Development of a Public Transit Information System Using GIS and ITS Technologies

Submitted by Sarah J. Riley, B.Eng.
Carleton University
Ottawa, Ontario, 2002

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Applied Science

Department of Civil and Environmental Engineering
Carleton University
Ottawa, Ontario

The Master of Applied Science in Civil and Environmental Engineering Program is a joint program with the University of Ottawa, administered by the Ottawa-Carleton Institute for Civil Engineering.

Acceptance Form

The undersigned hereby recommended to the Faculty of Graduate Studies and Research acceptance of the thesis:

Development of a Public Transit Information System Using GIS and ITS Technologies

Submitted by:
Sarah J. Riley, B.Eng.

in partial fulfillment of the requirements for the degree of Master of Applied Science

.....

Acting Chair, Department of Civil and Environmental Engineering

A.O. Abd El Halim

A. M. Khan
.....

Thesis Supervisor

A.M. Khan

Carleton University
July 2002

Abstract

In order to attract users to public transit, assist them in planning, and provide other useful information, most transit agencies use a telephone system and/or website that lists bus routes and schedules. But in order to determine an optimum route, an email or a phone call is required. This process is slow and tedious; therefore, an improved method is required to determine the optimal route for an individual, based on an origin and destination of travel.

To achieve this, data from OC Transpo and the City of Ottawa were used to develop a transit information system, which enables the identification of an optimum route for users. In solving these route problems, standard algorithms were modified in order to reflect the constraints of transit vehicles to adhere to fixed routes and schedules. Additionally, new methodology is intended to address the uncertainties in estimating travel time.

To facilitate this process, a Geographic Information System (GIS) software package called ArcView was used. This software utilizes the location of bus stops, scheduling, and route layouts in order to optimize a choice route. Bayesian statistical decision theory was also incorporated into the system to represent the possibilities of deficiencies in schedule adherence. In running the transit information system, the user enters start and end locations, the day, and time of travel. The output is given in the form of a map and trip itinerary.

The developed transit information system entails the latest advances in GIS technologies available, the Bayesian statistical decision theory addressing the uncertainty in schedule adherence and travel time, and accounting for the complexities of transit operation by the system.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. A. M. Khan, Carleton University, for his advice and support.

The City of Ottawa and OC Transpo provided transportation data and transit information. For this, I would like to thank two individuals, who for their invaluable time enabled me to attain the necessary schedule and statistical information. In particular, I would like to thank Jim Summers and Brian Barclay for their technical knowledge and feedback.

I would also like to thank my family and friends for their continued support and encouragement in completing this thesis. Finally, I would like to offer a sincere appreciation to my husband, Kevin, whose support and understanding has helped me pursue my goals.

Table of Contents

Pages

List of Figures

Pages

List of Tables

Page

List of Appendices

List of Symbols and Acronyms

AMS: Automated Mapping Systems

APC: Automatic Passenger Counter

APTS: Advanced Public Transportation Systems

ARTS: Automatic Radio Transmission Systems

ATIS: Advanced Traveller Information Systems

ATP: Automatic Trip Planning

AVL: Automatic Vehicle Location

AVLC: Automatic Vehicle Location and Control system

AVM/C: Advanced Vehicle Monitoring & Communication System

BDS: Bus Dispatch System

BIS: Bus Information System

CADD: Computer-Aided Drafting and Design Systems

CBD: Central Business District

CUMTD: Champaign-Urbana Mass Transit District

DGPS: Digital Global Positioning System

ESRI: Environmental Systems Research Institute, Inc.

FHWA: Federal Highway Administration

FTA: Federal Transit Administration

GPS: Global Positioning System

GUI: Graphical User Interface

HCM: Highway Capacity Manual

ITS: Intelligent Transportation Systems

IVHS: Intelligent Vehicle Highway Systems

IVR: Interactive Voice Response telephone systems

LACMTA: Los Angeles County Metropolitan Transportation Authority

MDI: Model Deployment Initiative

MDT: Mobile Data Terminals

O-D: Origin-Destination

O-OP: Object-Oriented Programming

OP: Official Plan

P(R1|T1): An example of Conditional Probability for T1

P1, P2, P3...Pn: Different probability conditions

PARIS: Passenger Routing and Information System

PCDs: Personal Communications Devices

PDA: Personal Data Assistant

R1: updated indication of early arrival; corresponds to T1

R2: updated indication of on-time arrival; corresponds to T2

R3: updated indication of late arrival; corresponds to T3

RCTIS: Regional Customer Telephone Information System

R-GRTA: Rochester-Genesee Regional Transportation Authority

SMS: Short Message Service

T1: early arrival at the destination

T2: on-time arrival at the destination

T3: late arrival at the destination

TATIS: Transit Advanced Traveller Information System

TMP: Transportation Master Plan

TRB: Transportation Research Board

Tri-Met: Tri-County Metropolitan Transportation District of Oregon

USDOT: United States Department of Transportation

UTA: Urban Transit Area

VMS: Vehicle Management System

WAP: Wireless Application Protocol

WSTA: Winston-Salem Transit Authority

WWW: World Wide Web

Chapter 1: Problem Identification and Methodology Framework

As a part of the development program for public transportation, and in order to improve the service and achieve customer satisfaction, new technologies are being introduced around the world. These technologies are helping with the increasing demand placed on the transportation networks. The ability to quickly respond to changing situations is essential to the success of transportation agencies. Questions such as where locations are, how to move around, how to quickly get to or get out of an area are faced by people of all walks of life on a daily basis.

The development of intelligent transportation systems (ITS) is bringing about a great change in the public transportation system. Traditional functions of transit services are undergoing important changes. Many transit agencies are considering, and some are implementing, new technologies to enhance their traditional services.

During the 1990's, Geographic Information Systems (GIS) has become one of the fastest growing software to be used within the field of transportation. One reason is that GIS is the ideal information management and analysis tool for many aspects of the transportation industry including both the public and the private sectors. Diverse areas of transportation, including highway, railway,

airport, bus and rail service planning, transportation modeling, and others, are applying GIS to their work. GIS is emerging as an important planning and management tool for transportation professionals.

1.1 Background

One of the important functions of public transit services is to provide customer services, including transit information (Syed and Khan, 2000). Traditional customer information systems provide users with information about routing and schedules by offering route maps, printed transit schedules, and telephone assistance with itinerary trip planning. Generally, printed brochures depict routes in colourful maps and present schedules in a tabular fashion. Smaller systems often include all route schedules in one brochure, while larger systems print multiple brochures that cover the regional subsets of their system.

The telephone system allows a customer to dial a number corresponding to a transit stop. A computer-based system tells the user when a particular bus will arrive at the transit stop. Through a customer service representative, a user can either phone or email the transit agency and provide the origin and destination locations, as well as time requirements. The customer service agent plans a trip from a user's origin to destination based on the printed route map, bus schedules, and notices reflecting updates not yet made on the printed schedules. Some telephone services include software support to assist the transit

representative in searching for an itinerary, while others simply require the representative to search standard printed route and schedule information. In all cases, the use of a customer service representative supplements standard route and schedule brochures (Smith, 2000). However, neither customer service agents nor users know whether the planned trip is the optimal one. Manual trip planning is a tedious, time-consuming, and often error-prone process, and information is sometimes inconsistent from one operator to another (Peng, 1997).

These traditional approaches have served transit customers well due to a number of strengths. First, printed brochures are portable, providing users with complete information in making decisions at their convenience. Also, they require little to no support from transit agencies' staff. On the other hand, the strength of the telephone service is that it provides immediate, interactive assistance to transit customers. Customers who may have difficulty deciphering the brochure can interact with a transit representative to identify a suitable trip itinerary.

Unfortunately, each approach also suffers from significant weaknesses. Printed brochures are static information distribution tools. They require that the transit user manually complete the complex analysis necessary to extract an itinerary that meets one's needs. In addition, if a change is made to a route or schedule, the brochures must be modified and reproduced. Not only is this costly, but it also results in the risk that a large portion of the transit users will continue to rely

on outdated brochures to identify trips in a system with a new route/schedule structure (Smith, 2000).

Getting information from a customer representative has weaknesses as well. First, a user can access the service only when a transit representative is working. By expanding the availability of the service temporally, the transit agency is faced with higher operating costs. Finally, given that the service relies purely on verbal communications, it provides no visual information to the customer (Smith, 2000). If the customer is unfamiliar with the region, it may be difficult for a transit representative to describe verbally where a particular stop is located.

Clearly, there is room for improvement in the provision of route and schedule information. In particular, a service that maintains the strengths of brochures and telephone service, while eliminating their weaknesses is most desirable.

1.2 Problem Identification

Public transit systems are rarely thought of as a competitive business. In reality, the transit riders are captive consumers of transportation, the transit agency is a producer of transportation, and transportation itself is a competitive marketplace with many different options. For this reason, transit systems are always looking for ways to increase visibility, enhance rider services, and convey traveller information quickly, accurately and effectively. Currently, transit users are

generally provided with only printed brochures, so there is a need to provide users with more effective tools to assist in selecting trip itineraries. Furthermore, existing methods are not capable of dealing with the uncertainties in estimating travel times from an origin of a trip to the destination.

The Ottawa-Carleton Transit Commission offers its users route and schedule information through its schedule booklets, a web site, and a telephone service. These methods for conveying information have been suitable for current transit users. However, new users sometimes have problems finding the appropriate information.

One problem is the difficulty in finding desired locations (origin or destination) on the available route maps. The bus route maps usually include only the portion of the region's road network that pertains to the particular route. However, accurately pinpointing one's location in relation to the closest bus stop may require a complete road map of the region along with the route map. Another problem is the general difficulty of determining a suitable route using the information available.

The subject of this research covers the methodology for the development of an optimum route identifier potentially applicable to OC Transpo and/or other transit agencies. This development will complement and enhance the efficiency of the current setup of the phone system and web information delivery.

1.3 Goals and Objectives

The main purpose of this thesis is to improve on the current transit services available to users and potential users. The goal is to increase the efficiency of the trip planning process for users and transit customer service personnel. Therefore, a user-friendly, computer-based transit information system was developed. The transit information system determines an optimal trip itinerary based on the traveller's origin and destination at a specific time and day. Hence the system has the following five major attributes:

- List of trip locations and the capacity to enter addresses
- Location of the transit stops
- Schedule Information
- Improvement in estimating travel time
- Route Optimization

In developing this transit information system, the specific objectives were:

- To investigate the current state of transit information services and technology available throughout North America and major cities in Europe, Asia and Australia

- To assess and compare different path finding algorithms

- To develop a user-friendly interface and algorithm using Geographic Information Systems

- To incorporate a method for dealing with the uncertainties in travel time estimates

- To enable the system to be applicable to any transit agency

- To carry out sample tests to postulate the effectiveness of the system

1.4 Study Methodology

A system development study of this nature has to deal with numerous issues and integrate many knowledge items. Figure 1.1 illustrates the general methodology followed for the conduct of this thesis research.

Major elements of the study methodology include:

- Problem identification and definition of the various facets of the problem to be solved.

- Development of the methodological framework and identification of the constituent elements.

- Literature review, covering ITS and GIS as these relate to the development of the transit information system.

- Definition of the variables of transit operations.

- Identification of the building blocks of the transit information system, including navigational process, general principles of transit information design, the traveller information component, and an introduction to Ottawa's public transit system.

- Selection and characterization of the ArcView GIS software and Network Analysis.

- GIS-based application process and algorithm development.

- Development of the statistical analysis method.

- Synthesis of the transit information system and application.

- Validation of the transit information system.

- Compilation of summary, conclusions, and recommendations.

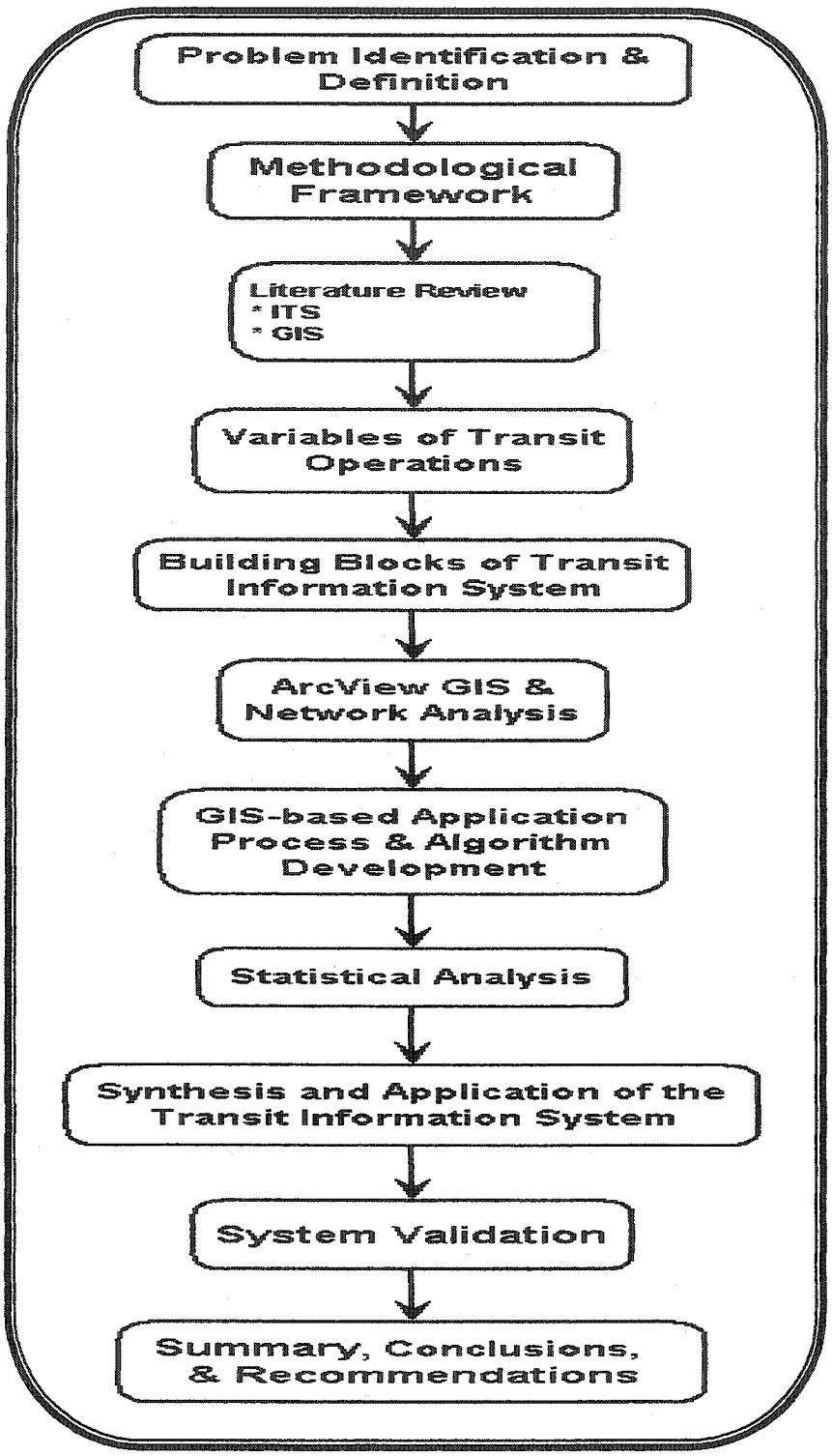


Figure 1.1: Study Methodology

1.5 Methodological Framework

Major elements of the transit information system development process are shown in Figure 1.2. These are noted below:

- Identification of user requirements
- Definition of system elements
- Abstraction of transit system within a GIS software (ArcView)
- Provision for a link with transit company's on-line information on the status of buses (e.g. an automatic vehicle location and control system supplemented with prediction software for bus arrival)
- Algorithm development for the identification of the optimum route
- Incorporation of Bayesian Statistical Analysis into the overall algorithm for the identification of revised optimum route and time solutions
- Provision of outputs to the user

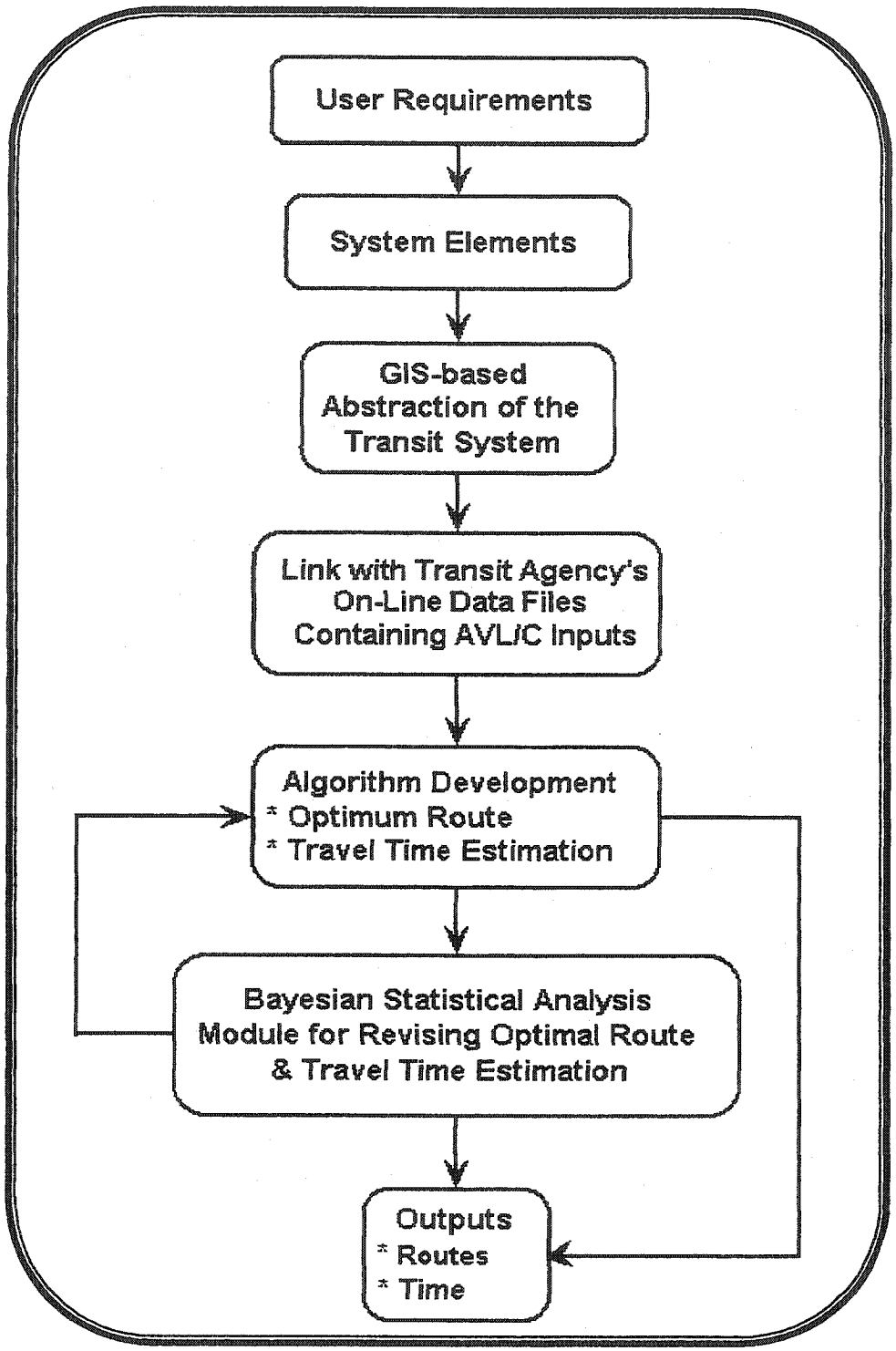


Figure 1.2: System Design Methodological Framework

A number of explanatory notes would be helpful in describing the selected elements of the methodological framework.

1) Although the generic aspects of the system design form an important part of the methodology, much effort was devoted to an operation system based on OC Transpo data. A selection of routes provided by OC Transpo along with their corresponding bus stop locations was used. This small sample of routes was chosen based on the complexity of the system. In total, 23 routes (Appendix B) were used along with almost 2000 bus stops. These routes were selected from each service time period, namely regular, rush hour, express, and early morning. The school and shuttle services were excluded. These routes were selected in order to obtain a good representation of the Urban Transit Area (UTA) in Ottawa (Figure 1.3). As well, scheduling and service availability was also considered to account for service in both peak and off-peak periods.

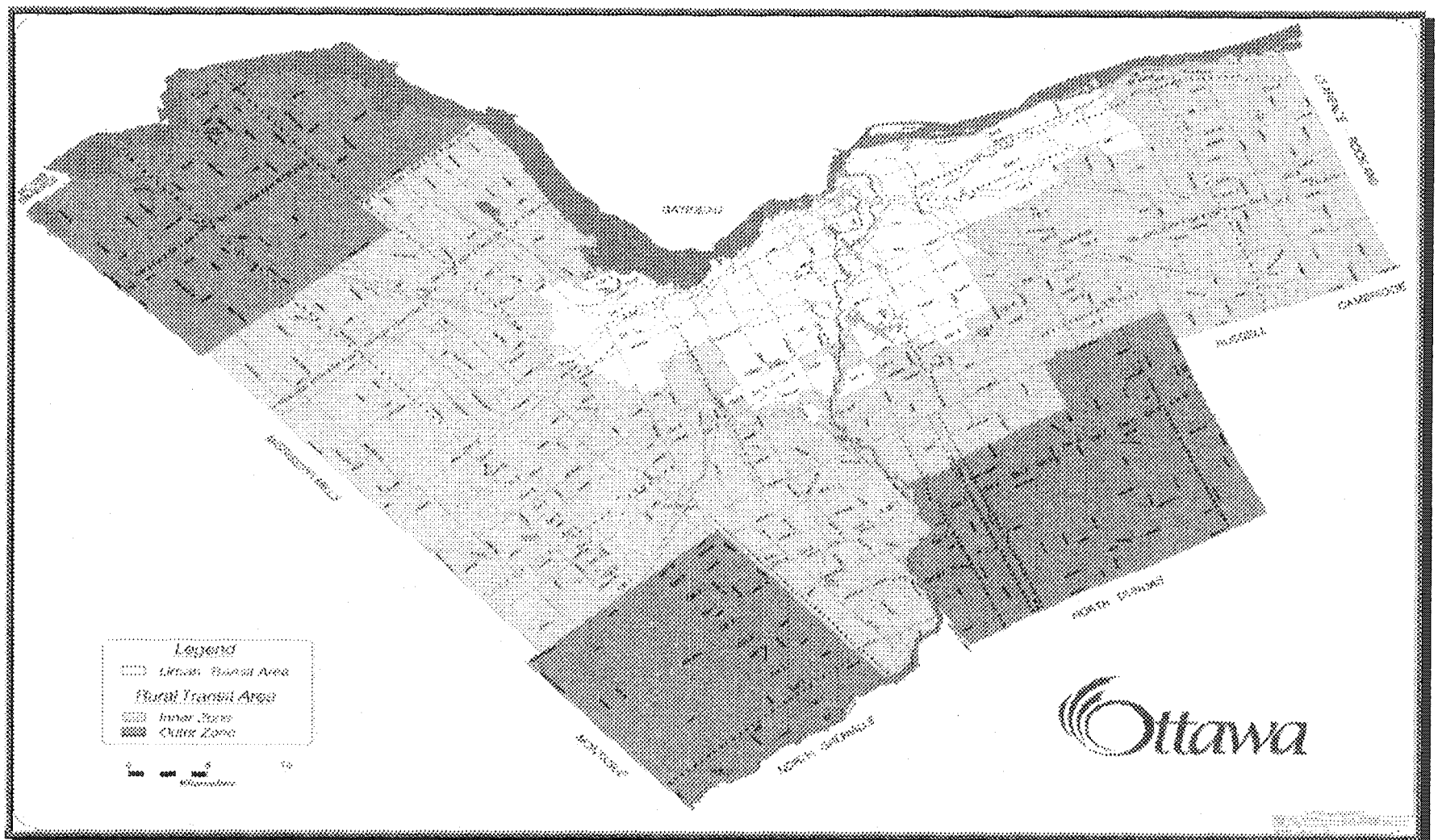


Figure 1.3: Urban Transit Area of the City of Ottawa (Leclair, 2002)

2) It was found that the path finding analysis capability was available in many GIS software programs. Based on a review of systems being used by transportation agencies, it became clear that ArcView and Arc/Info were predominant. Arc/Info is an excellent tool for storing, editing, and querying data as well as digitizing from images, but it requires a great deal of programming in order to perform tasks. ArcView can perform the above mentioned as well as provide an easy to use graphic user interface with menus, dialog boxes, and button bars for working. The system also enables simple programming capabilities as well as attainable system scripts for application. Therefore, ArcView3.2a was selected for use.

3) The algorithm developed will enable the user to request information based on origin and destination points as well as the time and day of departure. In solving the optimum path problem, many standard algorithms do not apply. Time travel on transit networks is constrained by a schedule as well as arrival times may be uncertain due to congestion, weather, or mechanical problems. As is well known, the transit network and service have many unique characteristics that differ from the street and highway network. This can become complex without real-time technology connections using AVL. Even if AVL is used, a prediction of bus arrival is necessary. Therefore it is assumed that users will time their arrivals with reference to the schedule and

that by applying Bayesian statistics, decisions can be made under uncertainty.

4) In looking at solving an optimum path for transit, there are a few things to consider before applying a standard algorithm. Firstly, transit service is time dependent since it must adhere to a schedule. The availability and the level of transit service vary by time of day and day of week because many transit routes have express and limited services at different times. To plan a trip at a specific time, the trip planning system needs service information for specific times and locations. A trip planning system that is based only on transit route configuration and assumes ever-present availability of service on every route and at all stops will generate erroneous trip plans. Therefore, the most important database requirement of the transit information system design is an incorporation of transit routes and the relevant transit services.

5) As well, in order to assign transit riders to specific routes and stops from the rider's trip origin, a trip planning system needs information on the relationship among transit routes, bus stops, and street network. This requires a relational database to link bus stops, transit routes, and street network.

6) Not every bus stop has a scheduled arrival and departure time. The bus operator is only required to adhere to the schedule for those stops that are on

the timetable. Therefore, for the purpose of trip planning the location of buses at specific times has to be estimated through linear interpolation.

7) Because of the unique features of a transit network noted above, a system has to be designed to address the time-dependent nature of transit services. It must be user-friendly and enable visualization of schedule information. In performing this task, network and geographic information is required along with a GIS software program.

8) The transit information system design process involves answering four major questions:

- Where is the nearest bus stop? (Location)
- When does the next bus leave? (Scheduling)
- How do I get from where I am to where I want to go? (Route Optimization)
- When will I arrive at my destination? (Travel Time)

In order to formulate these questions, the system will enable geocoding (method of deriving spatial coordinates for tabular data stored in an addressable form) of trip origin and destination points and link them to bus stops and routes. The respective bus routes will be selected and an optimum route will be found. The results will illustrate the preferred route and trip itinerary.

1.6 Scope of Research Work

This thesis research involved the development of a state-of-the-art transit information system by incorporating the latest ITS and GIS technologies and statistical analysis methodology for incorporating new travel time information as it becomes available.

The information system developed is operational. OC Transpo can use it with minor further development work. The system can potentially be accessed wherever web capability is available (e.g. desktop computers, kiosks, and handheld devices such as a cell phone).

The generic nature of the system design enables its application by public transit agencies other than OC Transpo.

1.7 Thesis Organization

This thesis is organized into 9 chapters. The first chapter introduces the reader to some background information regarding transit services provided to the customer. Research objectives, study methodology, methodological framework and scope of research are described. Chapter 2 is devoted to literature review covering ITS, GIS, and their application to transit. In Chapter 3, building blocks of transit information system are described. The navigational process is defined; the path-finding methods and the requirements for someone to navigate unfamiliar journeys by transit are explained. The general principles used in

transit information design are illustrated that include the types of aids utilized and current suggestions recommended by the Transportation Research Board. The traveller information system and its functions are described. Pre-trip traveller information systems are explained along with some potential obstacles. An overview of transit service in Ottawa is given along with some pre-defined long-term strategies that OC Transpo intends to implement. A review of the Smart-Systems and the current Automatic Passenger Counting System used by OC Transpo is described. In Chapter 4, ArcView3.2a is introduced along with its ability of customization. The programming language used in this application is explained as well as its relevance with the necessary extensions used during the development process. In addition, the route finding ability of ArcView is also explained. Chapter 5 begins the application process. In this chapter an overview of the steps taken in the design process are defined. The process used to input data, any modifications, and the applicable costs for the road network are explained. Any data preparation problems and solutions are also mentioned. The algorithm development used within the graphic user interface is explained. The statistical analysis used within the algorithm is introduced in Chapter 6. Here Bayesian statistical analysis is defined and how it is applied to the optimum route path. In Chapter 7, the application development is explained. Within this chapter the user interface design is explained along with the overall structure in order to describe the process taken by a traveller when using this system. Chapter 8 demonstrates some sample runs of the program in order to compare schedule information to that obtained by the transit information system. A

summary, conclusions, and recommendations along with some possible future work, are presented in Chapter 9.

This thesis also contains 8 Appendices. In Appendix A, the variables of transit operation are presented. It illustrates modal characteristics and trends along with present transit operations. The role of transit in urban centers is also illustrated as well as some parameters for bus facilities. Appendix B illustrates the transit routes used within the development of the transit information system. These are images of the entire route and the route classification (i.e. early morning, peak, regular). The following appendix is the coding developed for the user interface using the Avenue programming language. It includes the coding for user interface, the Bayesian analysis and the algorithm to determine the optimum route. Appendix D lists the attribute tables used within the user interface. These include all of the locations listed for the user and the bus routes and stops. In Appendix E the avenue coding used to solve for segment lengths is given. This code must be run before the network analysis can proceed. Another necessity for the network analysis is given in Appendix F. This is a table that enables the coding of speed limits in order to solve for travel time along line segments. The following appendix lists some of OC Transpo's operating statistics. These include the service area and facts. The final appendix lists the schedule tables used within the transit information system. These include weekday, Saturday, and Sunday timetables.

Chapter 2: Literature Review

In conducting this literature review, there were four main areas of concentration. Since the main objective of this thesis was to develop a transit information system, research was conducted in this area. This included uncovering past and present transit developments in providing users with customer services. The final section of this review gives an overview of predominantly American and European advances. It was found that very little has been done in the area of transit information systems in Canada.

Since technology plays an important role in the development of a transit information system, exploring technological and software advances in transportation was researched. It was discovered that the application of Intelligent Transportation Systems (ITS) is slowly being integrated to transit services. The most widely used subsections of ITS are the incorporation of Automatic Vehicle Location (AVL) devices and Geographic Information Systems (GIS). Hence, the literature begins with an overview of ITS technology and AVL system operations; the benefits and applications to transit services.

The following section discusses GIS, its components and areas of application. Since this transit information system was developed using a GIS software package, it was important to include an overview. One of the main areas that

GIS is increasingly being integrated is public transportation; therefore it was essential to find areas where GIS has been applied in the past and illustrate how GIS has helped to resolve these constraints.

As the transit information system is being used to solve for an optimum route, it was crucial to find information pertaining to optimum path algorithms. A review of past developments in optimum route algorithms pertaining to transit is discussed in the third section. Those discussed include generic data modeling and system operation modeling.

This literature review identifies past research and it provides an intellectual context for this thesis enabling a comparison to other work in this area.

2.1 Intelligent Transportation Systems

Currently, in the field of transportation, there is a push towards implementing Intelligent Transportation Systems (ITS) technologies. ITS is a broad range of diverse technologies applied to transportation to make systems safer, more efficient, more reliable and more environmentally friendly, without necessarily having to physically alter existing infrastructure. The range of technologies involved includes sensor and control technologies, communications, and computer informatics and cuts across disciplines such as transportation, engineering, telecommunications, computer science, finance, electronic commerce and automobile manufacturing (Transport Canada, 2000). ITS is an

emerging technology area and is benefiting public and private sectors alike. For example, ITS makes it possible to implement transportation safety compliance and road/bridge toll collection more economically, and to improve corporate productivity through time savings, reduced operating costs and energy consumption, and enhanced reliability and safety (Transport Canada, 2000).

The emphasis in much of ITS has been on road vehicular movement. Nonetheless, there is now a strong transit planning and management component to ITS efforts. Much of this takes the form of integrated information systems (kiosks, computer displays, Internet, postings, phone), common fare cards and SmartCards, articulated schedules (bus-rail), electronic payment, and variable fares. Also, ITS technology is being used to locate buses and improve schedule adherence, and research is continuing to improve ridership and travel time (Roess et al, 1998). Over the years, many other ITS applications have been implemented and planned both for roads and transit by a multiplicity of public and private sector organizations.

One of the preferred platforms for ITS technologies is a Geographic Information System. It is an area that has tremendous potential for better asset utilization and improved safety. Not all ITS applications involve GIS, but a growing number of implementations demonstrate that GIS adds a special new dimension to ITS, especially for incident management and traveller information. Most promising is the ability to use GIS to advise individual and fleet drivers of changing traffic

conditions en-route using wireless technologies and spatial databases. Like traditional planning and analysis GIS applications, spatially enabled ITS can also be used as a decision support system by incident management staff. Timely, easy access to accurate data helps people make better decisions on both an individual and community wide basis.

The future of ITS is promising. Already, real systems, products and services are at work throughout the world. Still, the wide-scale development and deployment of these technologies represents a true revolution in the way people think about transportation (USDOT, 1998).

2.1.1 ITS Technology

In many ways the goal of the ITS movement is to apply modern computer and communications technologies in our transportation systems, resulting in improved mobility, safety, air quality, and productivity. ITS technologies comprise numerous products and services that can influence the public, including (USDOT, 1998):

- **Intermodal transportation systems** that will make life easier for travellers switching from one mode (e.g., automobile travel) to another (e.g., air or rail travel).

- **Intelligent traffic control systems** that automatically adjust to the flow of traffic and reduce the time drivers stop at red lights for no apparent reason.

- **In-vehicle technologies**, such as traveller information and route guidance systems, and safety enhancement systems.

- **Safety enhancement technologies**, such as "smart" cruise control systems that take into account the location of the leading vehicle.

- **Traveller advisory systems**, including changeable message signs and advisory radio.

ITS technologies have the potential to improve the service and productivity of personalized public transit systems. These technologies are useful for both strictly curb-to-curb systems and hybrid systems. Various ITS technologies currently being used by the transit agencies include (Dessouky et al, 2001):

- Scheduling and dispatching software
- Automatic vehicle location devices (AVL)
- Advanced wireless communication
- Mobile data terminals (MDT)

- Computerized vehicle navigation
- Geographic database

The following emerging technologies are under development (Dessouky et al, 2001):

- Internet dispatching
- Wireless internet dispatching
- Superphones
- Personal data assistants (PDA)
- Smartmaps
- Smart bus technologies

2.1.2 Advanced Public Transportation Systems

Advanced Public Transportation Systems (APTS) are advanced navigation and communication technologies that are used in all aspects of public transportation. These include the application of advanced electronic technologies to the deployment and operation of high occupancy vehicles, shared-ride vehicles, conventional buses, rail vehicles and the entire range of paratransit vehicles. They encompass pre-trip travel information, en-route transit information, ride matching and reservation, electronic payment services, public transportation management, personalized public transit, and public travel security (Bang, 1998). These systems

enable transit agencies to make timely and needed transit information available to passengers, an element that is important to improving the convenience, reliability, and safety of public transportation.

APTS helps transit agencies manage a safe and efficient fleet and plan services to satisfy a broad range of consumer needs. When incorporated with a regional transportation system, APTS can also enable a system to manage its roadways with special accommodations for high occupancy vehicles. APTS applies advanced surveillance and communication technologies to rural area transportation systems with aim to improve safety, increase the efficiency of small community services and provide recreational travellers with location/navigation technologies (Bang, 1998).

2.1.3 Automatic Vehicle Location (AVL) Devices and Operations

In regards to transit, AVL devices are instrumental in ITS development. They are an electronic communication system for tracking and reporting the location of the vehicles to a central dispatching center. By knowing the location of the vehicle at the time of scheduling (assigning a specific trip request to a specific vehicle), it may be possible to improve the productivity of the system by better matching the passenger's request location with

the vehicle location. Actual position determination and relay techniques vary, depending on the needs of the transit system and the technology chosen (TRB, 1997). Typically, vehicle position information is stored on the vehicle for a time, which can be as short as a few seconds or as long as several minutes. Position information can be relayed to the control center in raw form or processed on-board the vehicle before its transmission. AVL systems also have the potential to reduce fleet sizes significantly (Dessouky et al, 2001, TRB, 1997).

Transit agencies have implemented AVL systems to assist them in a number of ways. These include the following (TRB, 1997):

Operations

- Improve schedule adherence.
- Improve service efficiency.
- Achieve better command and control of system.
- Improve bus schedules.
- Pre-process data for dispatcher.
- Facilitate systems integration.
- Improve information accuracy and availability.
- Provide better operations support.
- Reduce the number of street supervisors.

- Simplify operation of vehicle for the operator.
- Provide customers with real-time service information.

Safety

- Improve safety on buses.
- Improve response times to incidents and emergencies.

The dominant technology deployed today for locating the vehicle for AVL is the use of the Global Positioning System (GPS). GPS technology uses signals transmitted from a network of 24 satellites in orbit around the earth and received by a GPS antenna placed on the roof of each bus. A GPS receiver calculates its position based on signals received from at least three satellites.

GPS works anywhere the satellite signals reach. However, satellite signals do not reach underground and can be interrupted by the presence of tall buildings or dense foliage. In areas where this is a problem, GPS is often supplemented by another method of position determination, for extrapolation of location from the last GPS reading until the next GPS measurement (TRB, 1997).

Until recently, the accuracy of GPS data was about 100 meters. Through processing, most AVL systems achieve better accuracy

than standard GPS (USDOT, 2000). However, to further improve the position location accuracy, many transit agencies installed "differential GPS" (DGPS) when techniques such as signal priority were being employed or contemplated. In May 2000, the GPS accuracy was dramatically improved when the US military removed the intentional degradation to the signal that had been in place since GPS began operation. The accuracy of GPS today is between 10 and 20 meters (USDOT, 2000).

Prior to the availability of GPS, the most common form of AVL chosen by transit agencies was the signpost system in which a series of radio beacons is placed along the bus routes. An identification signal is transmitted by the signpost and is received by a short-range communication device on the bus. Since the location of each signpost is known, the location of the bus at the time of passing the signpost is determined. The distance traveled after passing the last signpost is measured by the bus odometer and is then used to estimate the bus position along its route (USDOT, 2000). However, this method can be limiting because signposts are placed at fixed locations. If a bus were to detour part of its route, it could not be tracked and if a transit route were altered additional signposts would be required.

Transit agencies often incorporate other advanced public transportation system features in conjunction with AVL system implementations. These include the following (USDOT, 2000, Casey et al, 2000):

Normally Integrated with AVL Systems

- Computer-aided dispatch software
- Mobile data terminals
- Emergency alarm
- Digital communications.

Sometimes Integrated with AVL Systems

- Real-time passenger information
- Automatic passenger counters
- Automated fare payment systems
- Automatic stop annunciation
- Automated destination signs
- Vehicle component monitoring
- Traffic signal preferential treatment.

2.1.3.1 AVL System Cost and Benefits

The capital cost of an integrated installation of AVL and other advanced public transportation system components is dependent on the size of the system, its level of sophistication, and the components to be included. Systems usually include a large amount of equipment (cameras, radio and microwave towers, mobile data terminals) and new software. There is a significant cost for the equipment and software that reside at the operations/dispatch center. Often, the installation of AVL coincides with a new or major upgrade to the communications system (USDOT, 2000).

The benefit and cost factors of AVL are numerous and varied. Basically, the AVL technology provides up to the minute information on exactly where the bus is located. This information can be broadcast to an automated dispatch system. This will present bus operational information in real-time to travellers and perhaps affect passenger demand patterns. The improved information enhances passengers' acceptance of the system and improves the perceived reliability of the system (Bang, 1998).

For the transit agency, the real-time information allows operators to monitor the vehicle condition and respond quickly to problems such

as falling behind schedule or traffic incidents. Enhanced control of vehicle operations and management improves service at lower cost (Bang, 1998). Since AVL technology is a major element of ITS, it provides a linkage by which transit agencies can participate in other fields of ITS applications. For example, a simple extension of the AVL can be applied to on-board passenger information, Advanced Vehicle Monitoring & Communication System (AVM/C), Vehicle Management System (VMS), pre-trip scheduling system, and demand responsive dispatching (USDOT, 2000).

Many public transit systems where AVL technology has been implemented experience more efficient and on-time operations as their schedules are improved, they are better able to respond to disruptions, (e.g., a disabled vehicle), and bus operators are aware of their schedule adherence (Casey et al, 2000). Safety and security typically increase, since the dispatcher knows immediately where to send help. AVL information also provides very useful inputs to passenger information and traffic signal preferential treatment systems. In the long term, the compiled historic bus position data can be used for planning route, schedules, fleet and personnel deployment (Bang, 1998).

From studies conducted by U.S. Department of Transportation, the following benefits from integrated implementation of AVL and other advanced public transportation system components were found (USDOT, 2000):

Operations

- Improved schedule adherence
- Improved transfer coordination
- Improved ability of dispatchers to control bus operations
- Facilitated on-street service adjustments
- Increased accuracy in schedule adherence monitoring and reporting
- Assisted operations during snowstorms and detours caused by accidents or roadway closings
- Effectively tracked off-route buses
- Effectively tracked paratransit vehicles and drivers
- Eliminated need for additional road supervisors
- Reduced manual data entry
- Monitored driver performance
- Received fewer complaints from operators.

Passenger Information

- Provided capability to inform passengers of predicted bus arrival times
- Helped meet Americans with Disability Act requirements by using AVL data to provide stop annunciation
- Increased number of customer information calls answered
- Eliminated need to add customer information operators.

Customer Relations

- Received fewer customer complaints
- Used playback function in investigating customer complaints
- Used AVL data to substantiate agency's liability position
- Improved image of agency.

Scheduling and Planning

- Provided more complete and more accurate data for scheduling and planning
- Expected to ultimately reduce schedule preparation time and staff
- Aided in effective bus stop placement
- Generated more accurate ridership counts with automatic passenger counters
- Expected to improve bus productivity.

Safety and Security

- Used AVL-recorded events to solve fare evasion and security problems
- Reduced the number of on-bus incidents by use of surveillance cameras
- Provided more accurate location information for faster response
- Foiled several criminal acts on buses with quick response
- Enhanced drivers' sense of safety.

2.2 Geographic Information Systems

As GIS was first introduced, most people used its software as methods of map-making, since manual drafting and map production was slow and tedious. This new method was more efficient and updates of maps were fast and more accurate in terms of data content (Clarke, 1997).

But to appreciate the aims of GIS, it is useful to understand the main components of its title. 'Geographic' is related to data and attributes that have some sort of spatial identity. These might be point features relating to specific sites such as schools, hospitals, or shopping centers; line features such as roads or rivers; or area features such as census wards, parliamentary constituencies, or postal districts/sectors. 'Information' is usually that, which defines something in decision making or planning. This may be present in data sets, data linkages or data

models. Hence, an information system is a set of organized procedures which, when executed, provides information to support decision-making. Therefore, a geographic information system possesses these properties in relation to geographical data and information (Birkin et al, 1996).

GIS has not only made the production and analysis of geographic information more efficient, it is changing the way geographic information is perceived and used. It is technology that makes geographic data more manageable, more easily shaped by the user into the form best suited to the application at hand. It makes geographic information more easily customized. In a GIS, the storage of data is independent of the mode of presentation. The physical map becomes a relatively inexpensive output product that can be generated quickly and customized for a single application. In addition, the geographic data used to produce the map can be continuously updated. As a result, the physical map becomes a customized "snapshot" of a continuously changing geographic database (Birkin et al, 1996, Clarke, 1997).

The processing power of the GIS has also enabled geographic information to be used in a quantitatively different way. Complex analyses can be iteratively refined towards an optimum solution, an approach that would be prohibitively expensive using manual methods.

A GIS has four necessary components: computer hardware (a personal computer or a workstation), a software package (for analysis of relationships and interfaces with databases), data (from various government, commercial and internal sources), and people to design it and use it.

Functionally, GIS provides tools for acquisition, editing, managing, analysis, modeling, display, output and dissemination of data. It is a computer-based tool that can perform the tasks of input, storage, retrieval, manipulation, analysis, and output for both spatial and non-spatial data (Clarke, 1997).

Data linkage is one of the most fundamental methods of adding value to data. In most geographic information systems, the concept of overlay is central to the process of linkage; Figure 2.1 illustrates an example of this. Data are stored in layers that can be retrieved from a system and then overlaid one on another to answer questions such as optimal site location. The overlay procedure is often undertaken in conjunction with spatial buffering (Aronoff, 1995). Buffering enables the user to determine an area at a chosen distance from a point or a line feature as shown in Figure 2.2.

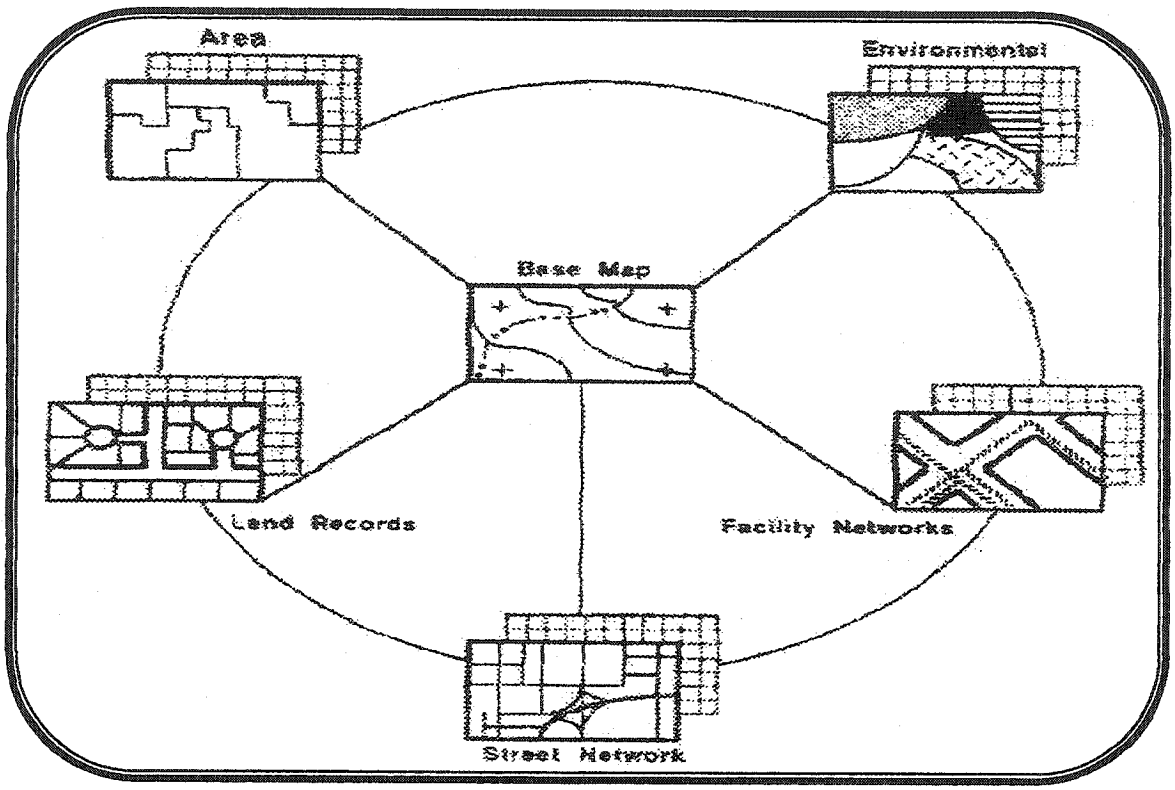


Figure 2.1: Concept of Overlay (Aronoff, 1995)

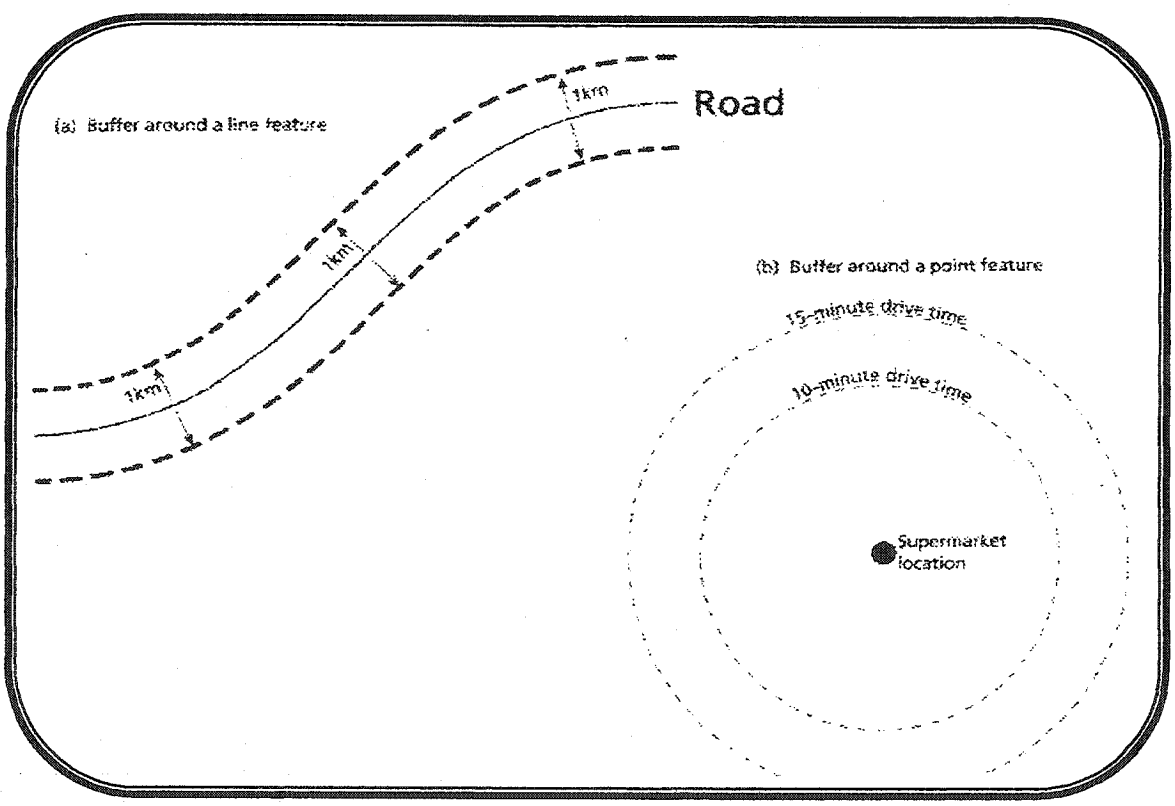


Figure 2.2: Buffering (Birkin, 1996)

Once data have been given a spatial reference (geocoded) then the system can perform a number of spatial queries. For example, once a specific map location has been highlighted, the GIS will search through all of its information banks to retrieve all recorded information associated with that geographical feature.

It is important however, to realize that geographic information systems provide data systems rather than information for policy making. Typically, existing systems can synthesize and integrate spatial data effectively, but do not have the capacity to forecast or plan ahead, or to provide the basis for impact analysis. An ideal GIS incorporates statistical methods as well as forecasting models (Aronoff, 1995).

Using GIS in the field of transportation opens up a wide variety of possible applications. GIS provides applications for ensuring smooth flow of the distribution of goods and people through aiding in design, routing, traffic control, and real-time navigation. In essence, a GIS application in transportation is a merger of GIS with Intelligent Transportation Systems, where GIS no longer exists as a stand-alone product (Husdal, 2000).

2.2.1 Applications to Public Transportation

Improving the mobility of today's citizens is a great challenge for public transit operators. Urban areas and employment centers are dispersing while automobiles have reduced the demand for public transportation. Therefore an economic and efficient system is required to enable transit agencies to provide sufficient services.

GIS has been in existence for many years, but only recently has its availability and application by transit been growing dramatically. GIS enables public transit to bring the power to communicate information through graphics to various aspects of transit planning, marketing, and decision-making. In an era of broad-based participation and complex assessments of information, GIS enables transit to leverage the evolution of computer technology and software development as a useful tool for improving the quality of transit services (Smith, 2000).

Transit planning applications in GIS are spreading throughout the world as new personal computer systems are being acquired and as several regional planning or city government GIS applications become familiar to transit planners. These applications range from simple uses of GIS for routine data analysis tasks to more sophisticated modeling tasks that usually involve programming

interfaces and data exchanges with other planning tools (Attanucci and Halvorsen, 1993).

As stated earlier, GIS is defined as a tool that supports the processing of spatial data into information. By having this ability to support spatial analysis, while also simultaneously allowing for the management and analysis of large quantities of attribute data describing the spatial entities, makes GIS well suited to support the identification of transit trip itineraries, in this case, the *information* in the definition. Furthermore, GIS is currently used extensively by public transportation agencies, primarily to support planning functions.

When transit agencies implement the GIS software and tools; tracking, routing, and scheduling, provide the ability to answer the where and how to travel questions. GIS helps agencies to reach new levels of success, customer service, efficiency and technological excellence (www.esri.com).

The beauty of GIS is in its wide scope and applicability to virtually any type of industry and activity. Thus GIS is not restricted to the conventional view of movement and transportation e.g. roads and railroads but extends to any from of motion and activity. GIS can

provide the tools one needs to gain the insight (www.corpweb.semcor.com).

Drivers, dispatchers, maintenance workers, route planners, management personnel, and even riders can make better decisions about their jobs and transportation options when they can make safe, reliable choices based on real information and not guesses. Information on bus routes, current location and schedule adherence, transit stop location, emergency situations and locations, road condition, demographic changes, employment centers, rideshare programs, and neighborhood characteristics are all the factors that can be used to improve transit performance. It enables transit operators to meet this challenge by putting information in the hands of people who can use it (www.esri.com).

