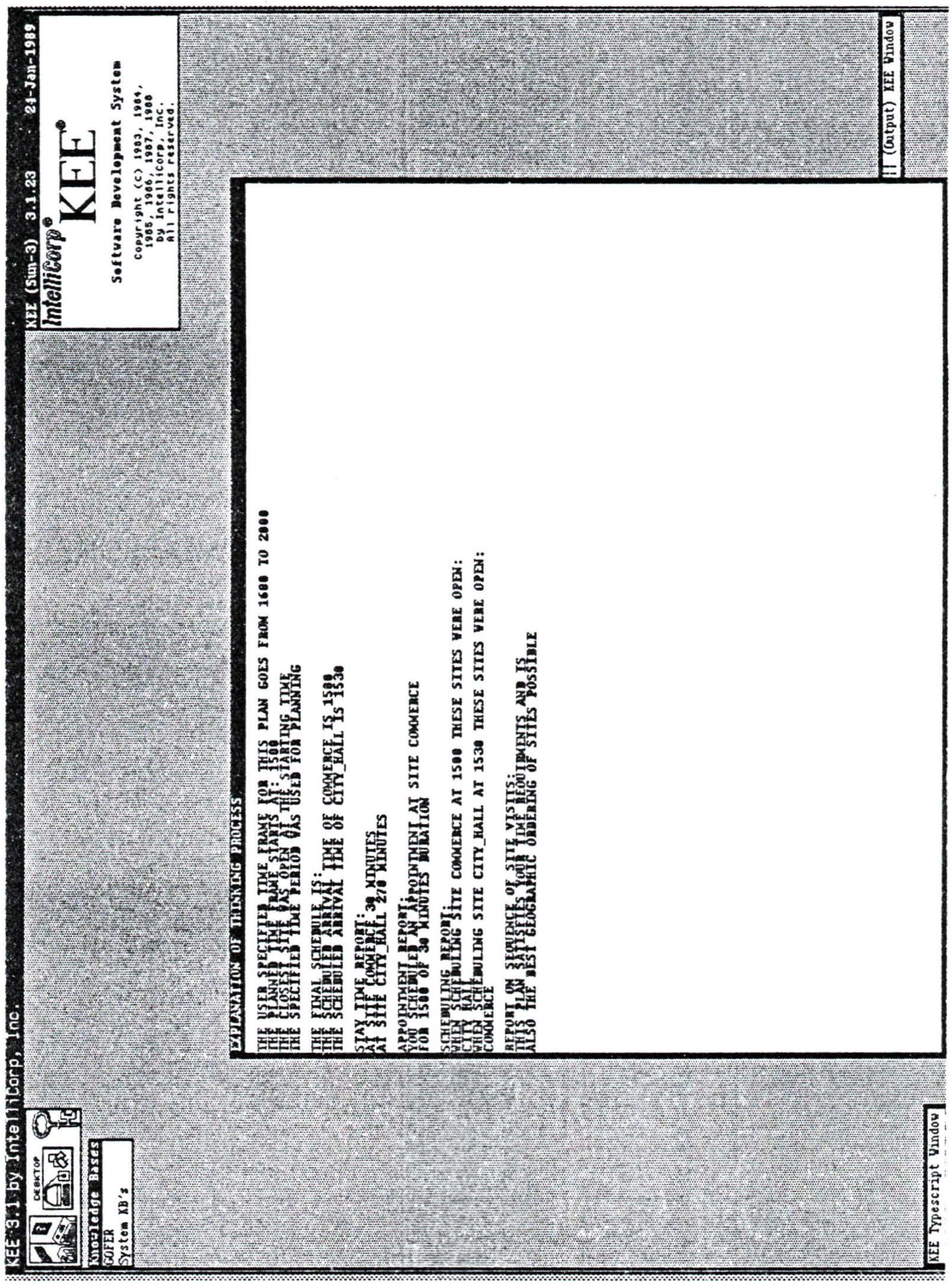

Figure A-24 Protocol 2 Test 4

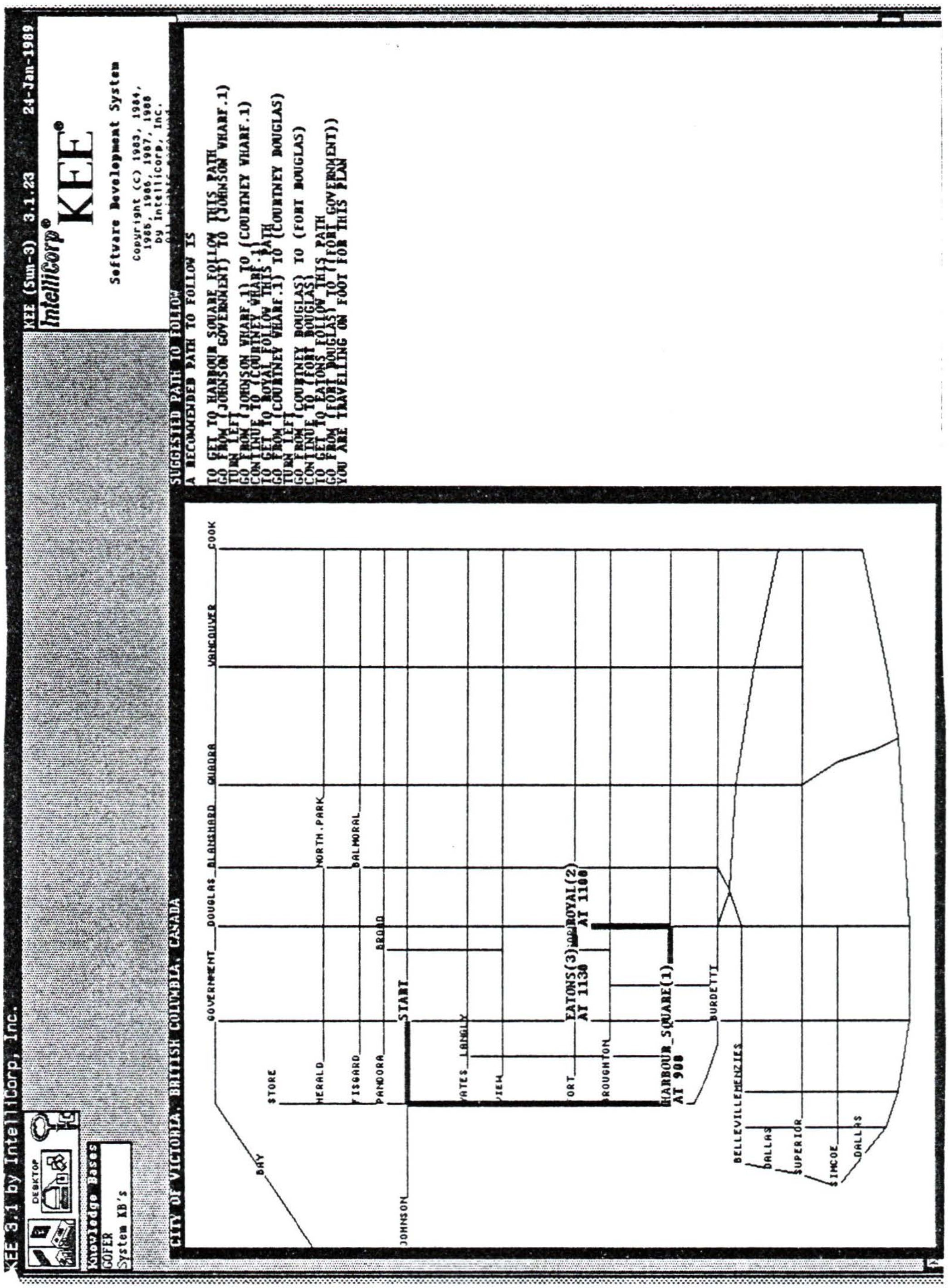

Figure A-25 Protocol 2 Test 5

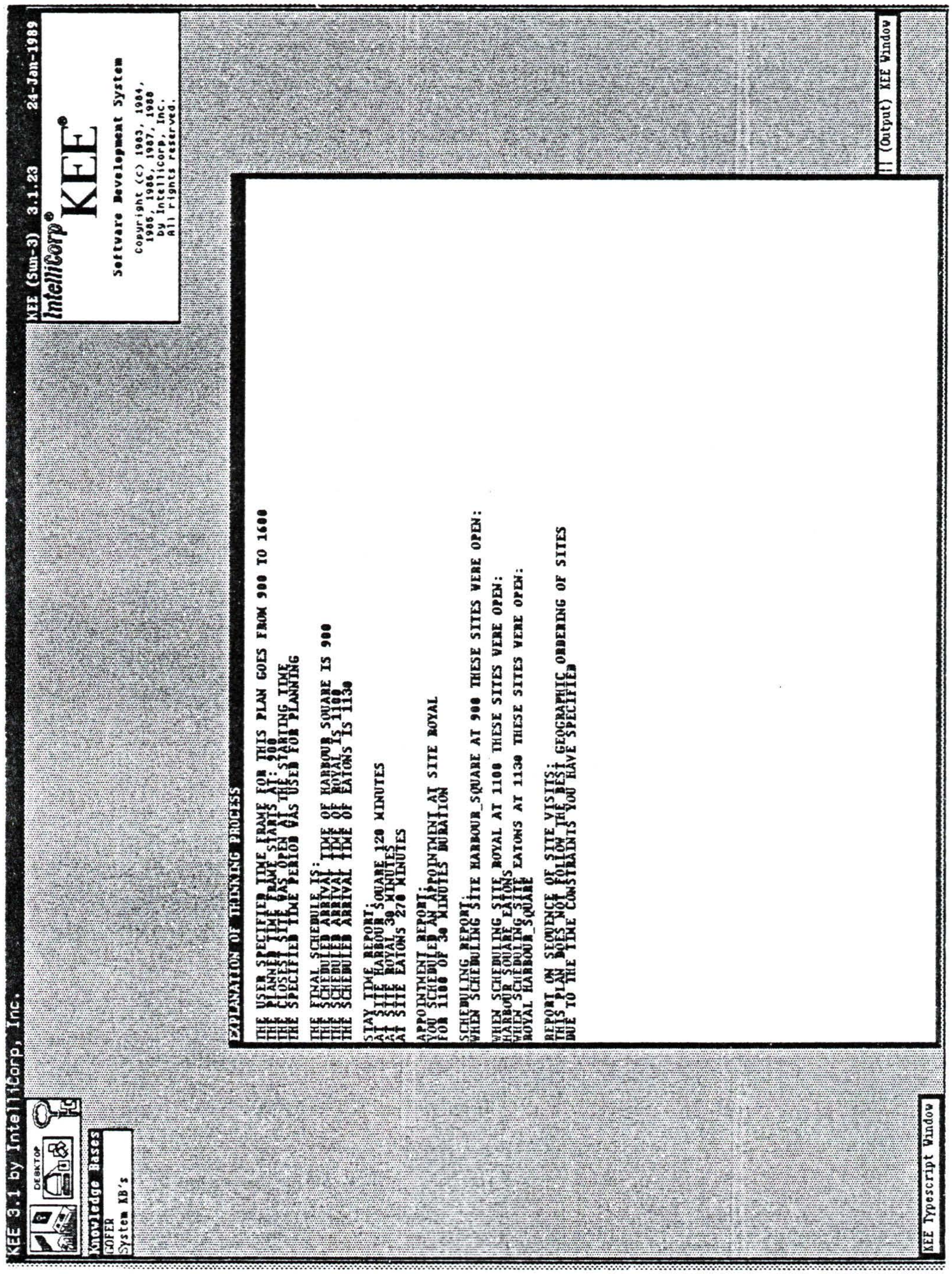

Figure A-26 Protocol 2 Test 5

A-1-3 Protocol 3

Test 1

Time Span: 0900-1900

Sites: Police Station, Empress Hotel

Mode of Travel: car

Starting Intersection: (Belleville Menzies)

Appointments: Nil

Notes: tourist accepts police site visit urgent

Processing Time: 0:35

Test 2

Time Span: 0900-1900

Sites: Police Station, Empress Hotel

Mode of Travel: car

Starting Intersection: (Belleville Menzies)

Appointments: Nil

Notes: tourist does not require police site be visited first

Processing Time: 0:31

Test 3

Time Span: 1000-1900

Sites: Wax Museum, Empress Hotel, Commerce Bank, Royal Bank

Police Station

Mode of Travel: car

Starting Intersection: (Wharf Government)

Appointments: Royal Bank at 1300 for 45 minutes

Notes: police site visit urgent

Processing Time: 3:58

Test 4

Time Span: 1000-1900

Sites: Wax Museum, Empress Hotel, Commerce Bank, Royal Bank

Police Station

Mode of Travel: car

Starting Intersection: (Wharf Government)

Appointments: Royal Bank at 1300 for 45 minutes

Notes: tourist does not require police site be visited first

Processing Time: 4:44

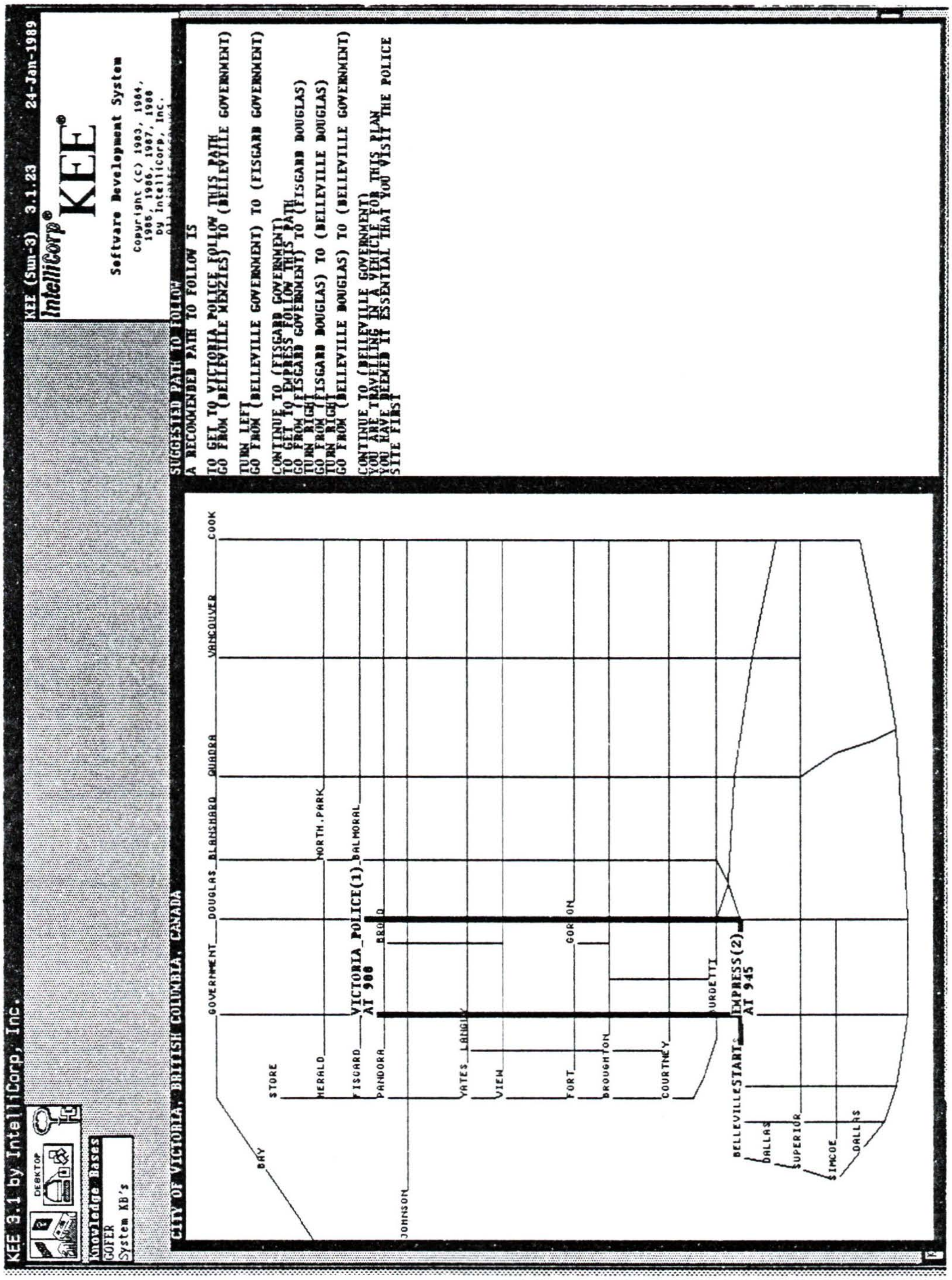

Figure A-27 Protocol 3 Test 1

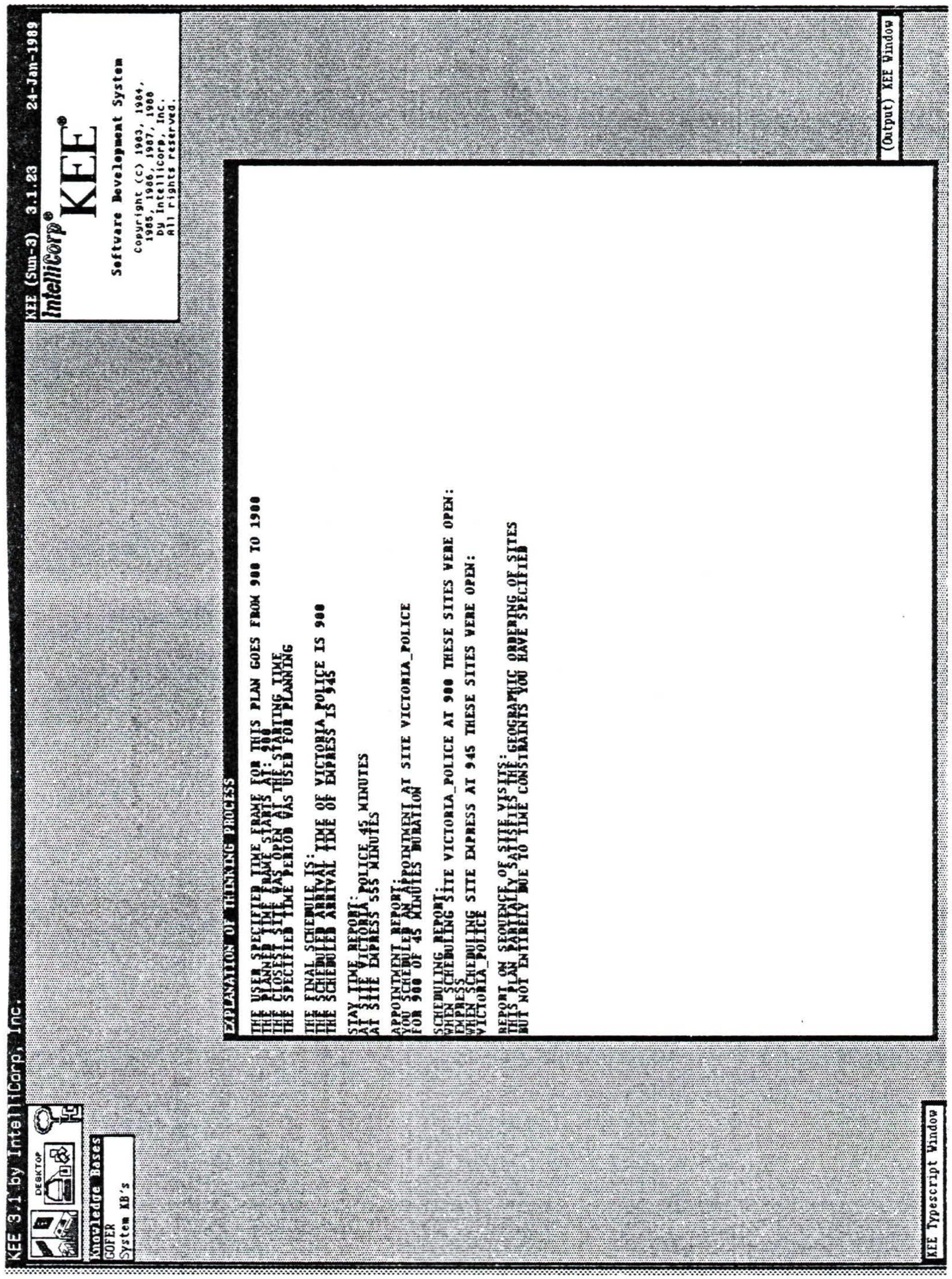

Figure A-28 Protocol 3 Test 1

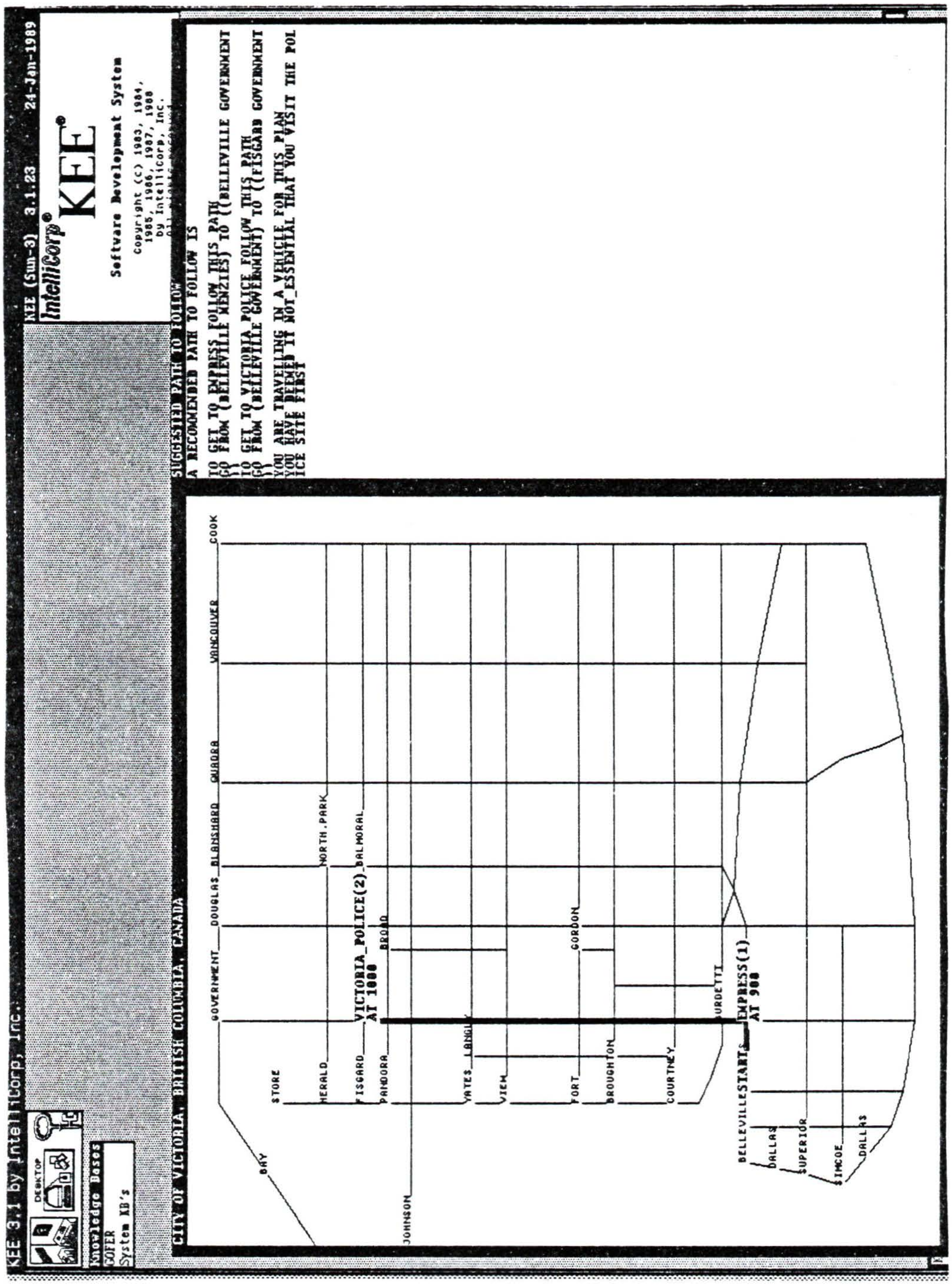

Figure A-29 Protocol 3 Test 2

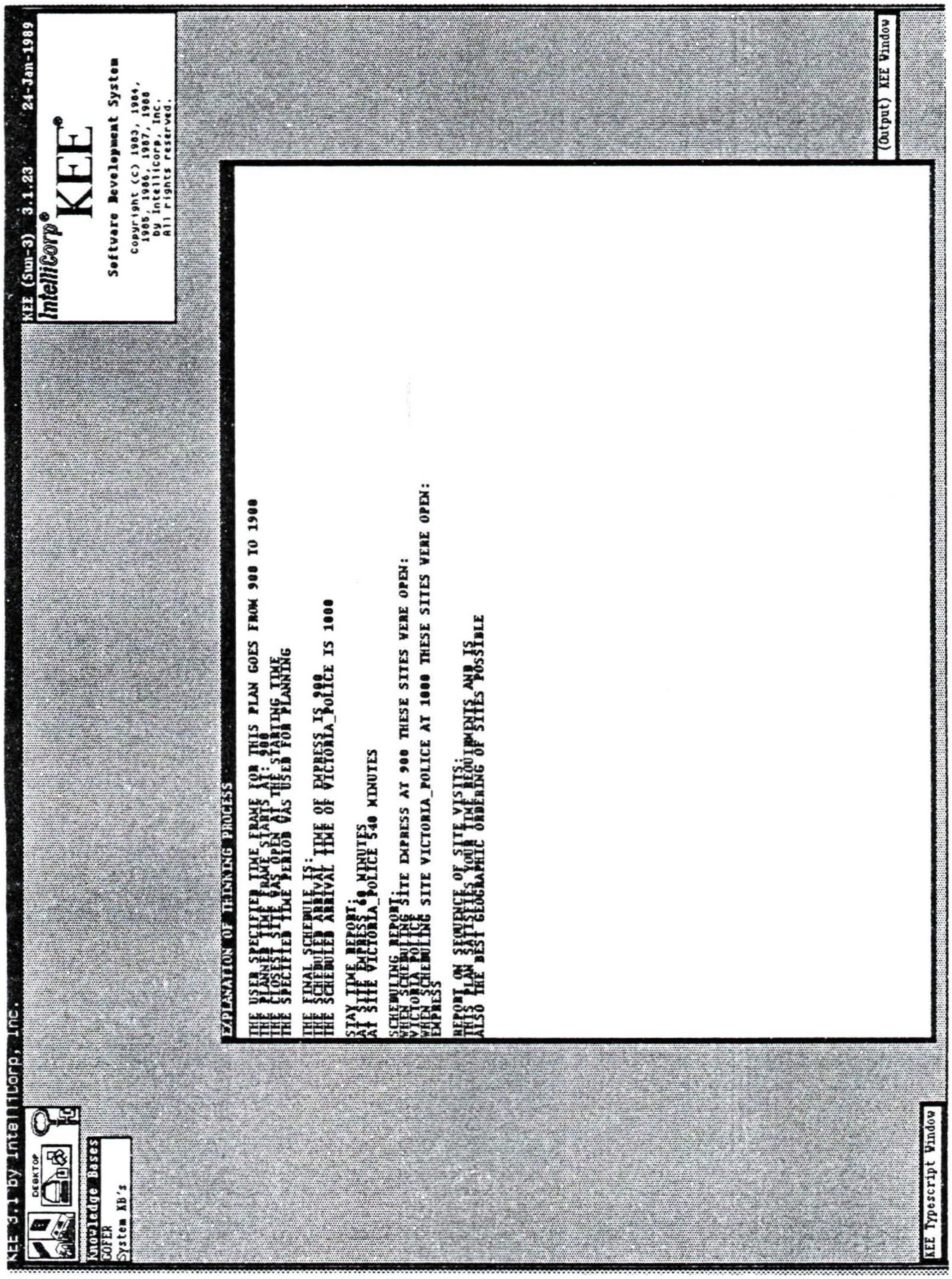

Figure A-30 Protocol 3 Test 2

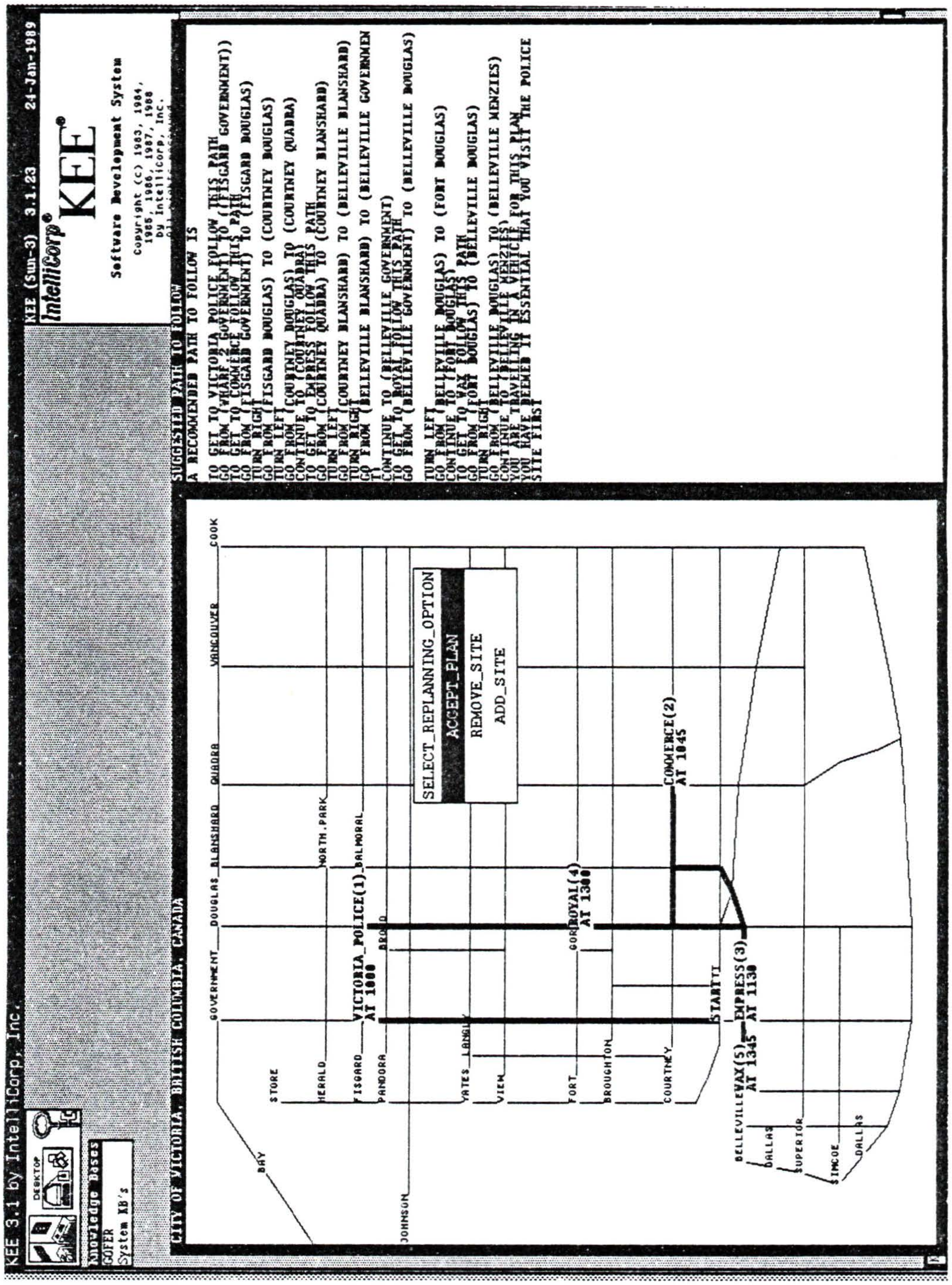

Figure A-31 Protocol 3 Test 3

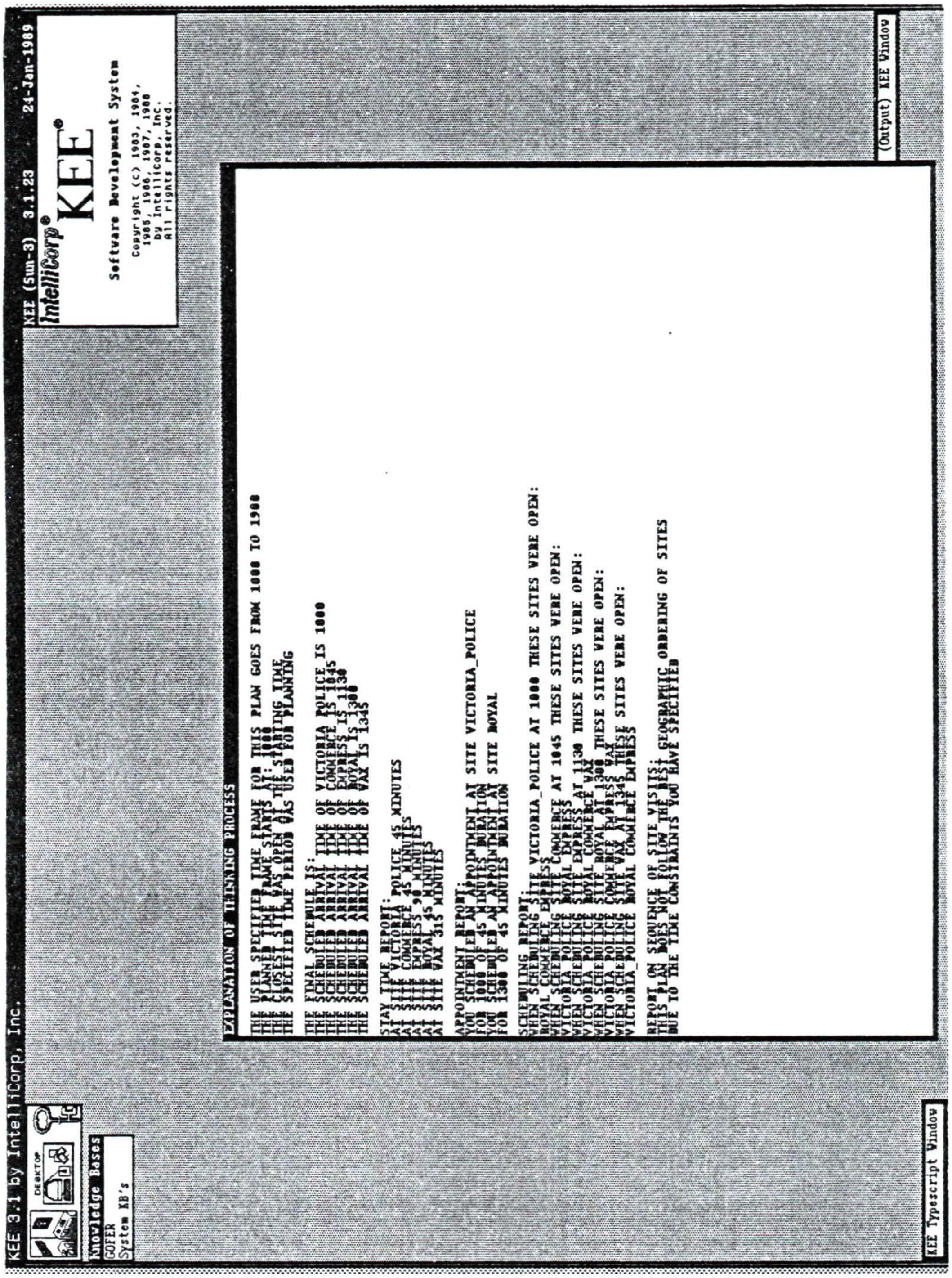

Figure A-32 Protocol 3 Test 3

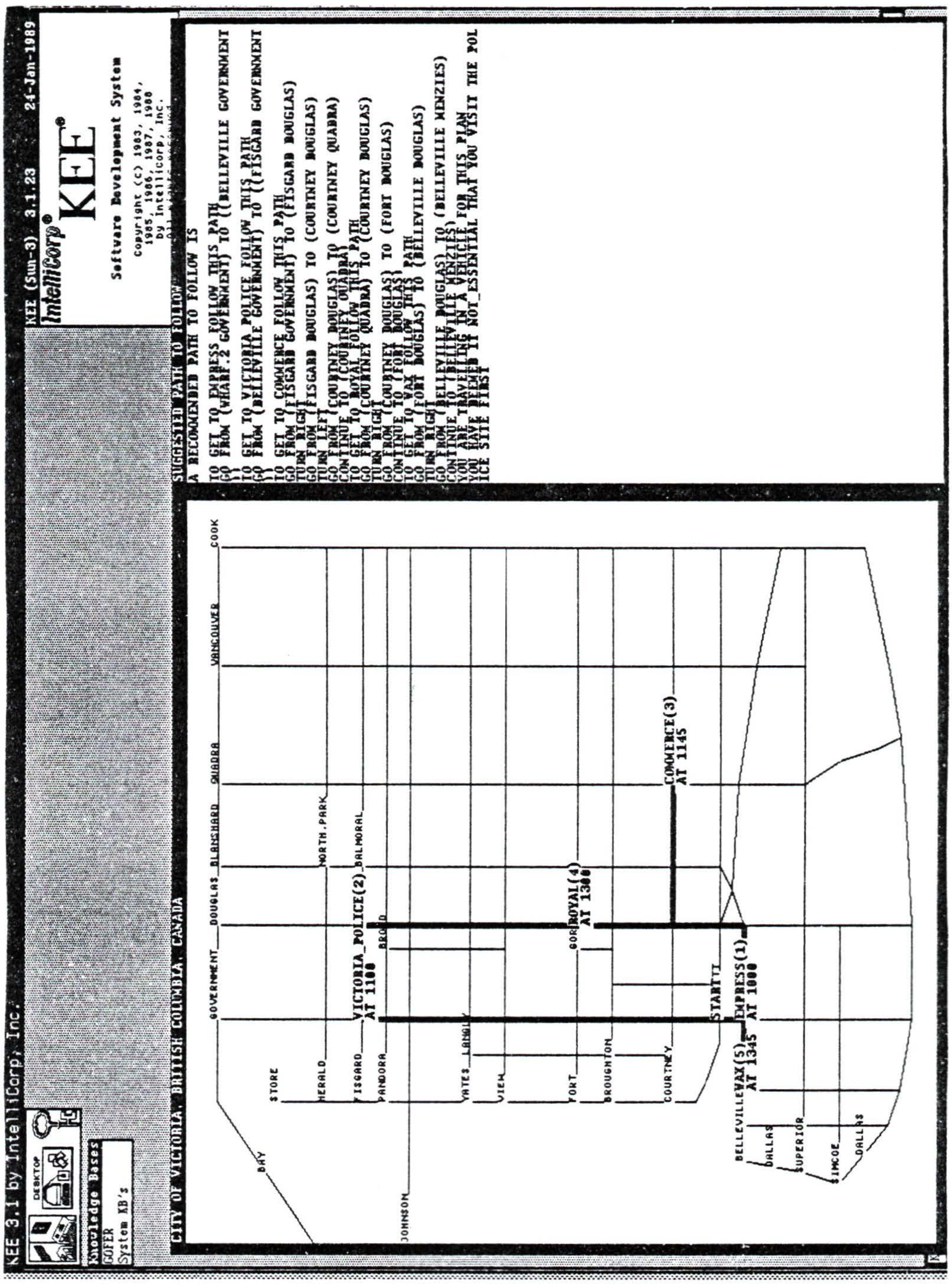

Figure A-33 Protocol 3 Test 4

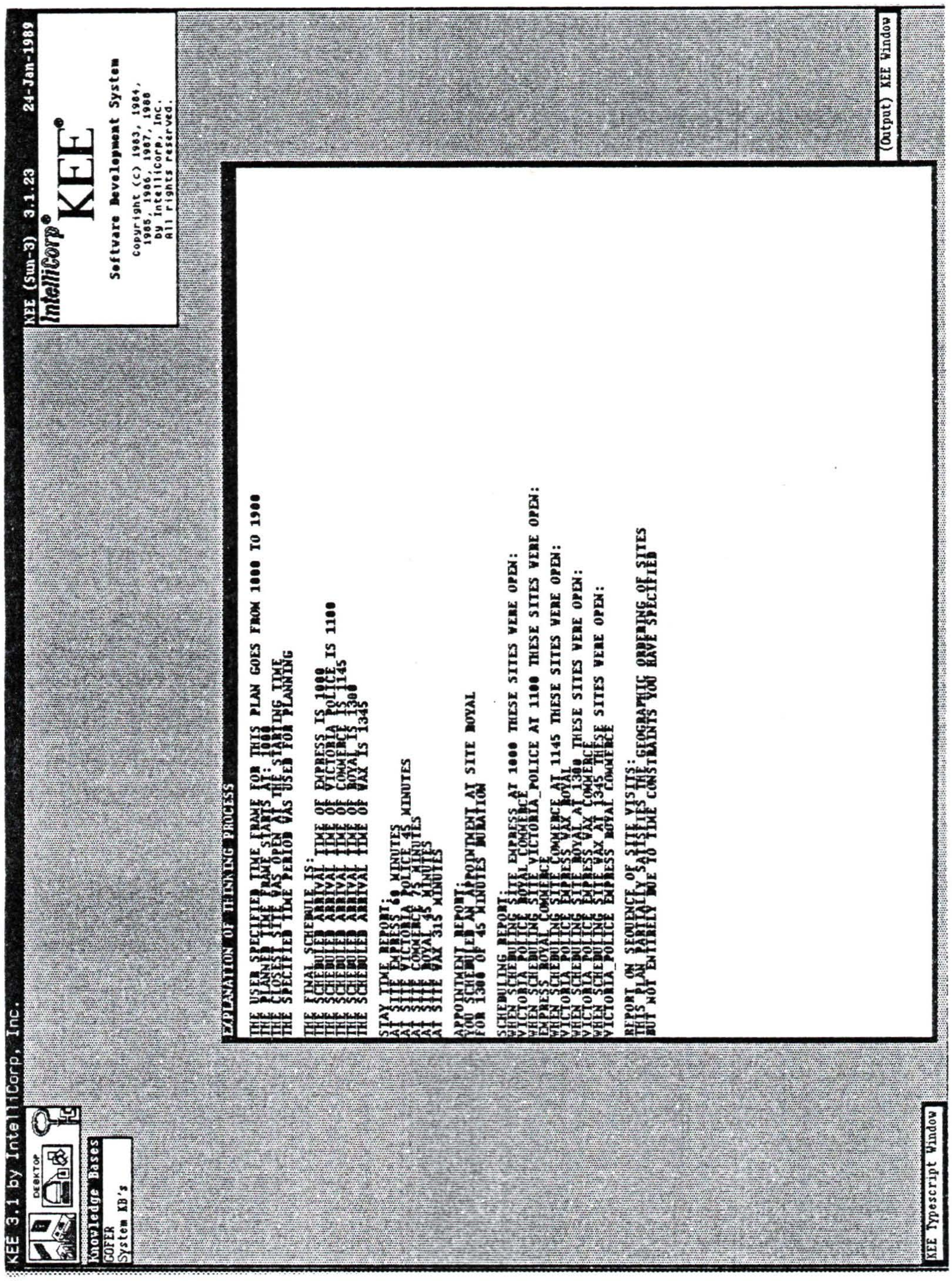

Figure A-34 Protocol 3 Test 4

A-1-4 Protocol 4

Test 1

Time Span: 1000-1800

Sites: Royal Bank, City Hall, Commerce Bank

Mode of Travel: car

Starting Intersection: (Pandora Cook)