There are a number of different uses for GIS in transit. Among its many applications, a GIS can be used for the display and/or analysis of the following (Casey et al, 2000):

- Bus routes, streets, parking lots, facilities, shelter locations, ridership loadings, running times, scheduling, bus assignments, dead-head routings, accidents, and customer complaints - for service and facilities planning;

- Street and route maps, service performance monitoring, vandalism location and history, and emergency call location identification - for operations purposes;

- Land uses, employer sites, demographic data, and travel patterns - for market development;

- Bus route maps, trip planning route choices, on-time performance data, multi-media displays, pass sales outlet planning, and customer complaint data - for customer information/service purposes;

- Customer address location, service qualification determination, and service performance statistics - for Americans with Disabilities Act service operations; and origin and destination of ridesharing applicants, custom bus service requests, and high occupancy vehicle (HOV) lane violations - for other transportation service analyses.

This extensive use of GIS is particularly advantageous in that most transit agencies will likely possess the data and software required to serve as the foundation for a transit trip itinerary planner. GIS

has the potential to significantly increase the quality of urban transportation planning data while reducing the cost of data collection and preparation by enabling transit and other local agencies to share and disseminate data (Schweiger, 1992). To further explore the applicability of GIS to this problem, consider some of the fundamental spatial analyses required by a transit trip planner. As stated earlier, proximity analysis (buffering) is important. It is key to determine if a candidate bus stop serves as a viable origin or destination point. The basic GIS tools of "overlays" and "buffers" provide the needed functionality to support this requirement. For example, once a customer's point of origin is located within the network, a GIS software can be used to see if that point falls within a user-defined distance from a bus stop. Furthermore, the use of a map-based GIS interface will allow a customer to determine if certain objects comprise unacceptable barriers to reaching a potential transit stop.

Nearly all commercially available GIS are built upon a relational database management system (RDBMS). The RDBMS serves as the engine that supports the management of attribute data describing the spatial entities (DeMers, 1999). This capability provides the functionality needed to store and analyze schedule information. The database, therefore, plays a key role in the trip

itinerary planner system. It can organize data describing the transit system, and provides a mechanism to store and manage the preferences of the user as they interact with the system.

The capabilities that GIS can bring to transit agencies help with the growing needs of the industry. For example, the ability of GIS to communicate detailed data in a format more easily understood by decision makers and citizens makes it extremely valuable. Transit agencies can also use GIS for planning and analysis and to deliver more cost-effective services. GIS can be used to analyze the service area of a transit system. The analysis is also useful in addressing a variety of policy and service planning issues (Smith, 2000).

Since GIS can associate large data files with particular geographic points, routes, or small street segments, it is also an ideal tool for organizing large amounts of data associated with public transit routes, schedules, and facilities. It provides a transit planner with the following general capabilities (Attanucci and Halvorsen, 1993):

- **Database Management:** A GIS will allow transit managers to store data such as route descriptions, bus stop files, ridership

information, population and employment demographic data, street and neighborhood characteristics.

- **Geographic and Thematic Mapping:** A GIS allows a planner to display and plot each database contained within it on a geographically accurate electronic map.

- **Planning Analyses and Evaluation:** Most GIS include a variety of preprogrammed data analysis routines that provide the transit planner with automated tools to aggregate and disaggregate various datasets according to predetermined criteria.

- **Presentation Tool:** The unique combination of accurate mapping and graphic representation of demographic and other statistical data provides transit planners with a new tool to present information to colleagues or to the public.

The capabilities that GIS brings to the professional planner and transit properties complement the evolving needs of the industry. For instance, the ability of GIS to communicate detailed data in a format more easily understood by decision makers and citizens makes it extremely valuable. Transit properties can also use GIS

for planning and analysis and for addressing issues such as those related to the equitable allocation of resources (Crowson et al, 1997).

GIS can provide a great deal of constructive and cost-effective assistance to transit agencies as they strive to deliver efficient and effective customer-focused transportation services.

> The evolution of user-friendly GIS software, efficient desktop personal computers, and refined geographic and demographic databases has resulted in moving GIS from a highly specialized capability requiring extensive training and support to one in which anyone can develop the necessary skills to provide a cost-effective GIS capability for transit properties (Ball, 1996).

2.2.2 Fixed-Route Service Planning and Scheduling

Transit service planning and scheduling encompass service demand and supply, as well as the spatial equity and efficient service allocations. This analysis of the transit demand and supply is accomplished by identifying spatial distribution of transit service, ridership, and their interaction with demographic and land use characteristics (Peng et al, 1998).

The developed GIS database provides transit planners a tool for planning analysis and performance evaluation. For example, service planners can use the database to identify and display major

sources of transit ridership by time of day at the stop level, route segment level or route level, so that they can determine the level of service on the route level and evaluate the route-on-route segment performance based on the observed ridership (Smith et al, 1998).

The challenge for route planning is to cover as many transit origins and destinations as possible while providing users with direct routings. The GIS database allows planners to identify potential transit stops to be served and to generate transit routes automatically. Transit planners can use the GIS data to identify the demographic and land use characteristics within walking distance of a proposed transit route and can then estimate potential transit ridership at the route-level (Peng et al. 1998). Every time planners design a new route or intend to make a route configuration change, they can know the demographic and land use characteristics of the route coverage area, and the potential ridership changes.

The relational structure in the GIS database that links temporal scheduling and spatial network is very valuable in service schedule planning. The GIS database allows the user to identify the topological and temporal relationship of different transit routes and stops. This is particularly helpful to facilitate the integration of fixed-

route bus and light rail dispatch to identify key system transfer points between bus and rail service (Smith et al, 1998).

2.2.3 Potential Obstacles to Transit GIS

The obvious question that may be asked at this point is if GIS is so great, why doesn't every transit agency use it? This question may be very difficult to answer if the same situation exists a few years from now, but for now some obstacles still exist to establishing an effective transit GIS. These include at least the following real or perceived problems.

Firstly, there are some difficulties in learning and using GIS software. Earlier versions of commercial GIS software required hours of training and repetitive use of different options to master the programs. While points and small segments of routes are generally handled well by most GIS, full or partial transit routes using multiple street segments are still difficult to manipulate using some commercial products. It should also be noted that the routing and scheduling features vary widely and only a few GIS packages can effectively address certain transit applications such as paratransit, trip itinerary planning, and other network modeling tasks. In general, the ease-of-use of the better GIS products has improved (Attanucci and Halvorsen, 1993).

Secondly, starting with no existing data can be a difficult task. Fortunately, GIS vendors now often can provide preloaded U.S., Canadian, or world data and other customized database development services through ESRI. In addition, many transit agencies can now look to other regional agencies to get at least a start on their databases

Thirdly, with as many as 100 different commercial GIS packages available, incompatibility of data files is likely in some cases when different agencies use different GIS tools. Even when the software outputs are compatible, inter-agency cooperation in sharing recent data can be lacking. Presently, users are investigating how a particular GIS handles other competitive systems' data before deciding on a standard for any specific application (Attanucci and Halvorsen, 1993).

Subsequently, and perhaps the largest obstacle to the most effective use of GIS in transit agencies is a lack of consideration about its broader use within the agency. In many cases, a particular department starts using the GIS application without determining how other databases might be interfaced or be inputted into their application. As a result, duplicate data entry is often

required with other text-based applications such as bus stop inventory programs or ridership count analysis software (Attanucci and Halvorsen, 1993).

Finally, until recently, initial GIS software and hardware purchases required substantial resources, with new mainframe and mini-systems running into hundreds of thousands of dollars and even up into the millions. While software and hardware prices have dropped, full-featured GIS packages are still not in the same league as typical office automation commercial software. The costs of specific application development within a GIS environment can vary widely depending on the product and the feature required (Attanucci and Halvorsen, 1993). GIS applications development will likely continue and resources need to be regularly budgeted to take advantage of a fast-developing new technology.

2.3 Optimum Route Algorithms

The transit network is much more complex than the road network because it is strongly time-dependent and highly dynamic. Transit-oriented network data models and routing algorithms are considerably underdeveloped compared to those of highway networks. With the growing demand for online transit information services and trip itinerary planning, there is an increasing need for efficient transit network data model and trip planning algorithms.

The analysis of transportation networks is one of many application areas in which the computation of shortest paths is one of the most fundamental problems. Several algorithms and data structures for algorithms have been put forward since the classic shortest path algorithm by Dijkstra (1959). In its modified version, this algorithm computes a one-to-all path in all directions from the origin node and terminates when the destination has been reached. Most computer applications, including ArcView, use this method to calculate an optimum route path to travel.

When it comes to developing an algorithm for transit, compared to its highway counterpart, transit networks receive much less effort in terms of data modeling. Transit is usually mentioned in a few words in the development of the general transportation data model.

Nevertheless, this all changed in the 1990's when data modeling for public transportation emerged. Work in the field of generic data model for Linear Referencing Systems brought together ideas from Koncz, Greenfeld, and Mouskos (1996), Deuker and Butler (1998), Peng and Huang (2000). These researchers are GIS-T professionals and practitioners and most of the research is focused on Linear Reference Systems for highways or on generic linear data models.

Koncz et al (1996) designed an approach to develop an algorithm and strategy for transit providers to find best route alternatives for the user, and to demonstrate how a geographic information system can be used in the development of transit advanced traveller information system (TATIS) to meet these needs. The major difference between this algorithm and others previously developed are the capabilities of handling multiple modes of transit, providing paths that include walking distances and provisions of multiple optimal paths to allow the user flexibility in choosing a path.

Dueker and Butler (1998) developed a generic GIS-T enterprise data model that is independent of map scale, specific entity attributes, mode of travel, and location measurements methods. With this data mode, criteria for path finding are attached or linked to traversal segment or nodes. Single or combined variable optimization based path finding algorithms, such as shortest path algorithm, can be employed to search the network. When this model is implemented in transit, a bus route with a direction can be expressed, as a traversal, and bus stops and transfer nodes are point events. This data model and others like it, have been implemented in many headway-based shortest path algorithms.

Another group of researchers in the transportation area are more focused on transit operations. Instead of working on GIS data modeling, they are more concerned with system operation, optimization, statistics and planning. Within

this group, transit is studied in terms of route assignment, trip planning, scheduling and providing service (Tong and Richardson (1984), Wong and Tong (1998), Peng and Huang (2000)).

Path finding and route assignment algorithms developed by this group fall into two categories: headway based and schedule based. Headway based algorithms assign passengers to the first arriving vehicle based on the combined bus frequencies on the same street. Several probabilistic or stochastic models can be linked to this category. The headway-based approaches are often applied to congested transit systems and work well with frequent service environment with headway of 5 minutes or less.

The optimal path search by schedule-based algorithms is strictly dependent on the schedule. Schedule-based algorithms are sometimes referred to as heuristic or deterministic approaches. In the model developed by Wong and Tong (1998), the estimation of time-dependent origin-destination (O-D) matrices for transit network based on observed passenger counts is given. The dynamic assignment framework uses a schedule-based transit network model to help determine the time-dependent least cost paths between all O-D pairs. An entropy-based approach is then employed to estimate the time-dependent O-D matrices based on the observed passenger counts at those observed links in the network. It was found that there was a good agreement between predicted and observed matrices.

Tong and Richardson (1984) developed a "branch-and-bound" algorithm approach to solving schedule-based optimum route finding. This time-dependent algorithm finds the minimum path between two stations in a multi-route, multi-mode transit system. Selection of the minimum path can be based on either journey time or on weighted time. This model can be used for generating route schedule information to guide transit users, for assisting in route schedule coordination, and for analyzing transit system accessibility.

Based on the data model, Peng and Huang (2000) developed two basic schedule-based path-finding algorithms and one basic non-schedule based minimal transfer path algorithm. Taking a spatio-temporal perspective, they developed an object-oriented data model in which the network itself is modeled as an object. The network object is composed of a set of subclass objects including a time map mechanism that monitors temporal behaviour and builds dynamic network topology. In addition, the Object-Oriented model substantially reduced network redundancy and enhances path search efficiency. Based on the data model, the authors develop two basic schedule-based path-finding algorithms and one basic non-schedule based minimal transfer path algorithm. By using various combinations of the basic algorithms an array of path options are generated to mimic human cognition in path finding.

As an interesting final note, three researchers in India developed a model for public transportation routing and scheduling using genetic algorithms. Rao, Muralidhar, and Dhingra (2000) presented the applicability of genetic algorithms as a possible technique instead of the current heuristic procedures used.

The design process was done in two phases based on two main objectives. These are to minimize the in-vehicle travel time and the transfer time for the whole network and the minimization of an overall cost generally a combination of user cost (sum of in-vehicle travel time, waiting time and transfer time) and the operator's cost.

This was applied to a small network where most of the node pairs had non-zero demands whereas in real networks transit demand matrices tend to be sparse, with fewer non-zero entities. The results were favorable but they found that work was needed to investigate in detail the effect of various parameters on larger networks.

2.4 Transit Information Systems

Within the last few years there has been an increase in the development and enhancement of aids to helping transit passengers travel more efficiently. Most of the early research and development has been done in Europe. The developed concepts and ideas have performed well and have recently spread to parts of

North America and other major cities around the world. Most have implemented new telephone operated systems, kiosk monitors and real-time Internet based systems. These variations of systems, all build on existing infrastructure and technology. Currently, most transit agencies rely on schedule information found in procures and on the Internet.

The most widely used and well-known system in the United States is NextBus. This company combines existing technology to provide real-time information for commuters in California (San Francisco, Oakland, Emeryville, Glendale, Santa Barbara), Boston and Lowell, Massachusetts, Vail, Colorado and Fairfax, Virginia. NextBus was the first company in the United States to serve passenger-focused information to the Internet. (Casey et al, 2000)

NextBus is not a static schedule listing; it is actual arrival information, updated at regular intervals. NextBus uses satellite technology and advanced computer modeling to track vehicles on their routes. GPS receivers are installed on transit vehicles, and this tracking information is used to calculate when a vehicle will arrive at a particular stop (www.NextBus.com).

Each bus receives frequent signals from satellites that are used to give an accurate location of the bus. This information is then passed by radio to a central computer and to displays at bus stops. Each display then determines the waiting time to bus arrival which is shown with updates every 30 seconds. Taking into

account the actual position of the buses, their intended stops, and the typical traffic patterns, NextBus can estimate vehicle arrivals with a high degree of accuracy (www.NextBus.com).

The accuracy of the predictions varies depending upon changes in traffic patterns, accidents, or the amount of Internet activity (number of users at one time). Therefore, NextBus cannot specify a precise overall accuracy. In general though, the margin of error is less than one minute for predictions of five-minutes or less. For ten-minute predictions, the margin of error is less than two minutes. These predictions are made available on the World Wide Web and to wireless devices including signs at bus stops and business, Internet capable cell phones, Palm Pilots, and other Personal Digital Assistants (PDAs). (www.NextBus.com)

Although NextBus is the most widely used system in the United States, there are other companies that have developed Automated Trip Planning Systems. These include Tidewater Consultants Inc., Megadyne Information Systems, and TeleRide Sage.

The system developed by Tidewater Consultants Inc. is designed to run in a Windows-based environment on a standard PC. The system relates stops, routes, and schedules to a GIS based on a Rapid Routing Module, an algorithm that computes a trip plan for a customer who has called. Travellers call the transit agency's telephone information center; give their origin and destination,

and the system computes a trip plan, generally in under 10 seconds. Looking at a pull-down menu, the telephone center agent then can describe to the caller the proposed itinerary, or can fax it or send it in the mail. These printouts can be multi-lingual, in Braille and/or in large type (Casey et al, 2000).

Megadyne markets several unique software products, as well as related marketable technology. One of Megadyne's products is directed toward public transit/transportation. Among the many Megadyne Information Systems developed most activities range from system development/integration, to professional services, to software products, and to process management. The systems developed by Megadyne include:

- PARIS: Passenger routing and information system
- PARISpeak: Interactive voice response system
- V-TRAX: Automated vehicle monitoring system
- SPARTAN: Schedule planning and real-time assessment
- ATHENS: Automated vehicle dispatch system
- Dial-A-Ride: Automated ride pooling system
- Rideshare: Carpooling and vanpooling system

First among these products are PARIS and PARISpeak. PARIS provides fast, accurate, and detailed information on transit services; generates optimum trip itineraries; and provides schedules, fares, stops, routes, travel times, multi-transfer itineraries, service mode, dispersed sales outlets, emergency updates,

handicapped access, and related management, trend, and cost information. PARISpeak is an automated voice response software that works either as a stand-alone product or in conjunction with PARIS. The greatest utility is achieved when the two products work together (www.Megadyneinfo.com).

Teleride Inc., a TTi Systems Company, is a global organization providing a unique spectrum of products and solutions to the transit industry. The products developed enhance the performance of transit agencies through scheduling and run cutting, itinerary planning, passenger information, driver management, demand-responsive scheduling and dispatch. All products are engineered on the Windows NT platform, using state-of-the-art Rapid Application Development tools and Object-oriented programming methodologies (www.Teleride.com).

The TeleRider Interactive Voice Response (IVR) System defines the state of the art for transit passenger information systems. It diverts calls from telephone information center operators, reducing abandoned calls, busy signals and call waiting times. It is available to passengers 24 hours a day, 7 days a week and can obtain schedule information within 20 to 30 seconds. It tells passengers if their bus is on time, delayed, detoured or canceled (www.Teleride.com).

TeleRider Plus provides route recommendations and route and schedule information on the basis of current transit network data. When a starting point and destination or a specified stop, address or point recognized by the

Geographic Information System and a date or day of the week and time of day of travel are entered, TeleRider Plus calculates the most effective connections. It is a client-server solution offering users access over a standard network. All transit information is calculated online by the TeleRider Plus server and then made available to users for viewing, further calculation, printing and/or fax distribution (www.Teleride.com).

TeleRider Plus is also Internet-enabled, allowing travellers with Internet access the convenience of planning their upcoming trips. It utilizes a highly developed data format to provide and calculate transit information efficiently on-line. A specially developed program is used to convert transit schedule and route information from the data management module into the format utilized by the TeleRider Plus server. TeleRider Plus offers the traveller accurate, detailed recommendations based on the actual date of requested travel (www.Teleride.com).

Individual cities and counties have also implemented programs themselves through their department of transportation. In San Diego County, the system implemented is called InfoExpress. It is an interactive voice response system available 24 hours a day, seven days a week. Callers can get basic route, schedule, and general information in Spanish or English for seven transit agencies in greater San Diego. This enables information agents to help with more complicated questions like trip planning (Casey et al, 2000).

The Winston-Salem Transit Authority, through the Mobility Manager model project being developed under the FTA's APTS program, is providing users with a menu of transportation services by telephone. By calling a single number, a prospective passenger will immediately be able to schedule a trip, ask about the status of a trip, make arrangements to transfer more readily from one mode of travel to another, or receive a schedule of transportation service available in the regional area. In the future, the Mobility Manager project will be expanded to include electronic variable message signs to provide passengers with real-time information about a vehicle en route, such as updates on delays or revised schedules (Casey et al, 2000).

In Los Angeles, California, CalTrans is directing the Smart Traveller program, which is a free automated information service to provide commuters with up-to-the-minute freeway conditions and traffic speeds, customized transit route planning, and real-time carpool matching. It has a telephone information service operating 24 hours a day, which provides Los Angeles County Metropolitan Transportation Authority information. As well, there are kiosks available. They utilize a personal computer based system, which is connected via dedicated digital telephone lines to a telecommunications system. This system accesses the rideshare, transit, and freeway conditions information residing in databases maintained by Commuter Transportation services, Inc., Los Angeles County

Metropolitan Transportation Authority, and CalTrans Transportation Management Center (Casey et al, 2000).

The kiosks provide personalized itineraries including routes, fares, schedules, and origin-to-destination travel times; carpooling possibilities; real-time freeway conditions; and videos on various transportation topics. The user interface is a touch-screen monitor. Freeway flow status is displayed on a map. All other information is displayed in textual form (Casey et al, 2000).

In Seattle, Washington, the King County Department of Metropolitan Services is employing several forms of electronic communication technology to access vital transportation information relating to the greater Seattle/Puget Sound region. Riderlink is an on-line information resource available on the Internet via the World Wide Web. It gives instant access to Seattle Metro bus routes, timetables, and maps, as well as information about vanpool and ridematch services, bicycle transportation, freeway congestion, the state's commute trip reduction law, and a variety of other transportation topics (Casey et al, 2000).

Seattle Metro's automated Bus-Time system makes schedule information available to anyone with a touch-tone phone; people can access information about their route and bus stop. As callers respond to a series of questions to identify their route and bus stop, a digitized script is packaged and passed on to

where schedule, route, and bus stop information, and a digitized vocabulary are stored (Casey et al, 2000).

Along with the current telephone system in Seattle, Washington and Portland, Oregon, MyBus was created. The goal of MyBus is to present to riders, in real-time, the predicted departure times of buses at specific locations throughout a transit region.

MyBus is designed as a distributed application. Schedule data and real-time AVL data streams flow between components. These components include an AVL source, a prediction generator, a set of filters for bad or erroneous AVL data, and a web server for final text formatting and delivery over the web (Dailey, 2001).

The buses are equipped with GPS units that daily capture observations. Onboard computers and Automatic Vehicle Location equipment are linked with the Bus Dispatch Center to improve communication with bus operators, locate buses in real time and enhance data collection. A stop record is created every time a bus passes a stop or a door is opened. Automatic Passenger Counters record the number of passengers boarding and leaving. In addition, an event record is created to record other activities, such as a bus travelling on or off schedule. Bus drivers also send and receive digital messages, and key in more

than 30 pre-programmed codes to capture field information. All records include the time and coordinate position (Kalmer and Beckel, 1999).

The prediction component of MyBus uses three inputs: a schedule data set, a set of historical trip realizations, and a real-time AVL stream. The schedule data provides the expected time for each event. Historical data provides ensemble-averaged statistics to the algorithm that predicts vehicle departure. The AVL stream supplies instantaneous bus location information approximately every 1 to 3 minutes per vehicle (Dailey, 2001).

The predictor algorithm uses an optimum filtering technique based on Kalman filter technique. An underlying assumption is that any real-time arrival prediction algorithm depends upon reliable real-time transit vehicle location information. A separate prediction process is started for each goal to minimize statistical errors. The prediction algorithm produces the optimal estimate of the departure time given the information provided to the filter. To compare the predictions with the vehicle behaviour, the actual departure time needs to be estimated. Since the vehicle is tracked irregularly, there is no guarantee that the location will be reported just as the vehicle departs. To get an estimate of the real departure, the location report just before arrival and just after the departure can be recorded and linearly interpolated to the actual departure time. The filter is continuously predicting the departure as a function of both time and space. The statistics of the deviation of the predictions from the actual can be expressed as a probability

surface in space and time. Moreover, predictions are accurate both far in advance of the scheduled departure time, as well as near departure time (Dailey, 2001).

The second major component of MyBus is the web server program. This component is responsible for receiving input from the predictor component and for storing and formatting this prediction data in a manner suitable for output over the Internet. Currently, the Seattle predictor and the combined Seattle/Portland web site run as two Java Virtual Machines on the same Windows NT host (www.MyBus.org).

Touch screen kiosks, featuring full-motion color video, stereo sound, on-screen maps, personalized public transit itineraries, and carpool matches for commuters, have been installed in the Coachella Valley area of Riverside County, California. The pilot project, called TransAction Network, has four kiosks at shopping centers with high pedestrian traffic. The TransAction Network is being introduced by Commuter Transportation Services Inc. and SunLine Transit Agency. SunBus, the personalized public transit option enables the user to obtain transit information. By entering a destination, arrival, or departure time and a transportation option, a user can receive a free printout of a complete SunBus itinerary including route, bus stop, fare, and schedule, as well as a carpool match list (Casey et al, 2000).

The Mass Transit Administration of Maryland's transit information center has been updated through the upgrading of the computer system to contain all of the latest schedule information. When a customer calls the information center, the operator will call up the schedule on a computer screen, instead of looking through books. Moreover, the system will keep a running memory of the customer's call based on the number where the call came from. The caller's last question asked and home address can be brought up on screen by the operator, which saves time in processing requested the information. There are also five kiosks installed in greater Baltimore to provide the same information as the operators. The full intent is to have all information be real-time, after the AVL system is fully installed and refined (Casey et al, 2000).

The Rochester-Genesee Regional Transportation Authority (R-GRTA) has implemented an automated telephone system. When a call comes into the system, Direct Talk sends the request for route, time of day, direction, and location to the mainframe, and then translates the data into a verbal response to the caller. The typical time to complete this transaction is two minutes. The request for information begins with the caller giving the route number. If the route number is not known, a listing is provided, and the caller can choose from the list to complete the request. If further information is needed, the caller can cut back into the system to speak with a customer service representative without having to hang up and redial to obtain the necessary information (Casey et al, 2000).

The Metropolitan Atlanta Rapid Transit Authority has implemented a trip planning system that supports MARTA's customer information center and kiosks that were available for the Olympics. The Passenger Routing and Information System (PARIS) (provided by Megadyne) requires an origin and destination to produce a nearest bus stop-to-nearest bus stop itinerary. The itinerary does not provide instructions to reach a bus stop, but it does take into account walking time and impossible walks (e.g., crossing a river without a bridge). PARIS ranks all of the options for a trip and the highest ranked option is provided. In addition, a limited amount of highway-related information can be obtained from PARIS (www.Megadyneinfo.com).

The AVL system, which was implemented by Transportation Management Solutions for the Olympics, provides real-time vehicle status information to the itinerary planning system. This capability enables the trip planning system to provide itineraries based on actual vehicle arrivals and departures as well as providing information on service disruptions (Casey et al, 2000).

In the Minneapolis/St. Paul, Minnesota area, transportation officials are looking to advanced technology to make transit operations more efficient and attractive to commuters. Called Travlink, this demonstration project is part of Minnesota Guidestar, Minnesota DOT's program for ITS, which is actively testing and deploying new technologies that improve the movement of people, goods and

services. Travlink represents the integration of a computer-aided dispatch and AVL system based on the global positioning system, an advanced traveller information system (ATIS) and an automatic vehicle identification system in the I-394 corridor in the Minneapolis/St. Paul metropolitan area. Travlink is using a variety of devices and systems to distribute both real-time and static transit and traffic information to travellers. Travlink is designed to encourage commuters to consider alternatives to single-occupant travel, especially public transit (Casey et al, 2000).

In the state of Florida, the Department of Transportation has developed a software program that allows one to quantify and visualize the mobility provided by a transit system at different times of the day and week at any location within the system's service area. Other helpful tools provided by the TLOS software include the ability to measure transit level of service, the ability to create mailing lists tailored to potential customers served by particular transit routes, and the ability to estimate the percentage of persons using transit, in those locations and during those hours when transit is available as an option to them. The results of a TLOS analysis can be used in a variety of transportation planning applications, including transportation planning models and geographic information systems (www11.myflorida.com).

The newest version of the TLOS software adds route-finding ability, allowing a user to find the minimum and average travel times for a trip, and the transit

route(s) one would take to make the trip. The software can also determine the number of people and jobs located within a user-defined transit travel distance of a location, calculate transit travel times between all combinations of transit services within an area, and calculate the auto-transit travel time level of service measurement. It is expected that these new features will have a variety of applications, from transit planning, to welfare-to-work, to customer service (www11.myflorida.com).

The Champaign-Urbana Mass Transit District (CUMTD) currently, in addition to schedule booklets, provides bus schedules through the World Wide Web (WWW). Recently a Bus Information System for CUMTD, which is an interactive and easy-to-use WWW-based application, was introduced. Dynamap 2000 by ESRI provided the street information and addressing and MapObject, NetEngine, and Arc/Info were used to develop the system. This system provides users with the shortest travel path and transfer information once the users enter an origin and a destination. The common algorithm provided by NetEngine 1.1, was used along with some modifications that combined the schedule data with the network data from the entire bus route network (Lee, Baumgartner, Kim, 1999).

BusView was successfully designed and demonstrated using real-time transit coach locations on a digital map to the University of Washington campus community. The project was further enhanced to encompass developing a graphical transit information system using data from King County Metro's existing

automatic vehicle location (AVL) system and the Puget Sound's regional intelligent transportation systems (ITS). It was created and launched on the World Wide Web providing real-time bus location information (Dailey, MacLean, Pao, 2000).

During the Seattle Smart Trek Model Deployment Initiative (MDI), the ideas developed in the campus-based version of BusView were used to create a version that could be widely supported on the Internet. Two new applications were created to provide real-time transit information. The first project, BusView Plus, is a desktop display of the real-time location of all the transit vehicles operated by the regional transit carrier. The second project is Transit Watch, which is a real-time arrival prediction system suitable for deployment in transit centers. Both of these applications are designed to operate over the Internet as Java Applets and are designed to be sufficiently general so that they can easily be ported to other cities (Dailey, 2001).

The first project, BusView Plus, implements a Java applet to display real-time transit vehicle locations on a variety of computing and operating system platforms. BusView is platform-independent with the goal of making transit vehicle location information accessible to anyone on the Internet. An additional goal is to develop an interactive interface that promotes modal change and encourages the use of transit (Dailey, 2001).

The second project called Transit Watch deployed an APTS/ATIS that provides a prediction of the arrival status of scheduled transit coaches. This prediction is categorized into four stages: on-time, delayed N minutes, departed, and no information. The database engine is central to the application. The data storage, relational queries, and serialization capabilities are used to synchronize the Predictor and Display Server components. The database stores information from the AVL stream necessary for the Predictor to make arrival time predictions as well as storing the resulting predictions for use by the Display Server (Dailey, 2001). Once reliable predictions are available in the database, the second specialized component of the Transit Watch architecture is used to distribute the information either to applet clients at transit centers or on the Internet (Dailey, 2001).

An interactive trip planner for users of PENTRAN, public transit for the Peninsula region of Virginia has been created. The interactive routing feature, or trip planner, allows a user that is unfamiliar with the PENTRAN routes and/or service area to plan a trip from their origin to a specific destination. The user will enter their origin and the routing program will calculate possible trips between the origin and destination by examining possible bus stops within a user-selected radius of the origin and destination. These trips will then be presented to the user to select and view a particular trip. Once a trip is selected, a trip itinerary will be presented to the user with all of the necessary travel information. Also included are a trip map that shows the desired bus stops for pickup and delivery

and their relation to the origin and destination through the use of a street network (Smith et al, 1998).

In Helsinki, Finland, the metropolitan area public transportation services has become more popular with the implementation of a system called mTravel to inquire about the routes and timetables. The system enables the user to search a route by entering two addresses. This facility tells the user immediately which public-service vehicle to get on to reach the desired destination. The user can also choose a certain point of time, on the basis of which the route service provides the next departure time of the public-service vehicle. The time can be the departure time of the vehicle or the time by which the user of the service wants to reach the intended destination (www.Novogroup.com).

The journey-planning algorithm compiles the best route at any given time by combining public transport connections and walking links, based on digital road networks. mTravel can also accommodate real-time positioning feeds from vehicles. This information can be used to correctly estimate arrival and departure times (www.Novogroup.com).

In 1996 the Copenhagen Transportation Company implemented a project called Priobus involving the real-time tracking of buses throughout the metropolitan area. The system, which was developed and implemented by the Dutch company Peek Traffic B.V., helps dispatchers prioritize buses moving about the

city and track arrival information via dynamic displays to passengers waiting at bus stops (www.esri.com).

Implementing the Priobus project included continuous accurate real-time localization calculated with on-board computers in the vehicles using a Digital Global Positioning System (DGPS). The information is communicated using radio transmission to the central system. All configuration and real-time information is stored in an Oracle database. The database is used to import new schedules that are automatically downloaded to the vehicles and to create reports with statistical information. In order to provide intuitive control over the system, a geographical interface was implemented using MapObjects (www.esri.com).

In South East Queensland, Australia, a journey planner has been implemented called TransInfo. It provides integrated public transport information on buses, ferries and trains. The service provides callers with detailed information on timetables, routes, stop locations, fares, connections, special events and other service provider information. The TransInfo journey planner helps on to organize travel times and explore alternate trip options by providing users with travel fare information on public transport for a selected journey. (www.esriau.com)

This area is also provided with a trip finder that is available through the Internet, kiosks, and mobile phones (using Wireless Application Protocol (WAP) and Short

Message Service (SMS) technology). The Tripfinder provides route planning and timetable information direct to public transport travellers. The user must choose an origin and a destination location and select a departure day and time from lists provided. The route planning and timetable algorithms have been developed based on predetermined routes to the list of origins and destinations (www.SydneyBuses.nsw.gov.au).

From each of these transit information systems mentioned, a strong conclusion can be made in that the uncertainty in the estimation of travel time is never a factor in the development process. While the simplest systems use a telephone system to let passengers know how the bus is running, the more complex systems rely heavily on the technology of AVL or GPS. This technology is put in place to allow for real-time placement of transit vehicles. A user can track the bus, but not be aware of travel time conditions. In other cases, a prediction method or a filtering system is used to estimate when a bus will arrive at the next transit stop. The prediction may be accurate but what happens if this bus never makes it to the next transmission point. By not considering the uncertainty in the travel time a bus can still be predicted incorrectly.

The system developed here is also building upon those systems that do not rely heavily on technology. Some agencies do not have the budget to acquire the latest technology, so they must design and adapt accordingly. By considering the advantages and disadvantages of each system described, an improved

methodology can be used to develop a well thought out system that can be applied to cities without the latest real-time technology. More options are also made available to the users within the user interface and the process of solving the optimum route considers uncertainty. As technology changes and becomes more affordable, this system can be updated.

Careful consideration is also required in making sure the system is generic in nature. While NextBus is widely used, many of the other sytems described were designed to serve a specific purpose of a particular city.

Overall the developed transit information system entails the latest advances in GIS technologies available. By applying the Bayesian statistical decision theory, the uncertainty in schedule adherence and travel time is considered. Finally, the system accounts for complexities of transit operation by the system.

Chapter 3: Building Blocks of Transit Information Systems

3.1 Navigational Process

3.1.1 Path Finding Methods

Whether walking through a building, driving in a city, or selecting routes on public transit, a traveller develops a mental image of the surrounding environment, of his or her own location within the environment, and of his or her progression through the environment to the desired destination. Such knowledge is acquired in three basic stages: (TRB, 1999)

In the first stage of path finding, a person identifies landmarks and begins to orient himself/herself using these landmarks as references. As landmark knowledge develops into route or procedural knowledge, a person starts to build travel directions and decisions around the framework of landmarks and can visualize travel plans as a series of actions that will take him or her from an origin to a destination.

Finally, with enough navigational experience in a particular environment, a person will develop a mental picture of that environment, including knowledge of the landmarks, the routes

from any one place to another, and approximate or relative distances between them. This survey knowledge allows travellers to describe routes they may never have traveled, by defining them in terms of this picture.

Although survey knowledge can be developed eventually from route knowledge and real-world navigation through the environment, it can often be acquired more quickly from map study. A printed map of landmarks and the spatial relation-ships between them helps to form the picture in a person's mind (TRB, 1999).

3.1.2 Requirements to Navigate Unfamiliar Journeys by Transit

Like any path finding experience, navigation through a transit system involves the three stages: orientation via landmarks, development of route knowledge to travel between those landmarks, and, finally, survey knowledge of the transit system. Transit information aids must translate the many elements of a transit system (the geography, connections, operations, and rules) into a base of knowledge that will allow a rider to identify and make decisions about the routes, transfers, and boarding and disembarking locations that will deliver them to the correct destination (TRB, 1999). User-friendly transit information aids provide this information in a way that allows the rider to travel confidently and easily through the various segments of a trip.

Ideally, passenger information should be available at every stage of the rider's transit trip. Pre-trip information helps the rider to plan routes and connections. In-transit information assists the rider at each decision point during the trip. Supportive/confirming information repeats and reinforces data and decisions and helps the rider to feel more confident that he or she is progressing toward the desired destination (TRB, 1999).

As defined by the Transportation Research Board, the pre-trip information requires the following:

- Location of the nearest bus stop,
- Routes that travel to the desired destination and transfer locations,
- Fare,
- Time of departure and approximate duration of the trip.

While in-transit information needs consist of the following:

- At the departure point: identification of the correct bus to board,
- On the bus: identification of bus stops for transfers or disembarking,
- At transfer points: how to transfer to another route; cost, time limits, and restrictions; and identification of the correct bus to board,

- At the destination: area geography and return trip information.

3.2 General Principles of Transit Information Design

Transit information aids are designed wherein users can learn transit routes, itinerary, and other relevant information like ticket costs very quickly. Users can educate themselves on transit information through the repetitive motion of viewing transit maps, speaking to others or through the media. This helps reduce the vagueness of how a transit system works. Transit information should be simple and easy to understand. Common names, terms, and references to known locations make it easier for users to navigate themselves through the transit network. When designing multiple aids, consistency should be maintained so the user is not confused. The rider should be able to build on initial information with data that confirm or reiterates the trip information (TRB, 1999).

Transit information can be presented in various ways like oral instruction, printed maps, and signage at bus stops or on buses. Each type of aid has benefits and drawbacks (see Table 3.1). Although no single information aid can meet all of the information needs of transit passengers, a combination of information types will accommodate different learning styles, different levels of transit experience, and different stages of a rider's transit trip (TRB, 1999).

Table 3.1: Types of Information Aids (TRB, 1999)

Information Aids	What They Provide	What They Don't Provide
Oral Instructions (Telephone information, bus operator, other passengers)	• Straightforward and personalized information. • Simplicity for new riders and for those who have difficulty reading maps. • Instant accessibility.	• An overall picture of the transit system. • Reference material for future or continued travel. • Flexibility or easy error correction; if a rider misses a step in the process, his/her frame of reference is lost unless he/she can converse further with the information source.
Maps	• "Bird's-eye" view of the transit system; spatial relationships of landmarks, routes, and connections. • Flexibility for changing trip plans. • Supportive information during a trip. • "Portable" information, useful for pre-trip and in transit.	• Instant accessibility. Not only is the map a physical object that a potential rider must obtain before trip planning can begin, but map reading presents difficulties for many people.
Signs	• Information at "decision points": bus stops, transfer points, terminals. • Supportive information.	• Detailed information and explanations. • Portable information; no help during pre-trip planning or on-board.
Timetables	• Portable information. • Detailed route information.	• Instant accessibility. Many riders have trouble reading and using timetables.

Although some transit riders use maps and timetables readily and regularly, many find difficulties in reading or understanding pieces of the information. The

information provided by the transit system's information operators or by automated recordings updates users on information pertaining to static maps and schedules. For many riders, the telephone will remain their primary choice for obtaining information about new trips. Even with the best system of signs and rider information, some still find the principal source of information about the transit system remains the bus operator. People with visual disabilities, cognitive problems, or functional illiteracy rely on the operator for route guidance, transfer information, and times (TRB, 1999).

3.3 Traveller Information Systems

Traveller Information Systems apply many technologies to allow customers to receive roadway, transit network, and other information important to their trip. This information assists the customers in selecting their mode of travel (car, train, bus), route and departure time. Transit schedule and status information may be obtained from Transit Management Systems. Most of the roadway-based information is collected by surveillance equipment (vehicle detectors, cameras, automated vehicle location systems) and is processed by computers in transportation management centers for further distribution to traveller information systems. Other information used in a traveller information system are map databases, emergency services information, and information on motorist services and tourist attractions and services (USDOT and FTA, 1998).

In the beginning of public transit, trip planning involved a person and a bus timetable. This timetable was divided into different stops and all of the different times of day a bus would be at that stop. Transit schedules fluctuated based on the current day of the week, so usually more than one timetable was needed. In the case of a large transit system, it was impossible to include every stop on the schedule, so occasionally only high volume stops would be given. A user had to know the physical location of a particular stop, relative to those listed on the schedule if it was not a high volume stop. In addition, the user would have to approximate the time, since the stop was not listed on the schedule. Of course, this assumed the user knew which stops to use and any route transfers that might be needed. Thus, route maps became important so a user could locate an origin and destination, and determine the needed routes for the trip. In the case of large transit systems, the map could be cluttered and confusing, and thus would have to be split into sections. What was needed was a single map, where a user could zoom in and out of view and not lose sight of the big picture. With the growth of the personal computer, this suddenly became possible. In a GIS environment, a user can search a route map of all available stops, with tools such as zooming and panning included. As transit systems have developed and become more complicated, the need has increased for a computer program to analyze the information quickly and accurately (Smith et al, 1998, Culp, 1994).

GIS can serve a variety of purposes for a transit provider in addition to trip planning. For example, the analysis of census data can benefit the overall transit

planning of the agency, including the determination of optimal locations for bus stop shelters. Executing an interactive trip planner is just one of the many applications of GIS, making it a worthwhile tool for any transit system (Attanucci and Halvorsen, 1993).

Traveller Information Systems provide travellers with information on one or more modes of transportation to facilitate decision-making before their trip and during the trip (en-route). Information can be provided to trip makers at home, work, transportation centers, wayside stops, and on-board vehicles. With links to automatic vehicle location, traveller information systems specifically for transit are beginning to provide ready real-time information, such as arrival times, departure times, and delays (Marks, 2001). The performance of such systems can be improved if models of bus delay are used to supplement AVL/C data (Abdelfattah and Khan, 1998). Travellers can access this information through a variety of media, including telephones, monitors, cable television, variable message signs, kiosks, and personal computers (Casey et al, 2000, Marks, 2001).

3.3.1 Traveller Information System Functions

Trip planning information and assistance may be provided pre-trip or en-route. Pre-trip planning assistance provides travellers with roadway information, including road condition, traffic information and travel times, and transit information which can be used to select route, mode, and departure time. This support may be requested from home, the workplace, park and ride facilities, transit stations, and other locations. En-route traveller assistance provides the traveller with roadway and transit information while travelling, including traffic information, roadway conditions, transit information, route guidance information, and other information such as adverse travel conditions, special events, and parking (USDOT and FTA, 1998).

3.3.2 Pre-Trip Information

Traffic volumes in cities across the world are consistently increasing every year, while transit ridership remains relatively static or is decreasing. Getting travellers to give up the convenience of driving their own cars is a difficult task. In doing so, it is essential to focus on a change in travel behavior, which entails providing accurate and timely information to travellers before their trip. This information will enable the traveller to make an informed decision about mode(s), as well as route(s) and departure times.

Pre-trip traveller information systems help travellers make decisions about the choice of transportation mode, route, and departure time before they begin their trip (Casey et al, 2000).

Pre-trip planning systems provide travellers transit information that can be used to plan a trip, including which route to take and what time to depart. Whether provided to travellers at home, the workplace, park and ride facilities, or transit stations, pre-trip traveller information can help relieve congestion by allowing travellers to make informed decisions. Pre-trip traveller information also supports itinerary planning, which can provide information on the entire trip from one point to another. In providing pre-trip traveller information, the focus is on travel behavior and decision support. This requires providing accurate and timely information to travellers before their trip. Touch-tone telephones, personal computers, personal digital assistants (PDAs), kiosks, and automated data retrieval systems that augment existing human-operator interfaces have the potential to substantially improve the accessibility of traveller information, thus influencing traveller behavior (USDOT and FTA, 1998, Marks, 2001).

Automated data retrieval systems to expand existing human-operator interfaces can provide information to the caller in a timely

manner. These media allow transit users to have good connection with transit information center. The Internet may be the most promising one because of its growing popularity and low cost. But many of transit users do not have Internet access. So the accessibility is an issue for Internet for the time being. The kiosk is good to deliver pre-trip information as well but with a higher cost. The combination of Internet with kiosks would increase the accessibility of Internet and enhance the information contents of kiosks. Telephones have good interactive function through the oral communication between passengers and telephone operators. It is a convenient tool especially for visual impaired people, but there is a high cost associated with a customer service agent. An automatic message center with real time connection with AVL information is a lower cost alternative (Peng, Jan, 1999).

Pre-trip information systems in the past were primarily directed towards riders who knew the transit system already, and only wanted updates on schedules and transfers. New systems provide relatively obtainable information to the beginner as well as the experienced transit passenger. Typically, these systems can be accessed by a touch-tone telephone. Newer systems include map displays of the service area based on geographic information systems (mostly for the operator providing information to the

customer who has requested information) and schedule information that is provided on the Internet through the World Wide Web. These newer systems, whether accessed directly or through a customer service operator can locate the closest transit stop to the caller's origin, provide directions to this stop, identify the closest stop to the caller's destination, provide detailed directions on transit between the origin and destination stops, and provide directions to the final destination from the last stop (Casey et al, 2000, Marks, 2001).

With the expanding implementation of Automatic Vehicle Location systems, the potential for providing real-time information on bus arrival and departure times is slowly becoming reality. Real-time schedule information is beginning to be provided by several transit agencies through a variety of media, including electronic signs and kiosks.

3.3.3 Effective Traveller Information Systems and Benefits

Effective traveller information systems should provide information that is timely, accurate, reliable, and relevant to making travel decisions. For travel information to be useful, it needs to be current and received in time to allow a user to act on it (e.g., route, time). Travellers also want and need to know that the travel information is

correct, whether it concerns the arrival time of the next bus or the location. As well, potential and current users want information to be available when needed, and are consistent in quality. The information provided should reflect the entire service area and it must contain sufficient details about locations, times, and possible alternates to be useful in planning travel. The system should be easily integrated with other ITS systems like emergency management, freeway management, and traffic signal control to obtain adequate traveller information. In designing a system, the developer must consider the user friendliness and access capabilities. Traveller information should be accessible in a variety of forms and locations such as personal devices, kiosks, and home computer and be easy to use, since users perceive access time as an additional cost associated with using traveller information. Finally, it must be easy to maintain and not require excessive costs and time to operate (USDOT, 1999).

In general, each traveller information system must be designed to meet the objectives of users within an urban area. Overall, traveller information systems attempt to reduce travel times and delays to users. In doing so, traveller stress for trips to unfamiliar destinations can be reduced. In many cases traveller information systems have demonstrated benefits in several areas including

travel time, consumer satisfaction, on-time performance, and environmental impacts (USDOT, 1999). Although users are receiving the necessary information in order to plan a trip within an urban area, improvements would be desirable.

3.4 Transit Service in Ottawa: Information System and Other Developments

OC Transpo has long been recognized as a leader in the public transit field. Its ridership levels are the highest among mid-sized North American cities, and its innovative Transitways have allowed the cost-effective introduction of rapid transit services throughout the region (OCTranspo, 1999)

OC Transpo provides transit service to the communities within the City of Ottawa, which include Nepean, Vanier, Rockcliffe, Gloucester, Kanata, and Cumberland. Some OC Transpo routes also operate to downtown Hull, Quebec. OC Transpo serves a population of 625,000 and on a typical weekday, ridership averages about 302,000 person trips. During a peak hour one-way trip, OC Transpo can carry approximately 10,000 passengers at the busiest location on approximately 190 buses per hour. Most of these operations occur on exclusive, grade-separated 80km/h busways (OCTranspo, 1999).

Its operations include 878 buses, distributed over 204 routes, and provide 5,500 bus stops. OC Transpo's Transitway system is by far the most comprehensive

bus based Rapid Transit System in North America. It provides a 31-kilometre Transitway that runs across the city with 28 bus stations and 5 light rail stations. It also has 3 km of mixed-use traffic in the downtown area, 21 km of freeway shoulder lanes, and an additional 10km of exclusive bus lanes and 32 stations (6 in CBD). Currently OC-Transpo is expanding the length of the Transitway to include areas outside the greenbelt. Along the Transitway there are 5 park-and-ride lots: Baseline, Greenboro, Place d'Orleans, Eagleson Rd/Kanata, and Fallowfield Rd/Barrhaven (www.OCTRANSPO.com). For more information on OC Transpo's operating statistics, please refer to Appendix G.

With an aging population and the changing nature of work, OC Transpo has struggled to keep its ridership. There has been a reduced significance of employment in the downtown core and the growth of suburban business parks that has led to OC Transpo's expansion in order to serve this population. The number of buses is up more than 6% in the last 5 years along with an increase of 35% in the number of routes. This value is mostly due to the introduction of school-service routes to decrease the congestion on regular routes. The number of bus stops has increased 10% due to the development of new suburban areas in the east and west of Ottawa. OC Transpo has been committed to the goals of providing economic, dependable, customer-friendly service from all levels and components of the organization and hence has seen an increase in annual ridership of 7% in the last 5 years (OCTranspo, 1999).

Currently, OC Transpo has setup a daily timetable for each bus route within its service area. Lists of buses, schedules and maps are presently available in paper format and on the website. These times are to include an estimated arrival time at each bus stop location. Also there is a telephone information system (the 560 system) for public use in order to update their schedule. In this system, a recorded message informs the users and announces the estimated arrival time for each bus stop location. The current interactive information system, run by OC Transpo, is a telephone and email service. Users are able to gain desired information regarding bus routes and schedules by speaking directly with telephone operators or sending them requests through emails. First, the users provide their starting point (origin), where they want to go (destination), and what time they want to travel. The operators, using information available to them and personal knowledge, find the most suitable route and travel time for the caller.

3.4.1 Long-Term Strategy

The long-term objectives for OC Transpo support the policies of the City's Official Plan (OP) and Transportation Master Plan (TMP) to reduce Ottawa-Carleton's dependency on private automobile use and promote more environmentally friendly modes of transportation. To meet these objectives, the City has adopted regional modal share targets that would significantly increase the proportion of trips made by more sustainable transportation modes and reduce dependence on automobile travel (OCTranspo, 1999).

3.4.1.1 Market driven and customer oriented services

Transit cannot be all things to all people therefore OC Transpo, like any other agency needs to give priority to key markets and provide effective services to them. Emphasis should be placed on providing a dependable basic system, supplemented with specific services emphasizing speed and convenience in markets where transit can compete with the private automobile. OC Transpo has laid out a long-term strategy that involves changes to the route structure; these are listed below:

- The development of an effective base network providing priority service on major arteries to address suburban trip patterns and to serve cross-town trips not necessarily oriented to the Central areas.

- More community oriented services that provide feeder service to/from the base network and Transitway stations.

- More express and limited stop services to major activity and employment centers located in suburban locations.

A second feature of the service strategy is to accommodate the travel needs of an aging population and an increasing number of

people with disabilities with an integrated fixed route/paratransit system. Low floor, accessible buses, accessible light rail vehicles, accessible Transitway stations and paratransit services designed to feed into the conventional system should help promote the integration and efficient use of the fixed route and paratransit services (OCTRANSPO, 1999).

A third feature of the service strategy is the transit priority network for all of the bus routes comprising the base network. This faster and more reliable service can be achieved through traffic signal pre-emption, responsive signal phasing, queue jump lanes, reserved bus lanes, automatic vehicle location and control and advanced passenger information (OCTRANSPO, 1999).

Enhanced passenger information systems will provide pre-trip information to plan a trip, bus arrival time at stops, next stop information, and other data. This will complement the "560" telephone system and traditional transit shelter maps, printed schedules, and ride guides (OCTRANSPO, 1999).

3.4.1.2 Route structure

The transit system currently consists of a hierarchy of routes that brings most urban residents within 400 meters of a transit route

(www.OCTranspo.com). The strength of the system is the route structure that provides basic region-wide mobility for residents, local services for communities, and express services for commuters. The Transitway and base routes in the main travel corridors, the community services supporting these routes, and the extensive express bus network throughout the region are the main reasons for the transit system's high ridership when compared to similar sized cities in North America (OCTranspo, 1999).

OC Transpo has expanded its route network in the suburbs to keep pace with the decentralization of population and employment, but has had to offset the increased service hours and costs by stretching its services in the older more established parts of the service area where ridership is highest. As a result, the transit routes have become more oblique, the transit services have become less frequent, and running times have become tighter and more difficult to control over the years (OCTranspo, 1999)

3.4.1.3 Smart systems

Recent advances in the effective use of information technology for transit referred to as 'smart systems' have provided a relatively low cost means for increasing and sustaining ridership. OC Transpo has already begun to implement smart system enhancements. An

action plan to cover the next five years has been developed and addresses the following:

- Electronic fare collection: A single "smart" card could replace paper tickets and passes, as well as reduce the volume of cash and paper transfers used.

- Fleet operations management: A vehicle location monitoring system would be more cost-effective to operate than OC Transpo's current system, it would support the ParaTranspo fleet, and it would provide the basis for improvements in on-time performance, through real-time fleet location information.

- Traffic signal priority: Along certain routes, buses tend to be delayed due to unpredictable traffic problems. Adjustments to the traffic signals, to provide a longer green phase can be essential in restoring on-time performance.

- Passenger information: Enhancements to OC Transpo's current techniques for communicating passenger information, including the "560" telephone information system, the website and the station displays will be made. In general, additional information will be provided to passengers before and during their trips at an

expanded range of locations. The station displays will provide ongoing updates to passengers about expected arrival times while they are waiting at major stations and terminals.

3.4.2 Automatic Passenger Counting (APC) at OCTranspo

3.4.2.1 Introduction

While many transit companies have found the idea of Automatic Passenger Counting tempting few have pursued this technical challenge and even fewer have been successful. As stated by Brian Barclay at OC Transpo, APC has been in full operation since 1986, collecting data on passenger movement and running times, and storing the data in a flexible database for reporting. Planners, Schedulers and Managers have a wide variety of APC reports at their disposal.

3.4.2.2 The System

The system uses a fleet of approximately 80 specially equipped buses and more recently Ottawa's own light rail service has been added. The APC fleet represents about ten percent of the total OCTranspo fleet, with each type of vehicle represented to ensure APC operation is unrestricted. Using a sample based approach, adequate data are gathered from these vehicles during one of four service periods that divide the year. Samples rates are often over

ninety percent. Many of the trips are captured multiple times, and with close to 7,500 scheduled one-way trips each day, this translates into over 21,000 trips in the database. Assignments for APC buses are prepared the day before using an automatic assignment system that weighs user needs and requests against the vehicles available for the next day.