Appointments: Royal Bank at 1000 for 60 minutes,

City Hall at 1700 for 60 minutes, Commerce Bank at 1400

for 60 minutes

Processing Time: 0:44

Test 2

Time Span: 1000-1600

Sites: Provincial legislature, Commerce Bank,

Montreal Bank, Harbour Square

Mode of Travel: car

Starting Intersection: (Johnson Cook)

Appointments: Provincial legislature at 1230 for 40 minutes,

Montreal Bank at 1100 for 45 minutes

Processing Time: 2:35

Test 3

Time Span: 1000-1500

Sites: Royal Bank, Wax Museum, Empress Hotel, City Hall,

Eatons

Mode of Travel: car

Starting Intersection: (Belleville Oswego)

Appointments: City Hall at 1100 for 40 minutes,

Royal Bank at 1400 for 30 minutes

Processing Time: 4:42

Test 4

Time Span: 0900-1500

Sites: City Hall, Provincial Legislature,Ports, Commerce

Mode of Travel: car

Starting Intersection: (Pandora Cook)

Appointments: City Hall at 1000 for 120 minutes,

Provincial Legislature at 1300 for 120 minutes

Notes: Ports for lunch

Processing Time: 1:33

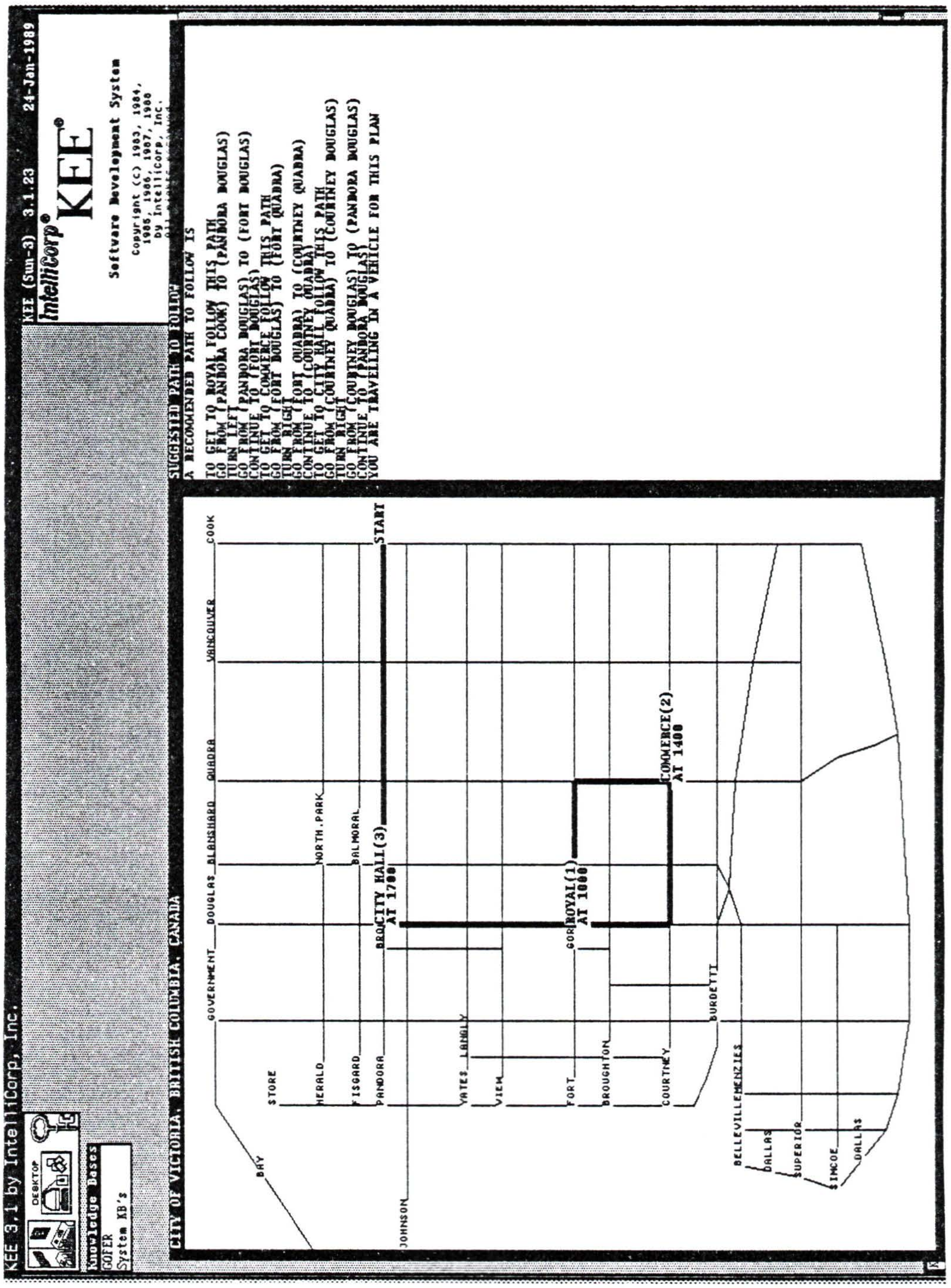

Figure A-35 Protocol 4 Test 1

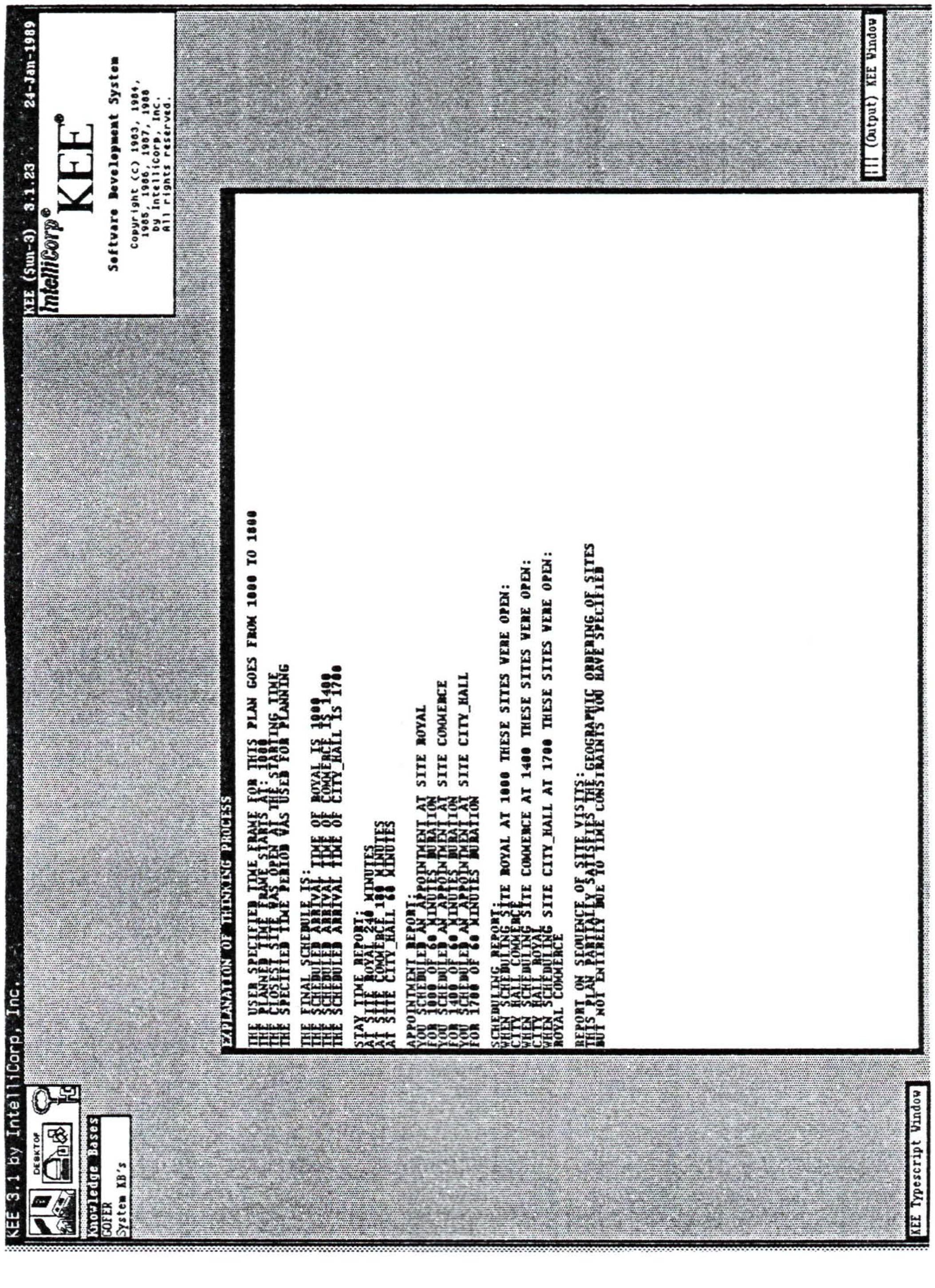

Figure A-36 Protocol 4 Test 1

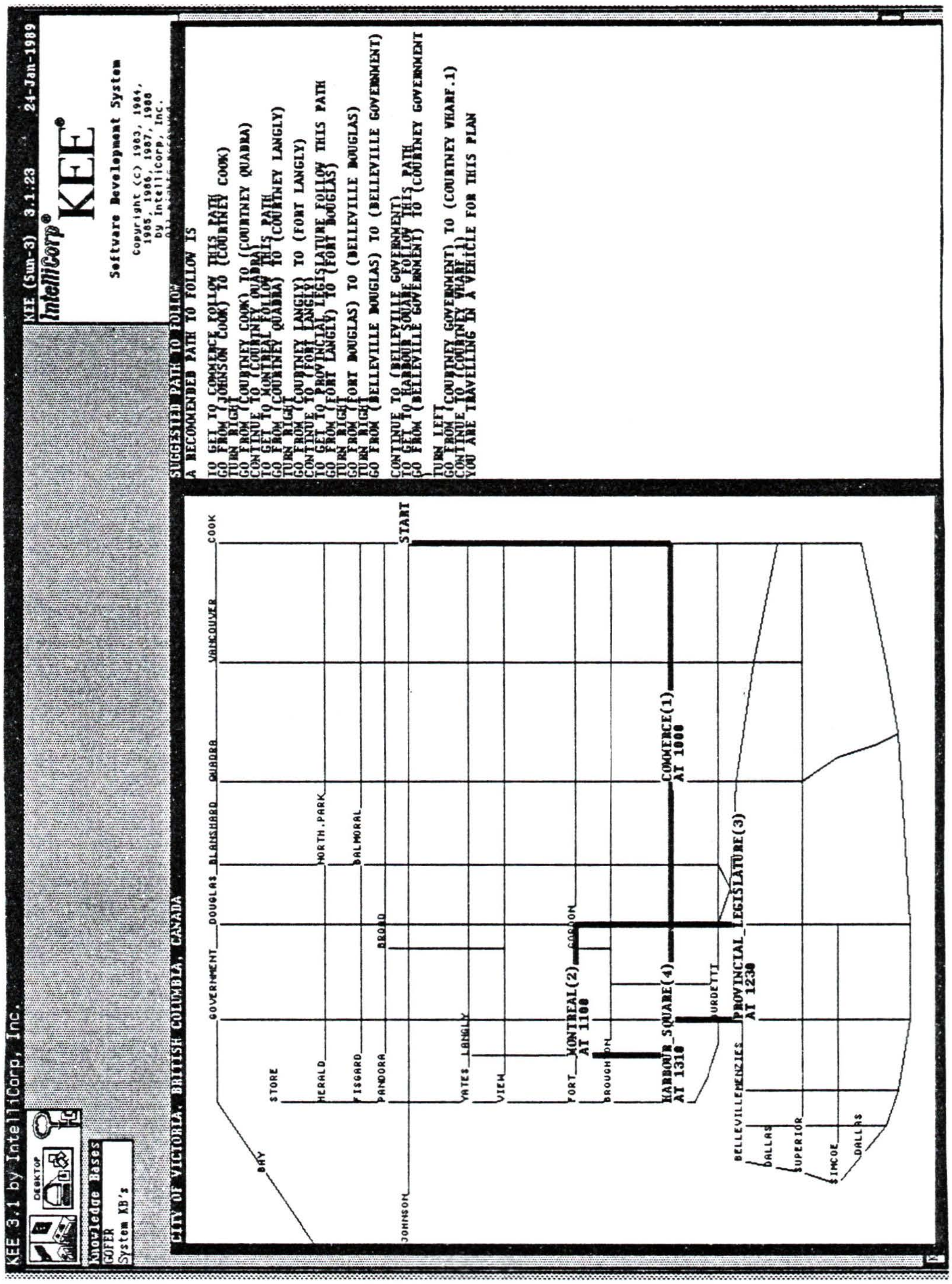

Figure A-37 Protocol 4 Test 2

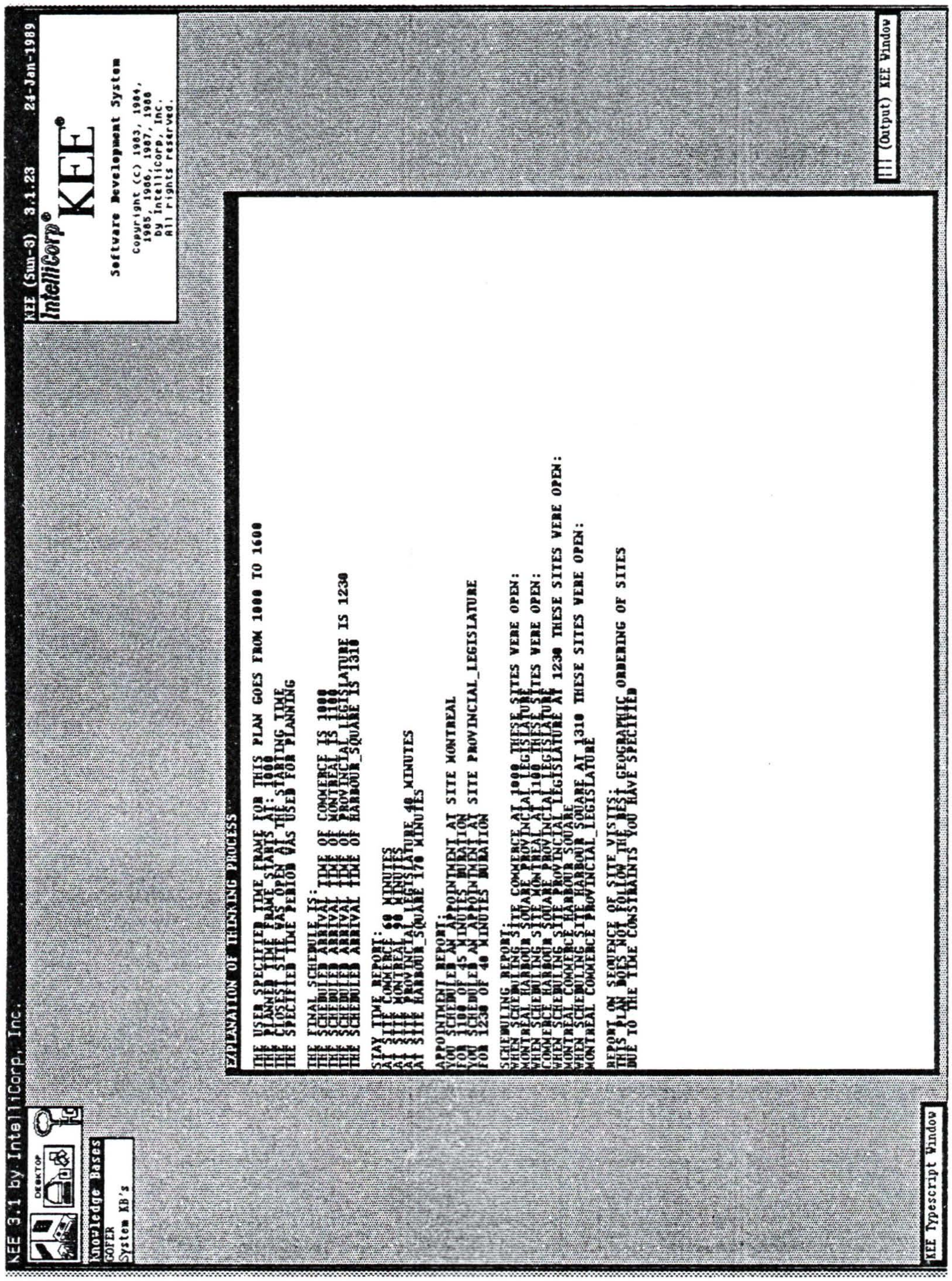

Figure A-38 Protocol 4 Test 2

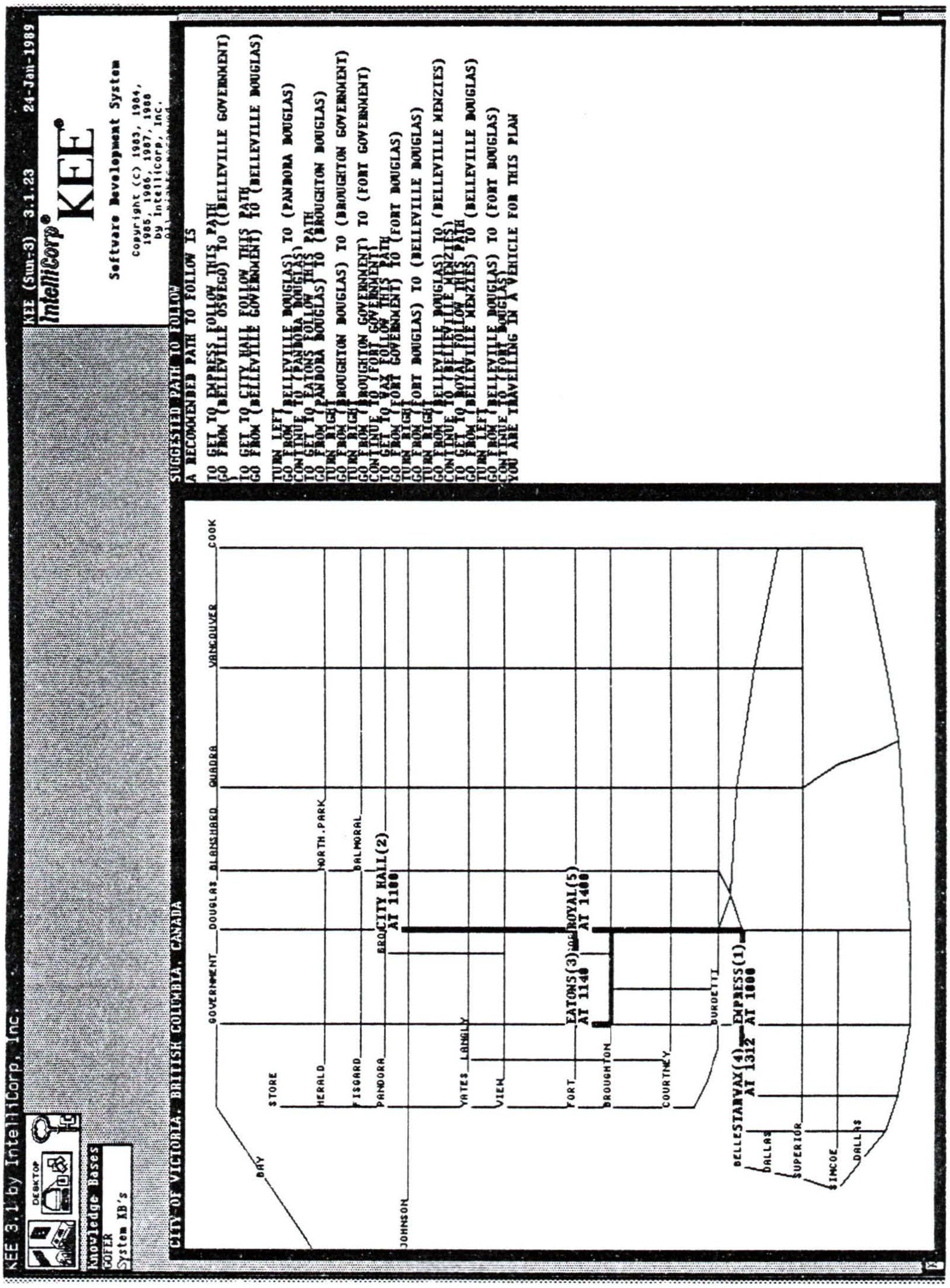

Figure A-39 Protocol 4 Test 3

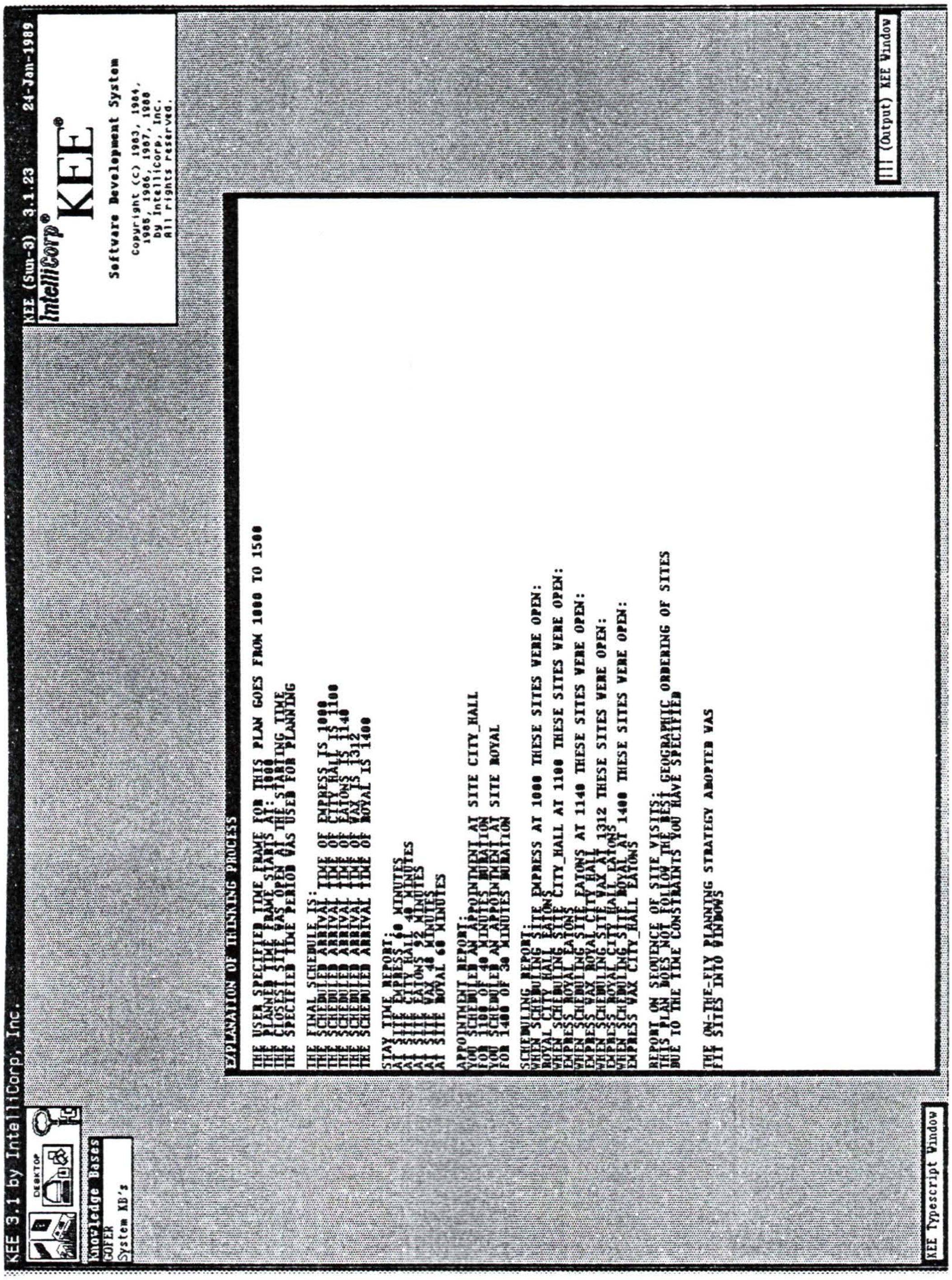

Figure A-40 Protocol 4 Test 3

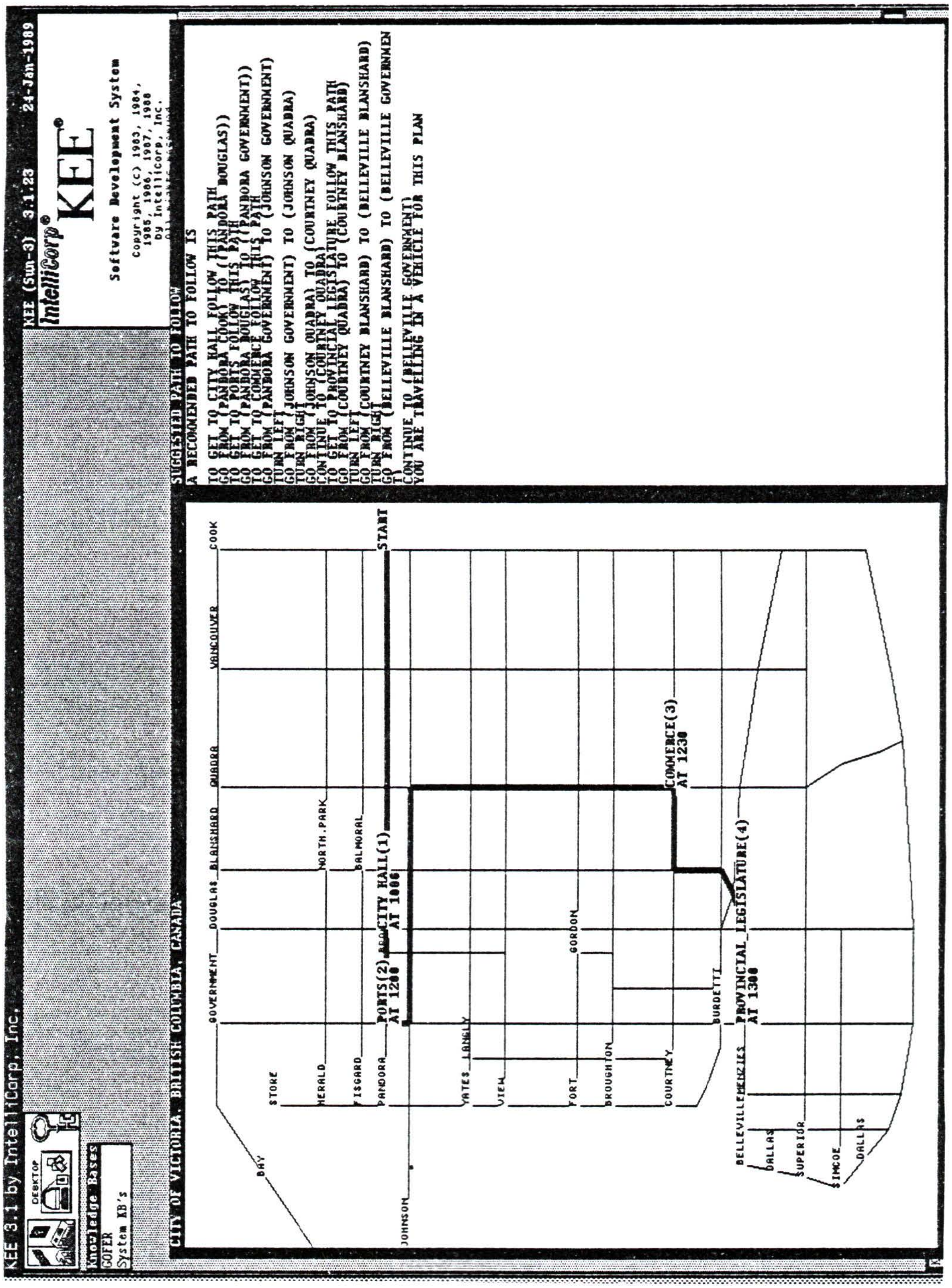

Figure A-41 Protocol 4 Test 4

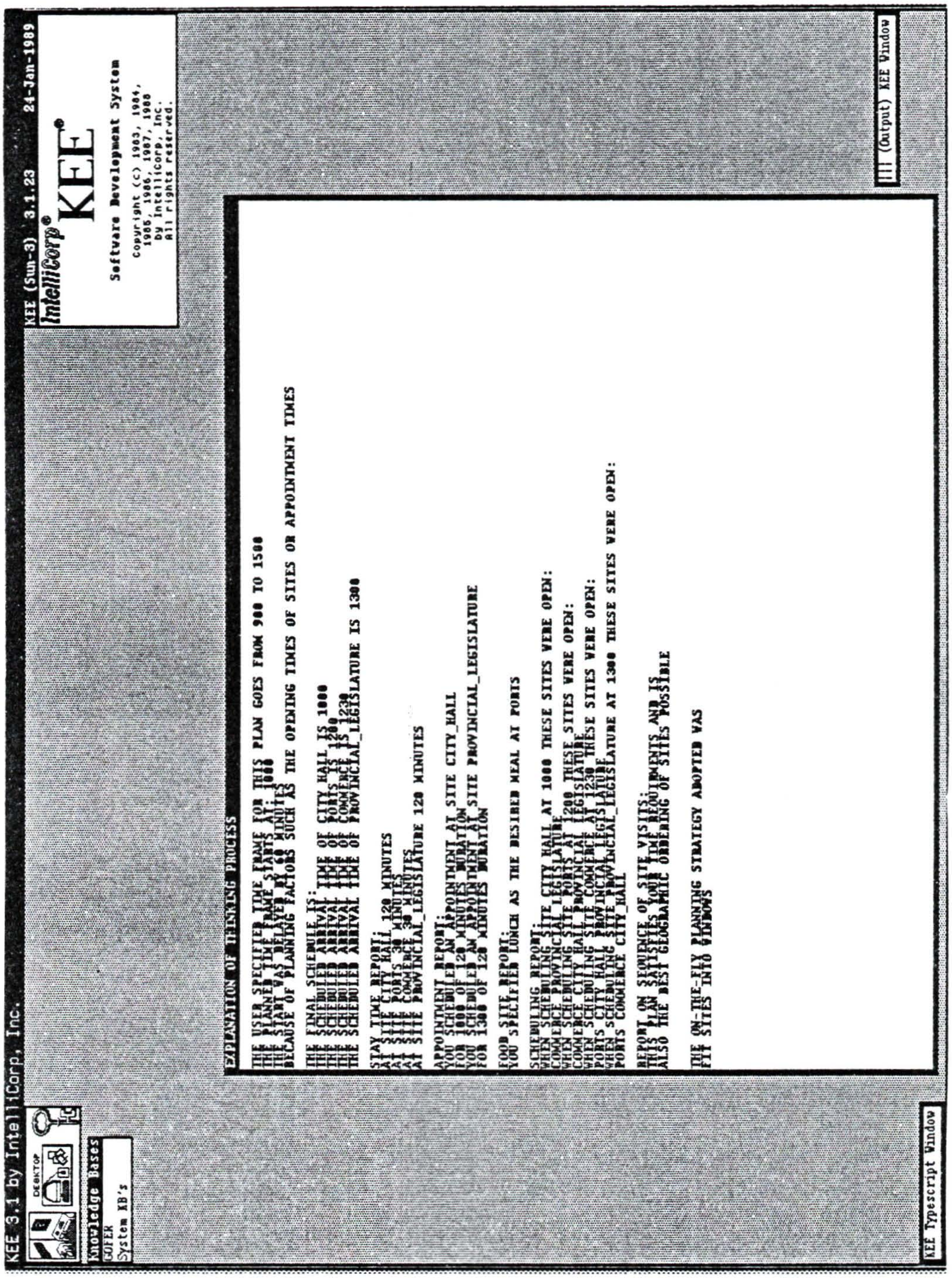

Figure A-42 Protocol 4 Test 4

A-1-5 Protocol 5

Test 1

Time Span: 1000-1900

Sites: Goodies, Ports, Royal Bank

Mode of Travel: foot

Starting Intersection: (Wharf Government)

Appointments:Nil

Notes: Goodies for lunch, Ports for supper

Processing Time: 0:55

Test 2

Time Span: 1100-1800

Sites: Goodies, Ports, Eatons, Harbour Square

Mode of Travel: foot

Starting Intersection: (Belleville Government)

Appointments: Nil

Notes: Goodies for lunch, Ports for supper

Processing Time: 1:46

Test 3

Time Span: 1000-1900

Sites: Royal Bank, Goodies, Rattenbury, Commerce Bank

Mode of Travel: car

Starting Intersection: (Courtney Cook)

Appointments: Royal Bank at 1130 for 60 minutes

Notes: Rattenbury for lunch, Goodies for supper

Processing Time: 1:05

Test 4

Time Span: 1100-1900

Sites: Goodies, Ports, Commerce Bank

Mode of Travel: car

Starting Intersection: (Courtney Douglas)

Appointments: Nil

Notes: Goodies for breakfast, Ports for supper

Processing Time: 1:28

Test 5

Time Span: 1100-1500

Sites: Market Square, Commerce Bank, Wax Museum,

City Hall, Rattenbury

Mode of Travel: car

Starting Intersection: (Johnson Cook)

Appointments: Commerce Bank at 1130 for 60 minutes,

CIty Hall at 1300 for 60 minutes

Notes: Rattenbury for lunch

Processing Time: 3:07

Test 6

Time Span: 0900-2000

Sites: Goodies, Commerce, Ports, Eatons, Harbour Square

Mode of Travel: car

Starting Intersection: (Johnson Cook)

Appointments: Nil

Notes: Goodies for lunch, Ports for supper.

Processing Time: 2:00

Test 7

Time Span: 0900-1900

Sites: Rattenbury, Wax Museum

Mode of Travel: car

Starting Intersection: (Pandora Cook)

Appointments: Nil

Notes: Rattenbury for lunch

Processing Time: 0:41

<u>Test 8</u>

Time Span: 0900-1900

Sites: Commerce Bank, Royal Bank, Montreal Bank, Ports

Mode of Travel: car

Starting Intersection: (Fort Cook)