Each data bus (or train) has been equipped with infrared sensors that monitor doors for passenger movement. Passengers are counted as they enter and exit at each stop; time spent in this exchange is also recorded. Vehicle location is tracked by tapping into the vehicle odometer so that relative distances can be recorded with each activity. Later during data processing passenger activities stored in memory are associated to actual bus stops by using distance and interpreted bus movement combined with a strict set of parameters. This "to the bus stop level" reporting offers data users the greatest possible detail for their work. To assist with vehicle location accuracy, several on street microwave transmitters are used to add location benchmarks to the data as the bus travels its route. Coded signals are received by the buses roof top receiver. GPS is being phased in to replace these transmitters. This change in technology will add even greater accuracy to location work.

Each night data are uploaded from all APC vehicles using an Automatic Radio Transmission Systems (ARTS). Data are also processed overnight. Processing puts captured data through very tight quality checks to maintain a high level of reliability. Reports can be run by the APC system staff each morning although, most requests use a completed service period to give a larger sample.

Passenger counting is only part of the work done by APC. Equally as important is APC's ability to report on running times collected by the system. The running time between any two bus stops on a route or collection of routes can be checked for any time period. Time utilization between stops can also be analyzed showing traffic congestion and time wasted; this is done using specialized logs (data records).

Schedule adherence reporting is yet another of the performance monitoring roles APC performs. APC also reports on revenue/cost ratio statistics. By adding the cost of service to the ridership and fare information, planners are quickly supplied with data that highlights those trips that are not operating efficiently. APC has gained acceptance through its demonstrated ability to provide timely input for solutions to practical problems.

Chapter 4: ArcView GIS and Network Analyst

4.1 Introduction

Transportation data providers, including government agencies and private organizations, are discovering the convenience of publishing and disseminating transportation information with the use of a geographic information system, like ArcView GIS.

As defined by the developer, Environmental Systems Research Institute, Inc. (ESRI), ArcView3.2a is a desktop geographic information system (GIS). With ArcView3.2a one can create intelligent, dynamic maps using data from virtually any source and across most popular computing platforms. ArcView3.2a provides the tools to let one work with maps, database tables, charts, and graphics all at once. It enables one to link information to location (it connects the what to the where), allowing one to see and analyze data in new and useful ways.

Below is a typical ArcView3.2a project, which was developed by the author. A map, chart, and table have been used to depict route characteristics and to illustrate general locations throughout Ottawa.

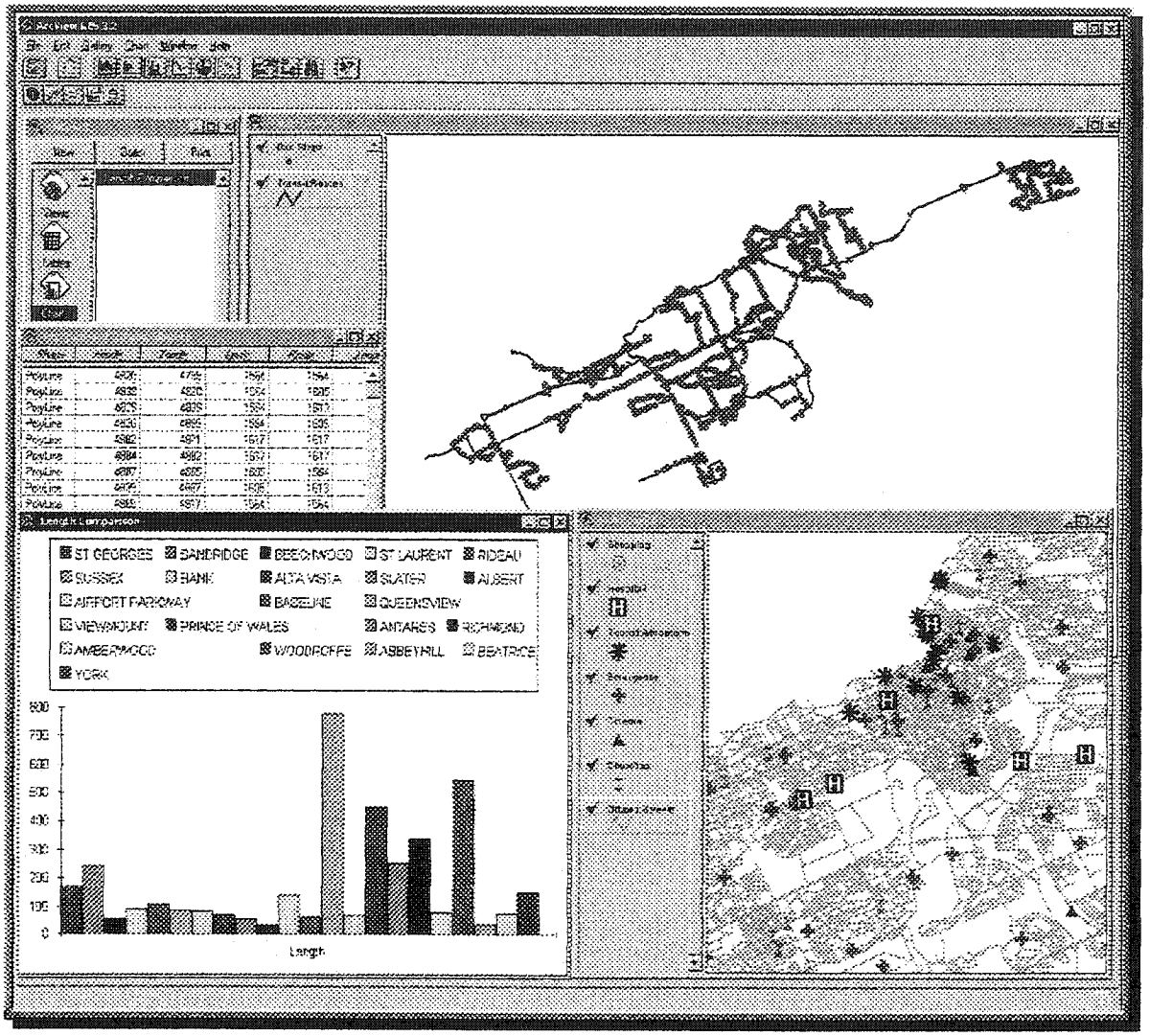


Figure 4.1: ArcView3.2a GIS Interface

The ArcView3.2a interface consists of windows that present information in different ways. Rows of menus, buttons, and tools at the top of the main application window allow you to view and perform analytical operations on the data in the database.

ArcView3.2a has a number of features that make it an excellent tool for learning and using GIS. Unlike most GIS software of the past, ArcView3.2a provides a graphical user interface. This provides some benefits as defined by ESRI:

- No need to remember thousands of commands and command arguments.
- Most commands and operations are available by navigating the ArcView3.2a menu and dialog structure.

Only recently have computers become cheap and powerful enough to supply the resources needed for a fully functional GIS on a typical office or school desktop. At the same time, much effort has been undertaken by software manufacturers to increase the functionality and ease of use of their products. In the recent past, GIS software could be used only on high-end UNIX or VMS workstations. Today, PCs are powerful enough to handle GIS applications, and the software is easy enough to get started. It is possible that a user can become productive within the first few hours of using ArcView3.2a (www.esri.com).

ArcView3.2a has a strong programming language called Avenue. These scripts (algorithms) can be used to automate complex or repetitive tasks, or to create entire applications.

Several extensions are shipped with ArcView3.2a, and others are available for free or at additional cost. These extensions increase the functionality of

ArcView3.2a by adding features commonly found in other GIS software such as the Network Analyst. Some of the built-in optional extensions are used for direct use of CAD data, digitizer support, projections, and modeling. It is also possible to create personal extensions to run custom tasks or applications.

4.2 Customizing with Avenue

ArcView3.2a can meet a wide variety of GIS needs as stated earlier but it can also be customized. This means that the user interface can be modified for personal or public use. Examples are noted below (ESRI, 1996):

- Buttons not required can be removed,
- ArcView scripts can be accessed (created either by the programmer or obtained from other application developers) along with other applications such as Netscape and Windows based programs,
- New elements that are either specialized for a particular application or that give access to additional code that is not accessible within ArcView can be added.

The customization process is relatively simple. Changes made to the ArcView3.2a interface can be applied to a specific project, a user's default project, or to the system default ArcView project. This provides a great deal of

freedom to ArcView3.2a programmers in developing applications that fit the needs of system users (ESRI, 1997).

When customizing in ArcView3.2a, Avenue is used to accomplish different tasks. One can use Avenue software to customize the standard graphical user interface (GUI) that comes with ArcView3.2a. For example, one can reorganize the controls (menus, buttons, tools, and popups), change text or icons, and add or remove choices. With Avenue, new functions for a specific application can also be created. One can combine a series of steps frequently used into a single click of a button. Avenue can also be used to develop a complete application that has its own GUI.

Avenue gives an easy-to-use framework for customizing controls and creating new functions. Dialog boxes can be created to modify controls and the Script Editor to write Avenue programs, called scripts. The scripts contain code that executes a new function. By using controls and scripts together, a new ArcView3.2a GUI is built (ESRI, 1997).

Avenue is able to perform all of these tasks because it was created as an object-oriented programming language. An object is an element, such as a view, theme, button, or a symbol, that one works with in ArcView3.2a. Objects with common characteristics belong to the same class. Each object is associated with a set of actions or requests. For example, requests for a layout object

include opening, closing, and printing. A marker symbol object has a different set of requests, such as setting marker size and colour. If one can identify objects and their associated requests, one can write Avenue scripts (ESRI, 1996).

Object oriented programming addresses many problems inherent in the procedural approach to programming. The principal advantage of the object-oriented language is its ability to handle complexity in a clear manner. Using this programming, functionalities are placed inside the objects and out of one's sight. Consequently, to execute a specific functionality, one simply makes a request to the object (ESRI, 1997). This type of programming eliminates redundant code. Also, this saves time by building a program with objects, and that the quality of the programs improves because object functions are protected from the code that is written.

4.2.1 Network Analyst

Within ArcView3.2a, extensions can be installed for greater analysis than a basic query. One of these extensions is Network Analyst. This extension enables the user to find the best route, closest facility or service area. In ArcView3.2a, any system of interconnecting line features is a network. The user chooses the points of interest and manually steps through choosing specific constraints and options. ArcView3.2a uses the information and

creates a Network theme be it a line or a polygon, to be used for interpolation.

When the Network Analyst is installed there are several new Avenue classes and requests added for solving network problems. These classes and requests can be used to automate tasks, add new capabilities, and build applications. Each of these can be changed or manipulated through the user interface in order to solve different and complex network problems (ESRI, 1997).

The Network class is the primary class used in solving network problems. It represents the network problem to be solved. Each of the problem solving requests operates on the Network class, which include finding routes, closest facilities, and service areas. Besides representing the network problem, the Network class can have several properties associated with it, such as cost fields or a customized search tolerance. There are several requests that operate on the Network class to set and retrieve these additional properties (Meyer and Miller, 2001).

The Network Analyst can solve network problems on any theme containing lines that connect. This extension is designed to help

use networks more efficiently. The theme can be a shapefile, an ARC/INFO coverage, or a CAD drawing (Meyer and Miller, 2001).

Network analysis is one of the most frequently used components of ArcView3.2a. Once a road system has been digitized, a GIS can quickly be used to answer questions such as "what is the quickest way from A to B?". It can be used to model the real world conditions and hence provide the shortest or quickest route as long as the network, calling points (stops) and the cost of traversing the individual links are identified. Two parameters normally define the cost of traversing a link, the length of the link and the time taken to traverse the link. The response algorithm can be made more complicated by adding impedances or "costs" onto major roadways. These may include speed restrictions, one-way systems, or temporary blockages caused by accidents or road repairs (Aronoff, 1995). Figure 4.2 summarizes the simple network operations of routing a vehicle or person through a street network. Based on predefined costs assigned to route segments, the illustration indicates the optimum path to travel.

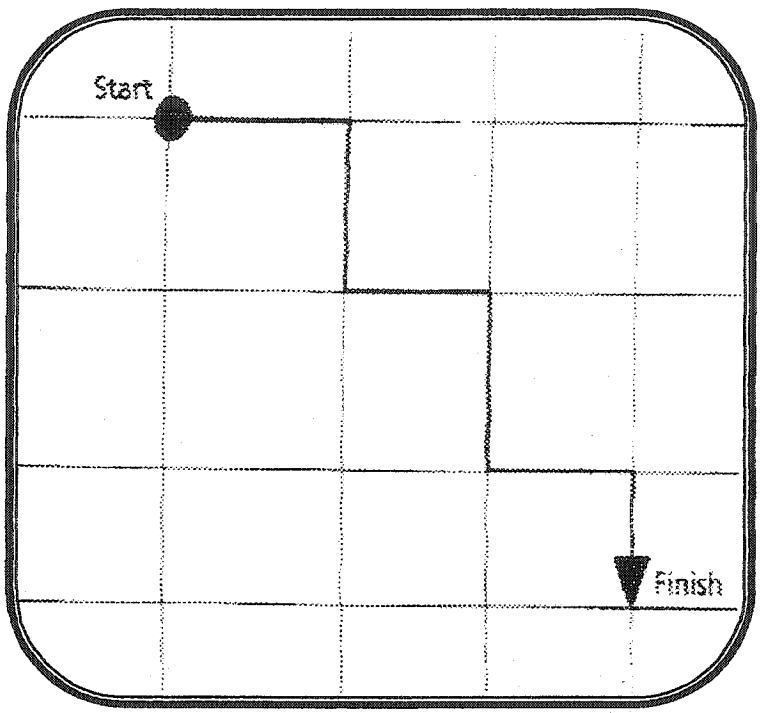


Figure 4.2: Shortest Path Routing (Aronoff, 1995)

Network Analyst supports different objectives, such as travelling quickly, travelling by the shortest route, travelling by the most interesting route or any criteria the user specifies. Network Analyst only needs a specified cost field in the line theme's attribute table when solving the problem. The cost fields that are used can be in any units of distance or travel time, such as minutes for finding a route that minimizes travel time, or kilometers for finding a route that minimizes travel distance.

An example of an application of a geographic information system's network analysis is a car navigation system based on GPS, designed to give up-to-date information as to how to avoid the

latest construction or accident hold-ups. As well, it could keep precise knowledge on the locations of fire engines and ambulances so that the nearest vehicle can always be dispatched and response times improved. Coupled with navigation systems, these allow the drivers to find the quickest and safest routes to accidents or emergencies given the latest local traffic conditions.

4.2.2 Route Finding

The obvious approach to finding the shortest path to travel is to examine all the possible paths. A large network contains so many possibilities that it becomes impractical. Several algorithms have been developed to find the shortest (or least cost) path through the network. Network Analyst uses the most well known of these, developed by Dijkstra. In solving for the optimum route, ArcView scans the network for connectivity and travel cost relationships among the nodes. Once it has this information it derives the optimum path.

Route planning is a process that helps a person plan a route prior to or during a journey. It is widely recognized as a fundamental issue in the field of vehicle navigation. A variety of route optimization criteria may be used in route planning. The quality of a route depends on the distance, the travel speed, prohibited turns,

one-way streets and the number of stops. All of these factors are referred to as travel costs. If the existing network has these rules of the road, they should be represented in the roads' line theme through cost factors before the Network Analyst is used. This will enable realistic solutions.

In the computer literature, people often refer to finding a route from point A to point B as the shortest path problem. By using Dijkstra's algorithm, ArcView can determine this least cost path between stops.

4.2.3 Dijkstra's Algorithm

Dijkstra's algorithm computes the shortest path from a given node, called the root node, to every other node in the network. A node exists wherever line segments connect. The set of paths found forms a sub-graph of the original network. This sub-graph turns out to be a tree (a *tree* is a graph having no loops in it). Hence, Dijkstra's algorithm finds the shortest path tree. (www.princeton.edu, 1959)

Consider a network, $R = \{N, A\}$ where N is the set of nodes and A is the set of arcs. Let a_{ij} denote the length of arc $(i, j) \in A$.

Dijkstra's algorithm is an iterative procedure where at each iteration, one new node is added to the shortest path tree. The process starts with no edges in the shortest path tree and with the root node as the only node in the tree.

At each iteration, an array is built that contains the shortest distance from each node to the root node that is subject to the constraint that all nodes in the connecting path belong to the path-tree constructed. Initially, the array is equal to zero at the route node, then it is set to the distance from the root node to every adjacent node within the route system, and all other node paths are set to infinity. (www.princeton.edu, 1959)

Also, there is another array that indicates the node in the shortest-path-tree to which each node j not yet in the tree has to be connected in order to obtain a path from the root node to this node. Initially, all adjacent nodes are equal to the route node, and the other j^{th} nodes are set to some prohibited value (used as a flag). (www.princeton.edu, 1959)

Until all of the nodes have been included in the shortest path tree, the following steps are performed:

- Loop through the nodes not in the shortest path tree to find the node with the smallest value of path length connecting it to the path tree. This node is now the new node. If its path length is infinite, then the network must be disconnected.

- The new node is added to the shortest-path-tree and an arc is added to the shortest-path-tree linking the new node back to the origin node.

- The path length and origin node are updated for every adjacent node that still needs to be added to the tree. Each new length is scanned for the lowest value. Once found, this adjacent node now becomes the new node, an arc is added, and the system advances until all nodes have been accounted for.

During the process, the program is determining the connectivity and travel cost relationships among nodes. Once it has the information, it derives the shortest path. This process can be done for any relevant cost field. (Ormsby and Alvi, 1999)

Chapter 5: Application Process and Algorithm Development

5.1 Overview

The overall goal of this project was to design a user interface that would enable a person to find the optimum transit route to travel from an origin point to a destination point. It would enable visualization of schedule information and link to OC Transpo's website. As well, it would account for uncertainty in schedule adherence. In performing this task, network and geographic information is required along with a GIS software program.

5.2 Software

The software utilized is listed below.

ArcView3.2a/Avenue: ArcView3.2a desktop GIS software provides a user-friendly Graphic User Interface (GUI). Avenue, as an object-oriented programming language, provides an application development environment. Developing a Transit Information System in ArcView3.2a utilizing Avenue can make full use of the existing capabilities of ArcView3.2a.

Dialog Designer Extension: ArcView3.2a Dialog Designer provides flexibility to design and build custom user interfaces for different data entry and for organizing functions of the transit information system.

Network Analyst Extension: Enables analysis of networks based on cost restrictions. It can solve common network problems on any theme containing lines that connect.

5.3 Geographic Data

5.3.1 Data Source

Before using the data each geographic file and database had to be created and/or prepared. Data were collected from the City of Ottawa and OC Transpo. This included full street network files both database and shape information from a compact disc provided by the city. These files were digitized by the City and contain service and characteristic information about the street networks. Bus stop locations, route layouts, schedules and timetables were collected from OC Transpo. Latitude and Longitude information pertaining to each bus stop was used to input the data into ArcView3.2a. The schedule information was entered into database tables. A small sample of landmark addresses was gathered from the phone book. All of these point locations were individually digitized into ArcView3.2a.

The following table provides the shape type for each map. The graphic and data information can be found in Appendix D.

Table 5.1: Shape Types of Maps Used in the Transit Information System

Maps	Shape Type
Ottawa Street Network	Line
Transit Route Network	Line
Individual Bus Routes	Line
Bus Stops	Point
Landmarks	Point

Once the data have been added to the view, the map units are set so that the interface will yield meaningful distance units. These have been set to 'meters'. In the project, there are seventeen themes, the street network, the transit route network, geocoded locations (cinemas, churches, emergency, hospitals, hotels-motels, libraries, museums, tourist attractions, schools, shopping malls, grocery stores, recreation centers), bus stops, an empty theme for input information, and a hotlink.

The geocoded addresses were created from a list of digitized points. This is a function under the Theme | Properties menu in ArcView3.2a. For further information consult the ArcView3.2a help menu. The empty theme is used when the user enters a location by an address and to join multiple locations through landmark selection. This theme is then geocoded for use in finding the

optimum route. The hotlink theme is used to link the user to OC Transpo's website.

5.3.2 Setting the Rules of the Network

Road networks have rules about how vehicles progress through them. For a given street network, roads may be closed, or one-way streets and speed limits in place. These could determine if a street is traveled or not. Therefore, if the Network Analyst is to solve a problem based on the existing street network, care is required in representing correct traffic control to enable a realistic solution. The cost fields are important because ArcView3.2a uses these to determine either the quickest or shortest route to travel. The cost field is assigned the units of distance (kilometers, meters, miles) or travel time (minutes, hours).

In the Network Analyst, the rules of the road are as follows:

- One-way streets
- Prohibited turns
- Overpasses and underpasses
- Closed streets or authorized vehicles only

The following sections define these rules and how they were entered.

5.3.2.1 Travel Costs

The units specified in the projection are used to specify length in the transit routes' and street's attribute tables. The script 'calalp.ave' (Appendix E) is used to input the length field into the tables. When the script is run, it calculates the length of each line segment and inputs a field named 'length' into the attribute tables. This script is an ArcView3.2a system script available for anyone's use.

A speed limit for each road was entered into the same attribute tables manually using the field calculator. This field is called 'speed_lim' and the units are specified in kilometers per hour. As well, a new field must be added called 'cfcc'. These are codes based on the type of road (freeway, two-way with median, residential) the speed limit corresponds to. All of the coding is available from a systems table (refer to Appendix F). A sample extension called Speed Limit Calculator applies speed limit information to a cost distance field (e.g. minutes, meters). By selecting the option of Speed Limit Calculator from the Network Analyst menu, this new field is created. The 'minutes' field represents travel according to drive time.

In addition to the above speed limit costs, the dwell time costs of opening and closing doors, boarding and deboarding passengers, and merging back into traffic was added on to the sections of the network that contained bus stops. As defined earlier in chapter 2, dwell time can be assumed 2 to 5 s for door opening and closing plus an additional time to be 60s for CBD, transit stations, or major on-time transfer points; 30s for major outlying stops; and 15s for typical outlying stops. These values were added on in accordance to the location of the stop along the route and the location within the city.

5.3.2.2 One-way Streets

The city of Ottawa has many one-way streets, especially in the downtown. Therefore this cost cannot be assumed insignificant. A field was created to represent streets that can only be traveled in one direction. This is a string field that is added to the roads theme. Its name must be ONEWAY in order for the Network Analyst to recognize it.

When setting one-way streets, the direction of the digitizing must be known. The direction of the digitization can be found by having the roads indicated by arrows in the view. Once this is known, the one-way field can be populated with the correct string information. If the

one-way street runs in the same direction as the digitization, then the field should have the value of 'FT'. If it runs in the opposite direction it has a value of 'TF'. 'F' corresponds to 'from' and 'T' corresponds to 'to'. If there is no entry permitted, it can be represented by using 'N'. Any other value or a blank cell represents travel in both directions (Ormsby, 1999).

5.3.2.3 Overpasses and Underpasses

Street themes contain many locations where one line feature crosses another. Some of these locations designate an intersection where it is possible to make a left or a right turn from one street onto another. Other locations designate an overpass or underpass, where in reality there is no physical connection between the two streets. Modeling overpasses and underpasses allows a definition of which streets are physically connected (Ormsby, 1999).

The overpasses in this network were not specified as a field in a table. This is because the map was digitized with nonplanar streets for overpasses and underpasses. This means that the streets do not have connections or nodes. This method involves two unbroken lines, one passes completely over the other to represent an overpass. Therefore it is not necessary to explicitly set them

because Network Analyst considers them not to be connected and does not stop at them to consider a route left or right.

5.3.2.4 Closed Streets

Streets currently closed to traffic or certain types of streets to avoid should be used so that the Network Analyst knows not to select them for travel. For this project, the closed streets cost was performed at the same time as the one-way streets' field was created. As mentioned under one-way streets, an 'N' was used to indicate a road is closed to traffic. This cost field can also be indicated as a separate field. The other methods used can be found by consulting the ArcView3.2a help menu.

Figure 5.1 shows a sample of the transit route attribute table showing the added fields. As seen in this figure, the fields created are: speed_lim, length, and minutes.

transit routes.dbf

Dirc	Speed_km	Length	Minutes	Coverage	F_elev	T_elev
A63	25	41.514	0.629	FT	0	0
A63	25	38.361	0.624	FT	0	0
A25	55	115.173	0.645		0	0
A63	25	46.273	0.636	FT	0	0
A63	25	47.187	0.637	FT	0	0
A25	55	237.769	0.728		0	0
A41	25	89.069	0.450	TF	0	0
A41	25	76.949	0.115	TF	0	0
A25	55	115.303	0.078		0	0
A41	25	81.243	0.438	TF	0	0
A41	25	105.045	0.474	TF	0	0
A63	25	229.026	0.909	FT	0	0
A41	25	68.446	0.102	TF	0	0
A25	55	144.935	0.098		0	0
A41	25	57.601	0.403	TF	0	0
A41	25	114.255	0.487	TF	0	0
A41	25	101.515	0.468	TF	0	0
A41	25	102.346	0.470	TF	0	0
A01	30	278.077	0.913	TF	0	0
A41	25	83.844	0.125	TF	0	0
A41	25	149.979	0.791	TF	0	0
A41	25	94.726	0.141	TF	0	0
A25	55	465.589	0.883		0	0

Figure 5.1: Sample Attribute Table

5.4 Purpose of maps

5.4.1 Base Street Network

The base street network provides the core geographic data for the transit information system. It is used to create individual bus routes, display street map to users, and provide geocoding capabilities for data preparation and for operation.

5.4.2 Individual Routes and Route Network

Like the base network, individual route information allows specific routes to be displayed as needed by analysis results. Individual routes are also used to create the entire route network and allow for easy maintenance of the network if one route from the network requires alterations.

Network Analyst was used to find the optimum travel path. Individual costs were inserted so that additional information would be included in the calculation so that the optimum travel path was implemented.

5.4.3 Bus Stops and Landmarks

By using the bus stop map, the location of the closest bus stop to the users' specified locations can be pinpointed. In addition, these points are used to display for users the places where they would start and end their trip in relation to their selected origin and destination. The maps of landmarks are not only utilized for displaying landmark locations, but also allow users to select their origin and destination location directly from the list of landmarks.

5.5 Data Preparation Problems and Solutions

As the project progressed various obstacles and data problems presented themselves. The problems encountered and the degree to which a solution was necessary to continue with the research project are discussed below.

5.5.1 Base Street Network

Although the Ottawa area street network from the city of Ottawa was acquired in September 2001, some of the data it contained needed to be updated and addresses fixed. A limited number of routes were acquired; therefore if there were major problems with certain parts of the street network where roads were missing these routes were eliminated. This aided in the development process. For, if these routes were used, the street network would have required additional digitizing risking problems with compatibility with the original street network design.

In addition to missing streets, the geocoded information provided on the map also created problems. First, address ranges provided by the city were incomplete or incorrect. In some cases additional maps were required to fix the problems in order for the geocoding process to work correctly. Due to the scope of the project, corrections made to the acquired map were limited, as correcting all

of the errors in the geocoding information of the base map was not possible. However, during manual digitization of missing streets, respective address ranges were entered as accurately as possible to minimize additional errors in address matching.

5.5.2 Individual Bus Routes and Route Network

Shape files (themes) of individual routes in addition to the entire route network were essential to the system. To create these shape files, each bus route was digitized from the base road network and saved independently. Merging the individual routes into one shape file using an ArcView3.2a tool created the route network.

Unlike the base network, the problems encountered when creating the individual bus routes were not as extensive. The primary concern regarded the street network. Again, it was not possible to make manual corrections in every case. Corrections were made in extreme cases that were recognized early enough to not create an exorbitant amount of backtracking in the project progression.

Cases also arose where bus routes travel through parking lots that are not represented on the base network map. In these cases, the road closest to the location was used as the approximate route or a segment was digitized to represent approximately where the route would travel. Information was also added to describe what the

segment represented (e.g., Nepean SportsPlex, Carlingwood Mall). In the future, buildings/parking lots will be added and the corresponding costs to travel these areas will be incorporated. In some instances the wait time through these areas (i.e., General Hospital) will most likely be more significant than that presented in this thesis.

5.5.3 Bus Stops and Landmarks

Although the OC Transpo schedule book only lists times at specific points (time points) on the routes, all stops were required to be extracted from the entire route network. Therefore all of the stops were digitized based on latitude and longitude coordinates collected by OC Transpo to create the bus stop shape file. Although the stops were recognized, their descriptions needed to be entered. Users need an explanation or intersection name to locate a bus stop, because providing them with a number would just create confusion. Thus, intersection names (e.g., Merivale and Meadowlands, Montreal and St. Laurent) or location descriptions (e.g., on Baseline Station at stop 1A) were manually entered for respective bus stops to complete the bus stop map and database.

To create the Landmark geographic file, an Excel database of landmarks was created. Using the addresses in the Excel

database, the landmark shape file was created through address matching (geocoding) with addition location corrections made manually in the shape file.

It is essential that the final location of each landmark in the landmark database is accurate; therefore an initial inspection of addresses was made. Each address was checked to make sure that it matched the base network. Problems encountered after inspection included landmarks not recognized by the system because the address on the streets was incorrect or out of date. Hence, the base network was recoded for address placement and the landmarks were re-matched.

5.5.4 Schedule Database

One of the important elements in the system is the OC Transpo schedule database, developed in Excel. The database contains information for the entire bus schedule, data of time-points, which are the bus stops provided in the schedule booklet, and descriptions of routes and time-points for weekend, Saturday and Sunday (Appendix H). The database is used for finding the shortest bus route and to provide schedule information to the users.

The schedule databases are considered relational databases made up of multiple tables rather than a single table. These tables are related to each other through the information they contain. To establish the relationship, a common field between the tables is needed. This field enables the tables to be linked together and can be used to search for information throughout the tables.

Three tables are contained in the developed schedule database. They are the schedule table, the bus route table, and the bus stop table. The schedule table contains the entire schedule for each trip of each route. The bus route table includes information related to routes such as route names, descriptions, and their availability during the week. Lastly, the bus stop table provides data related to bus stops, such as the stop number, descriptions, and arrival times. Figure 5.2 provides a flow chart describing the relationship between these three tables and highlights the identifiers between each table that allow for the link to exist.

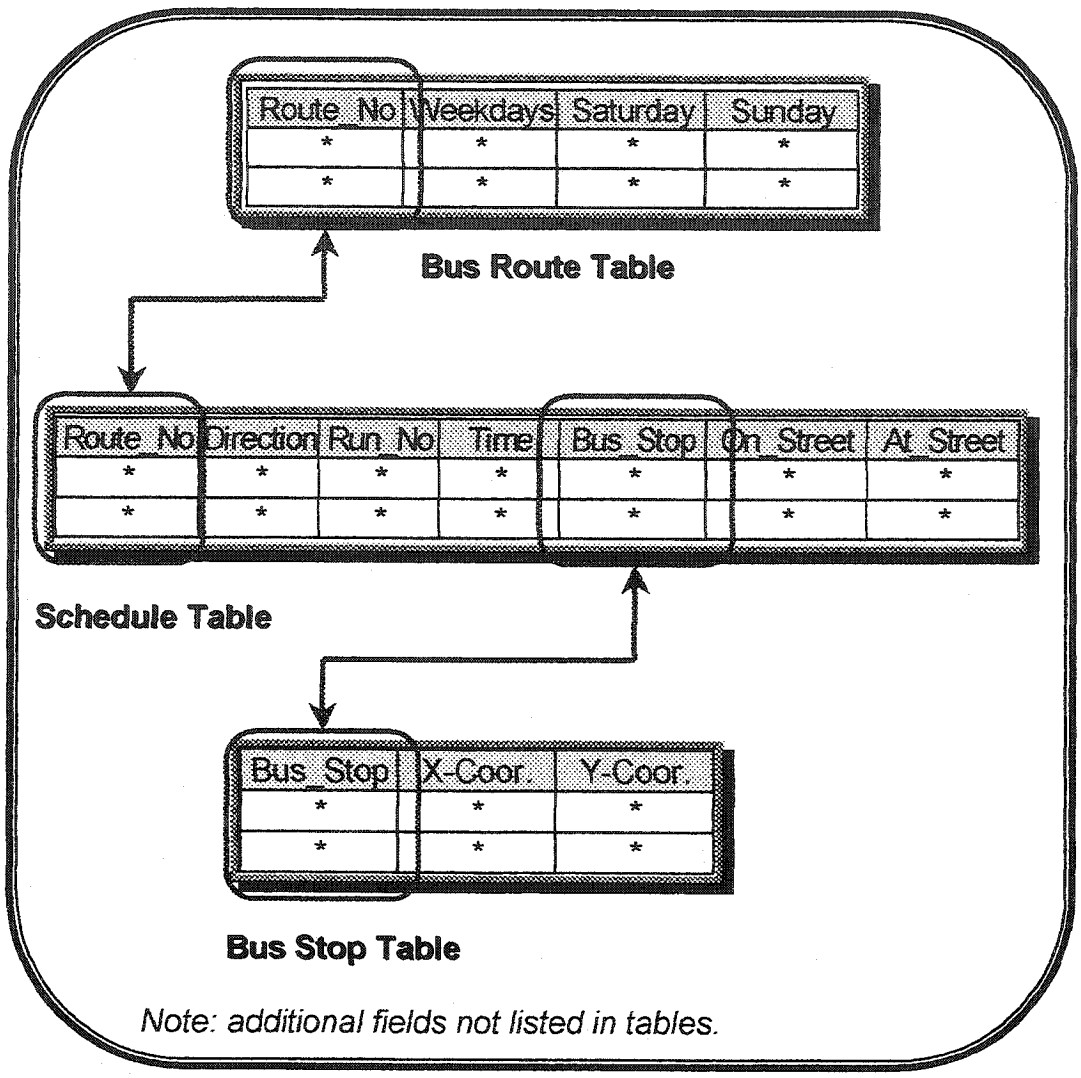


Figure 5.2: Relationships between Tables in the Schedule Database

As shown, there are two relationships among the tables: one between the schedule table and the bus route table, and the other between the schedule table and the bus stop table. The field establishing the relationship between the bus route table and the schedule table is given as "Route_No", which is the route number,

while "Bus_Stop" is the field that establishes a link between the schedule table and the bus stop table.

Detailed route and stop information can be derived through the schedule table because of these links. Even if the schedule table does not physically contain detailed information concerning routes and stops, information in the bus stop table and the bus route table can be located using the links between the tables. Therefore, after the calculations in the schedule table, other detailed user information can readily be derived from the bus route and stop tables.

5.6 Algorithm Development

5.6.1 Overview

A crucial part of the development of the transit information system is the algorithm for finding the optimum path. In solving optimum path problems, static methods have been the foremost approach. However, on transit networks, there is a level of uncertainty pertaining to arrival times due to factors such as traffic congestion, demand on the route, and malfunctions. Therefore true representation of transit networks should be probabilistic. In addition to being probabilistic, travel time on transit networks is constrained by a schedule; transit schedules are time-dependent.

The optimal path varies according to the time of entry into the system (Koncz, Greenfeld, Mouskos, 1996). If the system was not time dependent then a Dijkstra type algorithm can be used. However, optimum paths that are time-dependent can be found equally efficiently by using a Dijkstra algorithm with some modifications (Tong, Richardson, 1984). Therefore, it is assumed that transit travel will adhere to arrivals according to a schedule and that Bayesian statistics will be applied to compensate for the probabilistic nature of travel.

Since the transit system varies from that of a private automobile. The departure and arrival time from the origin to the destination are dictated by the bus schedule; they vary from daytime to evening, and from weekdays to weekends. Therefore, by combining the schedule data, extracted from the OC Transpo schedule database, with the network data from the entire bus route network, a new algorithm is developed.

5.6.2 Development within ArcView

In order to perform this task effectively, ArcView3.2a is used since it applies Dijkstra's algorithm (Chapter 4). ArcView3.2a contains four capabilities that are useful and necessary for building a transit information system. First is the capability of the input of standard street network files by any agency. Second is the ability to do

address matching or geocoding. Third is the ability to do pathfinding, whereby an optimal path based on time or length is found from an origin to a destination through any desired stops. Last is the ability to answer spatial analysis questions or queries such as finding the nearest facility. These capabilities, in addition to the visualization tools provided in a GIS, allow for ease of implementation (Koncz and Greenfeld, 1995).

For further modification, instead of starting at time zero at the origin node, the actual clock time is used. The arrival time at various nodes in the network will also be recorded as the actual clock time. As well, the length of an arc is dependent on the arrival time at the beginning of the arc. Hence, in order to determine the shortest travel time from node A to B the procedure is to search through the network file and find all arcs which match, and then for each of these arcs in turn read their start time from node A and then read the arrival time at node B and find the difference in time.

Before assigning a time to a bus stop, it is assumed that the bus gets to the time point on time. That is, the scheduled time at a time point is the real time a bus gets to that point. To assign a time to each stop, it is necessary to know the direction and distance of a stop to a time point where the scheduled time is known, as well as the speed of the bus between two time points. The distances of

each time point and bus stop has been obtained through one of ArcView3.2a's system algorithm (calalp.ave). Therefore, the distance and the direction between a stop and a time point can be identified. Through linear interpolation, each bus stop will read an arrival time when the information system is run.

The nearest bus stops to the chosen origin and destination are found using another ArcView3.2a script. This script enables a bus stop to be chosen within a distance of the chosen origin and destination points. An example of this script is shown below.

Syntax

```
aNetwork.FindClosestFac (aOriginPointList, aFacilityPointList,
NumFacs, CutOff, ToFacilityBoolean)
```

Example

```
' Calculate the closest tow trucks to a list of cars.
'
NumTowTrucksToFind = 1        ' find the closest tow truck
CuttOffDistance = 0           ' search the entire network
ToFacilityBoolean = False     ' travel is from the tow trucks to the
                                cars
'
aNumFoundList = aNetwork.FindClosestFac(aPointListOfBrokenDownCars,
                aPointListOfTowTrucks, NumTowTrucksToFind, CuttOff,
                ToFacilityBoolean)
```

Figure 5.3: Syntax of Closest Facility (ArcView3.2a)

5.6.3 Structure

In developing the structure of the transit information system, an attempt was made to simulate the communication that takes place

when the telephone service is utilized. Like telephone service interaction, the structure of the transit information system has three components: input of users' information, calculation of the shortest bus route, and display of the result. The following steps provide a description of the process used by the algorithm (Figure 5.4):

- Obtain information from user and step through user interface
- Geocode the origin and destination stops
- Search for bus arrival times at the origin point closest to the user's input
- Search all bus routes containing the origin or the destination point
- Obtain the closest bus stop to the origin and destination
- Obtain the nearest bus stop with a time point for linear interpolation
- Calculate the arrival time at the bus stop closest to the origin and destination
- Using the applied costs, calculate the travel time from the origin to destination
- Apply Bayesian statistics to the travel time
- Output the results graphically and with a written itinerary

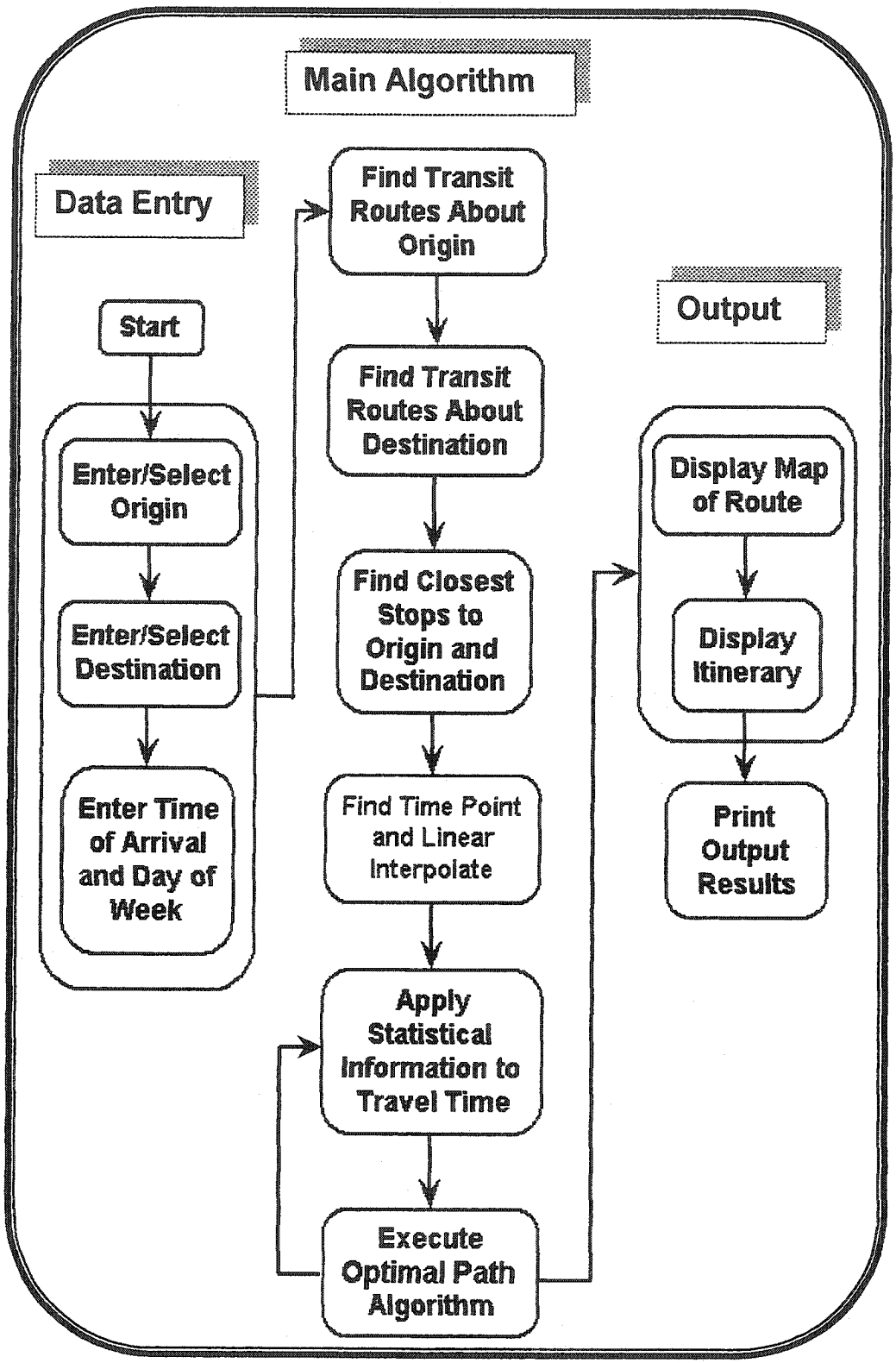


Figure 5.4: Algorithm Used Within Transit Information System

Within the transit information system, the database plays an important role. It organizes data describing the transit system, and provides a mechanism to store and manage the preferences of the user as they interact with system. Considerable effort is required when designing the structure of the database tables.

Chapter 6: Statistical Analysis

6.1 Introduction

Statistics are generally defined as the collection, classification, presentation, and analysis of numerical data (McGrew, Monroe, 1993). The use of statistics allows one to:

- Describe and summarize spatial data
- Make generalizations concerning complex spatial patterns
- Estimate the probability of outcomes for an event at a given location.

Engineers commonly use statistical analysis to solve problems and make decisions based on limited information about one or more of the parameters of the problem. The types of information available to them can be either objective or subjective. Objective information is based on experimental results or actual observations of a phenomenon. Subjective information is based on experience, intuition, other previous problems that are similar to the one under consideration, or the physics of the problem (Ayyub and McCuen, 1997).

Variables in the data sets used can be characterized as either discrete or continuous. A discrete variable has some restriction placed on the values that the variable can assume. A continuous variable has an infinitely large number of possible values along some interval of a real number line. In general, discrete

data are the results of counting, and potential values are limited to whole integers. Continuous data are the result of measurement, and values can be expressed as decimals (McGrew and Monroe, 1993).

If objective information has been gathered, it can be dealt with using the theories of probability and statistics. In this type, probability is interpreted as the frequency of occurrence assuming sufficient repetitions of the problem, its outcomes, and parameters, as a basis of the information. Subjective information reflects the judgment of the decision maker, the engineer, or analyst studying the problem. Subjective probabilities are used for decision making under uncertainty. These reflect the state of knowledge of the engineer or the analyst (Ayyub and McCuen, 1997).

It is common in engineering to encounter problems that are solved with both objective and subjective types of information. In these cases, it is desirable to utilize both types of information to obtain solutions or make decisions. The subjective probabilities are assumed to constitute a prior knowledge about a parameter, with gained objective information (or probabilities). Combining the two types produces posterior knowledge. The combination is performed based on Bayes' theorem (Ayyub and McCuen, 1997).

Bayes' theorem follows from the basic rules of probability. Its importance is that it enables analysts to update their knowledge, expressed in probabilistic terms, as new data are obtained.

6.2 Bayesian Analysis

Although Bayesian methods are attracting increasing attention there have been relatively few Bayesian applications in the transportation field. Researchers in other disciplines are adopting Bayesian techniques because they provide a principled approach for incorporating non-sample prior information, and they avoid erroneous approximations (Brownstone, 2000). For transit applications, consistent examinations of route conditions are required so that the public is informed and updated. By applying Bayesian techniques, scheduled travel information can be continually updated based on observations and expert knowledge. Therefore, the Bayesian technique was adopted for this research.

6.2.1 Introduction

This transit information system takes into account the initial estimate of travel time based on the latest available information (schedule information). The system then checks to see if OC Transpo has given any revised information based on travel conditions.

Within the database, travel time estimates are noted as R1, R2, R3, meaning early, on time, or late respectively. The actual travel times are treated as uncertain, and are revised based on gathered statistical information.

Once the journey is complete, the actual time from origin to destination becomes known. The actual time can be different from the estimated time, given a large number of sources of uncertainty. To name a few, weather-related delays, lane blockage due to snow clearing activities, accidents or incidents that result in lane blockage, unforeseen traffic conditions on mixed traffic links, blockage of bus-only lanes on city streets, passenger loading/unloading and other processes on the Transitway that cause deviations from the schedule.

6.2.2 Bayesian Approach

The Bayesian statistical approach to model a phenomenon is to systematically combine prior knowledge and experience with new data to improve the predictive relationship. The Bayesian methodology enables decisions to be made in the short term while improvements to the data, judgment, and the model continue to be made. The Bayesian solution achieves a balance between two

solutions based on initial observations and updated information (CSHRP, 2000).

In assembling information, the data collected are supplemented with prior knowledge. This approach is summarized Figure 6.1. When further information is obtained, the probability is updated using Bayes' theorem. The distribution before updating is called the prior distribution and the updated distribution is the posterior distribution (McGrew and Monroe, 1993).

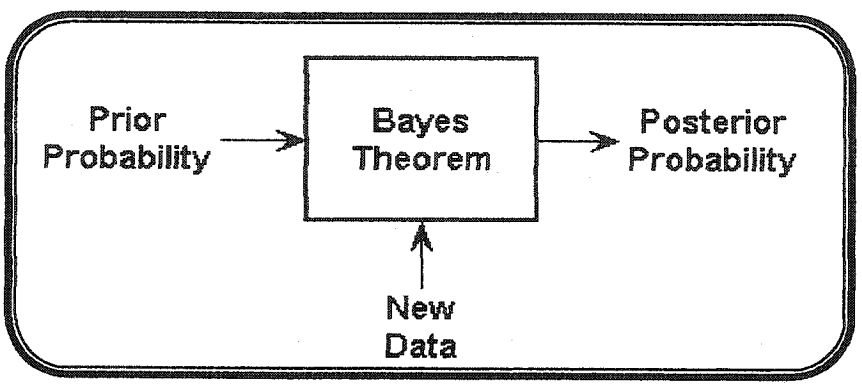


Figure 6.1: The Bayesian Statistical Approach (adapted from CSHRP, 2000)

The advantages of using Bayesian methods justify the increased effort in computation and prior determination. Bayesian methods provide one with the ability to formally incorporate prior information. The assumptions are conditional on the actual data and all of the analyses follow directly from the posterior distribution; no separate theories of estimation, testing, multiple comparisons are needed.

Any question can be directly answered through Bayesian analysis (Carlin and Louis, 2000).

6.2.3 Bayes' Theorem

Bayes' Theorem allows one to assess the probability of A given that B has already occurred. For this development, there are two simple formulas from probability theory. If A and B are two events, then:

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)} \qquad \text{if } P(B) \neq 0 \qquad (1)$$

$$P(A) = P(A \text{ and } B) + P[A \text{ and (not } B)] \qquad (2)$$

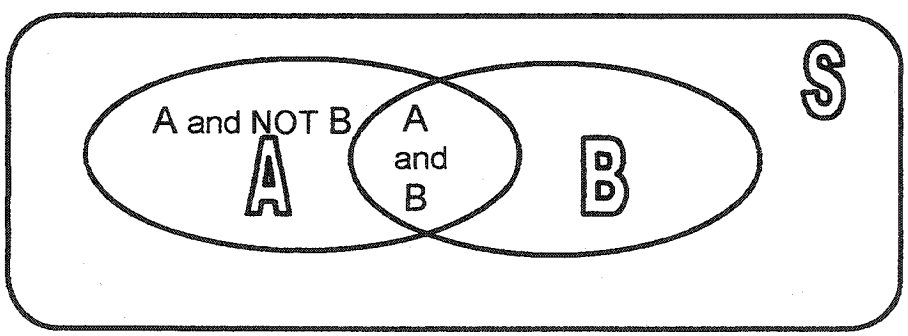


Figure 6.2: Venn Diagram of Event A and Event B

Now if a set of mutually exclusive events A_1, A_2, .. A_N (prior, subjective information) form a partition of a sample space where $A_1 + A_2 + .. + A_N = S$ and E (objective information) is an arbitrary event

(Ayyub and McCuen, 1997) as shown in the figure below, then the theorem of total probability states that:

$$P(E) = P(A_1)\,P(E|A_1) + P(A_2)\,P(E|A_2) + \ldots + P(A_N)\,P(E|A_N) \qquad (3)$$

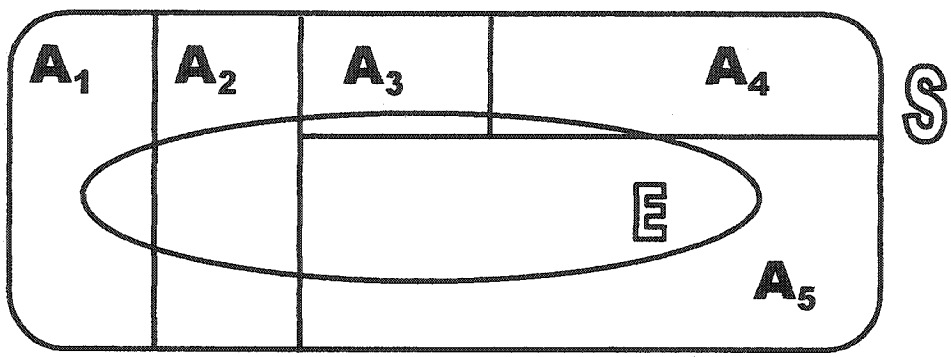


Figure 6.3: Venn Diagram of Total Probability

This theorem is very important in computing the probability of the event E, especially in practical cases where the probability cannot be computed directly, but the probabilities of the partitioning events and the conditional probabilities can be computed.

Bayes' theorem is very useful in computing the posterior (or reverse) probability of the type $P(A_i|E)$, for i = 1, 2,..., n. (Raiffa, 1968, Ayyub and McCuen, 1997). The posterior probability can be computed as follows:

$$P(A_i|E) = \frac{P(E|A_i)\,P(A_i)}{P(E|A_1)\,P(A_1) + P(E|A_2)\,P(A_2) + \cdots + P(E|A_n)\,P(A_n)} \qquad (4)$$

Or in more general terms as:

$$P(A_i|E) = \frac{P(E|A_i)\,P(A_i)}{\sum_{j=1}^{n} P(E|A_j)\,P(A_j))} \qquad (5)$$

The denominator of this equation is P(E), which is based on the theorem of total probability. According to the above, the prior knowledge, $P(A_i)$, is updated using the objective information, P(E), to obtain the posterior knowledge, $P(A_i|E)$. In general as more and more data are added to the problem, the posterior will continue to become more and more definitive.

6.3 Method of Application

This approach treats origin-destination travel time as uncertain. To begin the analysis, the system was assigned three states-of-nature (T1, T2, T3), which are uncertain travel times. T1, T2, and T3 correspond to early, on time, and late arrival respectively. As defined by OC Transpo during data collection: "The allowance for 'early' is tight but the 'late' allows a few minutes". As explained further, a bus was categorized as early if it was one to two minutes earlier than scheduled. A bus was considered late if it arrived 3 to 4 minutes after the scheduled time. Therefore the probabilities obtained are based on these conditions.

Since the Ts are discretely defined, meaning the travel time can either be one of three choices, the probabilities have been regarded as 'discrete' and assigned on the basis of such rationale as: low probability of early arrival at the destination, high probability of on-time arrival, moderate probability of late arrival. The sum of these equals one. The probabilities have then been estimated on the basis of frequency analysis of OC Transpo data.

When the data were compiled, there was a relatively lower number collected over the weekend. The lower number of observations is acceptable for Bayesian analysis because as more data are collected, the system can be updated and new posterior probabilities can be obtained.

For the purpose of calculating travel time, the three travel times were based on OC Transpo's information on what is categorized as early, on time, and late. Therefore T1, T2, and T3 were assigned travel times as follows:

T1: Travel time - 1min or more
T2: Travel time (if arrival is less than 1 minute early or less than 3 minutes late)
T3: Travel time + 3min or more

The prior probabilities of the Ts, P'(T), are found in Table 6.1. These have been defined based on OC Transpo's time periods. These are for off peak periods (18:31 – 6:30 & 9:31 – 15:30), peak periods (6:31 – 9:30 & 15:31 – 18:30), and peak hours (7:31 – 8:30 & 15:31 – 16:30). As an overall summary of all day performance, these were also compiled. To represent the sample routes within the application, probabilities were estimated for both Transitway (95, 97) and

non-Transitway (3, 118) routes. The data values were obtained through OC Transpo's AVL/C system.