Appointments: Nil

Notes: Ports for lunch

Processing Time: 0:59

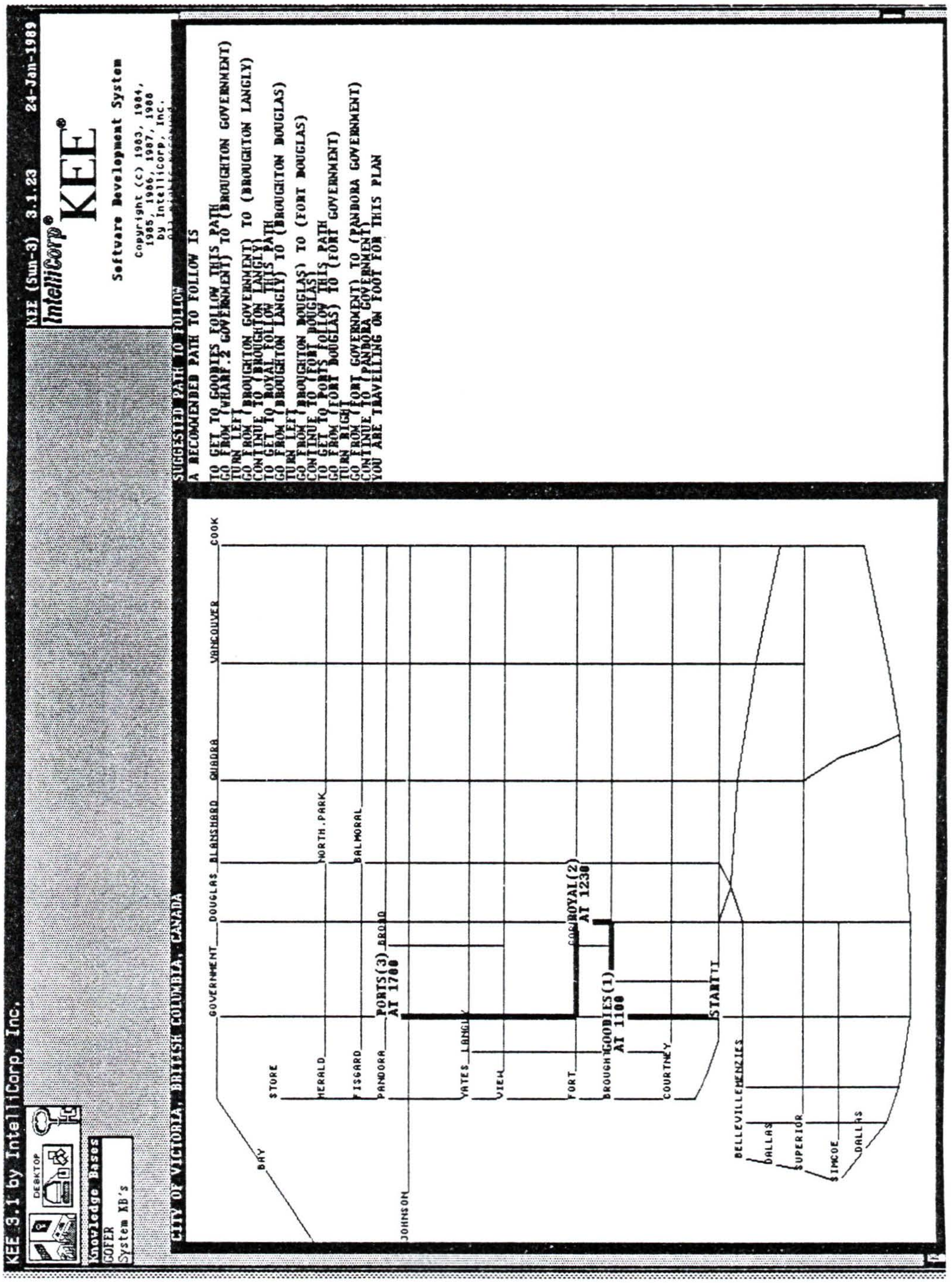

Figure A-43 Protocol 5 Test 1

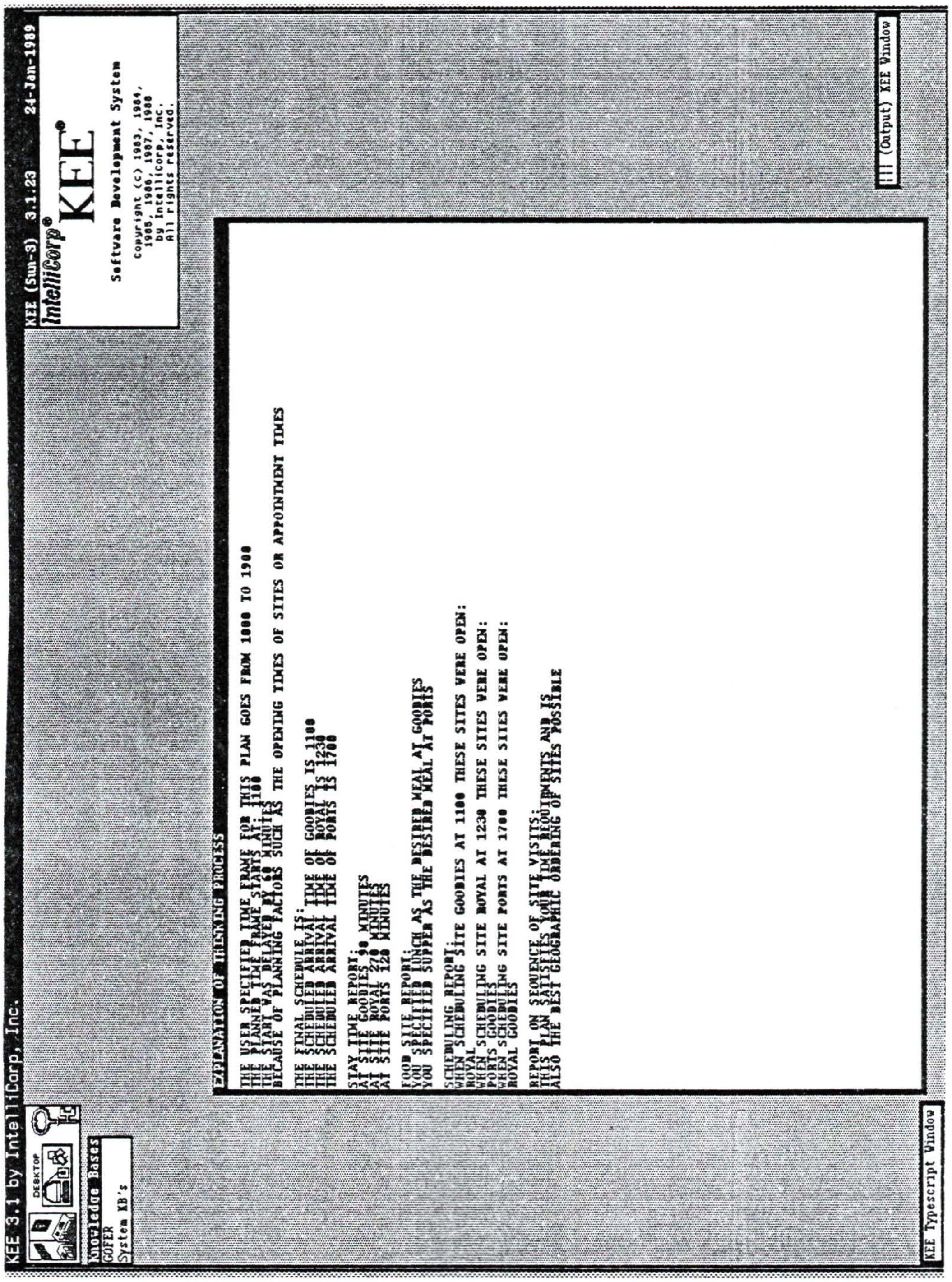

Figure A-44 Protocol 5 Test 1

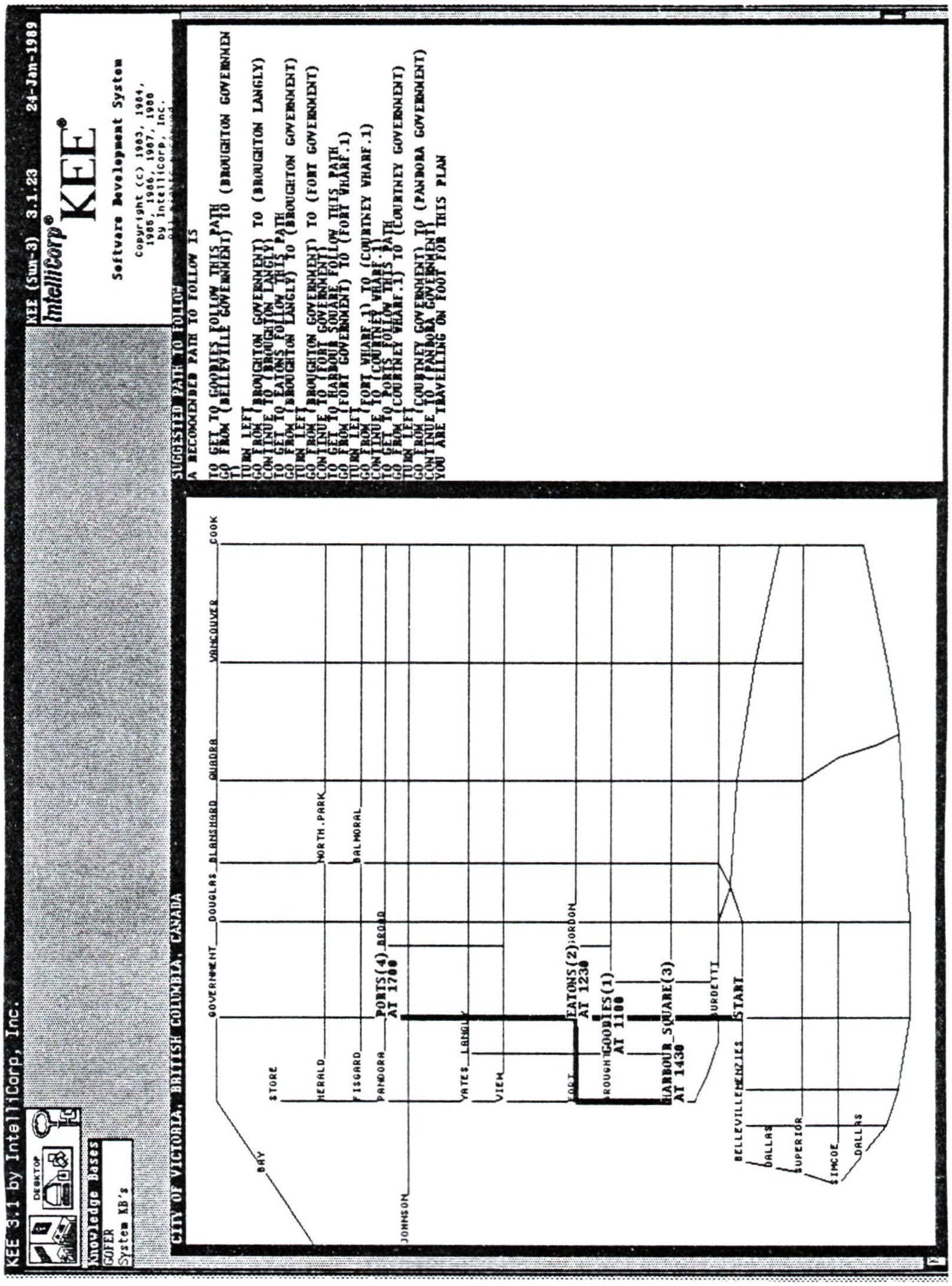

Figure A-45 Protocol 5 Test 2

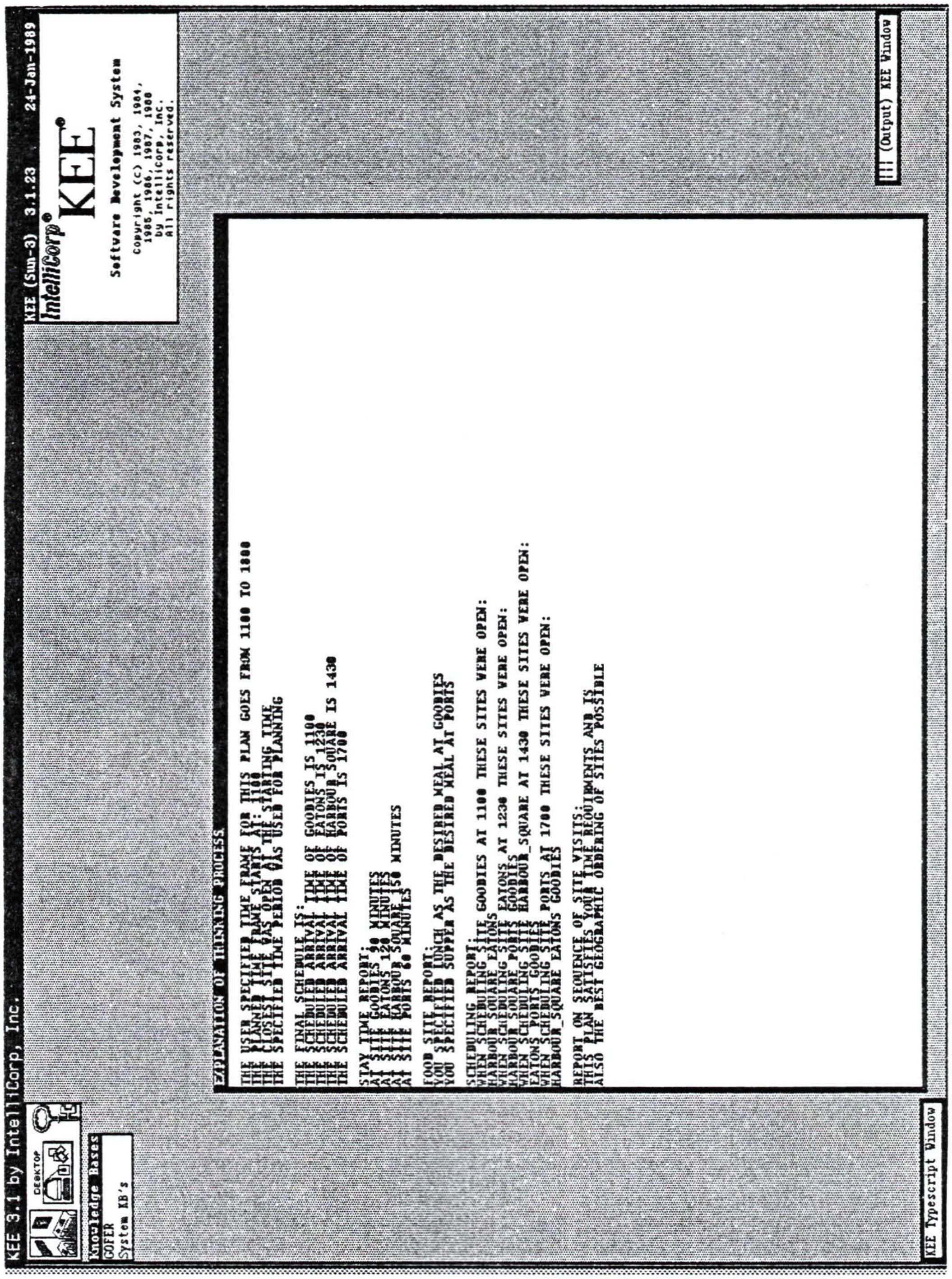

Figure A-46 Protocol 5 Test 2

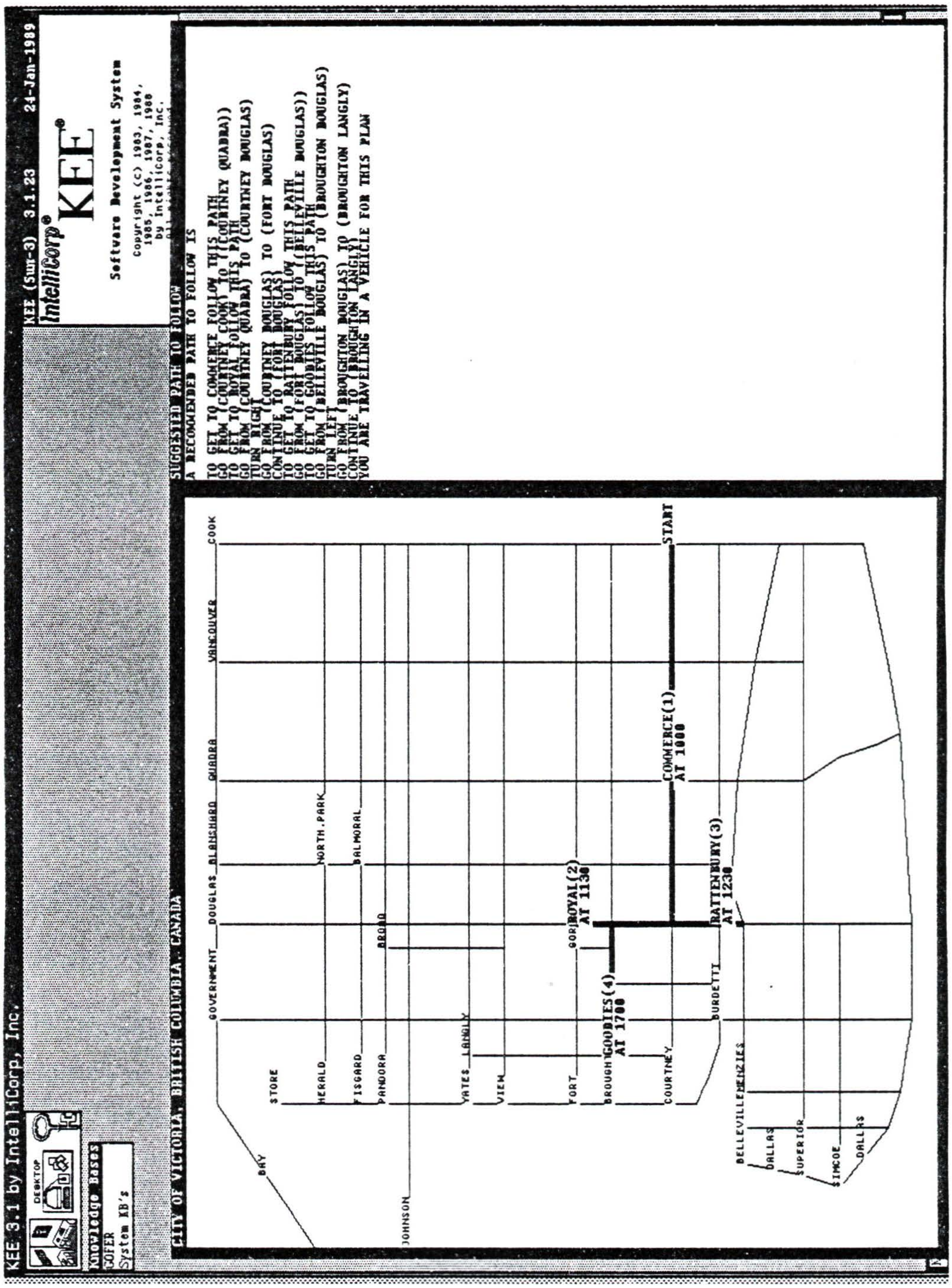

Figure A-47 Protocol 5 Test 3

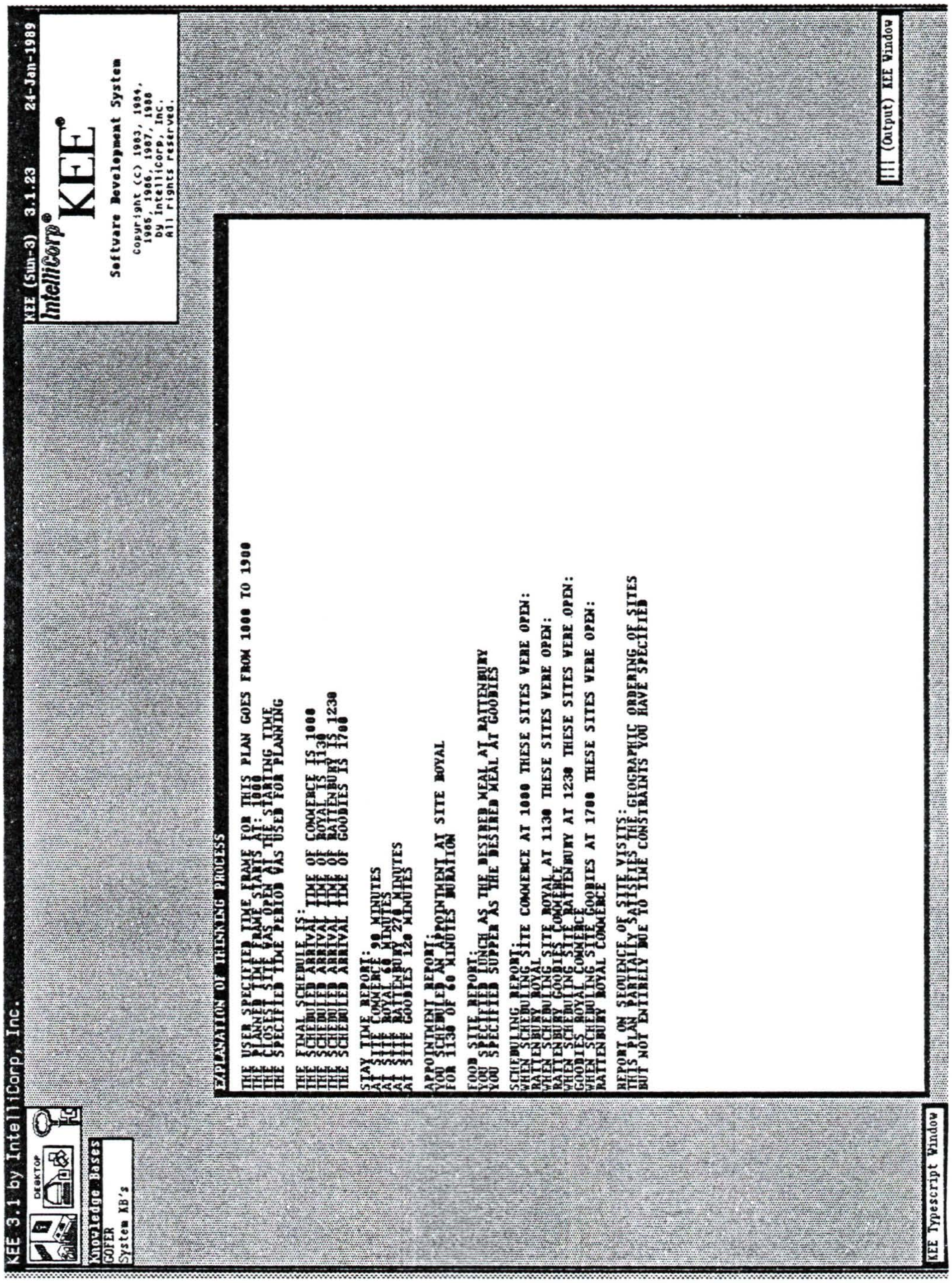

Figure A-48 Protocol 5 Test 3

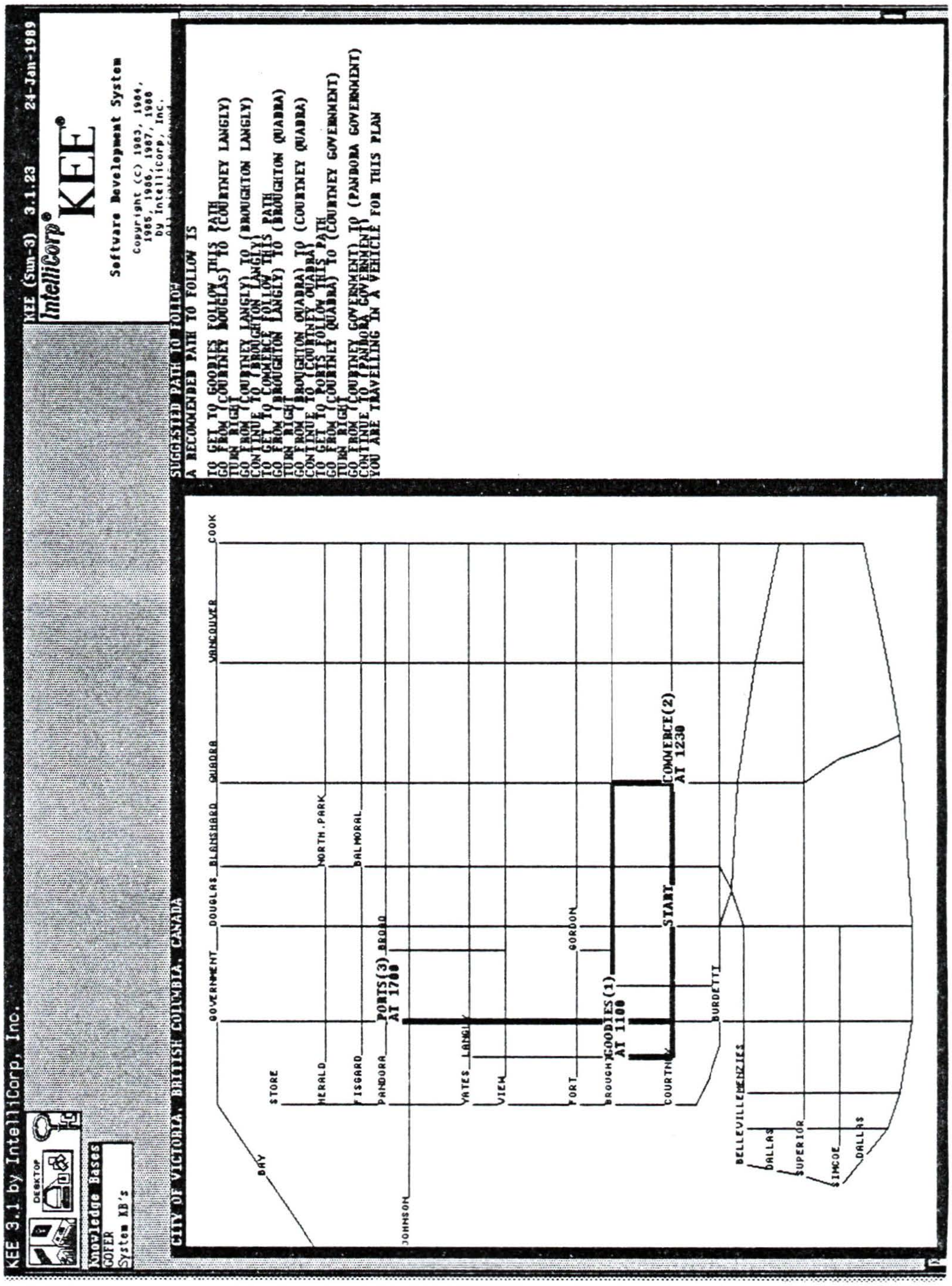

Figure A-49 Protocol 5 Test 4

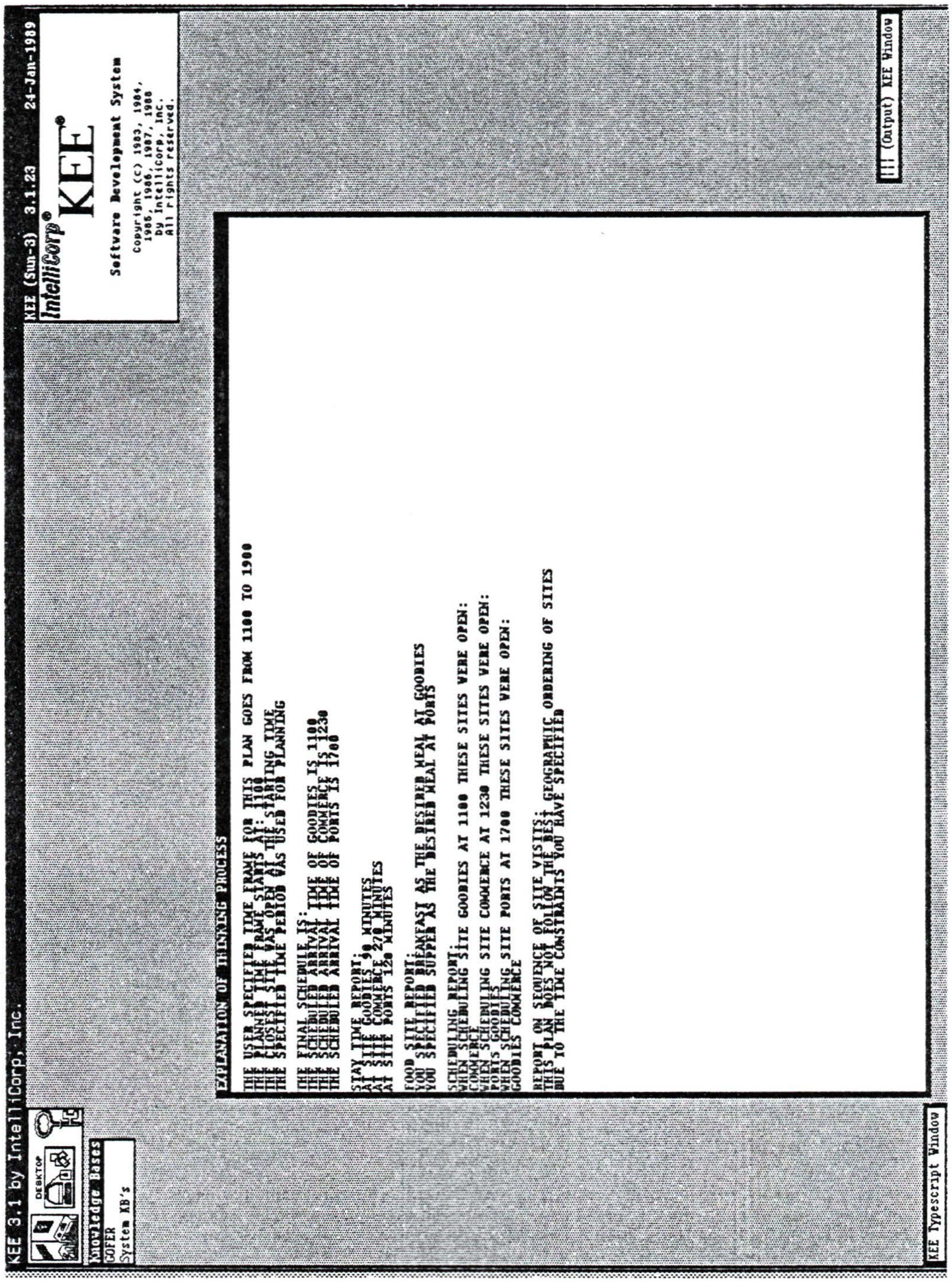

Figure A-50 Protocol 5 Test 4

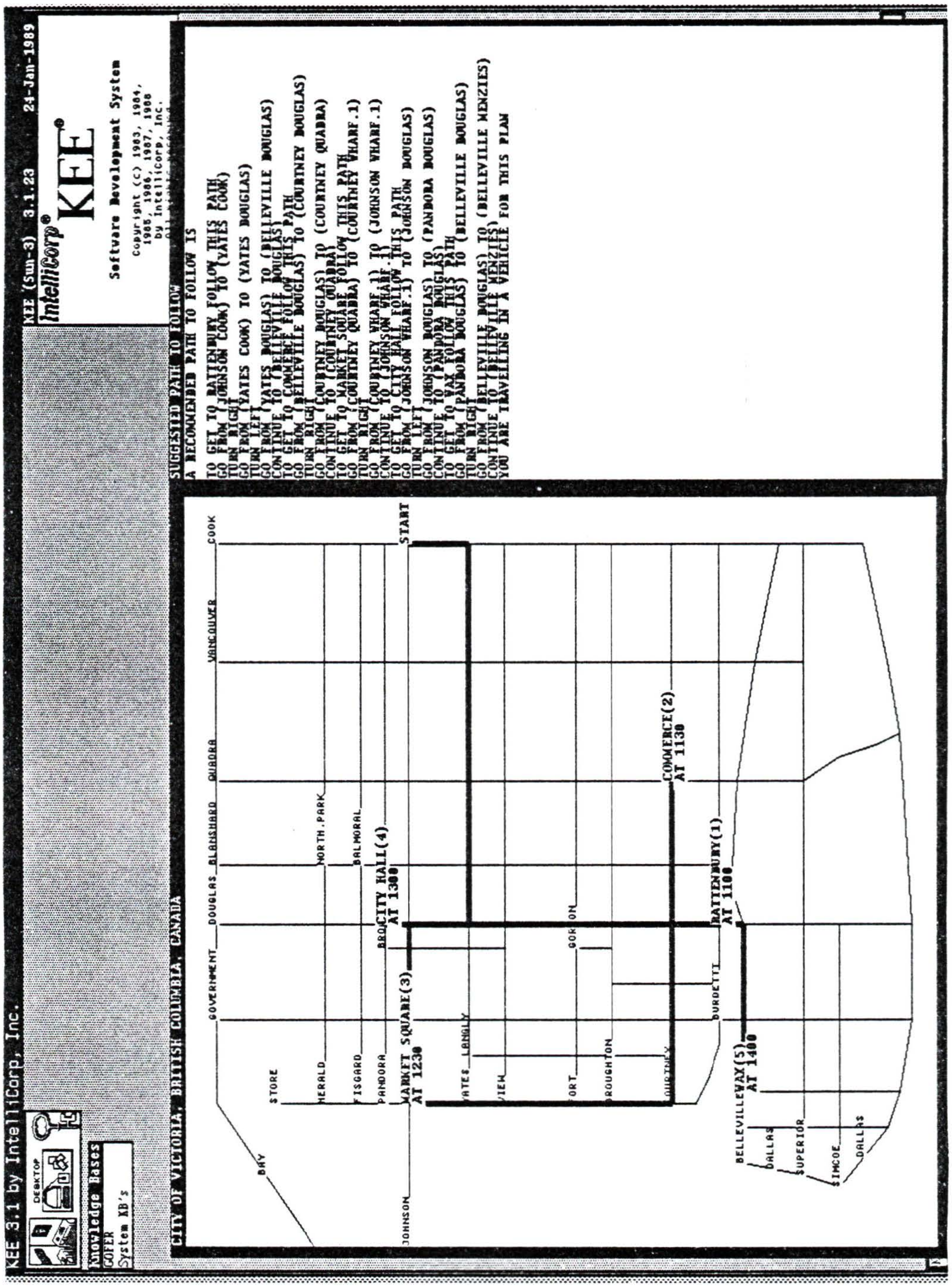

Figure A-51 Protocol 5 Test 5

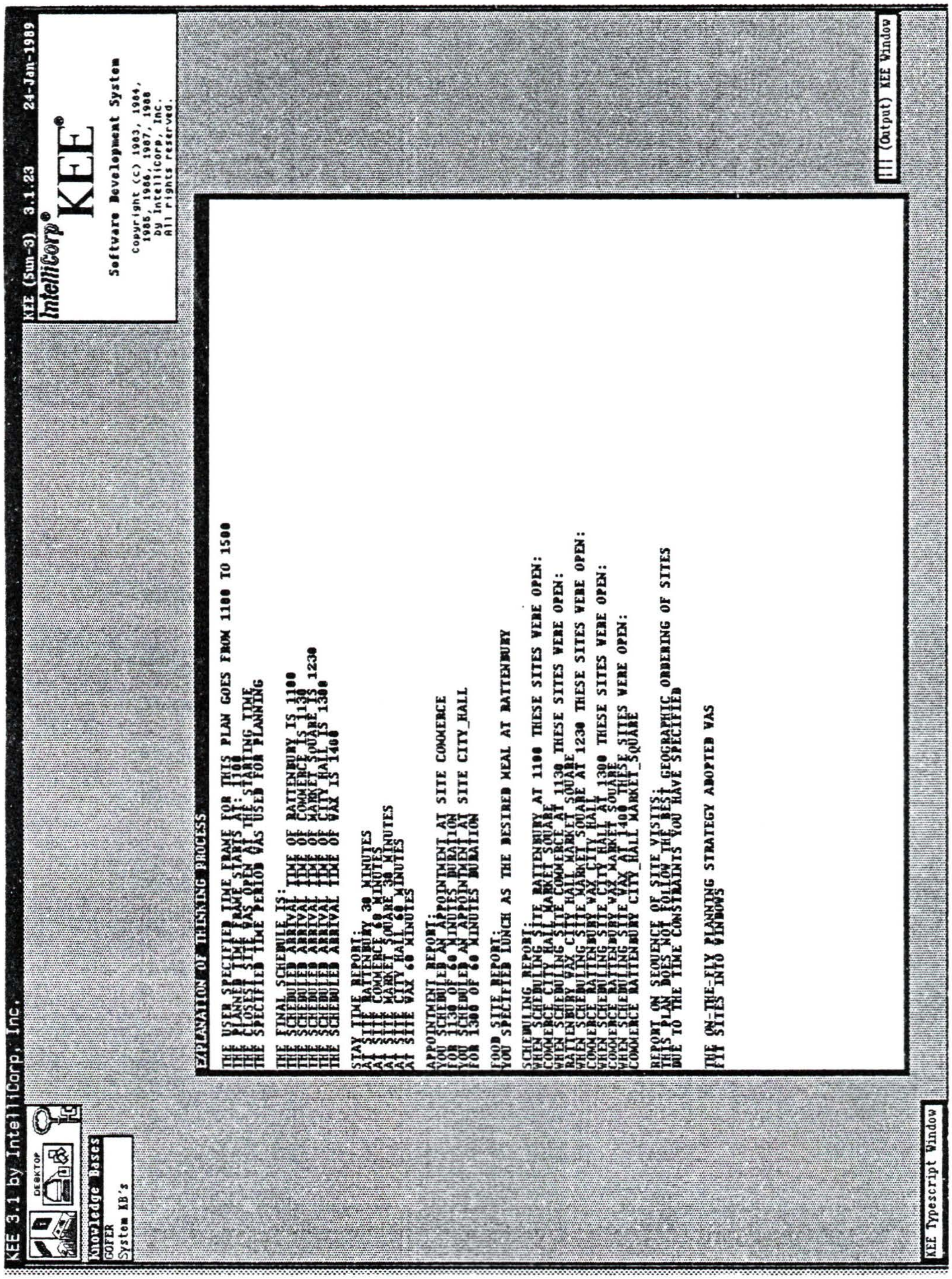

Figure A-52 Protocol 5 Test 5

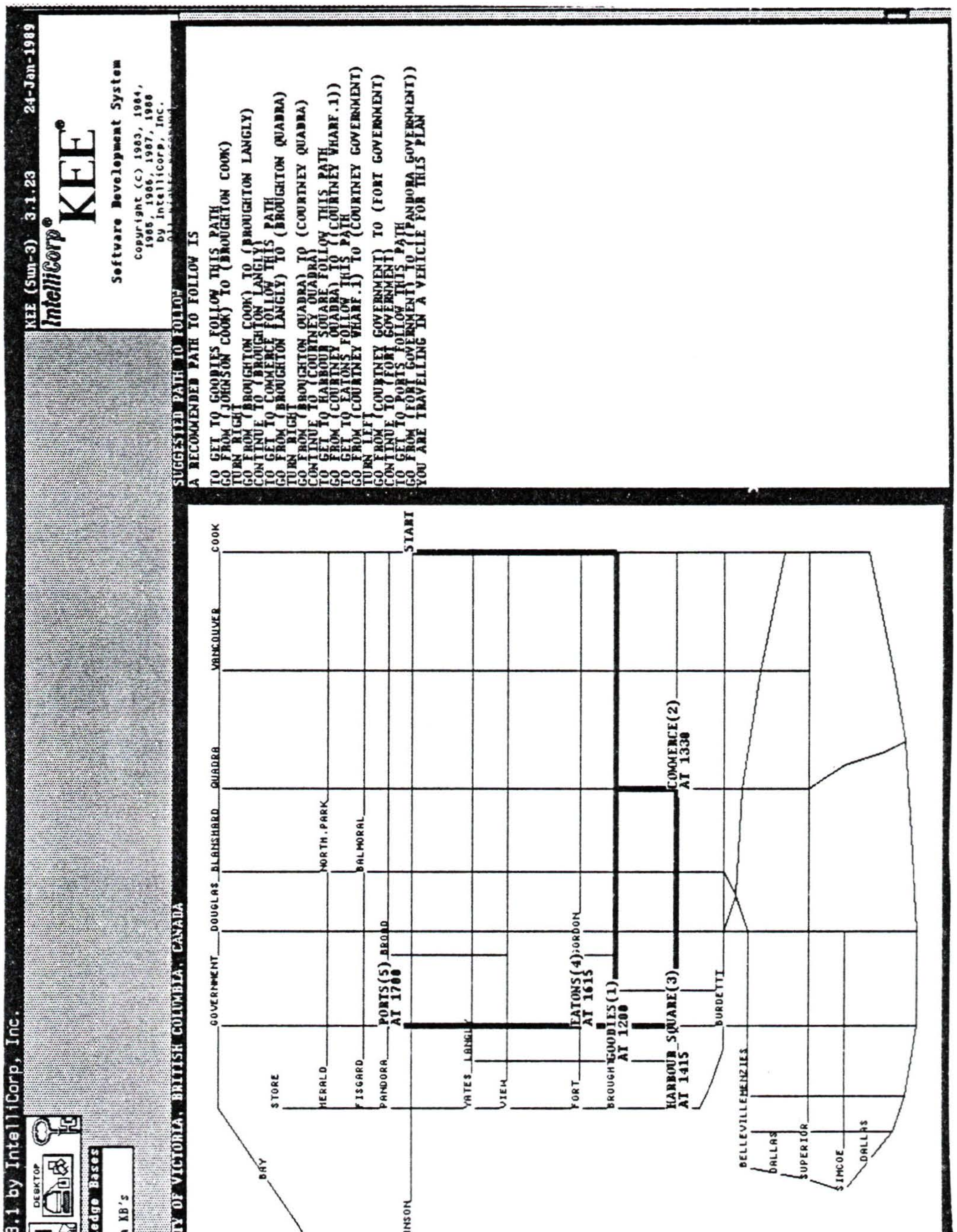

Figure A-53 Protocol 5 Test 6

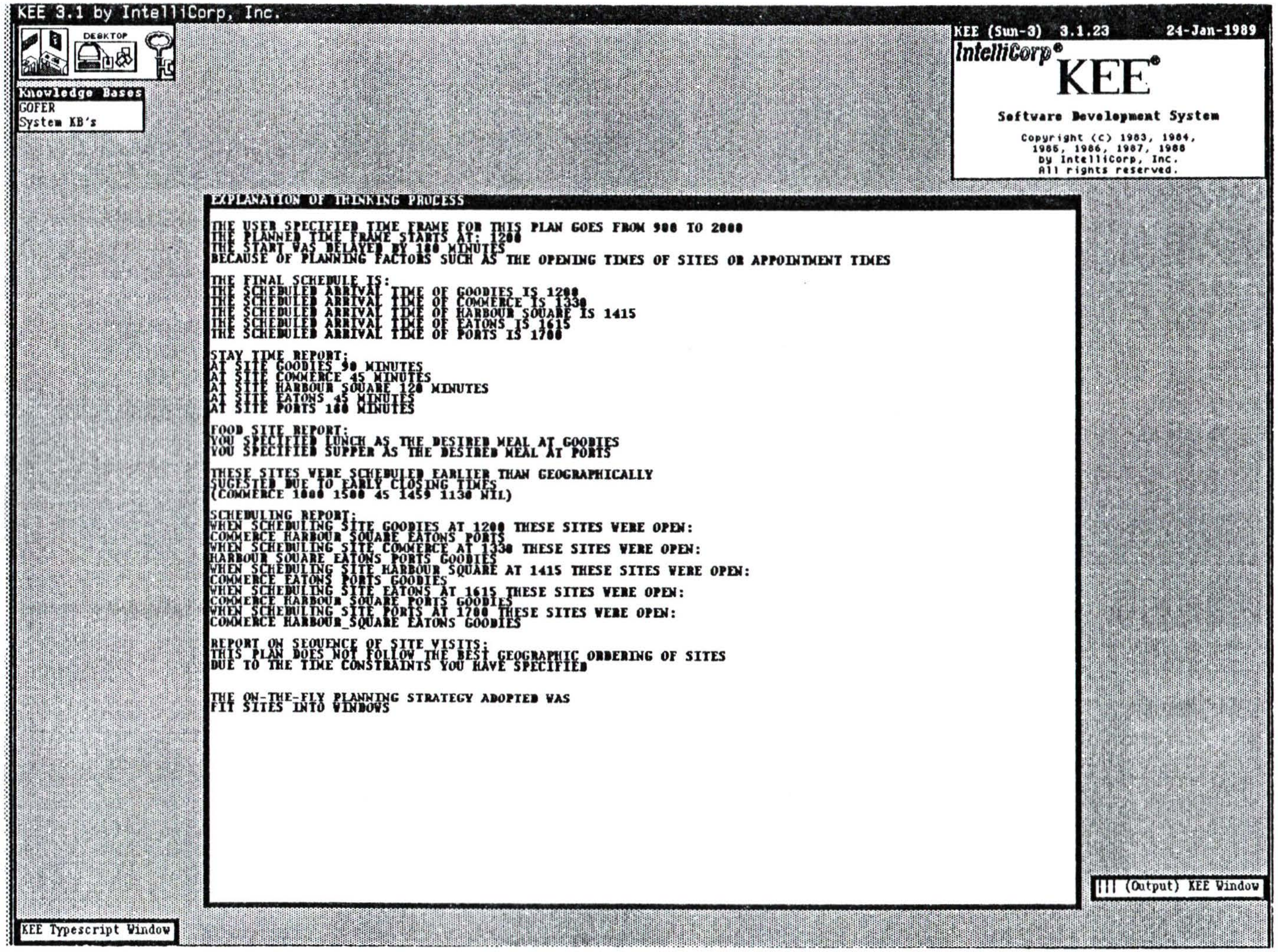

Figure A-54 Protocol 5 Test 6

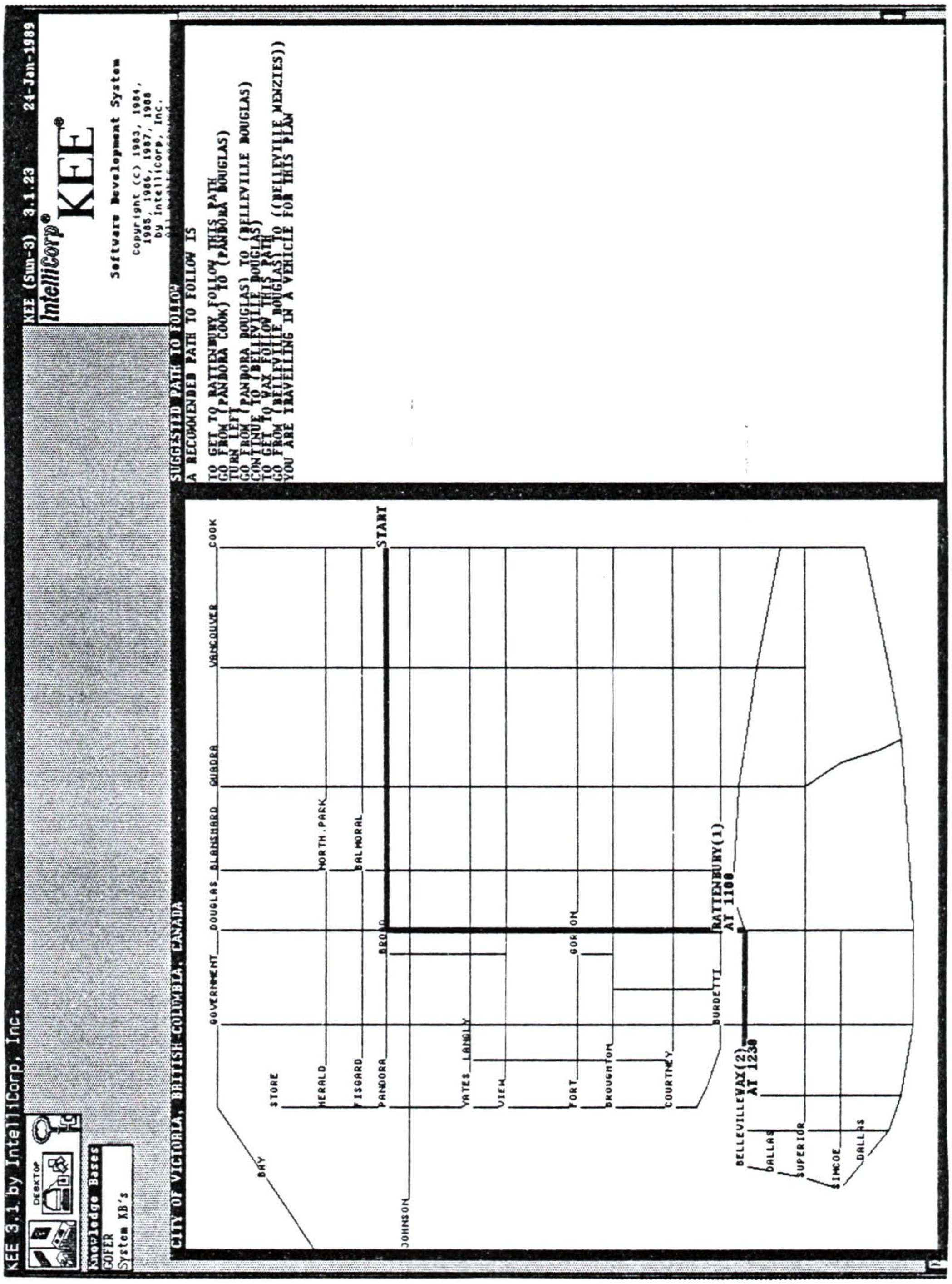

Figure A-55 Protocol 5 Test 7

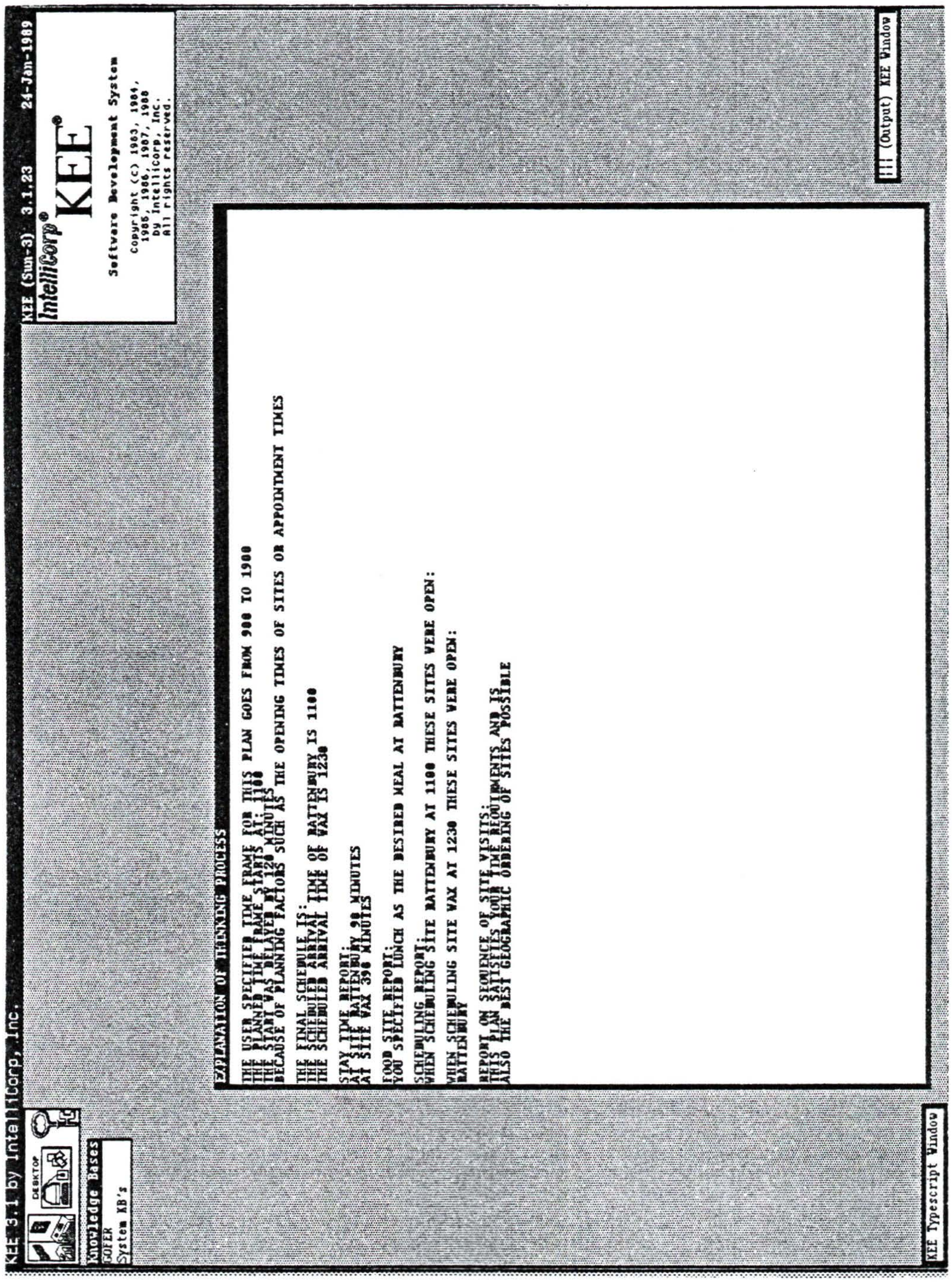

Figure A-56 Protocol 5 Test 7

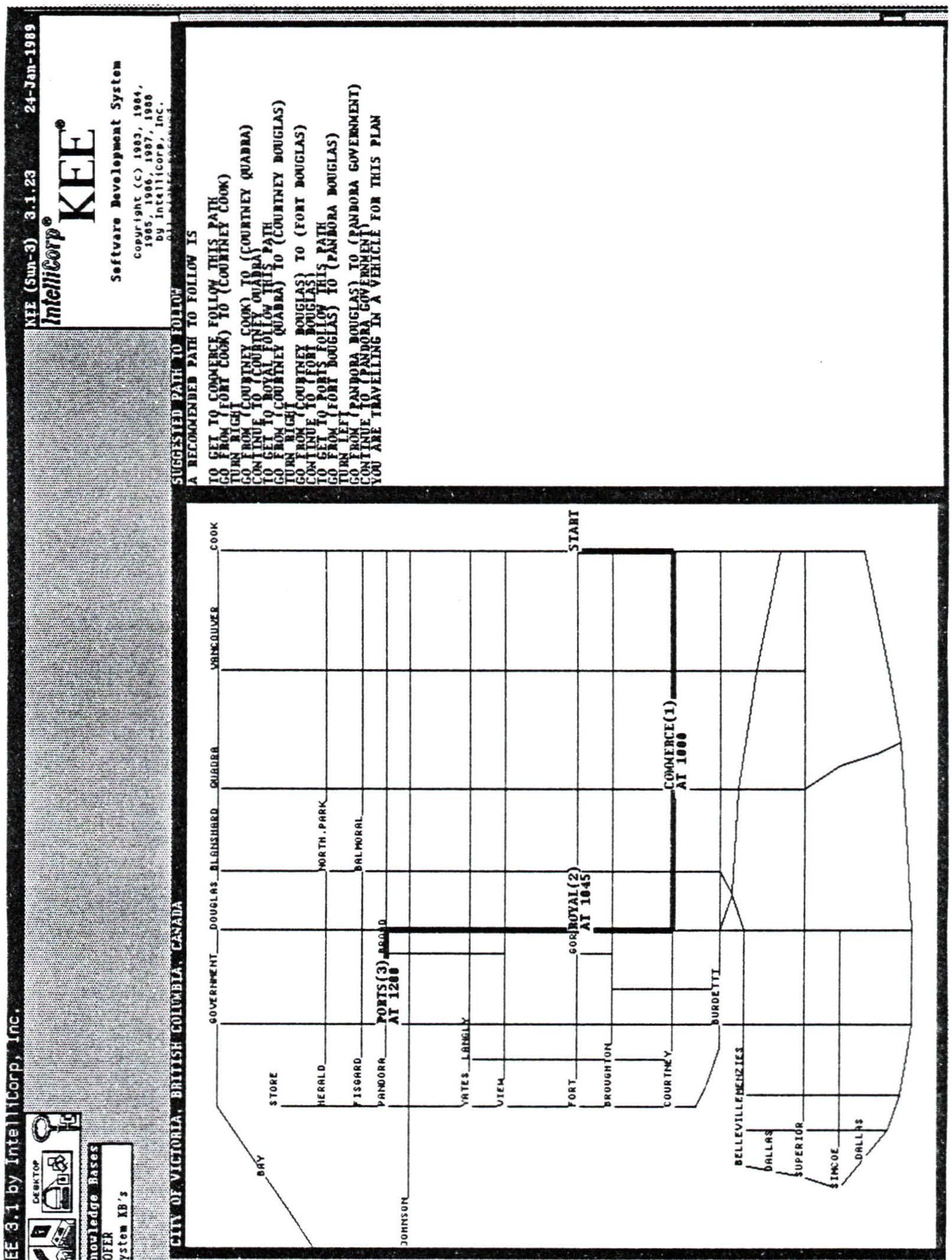

Figure A-57 Protocol 5 Test 8

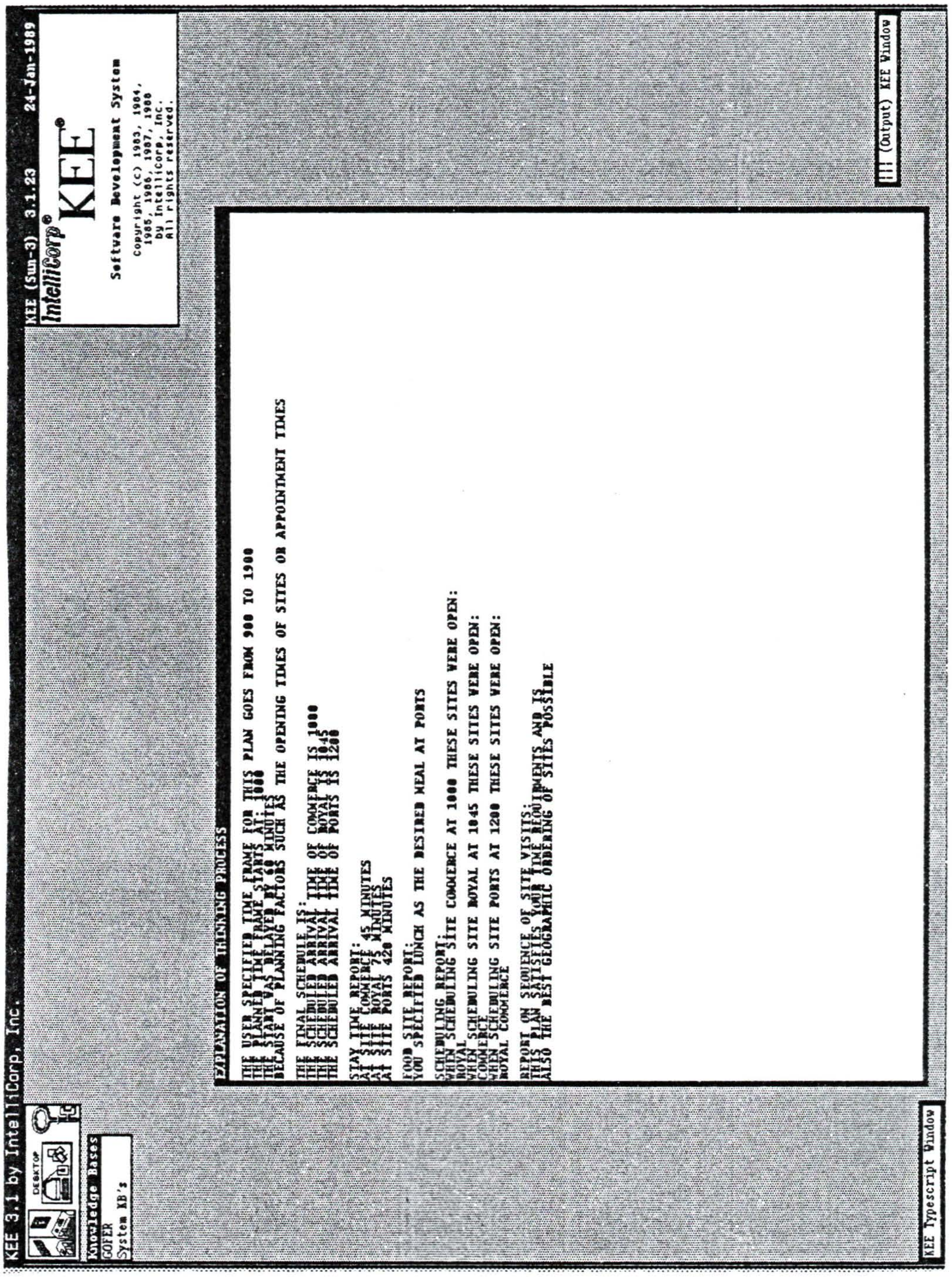

Figure A-58 Protocol 5 Test 8

A-1-6 Protocol 6

This protocol consists of tests which attempt to generate problems requiring the 'on-the-fly specialist to complete planning. The comments in the plan assessment window include a description of the 'on-the-fly' strategy adopted to solve a problem.

Test 1

Time Span: 1100-1500

Sites: Commerce Bank, Royal Bank, Wax Museum, Empress Hotel

Mode of Travel: car

Starting Intersection: (Wharf Government)

Appointments: Royal Bank at 1300 for 60 minutes

Processing Time: 2:43

<u>Test 2</u>

Time Span: 1100-1500

Sites: Commerce Bank, Royal Bank, Wax Museum, Empress Hotel

Mode of Travel: car

Starting Intersection: (Wharf Government)

Appointments: Royal Bank at 1130 for 45 minutes, Commerce Bank at 1330 for 30 minutes

Processing Time: 1:52

<u>Test 3</u>

Time Span: 1000-1500

Sites: Goodies, Eatons, Harbour Square, Empress Hotel,

Montreal Bank

Mode of Travel: car

Starting Intersection: (Fort Quadra)

Appointments: Montreal Bank at 1100 for 45 minutes

Notes: Goodies for lunch

Processing Time: 2:10

Test 4

Time Span: 1100-1500

Sites: Goodies, Eatons, Harbour Square, Empress Hotel,

Montreal Bank

Mode of Travel: car

Starting Intersection: (Fort Quadra)

Appointments: Montreal Bank at 1100 for 45 minutes

Notes: Goodies for lunch

Processing Time: 1:48

Test 5

Time Span: 1100-1500

Sites: Goodies, Eatons, Harbour Square, Empress Hotel,

Montreal Bank

Mode of Travel: car

Starting Intersection: (Fort Quadra)

Appointments: Montreal Bank at 1200 for 45 minutes

Notes: Goodies for lunch

Processing Time: 2:52

Test 6

Time Span: 1100-1430

Sites: Goodies, Eatons, Harbour Square, Empress Hotel,

Montreal Bank

Mode of Travel: car

Starting Intersection: (Fort Quadra)