Therefore, the initial expected travel time based solely on prior probabilities is calculated as follows:

Expected Travel Time = P(T1)(T1)+P(T2)(T2)+P(T3)(T3)

6.3.1 Information Updates

By making frequent checks with road conditions and travel times, the system can obtain the latest estimates of time. Within the transit information system, the transit agency must specify their perception of how the buses are running. A code is entered into the transit route table corresponding to early, on time or late conditions; R1, R2, and R3 respectively.

6.3.2 Conditional Probabilities

The conditional probability, for example P(R1|T1), is the probability of getting the indication of 'early arrival', given that the actual state is T1 (early arrival). In other words, if it turns out that the traveller will arrive early, what is the probability that the information obtained implies the same. Likewise P(R2|T1) defines the probability of getting the on-time indication, given that the true state is early arrival.

Table 6.1: Discrete Prior Probabilities based on OC Transpo Defined Travel Periods

Routes

	03:01-06:30			07:31-08:30			06:31-09:30			09:31-15:30		
Weekday	Early	On Time	Late	Early	On Time	Late	Early	On Time	Late	Early	On Time	Late
95	0.330	0.670	0.000	0.130	0.700	0.170	0.180	0.700	0.120	0.310	0.650	0.040
97	0.180	0.770	0.050	0.200	0.710	0.090	0.220	0.710	0.070	0.210	0.630	0.160
AVG	0.255	0.720	0.025	0.165	0.705	0.130	0.200	0.705	0.095	0.260	0.640	0.100
3	0.120	0.830	0.050	0.110	0.580	0.310	0.140	0.660	0.200	0.390	0.540	0.070
118	0.330	0.590	0.080	0.280	0.530	0.190	0.340	0.540	0.120	0.290	0.630	0.080
AVG	0.225	0.710	0.065	0.195	0.555	0.250	0.240	0.600	0.160	0.340	0.585	0.075
Saturday	Early	On Time	Late				Early	On Time	Late	Early	On Time	Late
95	0.120	0.860	0.020				0.470	0.470	0.060	0.330	0.640	0.030
97	0.370	0.630	0.000				0.230	0.760	0.010	0.190	0.720	0.090
AVG	0.245	0.745	0.010				0.350	0.615	0.035	0.260	0.680	0.060
3							0.000	0.740	0.260	0.120	0.680	0.200
118	0.370	0.500	0.130				0.430	0.570	0.000	0.340	0.600	0.060
AVG	0.370	0.500	0.130				0.215	0.655	0.130	0.230	0.640	0.130
Sunday	Early	On Time	Late				Early	On Time	Late	Early	On Time	Late
95	0.500	0.350	0.150				0.320	0.650	0.030	0.220	0.700	0.080
97	0.520	0.380	0.100				0.220	0.710	0.070	0.090	0.780	0.130
AVG	0.510	0.365	0.125				0.270	0.680	0.050	0.155	0.740	0.105
3							0.280	0.600	0.120	0.200	0.610	0.190
118							0.360	0.610	0.030	0.220	0.720	0.060
AVG							0.320	0.605	0.075	0.210	0.665	0.125

Table 6.1: Discrete Prior Probabilities based on OC Transpo Defined Travel Periods (cont'd)

Routes

	15:31-16:30			15:31-18:30			18:31-03:00			ALL DAY		
Weekday	Early	On Time	Late	Early	On Time	Late	Early	On Time	Late	Early	On Time	Late
95	0.220	0.670	0.110	0.230	0.640	0.130	0.220	0.680	0.100	0.250	0.660	0.090
97	0.280	0.640	0.080	0.270	0.630	0.100	0.200	0.610	0.190	0.230	0.630	0.140
AVG	0.250	0.655	0.095	0.250	0.635	0.115	0.210	0.645	0.145	0.240	0.645	0.115
3	0.250	0.620	0.130	0.360	0.530	0.110	0.260	0.560	0.180	0.290	0.580	0.130
118	0.430	0.430	0.140	0.450	0.450	0.100	0.350	0.570	0.080	0.360	0.540	0.100
AVG	0.340	0.525	0.135	0.405	0.490	0.105	0.305	0.565	0.130	0.325	0.560	0.115
Saturday				Early	On Time	Late	Early	On Time	Late	Early	On Time	Late
95				0.260	0.660	0.080	0.240	0.700	0.060	0.310	0.630	0.060
97				0.200	0.630	0.170	0.200	0.670	0.130	0.210	0.690	0.100
AVG				0.230	0.645	0.125	0.220	0.685	0.095	0.260	0.660	0.080
3				0.190	0.520	0.290	0.240	0.700	0.060	0.170	0.650	0.180
118				0.290	0.500	0.210	0.400	0.500	0.100	0.360	0.540	0.100
AVG				0.240	0.510	0.250	0.320	0.600	0.080	0.265	0.595	0.140
Sunday				Early	On Time	Late	Early	On Time	Late	Early	On Time	Late
95				0.140	0.670	0.190	0.210	0.690	0.100	0.220	0.670	0.110
97				0.120	0.760	0.120	0.170	0.810	0.020	0.140	0.760	0.100
AVG				0.130	0.715	0.155	0.190	0.750	0.060	0.180	0.715	0.105
3				0.320	0.570	0.110	0.350	0.600	0.050	0.280	0.590	0.130
118				0.280	0.670	0.050	0.350	0.530	0.120	0.290	0.640	0.070
AVG				0.300	0.620	0.080	0.350	0.565	0.085	0.285	0.615	0.100

This conditional probability information was obtained through OC Transpo's AVLC system. Based on previous experience, the conditional probabilities were estimated. These are shown in Figure 6.4. That is to say if the actual travel time turns out to be 'early', the probabilities of getting the advisory as early, on time or late are 70%, 25%, and 5%, respectively. The probabilities of receiving advisory are early, on time, or late given that the true travel time will be 'on time' are 15%, 70%, and 15%, respectively. And the probabilities of receiving advisory as early, on time, or late given that the true travel will be 'late' are 5%, 25%, and 70%.

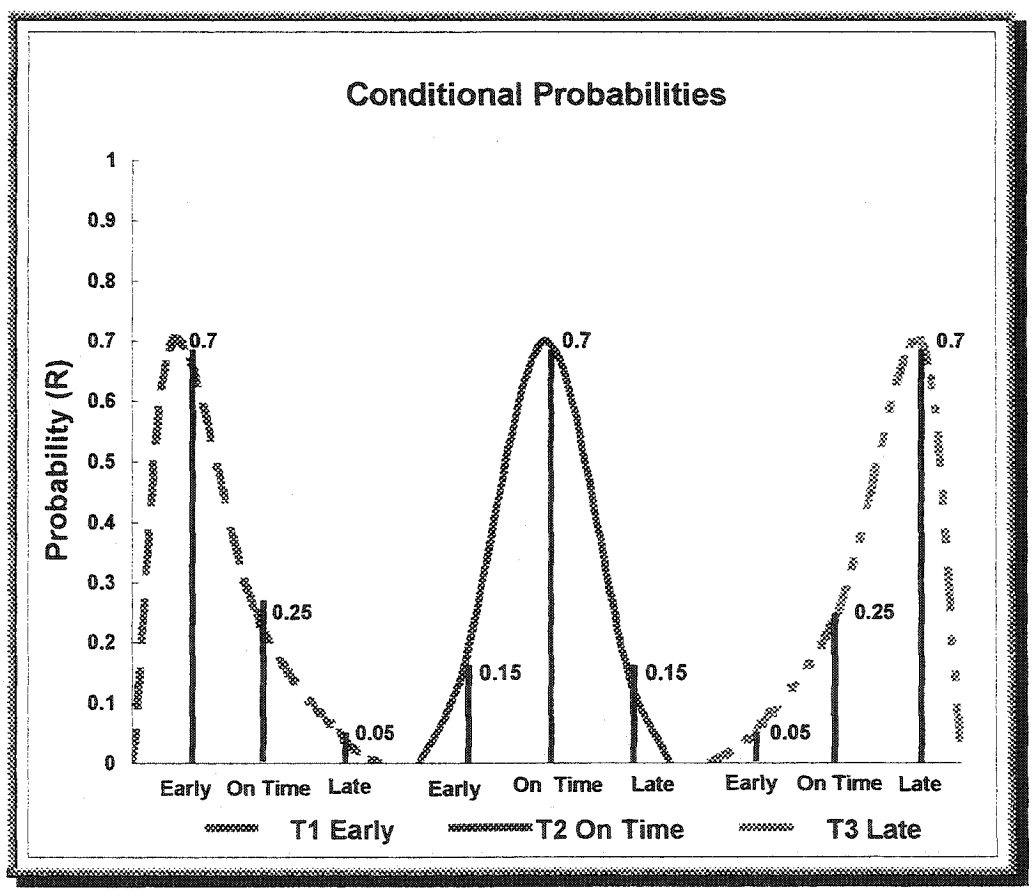


Figure 6.4: Conditional Probabilities

These conditional probabilities can be regarded as the reliability of information. As more data are obtained on a seasonal or yearly basis, modifications can be made.

6.3.3 Posterior Probabilities

These are revised probabilities of states of nature, found by combining old information with new information. Bayes' Theorem is used to calculate posterior probabilities.

If R1 is received as the information:

$$P''(T1|R1) = \frac{[P(T1)P(R1|T1)]}{[P(T1)P(R1|T1)+P(T2)P(R1|T2)+P(T3)P(R1|T3)]} \quad (6)$$

$$P''(T2|R1) = \frac{[P(T2)P(R1|T2)]}{[P(T1)P(R1|T1)+P(T2)P(R1|T2)+P(T3)P(R1|T3)]} \quad (7)$$

$$P''(T3|R1) = \frac{[P(T3)P(R1|T3)]}{[P(T1)P(R1|T1)+P(T2)P(R1|T2)+P(T3)P(R1|T3)]} \quad (8)$$

If R2 is received as the information:

$$P''(T1|R2) = \frac{[P(T1)P(R2|T1)]}{[P(T1)P(R2|T1)+P(T2)P(R2|T2)+P(T3)P(R2|T3)]} \quad (9)$$

$$P''(T2|R2) = \frac{[P(T2)P(R2|T2)]}{[P(T1)P(R2|T1)+P(T2)P(R2|T2)+P(T3)P(R2|T3)]} \quad (10)$$

$$P''(T3|R2) = \frac{[P(T3)P(R2|T3)]}{[P(T1)P(R2|T1)+P(T2)P(R2|T2)+P(T3)P(R2|T3)]} \quad (11)$$

If R3 is received as the information:

$$P''(T1|R3) = \frac{[P(T1)P(R3|T1)]}{[P(T1)\ P(R3|T1)+P(T2)P(R3|T2)+P(T3)P(R3|T3)]} \quad (12)$$

$$P''(T2|R3) = \frac{[P(T2)P(R3|T2)]}{[P(T1)P(R3|T1)+P(T2)P(R3|T2)+P(T3)P(R3|T3)]} \quad (13)$$

$$P''(T3|R3) = \frac{[P(T3)P(R3|T3)]}{[P(T1)P(R3|T1)+P(T2)P(R3|T2)+P(T3)P(R3|T3)]} \quad (14)$$

Once the posterior probabilities are found, the expected travel time can be calculated as follows:

For R1 case:

The Expected Travel Time =

$$P''(T1|R1)(T1) + P''(T2|R1)(T2) + P''(T3|R1)(T3) \quad (15)$$

For R2 case:

The Expected Travel Time =

$$P''(T1|R2)(T1) + P''(T2|R2)(T2) + P''(T3|R2)(T3) \quad (16)$$

For R3 case:

The Expected Travel Time =

$$P''(T1|R3)(T1) + P''(T2|R3)(T2) + P''(T3|R3)(T3) \quad (17)$$

The following section illustrates the above equations through an example.

6.3.4 Sample Calculations

With the above defined, sample calculations have been performed to observe how Bayesian analysis can handle the uncertainty with travel time.

Carleton University to My House Journey

T1 = 21 minutes

T2 = 22 minutes

T3 = 25 minutes

Priors: Using 'ALL DAY' WEEKDAY probabilities for Transitway routes.

P'(T1) = 0.24

P'(T2) = 0.645

P'(T3) = 0.115

1.0

Expected Time Based on Prior Probabilities

Exp. Time = (0.24)(21)+(.645)(22)+(.115)(25) = 22.1 minutes

Conditional Probabilities

P(R1|T1) = 0.700	P(R1|T2) = 0.150	P(R1|T3) = 0.050
P(R2|T1) = 0.250	P(R2|T2) = 0.700	P(R2|T3) = 0.250
P(R3|T1) = 0.050	P(R3|T2) = 0.150	P(R3|T3) = 0.700
1.000	1.000	1.000

Posterior Probabilities

P"(T1|R1) = 0.621	P"(T1|R2) = 0.053	P"(T1|R3) = 0.064
P"(T2|R1) = 0.358	P"(T2|R2) = 0.836	P"(T2|R3) = 0.511
P"(T3|R1) = 0.021	P"(T3|R2) = 0.111	P"(T3|R3) = 0.425
1.000	1.000	1.000

As shown in Figures 6.5 to 6.7, there is a significant change in the probabilities due to a change in the travel advisory. The prior probabilities are 24, 64.5, and 11.5 percent for early, on time, and late travel respectively.

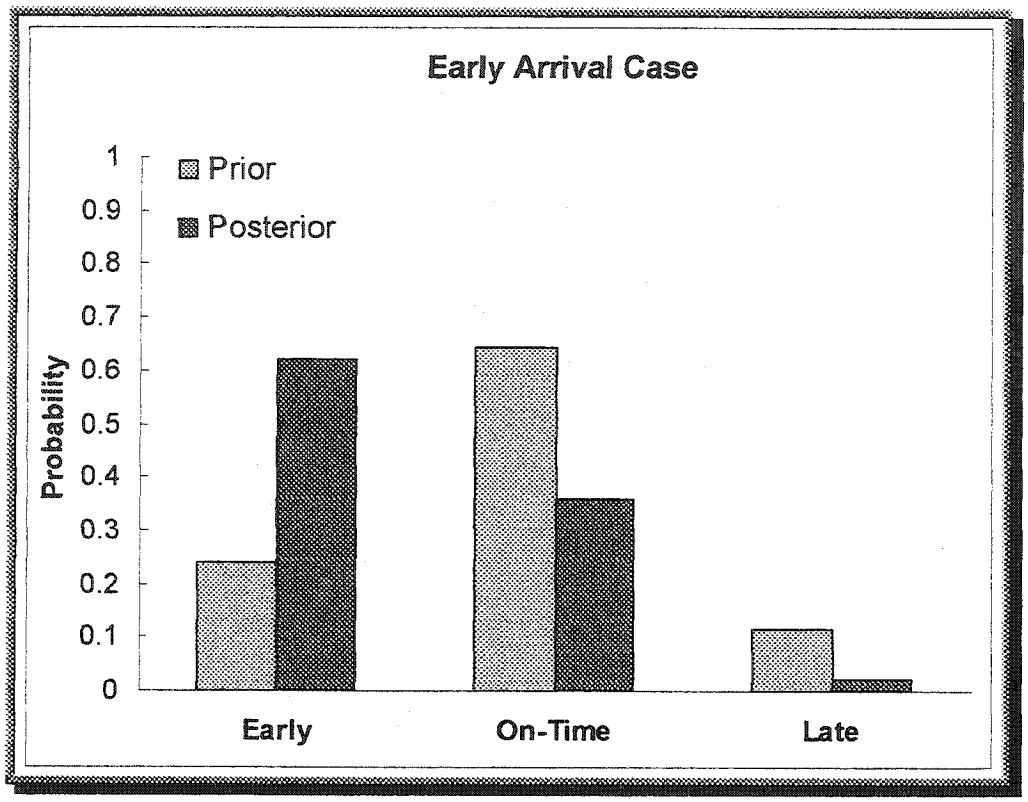


Figure 6.5: Early Arrival Case

Given that the arrival is early, Figure 6.5 illustrates the probabilities of a bus being early, on time, or late. As expected, the probability of early arrival is significantly increased. To compensate, there is a significant decrease in the on time probability and a small change is found in the late probability. If the bus is travelling early, there is a high probability of it arriving early, while it will seldom arrive late. Once these posterior probabilities are applied to update the travel time, the uncertainty in early travel can be noticeably reduced.

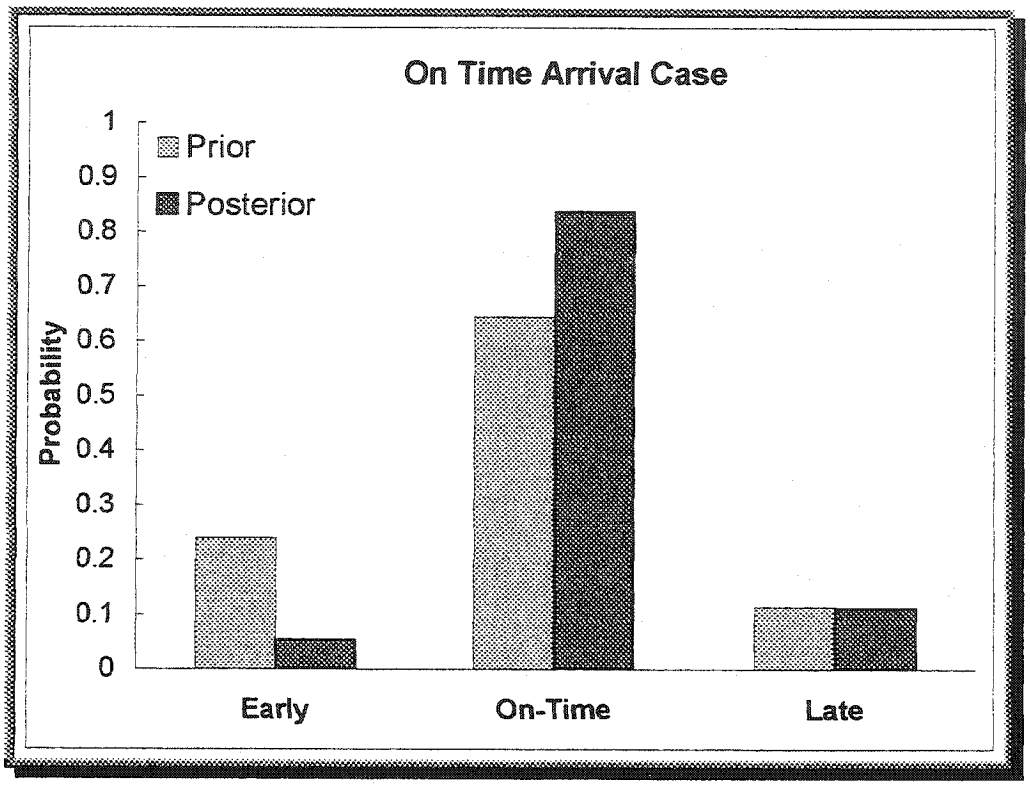


Figure 6.6: On Time Arrival Case

In Figure 6.6, the on time advisory is given. As expected in this case, the probability for being on time is increased. To compensate for this increase, the early and late probabilities have decreased. When applied to update the travel time, the user will notice that the travel time probability will shift from having a fairly good probability of being on time to a clear-cut on time probability.

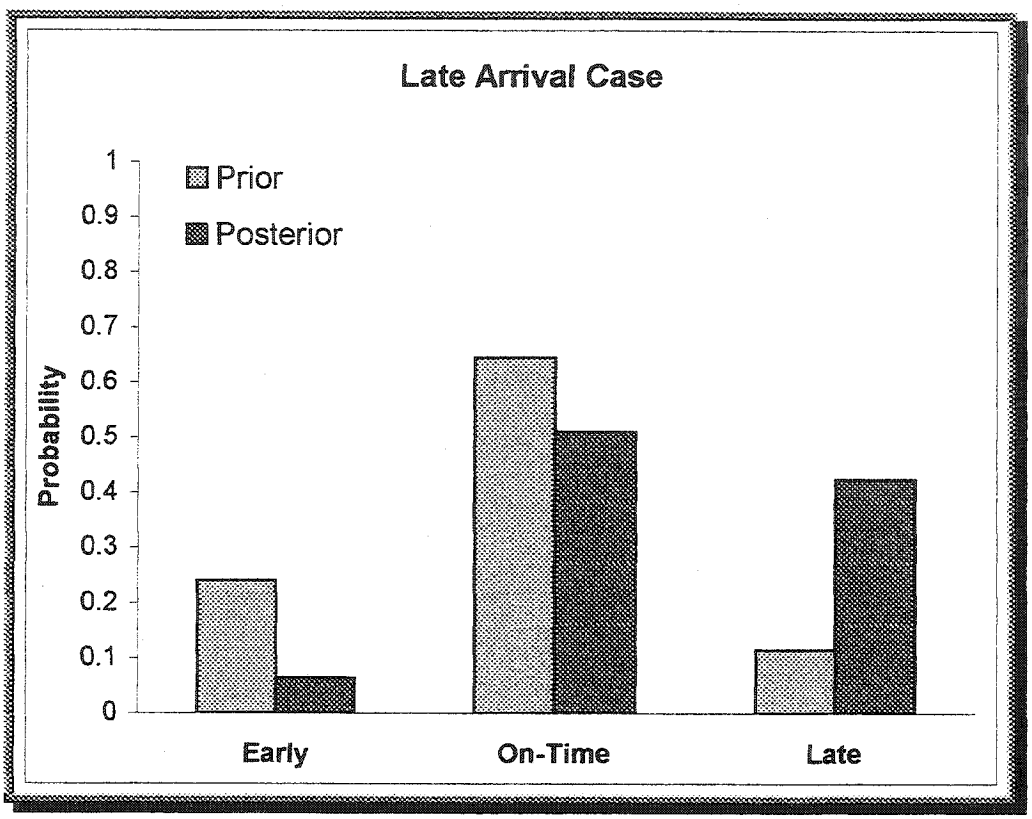


Figure 6.7: Late Arrival Case

Likewise for the late advisory shown in Figure 6.7, the late probability has significantly increased. As expected, when a bus is travelling late, the probability of it arriving late is high. The likelihood of a bus arriving early is small, where there is still a notable probability of it arriving on time. Since the prior probability of being late is quite small in comparison to the other conditions, the on time posterior probability is still greater than the late probability. But, with the large increase in the late probability, the uncertainty has been greatly minimized for the user.

Expected Time Based on Posterior Probabilities

Expected Travel Time: if R1 is the information received:

Time = (0.621)(21)+(0.358)(22)+(0.021)(25) = 21.44 minutes

Expected Travel Time: if R2 is the information received:

Time = (0.053)(21) + (0.836)(22) + (0.111)(25) = 22.28 minutes

Expected Travel Time: if R3 is the information received:

Time = (0.064)(21)+(0.511)(22)+(0.425)(25)= 23.21 minutes

From this example, it can be seen that the Bayesian analysis results in more realistic estimates of travel time as compared to a complete reliance on scheduled travel time. Likewise, a complete reliance is not placed on the bus status advisory, given that such information on a projected basis is not absolutely reliable. By applying Bayesian analysis, the uncertainty is considered and minimized to help users plan their arrivals at bus stops and their trips more efficiently.

6.4 Calculation Procedure

Figure 6.8 shows the computational procedure used for producing inputs to the optimal route algorithm. Following the establishment of the states of nature T1, T2, and T3 and the assignment of prior probabilities, the next step is to check on the transit agency's on-line database for changes in bus arrival times (e.g. a bus running late).

If there is no change from the initial times, prior probabilities are used to find expected travel time. On the other hand, if there is a travel time change advisory, then the posterior probabilities are computed and used as a basis for estimating expected travel time.

The estimated travel time information is provided to the optimum route algorithm for establishing the best route and finding the estimated travel time.

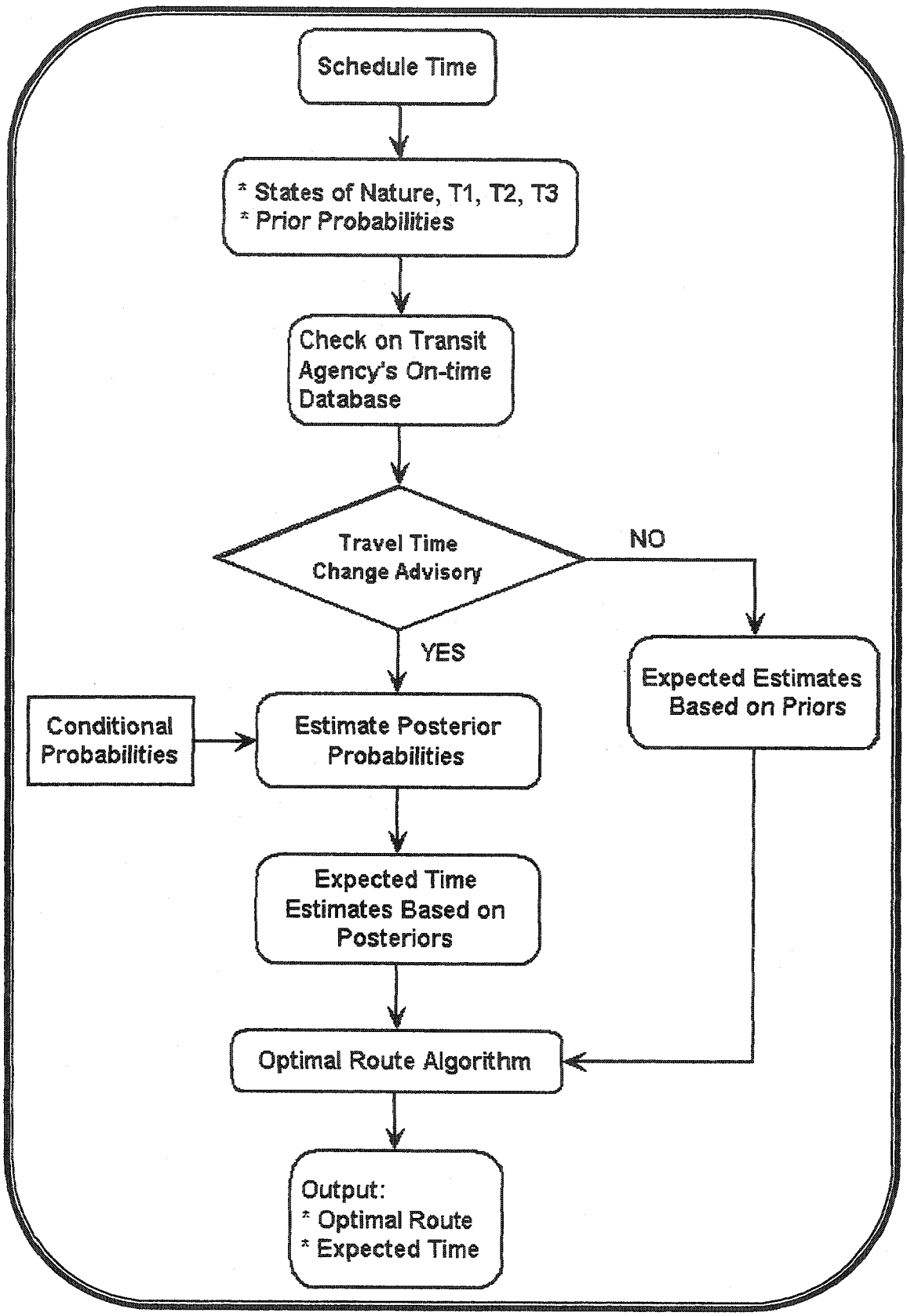


Figure 6.8: Bayesian Statistical Analysis of Travel Time

Chapter 7: Synthesis and Application of the Transit Information System

7.1 Introduction

The Transit Information System design process using GIS is an application process requiring many steps. The design process starts with defining a GIS data structure to represent a logical relationship among individual elements of transit data. Next, the design enables the representation of the control relationships among functions. Further, the design focuses on the interface by showing how the application communicates within itself and with users. Finally, the transit information system was designed utilizing ArcView3.2a, Avenue, and Dialog Designer. The final product of the system consists of 75 scripts (algorithm), and 7 dialogs. The scripts can be found in Appendix C.

The transit information system was developed as a user-interface. In the user-interface, there are three parts: setting the user's preference, inputting the user's desired origin and destination or schedule information, and displaying the results.

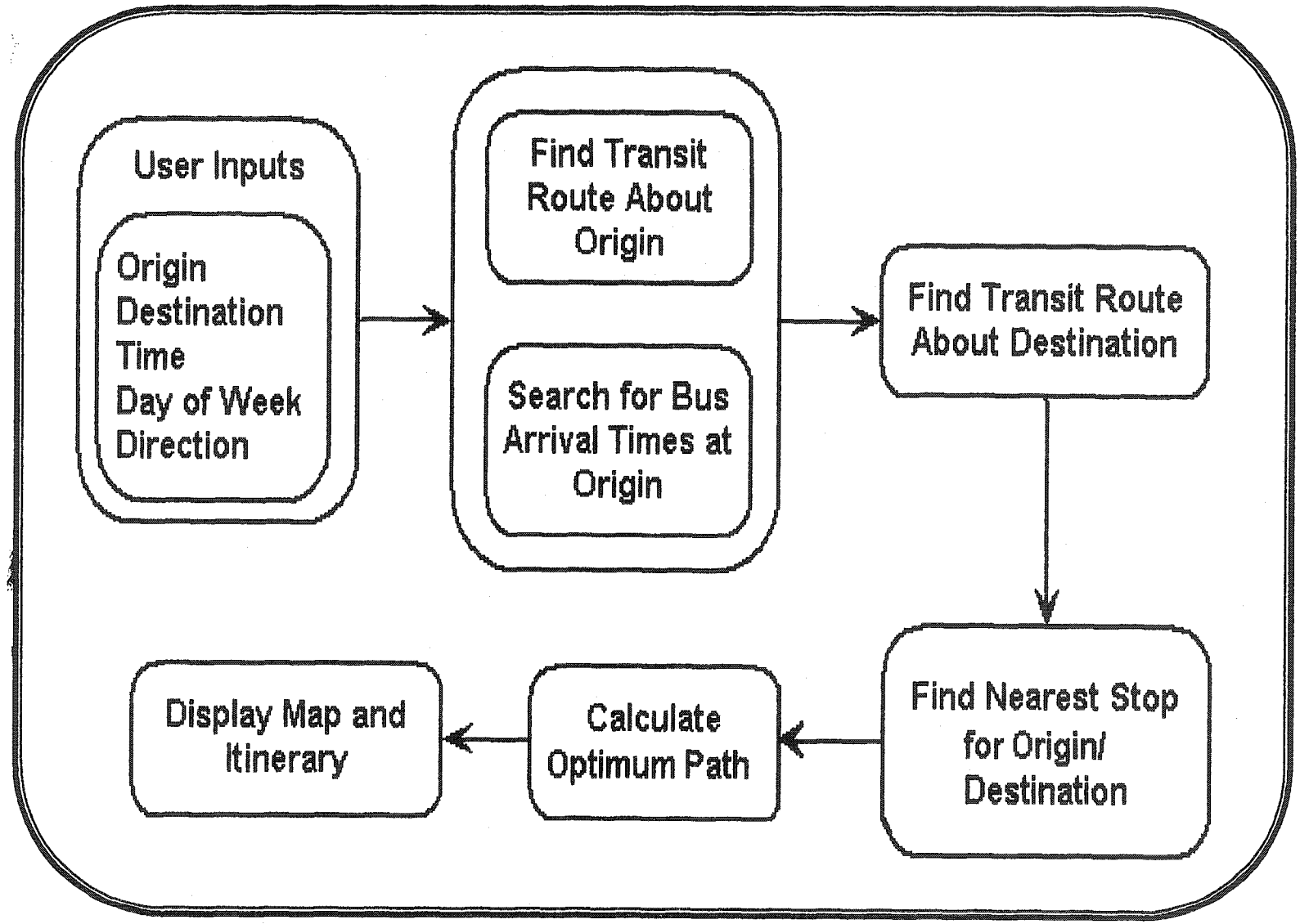


Figure 7.1: Overall Structure of Transit Information System

Figure 7.1, outlines the process. First, according to the set user preferences the application generates the subsequent pages, for selecting the origin and destination points. From these pages, the user's origin and destination data flow into the application for locating the points and their respective bus stops. Once it finds the origin and destination, and bus stops, the application displays these points and calculates the optimum path and the result is displayed to users to print.

7.2 User Interface Design

When the user enters the application, a screen showing the route network and bus location points is visible. If the user would like to view other categorical landmarks they can be chosen by selecting one or more on the left hand side of the screen. When implemented an instructions/help menu will also be available for further instructions and explanations for the user. Figure 7.2 illustrates this screen. To continue the user presses the button labeled 'RUN TIS' located on the main view.

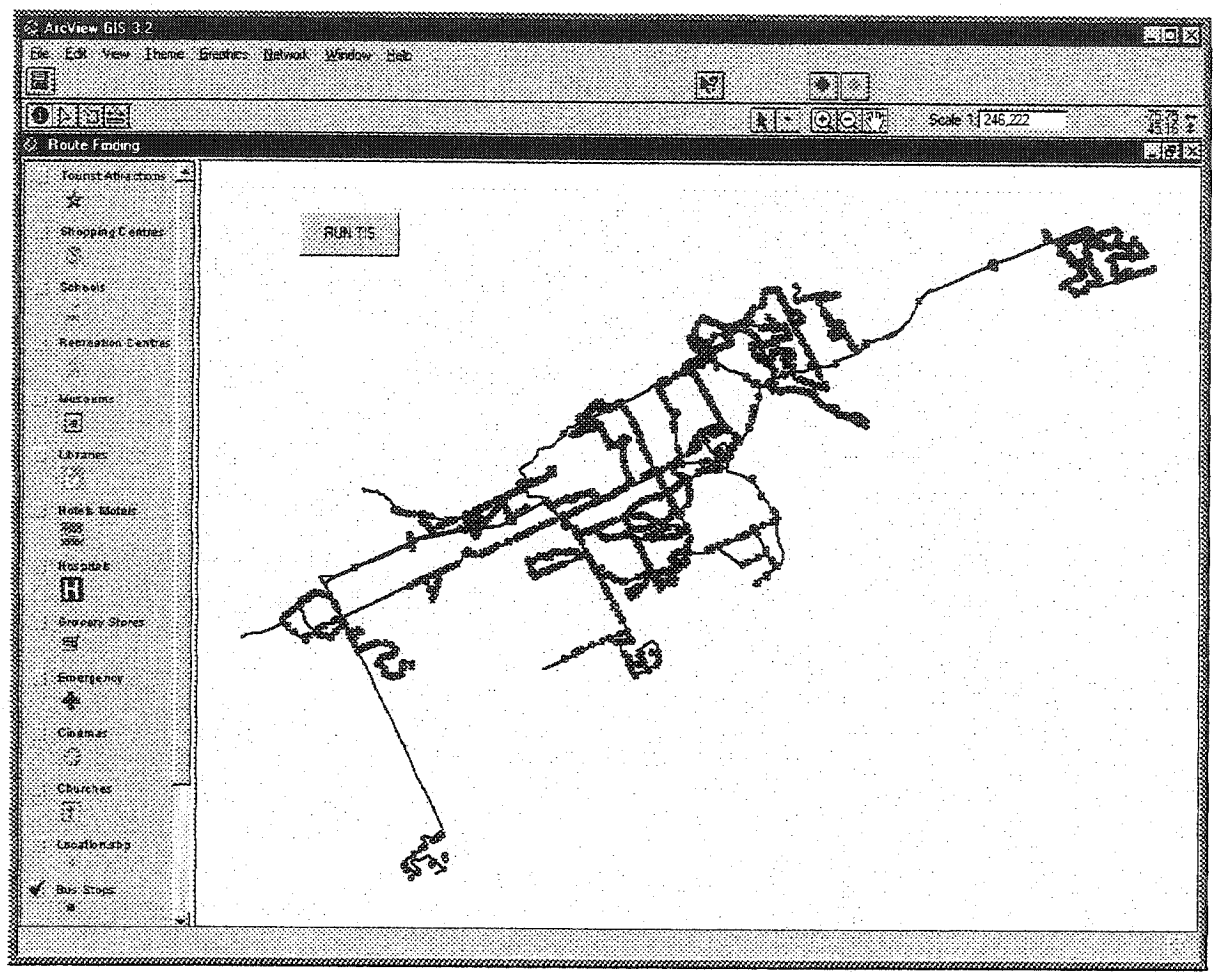


Figure 7.2: Initial View

When the user enters the transit information system, the window shown in Figure 7.3 appears. The user can either obtain schedule information pertaining to a specific route or continue to the user interface in order to find the optimum route. If exit is chosen, a message box will appear to verify this is what the user desires. This message box is shown in Figure 7.4.

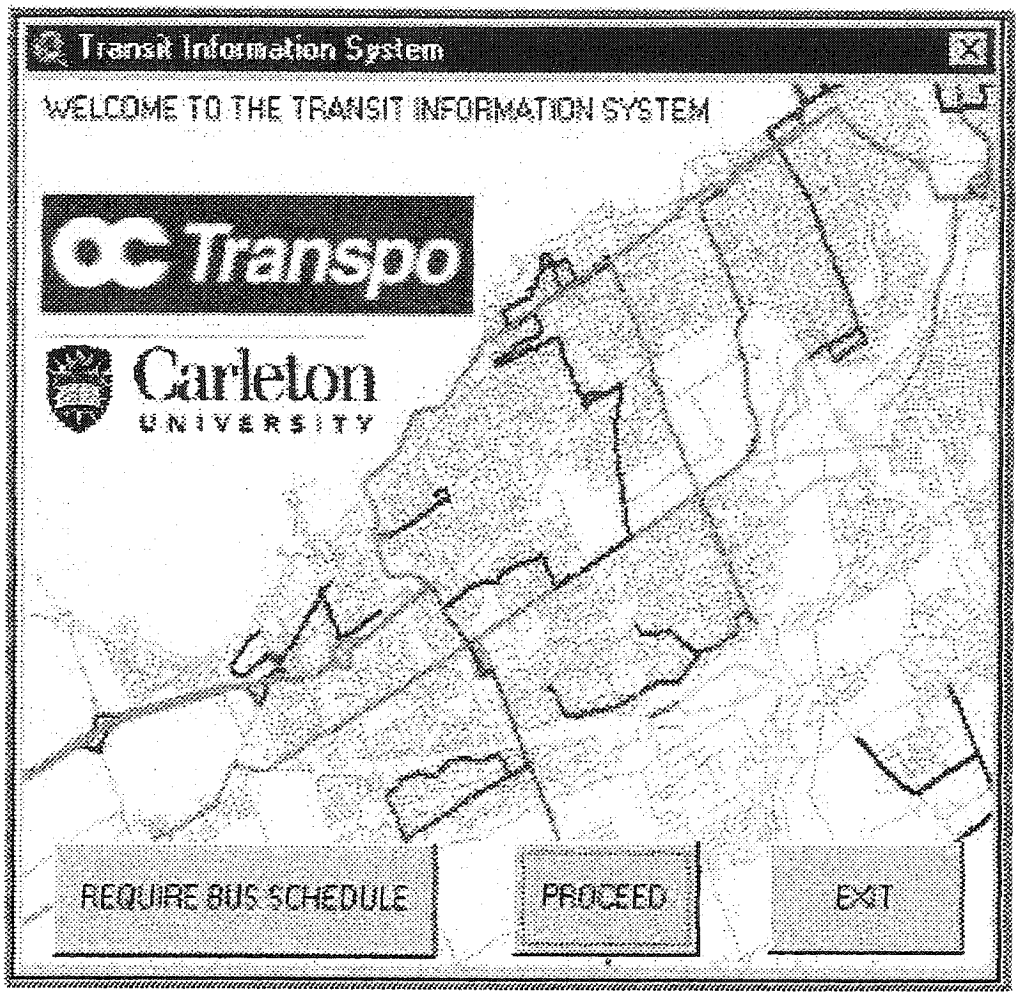


Figure 7.3: Initial dialog

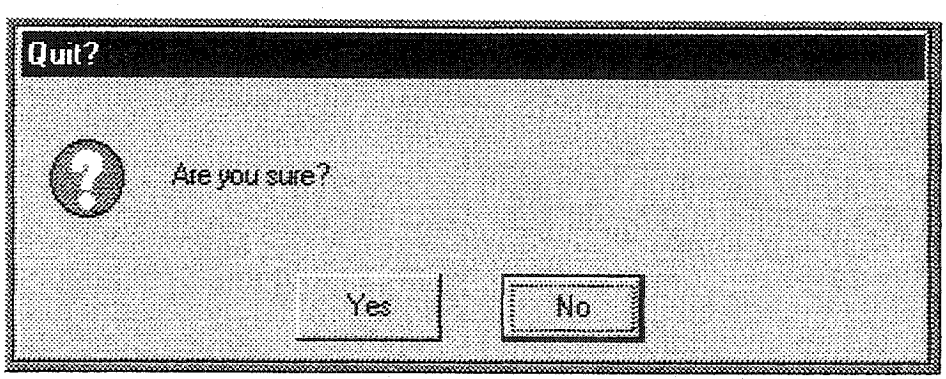


Figure 7.4: Verification Message Box

If the user selects 'REQUIRE BUS SCHEDULE' the dialog box shown in Figure 7.5 will appear. In this step, the user is asked to provide the time (hour, minute, and AM or PM) his/her trip will begin and the day of the week (Weekday, Saturday, or Sunday) the trip will take place. Radio buttons prevent the user from selecting both AM and PM or every day of the week. Since this dialog will produce a schedule, the user must also select a route number from the drop down menu. When all of the information is entered, the user has the option of canceling and restarting or to get the schedule information.

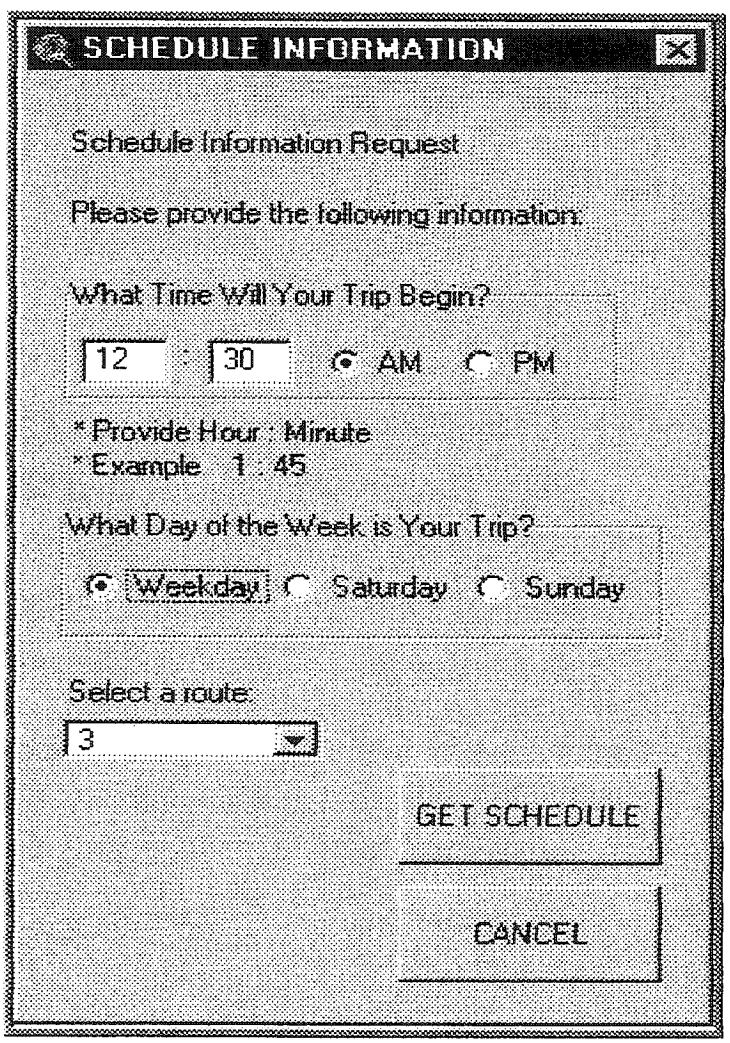


Figure 7.5: Dialog Box for Schedule Information

When the 'GET SCHEDULE' button is selected, the algorithm produces a highlighted table with the bus route and corresponding times of arrivals at different bus stops. The figure below shows an example of this.

weekday.dbf

Route_no	Direction	Run_no	Bus_stop	On_street	At_street	Time
117	EB	E-349	CG500	LIBRARY	ARTS TOWER	836
117	EB	E-189	CG500	LIBRARY	ARTS TOWER	851
117	EB	11603	CG500	LIBRARY	ARTS TOWER	906
117	EB	E-370	CG500	LIBRARY	ARTS TOWER	921
117	EB	11603	SH900	BASELINE		832
117	EB	E-370	SH900	BASELINE		847
117	EB	11601	SH900	BASELINE		907
117	EB	11601	SB330	MEADOWLANDS	MERIVALE FS	916
117	EB	E-370	SB330	MEADOWLANDS	MERIVALE FS	857
117	EB	11603	SB330	MEADOWLANDS	MERIVALE FS	842
117	WB	11601	SB320	MEADOWLANDS	MERIVALE #1 NS	831
117	WB	E-690	SB320	MEADOWLANDS	MERIVALE #1 NS	901
117	WB	11603	SB320	MEADOWLANDS	MERIVALE #1 NS	926
117	WB	E-690	SH900	BASELINE		910
117	WB	11601	SH900	BASELINE		840
117	WB	11603	CG500	LIBRARY	ARTS TOWER	909
117	WB	E-690	CG500	LIBRARY	ARTS TOWER	839
95	EB	9543	EE905	BLAIR	1B	2318
86	EB	8602	AL020	ELMVALE SC	11th STOP	928
97	EB	9702	NI915	LINCOLN FIELDS	2A: DOWNTOWN	444
86	EB	8603	AL020	ELMVALE SC	12th STOP	947
86	EB	8607	SH490	MEADOWLANDS	MERIVALE NS	1253

Figure 7.6: Schedule Information

If the user selects 'PROCEED', step one of the transit information system is run. This window, shown in Figure 7.7 establishes how the system will proceed through the system. In this step, the user is asked to provide the time, as previously mentioned, when the trip will begin, what day of the week (Weekday,

Saturday, or Sunday) the trip will take place and the direction. The travel information is established in this fashion due to the structure of OC Transpo's schedule system. In addition, the user establishes the selection method (by Address, Landmark, or Point_Click) that will be used to select his/her desired origin and destination. As per entering the time and day of travel, radio buttons also prevent the user from choosing more than one method of selecting an origin or destination. Under each method, a user may only select on of three choices. If the user selects Address, then the next dialog to appear will have the user enter a street address. It is not case sensitive, nor will the user need to enter a street type (Rd, St, Ave). The street name and number should be inputted correctly, although there is a tolerance for spelling errors.

If the user were to select Landmark, the next dialog to appear would have the user select a category (school, library, church), then the corresponding location from drop down menus. Finally, if Point_Click were chosen, the main view would re-appear, and the user would get instructions to select locations and solve the optimum route.

Depending on which selection the user has chosen for the destination, the transit information system will have the corresponding dialog appear for the user to enter his/her destination. Each dialog will appear in the same manner as described above.

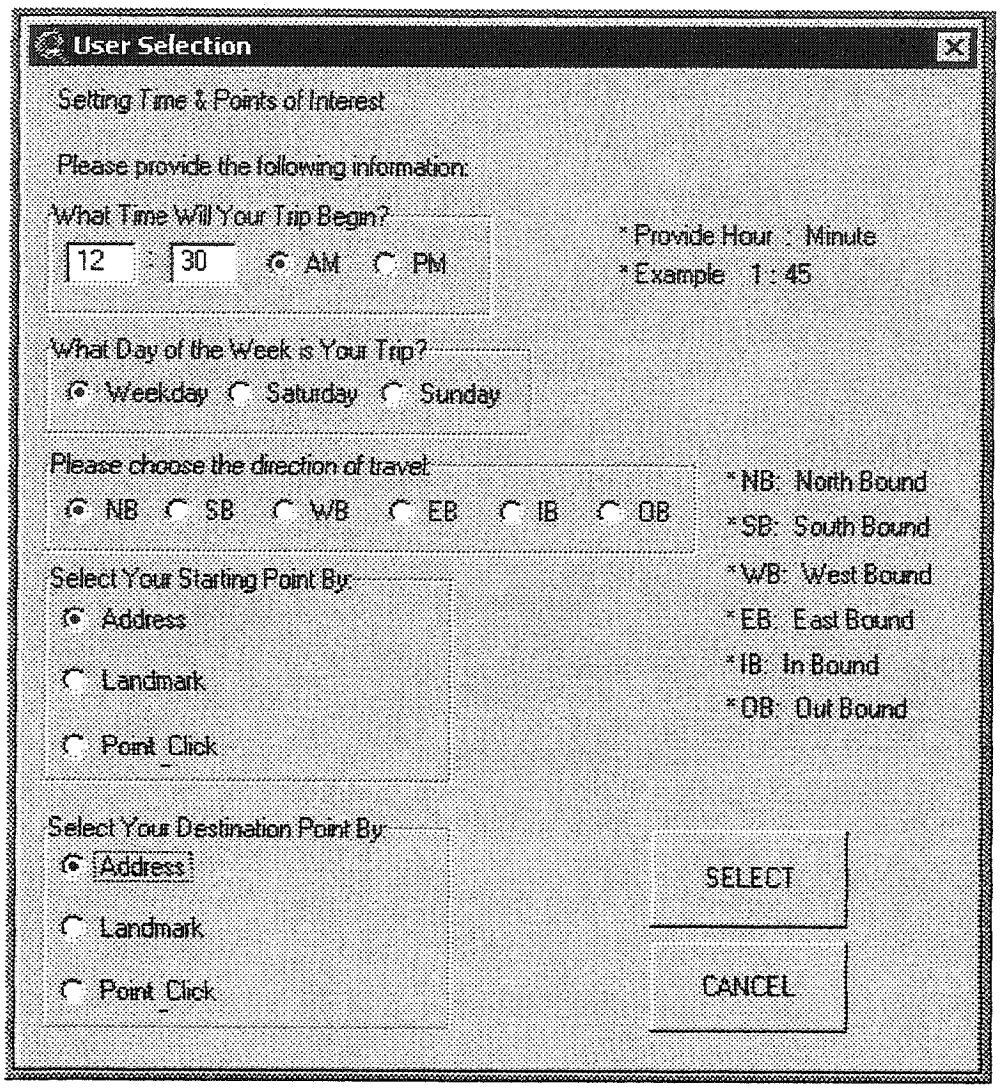


Figure 7.7: Dialog to Select Trip Information

When all information is provided, buttons allow the user to either continue to the next step or to cancel and restart the application.

Step two allows the user to select his/her desired starting point per the set selection methods requested in step one. Depending on the selection method set by the user, the page will vary to accommodate the information required.

If the user selects Address for either the origin or destination locations, the windows in Figure 7.8a/b will appear. The user enters the origin and/or destination location. When the user selects 'OK', this address is geocoded into the network through address matching with the street network. A point will appear in the view.

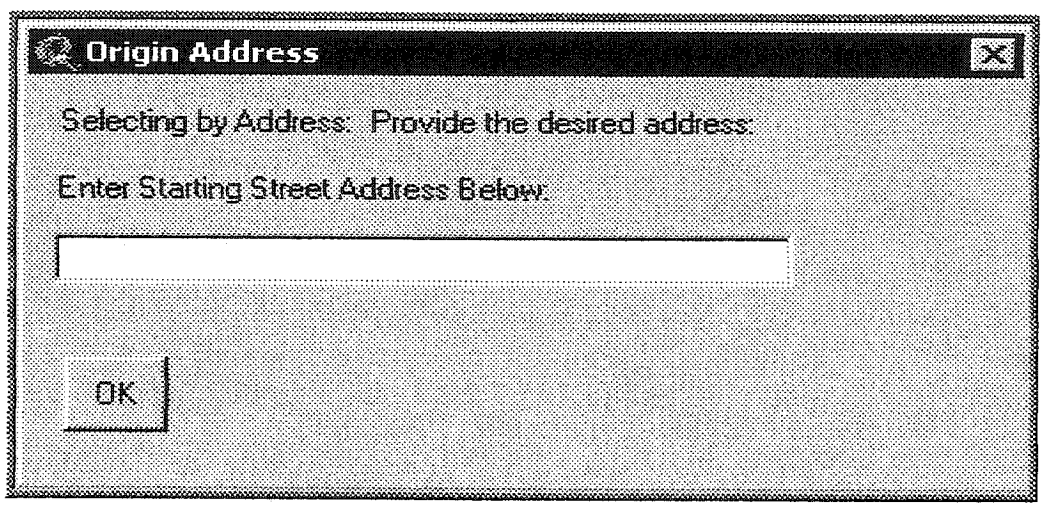


Figure 7.8a: Dialog to Enter the Origin Location

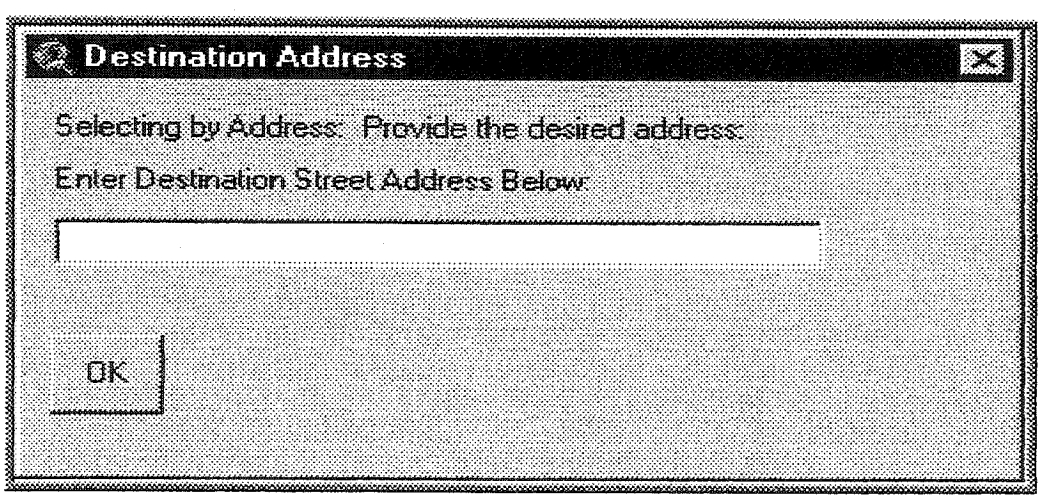


Figure 7.8b: Dialog to Enter the Destination Location

If the users were to select Landmark Selection from the User Interface, then the dialog shown in Figure 7.9a/b will appear. The users have the option of selecting their origin and/or destination location via a list provided by the application. These are the point themes added through address matching. The user will select a category and the corresponding attributes (landmarks) from the drop down menus. These will be added into the 'Location' table for route analysis.

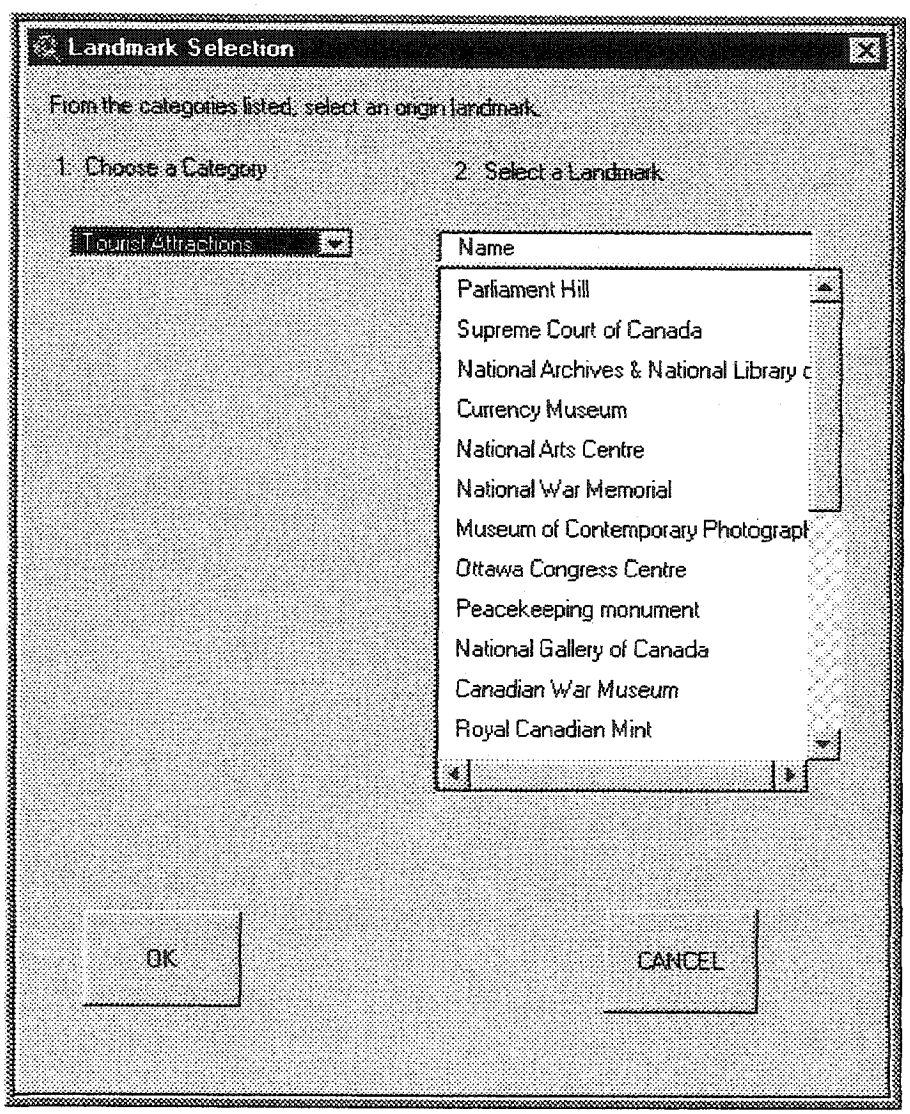


Figure 7.9a: Origin Landmark Selection

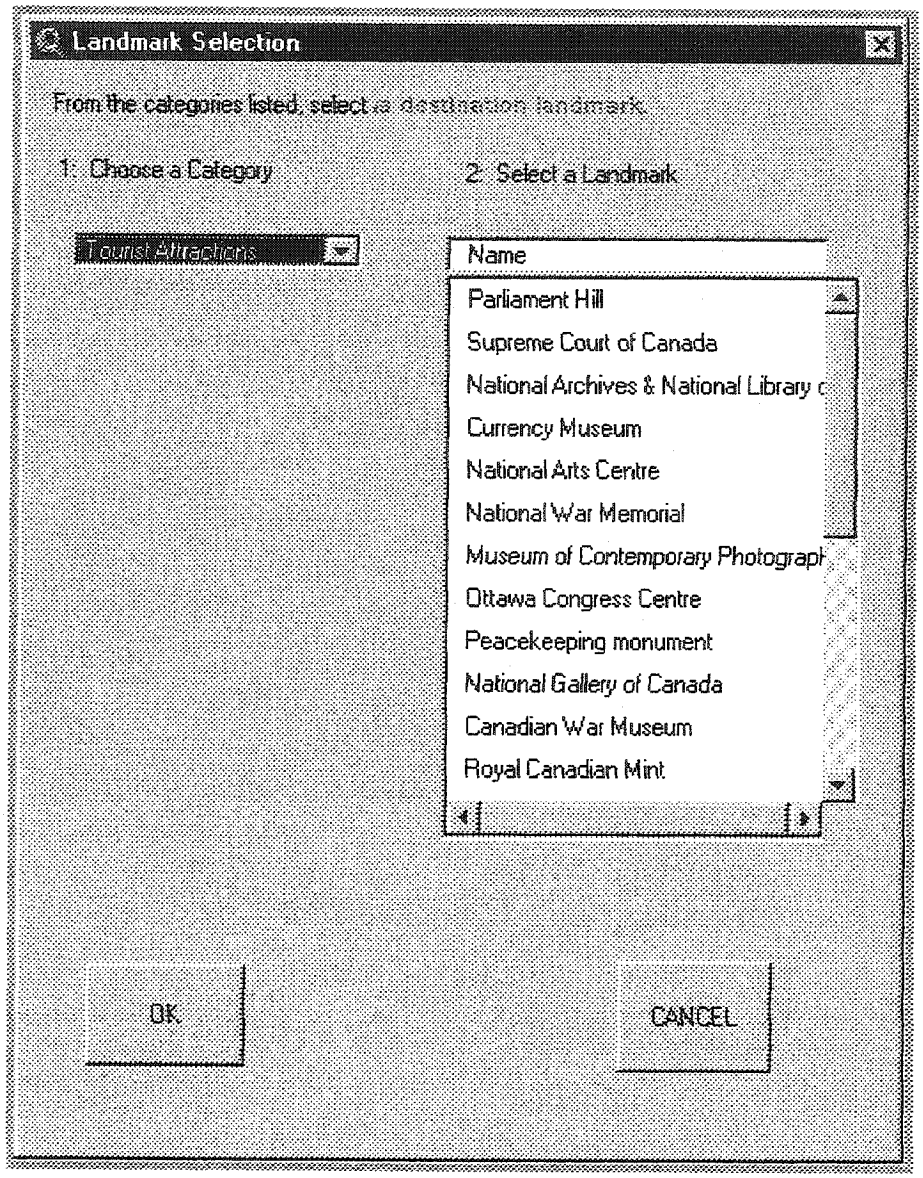


Figure 7.9b: Destination Landmark Selection

Once the user selects 'OK' for either the Address or Landmark Selection the message box in Figure 7.10 will appear. This directs the user to select the route button to solve the optimum route problem.

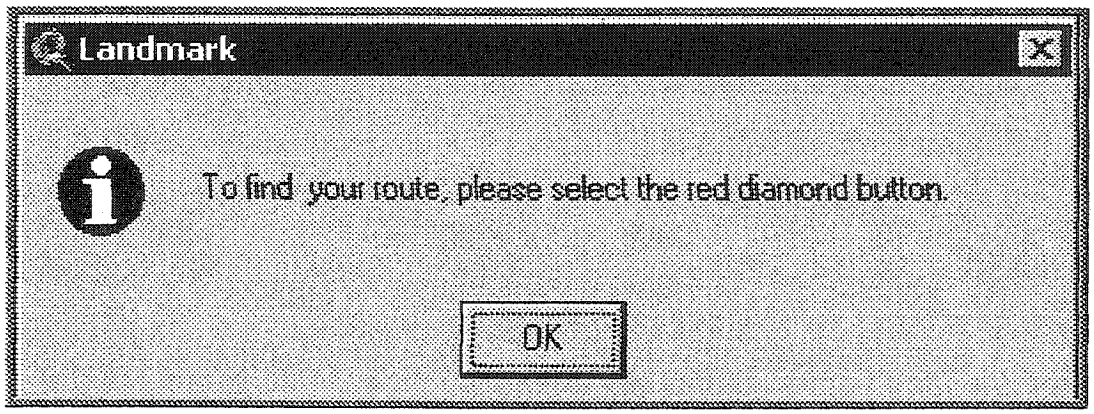


Figure 7.10: Message Box for Solving the Optimum Route

If the user is to select the origin and/ or destination by Point and Click then the dialog structure will close. Two message boxes will appear (Figures 7.11 & 7.12) indicating to the user, the method to select the origin and/or destination points to solve the route.

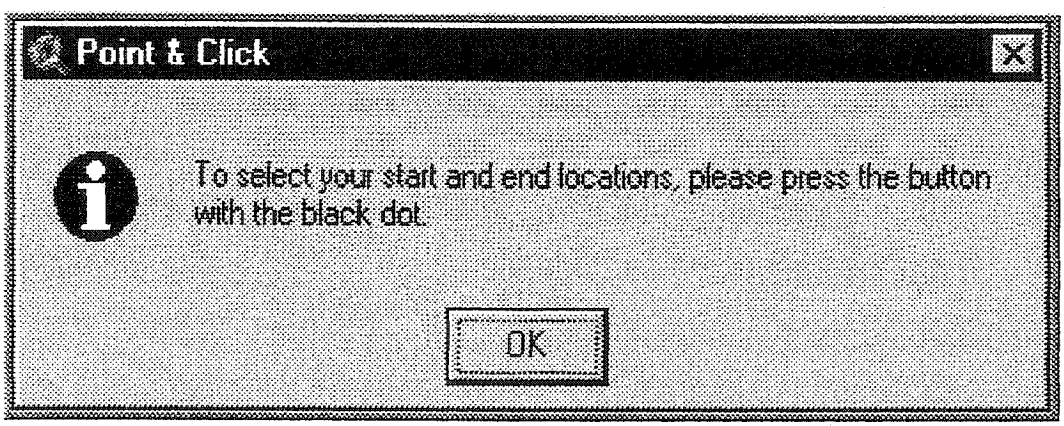


Figure 7.11: Message Box for Selecting the Origin/Destination

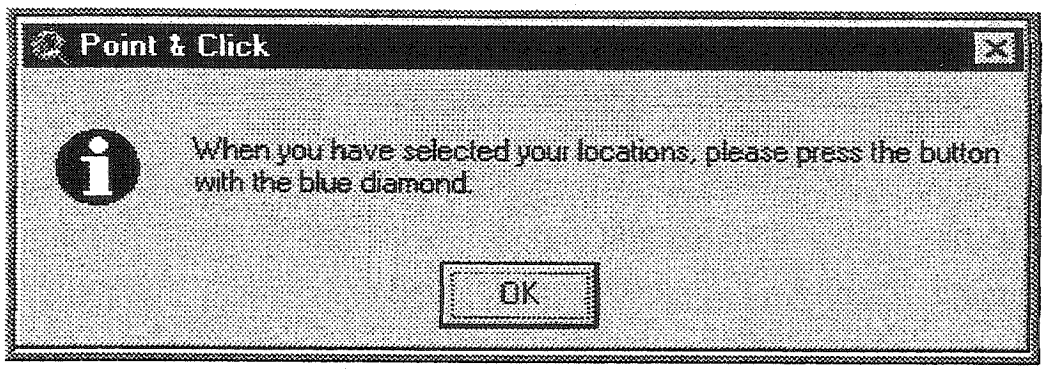


Figure 7.12: Message Box for Solving the Optimum Route using Point and Click

When designating the desired origin or destination point by using the Point_Click the user is provided with the transit routes network. This interface includes the map and functions to manipulate the map. These functions include zoom in, zoom out, pan, and select and are clearly located for easy access. Figure 7.13 illustrates the buttons and their location in the view. The Point_Click has a different route button since there is an additional step in solving the route problem. Given that the points entered by the user are graphics, they must be first converted to a theme in order for the application to solve the optimum route. This additional step is added into the route button along with the optimum route algorithm.

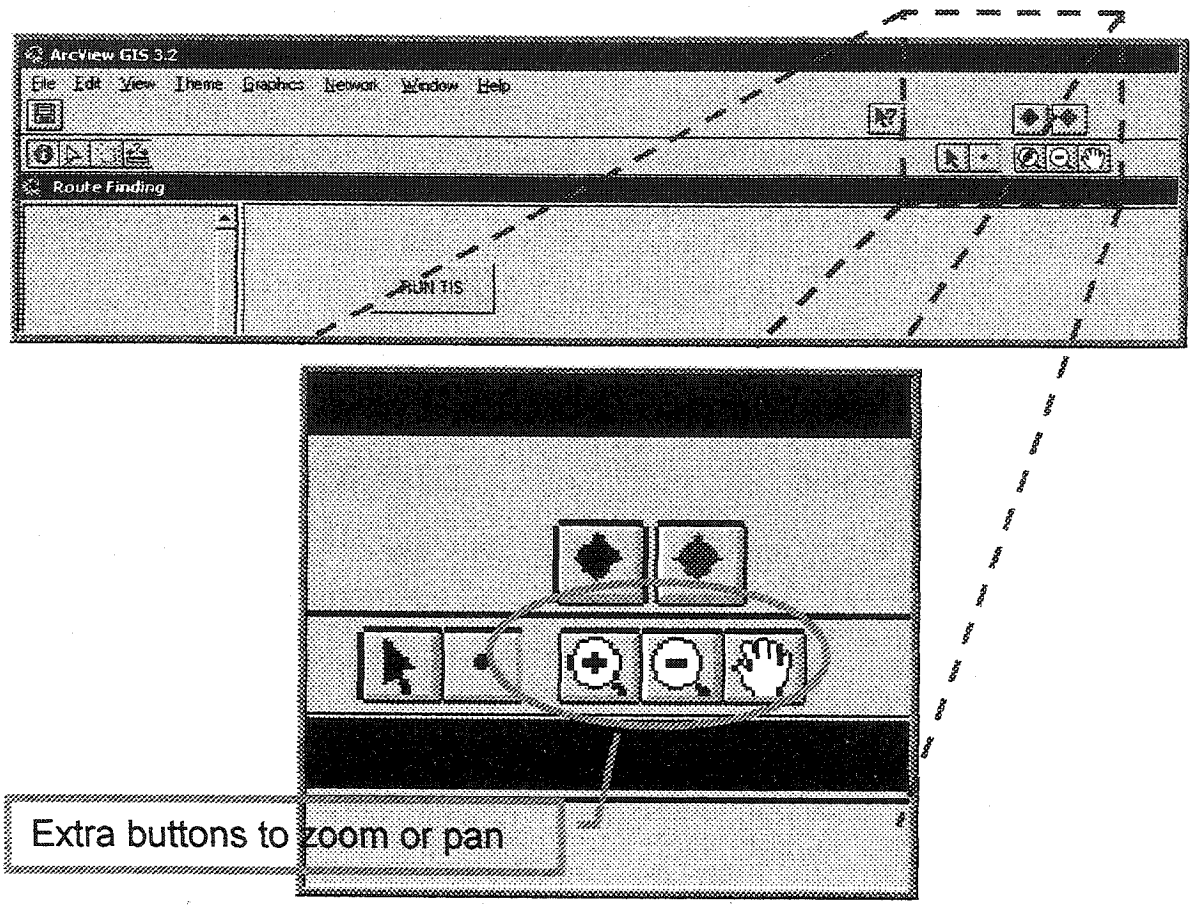


Figure 7.13: Buttons in Application

The final step is producing the results. To date the user must select the travel cost before the optimum route is found. This window produces only one cost (minutes) for the user to select. To eliminate this step the core system scripts for the Network Analyst need to be changed. This is a simple procedure, but due to the network violations it would create, this was not performed. If this application is to be implemented, the core system can be modified before use. Figure 7.14 illustrates this cost window.

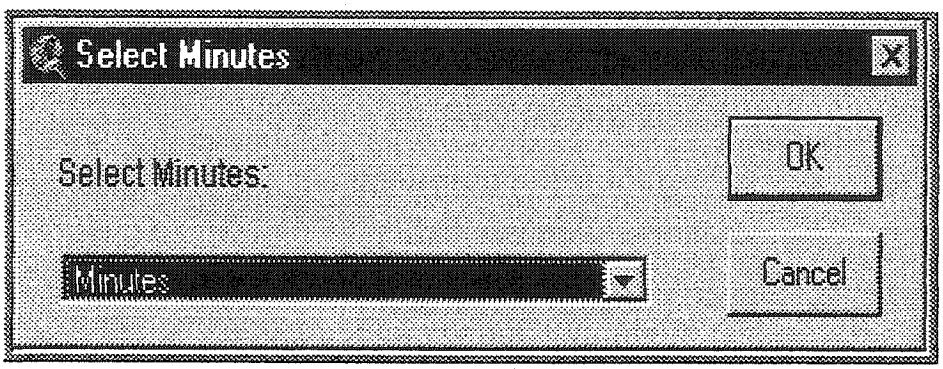


Figure 7.14: Cost Selection

The final step provides the user with the query results. This page consists of two parts. The first is an itinerary, which includes information on where the user must go to take the bus, which bus the user will board, when that bus will arrive, and approximately how long the trip will take. The second part of the results page is a map showing the origin and destination points and the optimum route. Sample itineraries and route illustrations are shown in Figures 7.15 and 7.16 respectively.

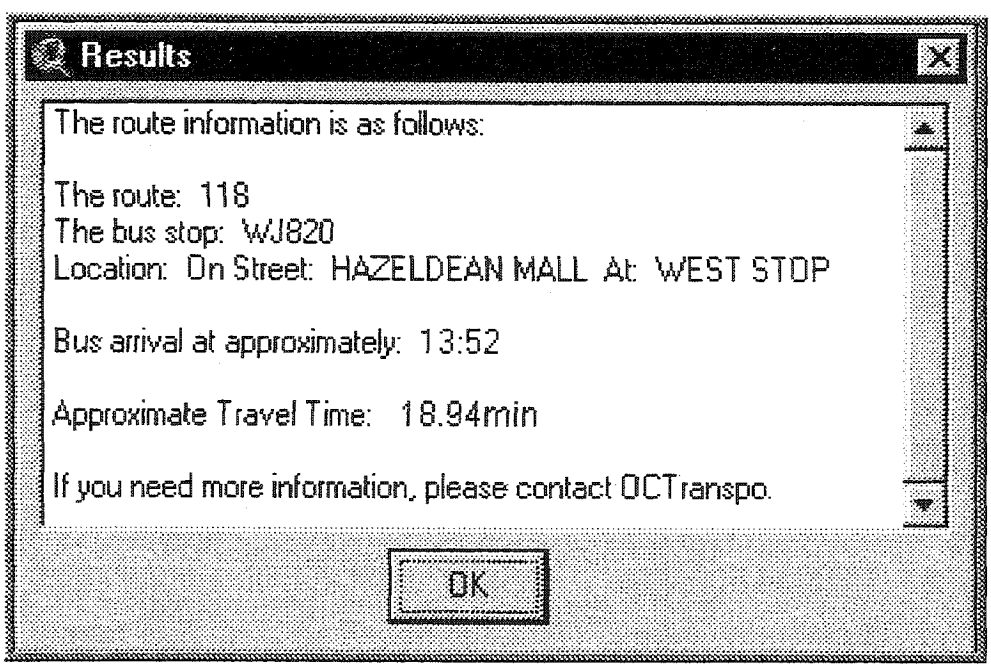


Figure 7.15: Report Box Reporting Results

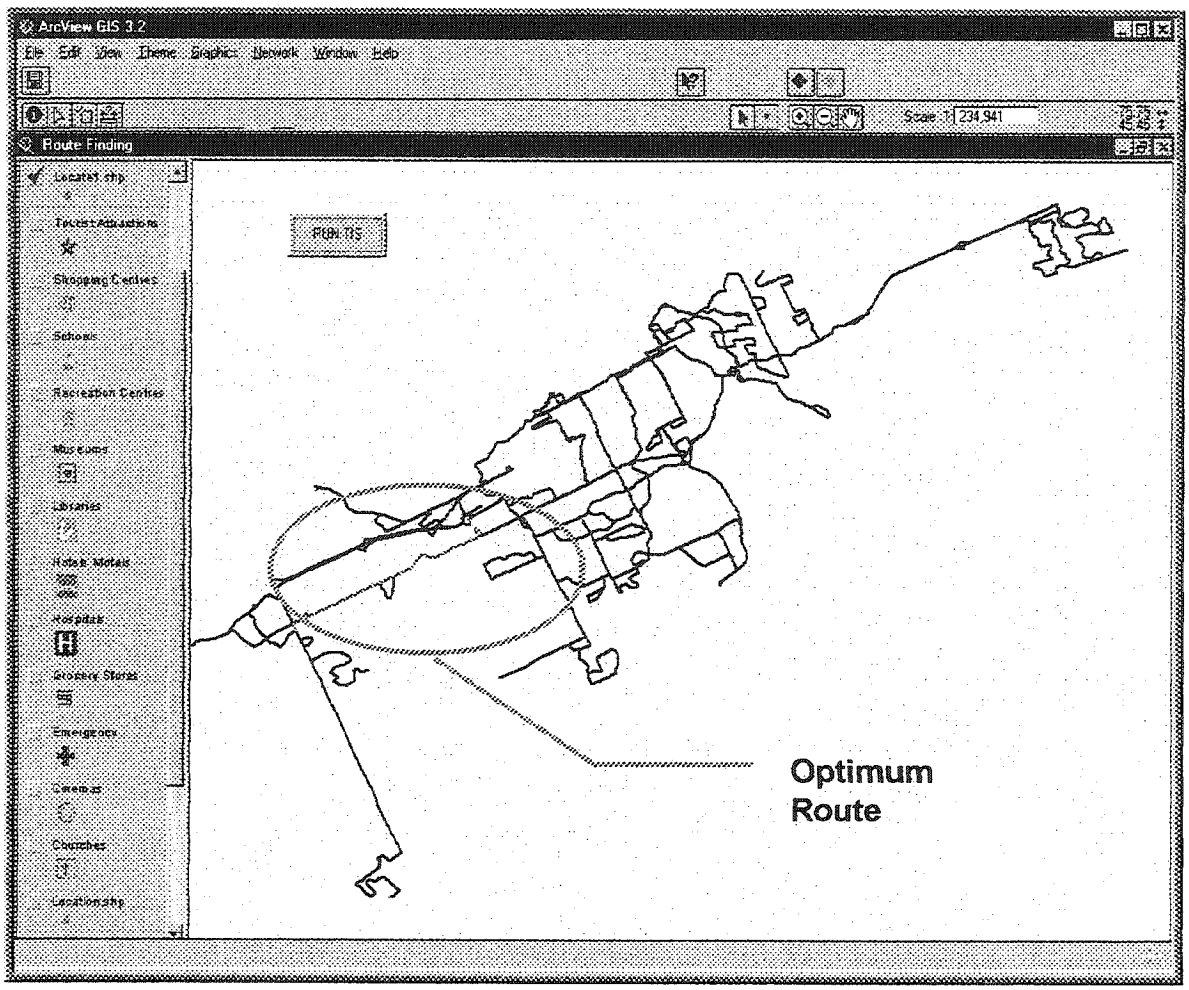


Figure 7.16: Results Image

The written results Itinerary will automatically print once the 'OK' button is selected and the user has the option to print the graphic route in the figure above. As well, the user can re-enter the transit information system to plan another trip or select landmarks from the left hand side of the view.

Chapter 8: System Validation

Once an application like this has been created, it needs to be tested. For this step, the key is to find all possible errors in the algorithm that a user might encounter. For this step several trial runs of the systems were conducted in an attempt to simulate possible user inputs. To date this has been successful and any errors encountered have been eliminated. A final version of the transit information system will ultimately indicate if this system is completely operational.

8.1 Criteria for Evaluation

8.1.1 Is the Trip Planner Functioning Properly and Returning Accurate Information?

This criterion is the most important for a successful trip planner. It is a necessity that the address entered by the user be recognized and located accurately by the transit information system. The system must be able to calculate an accurate route from the inputted information.

8.1.2 Is the System Attractive and Easy to Use?

For some transit riders, this trip planner will be their first introduction to OC Transpo's services. For this reason it is a necessity to create a positive impression and a pleasant trip

planning experience. The overall design and layout of the system should be colourful and the graphics should attract attention. If this service is successful, then the user will anticipate the same from the transit service. On the other hand, if the trip planner is not adequate, then OC Transpo will lose credibility with the users.

8.1.3 Further Criteria

Other criteria for evaluation could be developed when the system is implemented. These might be if the trip planner is used regularly. If it is then this will indicate that the public is aware of the new technology. If the public is not aware then further advertisement could be made. As well, criteria might be the satisfaction of the public with the trip planner. The main measure of the success of a topic like this is the response from the transit users themselves. If a person uses the trip planner to receive transit information and successfully completes the trip to his/her satisfaction then this system will be a success. This evaluation could be accomplished through user feedback through the Internet or informal surveys.

8.2 Performance Criteria Results

8.2.1 System Performance

Upon testing the system, sample runs were done in order to simulate a user trying to enter addresses, selected landmarks or

points. To date the system has been able to recognize addresses with minimal spelling errors and has no difficulties moving between selections.

In addition, sample trips between stations were made to test the optimum route algorithm with Transitway and non-Transitway trips. For the purposes of evaluation, routes 3, 7, and 118 were chosen for non-Transitway routes and routes 95 and 97 for Transitway routes. These trips were performed for weekday and weekend trips. Since the schedule databases for both weekday and weekend trips produced the same headways between time points only weekday trips are presented here. In Table 8.1, a comparison of the scheduled travel times designated by OC Transpo and that calculated through the transit information system is illustrated. Since it is necessary to validate the accuracy of the route algorithm, Bayesian Analysis was not initially applied at this stage. The percent difference is used to demonstrate the precision between the travel times.

Table 8.1: Weekday Comparison Between Schedule and Algorithm

Route No.	OCTranspo Schedule (min)	From Location	To Location	TIS Results (min)	Percent Difference
3	9	Colonnade	Fisher/Normandy	9.43	4.8%
off-peak	13	Fisher/Normandy	Preston/Carling	14.25	9.6%
morning	6	Preston/Carling	Albert/Booth	7.13	18.8%
	31	Albert/Booth	Hurdman Station	28.4	-8.4%
	59		Cumulative	59.21	0.4%
7	10	Carleton U/Library	Bank/Coliseum	9.1	-9.0%
off-peak	21	Bank/Coliseum	Rideau Center	20.81	-0.9%
afternoon	6	Rideau Center	St. Patrick Bridge	7.55	25.8%
	25	St. Patrick Bridge	St. Laurent Station	25.31	1.2%
	62		Cumulative	62.77	1.2%
95	15	Place d'Orleans	St. Laurent Station	11.97	-20.2%
peak	4	St. Laurent Station	Hurdman Station	4.5	12.5%
afternoon	14	Hurdman Station	Lebreton Station	14.56	4.0%
	10	Lebreton Station	Lincoln Fields Sta	11.06	10.6%
	4	Lincoln Fields Sta	Baseline Station	5.3	32.5%
	47		Cumulative	47.39	0.8%
97	24	Kanata Town Cent	Lincoln Fields Sta	21.45	-10.6%
off-peak	10	Lincoln Fields Sta	Lebreton Station	12.96	29.6%
evening	13	Lebreton Station	Hurdman Station	14.56	12.0%
	11	Hurdman Station	South Keys	12.1	10.0%
	6	South Keys	Airport	4.34	-27.7%
	64		Cumulative	65.41	2.2%
118					
off-peak	24	Hazeldean Mall	Baseline Station	23.8	-0.8%
afternoon	23	Baseline Station	Billings Bridge	22.63	-1.6%
	47		Cumulative	46.43	-1.2%

As seen, there are some discrepancies with some of the travel times. Many of these occur in short distances, where a difference of 30 seconds causes a significant percent difference. Route 95 is an example of this where half a minute to a minute causes a percent difference of up to 30%.

When the delay costs were entered into the algorithm, it took into consideration the wait time at stations from travellers leaving and boarding the bus. Linear interpolation is also used to estimate the arrival of buses at stops where there is no scheduled time. This can also add to the error within the calculated travel time. Estimates of the arrival of buses at certain bus stops based on the travel distance carry error which could explain the additional time in comparison to the schedule. Additional work may entail obtaining trip frequency results from OC Transpo on the travel time between stations for comparison. These trips could then be statistically analyzed for error calculations and incorporated into the algorithm.

In the case of travel time being less than that of the schedule, these occur in large distances between stations where travel speeds are greater than 80km/hr (Queensway and/or Transitway). This can be seen from routes 95 and 97. Both these routes travel

predominantly on the Transitway as well as a portion on the Queensway. Route 95 travels from St. Laurent to Place d'Orleans on the Queensway while route 97 travels the Queensway from Kanata to Lincoln Fields. In both cases there are no or few stops between stations leaving these travel times to the uncertainty of the road conditions. Future work may entail comparing travel times on high speed zones in order to represent these sections of the route more accurately. Nevertheless, in most cases when a bus actually arrives early at a time point, it waits until the designated time to depart. This allows for later arriving travellers to obtain their transit ride.

As observed in the cumulative trip time, the percent differences have been minimized. It balances because depending on the location of where the particular graphic point was placed within the transit stations; it would include or not include the individual wait time. As a suggestion to avoid this in the future, stations should be coded differently from bus stops. Presently the transit stations include multiple stop locations and if the user was unsure where the bus arrives or departs, an inconsistency would ensue. If a user was to put the point at the beginning of a transit station or at the end then the travel time would vary because the transit station costs would either be included or excluded. Overall, from one end

of the route to the other the costs have been accounted for accordingly and appropriate travel times have been recorded.

Based on the results of the algorithm, trial runs were executed given each condition of early, on time, or late. Bayesian analysis was applied for both weekday and weekend trips to visualize the different travel conditions instituted by the probabilities. These results are illustrated Tables 8.2 to 8.4.

As indicated in chapter 6, the prior probabilities are applied where there is no travel time change advisory. In this case, the travel time is recalculated using the appropriate prior probabilities. Where there is a travel time change advisory, conditional probabilities are used to estimate posterior probabilities that are then applied to update the travel time.

Table 8.2: Application of Probabilities to Weekday Trips

				No Statistics	Prior Probability	Posterior Probabilities		
Route No.	OCTranspo Travel Time (min)	From Location	To Location	Results (min)	Results On Time (min)	Results Early (min)	Results On Time (min)	Results Late (min)
3	9	Colonnade	Fisher/Normandy	9.43	9.32	8.74	9.37	10.32
	13	Fisher/Normandy	Preston/Carling	14.25	14.14	13.56	14.19	15.14
	6	Preston/Carling	Albert/Booth	7.13	7.02	6.44	7.07	8.02
	31	Albert/Booth	Hurdman Station	28.4	28.29	27.71	28.34	29.29
Sum	59		Cumulative	59.21	58.75	56.45	58.97	62.77
7	10	Carleton U/Library Rd	Bank/Coliseum	9.1	8.99	8.41	9.04	9.99
	21	Bank/Coliseum	Rideau Center	20.81	20.70	20.12	20.75	21.7
	6	Rideau Center	St. Patrick Bridge	7.55	7.44	6.86	7.49	8.44
	25	St. Patrick Bridge	St. Laurent Station	25.31	25.20	24.62	25.25	26.2
Sum	62		Cumulative	62.77	62.31	60.01	62.53	66.33
95	15	Place d'Orleans	St. Laurent Station	11.97	12.01	11.38	11.99	13.07
	4	St. Laurent Station	Hurdman Station	4.5	4.54	3.9	4.51	5.6
	14	Hurdman Station	Lebreton Station	14.56	14.60	13.97	14.57	15.66
	10	Lebreton Station	Lincoln Fields Station	11.06	11.10	10.47	11.07	12.16
	4	Lincoln Fields Station	Baseline Station	5.3	5.34	4.71	5.32	6.4
Sum	47		Cumulative	47.39	47.59	44.43	47.46	52.89
97	24	Kanata Town Center	Lincoln Fields Station	21.45	21.68	20.95	21.55	22.84
	10	Lincoln Fields Station	Lebreton Station	12.96	13.19	12.46	13.06	14.36
	13	Lebreton Station	Hurdman Station	14.56	14.79	14	14.6	15.77
	11	Hurdman Station	South Keys	12.1	12.33	11.6	12.2	13.5
	6	South Keys	Airport	4.34	4.57	3.84	4.44	5.74
Sum	64		Cumulative	65.41	66.54	62.85	65.85	72.21
118	24	Hazeldean Mall	Baseline Station	23.2	23.09	22.5	23.1	23.76
	23	Baseline Station	Billings Bridge	22.63	22.55	21.94	22.57	23.52
Sum	47		Cumulative	45.83	45.64	44.44	45.67	47.28

Table 8.3: Application of Probabilities to Saturday Trips

				No Statistics	Prior Probability	Posterior Probabilities		
Route No	OCTranspo Travel Time (min)	From Location	To Location	Results (min)	Results On Time (min)	Results Early (min)	Results On Time (min)	Results Late (min)
3	9	Colonnade	Fisher/Normandy	9.43	9.59	8.89	9.5	10.75
	13	Fisher/Normandy	Preston/Carling	14.25	14.41	13.71	14.32	15.56
	6	Preston/Carling	Albert/Booth	7.13	7.29	6.59	7.2	8.44
	31	Albert/Booth	Hurdman Station	28.4	28.56	27.86	28.47	29.71
Sum	59		Cumulative	59.21	59.85	57.05	59.49	64.46
7	10	Carleton U/Library Ro	Bank/Coliseum	9.1	9.26	8.59	9.37	10.04
	21	Bank/Coliseum	Rideau Center	20.81	20.97	20.3	21.08	22.57
	6	Rideau Center	St. Patrick Bridge	7.55	7.71	7.04	7.82	9.49
	25	St. Patrick Bridge	St. Laurent Station	25.31	25.47	24.8	25.57	27.25
Sum	62		Cumulative	62.77	63.41	60.73	63.84	69.35
95	15	Place d'Orleans	St. Laurent Station	11.97	11.89	11.37	11.93	12.69
	4	St. Laurent Station	Hurdman Station	4.5	4.42	3.89	4.46	5.21
	14	Hurdman Station	Lebreton Station	14.56	14.48	13.96	14.52	15.28
	10	Lebreton Station	Lincoln Fields Station	11.06	10.98	10.46	11.02	11.77
	4	Lincoln Fields Station	Baseline Station	5.3	5.22	4.69	5.26	6.01
Sum	47		Cumulative	47.39	46.99	44.37	47.19	50.96
97	24	Kanata Town Center	Lincoln Fields Station	21.45	21.52	20.92	21.48	22.49
	10	Lincoln Fields Station	Lebreton Station	12.96	13.03	12.42	12.98	14
	13	Lebreton Station	Hurdman Station	14.56	14.63	14.02	14.58	15.6
	11	Hurdman Station	South Keys	12.1	12.17	11.56	12.12	13.14
	6	South Keys	Airport	4.34	4.41	3.8	4.36	5.37
Sum	64		Cumulative	65.41	65.74	62.72	65.52	70.6
118	24	Hazeldean Mall	Baseline Station	23.2	23.36	23.24	23.21	24.66
	23	Baseline Station	Billings Bridge	22.63	22.79	22.09	22.7	23.94
Sum	47		Cumulative	45.83	46.15	45.33	45.91	48.6

Table 8.4: Application of Probabilities to Sunday Trips

				No Statistics	Prior Probability	Posterior Probabilities		
Route No	OCTranspo Travel Time (min)	From Location	To Location	Results (min)	Results On Time (min)	Results Early (min)	Results On Time(min)	Results Late (min)
3	9	Colonnade	Fisher/Normandy	9.43	9.60	8.92	9.5	10.7
	13	Fisher/Normandy	Preston/Carling	14.25	14.42	13.74	14.32	15.52
	6	Preston/Carling	Albert/Booth	7.13	7.30	6.62	7.2	8.4
	31	Albert/Booth	Hurdman Station	28.4	28.57	27.89	28.47	29.67
Sum	59		Cumulative	59.21	59.87	57.17	59.49	64.29
7	10	Carleton U/Library Rc	Bank/Coliseum	9.1	9.27	8.59	9.17	10.37
	21	Bank/Coliseum	Rideau Center	20.81	20.98	20.3	20.88	22.08
	6	Rideau Center	St. Patrick Bridge	7.55	7.72	7.04	7.62	8.82
	25	St. Patrick Bridge	St. Laurent Station	25.31	25.48	24.8	25.38	26.58
Sum	62		Cumulative	62.77	63.43	60.73	63.05	67.85
95	15	Place d'Orleans	St. Laurent Station	11.97	12.13	11.55	12.03	13.08
	4	St. Laurent Station	Hurdman Station	4.5	4.66	4.08	4.56	5.61
	14	Hurdman Station	Lebreton Station	14.56	14.72	14.14	14.62	15.66
	10	Lebreton Station	Lincoln Fields Station	11.06	11.22	10.64	11.12	12.16
	4	Lincoln Fields Station	Baseline Station	5.3	5.46	4.88	5.36	6.4
Sum	47		Cumulative	47.39	48.19	45.29	47.69	52.91
97	24	Kanata Town Center	Lincoln Fields Station	21.45	21.44	21.12	21.59	22.88
	10	Lincoln Fields Station	Lebreton Station	12.96	12.95	12.63	13.1	14.39
	13	Lebreton Station	Hurdman Station	14.56	14.55	14.23	14.7	15.99
	11	Hurdman Station	South Keys	12.1	12.09	11.77	12.24	13.53
	6	South Keys	Airport	4.34	4.33	4.01	4.48	5.77
Sum	64		Cumulative	65.41	65.36	63.76	66.11	72.56
118	24	Hazeldean Mall	Baseline Station	23.2	23.37	23.14	23.13	24.39
	23	Baseline Station	Billings Bridge	22.63	22.80	21.98	22.6	23.56
Sum	47		Cumulative	45.83	46.16	45.12	45.73	47.95

In the previous tables, the results obtained from the Bayesian Analysis clearly show how the uncertainty of transit travel can be minimized. There is an obvious difference in travel time given the criteria of being early, on time, or late.

Depending on the applied probabilities to the corresponding time period, the bus arrival can vary up to 10 - 15% representing early to late at a particular bus stop. This percentage demonstrates how a bus could arrive between 3 to 8 minutes early or late as compared to the actual scheduled time.

An excellent example of this variability is shown in routes 95 and 97 for weekday travel. The total travel time varies from 3 minutes early to roughly 5 to 7 minutes late. This difference can significantly impact when a person arrives at a transit stop and the entailed waiting time. By minimizing the uncertainty, the anxiety and frustration of waiting or missing a bus is reduced. People can be more aware and plan daily routines more accurately given the travel time with the inclusion of uncertainty through Bayesian Analysis.

The probabilities also indicate how variability changes from weekday, Saturday, and Sunday travel. Given the departure time is within the same time period, the resulting travel times vary from day to day. Since travel time on weekends tend to vary differently due to reduced passenger loads and congestion changes, the probabilities are altered.

Depending on the route and which area of the city it serves, the probabilities will differ. If a route serves a large number of passengers and the impact by congestion from mixed traffic does not change significantly, the results obtained from weekday travel to weekend travel can be small.

An example of this is shown using Route 118. From the previous three tables, route 118 only has travel times varying from 1 to 3 minutes of being early or late arriving at a transit stop. This route travels directly between Kanata and Billings Bridge stations having two shopping centers at either end. The route uses two major roads where traffic conditions do not vary significantly between weekday and weekend travel. Therefore, its uncertainty between days varies less throughout the week. .

As shown in chapter 5, a loop back in to the algorithm is performed to check for another route with less travel time based on the Bayesian application. An attempt was made to show a change in the optimality of a route due to the Bayesian Analysis. Unfortunately due to the limited number of routes used for the transit information system, no alternate routes were obtained for illustration purposes. Since only 23 routes were used instead of the entire network, many routes do not overlap or travel to identical origins and destinations. There are restrictions to where and when transit vehicles travel, therefore it is difficult to obtain another optimal route with a small number of routes. Given the opportunity of applying the entire network, as shown in Figure 8.1, the possibility of having the Bayesian Analysis altering an optimal bus route would be greatly enhanced.

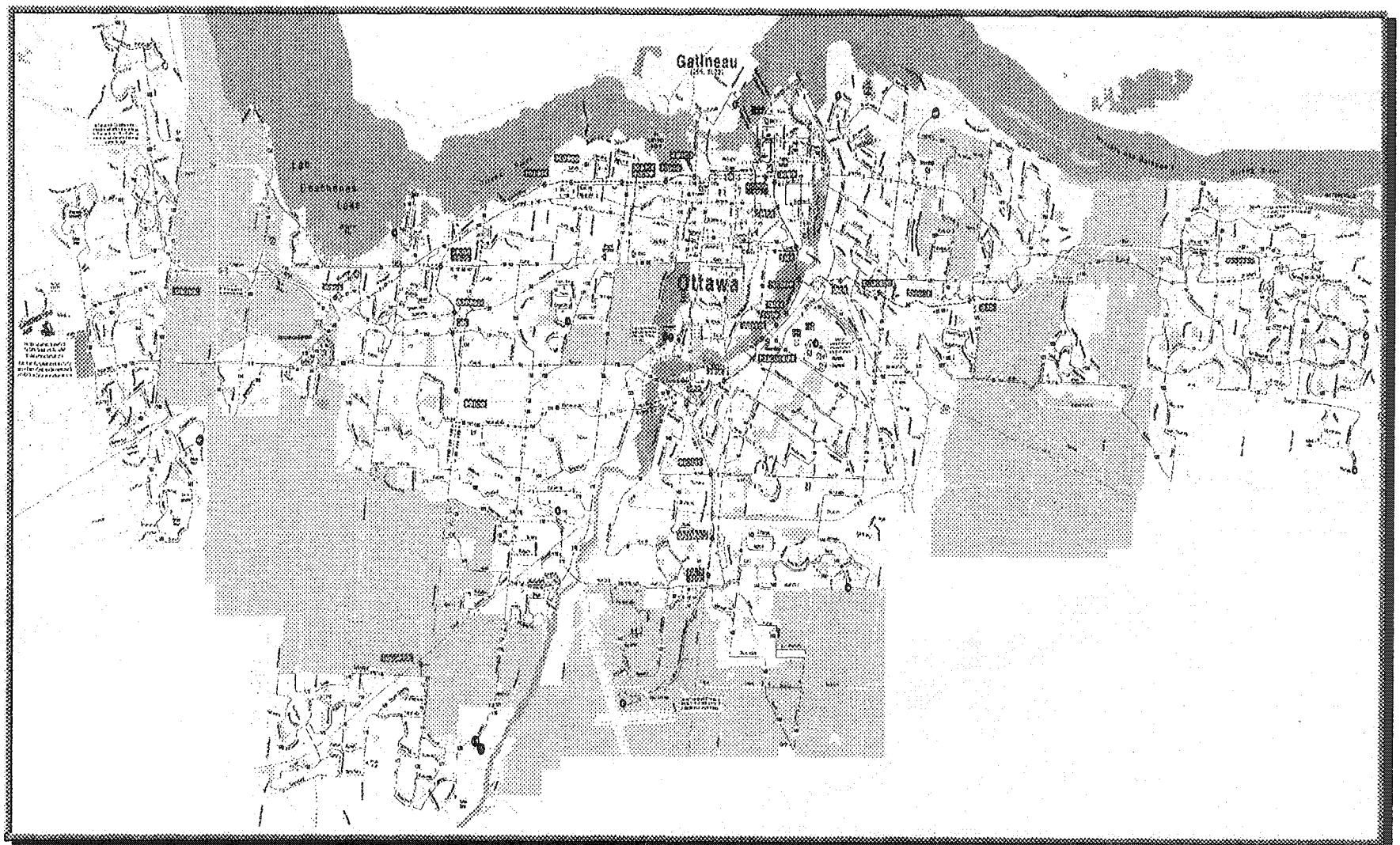


Figure 8.1: OC Transpo Route Network (www.octranspo.com)

The Bayesian analysis is also useful and is applied in cases where a route could be detoured or diverted for a long or short period of time due to road construction, protests, Canada Day festivities, or a major fire. In cases where there is a major event being celebrated or construction occurring, the City of Ottawa and OC Transpo work together to let the public know of street closures and schedule changes ahead of time. OC Transpo and the emergency services in Ottawa presently work together to alert each other of problems along the roadways. So, if there is a major traffic accident along a particular route, drivers can let the dispatchers know and can then work together to help in establishing a detour in order to avoid the situation.

When route changes are required, OC Transpo has the ability to recode the line segments pertaining to the specific route within the attribute table. As stated in Chapter 5, line segments can be coded to represent one-way streets and closed streets. Using the appropriate code, routes may be changed and an advisory given for a late arrival. The Bayesian analysis would be applied to the new travel time with respect to the revised route and the suitable conditional probabilities will be assigned based on the time period.

A new travel time is calculated based on the pertaining costs for the new section of road. For the most part, the bus will not stop and pick-up or drop-off any passengers until the regular route is once again being traveled upon. Therefore the new costs only include the speed limit and other relevant road restrictions. For the bus stops not being used for this period of time, OC Transpo can code them as 'not in use' for the designated period of time.

An example is given below where Route 137 has been given a detour and a late advisory. In Figure 8.2, the prior route is shown in the solid line and the new route upon given the detour is shown in the dashed line. In this case, a section of Centrum Road is closed and the bus must now travel along St. Joseph Blvd. The travel time given by the schedule is 14 minutes and the prior expected travel time is 14.63 minutes. Once the late advisory is given, the new travel time is calculated to be 16.86 minutes based on the new travel route and the conditional probabilities. In this case the route travel time has increased since the detour is somewhat lengthier and the late advisory takes into consideration any uncertainty that may arise within the detour.

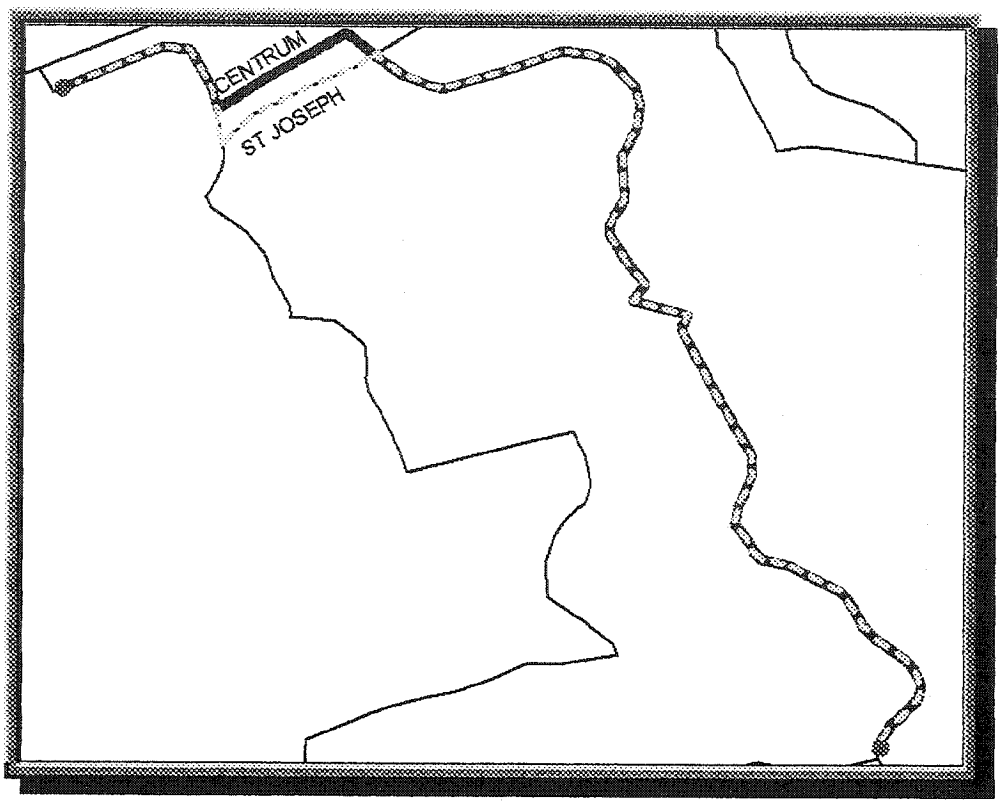


Figure 8.2: Path Change Due to Detour

In another example, the bus route number to travel may change. In this case the sub optimal, indirect route may be required. Two bus routes travel to the same location but one has been blocked off due to a residential fire. The indirect route is being used to detour traffic. As shown below, the prior path is route 116 (solid line) travelling from Meadowlands and Woodroffe to Merivale Mall. This expected travel time found through the travel information system is 8.92 minutes.

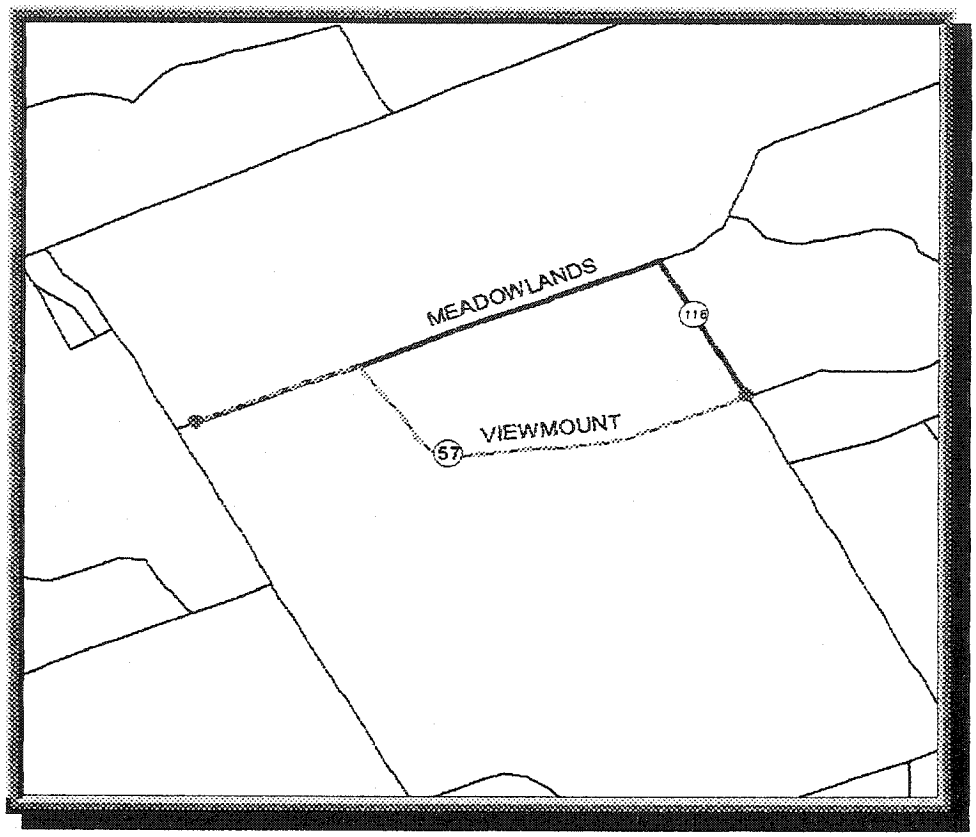


Figure 8.3: Route Number Change Due to Detour

Given that Meadowlands Drive from Viewmount Drive to Merivale Road is blocked off, vehicles must now travel via Viewmount Drive. The new route found from the transit information system is route 57 (dashed line) and its expected travel time is 9.91 minutes. As mentioned above, OC Transpo would give a late advisory and the corresponding conditional probabilities would be applied to give a new expected travel time of 11.2 minutes. This new time takes into consideration route 57's uncertainty due to passenger loading and congestion. The bus stops located along the closed section of

Meadowlands Drive would be given a 'not in use' label and the closest bus stop to Viewmount Drive would be designated for the user.

Since a detour is a major deterrent, OC Transpo can also include a written advisory on its web site and within the transit information system to allow users to plan their trips wisely and allow extra time for travel.

Overall, by distinguishing between the travel time periods of peak and off-peak, evening to early morning, the travel time uncertainty is appropriately accounted for. Each time period and day of the week vary in accordance with the amount of congestion on the route be it passenger or traffic. By applying Bayesian analysis, the user can obtain reasonable information pertaining to route conditions and plan their arrival at transit stops more efficiently. They become more confident and less worrisome of travel time to a destination. Passengers appreciate knowing how travel will vary from day to day.

8.2.2 Attractiveness and Ease of Use

In regard to the attractiveness and ease of use of the system, this can only be truly evaluated once executed. As it stands, the author

believes, the overall design of the system is structured for easy understanding, the layout is colourful, and the graphics draw attention. Since, this trip planner will be the first of this type introduced to OC Transpo's services, it is assumed that it will create a positive impression and a pleasant trip planning experience. The transit information system should be a successful and credible implementation.