Appointments: Montreal Bank at 1100 for 45 minutes

Notes: Goodies for lunch

Processing Time: 2:02

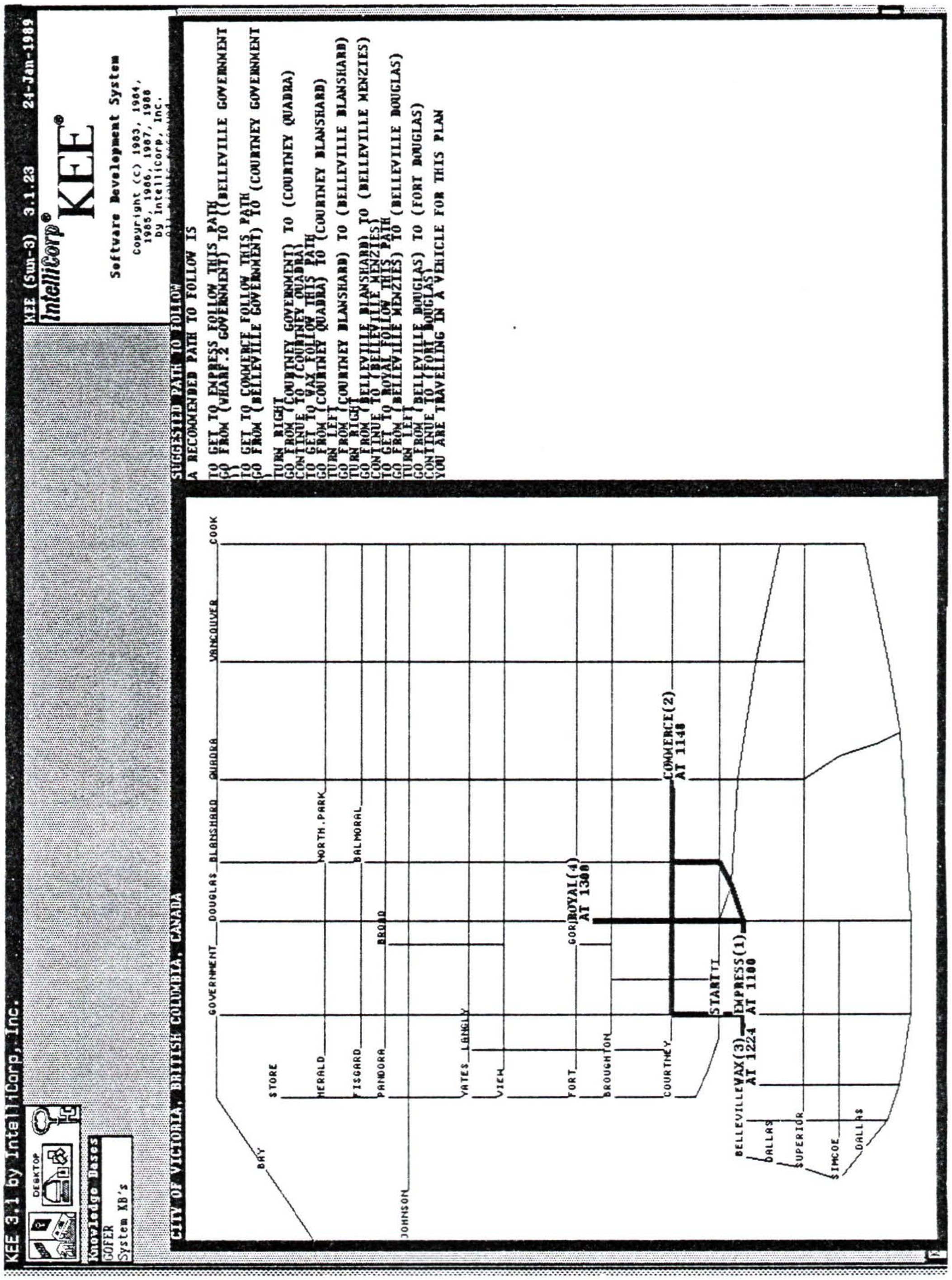

Figure A-59 Protocol 6 Test 1

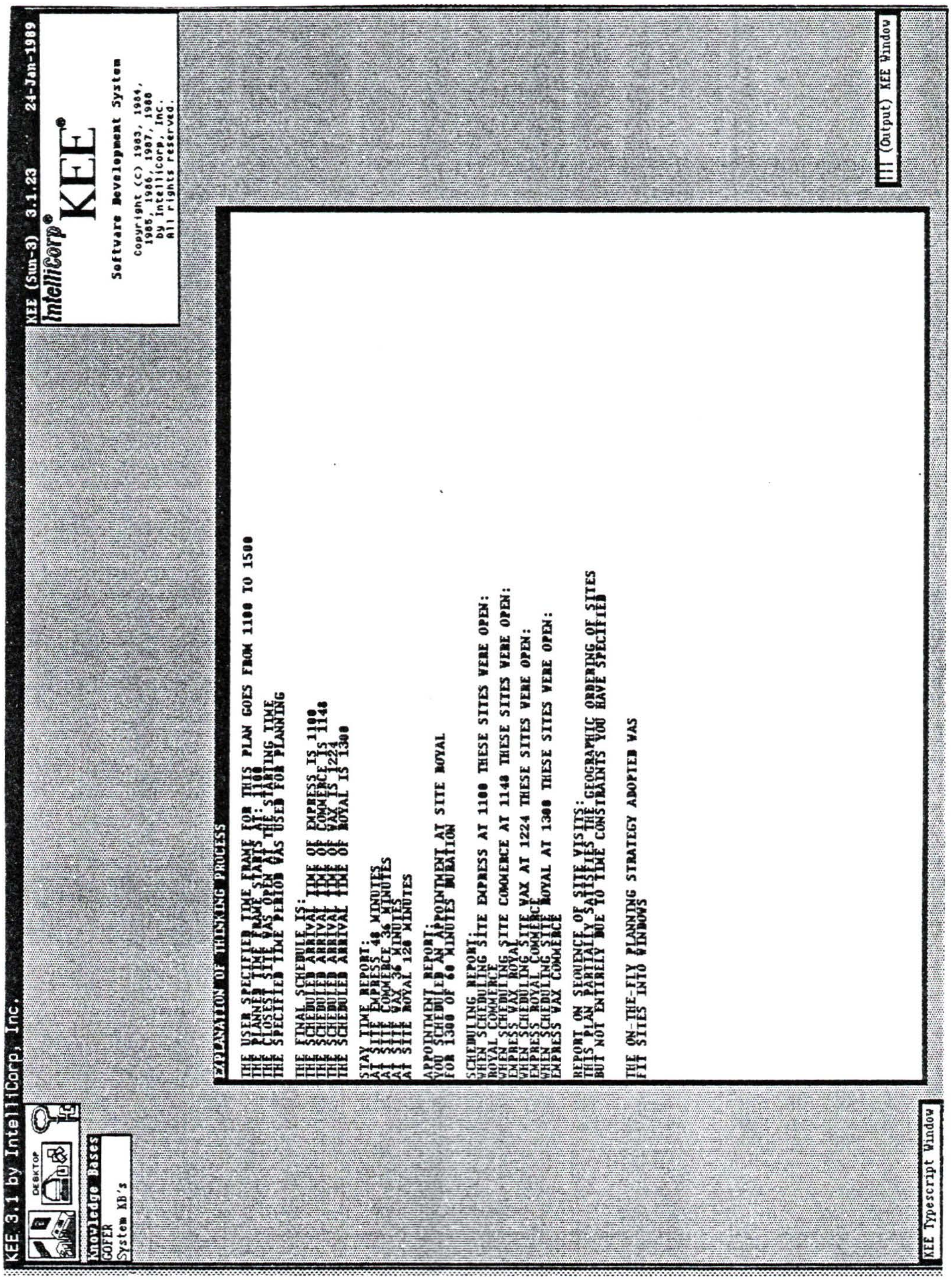

Figure A-60 Protocol 6 Test 1

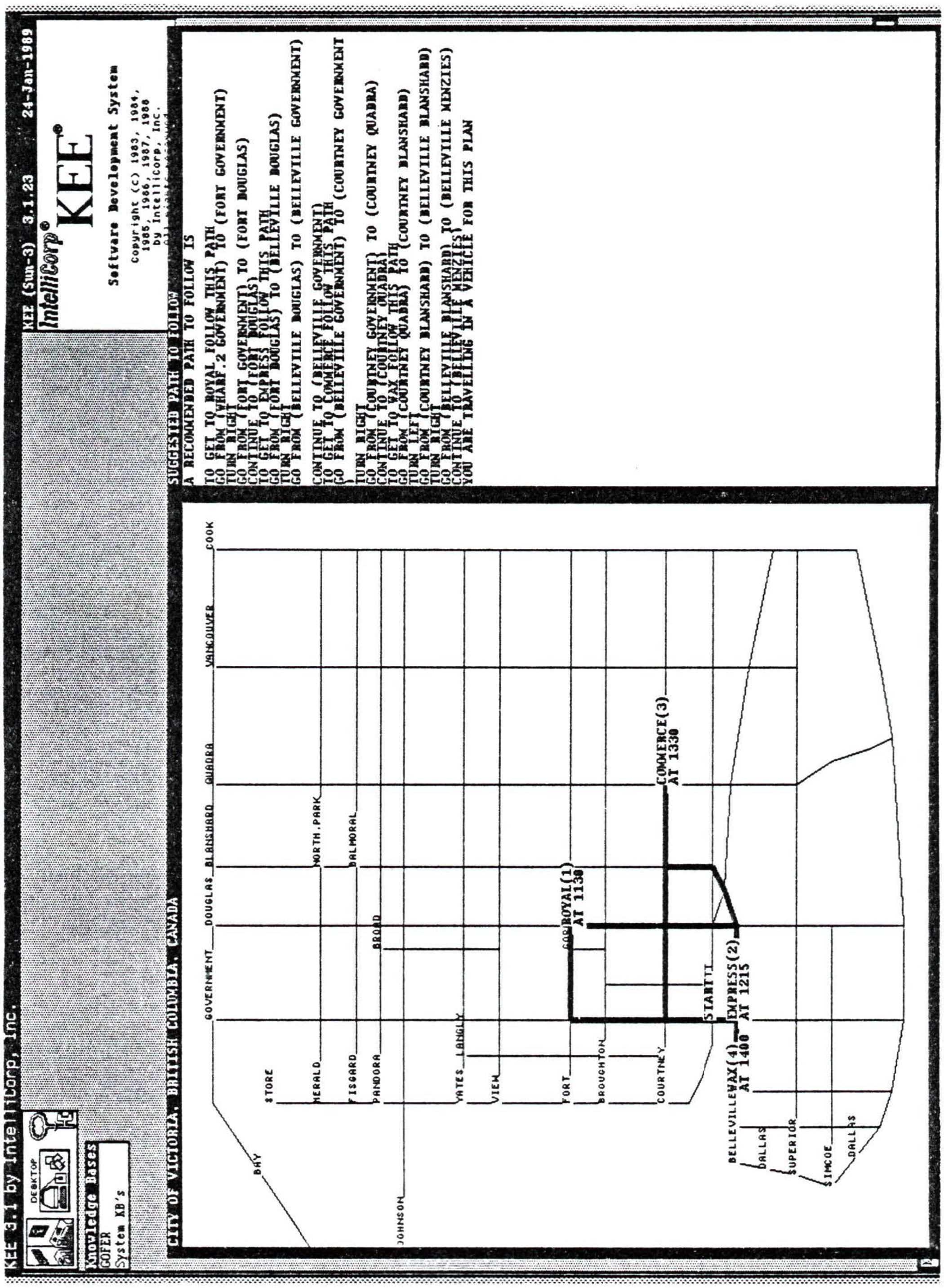

Figure A-61 Protocol 6 Test 2

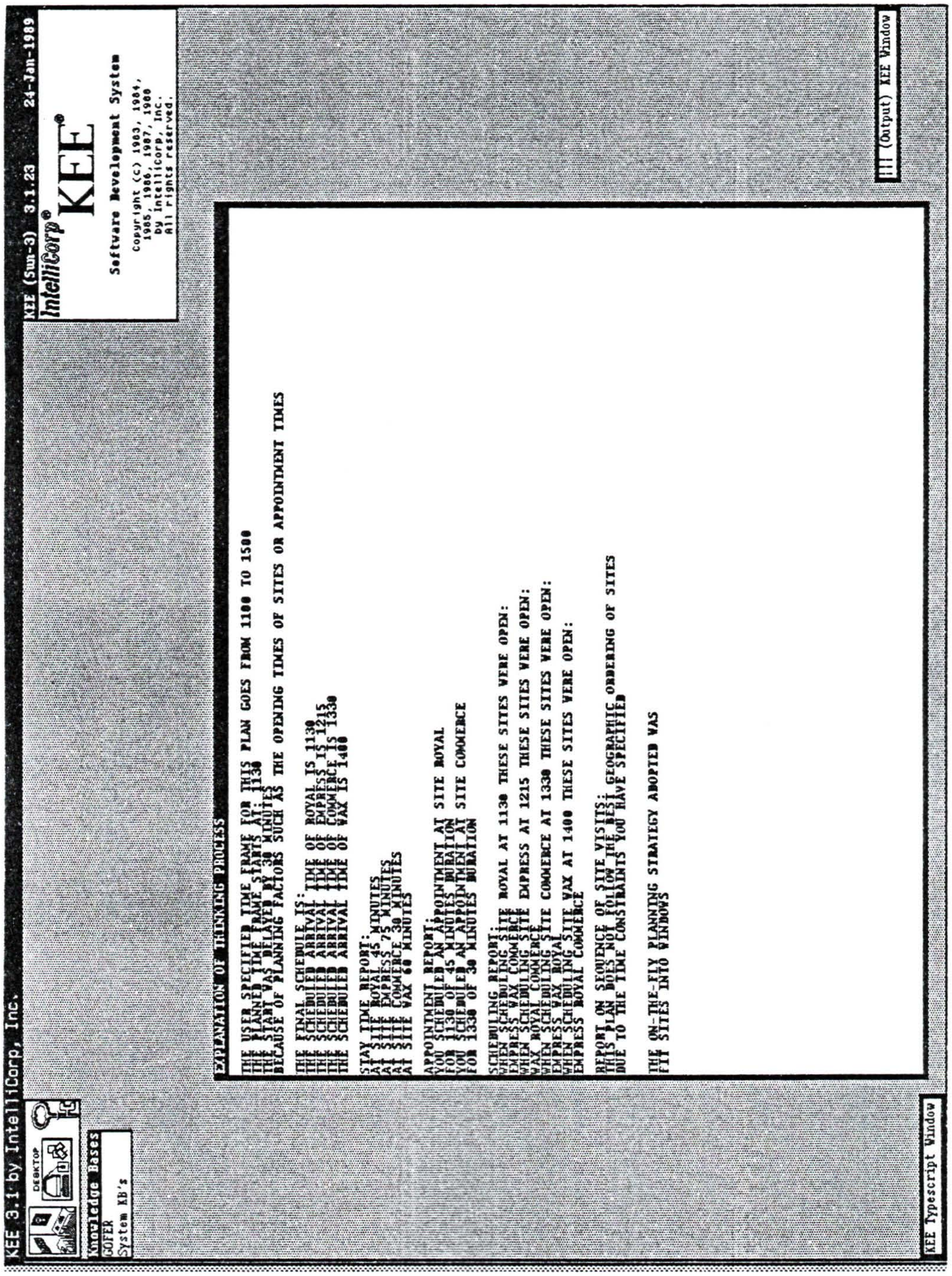

Figure A-62 Protocol 6 Test 2

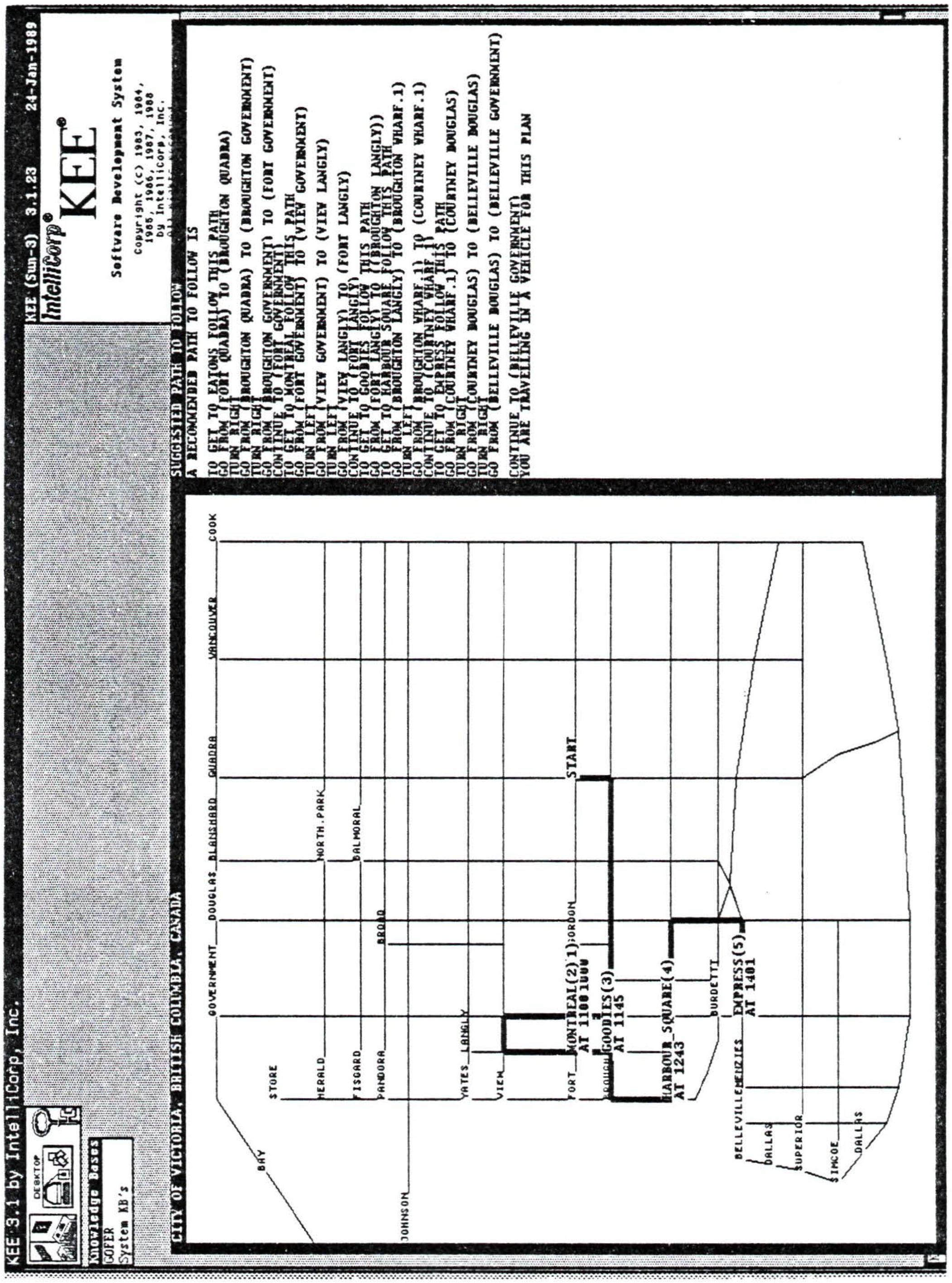

Figure A-63 Protocol 6 Test 3

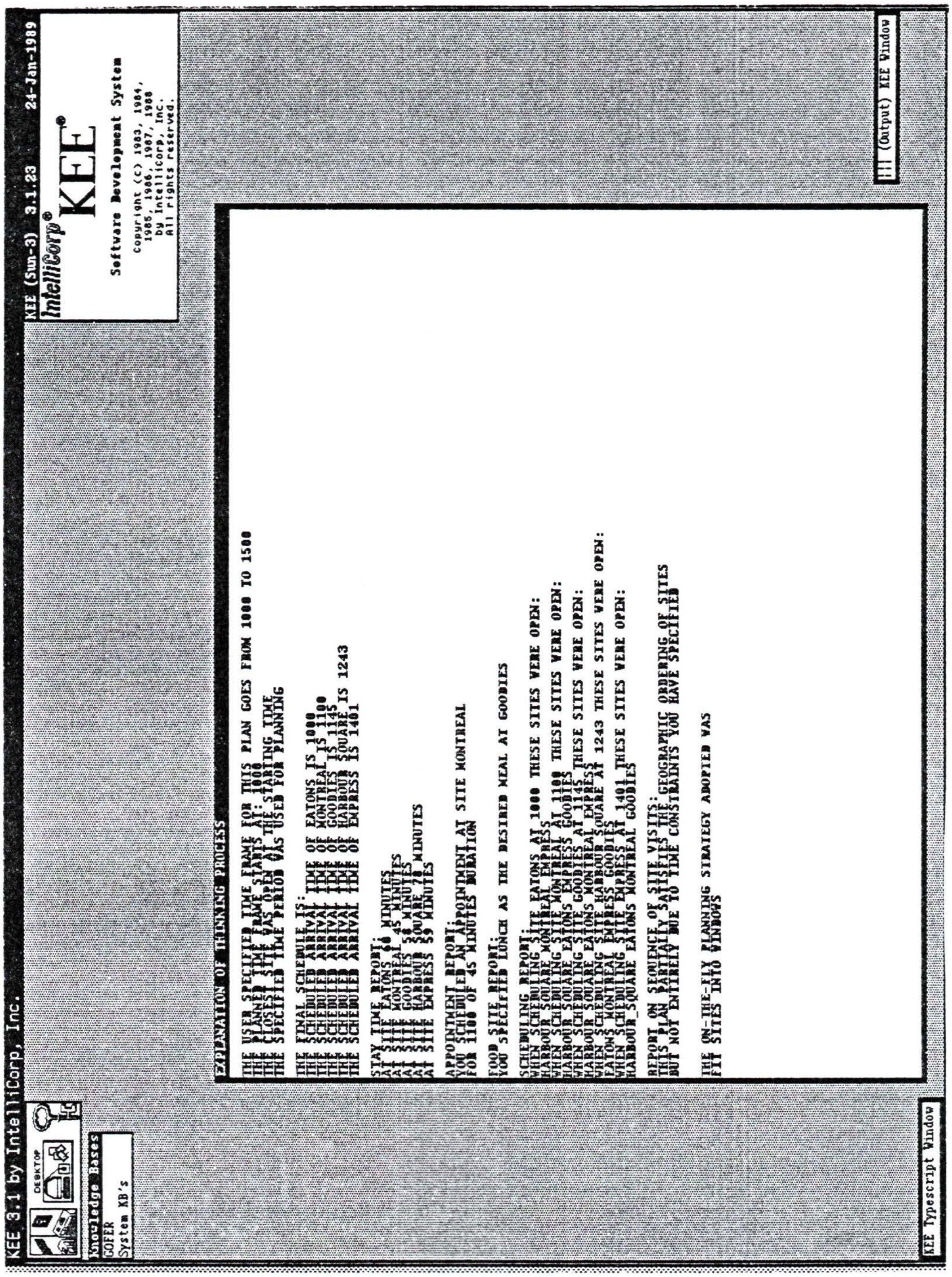

Figure A-64 Protocol 6 Test 3

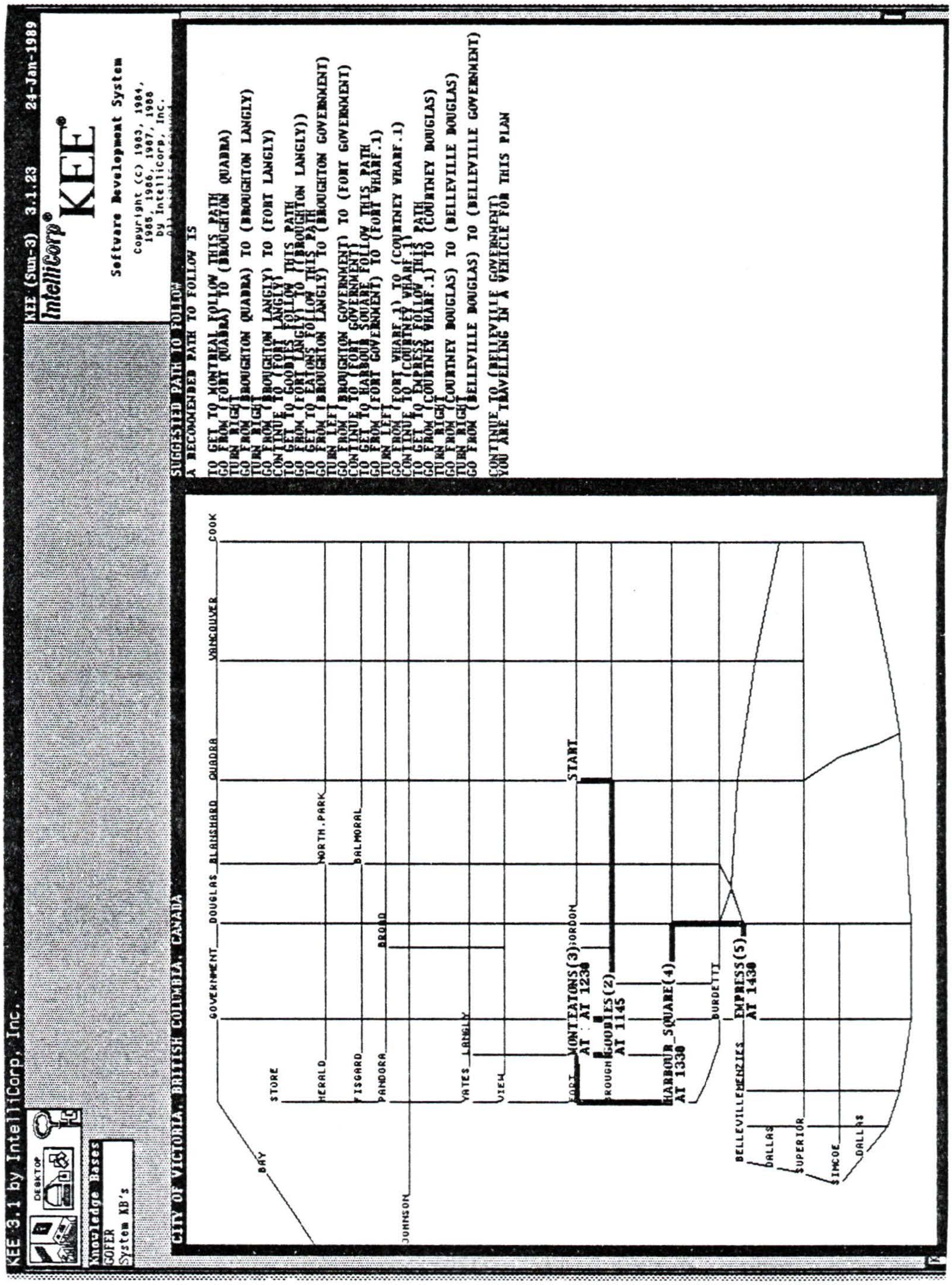

Figure A-65 Protocol 6 Test 4

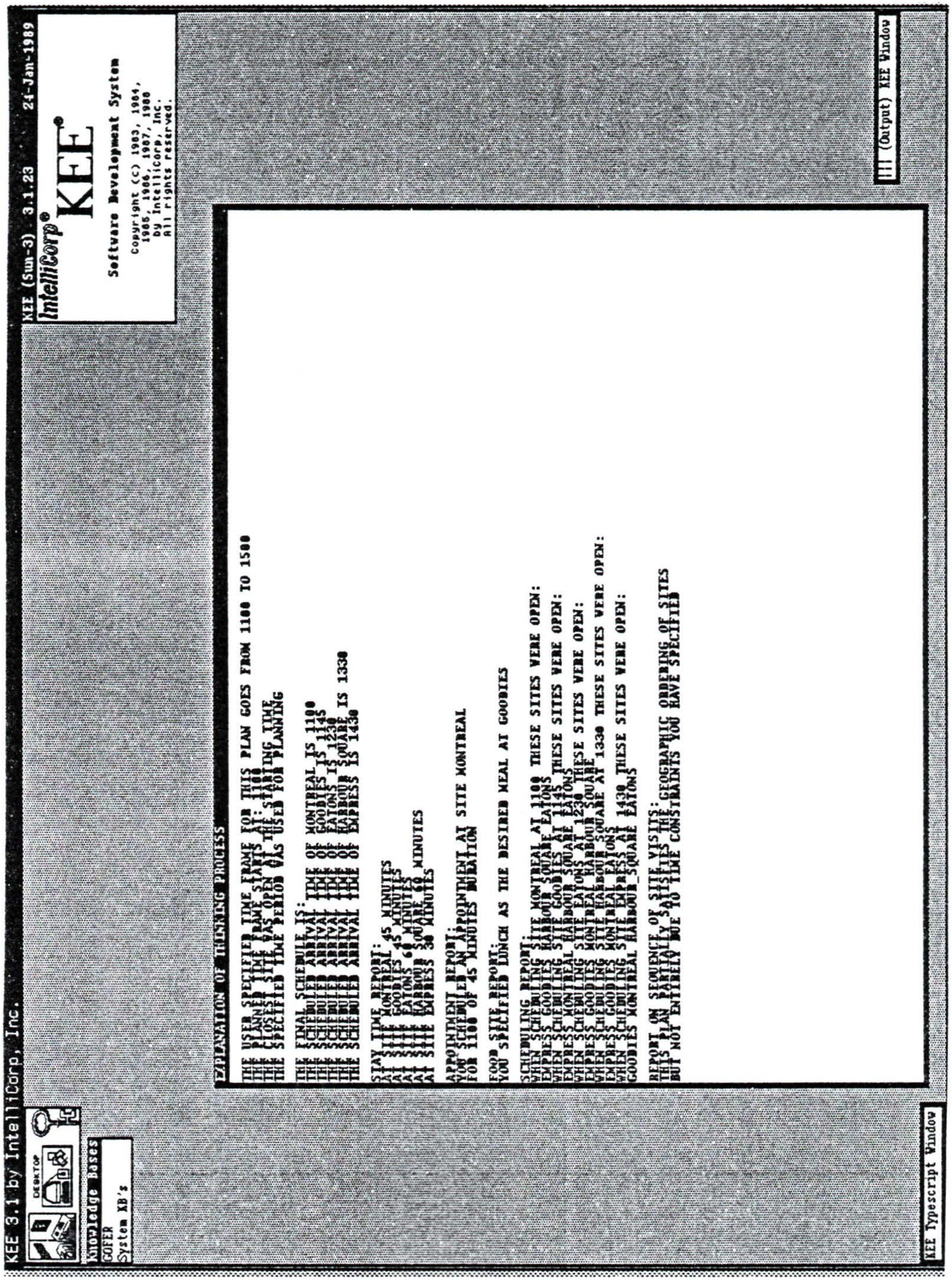

Figure A-66 Protocol 6 Test 4

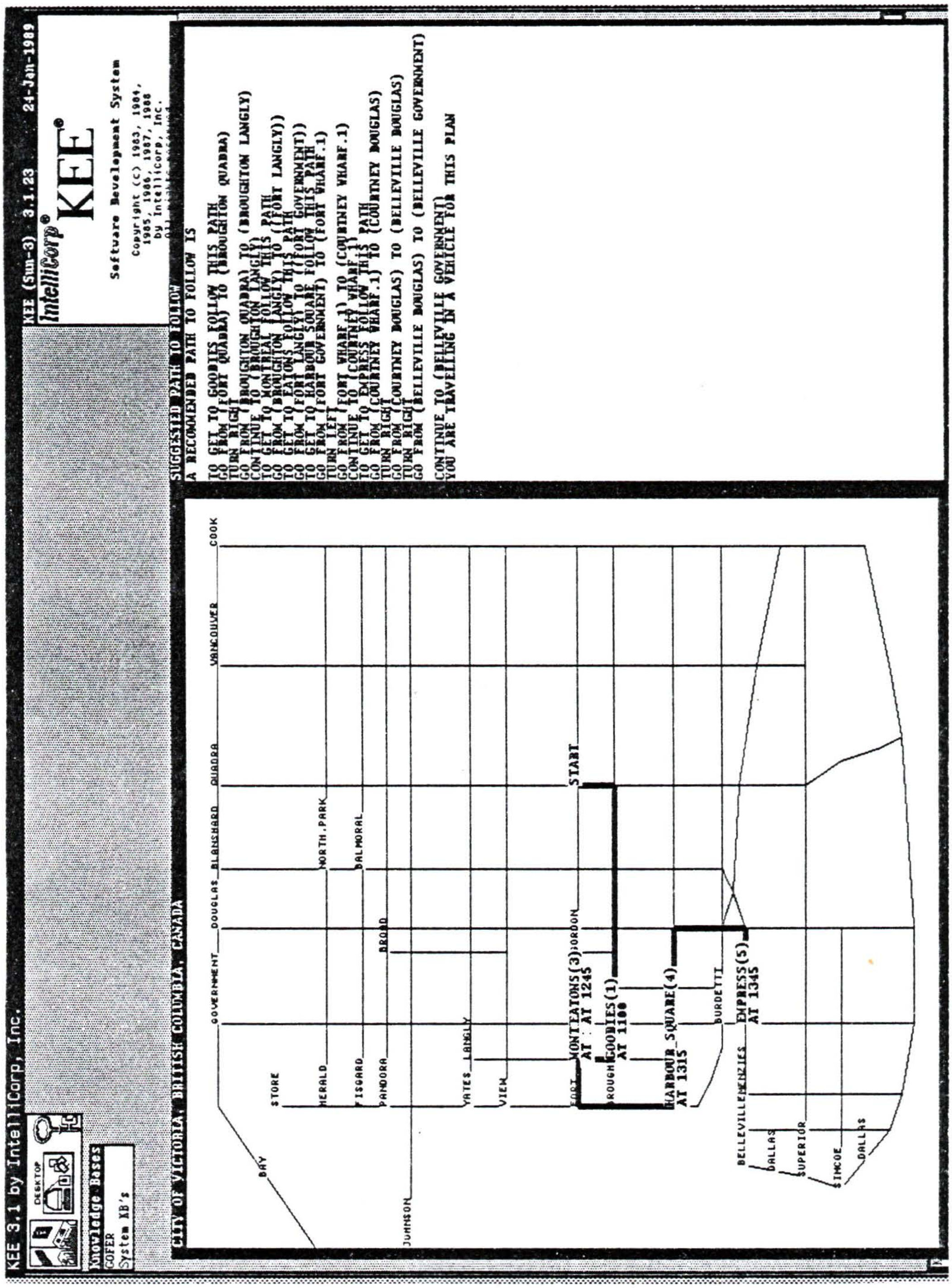

Figure A-67 Protocol 6 Test 5

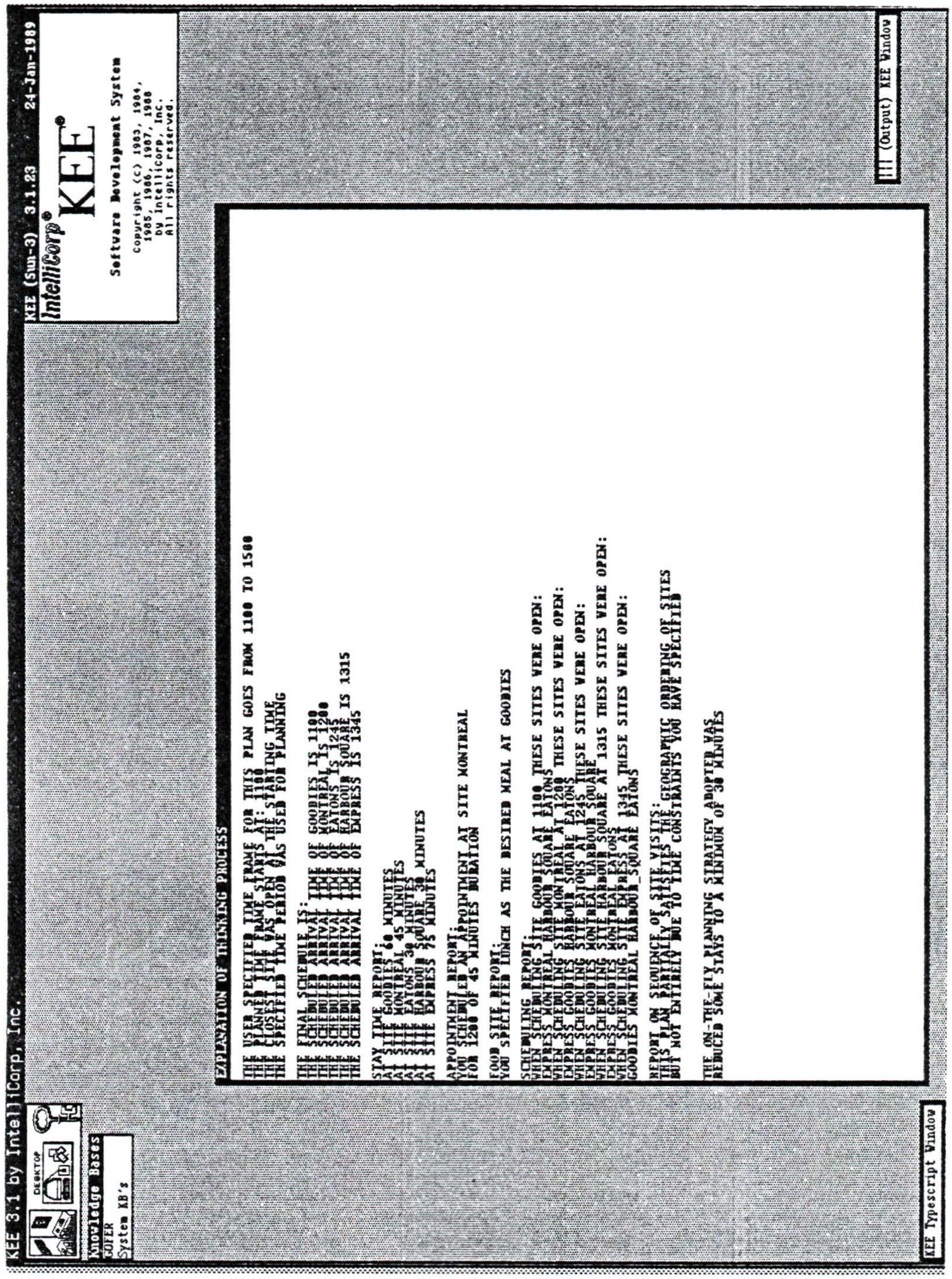

Figure A-68 Protocol 6 Test 5

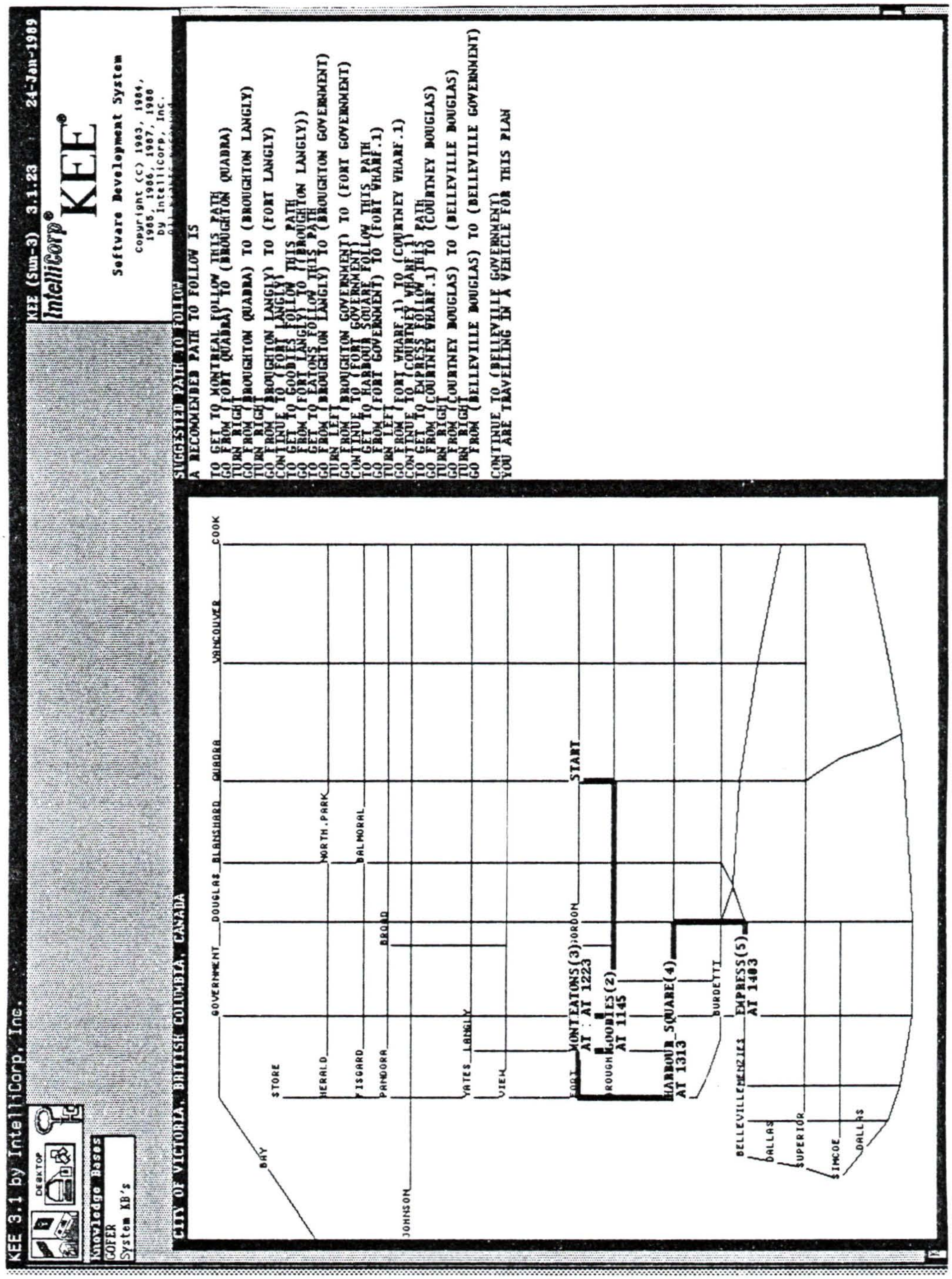

Figure A-69 Protocol 6 Test 6

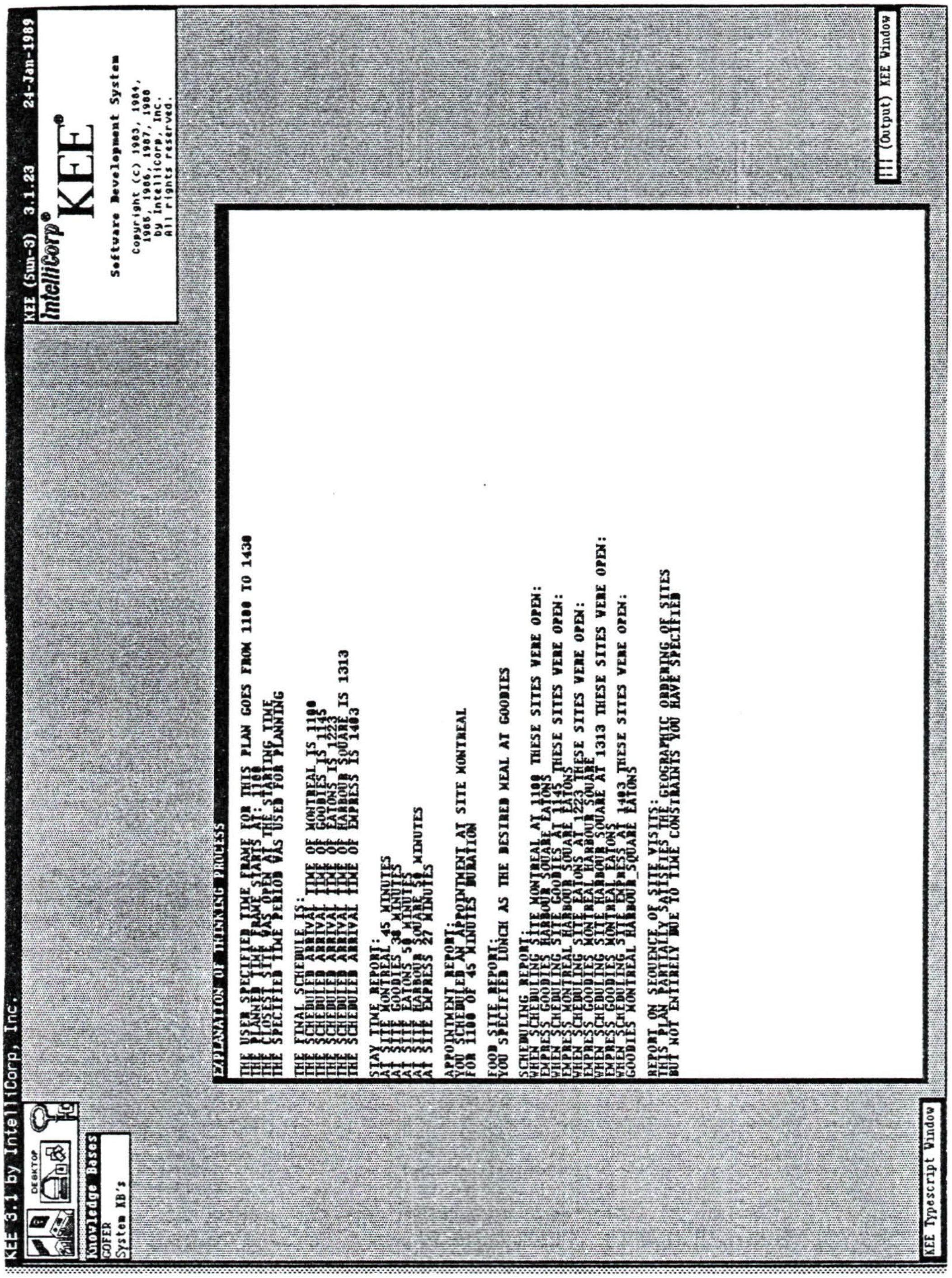

Figure A-70 Protocol 6 Test 6

A-1-7 Protocol 7

The tests in this protocol have all been generated by testers acting as tourists. They were given a brief summary of the features of the tourist information planner and allowed to generate three tests. The tester graded the test either easy, medium or hard and scored the plan as one of unacceptable, acceptable or excellent. A discussion of the results can be found in Section 4.5.8.

Test 1a

Time Span: 0930-1500

Sites:Goodies,Commerce Bank,Royal Bank,Montreal Bank,Eatons

Mode of Travel: car

Starting Intersection: (North Park Vancouver)

Appointments:

Royal Bank at 1000 for 45 minutes

Commerce Bank at 1100 for 40 minutes

Montreal Bank at 1300 for 45 minutes

Notes: Goodies for lunch

Processing Time: 3:04

Test1b

Time Span: 0900-2100

Sites:Joes,Benny's Bagels,Montreal Bank,Provincial Legislature, Ports

Mode of Travel: car

Starting Intersection: (Herald Government)

Appointments:

Benny's Bagels at 1200 for 30 minutes

Ports for dinner at 1900

Notes: Joes at (Simcoe Dallas), Benny's Bagels at (Pandora Store) are sites not in the knowledge base but were added interactively by the tourist.

Processing Time: 1:55

Test 1c

Time Span: 0800-2200

Sites: London Drugs, Cruise Ship, Eatons, Goodies, Classic Car Museum, Wax Museum, Ports

Mode of Travel: car

Starting Intersection: (Pandora Government)

Appointments:

London Drugs at 0800 for 60 minutes

Ports for dinner at 2000 for 120 minutes

Notes: Goodies for lunch, London Drugs at (Yates Vancouver) and Cruise Ship at (Simoce Dallas) were added interactively

Processing Time: 2:37

Test 2a

Time Span: 0900-1400

Sites: Wax Museum, Classic Car Museum, Ports

Mode of Travel: car

Starting Intersection: (Bay Quadra)

Notes: Ports for lunch

Processing Time: 1:10

Test 2b

Time Span: 0800-1630

Sites: Police, Classic Car Museum, Maritime Museum, Eatons, Market Square

Mode of Travel: foot

Starting Intersection: (Simcoe Dallas)

Appointments:

Police at 1100 for 15 minutes

Eatons at 1430 for 30 minutes

Notes: Police site not an urgent visit.

Processing Time: 2:10

Test 2c

Time Span: 0800-1630

Sites: Police, Wax Museum, Maritime Museum, Commerce Bank, Smittys, Rattenbury

Mode of Travel: car

Starting Intersection: (View Cook)

Appointments:

Police at 1000 for 30 minutes

Commerce at 1430 for 15 minutes

Smittys at 0800 for 60 minutes

Notes: Rattenbury for lunch, Smittys at (Courtney Government) was an added site.

Processing Time: 2:15

Test 3a

Time Span: 1000-1400

Sites: Market Square, Montreal Bank, Wax Museum

Mode of Travel: foot

Starting Intersection: (North.Park Quadra)

Processing Time: 0:48

Test 3b

Time Span: 1000-1800

Sites: Eatons, Market Square, Wax Museum, Goodies, Royal Bank, City Hall

Mode of Travel: car

Starting Intersection: (Superior Vancouver)

Appointments:

Royal Bank at 1000 for 15 minutes

City Hall at 1100 for 30 minutes

Notes: Goodies for lunch

Processing Time: 3:43

Test 3c

Time Span: 1100-1800

Sites: Dots, Fabricland, Eatons, Harbour Square, Market Square, Wax Museum, Goodies, Commerce, Ports

Mode of Travel: foot

Starting Intersection: (Johnson Wharf)

Appointments:

Eatons at 1300 for 60 minutes

Commerce at 1100 for 15 minutes

Notes: Ports dinner reservation 1700 for 60 minutes, Goodies for lunch, Dots at (Yates Douglas) and Fabricland at (Johnson Quadra) were added sites

Processing Time: 3:17

Test 4a

Time Span: 0900-1700

Sites: McDonalds, Harbour Square, Classic Car Museum, Empress

Mode of Travel: car

Starting Intersection: (Bay Blanshard)

Appointments:

McDonalds at 0900 for 30 minutes

Notes: McDonalds at (Pandora Vancouver) was an added site

Processing Time: 2:14

Test 4b

Time Span: 0830-1300

Sites: Bistro, Rattenbury, Empress, Provincial Legislature

Mode of Travel: foot

Starting Intersection: (Belleville Menzies)

Appointments:

Bistro at 0830 for 60 minutes

Provincial Legislature at 1000 for 120 minutes

Notes: Bistro at (Belleville Menzies) was an added site, Rattenbury for lunch

Processing Time: 2:00

Test 4c

Time Span: 1100-2000

Sites: Maritime Museum, Ports, Police Station, Harbour Sqare, Market Square, City Hall, Goodies, Royal Bank

Mode of Travel: car

Starting Intersection: (Bay Douglas)

Appointments:

Police at 1400 for 30 minutes

City Hall at 1600 for 120 minutes

Ports at 1800 for 120 minutes

Notes: Goodies for lunch

Processing Time: 3:07

Test 5a

Time Span: 1200-1500

Sites: Goodies, Eatons, Commerce Bank

Mode of Travel: car

Starting Intersection: (Fort Douglas)

Notes: Goodies for lunch

Processing Time: 1:45

Test 5b

Time Span: 1200-2000

Sites: Ports, Rattenbury, Wax, Maritime, Royal, Police, Ferry

Mode of Travel: car

Starting Intersection: (Bay Cook)

Appointments:

Royal at 1400 for 30 minutes

Notes: Rattenbury dinner reservation for 1800 for 120 minutes, Ports for lunch, Ferry was an added site at (Simcoe Dallas)

Processing Time: 2:53

Test 6a

Time Span: 1200-2000

Sites: Classic Car Museum, Maritime Museum, Laurel Point Inn

Mode of Travel: foot

Starting Intersection: (Bay Vancouver)

Processing Time: 0:39

<u>Test 6b</u>

Time Span: 1000-1500

Sites: Harbour Square, Police, Ports, Commerce Bank

Mode of Travel: car

Starting Intersection: (Broughton Langly)

Appointments:

Commerce Bank at 1200 for 60 minutes

Notes: Ports for lunch, Police site must be visited first

Processing Time: 1:10

<u>Test 6c</u>

Time Span: 0900-2100

Sites: Montreal Bank, Cafe Mexico, Harbour Square, Police, Rattenbury, Ports, Wax Museum

Mode of Travel: car

Starting Intersection: (Dallas Quadra)

Appointments:

Police at 1000 for 30 minutes

Montreal Bank at 1100 for 30 minutes

Cafe Mexico at 1300 for 60 minutes

Ports at 1900 for 120 minutes

Notes: Cafe Mexico was an added site at (Johnson Wharf), Ports for dinner, Rattenbury for breakfast

Processing Time: 2:53

Test 7a

Time Span: 0900-1600

Sites: Eatons, Ports, Royal Museum, Wax Museum

Mode of Travel: foot

Starting Intersection: (Bay Cook)

Notes: Ports for lunch, Royal Museum at (Belleville Government) was an added site

Processing Time: 1:36

Test 7b

Time Span: 1000-1500

Sites: Royal Bank, Goodies, Market Square, Crystal Gardens, Rattenbury

Mode of Travel: car

Starting Intersection: (Bay Cook)

Appointments:

Royal Bank at 1000 for 20 minutes

Notes: Goodies for breakfast, Rattenbury for lunch, Crystal Gardens at (Belleville Douglas) was an added site

Processing Time: 2:59

Test 7c

Time Span: 0930-1600

Sites: Kins, Commerce Bank, The Bay, McDonalds

Mode of Travel: car

Starting Intersection: (Bay Cook)

Appointments:

McDonalds at 1200 for 30 minutes

Kins at 1400 for 120 minutes

Notes: McDonalds at (Courtney Douglas) and Kins at (Superior Menzies) were added sites