Chapter 9: Summary, Conclusions, and Recommendations

9.1 Summary

Like most businesses and government facilities, transit agencies are under increasing pressure to do more with less and to provide better customer services. The development of a transit information system provides part of the high-tech approach for transit agencies to seek more productive solutions. The system enables transit agencies to provide more efficient customer services to transit users.

The success of GIS implementation at transit agencies demands an integrated GIS database to meet the needs of a wide range of applications. The challenge of the transit GIS database design is to integrate spatial features in the transit network and the temporal service variations.

The spatial features in the transit network (route, segments, patterns, stops and time points) can be logically linked to linear and location referencing. It also registers the transit network to the street network and other geographic features like landmarks, which provides the basis for geocoding and trip planning. Considerable effort was required when designing the structure of the database tables. By considering the maintenance of the system throughout the design

process, changes can be easily completed through the databases, without changing the core source code.

By developing this user-friendly transit information system, transit agencies can produce efficient customer service while using an integrated geographic information system. Within the application, an analysis of the Ottawa street network, transit routes and transit stops is performed. Schedule databases were created to link to transit attribute tables while addresses were geocoded, network costs entered, and speed limits allocated. It provides a quick method to find a route and obtain a printed copy of an itinerary. It has the capability of instantly planning an optimal transit trip based on the user's trip origin, destination and the time of travel.

With the lack of real-time information and the limitations of making prediction of arrival times, the application of Bayesian statistics helps to reduce the uncertainty found with transit travel by applying the appropriate probability based on the travel time period and the AVL/C advisory. With this technique, the user can obtain information pertaining to the route in order to travel efficiently. Be it weekday or weekend travel, the suitable probabilities will be applied to represent the conditions of the route.

While this application has been developed based on the information gathered from OC Transpo and the City of Ottawa, careful consideration was made during

the design process to ensure that it is generic. This system can be integrated into any transit system by simply changing the graphical information (routes, bus stops, landmarks). It is designed so the agency can make changes easily and quickly to the spatial data in the GIS environment. The core algorithm merely requires the statistical information to be updated to enable accurate system performance. In accordance with the long-term strategies of OC Transpo, this implementation would enhance the current passenger information systems available and overall, this system can be widely used and become very useful to potential and current passengers.

9.2 Conclusions

An advanced transit information system is developed by incorporating the ITS and GIS technologies and statistical analysis methodology for updating travel time information on the basis of transit advisories regarding bus movements. Results obtained from the developed transit information system suggest that this system will be useful. Sample runs indicate the ease of use and identification of procedures in order to find the optimal route of travel and corresponding travel time. This kind of analysis enables visual orientation with written instructions. Overall, the following conclusions can be drawn that pertain to the development and applicability of the transit information system.

- The research presented in this thesis illustrates the development of a transit information system using GIS and ITS technologies as well as Bayesian Statistical Analysis.

- This system is unique by its consideration of the uncertainty in travel time and schedule adherence as compared to other transit path finding methods.

- The Bayesian methodology is used to handle uncertainty in schedule adherence. This feature makes the best use of AVL/C information obtained from the transit agency.

- The object-oriented data model design simplifies data management and maintenance as well as increases the efficiency of network analysis, including spatial search, query, and shortest path finding.

- The user interface identifies user inputs and enables abstraction of the transit system in order to determine an optimum route.

- It considers the conditions of travel as assigned by the transit agency. If the travel time is consistent with the scheduled time, then the prior probabilities applied account for the possibility of lack of adherence due to the projected nature of a journey yet to be completed. If the advisory indicates that travel

running off of the schedule, then this new information is used to estimate posterior probabilities to calculate travel time.

- The developed information system demonstrates that GIS is a powerful tool to develop flexible and versatile functions and to deliver large amount of information to the user.

- The use of ArcView GIS software was found to be very helpful because many functions are integrated into the software and require a minimal amount of new programming software code for any specific routine, with the exception of the transit route-planning and statistical analysis algorithm.

- The interactive map-based software allows the system to incorporate other information such as shopping centers, cinemas, and other local attractions.

- The optimum route finding algorithm combines schedule-based information and statistical analysis to fit the unique characteristics of transit networks.

- By linking to the transit agency, information on the status of buses is acquired for the algorithm procedures.

- This system has the capability of instantly planning an optimal transit trip based on the user's origin, destination, and the day and time of travel.

- The system ignores the routes that are not in service at the time of travel.

- An optimal route is found based on the information given by the user and updated travel times are outputted.

9.3 Recommendations for Future Research

The transit information system has been developed mainly as a stand-alone system, with the exception of assistance from transit agency's database on the status of buses. However, by linking it with other ITS applications and the World Wide Web it can be further enhanced. The next step is to obtain Global Positioning points for the entire Ottawa area. These points can be used to adjust the base map in its entirety for positional accuracy. Additional information can be collected at each bus stop and it can then be used in the creation of a database that will be used to maintain information on the attributes of each stop. GPS would not only provide a tool for operations to improve service, it could also improve relations between the public and the transit system. A GPS-based AVL system would enrich the coverage and accuracy of real-time data on transit vehicles. In turn, the real-time location information would enhance the trip planning process. Users could then receive more accurate bus schedules and route information from the system.

The transit information system can also be linked with real-time bus scheduling, dispatch, and control. That will take incidents such as accidents, traffic congestion, and special events into consideration to make trip planning more efficient and more effective. The system can be further linked with paratransit scheduling and the advanced highway traveller information system to become a multi-modal traveller information system.

The data for a project like this needs to be standardized. Since no geographic data for transit systems exist and street networks are generally designed for the automobile mode of transportation, not bus, new data needs to be generated from the existing street network. Although transit is affected by characteristics of the street network, a bus network has unique characteristics, such as designated bus stops and operation times. The data should be designed to be interoperable between GIS software programs and transit GIS usage. Any transportation agency should have the ability to use the data proficiently. The transit sytem data must be capable of being linked to the road network used by passenger vehicles.

Within the system, more travel options can be offered to the user so that travel preferences will better fit the individual. The current system provides the users with only the shortest route determined by the travel time. Possible options can be travelling on buses with bike-racks throughout the trip or riding low-floor buses. In addition, a possible travel criterion could be added to enable the user

to select a desired arrival time. Currently, the system can only accept desired departure time. Additionally, by adding in walk times to and from the origin and destination and drive time to park and ride lots, this would enable the users to better plan their schedule. Again, by offering users more options for setting their travel preferences each query can be personalized to meet the priorities of the specific user.

In a completely accurate system, the topography would also be integrated. This would include, railway lines, lakes, rivers, and streams. The central business district could also be incorporated, by enabling the representation of buildings where no passage is allowed between streets. This could also include no trespass areas corresponding to private property.

A level of complexity transpires when more than one path exists from an origin to a destination. Often the most direct or simplest path is given to the user. This is often misleading in that there may be several routes that are in the vicinity of the passenger's origin, or a variety of routes with different travel speeds at a nearby transit stop. Additionally, better routes may be possible by making one or two transfers with or without walking links.

Although, the application is complete, there are some trivial requirements before implementation. Presently, the system only has a small sample of the existing routes and bus stop locations within the UTA. Therefore to complete the

application, the remaining routes and stops need to be added. These include the school routes, shuttle buses and the O-Train.

In determining the transit information system is operational, it now requires that it be monitored upon implementation. The true test will be user response to assess the effectiveness of improved customer service.

References

Abdelfattah, Ali and Ata M. Khan. *Models for Predicting Bus Delays.* Transportation Research Record 1623, 1998, pp. 8- 15.

Aronoff, Stan. *Geographic Information Systems: A Management Perspective.* Ottawa: WDL Publications, 1995.

Attanucci, John P., Rick Halvorsen. *What GIS Can Do For Transit Planning.* Papers for the Operational And Service Plannings Symposium, Dec. 1993. http://www.fta.dot.gov/library/technology/symops/attanucc.htm

Ayyub, Bilal M., Richard H. McCuen. *Probability, Statistics, and Reliability for Engineers.* New York: CRC Press, 1997.

Ball, William. *The Use of GIS in Public Transportation.* 1996. http://www.azfms.com/DocReviews/Oct96/art16.htm

Bang, Chulho. *Integrated Model to Plan Advanced Public Transportation Systems.* 1998. http://scholar.lib.vt.edu/theses/available/etd-122898-222857/

Birkin, Mark, Graham Clarke, Martin Clarke, Alan Wilson. *Intelligent GIS Location Decisions and Strategic Planning.* New York: Pearson Professional Ltd., 1996.

Brownstone, David. *Discrete Choice Modeling for Transportation.* IATBR Travel Behavior Conference, Australia, July 2000. http://orion.oac.uci.edu/~dbrownst/discret.pdf

BusView http://www.its.washington.edu/busview

Canadian Strategic Highway Research Program (CSHRP) Bayesian Modeling: A User's Guide. 2000. http://www.cshrp.org/products/bayesian/guide/gchone.htm

Carlin, Bradley P., Thomas A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis, 2nd Ed.* New York: Chapman and Hall/CRC, 2000.

Casey, Robert F., Lawrence N. Labell, Leisa Moniz, Jackson W. Royal, Michael Sheehan, Terry Sheehan, Alex Brown, Malinda Foy, Margaret Zirker, Carol L. Schweiger, Buck Marks, Bruce Kaplan, and Doug Parker. *Advanced Public Transportation Systems: The State of the Art Update 2000.* US

Department of Transportation, Federal Transit Administration, December 2000. http://www.itsdocs.fhwa.dot.gov//JPODOCS/REPTS_TE//13583.pdf

Clarke, Keith C. *Getting Started with Geographic Information Systems.* New Jersey: Prentice Hall, 1997.

Conditions and Performance Report http://www.

Crowson, Jane L., David E. Leasure, Robert W. Smith, Frederick P. Worthen. *A GIS for Public Transit*. ESRI International User Conference 1997. http://www.esri.com/library/userconf/proc97/proc97/to400/pap359/p359.htm

Culp, Linda. *Short Range Transit Planning and Marketing Using Desktop Geographic Information Systems*. San Diego Association of Governments, December 1994. www.fta.dot.gov/library/planning/CULP/CULP.htm

Dailey, D. J. *Smart Trek: A Model Deployment Initiative*. Washington State Transportation Commission and US Department of Transportation Federal Highway Administration, May 2001. http://www.its.washington.edu/pubs/smart_trek_report.pdf

Dailey, D. J., S. MacLean, I. Pao. *BusView: An APTS Precursor and a Deployed Applet*. Washington State Transportation Commission and US Department of Transportation Federal Highway Administration, June 2000. http://www.its.washington.edu/pubs/busview_report.pdf

DeMers, Michael. *Fundamentals of Geographic Information Systems*. New York: J. Wiley & Sons Inc., 1999.

Dessouky, Maged M., Randolph W. Hall, Rutvij Shah, Majid Aldaihani. Evaluating the Impact of ITS on Personalized Public Transit. March 2001. www.its.berkeley.edu/publications/UCB/ 2001/PWP/UCB-ITS-PWP-2001-12.pdf

Dijkstra's Algorithm, 1959. http://www.princeton.edu/~alaink/lab8_html/lab8.html

Dueker and Butler. *GIS-T Enterprise Data Model with Suggested Implementation Choices*. The Center for Urban Studies, Portland State University, 1998.

Edwards, John D. *Transportation Planning Handbook*. New Jersey: Prentice Hall, 1992.

ESRI. *Getting to Know ArcView GIS*. Redlands, CA: ESRI Press, 1997.

ESRI. *Avenue Customization and Application Development for ArcView*. Redlands, CA: ESRI Press, 1996.

http://corpweb.semcor.com/gis/solutions/type/route/

http://www.esri.com

http://www.esriau.com.au/ozri/info.asp

http://www.sydneybuses.nsw.gov.au/sta/tp_UI.html

http://www11.myflorida.com/transit/Pages/

Husdal, J. *Road Transportation Management – Vehicle Routing and Tracking*. University of Leicester, UK. 2000. http://www.husdal.com/mscgis/transport.htm

Kalmer, Bibiana, Beckel Marc. *A Magic Bus, Portland's Mass Transit System Benefits from GIS/GPS Technology*. GeoWorld, September, 1999.

Khisty, C. Jotin, B. Kent Lall. *Transportation Engineering An Introduction, 2nd Edition*. New Jersey: Prentice Hall, 1998.

Khon, Harold M. *Factors Affecting Urban Transit Ridership*. Statistics Canada, June 2000. http://www.statcan.ca/cgi-bin/downpub/listpub.cgi?catno=53F0003XIE

Koncz, Nicholas, Joshua Greenfeld, Kyriacos Mouskos. *A Strategy for Solving Static Multiple-Optimal-Path Transit Network Problems*. Journal of Transportation Engineering, American Society of Civil Engineers, Vol. 122, No. 3, May/June, 1996, pp218-225.

Koncz, Nicholas, Joshua Greenfeld. *GIS-Based Transit Information Bolsters Travel Options*. GIS World, July 1995.

Leclair, R. T. *Transit Service In Villages and Rural Areas Report to Transportation and Transit Committee and Council.* City of Ottawa, March 2002. http://www.city.ottawa.on.ca/calendar/ottawa/citycouncil/ttc/2002/03-20/ACS2002-TUP-TRN-0003.htm

Lee, Jong Sung, Jenny A. Baumgartner, and Tschangho John Kim. *Champaign-Urbana Mass Transit Web-based Bus Information System*. 1999 ESRI User Conference Proceedings http://www.esri.com/library/userconf/proc99/proceed/papers/pap494/p494.htm

Marks, James. *How Transit Agencies are Leveraging the Web for Traveller Information*, 2001. http://www.nawgits.com/icdn/transitweb.html

McGrew, J. Chapman, Charles B. Monroe. *An Introduction to Statistical Problem Solving in Geography*. Dubuque, Iowa: Wm. C. Brown Publishers Inc., 1993.

MegaDyne http://www.megadyneinfo.com

Meyer, Michael D., Eric J. Miller. *Urban Transportation Planning 2nd Edition*. Toronto: McGraw-Hill, 2001.

Mouskos, Kyriacis C., Joshua Greenfeld, Louis J. Pignataro. *Toward a Multi-Modal Advanced Traveller Information System*. NJIT Research, Vol. 4, Spring 1996.

Mouskos, Kyriacos C., Joshua Greenfeld. *A GIS-Based Multimodal Advanced Traveller Information System*. Computer-Aided Civil and Infrastructure Engineering Vol. 14, 1999, pp267-279.

MyBus http://www.mybus.org

NextBus http://www.nextbus.com

NovoGroup. *Increasing Popularity of Public Transport Route Search Facility*. Press Release September 2001. http://www.novogroup.com

OC Transpo Comprehensive Review The Way Ahead Becoming The Best of The Best. Ottawa: February, 1999. http://www.octranspo.com/admin/Comprehensive_Review.htm

OC Transpo Statistics. http://www.octranspo.com/admin/

Ormsby, Tim, Jonell Alvi. Extending ArcView GIS. ESRI Press, 1999.

Orski, C. Kenneth. *The Traveller Information Industry Today*, 2000. http://www.nawgits.com/ko_ti_today.html

Peng, Zhong-Ren, Jonathan N. Groff, Kenneth J. Dueker. *An Enterprise GIS Database Design for Agency-Wide Transit Applications*. URISA Journal, Vol. 10, No. 2, 1998.

Peng, Zhong-Ren, Oliver Jan. *Assessing Means of Transit Infomration Delivery for Advanced Public Transportation Systems*. Transportation Research Record 1666, 1999.

Peng, Zhong-Ren, Ruihong Huang. *Design and development of interactive trip planning for web-based transit information systems*. Transportation Research Part C, Vol. 8, pp409-425, 2000.

Peng, Zhong-Ren. *A Methodology for Design of a GIS-Based Automatic Transit Traveller Information System*. Computer, Environment, and Urban Systems, Vol. 21, No. 5, pp359-372, 1997.

Peng, Zhong-Ren. *GIS Data in the Internet: Issues and Solutions*. http://www.bts.gov/gis/reference/abstracts/abs23.html

Raiffa, Howard. *Decision Analysis Introductory Lectures on Choices under Uncertainty*. Addison-Wesley Publishing Company Inc., 1968.

Rao, K. V. Krishna, S. Muralidhar, S. L. Dhingra. *Public Transport Routing and Scheduling Using Genetic Algorithms*, 2000. http://www.winforms.phil.tu-bs.de/caspt/program.htm

Region of Ottawa Carleton. *Transportation Master Plan*. 1997.

Regional Municipality of Ottawa-Carleton. *Transportation in Ottawa-Carleton*. Status Report, December 1994.

Roess, Roger P., William R. McShane, Elena S. Prassas. *Traffic Engineering 2nd Edition*. New Jersey: Prentice Hall, 1998.

Schweiger, Carol L. *Current Use of Geographic Information Systems in Transit Planning*. Transportation Research Record 1349, 1992.

Shaw, Shih-Lung, Phillip Lall Dass. *A GIS Analysis of Geographic Variations in Travel Characteristics*. May 1996. http://www.esri.com/library/userconf/proc96/TO300/PAP295/P295.HTM

Smith, Brian L. *Using Geographic Information Systems and the World Wide Web for Interactive Transit Trip Itinerary Planning*. Journal of Public Transportation, Vol. 3, No. 2, 2000.

Smith, Brian L., Priya K. Durvasula, Michael J. Demetsky, Stephen C. Brich. *Decision Support Software for Real-Time Route Deviation Transit Service*. 1998 User Conference Proceedings. http://www.esri.com/library/userconf/proc98/PROCEED/TO300/PAP272/P272.HTM

Syed, Sharfuddin and Ata M. Khan. *Factor Analysis for the Study of Determinants of Public Transit Ridership*. Journal of Public Transporation, Vol. 3, No. 3, 2000, pp.1-18.

Teleride http://www.teleride.com

Tong, C. O., A. J. Richardson. *A Computer Model for Finding the Time-Dependent Minimum Path in a Transit System with Fixed Schedules*. Journal of Advanced Transportation, Vol. 18, No. 2, pp145-161, 1984.

Transport Canada. *What is ITS?* 2000. http://www.its-sti.gc.ca/en/its_what_is_its.htm

Transportation Research Board (TRB), National Research Council. *AVL Systems for Bus Transit.* National Academy Press, 1997. http://ntl.bts.gov/display.cfm?sub=d2&cat=4

Transportation Research Board (TRB), National Research Council. *TCRP Report 45 Passenger Information Services: A Guidebook for Transit Systems.* Transit Cooperative Research Program, Sponsored by the Federal Transit Administration, 1999. www.nationalacademies.org/trb/publications/tcrp/tcrp_rpt_45.pdf

Transportation Research Board (TRB). *Highway Capacity Manual*, 2000.

U. S. Department of Transportation (USDOT). *Developing Traveller Information Systems Using the National ITS Architecture.* Intelligent Transportation Systems Joint Program Office, August 1998. http://www.itsdocs.fhwa.dot.gov/jpodocs/repts_te/37n01!.pdf

US Department of Transportation (USDOT) and Federal Transit. Administration (FTA). *Traveller Information.* http://www.fta.dot.gov/library/technology/APTS/update/CHAP3.HTM

US Department of Transportation (USDOT). *Automatic Vehicle Location Successful Transit Applications A Cross Cutting Study Improving Service and Safety.* August 2000. http://ntl.bts.gov/data/8v301.pdf

US Department of Transportation (USDOT). *Conditions and Performance Report.* Sponsored by Federal Highway Administration and Federal Transit Administration, 1999. http://www.fhwa.dot.gov/policy/1999cpr/ch-2/cpm02_8.htm

Wong, S. C., C. O. Tong. *Estimation of Time-Dependent Origin-Destination Matrices for Transit Networks.* Transportation Research-B, Vol. 32, No. 1, pp 35-48, 1998.

Appendix A
Variables of Transit Operation

Appendix A: Variables of Transit Operation

A.1 Modal Characteristics and Trends

Throughout the world, the last few decades have seen an upgrading of bus systems and an expansion of rail transit systems. Urban expansion and growth have triggered this need for public transportation development. In the United States and Canada, transit investment reflects expansion of downtown office space; suburban and urban traffic congestion; the desire to provide a viable alternative to the car and freeway; and the realization by many local and federal governments that the automobile cannot exclusively provide commuter transportation (Edwards, 1992).

Bus transportation is the dominant form of public transport in most North American cities. Most bus service operates in mixed traffic over streets and highways. Bus priority lanes are provided on city streets and highways. Transitways have been introduced in some cities including Ottawa.

The speed, capacity, and reliability of buses can be enhanced by a variety of techniques, such as:

- Exclusive or preferential transit lanes on sections of streets and freeways.
- Exclusive transit turns at intersections.

- Passive traffic signal priority measures, such as cycle length adjustments, split phases, and timing plans selectively favouring buses.
- Exclusive transit streets, malls, and ramps.
- Exclusive busways.

A.2 Role of transit

Within the urban environment, the automobile is by far urban transit's greatest competitor having the remainder walk or use bicycles. Around the world, those that have implemented policies that support urban transit have, in some cases, encouraged people to switch from the automobile to forms of public transportation such as buses, streetcars and subways. Prohibitive taxing and pricing policies have the potential to force drivers out of their cars into urban transit vehicles. Nevertheless, with the exception of a handful of major population centers, most communities in Canada are sparsely populated, which makes the provision of urban transit services relatively expensive and difficult to offer on a comprehensive basis (that is, providing all parts of the community with frequent service). Changes in work patterns such as work-at-home, shift work outside the traditional 9 to 5 time frame, the location of work sites away from downtown areas and other similar factors have given transit planners a multitude of challenges.

However transit plays three major roles in North America. First, it accommodates choice riders; those who choose transit for their trip making even though they have other means of travel, such as a passenger vehicle. Many commuters choose transit because they are unwilling to deal with traffic congestion during peak periods for work trips. Choice riders dominate transit during the peak periods for work trips. In this way, transit increases the number of people who can be carried by urban transportation systems and constrains the growth of more passenger vehicles. Accommodating choice riders is especially critical in cities with high central business district densities and costly and limited parking (TRB, 2000).

Transit assists in congestion relief. If transit service consistently provides door-to-door travel times that are competitive with those of private automobile trips, then transit will provide a meaningful substitute for automobiles as the travel mode of choice. In doing so, transit can effectively reduce roadway congestion. This is especially important for commuting trips, which are often made during times of peak period congestion. With the busways and light rail systems riders are provided with a separate system that avoids all automobile traffic and congestion (USDOT, 1999).

The other major role of transit is providing basic mobility for segments of population that are too young, too old, or otherwise unable to drive due to physical, mental, or financial situations. These transit users have been called

captive riders. These users benefit from a transit system that provides regular access to multiple destinations at a low cost (USDOT, 1999, TRB, 2000).

These three roles will obviously overlap with and support each other. For example, a transit vehicle may primarily serve as a congestion relief tool during peak travel periods while supporting basic mobility in off-peak hours. An individual may choose a housing location near a transit station in order to both avoid rush-hour congestion and to access shopping and entertainment activities in the evening.

The cost of providing comprehensive services, especially for communities that are characterized by urban sprawl, has meant a requirement for subsidization. In 1998, governments in Canada paid approximately $2.4 billion in capital and operating subsidies to urban transit companies (Khon, 2000). Nevertheless, transit companies have sought out new sources of revenue such as fees from parking lots and advertising. Revenues from these sources grew from $82.4 million in 1995 to $110.4 million in 1998 (Khon, 2000).

Most would agree that there are many factors that affect both the demand for and supply of urban transit. These include (Khon, 2000):

- Family size – Families with children may choose to use personal vehicles rather than urban transit because the monthly cost of transit passes may be more expensive;
- Economic changes such as employment opportunities, taxes, fuel costs, parking fees, automobile insurance costs, and vehicle operating costs;
- Demographic impacts such as population growth, immigration rates and fertility rates;
- Ridership loyalty;
- Parking rates and distance to work;
- Other factors such as convenience, a change in work schedules and the work-at-home phenomenon;
- Community size – Large cities with long commutes, expensive downtown parking and relatively greater distances to destination points may positively influence the use of public transit because of the level and frequency of service as well as time and cost savings. Travel times in smaller communities may not be as lengthy, although in smaller communities, the availability of urban transit may not be as comprehensive as in larger cities.

Factors that affect supply and demand are complex, constantly changing and difficult to identify and discern. Therefore it is safe to say that many factors individually or combined influence transit ridership.

A.2.1 Transit Characteristics

Several characteristics differentiate transit from the automobile in terms of availability and capability. Although the automobile has widespread access to roadway facilities, transit service is available only in certain locations during certain times. Roadway capacity is available 24 hours a day seven days a week once constructed, but transit capacity is limited by the number of transit vehicles operating at a given time.

Transit passengers frequently rely on other modes to gain access to transit. Transit use is greatest where population densities are highest and pedestrian access is good. A typical transit user does not have transit service available at the door and must walk, bike, or drive to a transit stop and then must walk or bike from the transit discharge point to the destination. In contrast, suburban areas are mainly automobile-oriented, with employment and residents dispersed, often without sidewalks, and without direct access to many transit lines. If potential passengers cannot have access to transit from both their trip origin and destination, transit is not an option.

Finally, transit is about moving people rather than vehicles. Transit operations at their most efficient involve relatively few vehicles,

each potentially carrying a relatively large number of passengers. In contrast, roadway analysis traditionally involves relatively large numbers of vehicles, each usually only one occupant. When evaluating transit priority measures for transit and automobile users, it is the number of people affected that should be compared, rather than the number of vehicles (TRB, 2000).

A.2.2 Classification

Transit users can be classified in two groups "captive" and "choice" riders. Captives are those who have no alternative transportation mode available; they have no access to an automobile either as driver or passengers, nor is walking or bicycling a viable alternative for those trips for which transit is used. The majority of this group depends on transit so frequently that they must choose their home, work location, shopping, and other facilities in relation to well-established transit lines. Choice riders are attracted to transit if they feel that the travel time and cash costs involved are competitive with those of the automobile alternative. This is most likely to occur for peak period trips with one end in the CBD because of the likelihood of congestion at the approaches to the downtown area and the high parking costs within it (Khisty and Lall, 1998).

A.2.3 Types of service

Bus transit can be either fixed route or demand responsive. Fixed-route service is ideal for large, densely populated urban areas. In less dense areas, which cannot support fixed-route service, demand responsive transit can be an essential part of transportation of the non-driving population (TRB, 2000). In Ottawa, OC Transpo operates as a fixed route system.

Local transit networks operate in mixed traffic (shared right-of-way) on suitable streets and freeways. For environmental reasons, transit routes are generally confined to major arterials and collector streets. However to serve outlying neighbourhoods adequately, some residential streets may have to be used. Where applicable busways are built to accommodate bus traffic and more efficient service (khisty and Lall, 1998).

There is no precise definition of what differences in right-of-way, technology, or service constitute a specific transit type. Commonly, transit is classified into three generic classes, based mostly on right-of-way (Region of Ottawa-Carleton, 1997):

- Street Transit—(surface transit) designates transit operated on roadways with mixed traffic. Its reliability is often low because

of various interferences, and its speed is lower than the speed of traffic due to the time spent at passenger stops. Buses, trolley buses and streetcars fall into this category.

- Semi-Rapid Transit uses mostly longitudinally separated right of way although other classes of right of way may be used. Some LRT and bus-based systems fall into this category.

- Rapid Transit operates exclusively on fully separated right-of-way. They generally feature high speed and capacity. Busways such as the OC Transpo's Transitway fall into the category of rapid transit. Busways are controlled access facilities dedicated for bus services separated from general traffic, often with grade separated right-of-way. Buses, therefore, operate under conditions achieving speeds equal to or in excess of light rail systems. The busway combines the flexibility of a bus, which can go anywhere there is an adequately paved street, with freedom from general traffic conditions (Region of Ottawa-Carleton, 1997).

A.2.4 Quality of service

Whether or not transit is provided near a person's origin and destination is key in use of transit. Ideally, transit service is

provided within a reasonable walking distance of the origin and destination, or demand-responsive service is available. The reasonableness of the walking distance varies from source to source and depends on the situation (TRB, 2000).

If transit service is not provided near the origin, other options include driving to a park-and–ride lot or riding a bike to transit. Both of these options require that the transit operator provide additional facilities, such as parking lots, bicycle storage and racks.

However, if transit service is not provided near the destination, the choices are more limited. A bicycle might be carried on a rack, but a customer must have some degree of confidence that space will be available in the bike rack when the bus arrives.

How often and when a transit service is provided are important factors in the decision to use transit. The more frequent the service, the shorter the wait when a bus or train is missed or when the exact schedule is not known, and the greater the flexibility customers will have in selecting travel times. The number of hours during the day when service is provided is also important. It does not matter whether a transit stop is located within walking distance

if service is not provided at the desired time of travel; transit then cannot be an option (TRB, 2000).

A.2.5 Reliability

Reliability affects the amount of time passengers must wait at a transit stop, as well as the consistency of a passenger's arrival time at a destination from day to day. Reliability encompasses on-time performance as well as the regularity of headways between successive transit vehicles. Uneven headways result in uneven passenger loadings, so that a transit vehicle arriving late picks up not only its regular passengers but also others who have arrived early for the following vehicle. As a result, the vehicle falls further and further behind schedule. In contrast, the vehicles following will have lighter-than-normal passenger loads and will tend to run ahead of schedule. Reliability is influenced by transit conditions, staff availability and vehicle maintenance, and by how well vehicle operators adhere to schedules (TRB, 2000).

A.2.6 Total Trip Time

Total trip time includes the travel time from the origin to a transit stop, waiting time for a transit vehicle, travel time onboard a vehicle, travel time from transit to the destination, and any required transfers between routes during the trip. In general, both the

absolute travel time and the travel time in relation to competing modes will factor in a traveller's decision about transit (TRB, 2000). However, the apparent inconvenience of a longer transit trip can be diminished if the passenger can use the time onboard productively (i.e. reading, preparing or reviewing work, or even catching up on sleep).

Travel speed is a useful route segment performance measure because it reflects how long a trip may take without depending on how long a route segment might be. Transit priority measures, improvements to fare collection procedures, the introduction of low-floor buses and other similar actions implemented along a route segment will be reflected as improvements in travel speed (TRB, 2000).

Total trip time is influenced by several factors, including the route spacing, the service frequency, the frequency of stops, traffic congestion, signal timing, and the fare collection system (TRB, 2000). Regardless of the kind of bus facility being analyzed, there are some fundamental components common to each that are required to calculate the facility's vehicle and person capacity. Dwell time is the most important of these, but all have some effect on capacity.

The average speed along a route is a function of stop spacing and either the top speed capacity of the vehicle or train or the speed limit imposed by law or physical conditions. Most choice riders seek the shortest total travel time door to door, and that maximizing the speed of transit routes at the cost of accessibility is not necessarily advantageous. However, if accessibility remains about the same, increases in speed are beneficial to users (Khisty and Lall, 1998).

A.2.7 Capacity

Transit capacity is different from highway capacity. It deals with the movement of both people and vehicles; it depends on the size of the transit vehicles and how often they operate; and it reflects the interaction of passenger traffic and vehicles flow. Transit capacity depends on the operating policy of the transit agency, which specifies service frequencies and allowable passenger loadings. Accordingly, the traditional concepts applied to highway capacity must be adapted and broadened (TRB, 2000).

A.2.8 Dwell time

Just as dwell times are key to determining capacity, passenger demand volumes and passenger service times are key to

determining dwell times. Dwell times may be governed by boarding demand, alighting demand, or total interchanging passenger demand. In all cases, dwell time is proportionate to the boarding and alighting volumes times the service time per passenger. Dwell time also can influence a transit operator's service costs; if average vehicle speeds can increase by reducing dwell time, and it the cumulative change exceeds the route headway, then fewer vehicles may be required to provide the same service frequency (TRB, 2000).

As defined by the Highway Capacity Manual, there are six main influences on dwell time. Two relate to passenger demands; the others relate to passenger service time:

- Passenger demand and loading.
- Stop and station spacing.
- Fare payment procedures.
- Vehicles types.
- On-board circulation.
- Wheelchair and bicycle boarding.

Dwell time can be summed as the time required to serve passengers at the busiest door plus the time required to open and

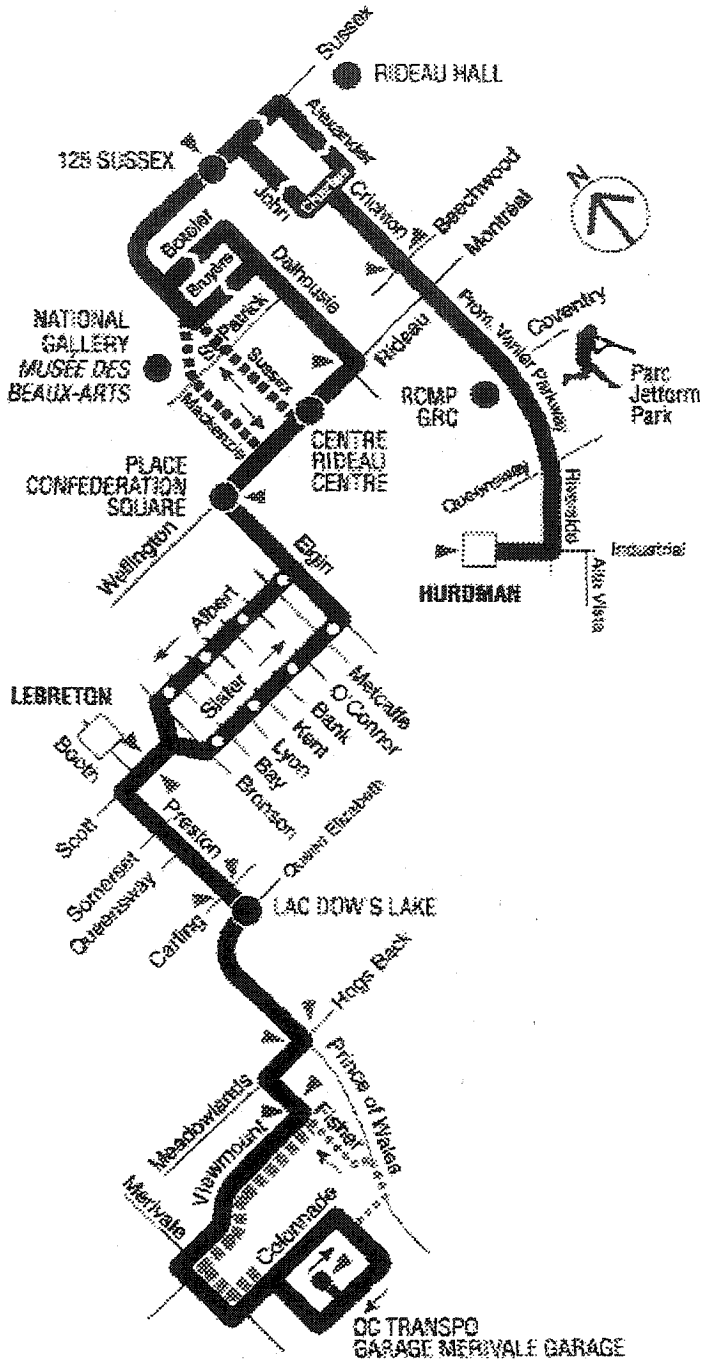
close the doors. A value of 2 to 5 s for door opening and closing is reasonable for normal operations (TRB, 2000).

Dwell time can be measured in the field. Field measurement of dwell time is best suited for determining the capacity and LOS of an existing transit line. In the absence of other information, dwell time can be assumed to be 60s for CBD, or transit stations; 30s for major outlying stops; and 15s for typical outlying stops (TRB, 2000).

A.2.9 Clearance Time

Clearance time includes two components, the time for a bus to start up and travel its own length while exiting a bus stop and the reentry delay associated with the wait time for a sufficient gap in traffic to allow a bus to pull back into the travel lane. Various studies have looked at these factors, either singly or together. Research has found that bus start-up times range from 2 to 5s. The time for a bus to travel its own length after stopping is approximately 5 to 10s, depending on acceleration and traffic conditions. Other research recommends a range of 10 to 15s for clearance time (TRB, 2000).

Appendix B
Transit Routes



3

NEPEAN CENTRE

HURDMAN

OC Transpo

Legend • *Légende*

Transitway Station / *Station du Transitway*

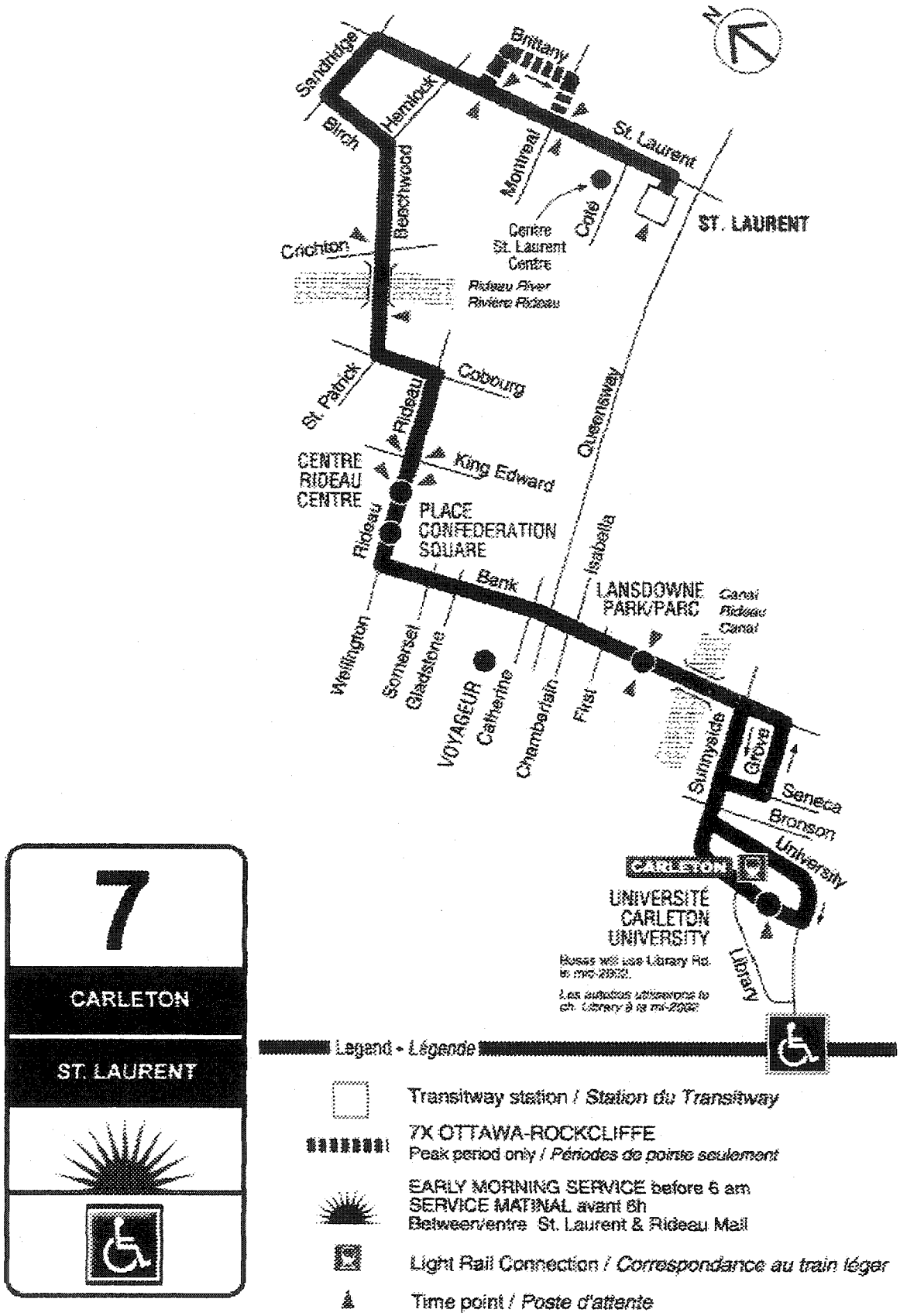
Some Trips (3X) / *Quelques trajets (3X)*

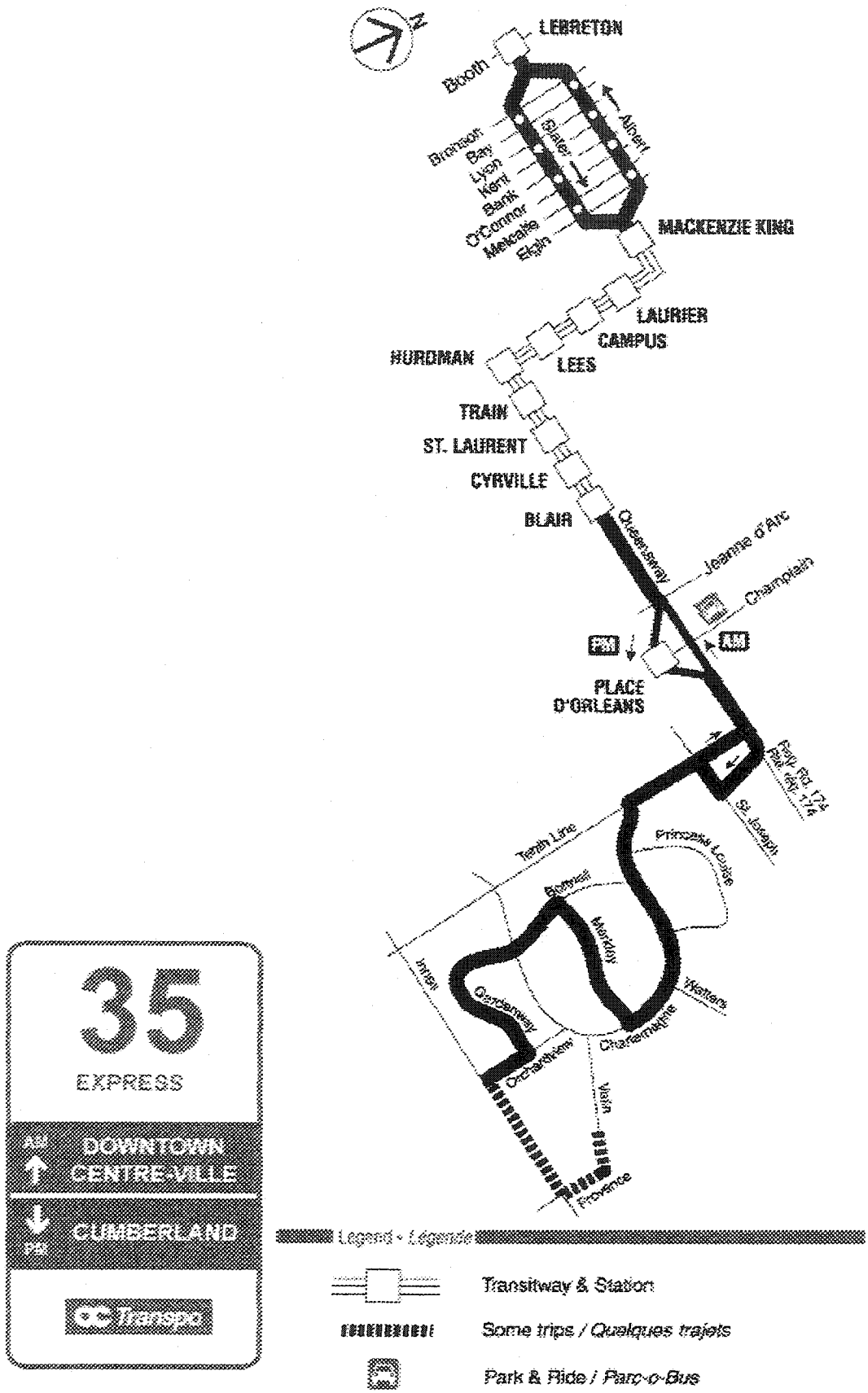
EARLY MORNING SERVICE before 6 am
SERVICE MATINAL avant 6h

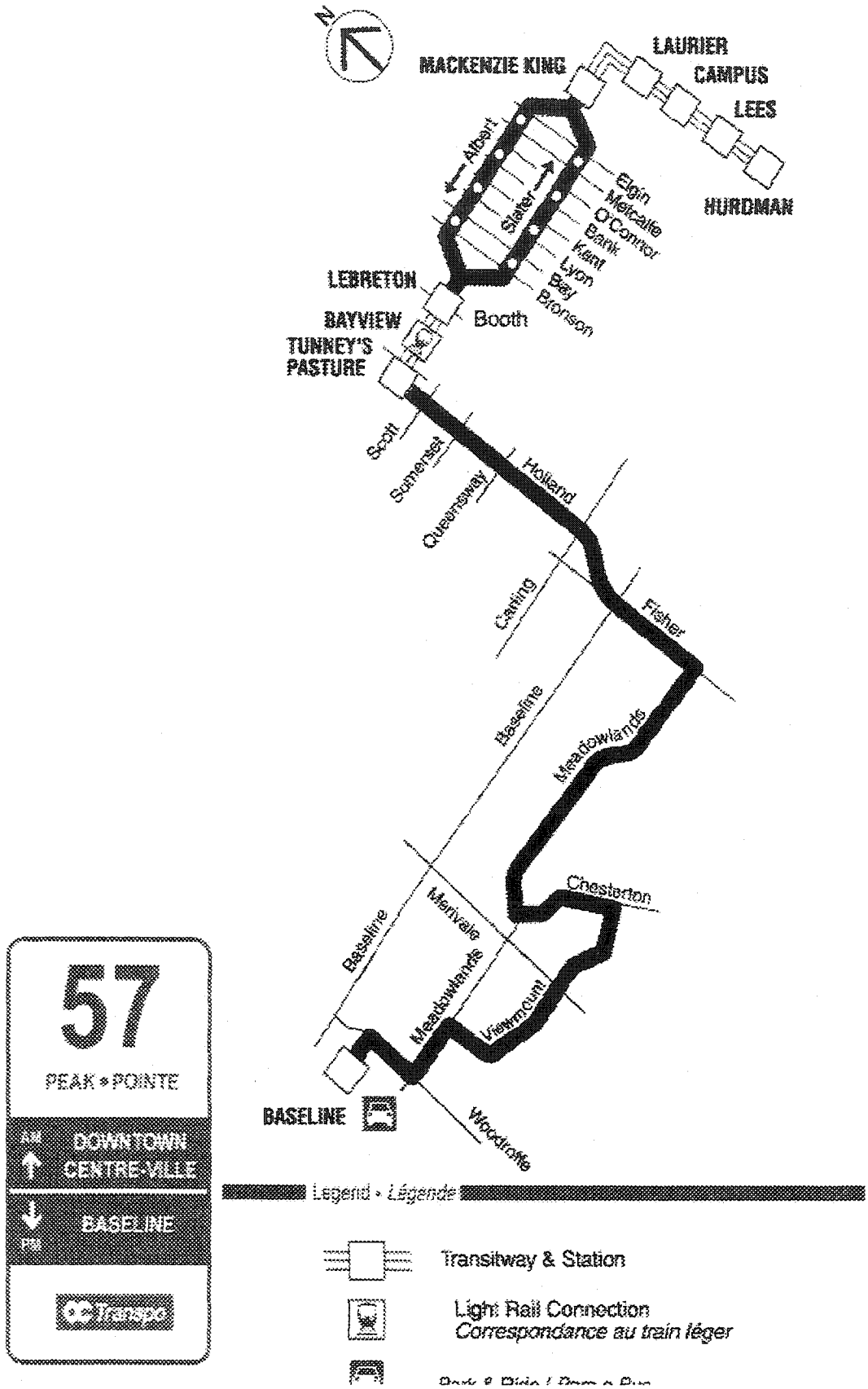
Early morning service only / *Service matinal seulement*

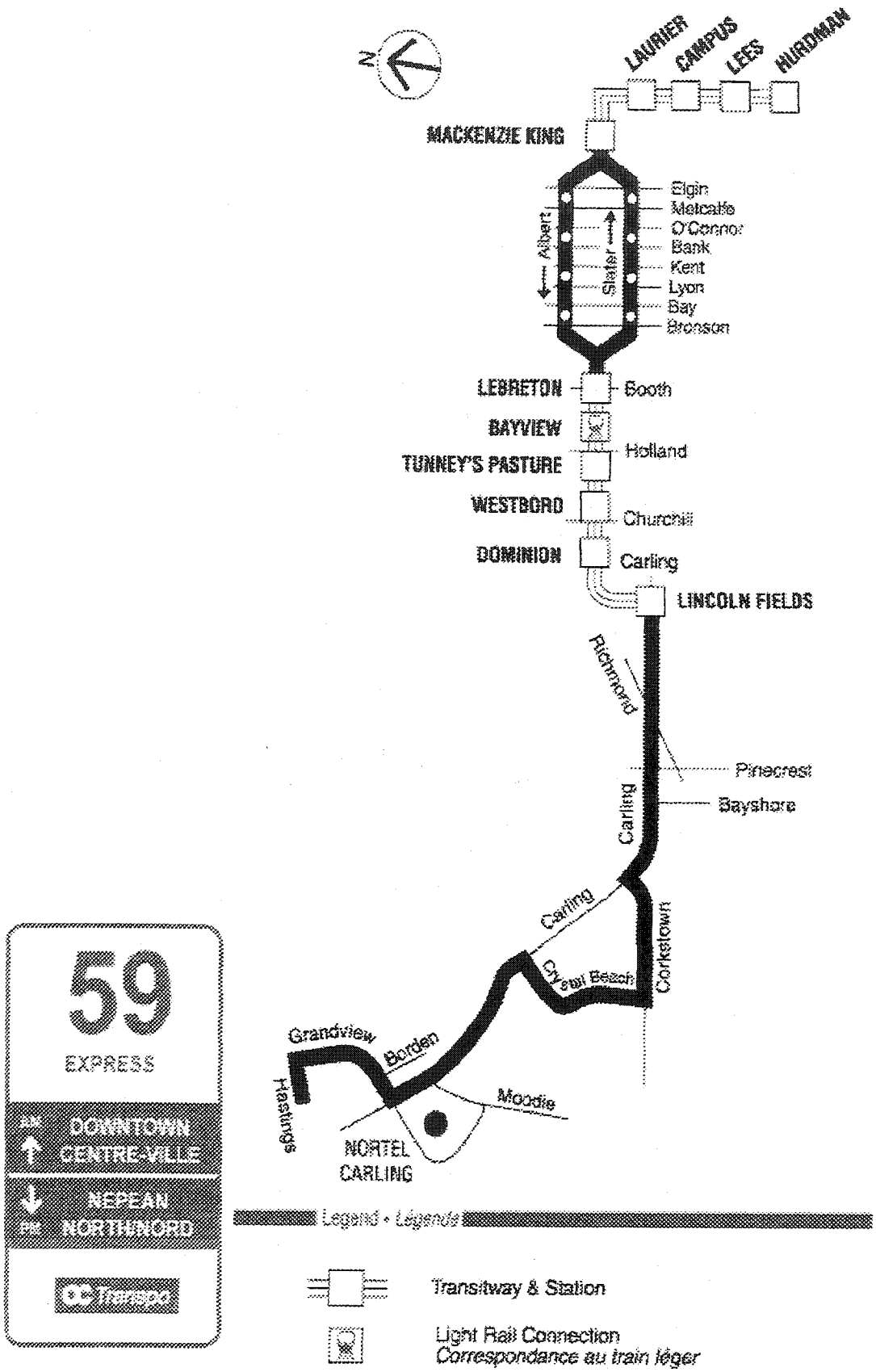
No early morning service / *Aucun service matinal*

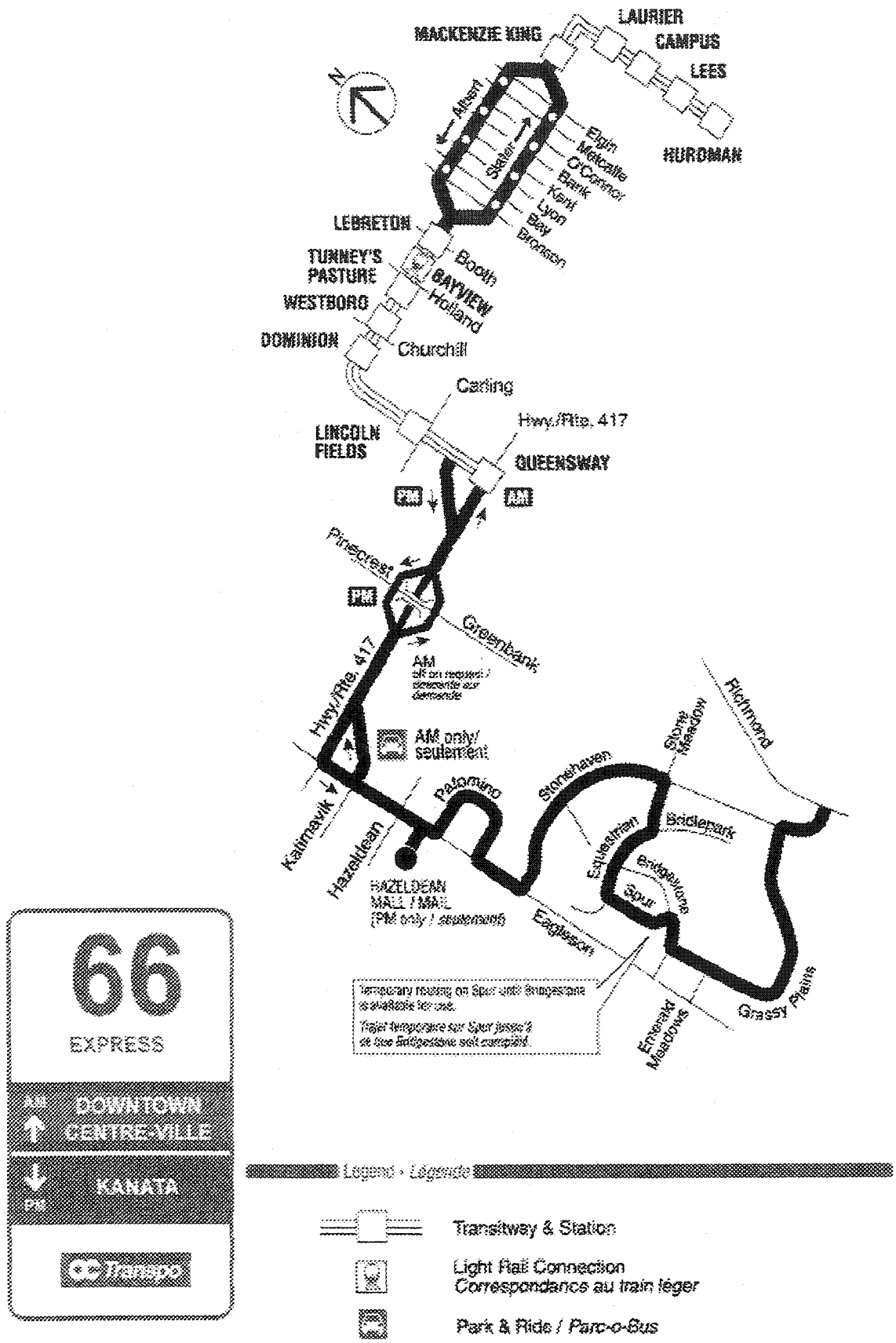
Time point / *Poste d'attente*

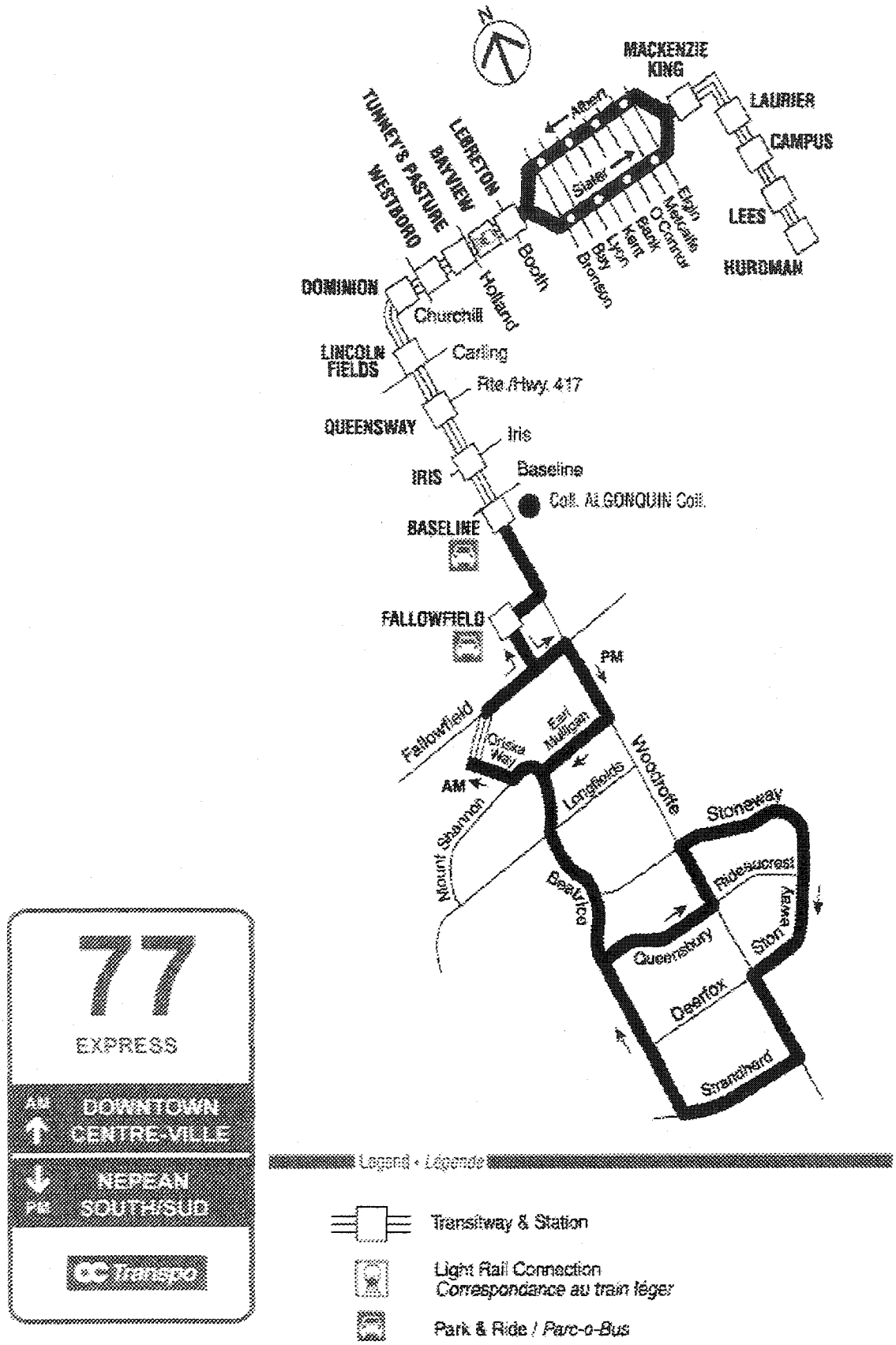

7
CARLETON
ST. LAURENT
Sandridge
Birch
Hemlock
Beechwood
Brittany
St. Laurent
Montreal
Cité
Centre St. Laurent Centre
ST. LAURENT
Crichton
Rideau River
Rivière Rideau
St. Patrick
Cobourg
Rideau
Queensway
CENTRE RIDEAU CENTRE
King Edward
PLACE CONFEDERATION SQUARE
Rideau
Bank
Isabella
LANSDOWNE PARK/PARC
Canal Rideau Canal
Wellington
Somerset
Gladstone
VOYAGEUR
Catherine
Chamberlain
First
Sunnyside
Grove
Saneca
Bronson
University
CARLETON
UNIVERSITÉ CARLETON UNIVERSITY
Library
Legend • Légende
Transitway station / Station du Transitway
7X OTTAWA-ROCKCLIFFE
Peak period only / Périodes de pointe seulement
EARLY MORNING SERVICE before 6 am
SERVICE MATINAL avant 6h
Between/entre St. Laurent & Rideau Mall
Light Rail Connection / Correspondance au train léger
Time point / Poste d'attente

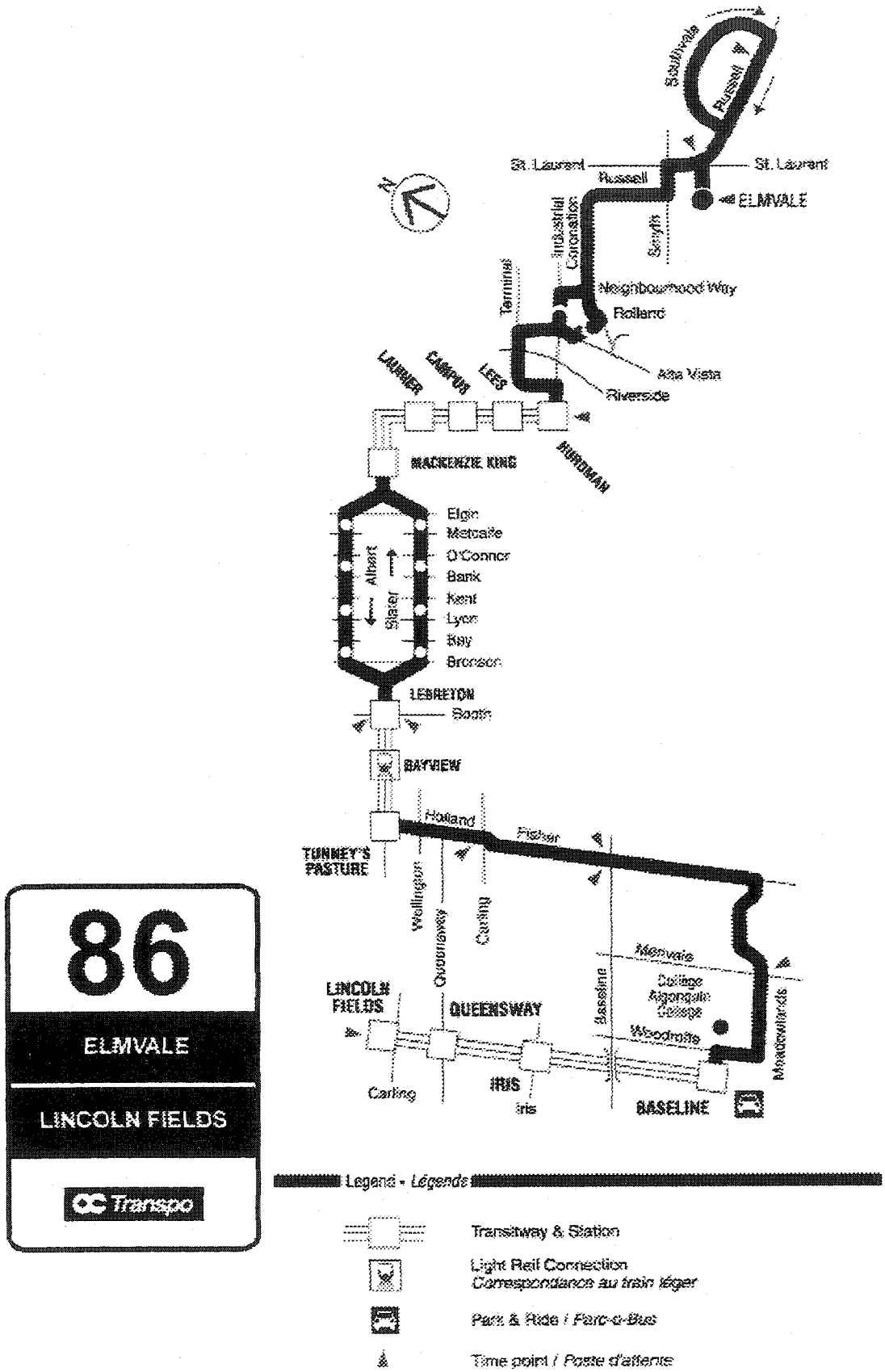

N
LEBRETON
Booth
Bronson
Bay
Lyon
Kent
Bank
O'Connor
Metcalfe
Elgin
Bluter
Albert
MACKENZIE KING
LAURIER
CAMPUS
LEES
HURDMAN
TRAIN
ST. LAURENT
CYRVILLE
BLAIR
Queensway
Jeanne d'Arc
Champlain
PM
AM
PLACE D'ORLEANS
Tenth Line
Bathurst
Markham
Innisil
Queensway
Orchardview
Charlemagne
Watkins
Mainprise Lodge
St. Joseph
Mainprise
Province
35
EXPRESS
AM
DOWNTOWN
CENTRE-VILLE
PM
CUMBERLAND
Transpo
Legend - Légende
Transitway & Station
Some trips / Quelques trajets
Park & Ride / Parc-o-Bus

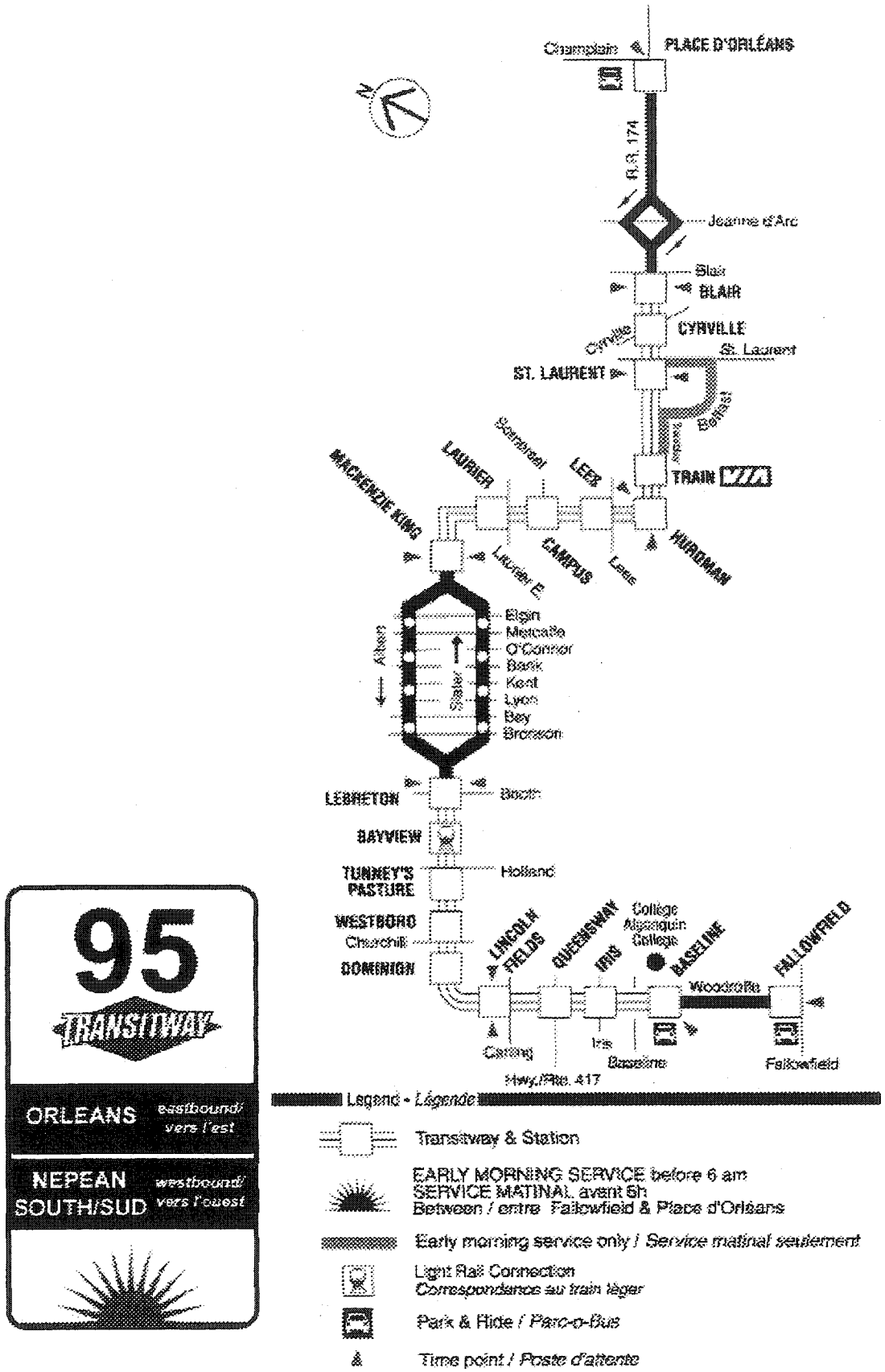

N
MACKENZIE KING
LAURIER
CAMPUS
LEES
HURDMAN
Albert
Slater
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
LEBRETON
BAYVIEW
Booth
TUNNEY'S PASTURE
Scott
Somerset
Queensway
Holland
Carling
Fisher
Baseline
Meadowlands
Chesterton
Merivale
Baseline
Meadowlands
Viewmount
BASELINE
Woodroffe
57
PEAK • POINTE
AM
DOWNTOWN
CENTRE-VILLE
PM
BASELINE
Transpo
Legend • Légende
Transitway & Station
Light Rail Connection
Correspondance au train léger

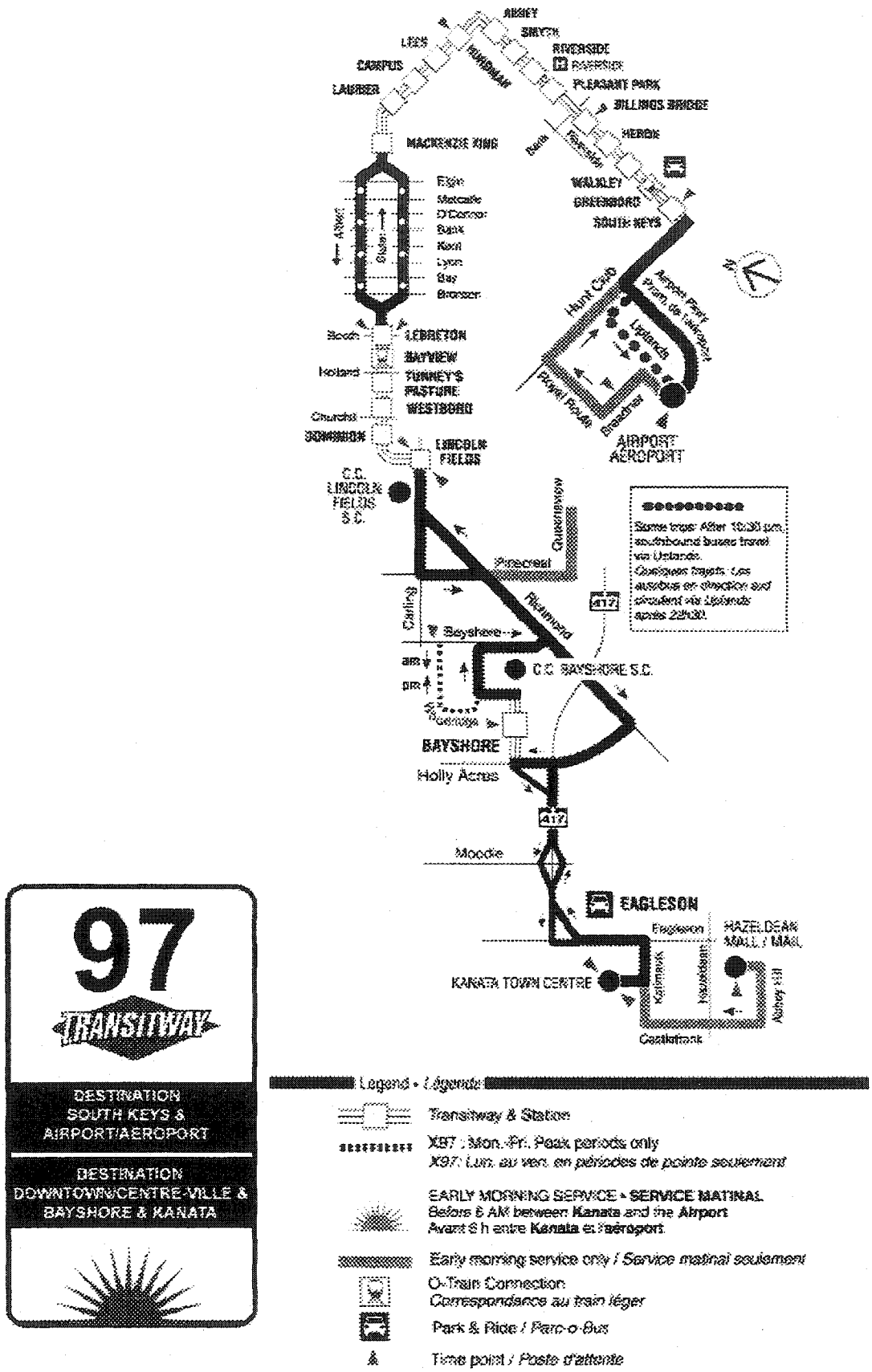

N
LAURIER
CAMPUS
LEES
HURDMAN
MACKENZIE KING
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
Albert
Slater
LEBRETON
Booth
BAYVIEW
TUNNEY'S PASTURE
Holland
WESTBORO
Churchill
DOMINION
Carling
LINCOLN FIELDS
Richmond
Pinecrest
Bayshore
Carling
Carling
Corkstown
Cyrwall Beach
Grandview
Borden
Hastings
Moodie
NORTEL
CARLING
59
EXPRESS
AM
DOWNTOWN
CENTRE-VILLE
PM
NEPEAN
NORTHWOOD
Transpo
Legend • Légende
Transitway & Station
Light Rail Connection
Correspondance au train léger

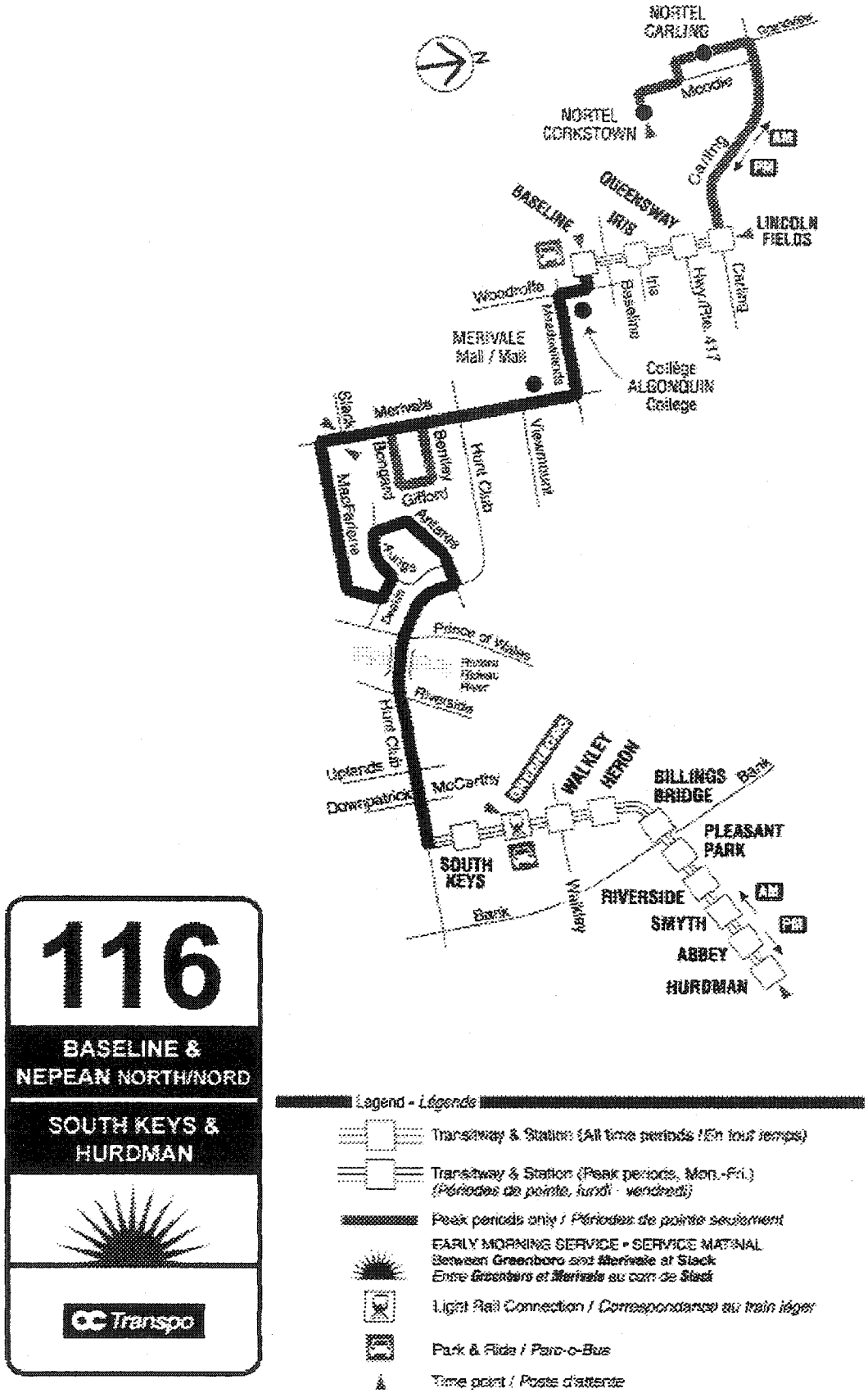

MACKENZIE KING
LAURIER
CAMPUS
LEES
HURDMAN
N
Albert
Slater
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
LEBRETON
TUNNEY'S PASTURE
Booth
BAYVIEW
Holland
WESTBORO
DOMINION
Churchill
Carling
Hwy./Rte. 417
LINCOLN FIELDS
QUEENSWAY
PM
AM
Pinecrest
PM
Greenbank
AM
Hwy./Rte. 417
AM only/ seulement
Kanatavik
Hazeldean
Paterno
Stonehaven
Stone Meadow
Richmond
Equestrian
Bridlepark
Bridlewood
Spur
HAZELDEAN MALL / MALL (PM only / seulement)
Eagleson
Emerald Meadows
Grassy Plains
Temporary routing on Spur until Bridgestone is available for use.
Trajet temporaire sur Spur jusqu'à ce que Bridgestone soit complété.
66
EXPRESS
AM
DOWNTOWN
CENTRE-VILLE
PM
KANATA
Transpo
Legend • Légende
Transitway & Station
Light Rail Connection
Correspondance au train léger
Park & Ride / Parc-o-Bus

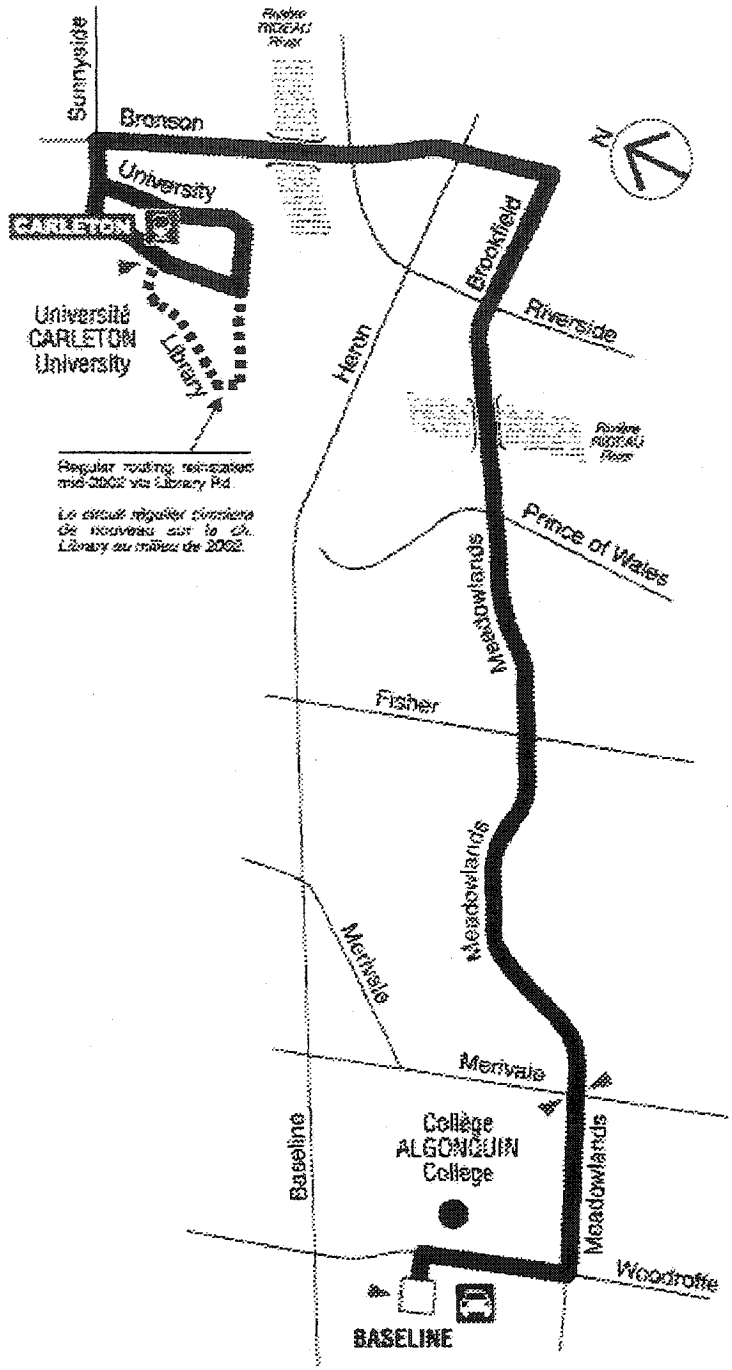

N
MACKENZIE KING
LAURIER
CAMPUS
LEES
HURDMAN
Albert
Slater
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
Booth
Holland
TUNNEY'S PASTURE
WESTBORO
BAYVIEW
LEBRETON
DOMINION
Churchill
LINCOLN FIELDS
Carling
Rte./Hwy. 417
QUEENSWAY
Iris
IRIS
Baseline
Coll. ALGONQUIN Coll.
BASELINE
FALLOWFIELD
PM
Fallowfield
Orsise Way
Earl Mullion
Woodroffe
AM
Longfields
Mount Shannon
Stoneway
Beatrice
Rideaucrest
Queensbury
Stoneway
Deerfox
Strandherd
77
EXPRESS
AM
DOWNTOWN
CENTRE-VILLE
PM
NEPEAN
SOUTH/SUD
Transfers
Legend • Légende
Transitway & Station
Light Rail Connection
Correspondance au train léger
Park & Ride / Parc-o-Bus


86
ELMVALE
LINCOLN FIELDS
Transpo
Southvale
Russell
St. Laurent
St. Laurent
Russell
ELMVALE
Industrial
Coronation
Smyth
Terminal
Neighbourhood Way
Rollerd
Alta Vista
Riverside
LAURIER
CAMPUS
LEES
HURTEBAISE
MACKENZIE KING
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
Albert
Slater
LEBRETON
Booth
BAYVIEW
Holland
Fisher
TUNNEY'S PASTURE
Wellington
Queensway
Carling
Merivale
College Algonquin College
Woodroffe
Baseline
Meadowlands
LINCOLN FIELDS
QUEENSWAY
Carling
IRIS
Iris
BASELINE
Legend - Légende
Transitway & Station
Light Rail Connection
Correspondance au train léger
Park & Ride / Parc-o-Bus
Time point / Poste d'attente


95
TRANSITWAY
ORLEANS eastbound/ vers l'est
NEPEAN SOUTH/SUD westbound/ vers l'ouest
PLACE D'ORLÉANS
Champlain
R.R. 174
Jeanne d'Arc
Blair
BLAIR
CYRVILLE
Cyrville
St. Laurent
ST. LAURENT
Bathurst
TRAIN
Somerset
LAURIER
LEES
MACKENZIE KING
CAMPUS
HURDMAN
Laurier E.
Lees
Albert
Slater
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
LEBRETON
Booth
BAYVIEW
TUNNEY'S PASTURE
Holland
WESTBORO
Churchill
DOMINION
LINCOLN FIELDS
QUEENSWAY
IRIS
Collège Algonquin College
BASELINE
FALLOWFIELD
Woodroffe
Carling
Iris
Baseline
Fallowfield
Hwy./Rte. 417
Legend - Légende
Transitway & Station
EARLY MORNING SERVICE before 6 am
SERVICE MATINAL avant 6h
Between / entre Fallowfield & Place d'Orléans
Early morning service only / Service matinal seulement
Light Rail Connection
Correspondance au train léger
Park & Ride / Parc-o-Bus
Time point / Poste d'attente


97
TRANSITWAY
DESTINATION
SOUTH KEYS &
AIRPORT/AEROPORT
DESTINATION
DOWNTOWN/CENTRE-VILLE &
BAYSHORE & KANATA
ABBEY
SMYTH
LEES
RIVERSIDE
CARPUS
LAURIER
PLEASANT PARK
BILLINGS BRIDGE
MACKENZIE KING
HEROE
WALKLEY
GREENBORO
SOUTH KEYS
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
Hunt Club
Airport Pkwy
Uplands
Royal Roads
Broadway
AIRPORT
AEROPORT
LEBRETON
BAYVIEW
TUNNEY'S
PASTURE
WESTBORO
Holland
Churchill
LINCOLN
FIELDS
C.C.
LINCOLN
FIELDS S.C.
Queensview
Pinecrest
Richmond
Carling
417
Bayshore
C.C. BAYSHORE S.C.
BAYSHORE
Holly Acres
Moodie
EAGLESON
Eagleson
HAZELDEAN
MALL / MALL
KANATA TOWN CENTRE
Castlefrank
Legend • Légende
Transitway & Station
X97 : Mon.-Fri. Peak periods only
X97 : Lun. au ven. en périodes de pointe seulement
EARLY MORNING SERVICE • SERVICE MATINAL
Before 6 AM between Kanata and the Airport
Avant 6 h entre Kanata et l'aéroport
Early morning service only / Service matinal seulement
O-Train Connection
Correspondance au train léger
Park & Ride / Parc-o-Bus
Time point / Poste d'attente


116
BASELINE &
NEPEAN NORTH/NORD
SOUTH KEYS &
HURDMAN
Transpo
NORTEL CARLING
NORTEL CORKSTOWN
Carling
Moodie
BASELINE
QUEENSWAY
IRIS
LINCOLN FIELDS
Woodroffe
Iris
Baseline
Hwy/Rte. 417
Carling
MERIVALE Mall / Mall
Merivale
Mississippi
Collège ALGONQUIN College
Stack
Merivale
Bentley
Richmond
Gifford
Hunt Club
Viewmount
MacFarlane
Arlington
Prince of Wales
Riverside
Hunt Club
Uplands
Donnybrook
McCarthy
WALKLEY
HERON
BILLINGS BRIDGE
Bank
PLEASANT PARK
SOUTH KEYS
Walkley
RIVERSIDE
SMYTH
ABBEY
HURDMAN
Bank
Legend - Légende
Transitway & Station (All time periods / En tout temps)
Transitway & Station (Peak periods, Mon.-Fri.)
(Périodes de pointe, lundi - vendredi)
Peak periods only / Périodes de pointe seulement
EARLY MORNING SERVICE - SERVICE MATINAL
Between Greenboro and Merivale at Stack
Entre Greenboro et Merivale au coin de Stack
Light Rail Connection / Correspondance au train léger
Park & Ride / Parc-o-Bus
Time point / Poste d'attente



117

CARLETON

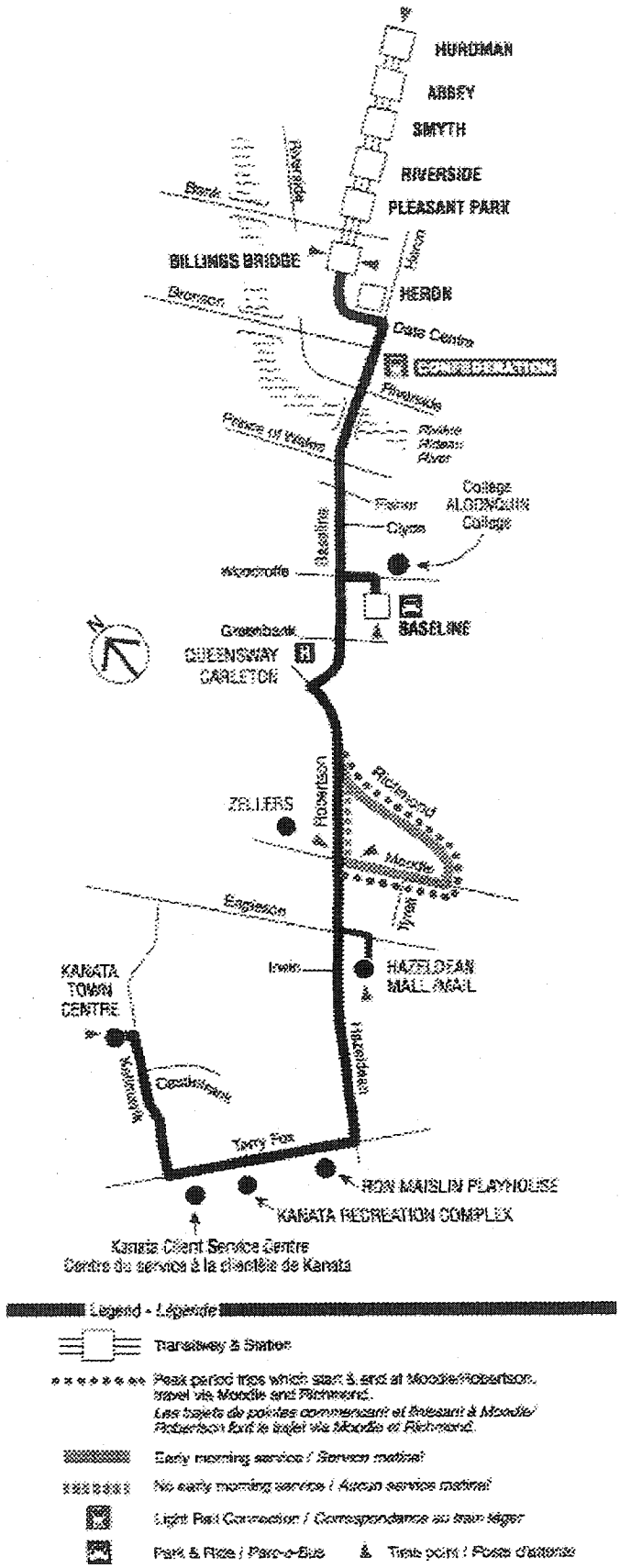
BASELINE

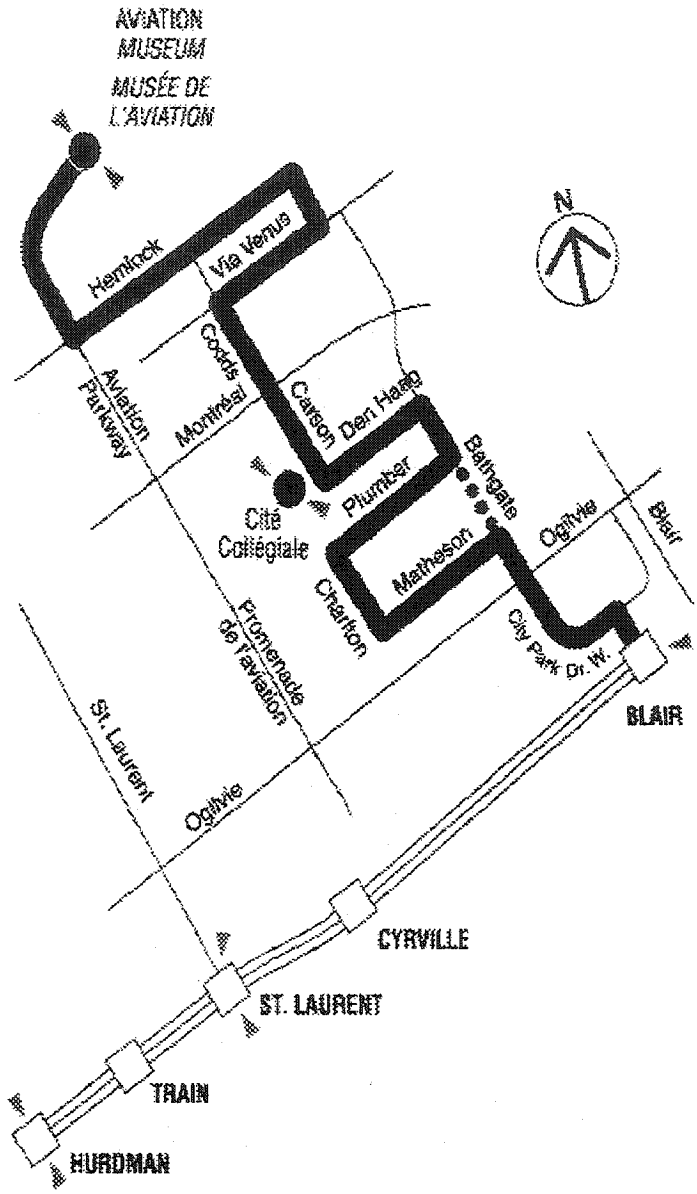
Transpo

Legend • *Légende*

- Transitway Station / *Station du transitway*
- Park & Ride / *Parc-o-Bus*
- Light Rail Connection / *Correspondance au train léger*
- Time point / *Poste d'attente*


118
HURDMAN
KANATA &
NEPEAN CENTRE
Transpo


HURDMAN
ABBEY
SMYTH
RIVERSIDE
PLEASANT PARK
BILLINGS BRIDGE
HERON
Data Centre
Bronson
Baseline
BASELINE
QUEENSWAY CARLETON
Greenbank
Richardson
Richmond
Moodie
ZELLERS
Eagleson
KANATA TOWN CENTRE
HAZELDEAN MALL/MAIL
Hazeldean
Terry Fox
RON MAISLIN PLAYHOUSE
KANATA RECREATION COMPLEX
Kanata Client Service Centre
Centre de service à la clientèle de Kanata
Legend - Légende
Transitway & Station
Peak period trips which start & end at Moodie/Robertson. Travel via Moodie and Richmond.
Les trajets de pointes commençant et finissant à Moodie/Robertson font le trajet via Moodie et Richmond.
Early morning service / Service matinal
No early morning service / Aucun service matinal
Light Rail Connection / Correspondance au train léger
Park & Ride / Parc-o-Bus
Time point / Poste d'attente



129

OTTAWA-ROCKCLIFFE

HURDMAN

OC Transpo

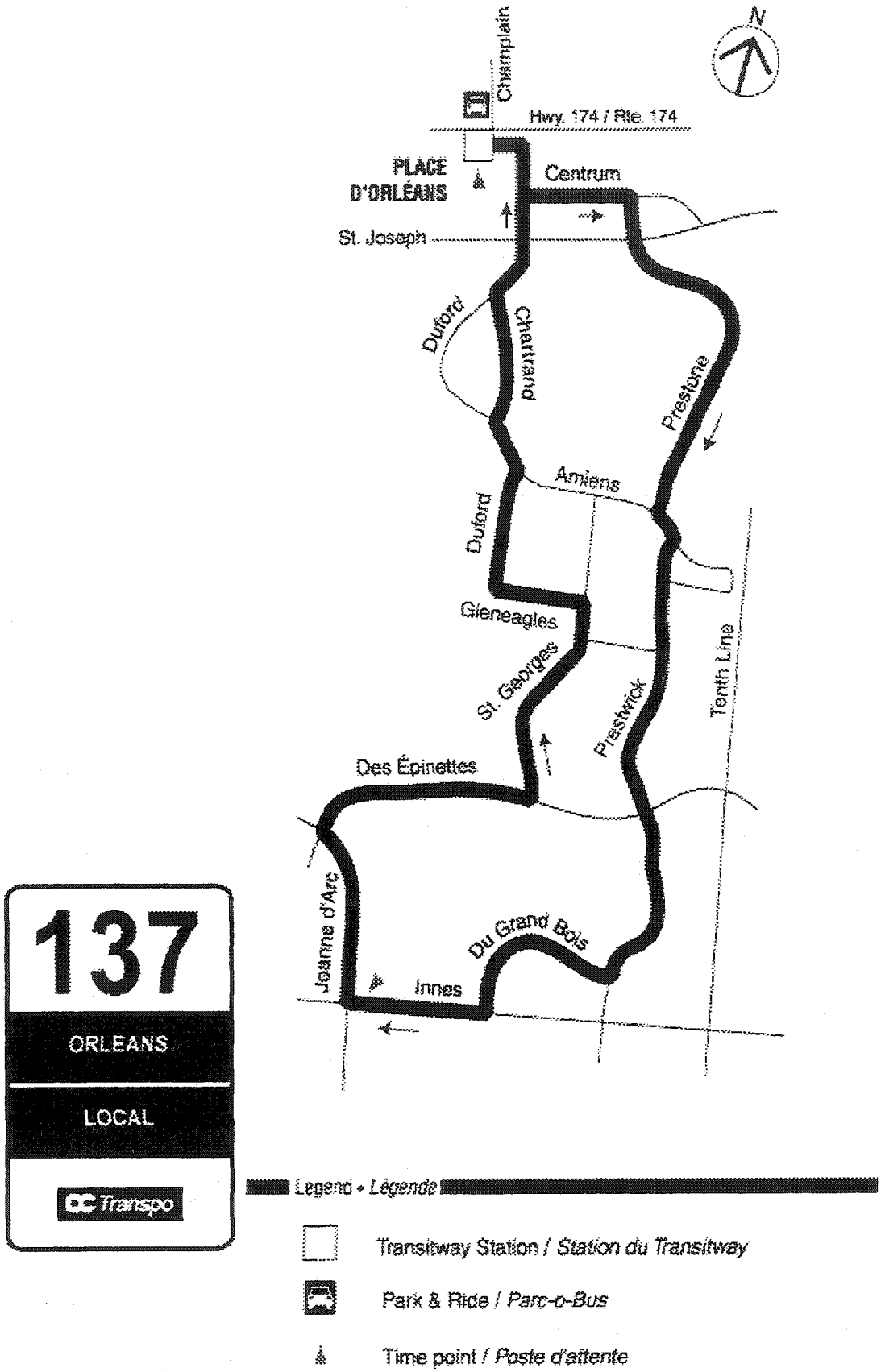
No Sunday service between Blair and Hurdman.
Aucun service le dimanche entre Blair et Hurdman.

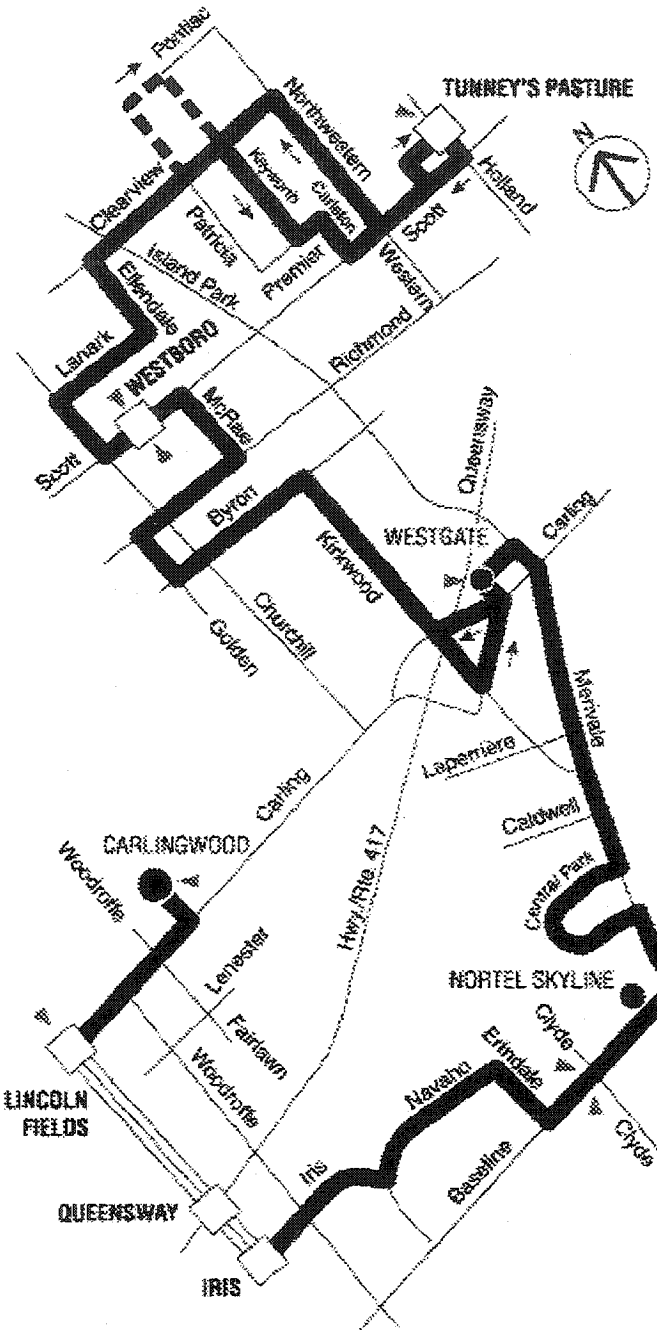
Legend • *Légende*

Transitway & Station

129 SPECIAL

Time point / *Poste d'attente*


N
Champlain
Hwy. 174 / Rte. 174
PLACE D'ORLÉANS
Centrum
St. Joseph
Dufford
Chartrand
Prestone
Amiens
Dufford
Gieneagles
St. Georges
Prestwick
Tenth Line
Des Épinettes
Jeanne d'Arc
Du Grand Bois
Innes
137
ORLEANS
LOCAL
OC Transpo
Legend • Légende
Transitway Station / Station du Transitway
Park & Ride / Parc-o-Bus
Time point / Poste d'attente



151

TUNNEY'S PASTURE

CARLINGWOOD

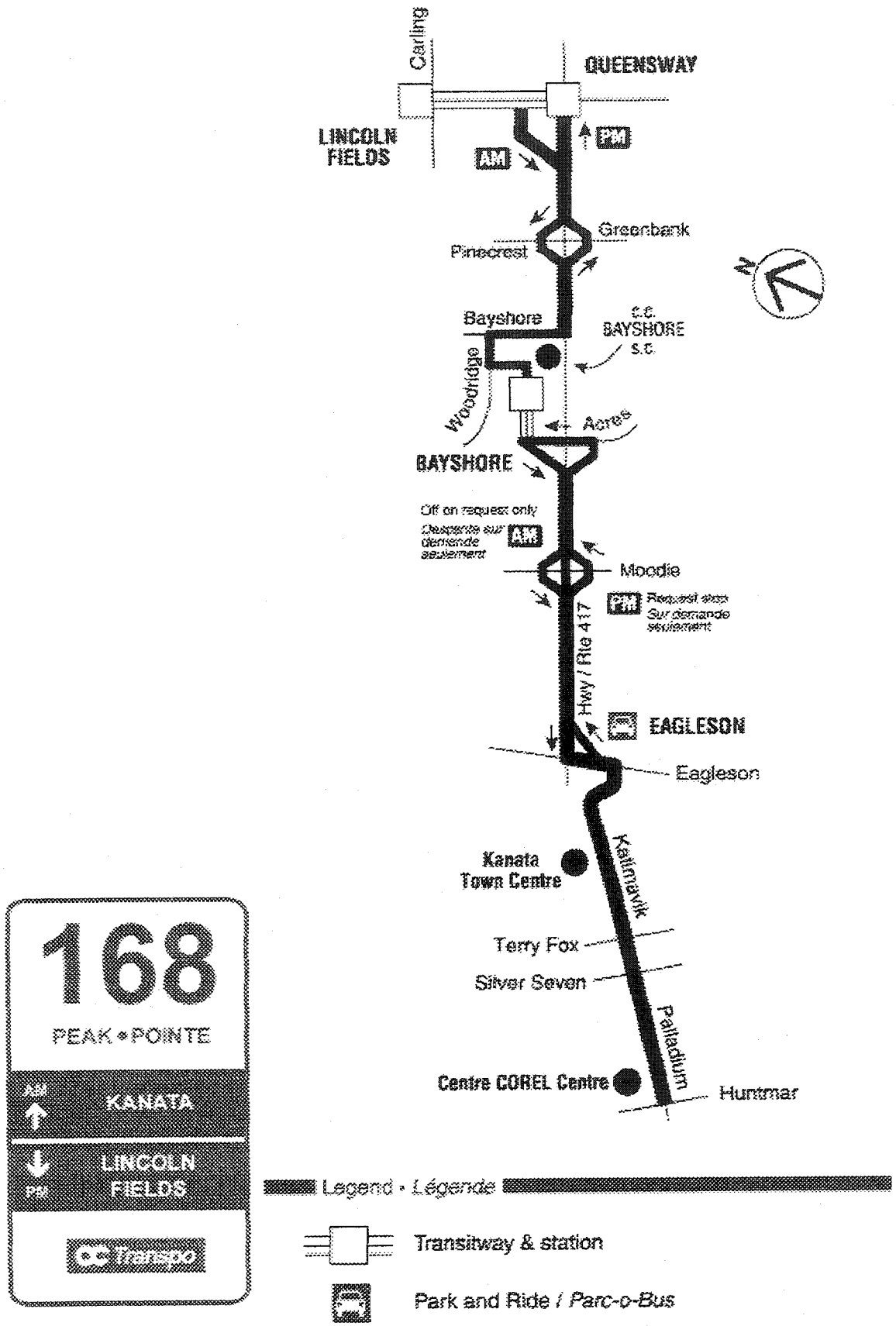
Transpo

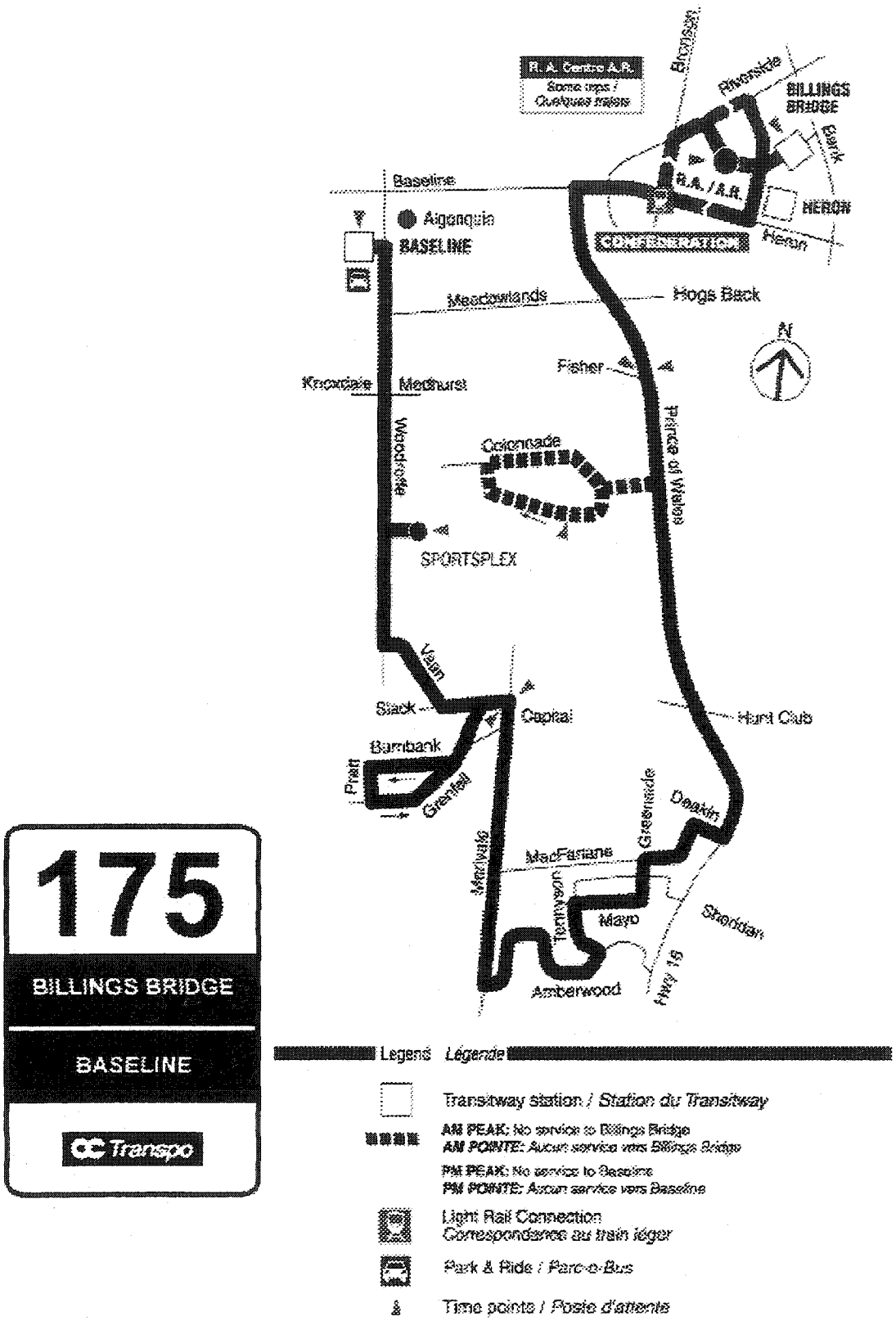
Legend • *Légende*

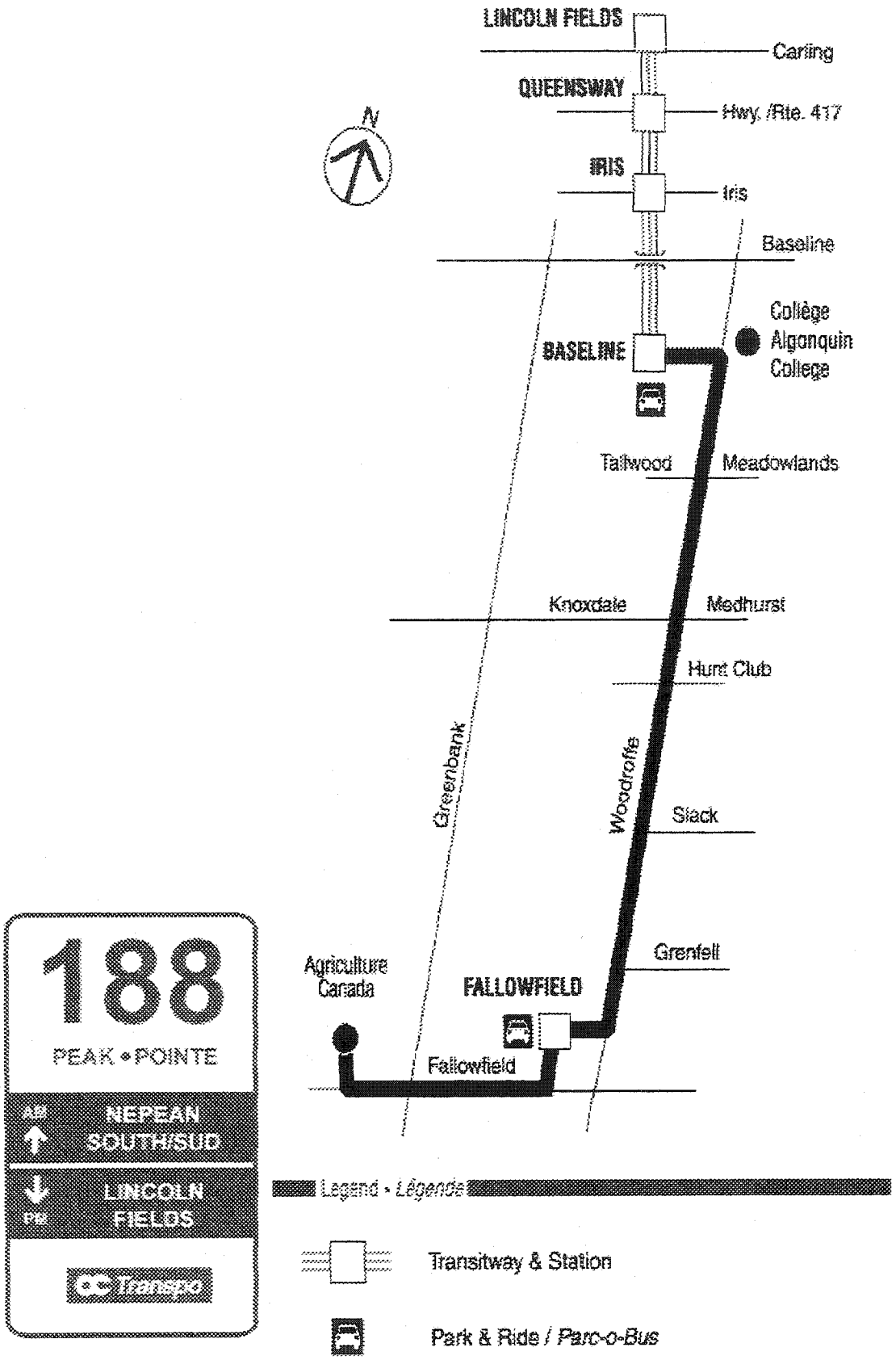
Transitway & Station / *Station et Transitway*