Processing Time: 3:10

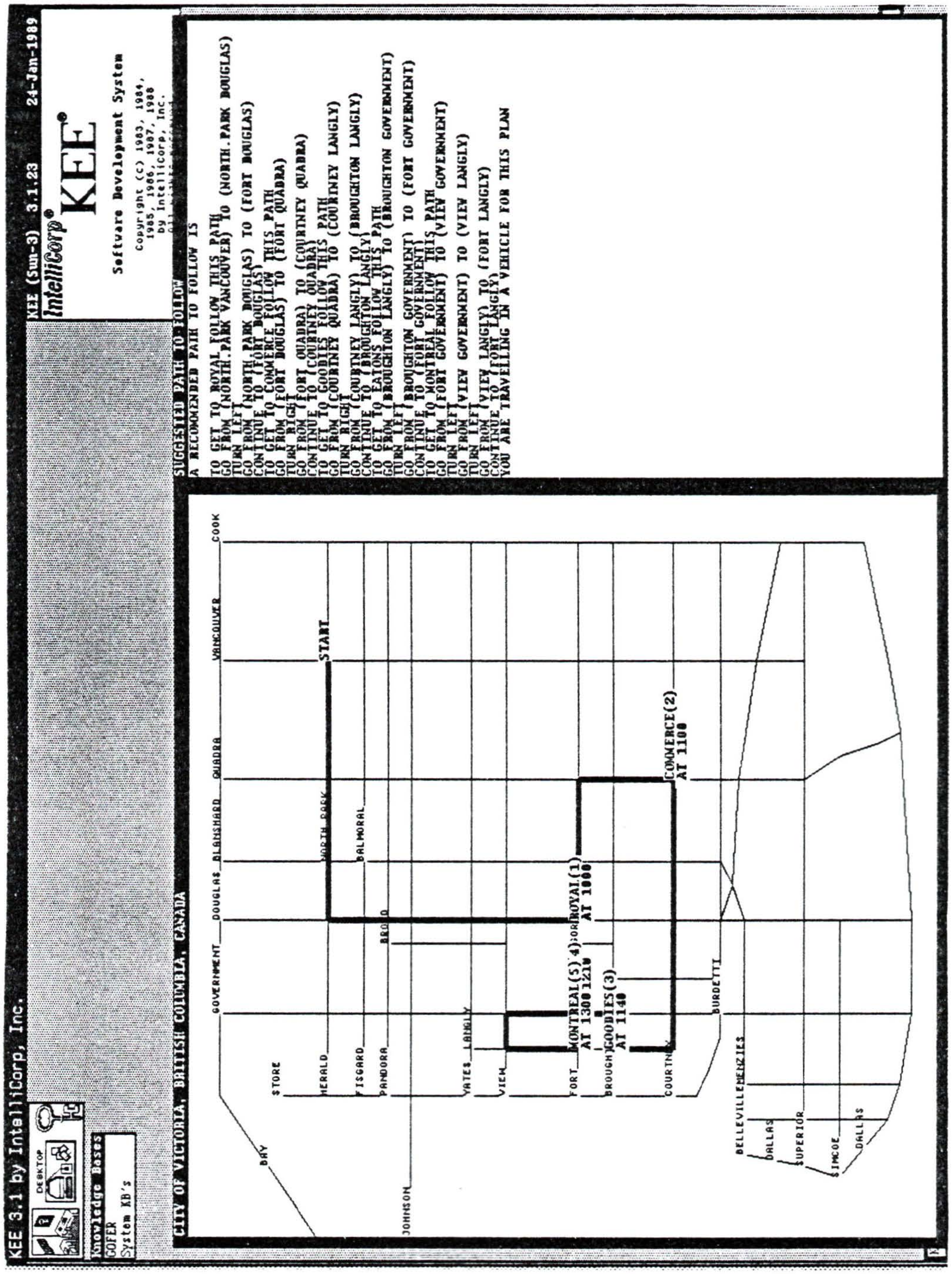

Figure A-71 Protocol 7 Test 1a

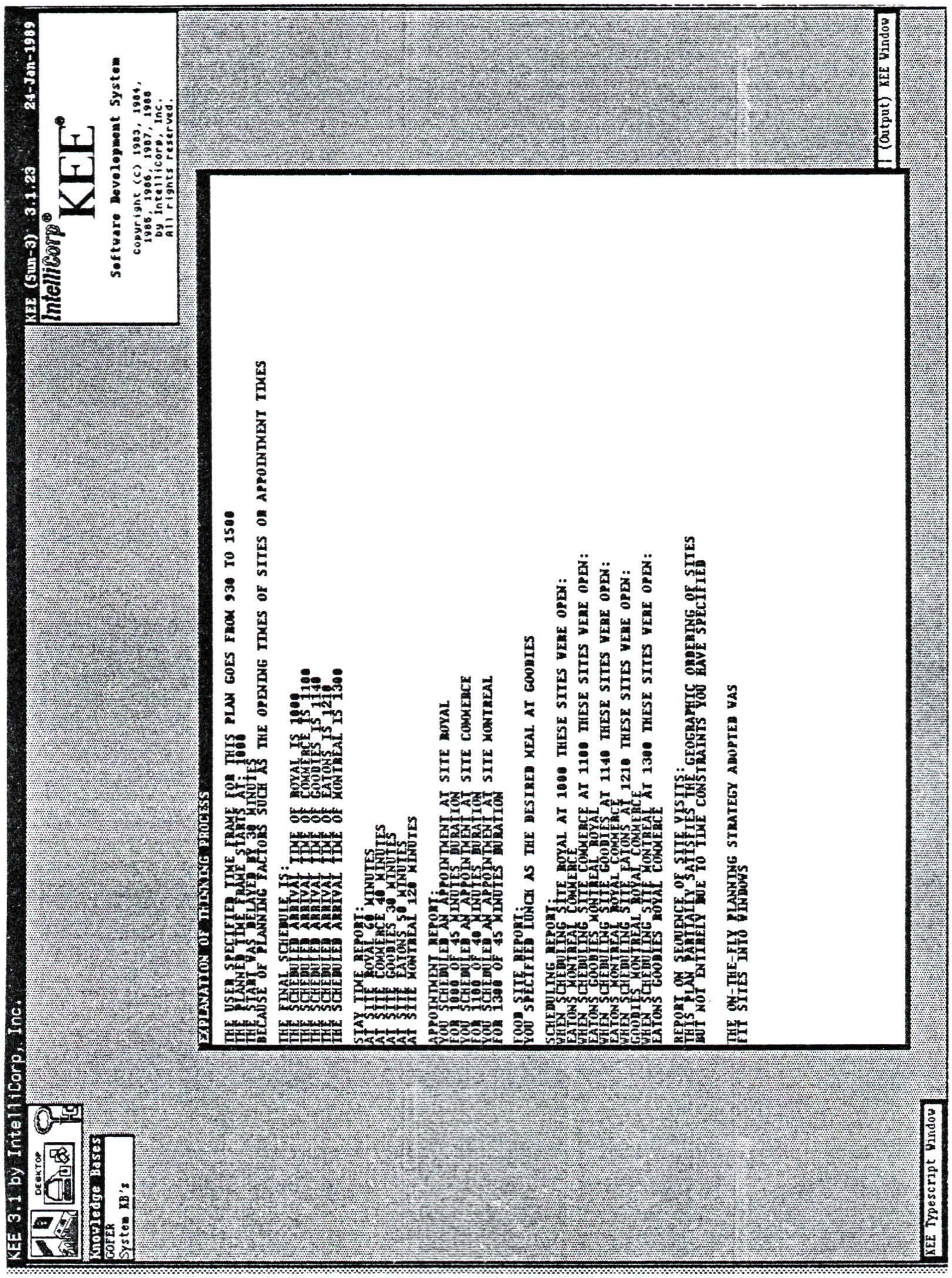

Figure A-72 Protocol 7 Test 1a

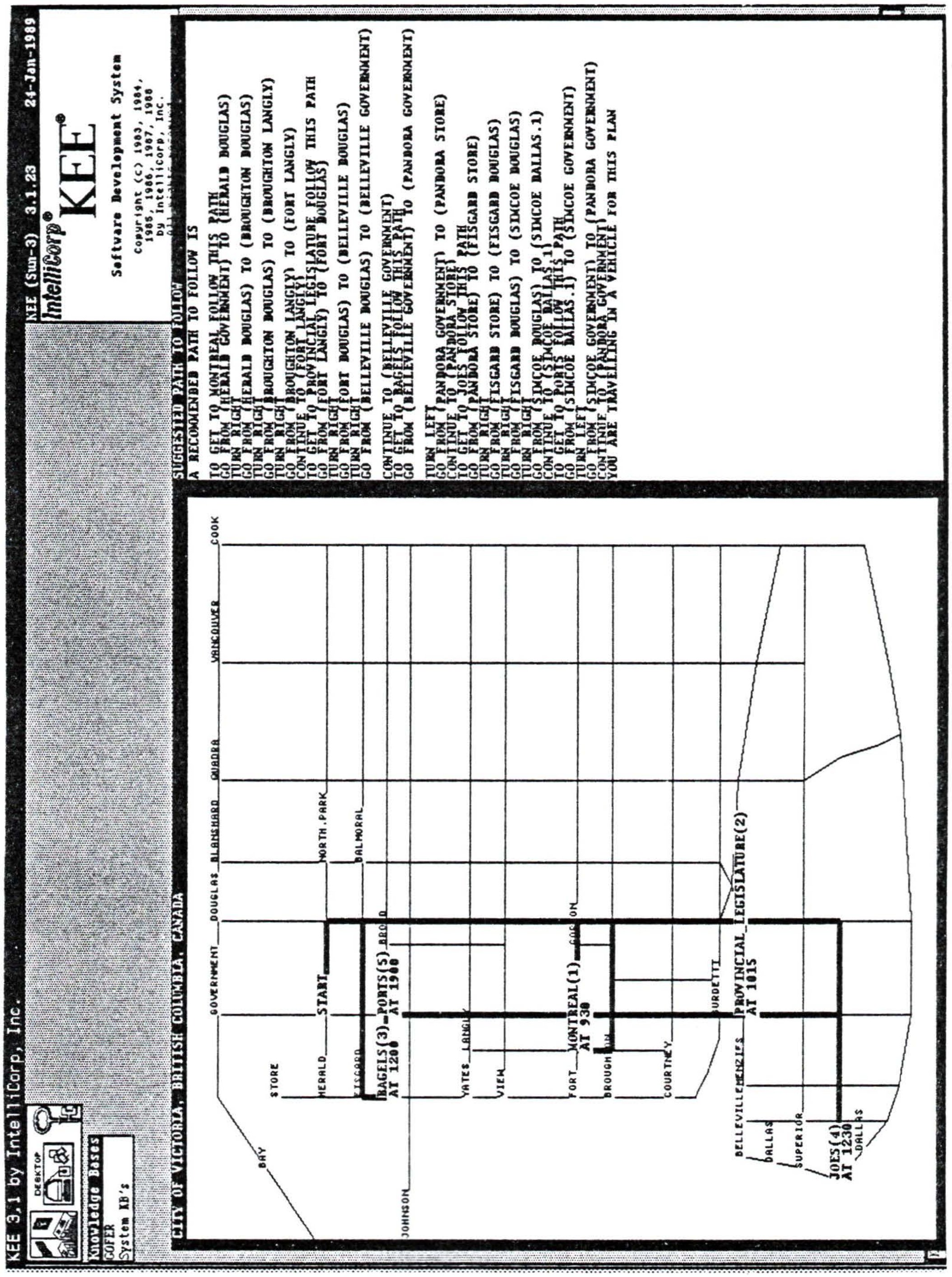

Figure A-73 Protocol 7 Test 1b

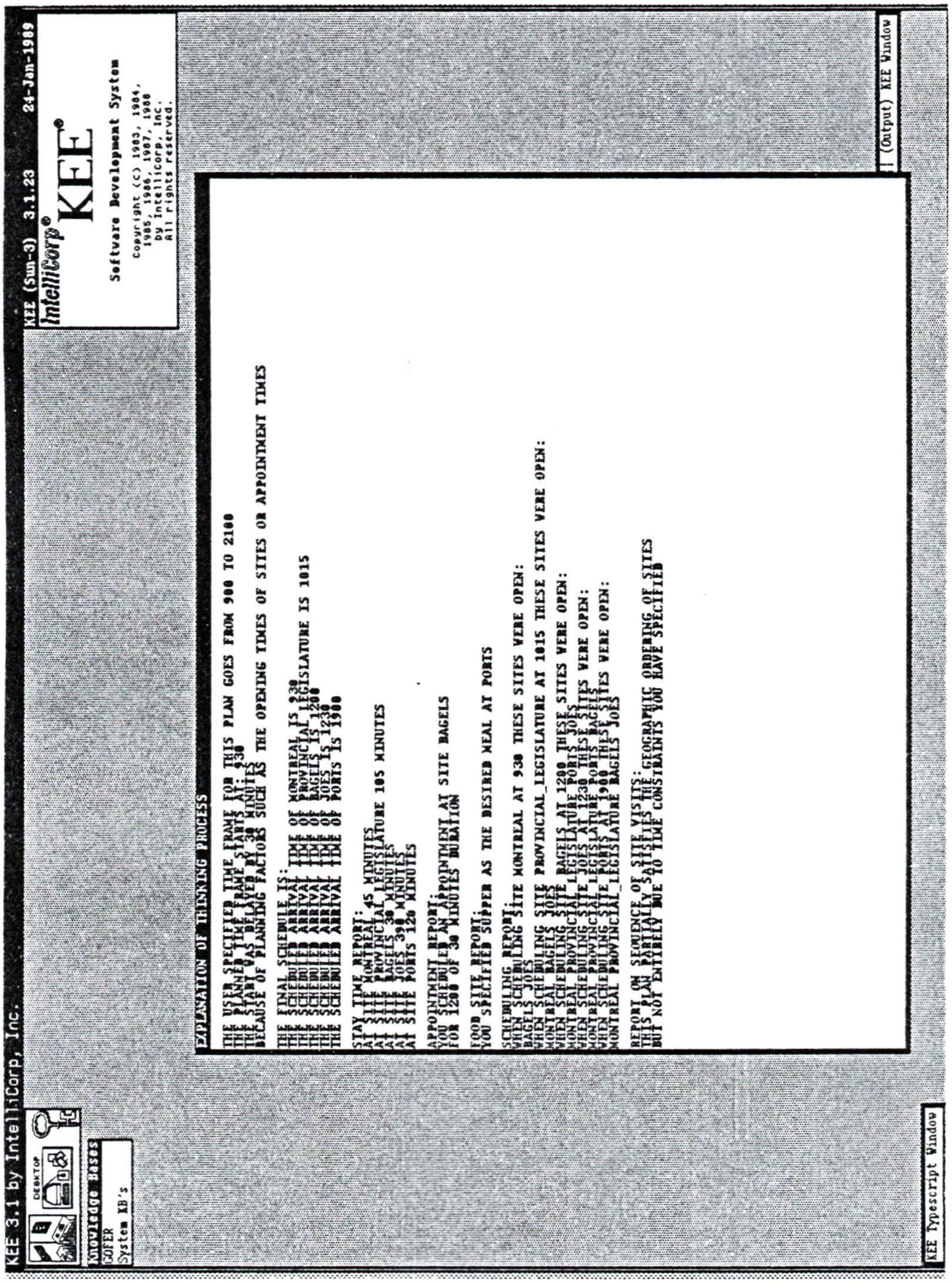

Figure A-74 Protocol 7 Test 1b

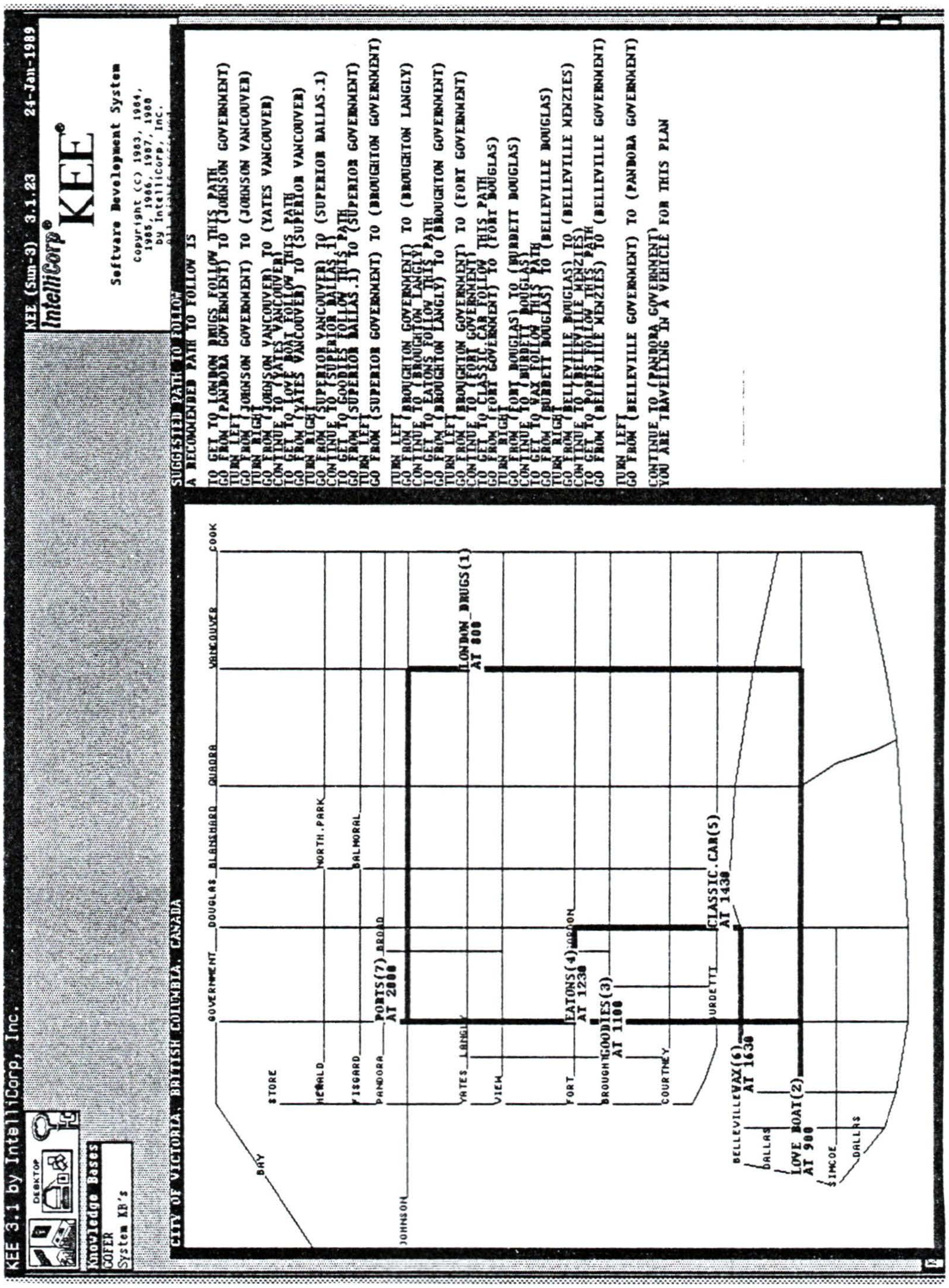

Figure A-75 Protocol 7 Test 1c

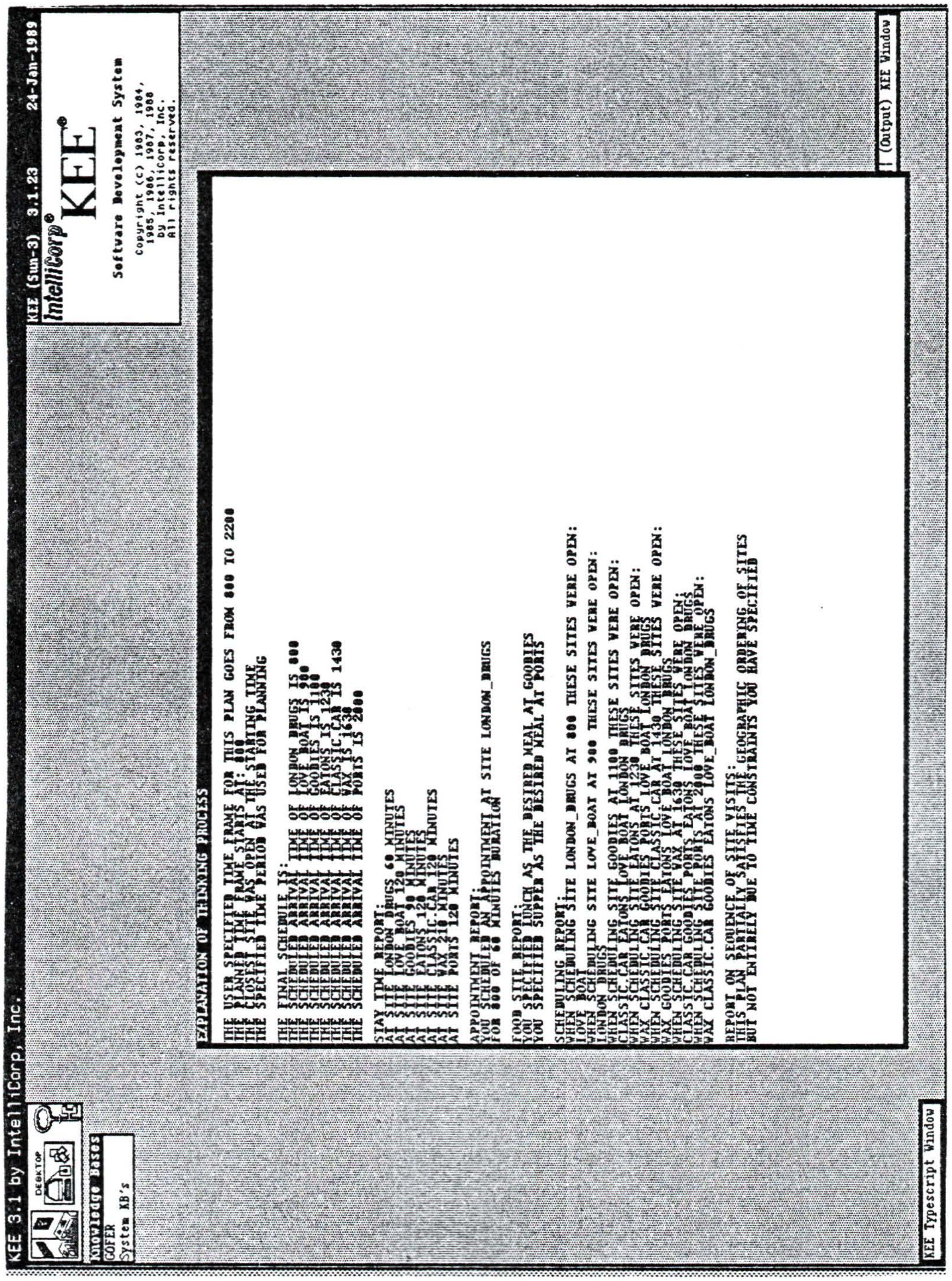

Figure A-76 Protocol 7 Test 1c

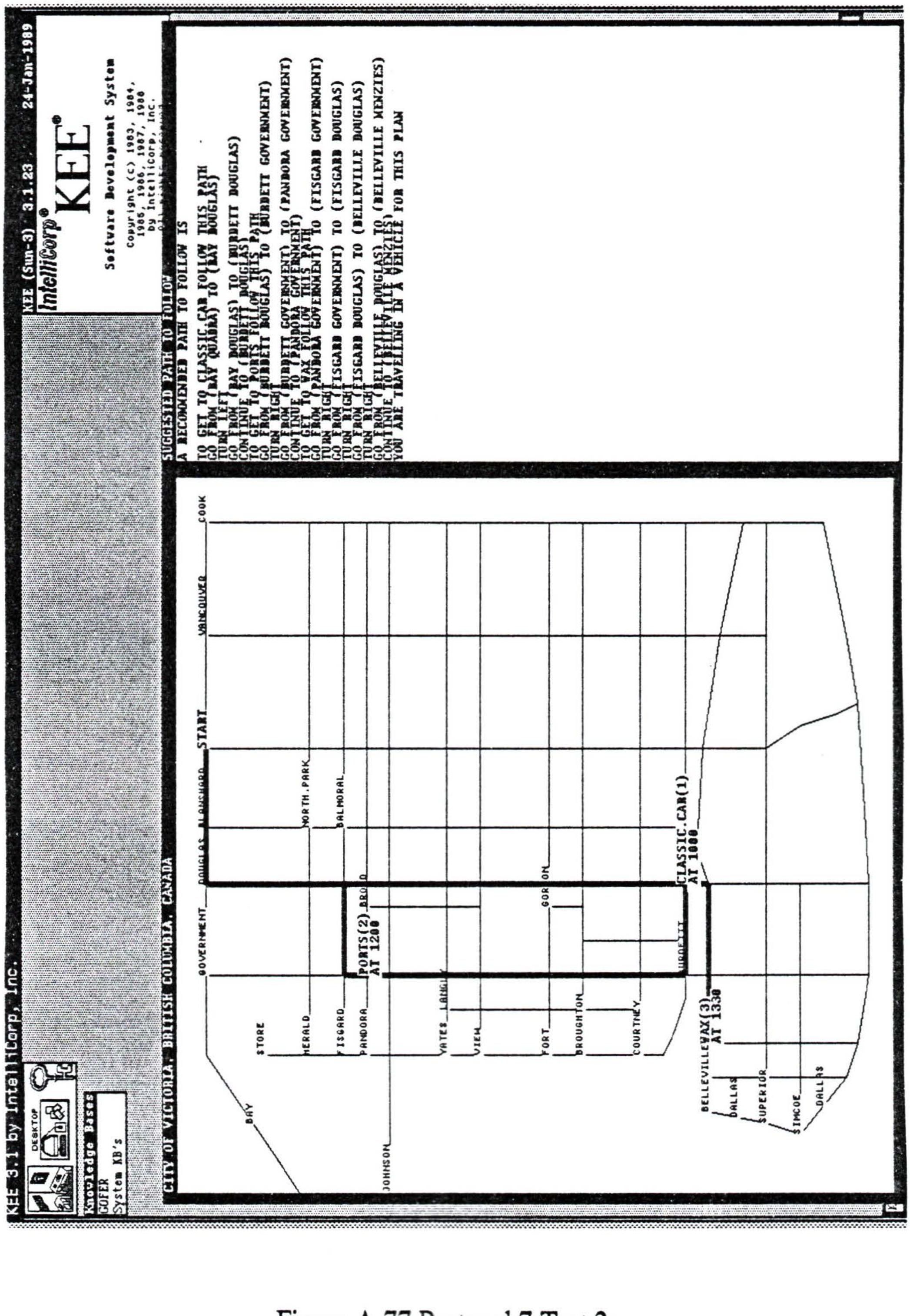

Figure A-77 Protocol 7 Test 2a

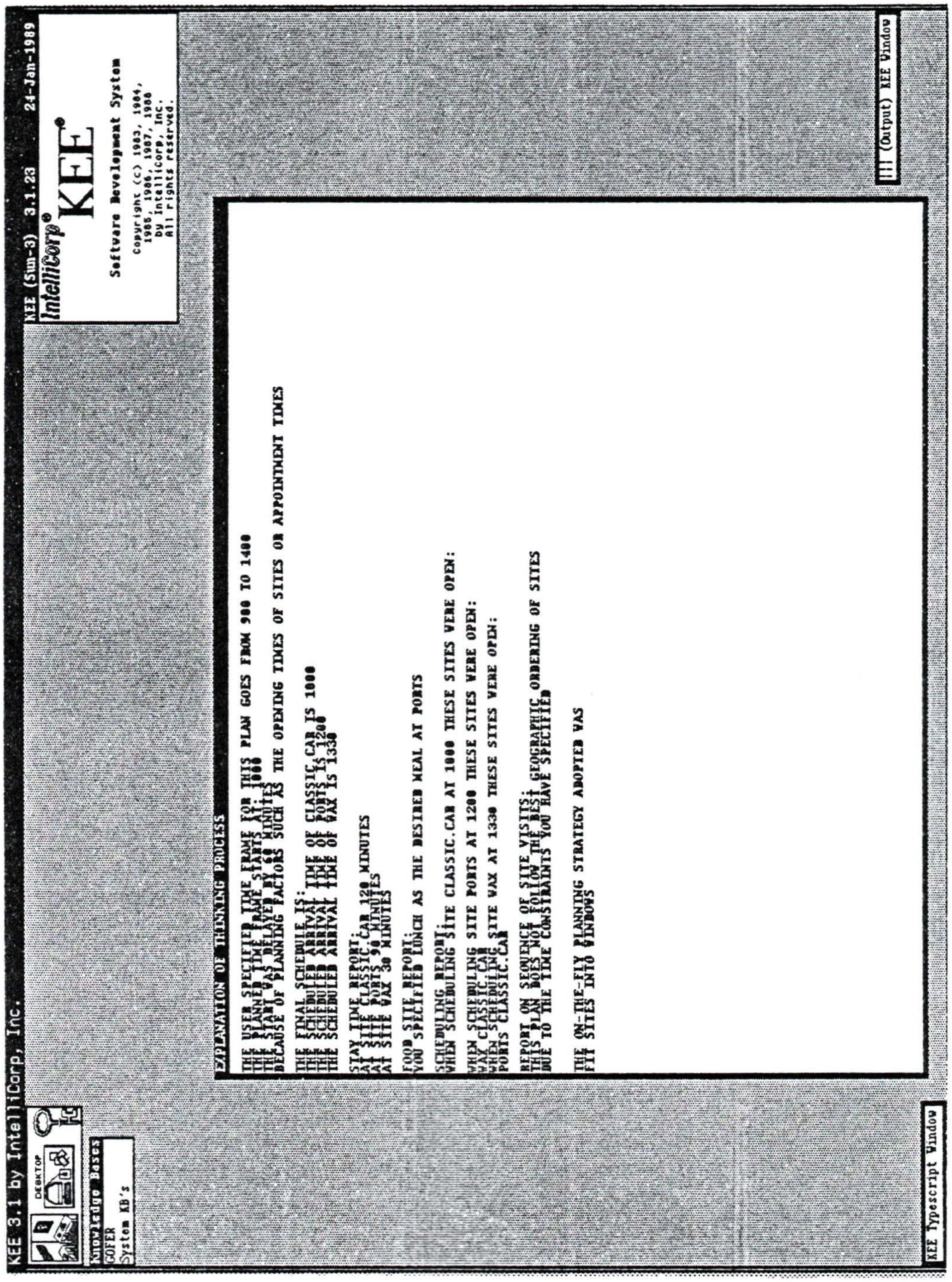

Figure A-78 Protocol 7 Test 2a

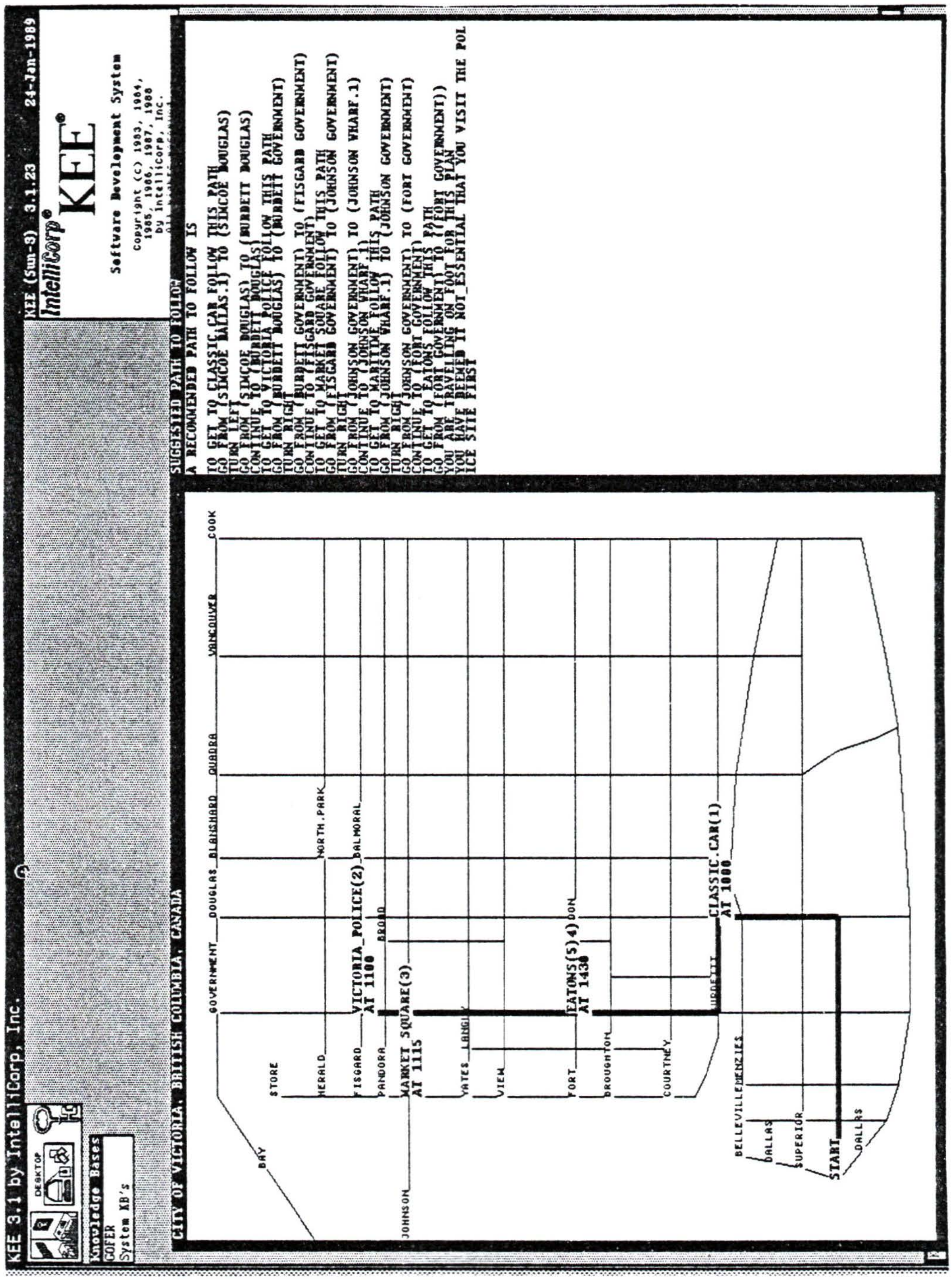

Figure A-79 Protocol 7 Test 2b

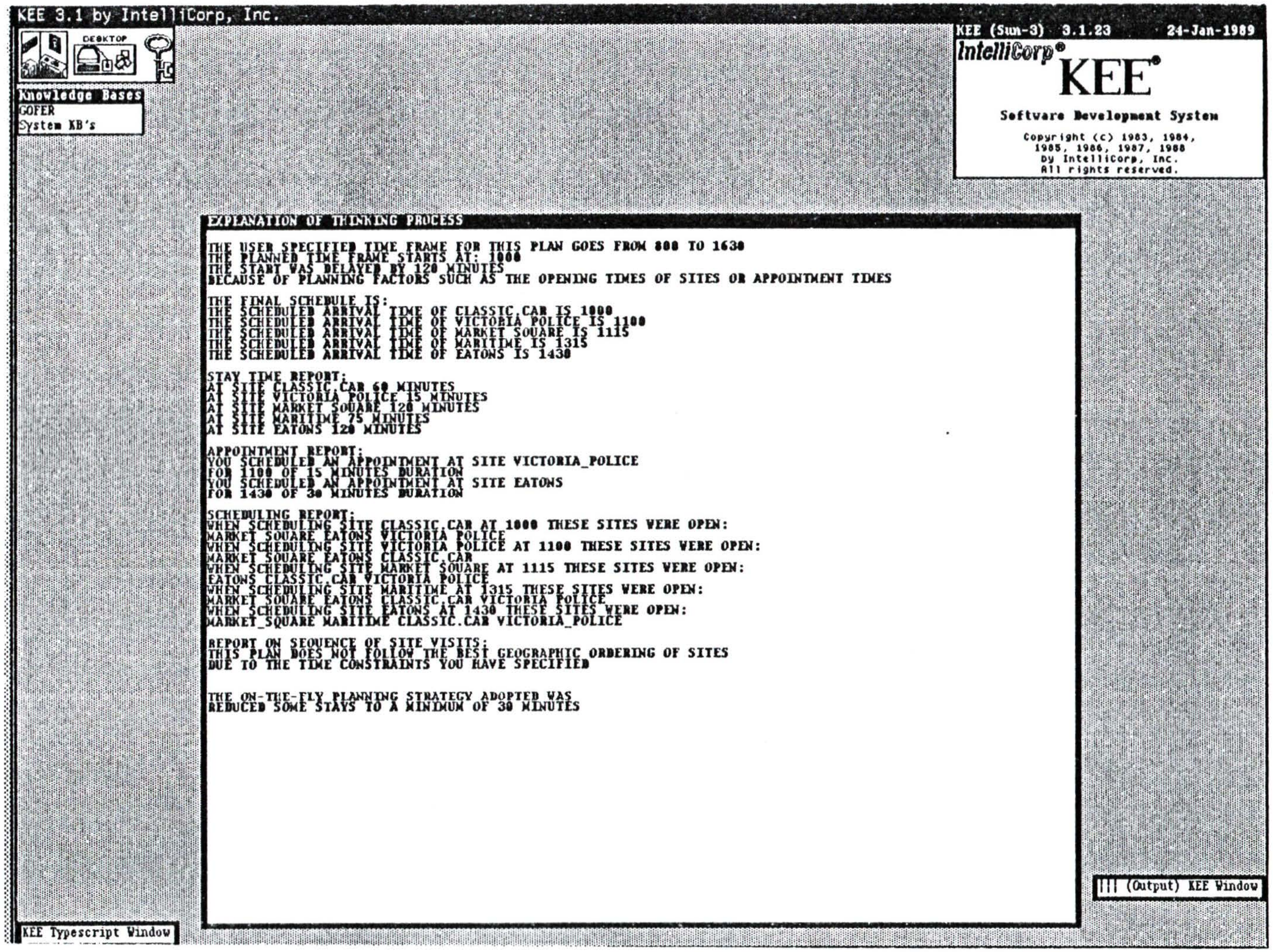

Figure A-80 Protocol 7 Test 2b

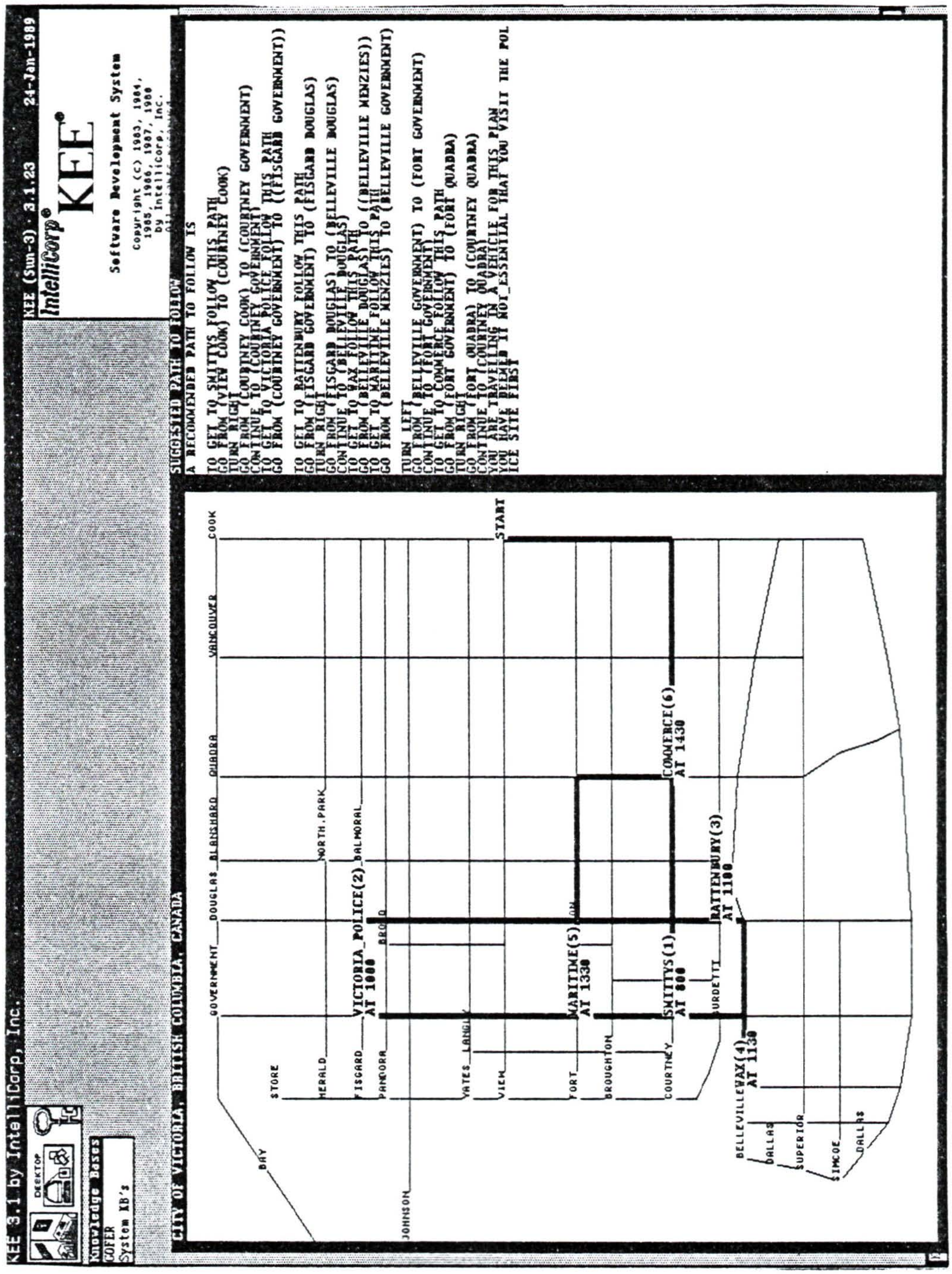

Figure A-81 Protocol 7 Test 2c

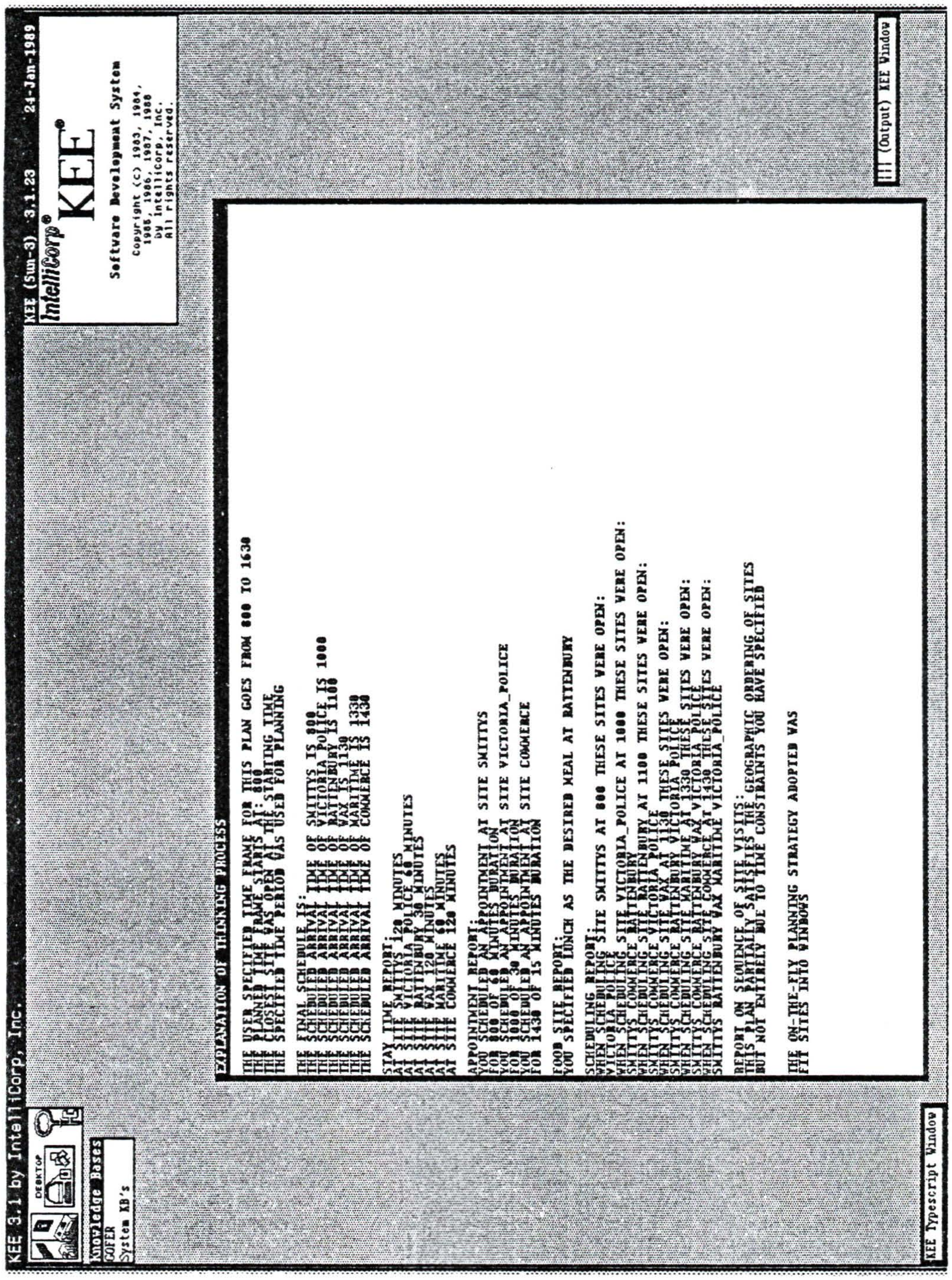

Figure A-82 Protocol 7 Test 2c

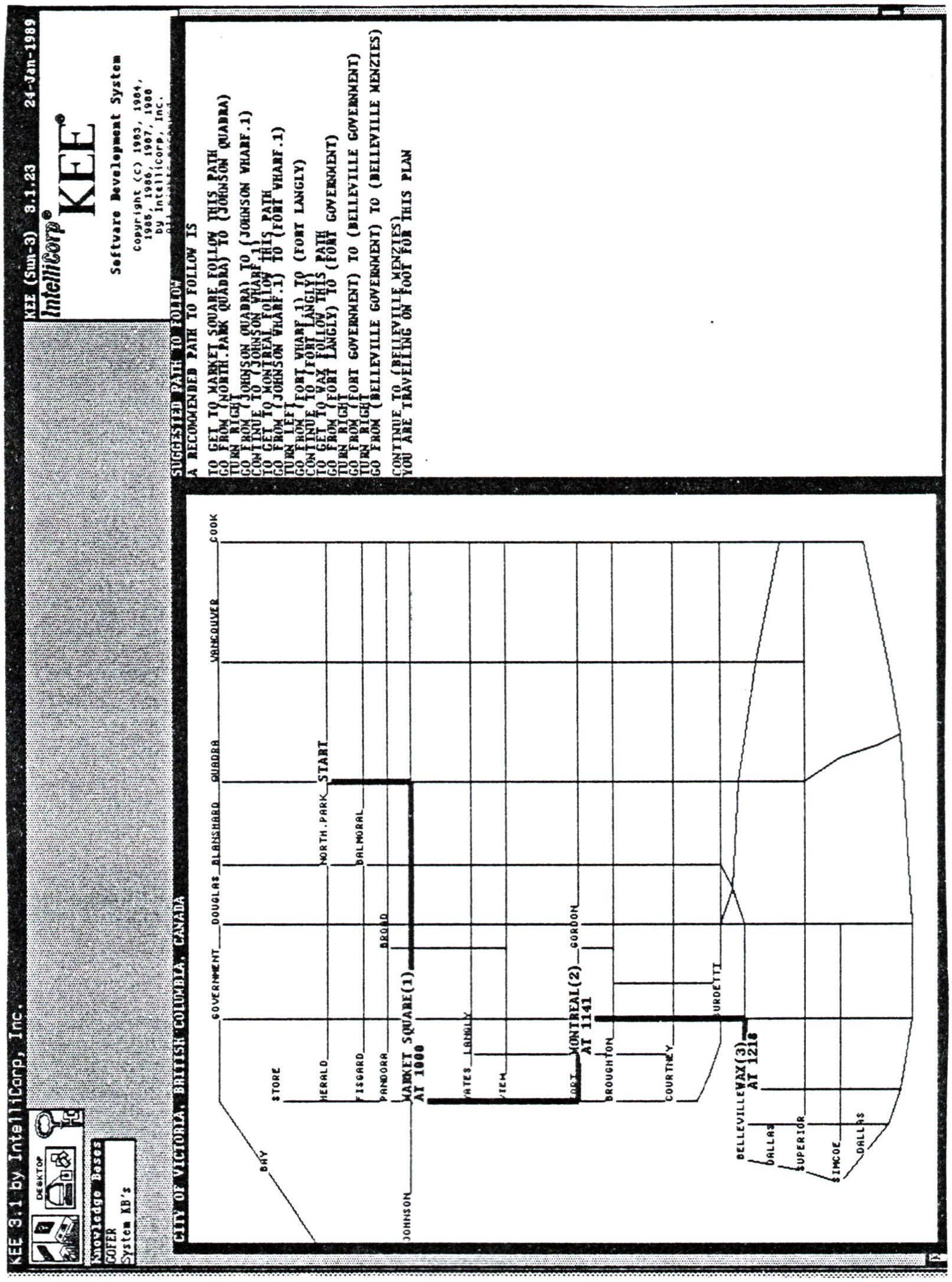

Figure A-83 Protocol 7 Test 3a

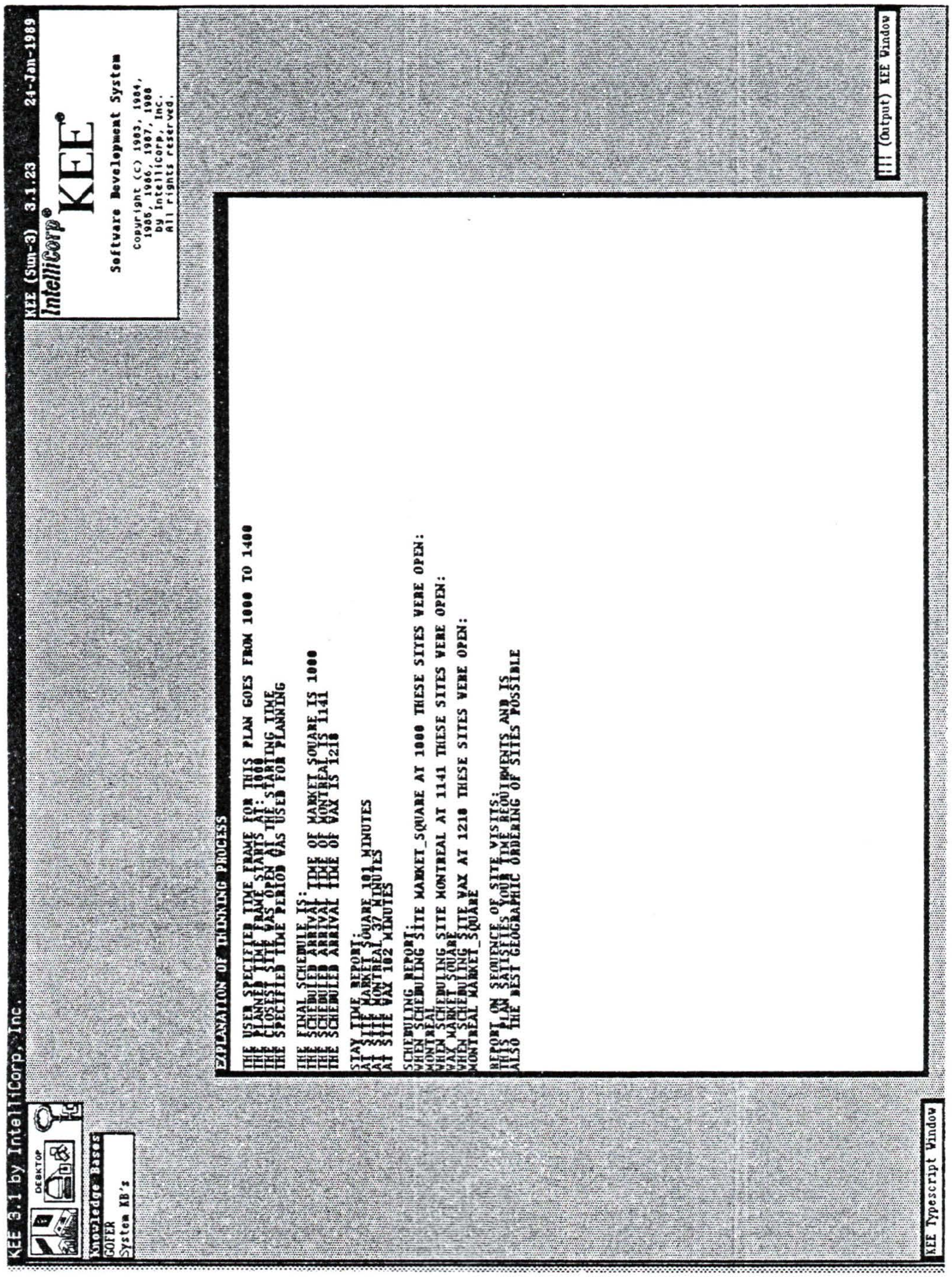

Figure A-84 Protocol 7 Test 3a

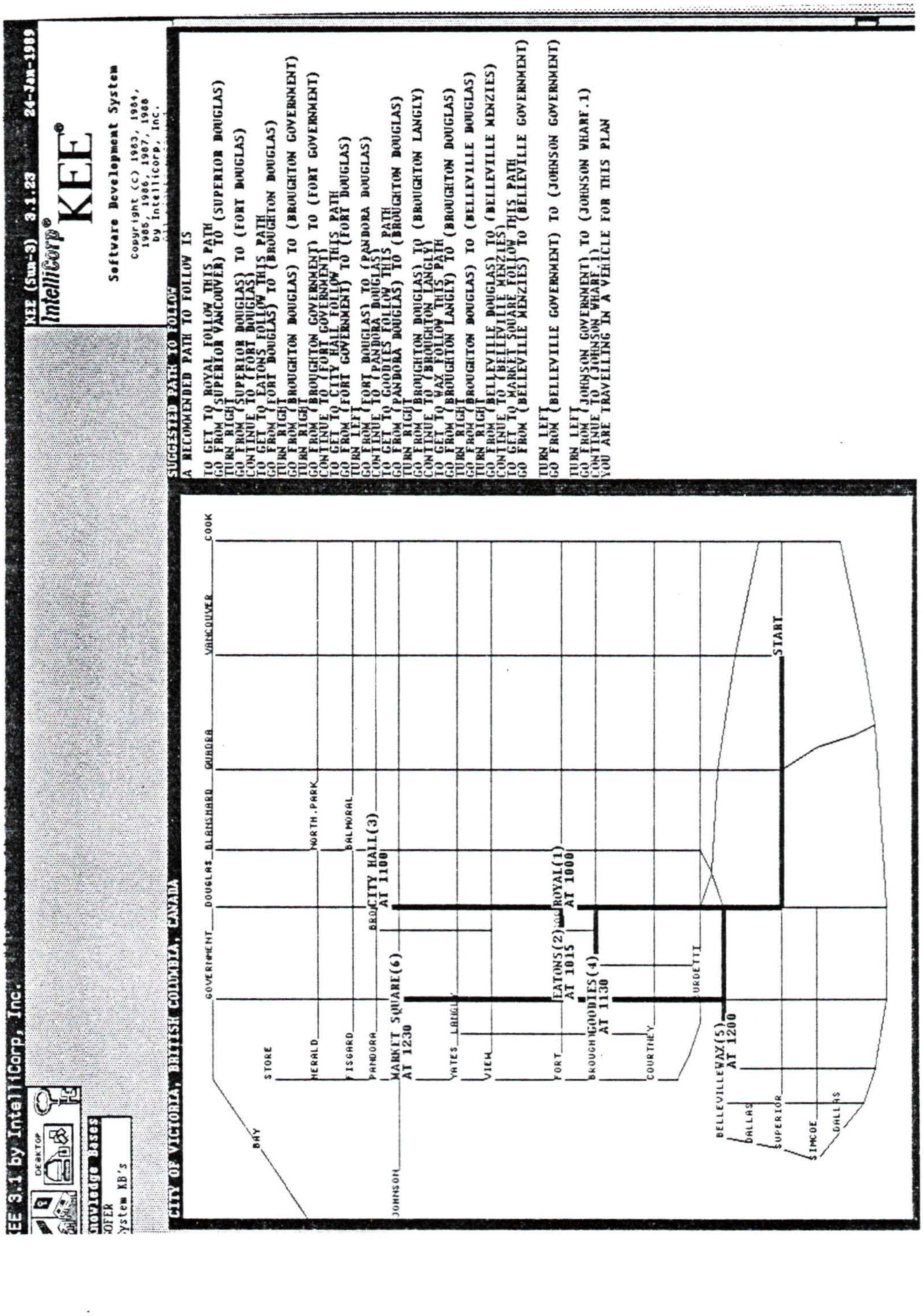

Figure A-85 Protocol 7 Test 3b

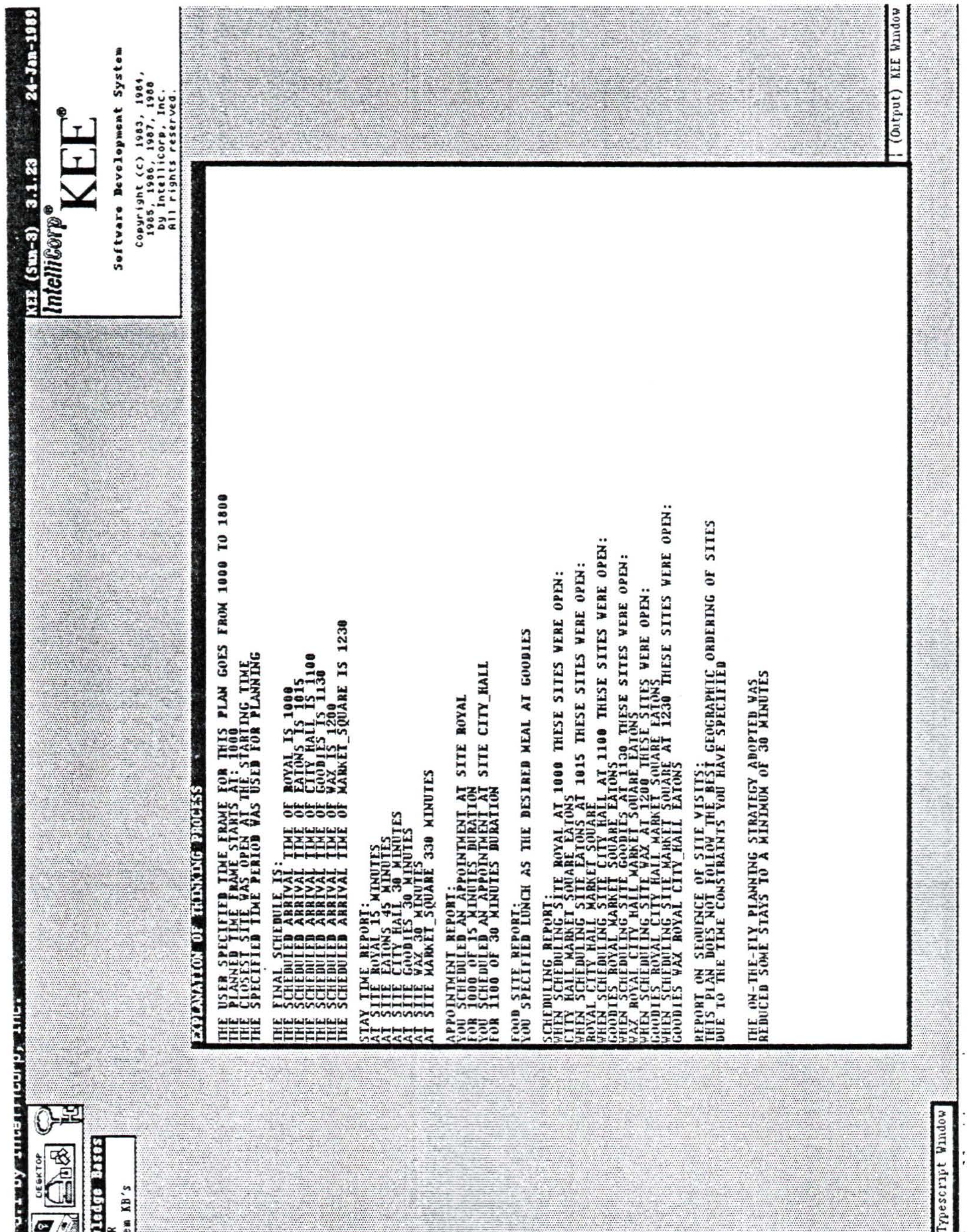

Figure A-86 Protocol 7 Test 3b

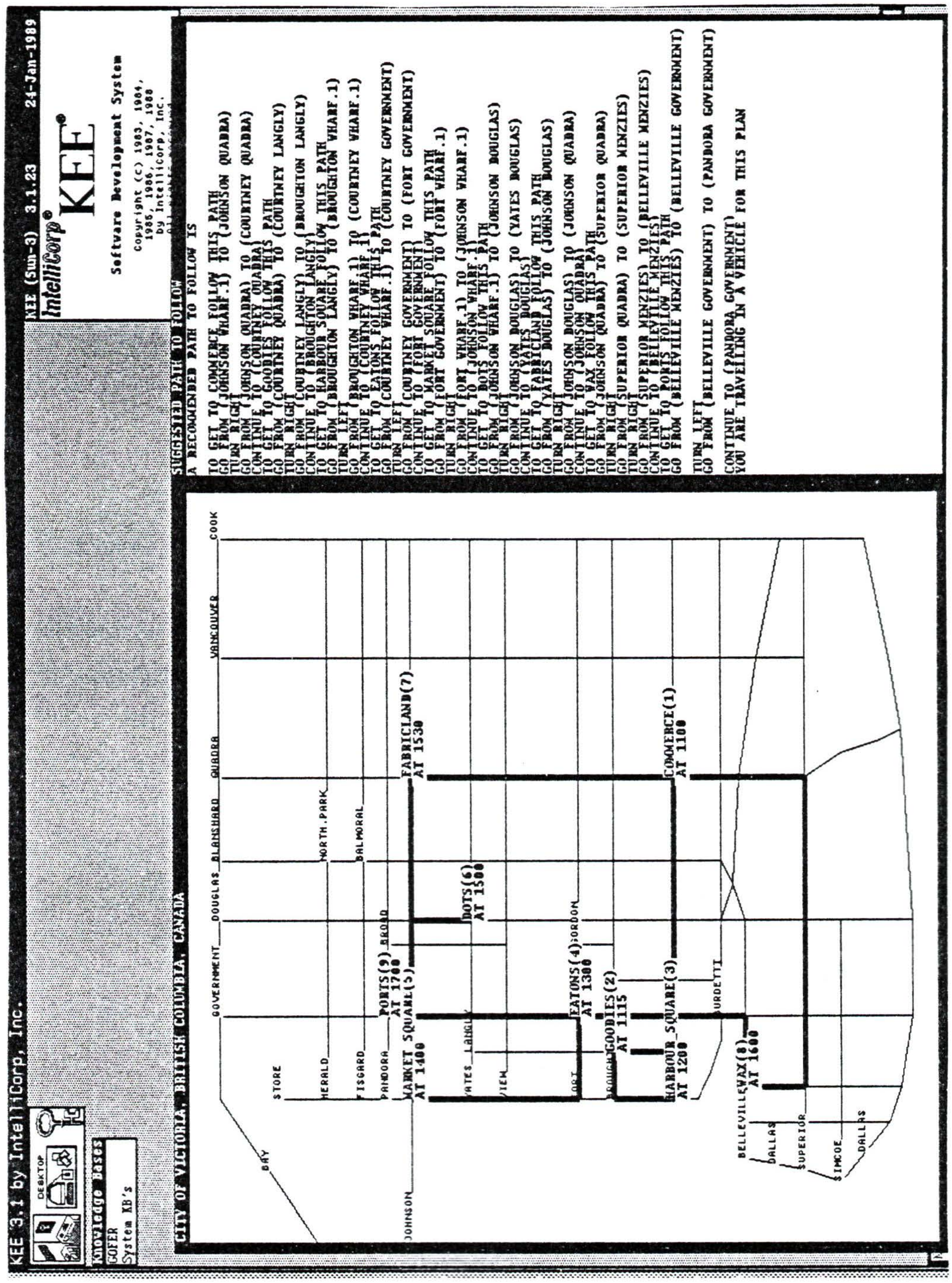

Figure A-87 Protocol 7 Test 3c

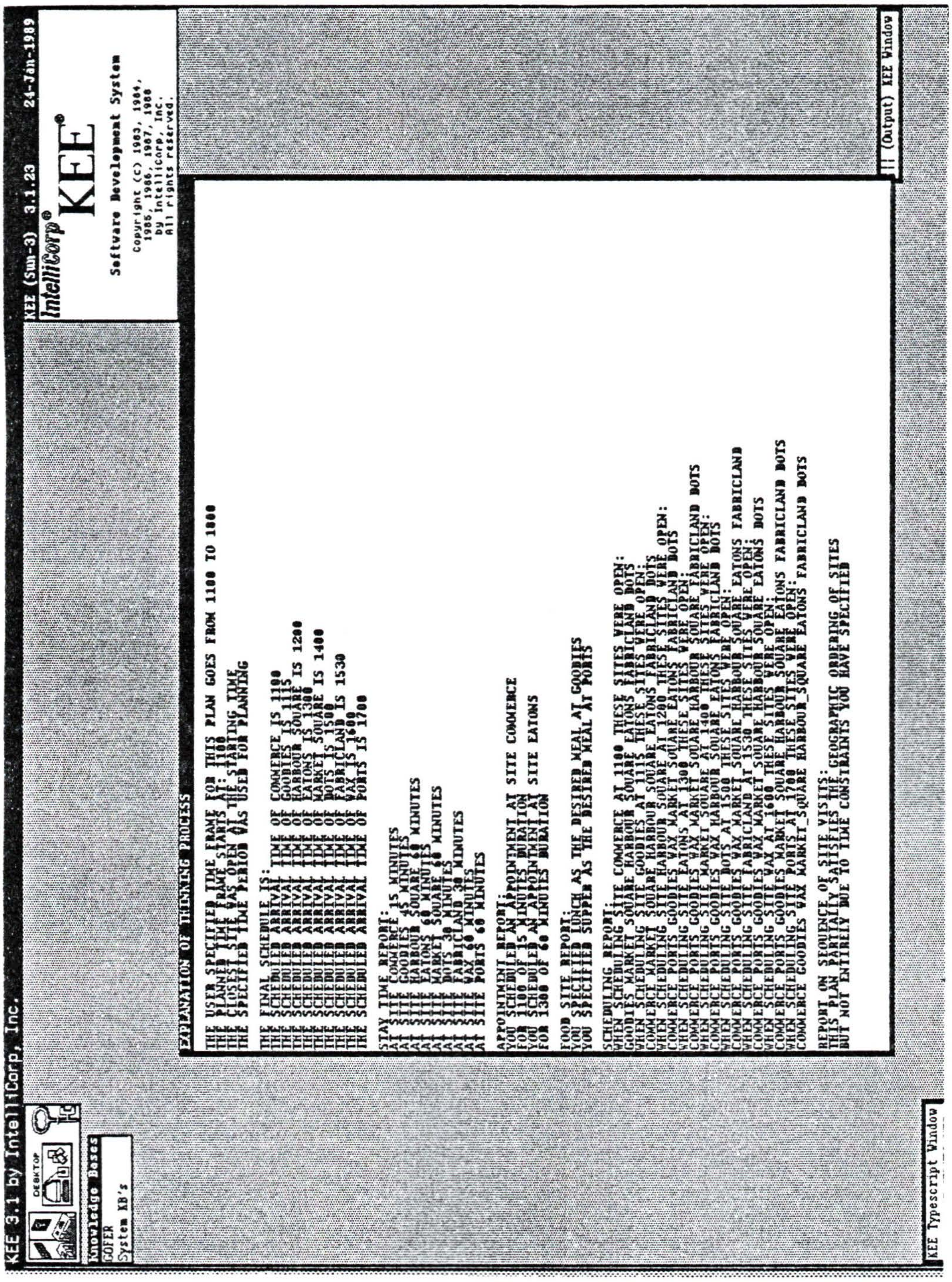

Figure A-88 Protocol 7 Test 3c

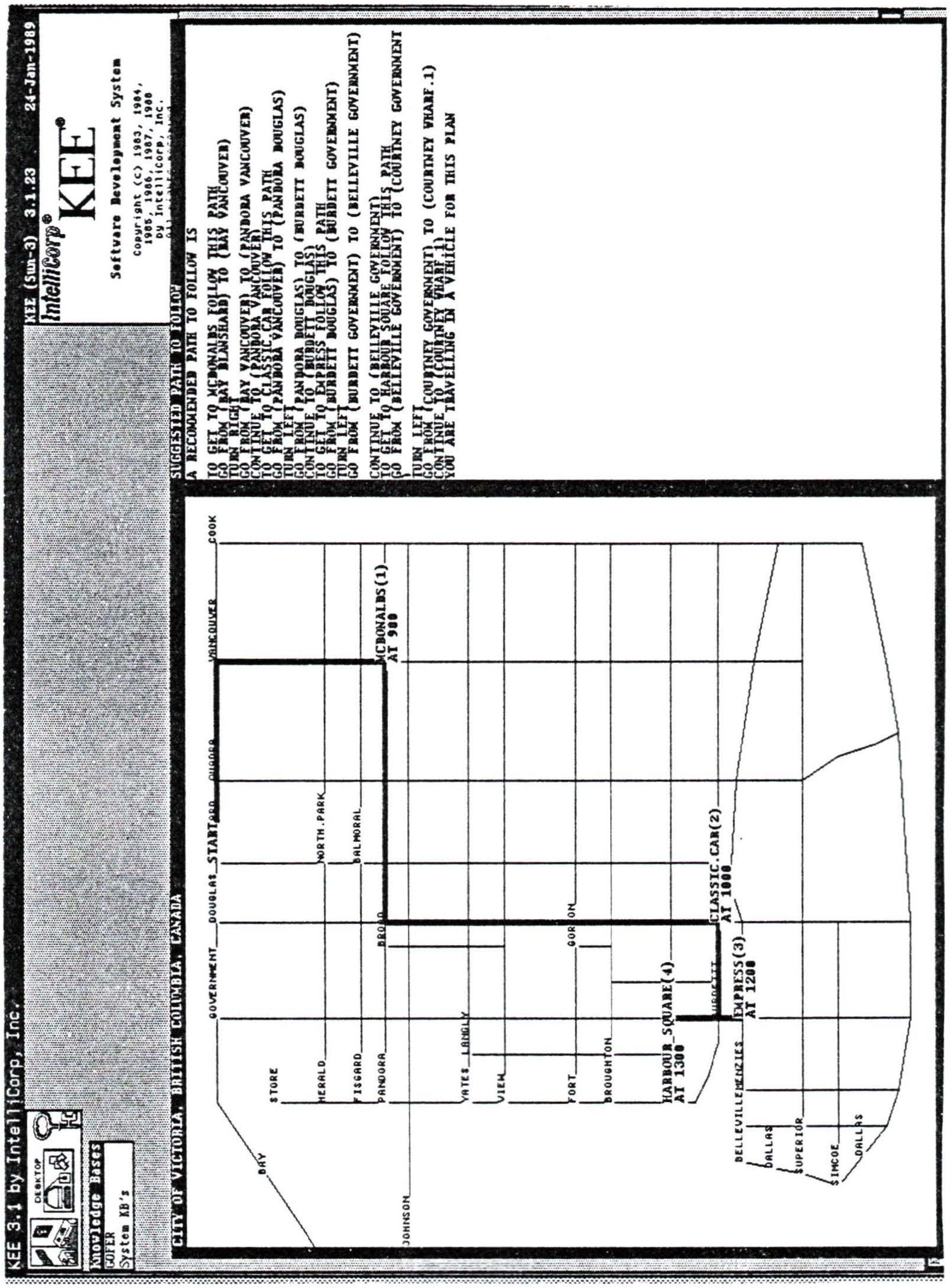

Figure A-89 Protocol 7 Test 4a

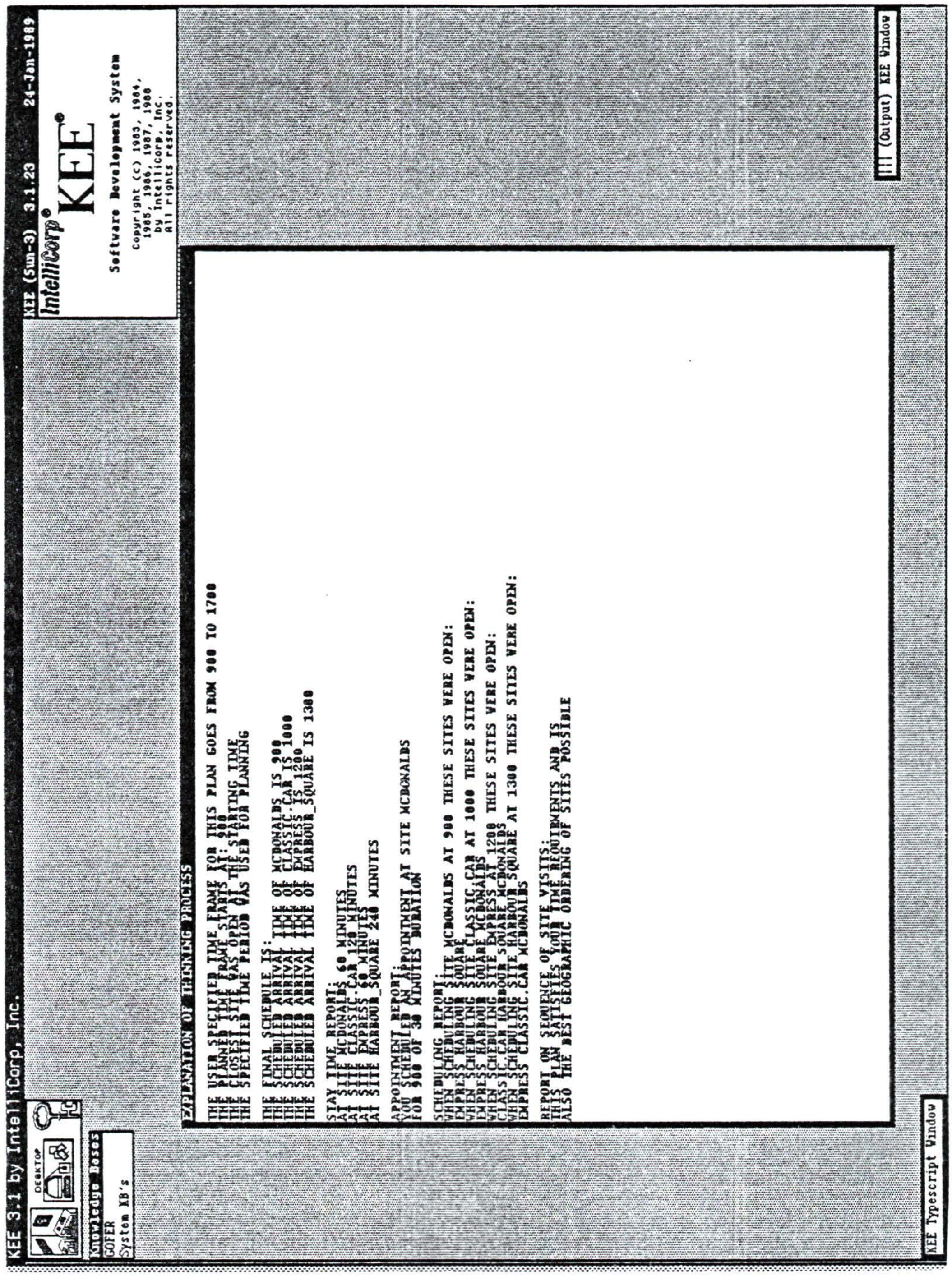

Figure A-90 Protocol 7 Test 4a

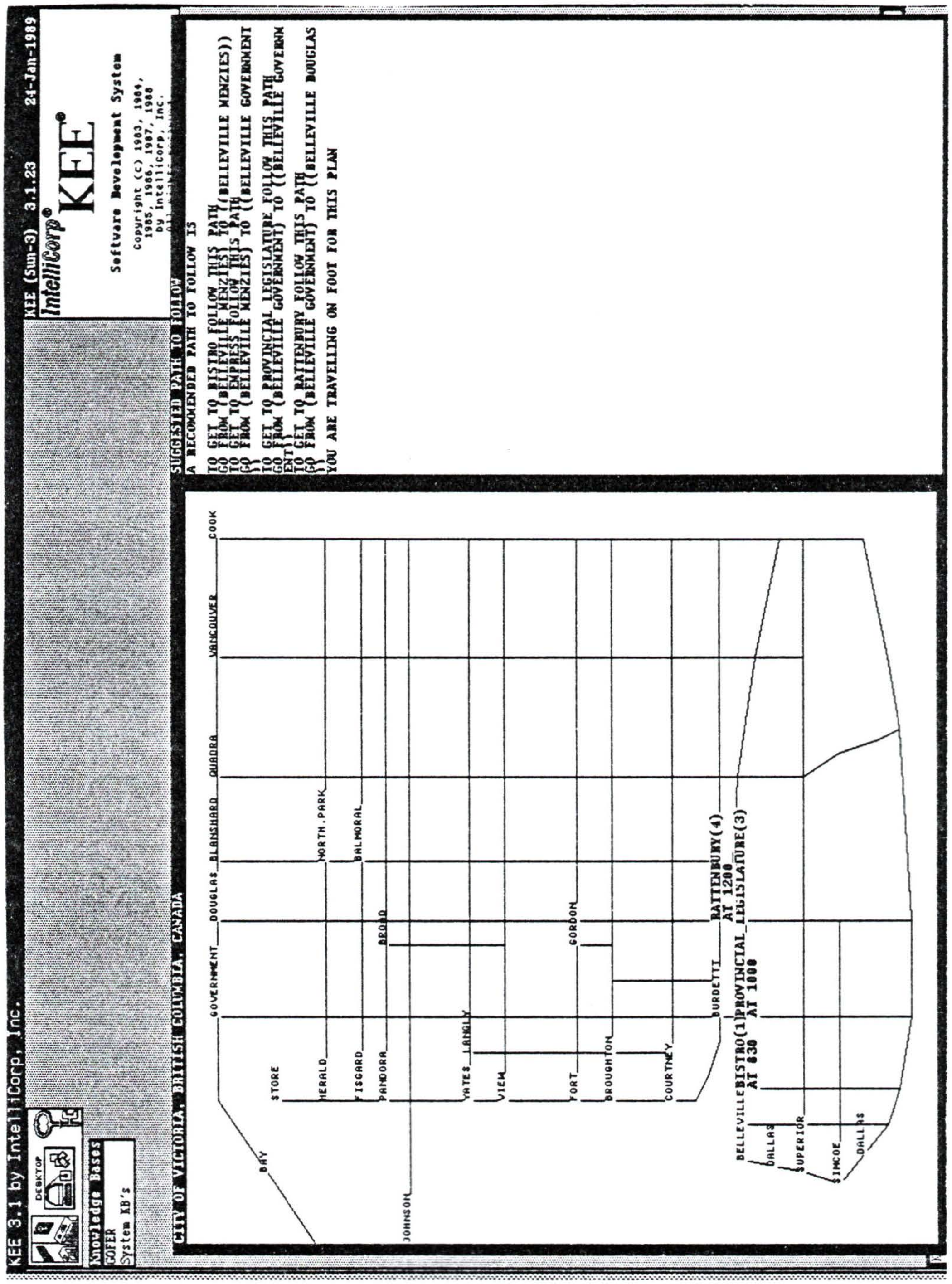

Figure A-91 Protocol 7 Test 4b

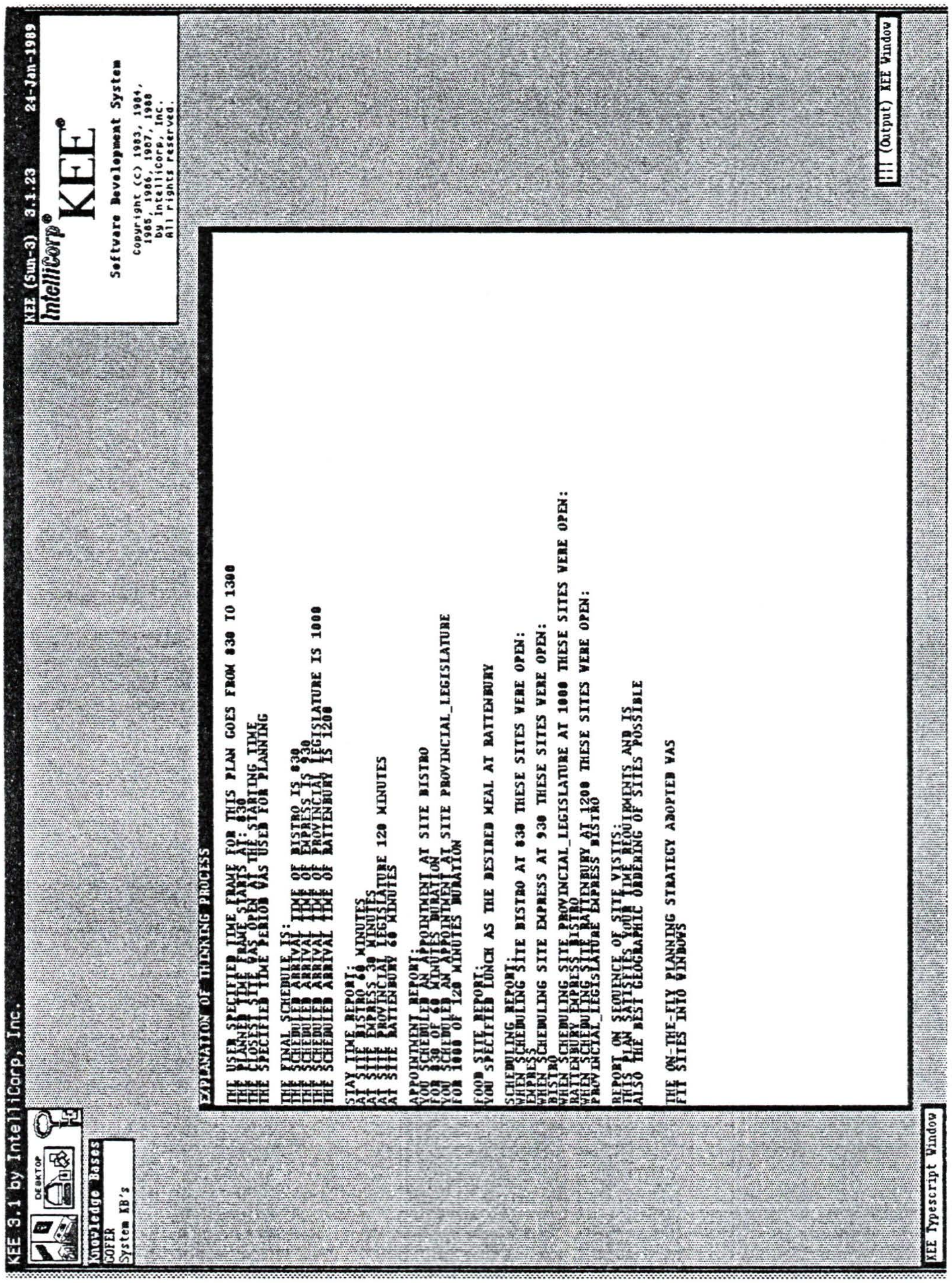

Figure A-92 Protocol 7 Test 4b

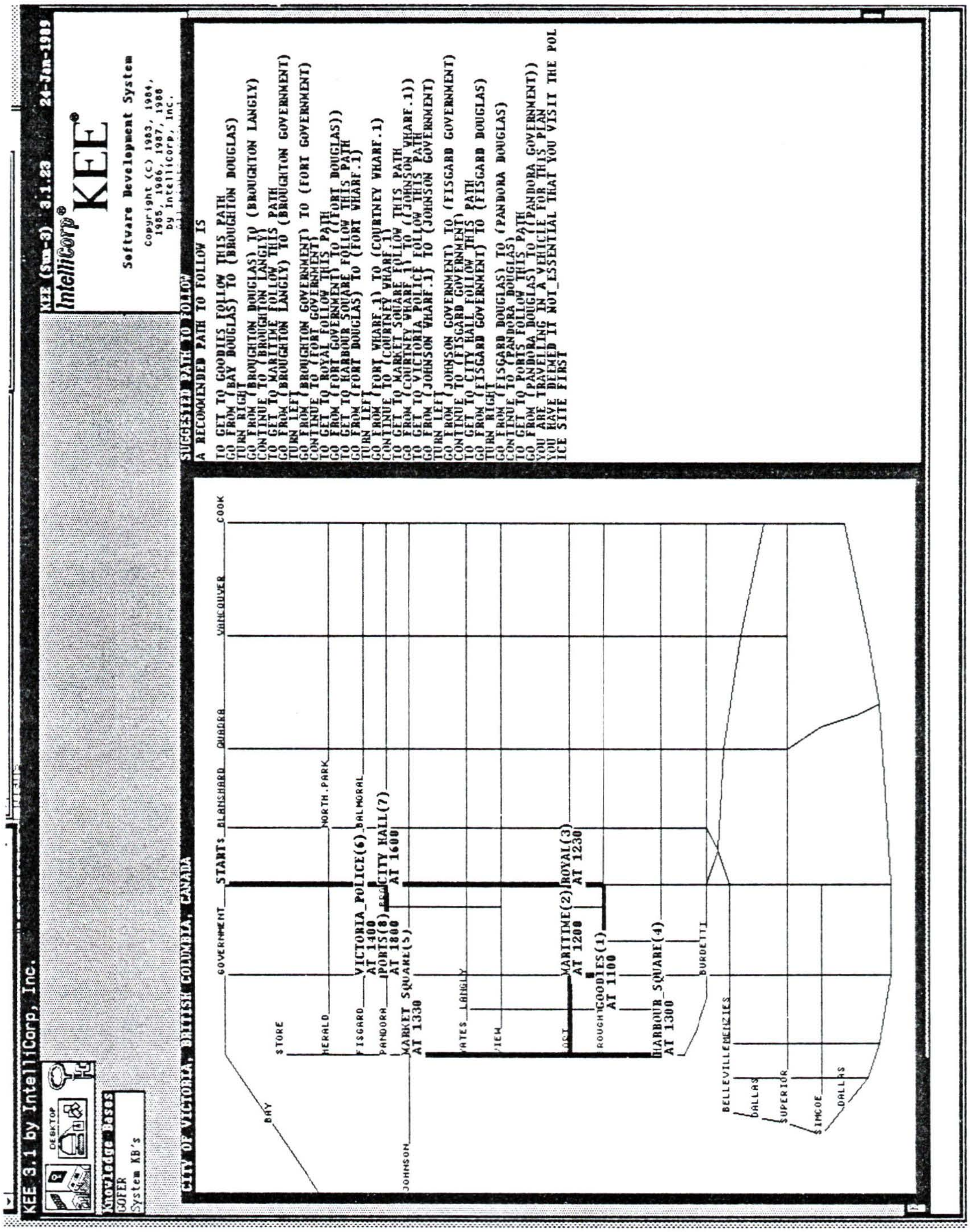

Figure A-93 Protocol 7 Test 4c

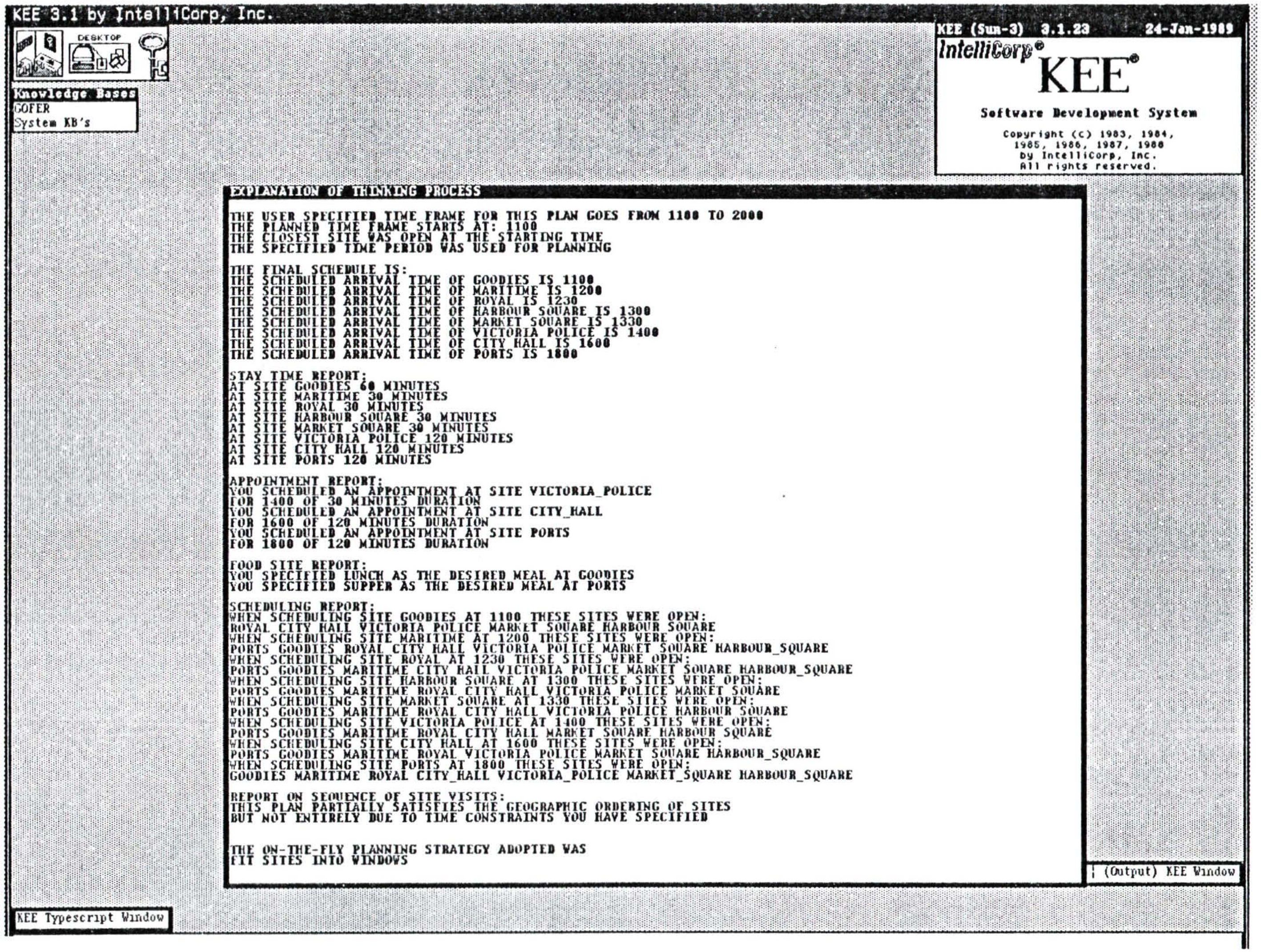

Figure A-94 Protocol 7 Test 4c

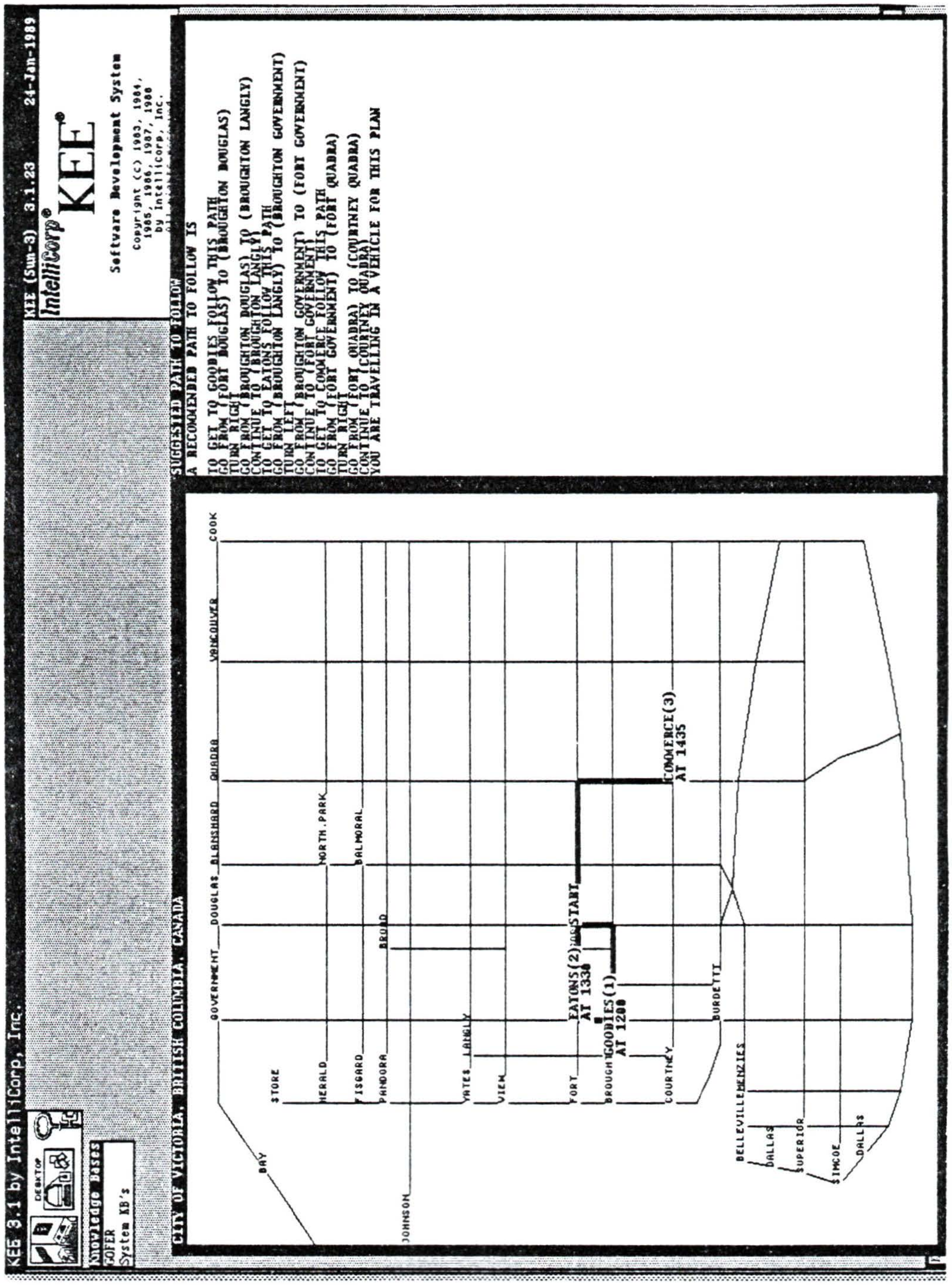

Figure A-95 Protocol 7 Test 5a

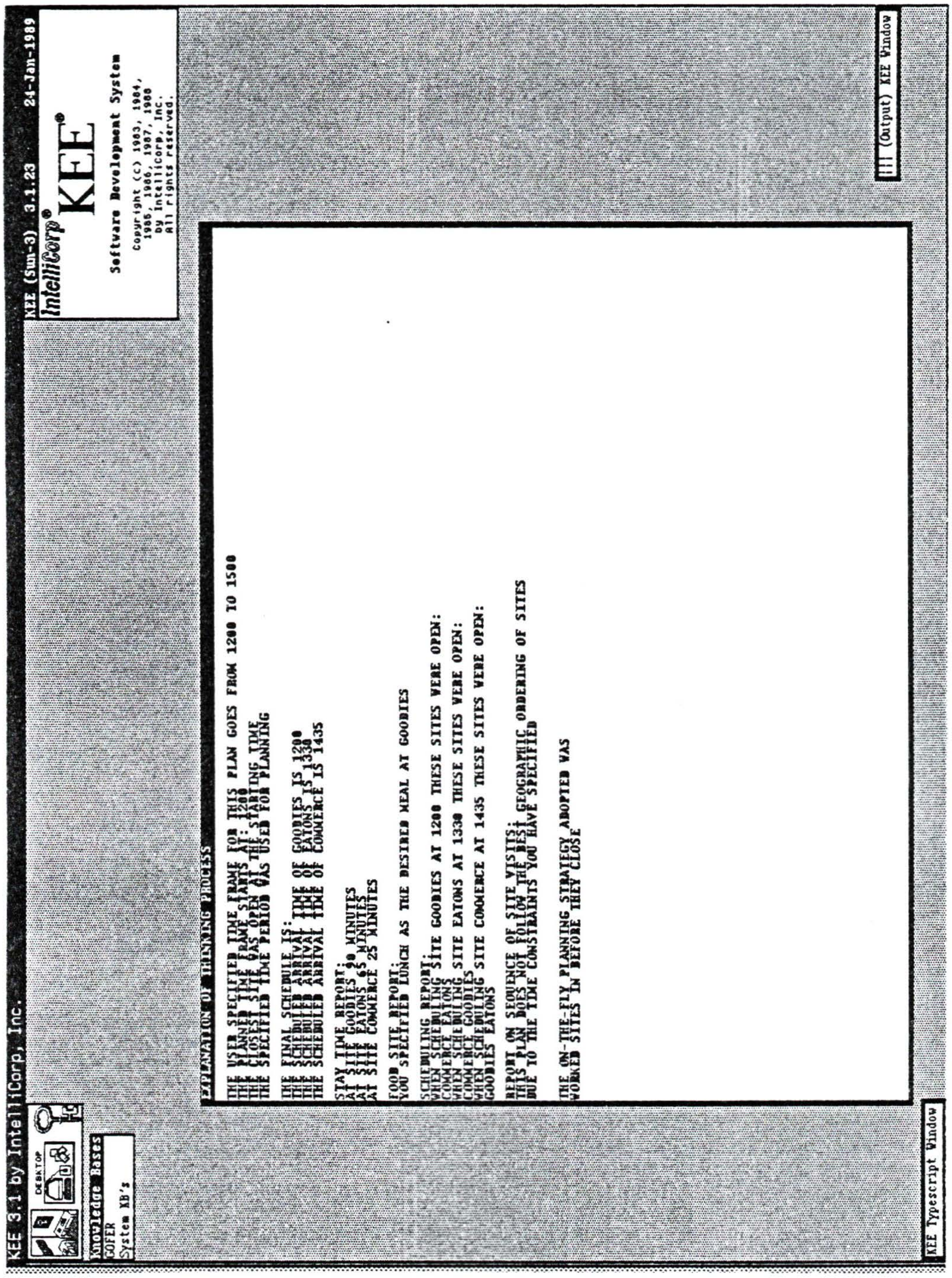

Figure A-96 Protocol 7 Test 5a

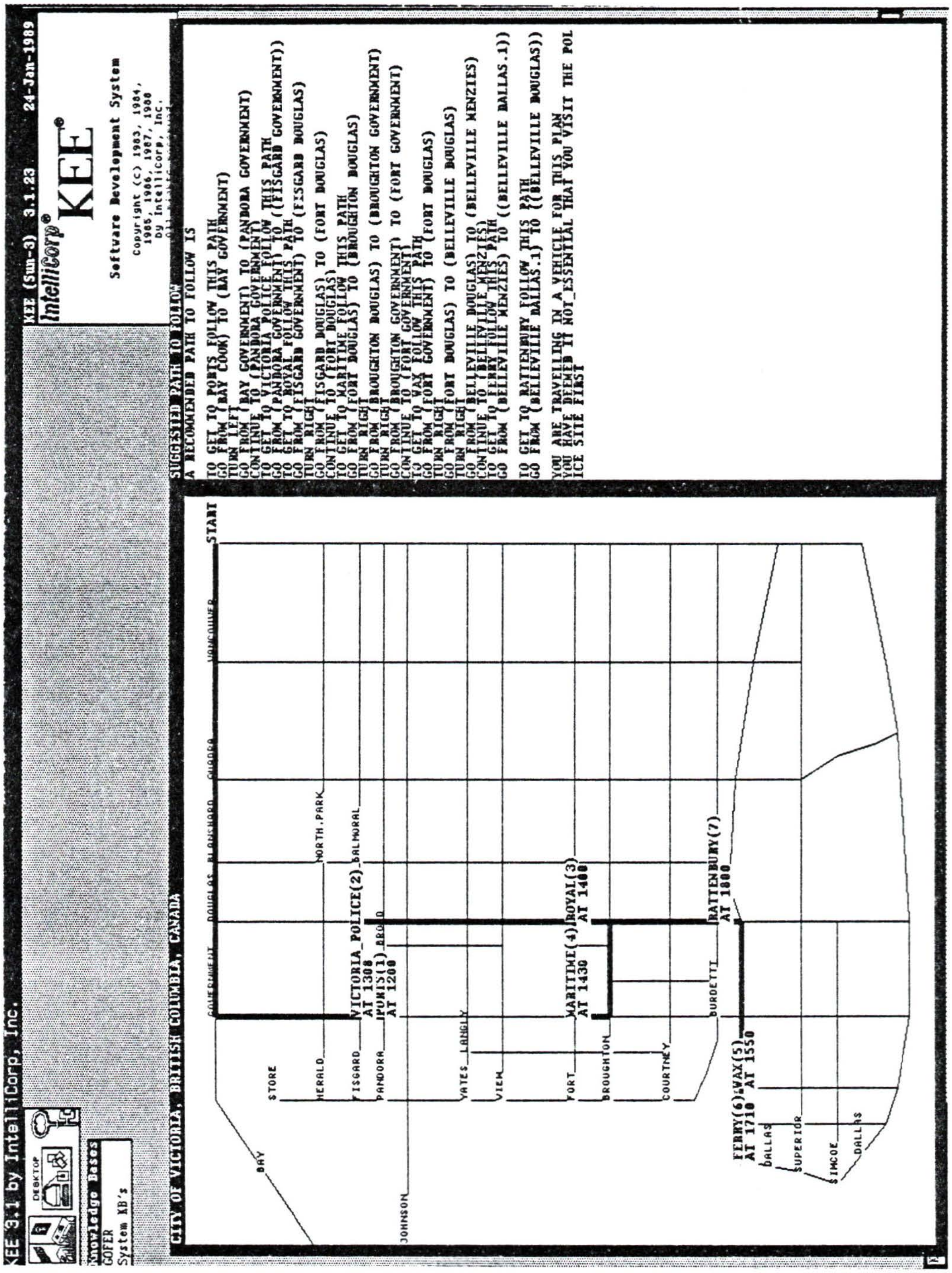

Figure A-97 Protocol 7 Test 5b

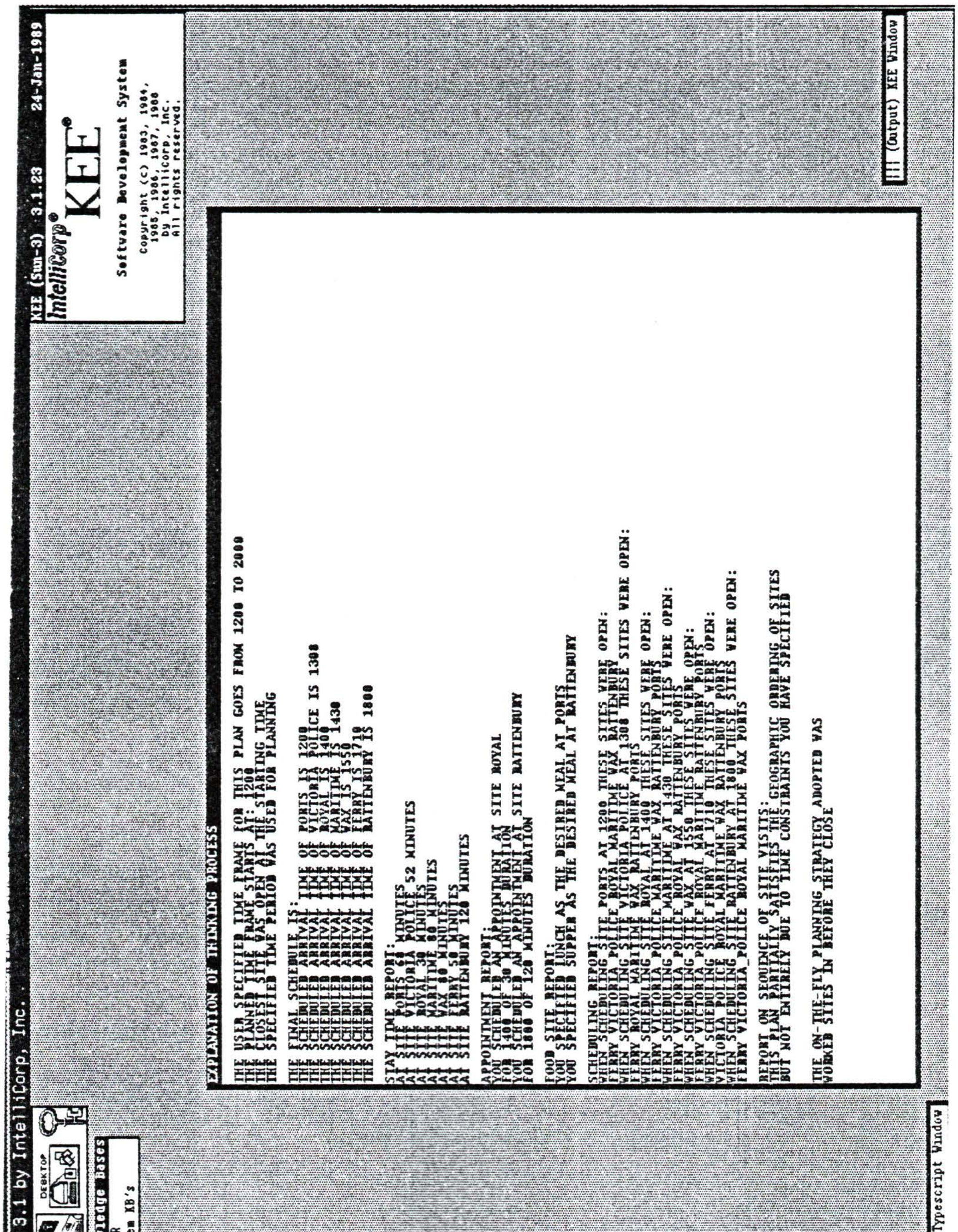

Figure A-98 Protocol 7 Test 5b

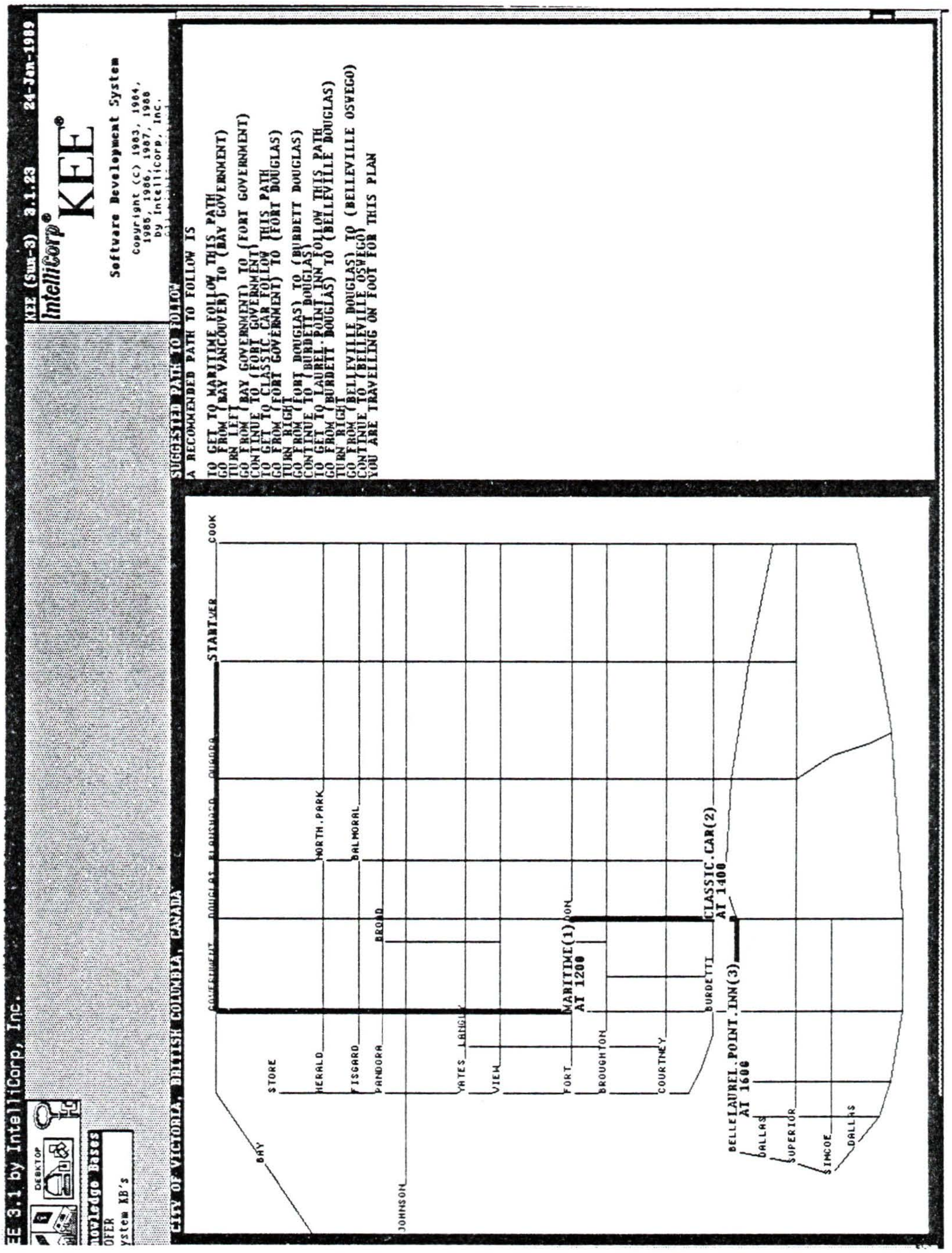

Figure A-99 Protocol 7 Test 6a

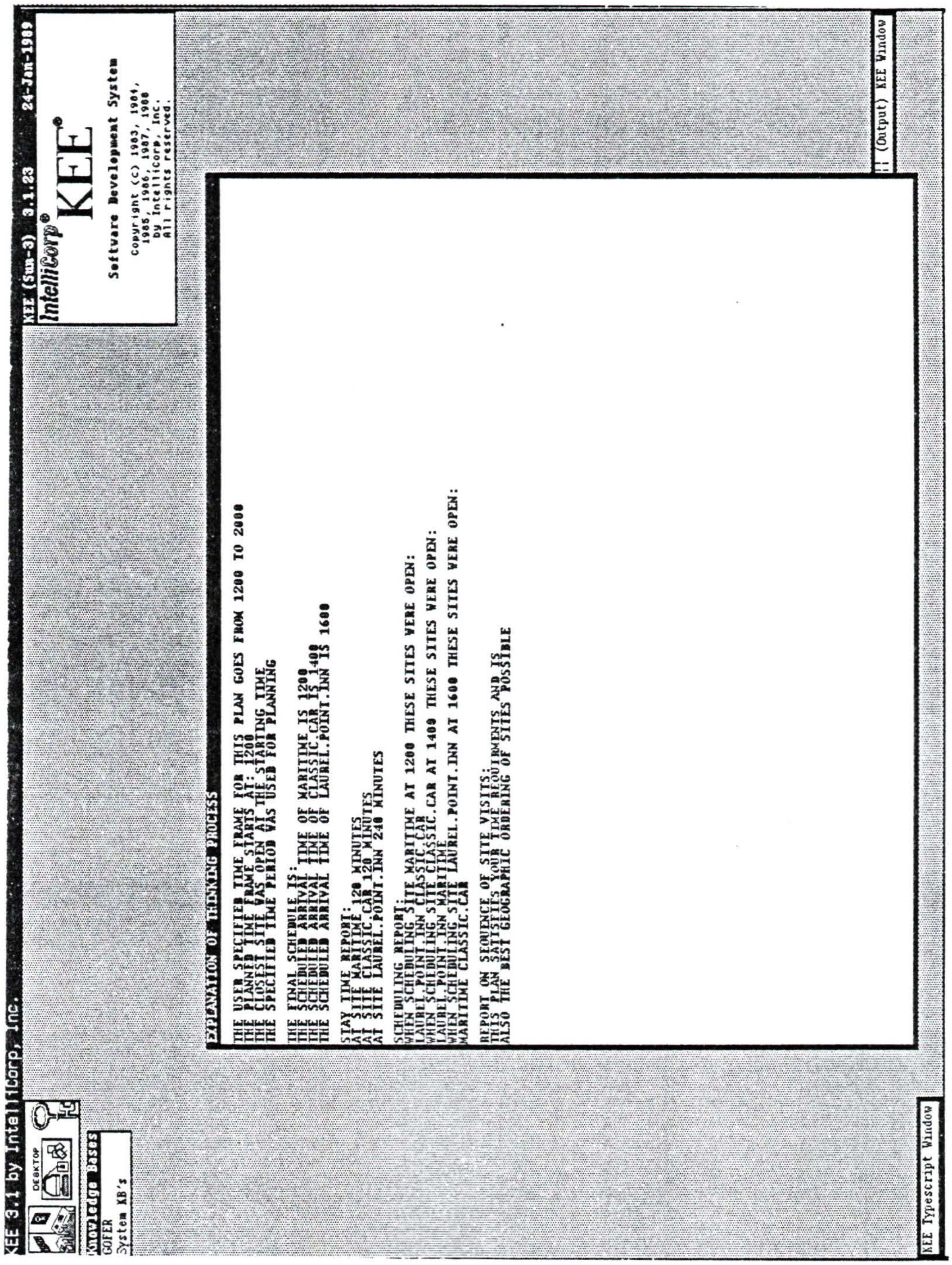

Figure A-100 Protocol 7 Test 6a

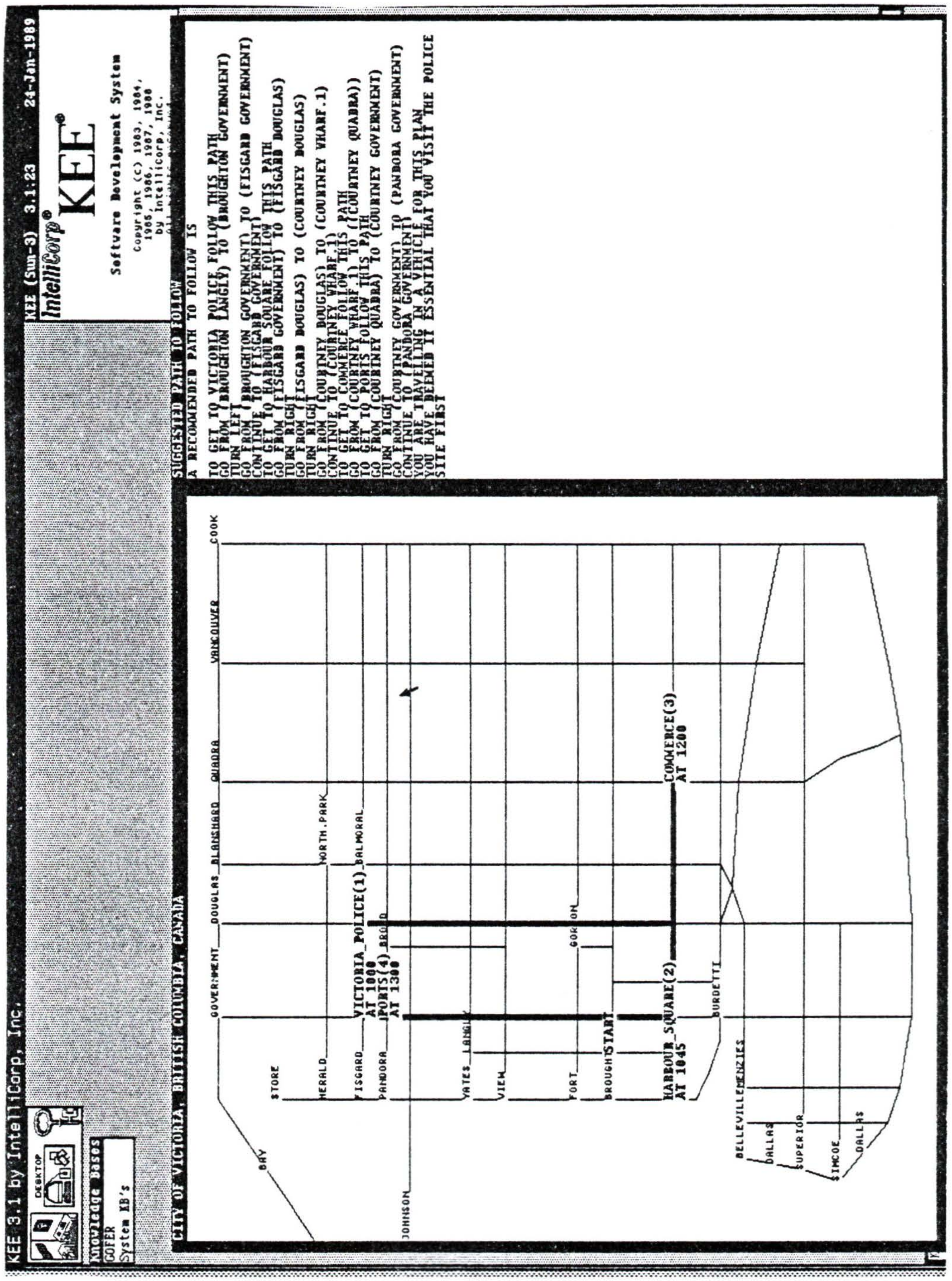

Figure A-101 Protocol 7 Test 6b

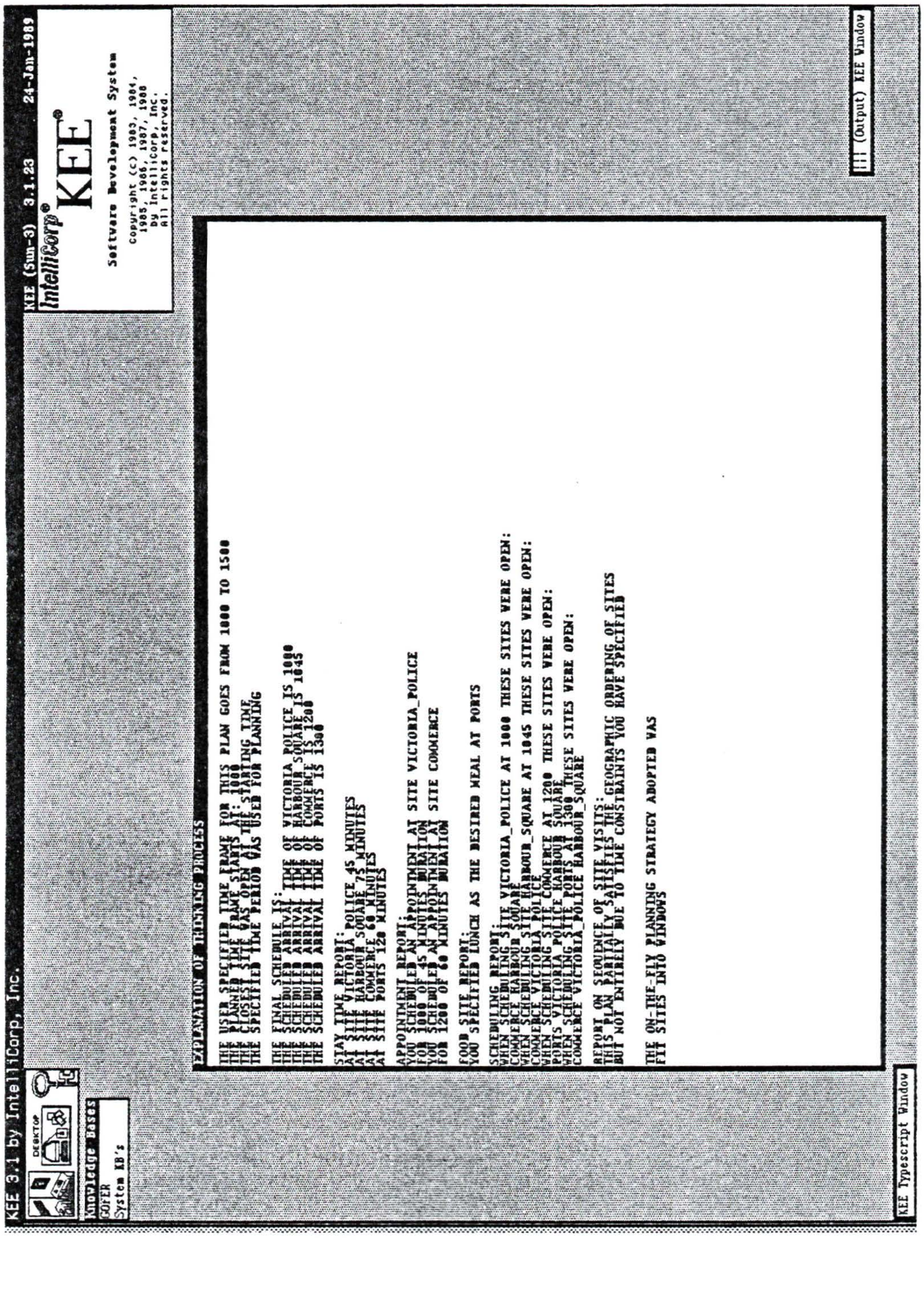

Figure A-102 Protocol 7 Test 6b

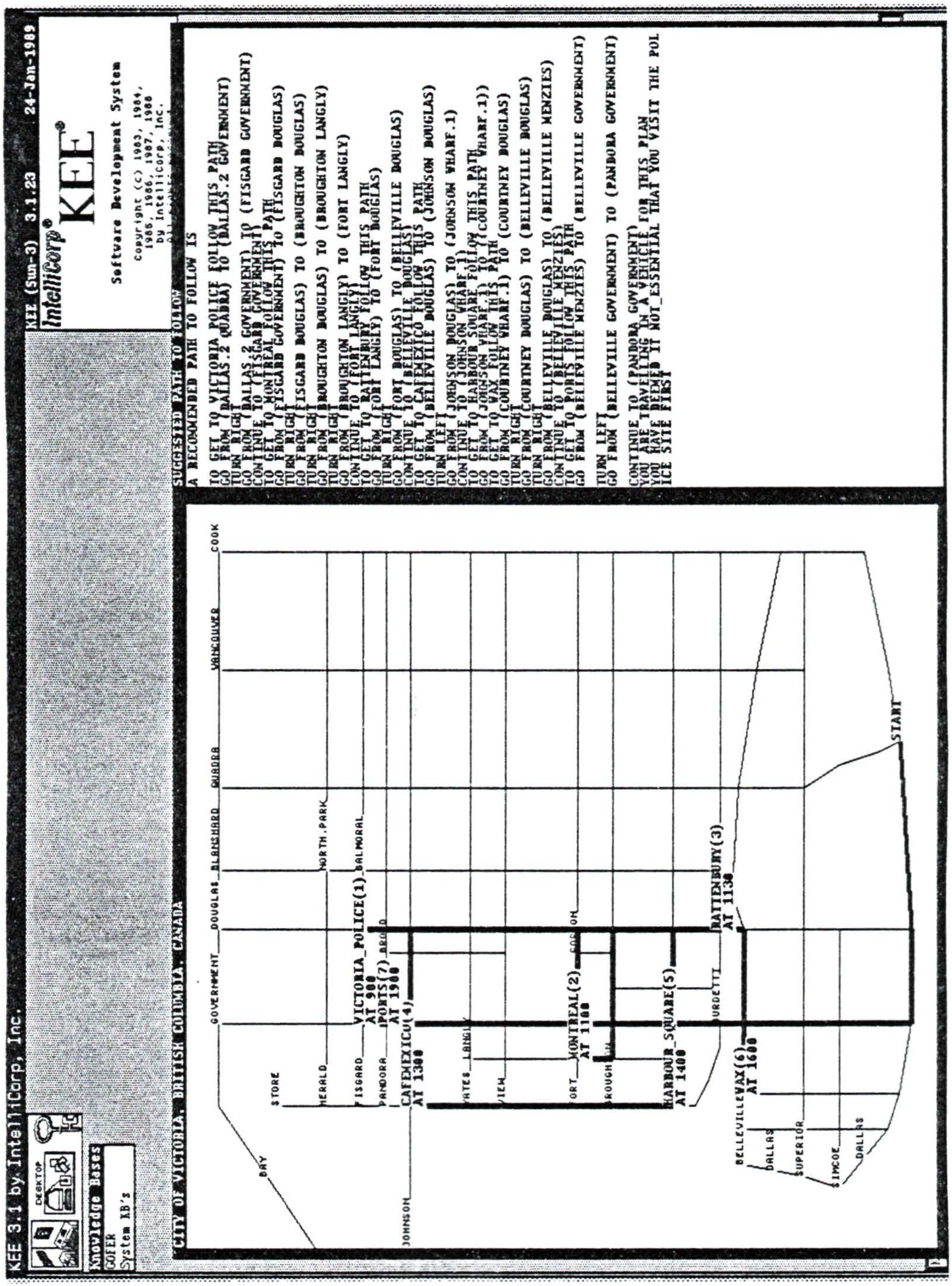

Figure A-103 Protocol 7 Test 6c

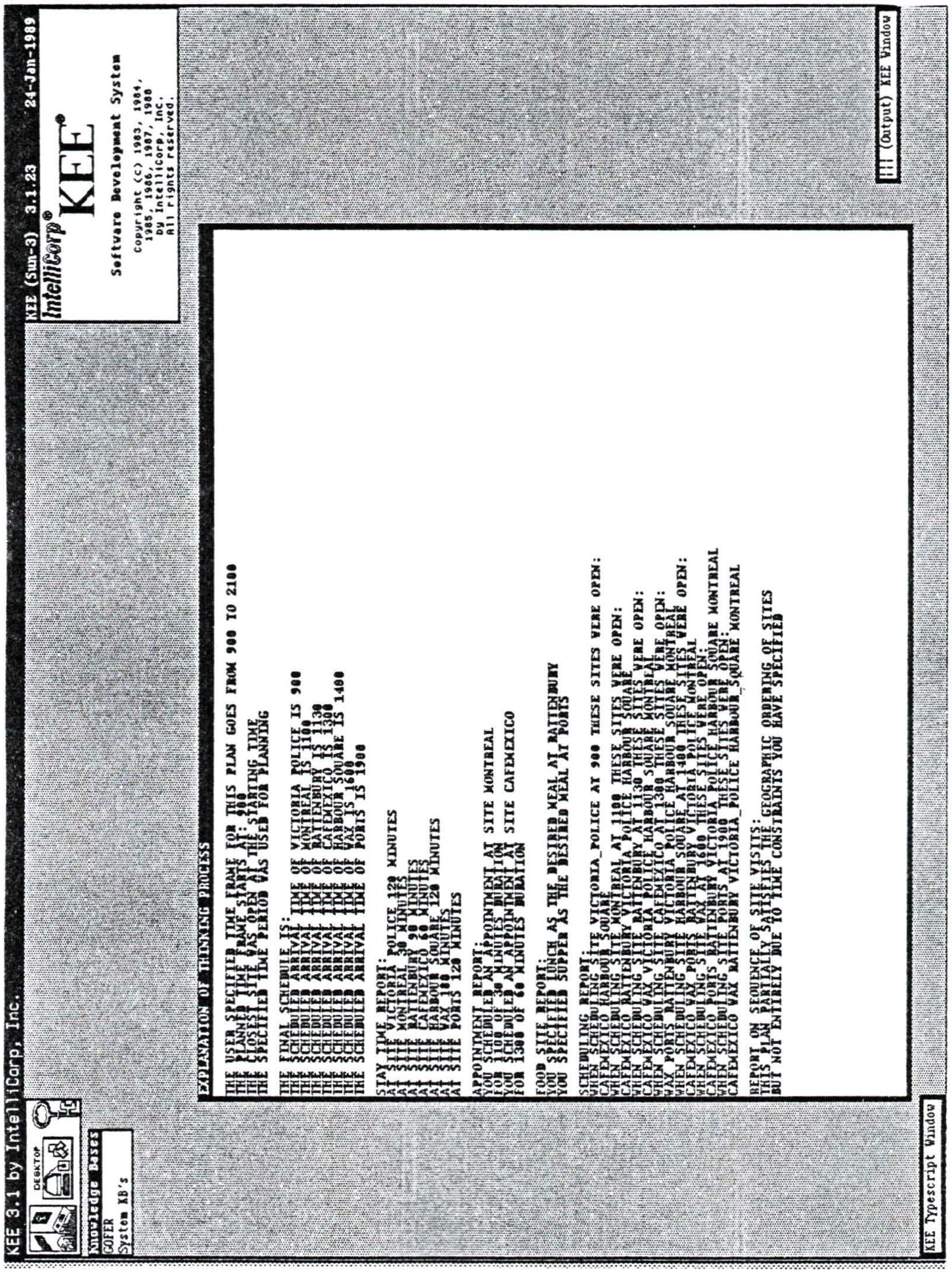

Figure A-104 Protocol 7 Test 6c

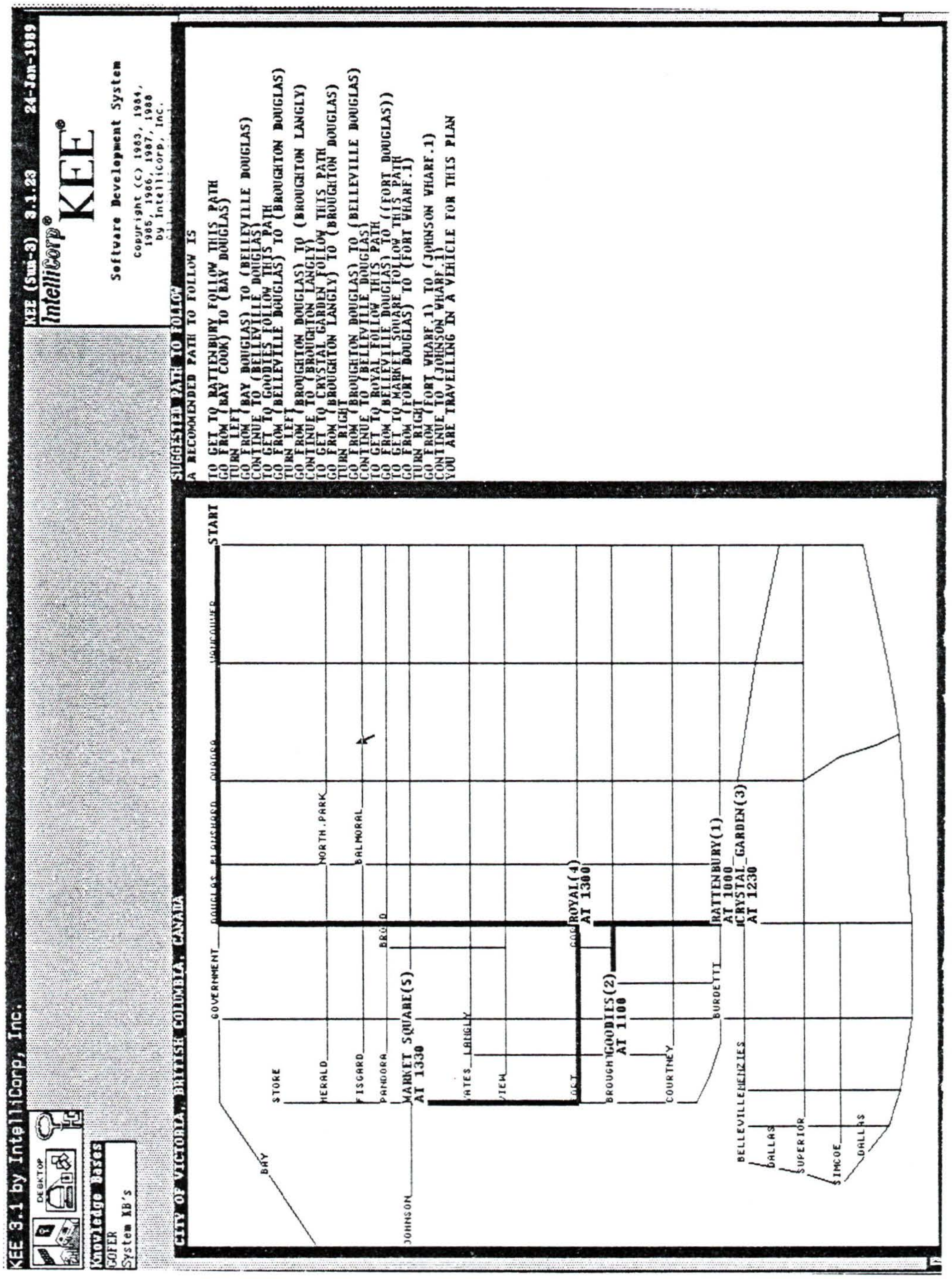

Figure A-105 Protocol 7 Test 7a

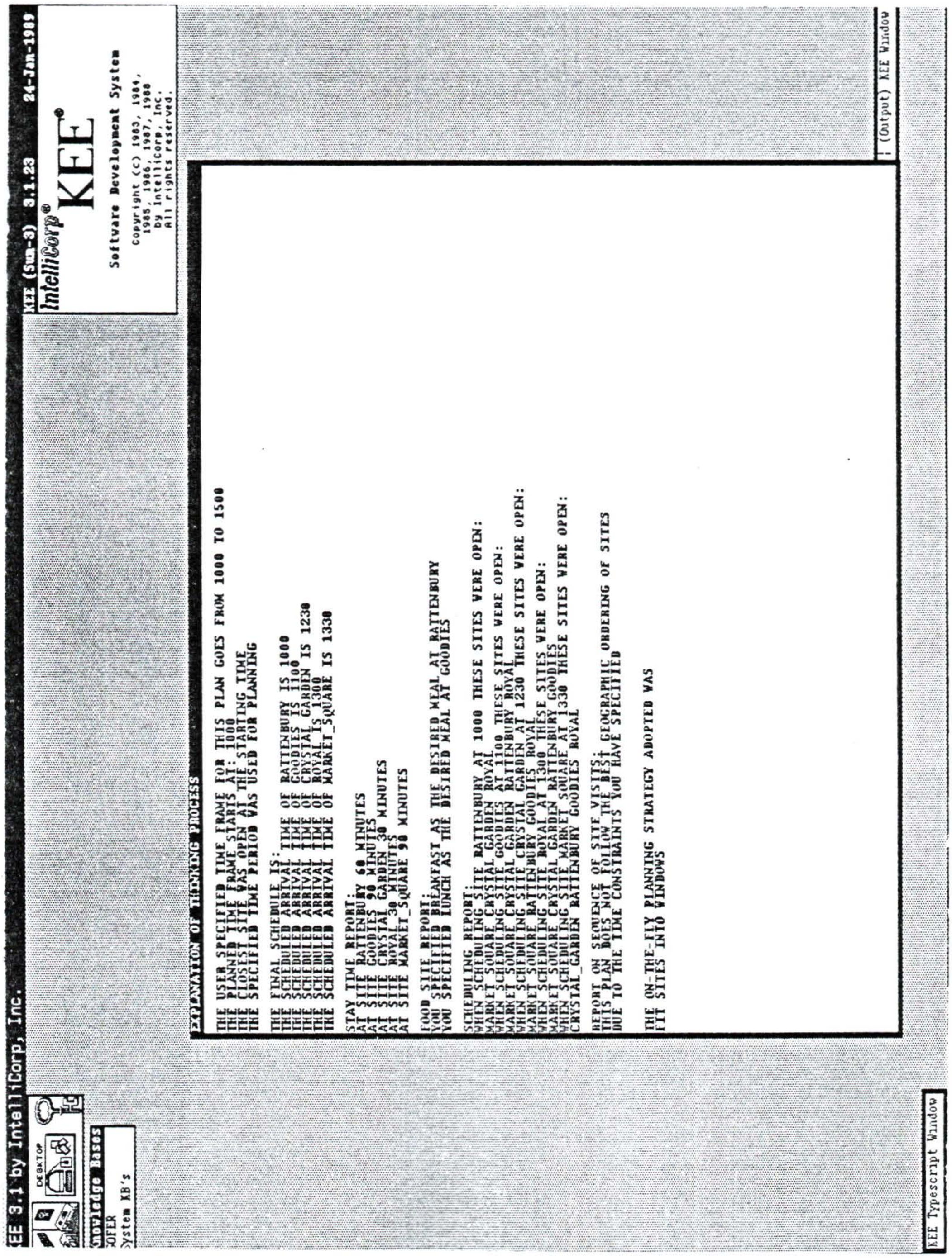

Figure A-106 Protocol 7 Test 7a

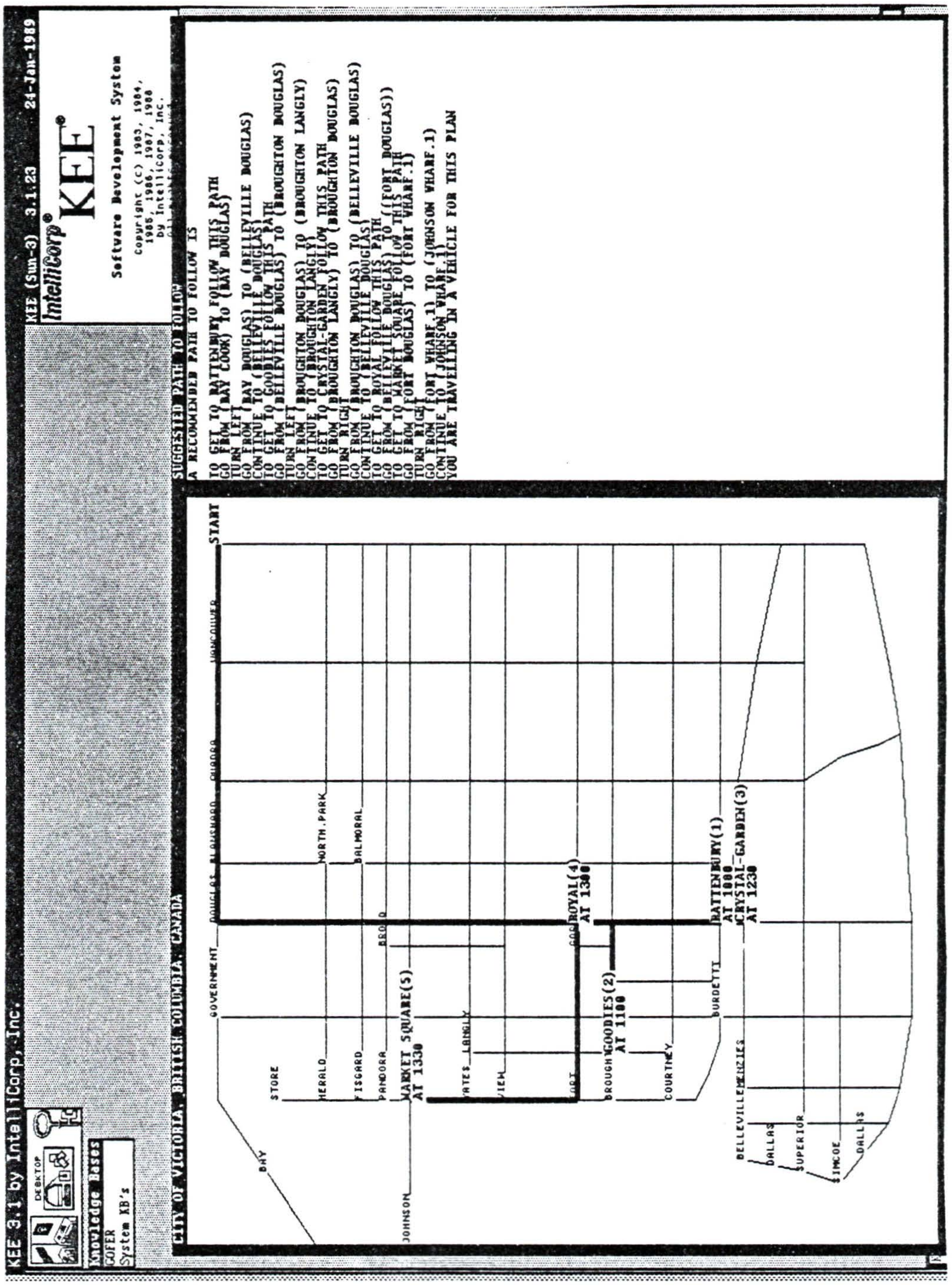

Figure A-107 Protocol 7 Test 7b

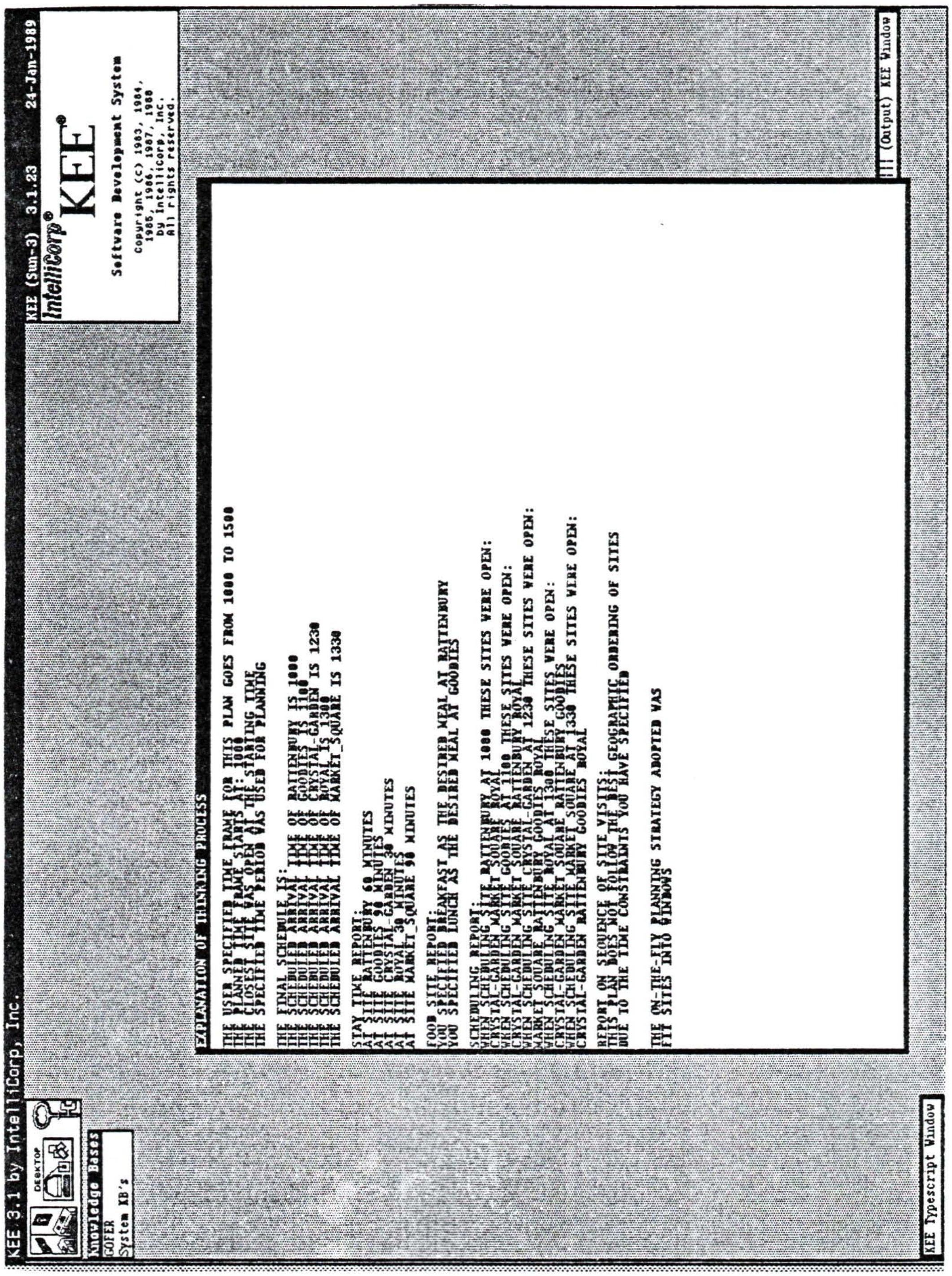

Figure A-108 Protocol 7 Test 7b

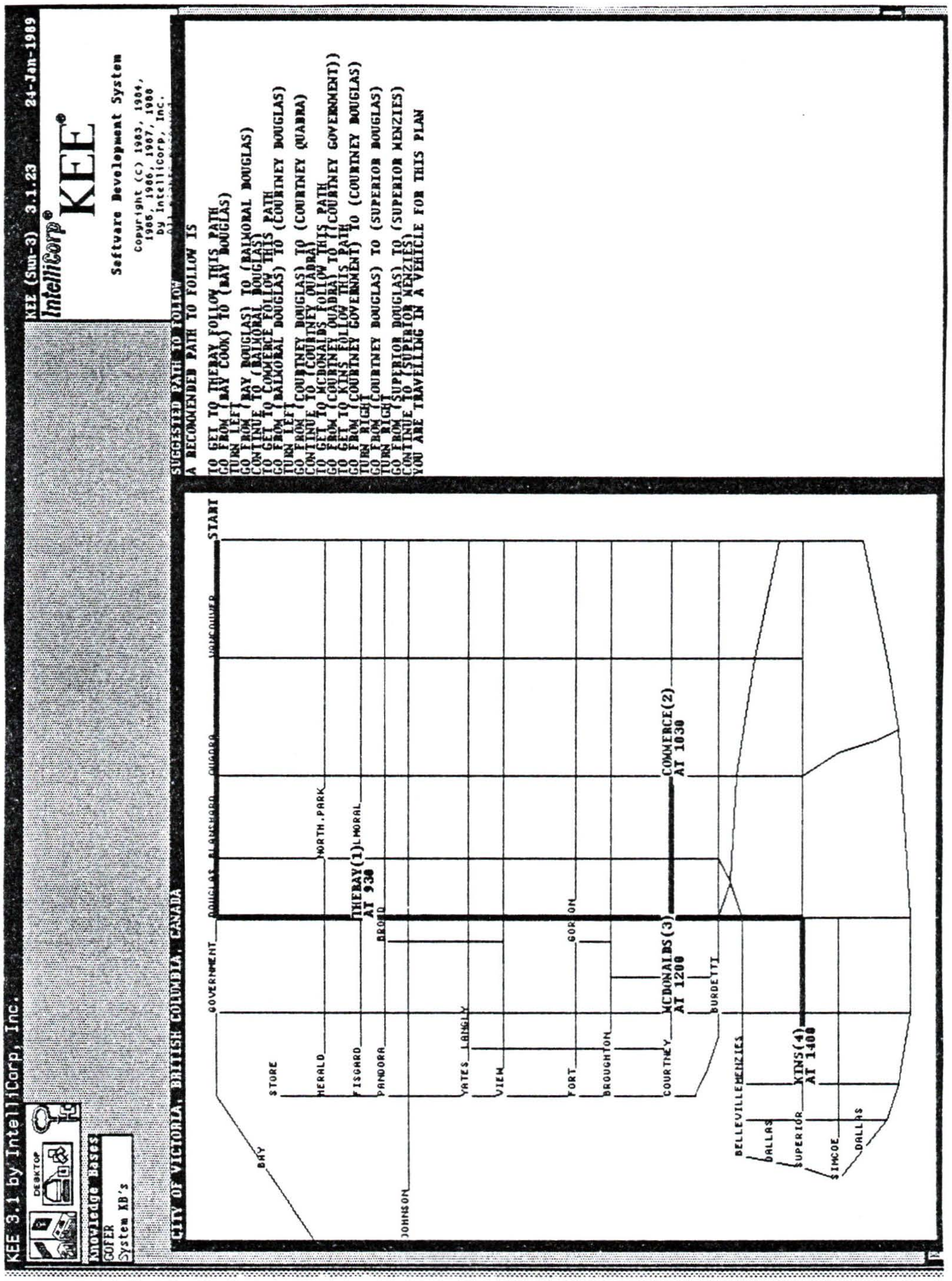

Figure A-109 Protocol 7 Test 7c

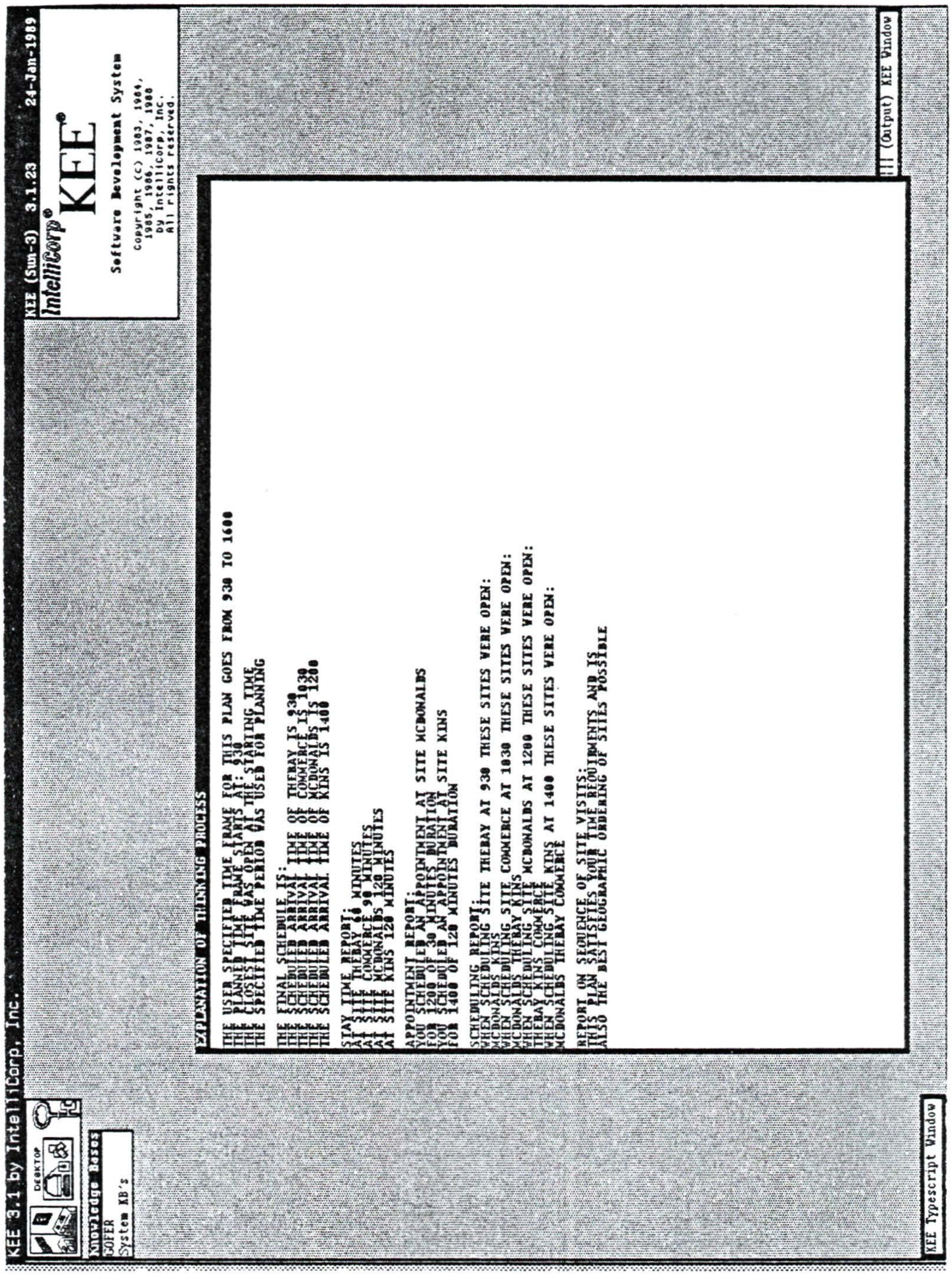

Figure A-110 Protocol 7 Test 7c

VITA

Surname: Leroux	Given Names: Joseph Georges Paul
Place of Birth: Ste-Foy, PQ	Date of Birth: 29 December 1958

Educational Institutions Attended:

University of Victoria	1988 to 1990
Royal Roads Military College	1978 to 1980
College Militaire Royal de St. Jean	1975 to 1978

Degrees Awarded:

B.Sc.	Royal Roads Military College	1980

Honours and Awards:

Governor General's Gold Medal, Royal Roads Military College	1980

Publications:

"An AI-Assisted Geographical Problem Solver", Canadian Conference on Electrical and Computer Engineering, Montreal, PQ, September 17-20, 1989, pp 134-137.

"An AI-Assisted Geographical Problem Solver", Proceedings of IEEE Section Conference, Victoria, BC, 2 December, 1989.

"An AI-Assisted Geographical Planner", to be presented at the IEE First International Conference on Expert Planning Systems EPS 90, London, UK, June, 1990.

"Geographical Problem Solving: A Practical AI Implementation Experience", to be presented at ICCI 90 Conference, Niagra Falls, Ontario, May 26 1990.

PARTIAL COPYRIGHT LICENSE

Title of Thesis:

The Study of Human Cognitive Processing and Planning in the Solution of Geographic Problems using Artificial Intelligence Techniques

Author [redacted]

(Signature)

PAUL LEROUX

(Name in Block Letters)

17 APR 90

(Date)