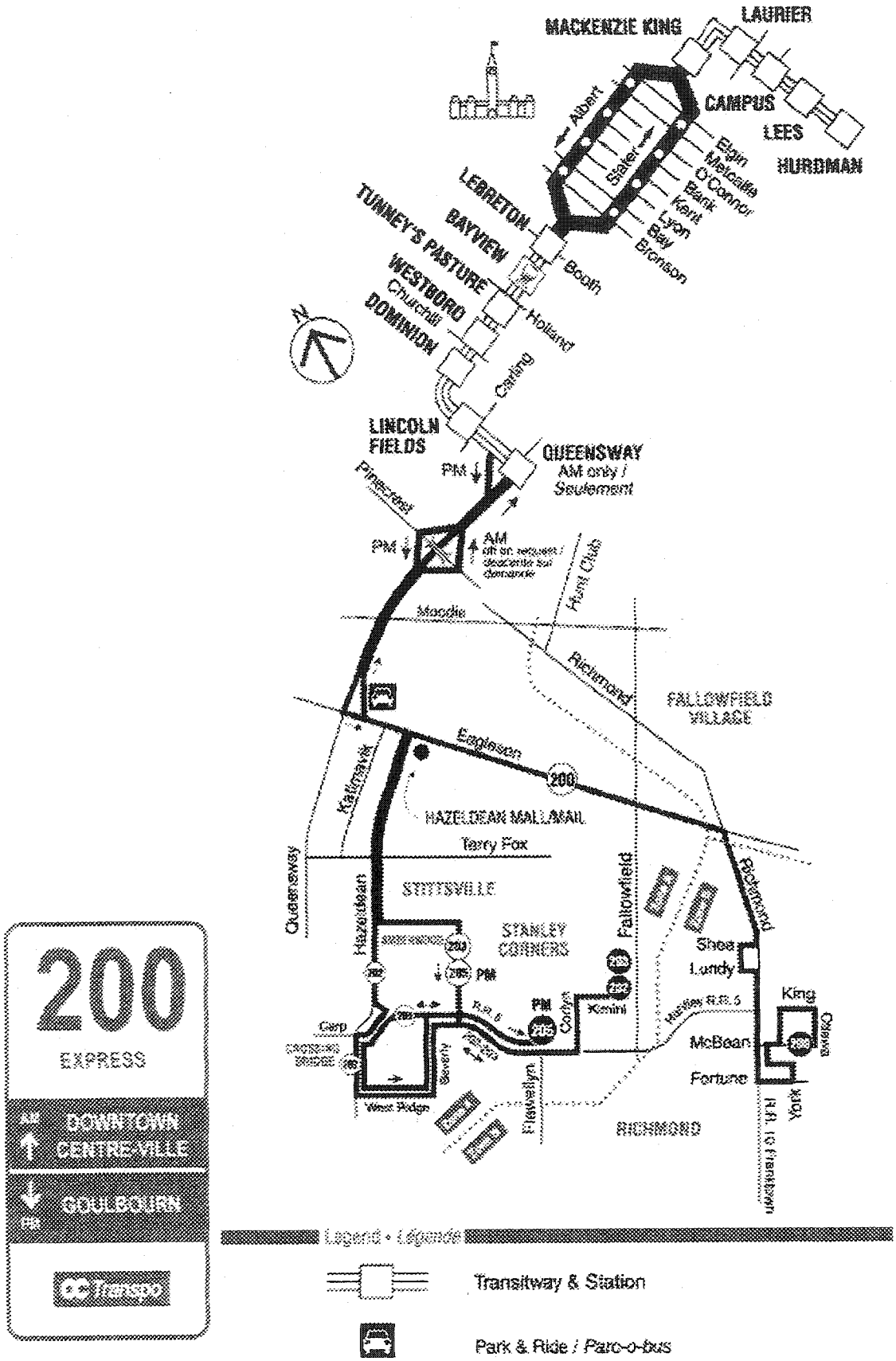
Some trips / *Quelques trajets*

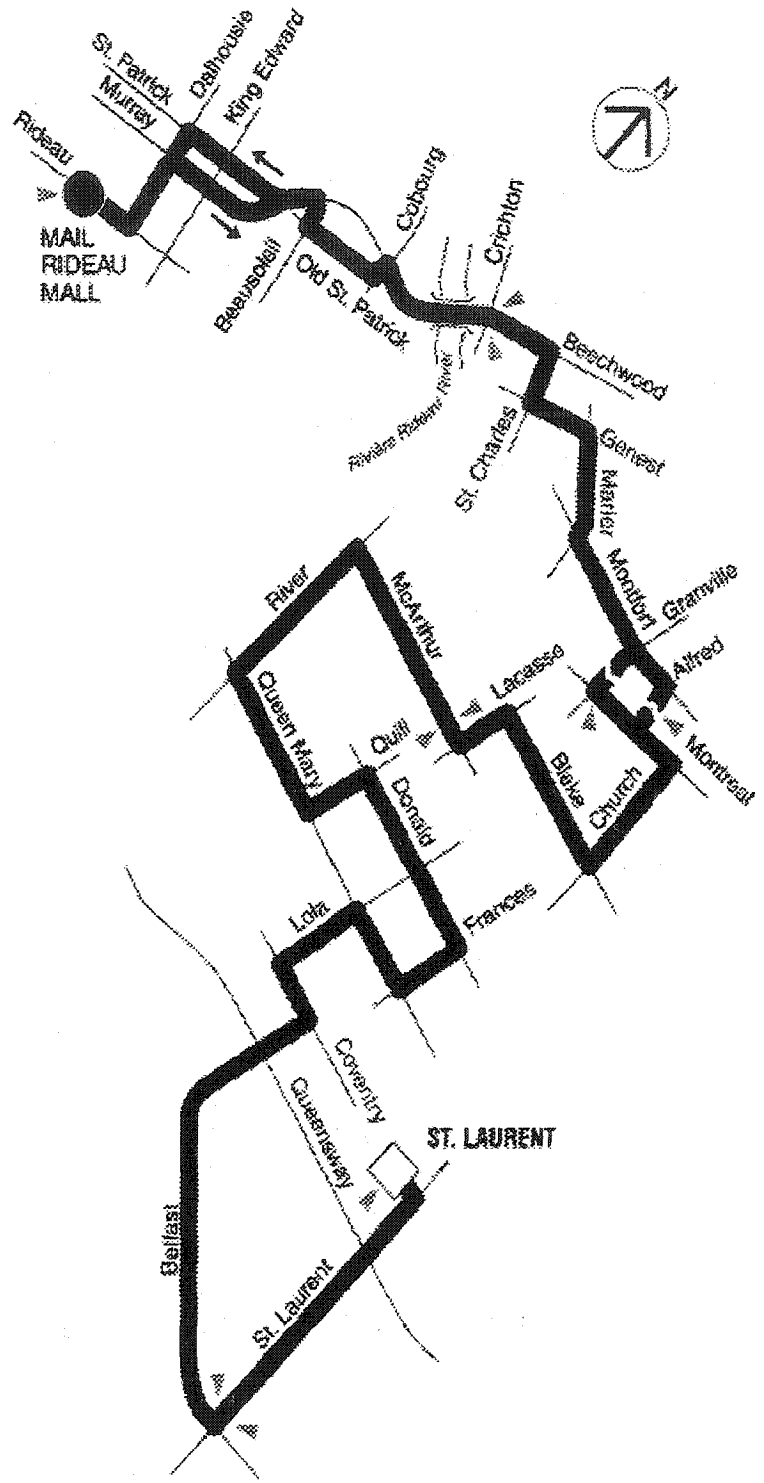
Time point / *Poste d'attente*


Carling
QUEENSWAY
LINCOLN FIELDS
AM
PM
Greenbank
Pinecrest
Bayshore
S.C. BAYSHORE S.C.
Woodridge
Acres
BAYSHORE
Off on request only
Descente sur demande seulement
AM
Moodie
PM
Request stop
Sur demande seulement
Hwy / Rte 417
EAGLESON
Eagleson
Kanata Town Centre
Kanatavik
Terry Fox
Silver Seven
Palladium
Centre COREL Centre
Huntmar
168
PEAK • POINTE
AM
KANATA
PM
LINCOLN FIELDS
Transitway
Legend • Légende
Transitway & station
Park and Ride / Parc-o-Bus


175
BILLINGS BRIDGE
BASELINE
Transpo
R.A. Centre A.R.
Some trips / Quelques trajets
Bronson
Riverside
BILLINGS BRIDGE
Bank
HERON
Heron
R.A. / A.R.
Baseline
Algonquin
BASELINE
Meadowlands
Hogs Back
N
Fisher
Knoxdale
Merivale
Woodroffe
Colonnade
Promenade du Moulin
SPORTSPLEX
Vanier
Slack
Capital
Hunt Club
Barrhaven
Pratt
Greenfield
Greenside
Deakin
Merivale
MacFarlane
Tennyson
Mayo
Sheridan
Hwy 16
Amberwood
Legend Légende
Transitway station / Station du Transitway
AM PEAK: No service to Billings Bridge
AM POINTE: Aucun service vers Billings Bridge
PM PEAK: No service to Baseline
PM POINTE: Aucun service vers Baseline
Light Rail Connection
Correspondance au train léger
Park & Ride / Parc-e-Bus
Time points / Points d'attente


LINCOLN FIELDS
Carling
QUEENSWAY
Hwy. /Rte. 417
N
IRIS
Iris
Baseline
Collège Algonquin College
BASELINE
Tallwood
Meadowlands
Knoxdale
Medhurst
Hunt Club
Greenbank
Woodroffe
Slack
Grenfell
Agriculture Canada
FALLOWFIELD
Fallowfield
188
PEAK • POINTE
AM
NEPEAN
SOUTH/SUD
PM
LINCOLN
FIELDS
Transitway
Legend • Légende
Transitway & Station
Park & Ride / Parc-o-Bus


MACKENZIE KING
LAURIER
CAMPUS
LEES
HURDMAN
Albert
Slater
Elgin
Metcalfe
O'Connor
Bank
Kent
Lyon
Bay
Bronson
LEBRETON
BAYVIEW
Booth
TUNNEY'S PASTURE
WESTBORO
Churchill
Holland
DOMINION
Carling
LINCOLN FIELDS
QUEENSWAY
AM only / Seulement
PM
Pinecrest
AM
Moodie
Hunt Club
Richmond
FALLOWFIELD VILLAGE
Eagleson
200
Kakamavik
HAZELDEAN MALL/MAIL
Terry Fox
Queensway
STITTSVILLE
Hazeldean
STANLEY CORNERS
Fallowfield
Richmond
Shea
Lundy
King
Ottawa
Kemi
McBean
Fortune
York
Carp
Beverly
Flewellyn
RICHMOND
West Ridge
200
EXPRESS
AM DOWNTOWN
CENTRE-VILLE
PM GOULBOURN
Transpo
Legend • Légende
Transitway & Station
Park & Ride / Parc-o-bus



829

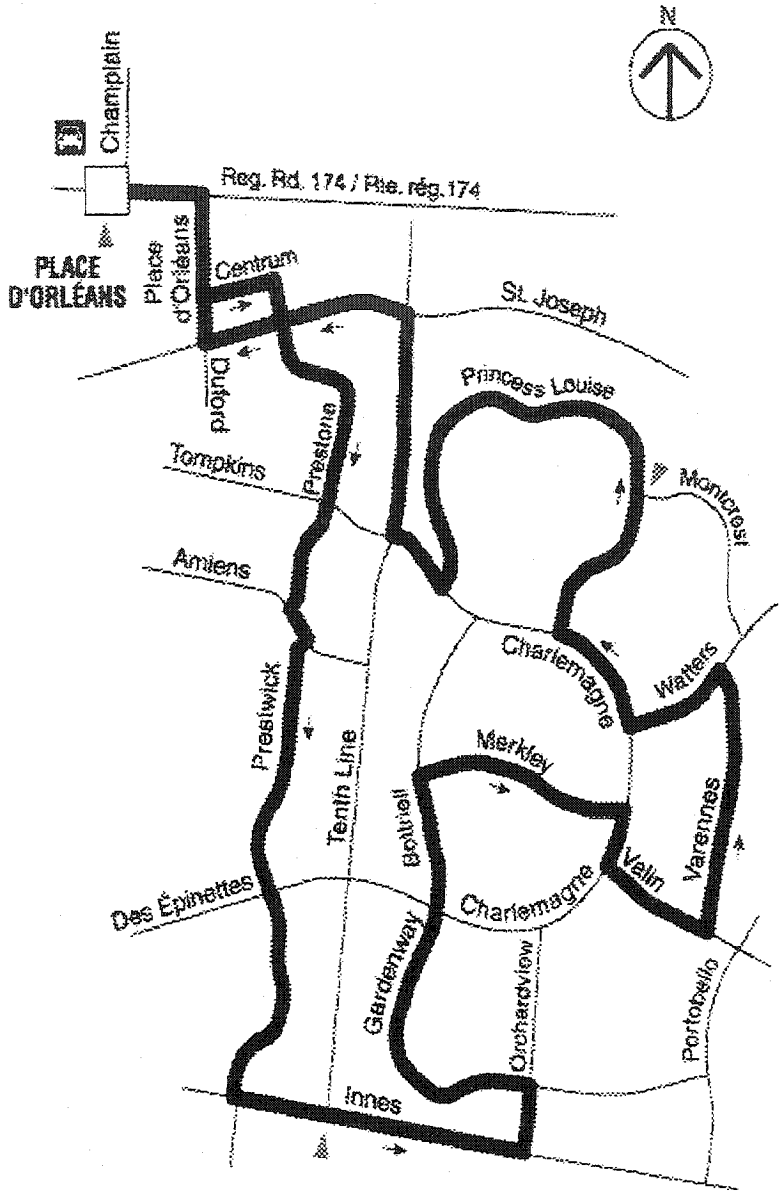
DOWNTOWN
CENTRE-VILLE

ST. LAURENT

Legend • *Légende*

Transitway station / *Station du Transitway*

Time point / *point d'attente*



835

ORLÉANS

LOCAL

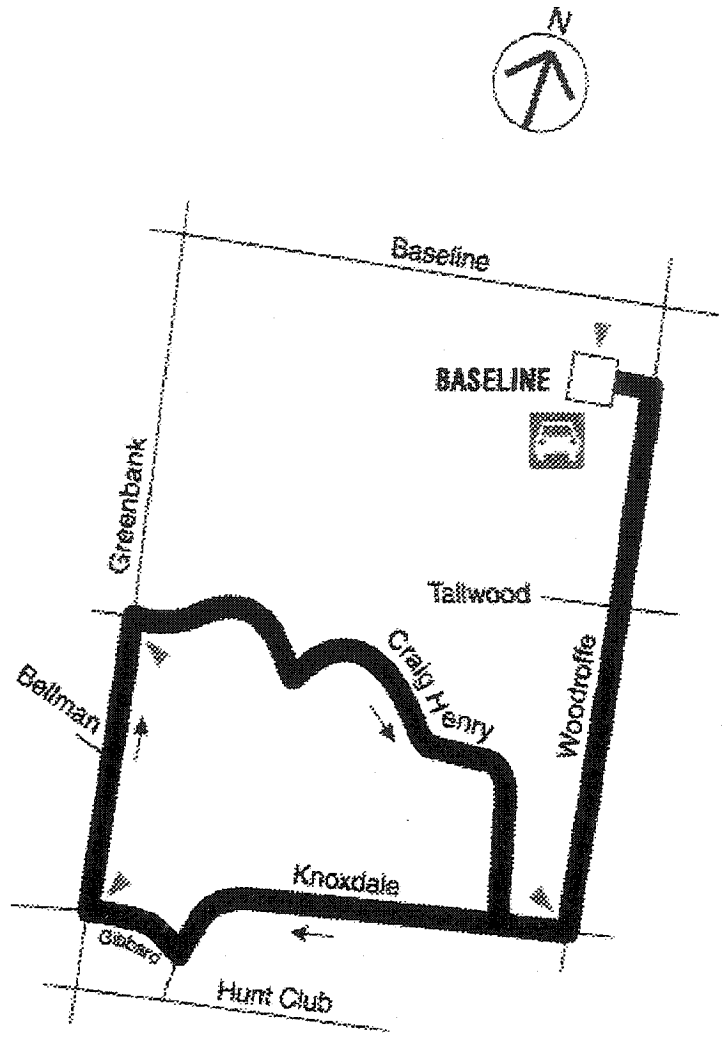
Legend • *Légende*

Transitway Station / *Station du Transitway*

Park & Ride / *Parc-o-Bus*

This route runs in the early morning only.
Monday to Friday 4am to 6am approx. /
Ce circuit circule tôt le matin seulement.
Du lundi au vendredi de 4h à 6h approx.

Time point / *Poste d'attente*



874

BASELINE

LOCAL

Legend • *Légende*

Transitway Station

Park & Ride / *Parc-o-Bus*

Time point / *Poste d'attente*

Appendix C
Avenue Coding

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Start
' Script Description: Begins the Transit Information System.
'                    Links to DeleteThemesAfterUse to delete
'                    files before use.
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    30-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'delete themes, tables, graphics before use

av.run("DeleteThemesAfterUse", "")


'starts the transit information system

av.FindDialog("TransitInfo").Open
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       DeleteThemesAfterUse
' Script Description: Deletes the themes and tables created during
'                     the Transit Info System as well as clears
'                     the selections from all of the permanent
'                     tables used.  This attached to TransitInfo
'                     dialog box to run before TIS begins.
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    15-Jun-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


' delete themes

theView = av.GetProject.FindDoc("Route Finding")
theTheme1 = theView.FindTheme("Route")

theTheme2 = theView.FindTheme("Route2")

theTheme3 = theView.FindTheme("Chosen Locations")

theTheme4 = theView.FindTheme("firstlocation")

theTheme5 = theView.FindTheme("Time")

theTheme6 = theView.FindTheme("Locate1.shp")

theTheme7 = theView.FindTheme("Route1")

' if the themes exist in the view delete them and their tables

if (theTheme1 = nil)then
  'msgbox.info("No theme found.", "")
else
  theTable1 = theTheme1.GetFTab

  theView.DeleteTheme(theTheme1)

 ' av.GetProject.RemoveDoc(theTable1)
'  theTable1 = nil

  'Deactivate the FTab
  theTable1.Deactivate
  theTable1 = nil

  'Delete the source files
  file.delete("F:\thesis data\Route.shp".asfilename)
  file.delete("F:\thesis data\Route.shx".asfilename)
  file.delete("F:\thesis data\Route.dbf".asfilename)

end

if (theTheme2 = nil)then
  'msgbox.info("No theme found.", "")
else
  theTable2 = theTheme2.GetFTab

  theView.DeleteTheme(theTheme2)

  'av.GetProject.RemoveDoc(theTable2)
'  theTable2 = nil
```

```
 'Deactivate the FTab
 theTable2.Deactivate
 theTable2 = nil

 'Delete the source files
 file.delete("F:\thesis data\Route2.shp".asfilename)
 file.delete("F:\thesis data\Route2.shx".asfilename)
 file.delete("F:\thesis data\Route2.dbf".asfilename)

end

if (theTheme3 = nil)then
 'msgbox.info("No theme found.", "")
else
 theTable3 = theTheme3.GetFTab

 theView.DeleteTheme(theTheme3)

 'av.GetProject.RemoveDoc(theTable3)
'  theTable3 = nil

 'Deactivate the FTab
 theTable3.Deactivate
 theTable3 = nil

 'Delete the source files
 file.delete("F:\thesis data\Chosen Locations.shp".asfilename)
 file.delete("F:\thesis data\Chosen Locations.shx".asfilename)
 file.delete("F:\thesis data\Chosen Locations.dbf".asfilename)

end

if (theTheme4 = nil)then
 'msgbox.info("No theme found.", "")
else
 theTable4 = theTheme4.GetFTab

 theView.DeleteTheme(theTheme4)

 'av.GetProject.RemoveDoc(theTable4)
'  theTable4 = nil

 'Deactivate the FTab
 theTable4.Deactivate
 theTable4 = nil

 'Delete the source files
 file.delete("F:\thesis data\firstlocation.shp".asfilename)
 file.delete("F:\thesis data\firstlocation.shx".asfilename)
 file.delete("F:\thesis data\firstlocation.dbf".asfilename)

end

if (theTheme5 = nil)then
 'msgbox.info("No theme found.", "")
else
 theTable5 = theTheme5.GetFTab

 theView.DeleteTheme(theTheme5)

 ' av.GetProject.RemoveDoc(theTable5)
'  theTable5 = nil

 'Deactivate the FTab
 theTable5.Deactivate
 theTable5 = nil

 'Delete the source files
 file.delete("F:\thesis data\Time.shp".asfilename)
```

```
 file.delete("F:\thesis data\Time.shx".asfilename)
 file.delete("F:\thesis data\Time.dbf".asfilename)

end

if (theTheme6 = nil)then
 'msgbox.info("No theme found.", "")
else
 theTable6 = theTheme6.GetFTab

 theView.DeleteTheme(theTheme6)

 'av.GetProject.RemoveDoc(theTable6)
'  theTable6 = nil

 'Deactivate the FTab
 theTable6.Deactivate
 theTable6 = nil

 'Delete the source files
 file.delete("F:\thesis data\Locate1.shp".asfilename)
 file.delete("F:\thesis data\Locate1.shx".asfilename)
 file.delete("F:\thesis data\Locate1.dbf".asfilename)

end


if (theTheme7 = nil)then
 'msgbox.info("No theme found.", "")
else
 theTable7 = theTheme7.GetFTab

 theView.DeleteTheme(theTheme6)

 'av.GetProject.RemoveDoc(theTable7)
'  theTable7 = nil

 'Deactivate the FTab
 theTable7.Deactivate
 theTable7 = nil

 'Delete the source files
 file.delete("F:\thesis data\Route1.shp".asfilename)
 file.delete("F:\thesis data\Route1.shx".asfilename)
 file.delete("F:\thesis data\Route1.dbf".asfilename)

end

'After you delete the themes from the View,
'send the command:

theView.Invalidate

'This will update the view with no themes in it.

'Also have to delete the tables that were created

'Remove the Table doc of the FTab if it exists

theTable8 = av.FindDoc("near.dbf")

if (theTable8 = nil)then
 'msgbox.info("No table found.", "")
else
 theVTab8 = theTable8.GetVTab

 av.GetProject.RemoveDoc(theTable8)
 theTable8 = nil

 'Deactivate the VTab
```

```
  theVTab8.Deactivate
  theVTab8 = nil

  'Delete the source files
  file.delete("F:\thesis data\near.dbf".asfilename)

end


theTable9 = av.FindDoc("Results.dbf ")

if (theTable9 = nil)then
  'msgbox.info("No table found.", "")
else
  theVTab9 = theTable9.GetVTab

  av.GetProject.RemoveDoc(theTable9)
  theTable9 = nil

  'Deactivate the VTab
  theVTab9.Deactivate
  theVTab9 = nil

  'Delete the source files
  file.delete("F:\thesis data\Results.dbf".asfilename)

end

'Delete extra source files
file.delete("F:\thesis data\Route1.shp".asfilename)
file.delete("F:\thesis data\Route1.shx".asfilename)
file.delete("F:\thesis data\Route1.dbf".asfilename)

file.delete("F:\thesis data\Route2.shp".asfilename)
file.delete("F:\thesis data\Route2.shx".asfilename)
file.delete("F:\thesis data\Route2.dbf".asfilename)

file.delete("F:\thesis data\Route3.shp".asfilename)
file.delete("F:\thesis data\Route3.shx".asfilename)
file.delete("F:\thesis data\Route3.dbf".asfilename)

file.delete("F:\thesis data\Time1.shp".asfilename)
file.delete("F:\thesis data\Time1.shx".asfilename)
file.delete("F:\thesis data\Time1.dbf".asfilename)

file.delete("F:\thesis data\Time2.shp".asfilename)
file.delete("F:\thesis data\Time2.shx".asfilename)
file.delete("F:\thesis data\Time2.dbf".asfilename)


'delete the temp script file created

theScript = av.FindDoc("tempscript")
if(theScript = nil)then
  'msgbox.info("No script.", "")
else
  av.GetProject.RemoveDoc(theScript)
  theScript = nil

  'Delete the source file
  file.delete("F:\thesis data\tempscript.ave".asfilename)
end

av.PurgeObjects

'Delete the source files
file.delete("F:\thesis data\ .dbf".asfilename)

'theProject = av.GetProject
'theDocs = theProject.GetSelectedDocs
```

```
'if (theDocs.IsEmpty) then
'  return nil
'end
'doc_names = ""
'all = false
'result = false
'for each d in theDocs
'  'if (NOT all) then
''    result = MsgBox.AllYesNo( "Are you sure you want to delete"++d.GetName, "Delete", NIL )
''    if (result = nil) then
''      all = true
''    elseif (result = false) then
''      continue
''    end
''  end
''  if(d.Is(View)) then
''    editThm = d.GetEditableTheme
''    if (editThm <> nil) then
''      doSave = MsgBox.YesNoCancel("Save Edits to "+editThm.GetName+" in "+
''      d.GetName+"?", "Stop Editing", true)
''      if (doSave = nil) then
''        continue
''      end
''      if (editThm.StopEditing(doSave).Not) then
''        MsgBox.Info("Unable to Save Edits to Theme "
''                  + editThm.GetName +
''                  ", please use the Save Edits As option", "")
''        continue
''      else
''        d.SetEditableTheme(NIL)
''      end
''    end
''  end
''  if (d.Is(Table)) then
''    if (d.GetVTab.IsBeingEditedWithrecovery) then
''      doSave = MsgBox.YesNoCancel("Save Edits to the table "+d.GetName+
''      "?", "Stop Editing", True)
''      if (doSave = nil) then
''        continue
''      end
''      if (d.GetVtab.StopEditingWithRecovery(doSave).Not) then
''        MsgBox.Info("Unable to Save Edits to Table " + d.GetName +
''                  ", please use the Save Edits As option", "")
''        continue
''      end
''    end
''  end
'  av.GetProject.RemoveDoc( d )
'end
'
'av.PurgeObjects


'delete graphics

theView = av.GetProject.FindDoc("Route Finding")
theView.GetGraphics.SelectAll
theView.GetGraphics.ClearSelected


'clear all of the tables

theView = av.GetActiveDoc
thetable1 = av.FindDoc("weekday.dbf")
theVTab = thetable1.GetVTab
theBitmap1 = theVTab.GetSelection
theBitmap1.ClearAll

thetable2 = av.FindDoc("saturday.dbf")
theVTab = thetable2.GetVTab
```

```
theBitmap2 = theVTab.GetSelection
theBitmap2.ClearAll

thetable3 = av.FindDoc("sunday.dbf")
theVTab = thetable3.GetVTab
theBitmap3 = theVTab.GetSelection
theBitmap3.ClearAll

theTheme1 = theView.FindTheme("Bus Stops")
theFTab = theTheme1.GetFTab
theBitmap4 = theFTab.GetSelection
theBitmap4.ClearAll
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       TransitInfo.Update
' Script Description: Enables the dialog to run only if there
'                   is an active view with themes.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


theDoc = av.GetActiveDoc
if ((theDoc.is(View)) and (theDoc.GetThemes.count>0)) then
 SELF.SetEnabled(true)
else
 SELF.SetEnabled(false)
end
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        TransitInfo.schedule_lbt.Update
' Script Description: Enables the schedule button.
'
' Programmer:         Sarah Riley
' Date Created:       Apr. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'
SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       TransitInfo.schedule_lbt.Click
' Script Description: Script runs when the schedule button is
'                  pressed.  It calls to open the userinterface2
'                  dialog.
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'SELF refers to the button

SELF.SetEnabled(true)

'closes the dialog
SELF.GetDialog.Close

'run the userinterface2 dialog
av.FindDialog("userinterface2").Open
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       TransitInfo.proceed_lbt.Update
' Script Description: Enables the proceed button in the dialog.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'
SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       TransitInfo.proceed_lbt.Click
' Script Description: Script runs when the proceed button is pressed.
'                     It calls to open the userinterface dialog.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley   Jan-2002     Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'SELF refers to the button

SELF.SetEnabled(true)

'closes the dialog
SELF.GetDialog.Close

'run the userinterface dialog
av.FindDialog("userinterface").Open
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        TransitInfo.Open
' Script Description: Opens the TransitInfo dialog and initializes
'                     all of the objects within.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'

'open the dialog box
theDialog = av.FindDialog("TransitInfo")

'finds the box within the dialog
proceed_lbt = theDialog.FindByName("proceed_lbt")
exit_lbt = theDialog.FindByName("exit_lbt")
schedule_lbt = theDialog.FindByName("schedule_lbt")

'runs the scripts for each selection
'proceed_lbt.Update
'exit_lbt.Update
'schedule_lbt.Update
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       TransitInfo.exit_lbt.Update
' Script Description: Enables the TransitInfo dialog.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'

'always true, always available

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       TransitInfo.exit_lbt.Click
' Script Description: Script runs when the exit button is pressed.
'                  It closes the dialog and makes sure the user
'                  really wants to.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley   Jan-2002     Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'SELF refers to the button
'closes the dialog

if(msgbox.yesno("Are you sure?","Quit?",false)) then
  self.GetDialog.Close
  quit = true
end

'SELF.GetDialog.Close
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       userinterface2.Update
' Script Description: Updates the Dialog.
'
' Programmer:        Sarah Riley
' Date Created:      April 16, 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'always true, always, enabled

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        userinterface2.self.Update
' Script Description: Broadcast the update events.
'
' Programmer:         Sarah Riley
' Date Created:       16-Apr. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'

SELF.BroadcastUpdate
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       userinterface2.select_lbt.Update
' Script Description: Makes sure the select button is always
'                    available for selection.
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 16, 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'always true, always available

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        userinterface2.select_lbt.Click
' Script Description: Takes the information entered within this
'                     Dialog and performs the appropriate queries.
'
' Programmer:         Sarah Riley
' Date Created:       April 16, 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    29-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'set SELF and default header

SELF.SetEnabled(true)

'get the view and set it active
theView = av.GetProject.FindDoc("Route Finding")
if (theView = nil) then
  MsgBox.Info("Can't find view: " + theView.AsString, "")
  SELF.GetDialog.Close
  exit
end

'close the dialog so that it won't appear once queries are performed
SELF.GetDialog.Close


'when the time of day is entered link it to the am/pm
'radio buttons---this is then used as a basis for a time
'in the schedule table
'if nothing is entered then only best route is calculated
'for the user

'get the time data
thehour = SELF.GetDialog.FindByName("timehour_txt")
theminute = SELF.GetDialog.FindByName("timeminute_txt")

'get time of day user radio button selection from control panel
cpa = SELF.GetDialog.FindByName("time_cpa")

if (cpa = nil) then
  MsgBox.Info("Can't find dialog", "")
  SELF.GetDialog.Close
end

cpaChoice = cpa.GetSelected

if (cpaChoice = nil) then
  MsgBox.Info("No time of day entered (am/pm)", "")
  SELF.GetDialog.Close
end

if (thehour.IsEmpty = true) then
  thehour = 0
else
  thehour = thehour.GetText.AsNumber
end

if (theminute.IsEmpty = true) then
  theminute = 0
  msgbox.Info("No time has been selected for trip.","")
else
  theminute = theminute.GetText.AsNumber
end
```

```
'get the am/pm selection
am = SELF.GetDialog.FindByName("am_rad")
pm = SELF.GetDialog.FindByName("pm_rad")

'take the time entered a put it in the format found in
'the schedule tables
if (cpaChoice = am) then
 time1 = thehour.AsString
 if (time1 = "12") then
  time = "0"
  time1 = time.AsString
  time2 = theminute.AsString
  time = time1.AsString + time2.AsString

  'for query
  less = time.AsNumber - 20
  lesstime = "0" + less.AsString
  more = time.AsNumber + 30
  moretime = "0" + more.AsString
  'MsgBox.Info(lesstime + NL + moretime, "")
 else
 time2 = theminute.AsString

 'join these two times
 time = time1.AsString + time2.AsString

 'for query
 less = time.AsNumber - 20
 lesstime = less.AsString
 more = time.AsNumber + 30
 moretime = more.AsString
 'MsgBox.Info(lesstime + NL + moretime, "")
 end

elseif (cpaChoice = pm) then
 time1 = thehour + 12
 if (time1 = 24) then
  time1 = thehour.AsString
  time2 = theminute.AsString
  time = time1.AsString + time2.AsString
 else
 times = time1.AsString
 time2 = theminute.AsString

 'join these two times
 time = times.AsString + time2.AsString
 end

 'for query
 less = time.AsNumber - 20
 lesstime = less.AsString
 more = time.AsNumber + 30
 moretime = more.AsString
 'MsgBox.Info(lesstime + NL + moretime, "")

else
  MsgBox.Info("The time will be ignored in finding a route.", "")
end


'get weekday user radio button selection from control panel
cpa2 = SELF.GetDialog.FindByName("week_cpa")

if (cpa2 = nil) then
 MsgBox.Info("Can't find dialog", "")
 SELF.GetDialog.Close
end
```

```
cpa2Choice = cpa2.GetSelected

if (cpa2Choice = nil) then
  MsgBox.Info("No day of week selected", "")
  SELF.GetDialog.Close
end

'get the route chosen by the user
alist = SELF.GetDialog.FindByName("route_cbx")
aRoute = alist.GetSelection

'if weekday selected then go to table schedule-weekday.txt for the query
'if Saturday selected then go to table saturday.txt for the query
'if Sunday selected then go to table sunday.txt for the query
'the query will find time in the format as follows: 6:40 AM

weekday = SELF.GetDialog.FindByName("weekday_rad")
saturday = SELF.GetDialog.FindByName("saturday_rad")
sunday = SELF.GetDialog.FindByName("sunday_rad")
aRoute1 = aRoute.AsString

'query the appropriate schedule table for time and route selected
if (cpa2Choice = weekday) then
  thetable1 = av.FindDoc("weekday.dbf")
  theVTab = thetable1.GetVTab
  theBitmap1 = theVTab.GetSelection
  myQueryString = "[Time] >= " + lesstime.quote
  myQueryString2 = "[Time] <= " + moretime.quote
  myQueryString3 = "[Route_no] = " + aRoute1
  theVTab.Query(myQueryString, theBitmap1, #VTAB_SELTYPE_NEW)
  theVTab.UpdateSelection
  theVTab.Query(myQueryString2, theBitmap1, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection
  theVTab.Query(myQueryString3, theBitmap1, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection

  ' show the table with promoted selected records
  ' sorted by direction and then run number

  'theVTab.GetSelection
  TabWin = thetable1.GetWin
  TabWin.Resize (600, 600)
  TabWin.Open

  thetable1.PromoteSelection
  theVTab = thetable1
  aField = theTable1.GetVTab.FindField("Direction")
  thetable1.Sort(aField, FALSE)
  thetable1.PromoteSelection
  aField2 = theTable1.GetVTab.FindField("Run_no")
  thetable1.Sort(afield2, FALSE)
  thetable1.PromoteSelection

  MsgBox.Info("The results that match your selections are highlighted in yellow.", "Route Information")


elseif (cpa2Choice = saturday) then
  thetable2 = av.FindDoc("saturday.dbf")
  theVTab = thetable2.GetVTab
  theBitmap2 = theVTab.GetSelection
  myQueryString = "[Time] >= " + lesstime.quote
  myQueryString2 = "[Time] <= " + moretime.quote
  myQueryString3 = "[Route_no] = " + aRoute1
  theVTab.Query(myQueryString, theBitmap2, #VTAB_SELTYPE_NEW)
  theVTab.UpdateSelection
  theVTab.Query(myQueryString2, theBitmap2, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection
  theVTab.Query(myQueryString3, theBitmap2, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection
```

```
  ' show the table with promoted selected records
  ' sorted by direction and then run number

  'theVTab.GetSelection
  TabWin = thetable2.GetWin
  TabWin.Resize (600, 600)
  TabWin.Open

  thetable2.PromoteSelection
  theVTab = thetable2
  aField = theTable2.GetVTab.FindField("Direction")
  thetable2.Sort(aField, FALSE)
  thetable2.PromoteSelection
  aField2 = theTable2.GetVTab.FindField("Run_no")
  thetable2.Sort(afield2, FALSE)
  thetable2.PromoteSelection

  MsgBox.Info("The results that match your selections are highlighted in yellow.", "Route Information")


elseif (cpa2Choice = sunday) then
  thetable3 = av.FindDoc("sunday.dbf")
  theVTab = thetable3.GetVTab
  theBitmap3 = theVTab.GetSelection
  myQueryString = "[Time] >= " + lesstime.quote
  myQueryString2 = "[Time] <= " + moretime.quote
  myQueryString3 = "[Route_no] = " + aRoute1
  theVTab.Query(myQueryString, theBitmap3, #VTAB_SELTYPE_NEW)
  theVTab.UpdateSelection
  theVTab.Query(myQueryString2, theBitmap3, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection
  theVTab.Query(myQueryString3, theBitmap3, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection

  ' show the table with promoted selected records
  ' sorted by direction and then run number

  'theVTab.GetSelection
  TabWin = thetable3.GetWin
  TabWin.Resize (600, 600)
  TabWin.Open

  thetable3.PromoteSelection
  theVTab = thetable3
  aField = theTable3.GetVTab.FindField("Direction")
  thetable3.Sort(aField, FALSE)
  thetable3.PromoteSelection
  aField2 = theTable3.GetVTab.FindField("Run_no")
  thetable3.Sort(afield2, FALSE)
  thetable3.PromoteSelection

  MsgBox.Info("The results are highlighted.", "Route Information")


else
  msgbox.Info ("Nothing", "")

end


SELF.GetDialog.Close
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        userinterface2.route_cbx.Update
' Script Description: This script fills the listbox in the Dialog
'                     with a list of bus routes.
'
' Programmer:         Sarah Riley
' Date Created:       Apr. 2002
' Returned Object:    none
'
' Changes:
'
' Author     Date          Details
'
' S. Riley   16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


' create a list of numbers

alist = {3, 7, 35, 57, 59, 66, 77, 86, 95, 97, 116, 117, 118, 129, 137, 151, 168, 175, 188, 200, 829, 874}

'Add the list contents to the listbox

self.DefineFromList(alist)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      userinterface2.Open
' Script Description: Opens the dialog box and initialized each
'                   item within the dialog.
'
' Programmer:       Sarah Riley
' Date Created:     Apr. 2002
' Returned Object:  none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    16-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'open the dialog box
theDialog = av.FindDialog("userinterface2")

'move the dialog out of view but not close it right away
theDialog.Move(0, 0)


'clear textlines
'declare and update all controls in first step

a = "12"
b = "30"
timehour_txl = theDialog.FindByName("timehour_txl")
timehour_txl.SetText(a)
timehour_txl.Update

timeminute_txl = theDialog.FindByName("timeminute_txl")
timeminute_txl.SetText(b)
timeminute_txl.Update

am_rad = theDialog.FindByName("am_rad")
'am_rad.Update

pm_rad = theDialog.FindByName("pm_rad")
'pm_rad.Update

time_cpa = theDialog.FindByName("time_cpa")
'time_cpa.Update

weekday_rad = theDialog.FindByName("weekday_rad")
'weekday_rad.Update

saturday_rad = theDialog.FindByName("saturday_rad")
'saturday_rad.Update

sunday_rad = theDialog.FindByName("sunday_rad")
'sunday_rad.Update

week_cpa = theDialog.FindByName("week_cpa")
'week_cpa.Update

route_cbx = theDialog.FindByName("route_cbx")
route_cbx.Update

select_lbt = theDialog.FindByName("select_lbt")
'select_lbt.Update

cancel_lbt = theDialog.FindByName("reset_lbt")
'cancel_lbt.Update

'Set listeners
timehour_txl.SetListeners({select_lbt})
timeminute_txl.SetListeners({select_lbt})
```

```
time_cpa.SetListeners({select_lbt})
week_cpa.SetListeners({select_lbt})
route_cbx.SetListeners({select_lbt})

'make one radio button active as a default

am_rad = theDialog.FindByName("am_rad")
am_rad.Select'(true)

weekday = theDialog.FindByName("weekday_rad")
weekday.Select'(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        userinterface2.cancel_lbt.Update
' Script Description: Allows the cancel button to always be available.
'
' Programmer:         Sarah Riley
' Date Created:       April 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley    16-Apr-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always available

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       userinterface2.cancel_lbt.Click
' Script Description: Closes the dialog when the cancel button
'                  is selected.
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    16-Apr-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'close the dialog box

SELF.GetDialog.Close
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       userinterface.Update
' Script Description:
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always, enabled

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        userinterface.self.Update
' Script Description: Broadcasts the update events.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author       Date        Details
'
' S. Riley     Jan-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

SELF.BroadcastUpdate
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       userinterface.select_lbt.Update
' Script Description: Enables the select button.  Always active.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always available

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       userinterface.select_lbt.Click
' Script Description: This script is run when the user submits their
'                     request.  It queries the schedule table and
'                     passes information to the next Dialog.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    2-May-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'set SELF and default header

SELF.SetEnabled(true)

'get the view and set it active
theView = av.GetProject.FindDoc("Route Finding")
if (theView = nil) then
  MsgBox.Info("Can't find view: " + theView.AsString, "")
  SELF.GetDialog.Close
  exit
end

'close the dialog so that it won't appear when Address Selection is run
SELF.GetDialog.Close


'when the time of day is entered link it to the am/pm
'radio buttons—this is then used as a basis for a time in the schedule table
'if nothing is entered then only best route is calculated for the user

'get the time data
thehour = SELF.GetDialog.FindByName("timehour_txl")
theminute = SELF.GetDialog.FindByName("timeminute_txl")

'get time of day user radio button selection from control panel
cpa = SELF.GetDialog.FindByName("time_cpa")

if (cpa = nil) then
  MsgBox.Info("Can't find dialog", "")
  SELF.GetDialog.Close
end

cpaChoice = cpa.GetSelected

if (cpaChoice = nil) then
  MsgBox.Info("No time of day entered (am/pm)", "")
  SELF.GetDialog.Close
end

if (thehour.IsEmpty = true) then
  thehour = 0
else
  thehour = thehour.GetText.AsNumber
end

if (theminute.IsEmpty = true) then
  theminute = 0
  msgbox.Info("No time has been selected for trip.","")
else
  theminute = theminute.GetText.AsNumber
end
```

```
'get the am/pm selection
am = SELF.GetDialog.FindByName("am_rad")
pm = SELF.GetDialog.FindByName("pm_rad")

'convert the time to correspond to the schedule table format
if (cpaChoice = am) then
  time1 = thehour.AsString
  if (time1 = "12") then
    time = "0"
    time1 = time.AsString
    time2 = theminute.AsString
    time = time1.AsString + time2.AsString

    'for query
    less = time.AsNumber
    lesstime = "0" + less.AsString
    more = time.AsNumber + 30
    moretime = "0" + more.AsString
    'MsgBox.Info(lesstime + NL + moretime, "")
  else
  time2 = theminute.AsString

  'join these two times
  time = time1.AsString + time2.AsString

  'for query
  less = time.AsNumber
  lesstime = less.AsString
  more = time.AsNumber + 30
  moretime = more.AsString
  'MsgBox.Info(lesstime + NL + moretime, "")
  end
elseif (cpaChoice = pm) then
  time1 = thehour + 12
  if (time1 = 24) then
    time1 = thehour.AsString
    time2 = theminute.AsString
    time = time1.AsString + time2.AsString
  else
  times = time1.AsString
  time2 = theminute.AsString

  'join these two times
  time = times.AsString + time2.AsString
  end

  'for query
  less = time.AsNumber
  lesstime = less.AsString
  more = time.AsNumber + 30
  moretime = more.AsString
  'MsgBox.Info(lesstime + NL + moretime, "")

else
    MsgBox.Info("The time will be ignored in finding a route.", "")
end


'get weekday user radio button selection from control panel
cpa2 = SELF.GetDialog.FindByName("week_cpa")

if (cpa2 = nil) then
  MsgBox.Info("Can't find dialog", "")
  SELF.GetDialog.Close
end

cpa2Choice = cpa2.GetSelected

if (cpa2Choice = nil) then
```

```
 MsgBox.Info("No day of week selected", "")
 SELF.GetDialog.Close
end

'get direction user radio button selection from control panel
cpa_d = SELF.GetDialog.FindByName("Direction_cpa")

if (cpa_d = nil)then
 MsgBox.Info("Can't find dialog", "")
 SELF.GetDialog.Close
end

cpa_dChoice = cpa_d.GetSelected

if(cpa_dChoice = nil)then
 MsgBox.Info("No direction selected", "")
 SELF.GetDialog.Close
end


'if weekday selected then go to table weekday.dbf for the query
'if Saturday selected then go to table saturday.dbf for the query
'if Sunday selected then go to table sunday.dbf for the query
'the query will find time in the format as follows: 6:40 AM

'in this step, the direction of travel is also selected from the tables

weekday = SELF.GetDialog.FindByName("weekday_rad")
saturday = SELF.GetDialog.FindByName("saturday_rad")
sunday = SELF.GetDialog.FindByName("sunday_rad")
NoB = SELF.GetDialog.FindByName("NB_rad")
SoB = SELF.GetDialog.FindByName("SB_rad")
WeB = SELF.GetDialog.FindByName("WB_rad")
EaB = SELF.GetDialog.FindByName("EB_rad")
InB = SELF.GetDialog.FindByName("IB_rad")
OtB = SELF.GetDialog.FindByName("OB_rad")

dir1 = "NB"   'northbound
dir2 = "SB"   'southbound
dir3 = "WB"   'westbound
dir4 = "EB"   'eastbound
dir5 = "IB"   'inbound
dir6 = "OB"   'outbound

if (cpa2Choice = weekday) then
 thetable1 = av.FindDoc("weekday.dbf")
 theVTab = thetable1.GetVTab
 theBitmap1 = theVTab.GetSelection

'query the schedule table for direction
if(cpa_dChoice = NoB)then
   myQueryString3 = "[Direction] = " + dir1.quote
   theVTab.Query(myQueryString3, theBitmap1, #VTAB_SELTYPE_NEW)
   theVTab.UpdateSelection
  elseif(cpa_dChoice = SoB)then
   myqueryString4 = "[Direction] = " + dir2.quote
   theVTab.Query(myQueryString4, theBitmap1, #VTAB_SELTYPE_NEW)
   theVTab.UpdateSelection
  elseif(cpa_dChoice = WeB)then
   myqueryString5 = "[Direction] = " + dir3.quote
   theVTab.Query(myQueryString5, theBitmap1, #VTAB_SELTYPE_NEW)
   theVTab.UpdateSelection
  elseif(cpa_dChoice = EaB)then
   myqueryString6 = "[Direction] = " + dir4.quote
   theVTab.Query(myQueryString6, theBitmap1, #VTAB_SELTYPE_NEW)
   theVTab.UpdateSelection
  elseif(cpa_dChoice = InB)then
   myqueryString7 = "[Direction] = " + dir5.quote
   theVTab.Query(myQueryString7, theBitmap1, #VTAB_SELTYPE_NEW)
   theVTab.UpdateSelection
```

```
  elseif(cpa_dChoice = OtB)then
    myqueryString8 = "[Direction] = " + dir6.quote
    theVTab.Query(myQueryString8, theBitmap1, #VTAB_SELTYPE_NEW)
    theVTab.UpdateSelection
  else
    msgbox.Info ("Nothing", "")
  end

 'query the schedule table for the time
  myQueryString = "[Time] >=" + lesstime.quote
  myQueryString2 = "[Time] <=" + moretime.quote
  theVTab.Query(myQueryString, theBitmap1, #VTAB_SELTYPE_NEW)
  theVTab.UpdateSelection
  theVTab.Query(myQueryString2, theBitmap1, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection

  'the schedule tables are permanently linked by me before
  'the program starts, but this is the code that is used

  "link to the bus stop theme to a schedule table
 '  theField = theVTab.FindField("Bus_stop")  'from dbf
 '  theTheme = theView.FindTheme("Bus Stops") 'fTheme
 '  aVTab = theTheme.GetFTab
 '  aField = aVTab.FindField("Bus_stop")
 '  theVTab.Link(theField, aVtab, aField)


 elseif (cpa2Choice = saturday) then
  thetable2 = av.FindDoc("saturday.dbf")
  theVTab = thetable2.GetVTab
  theBitmap2 = theVTab.GetSelection

   'query the schedule table for direction
   if(cpa_dChoice = NoB)then
     myQueryString3 = "[Direction] = " + dir1.quote
     theVTab.Query(myQueryString3, theBitmap1, #VTAB_SELTYPE_NEW)
     theVTab.UpdateSelection
   elseif(cpa_dChoice = SoB)then
     myqueryString4 = "[Direction] = " + dir2.quote
     theVTab.Query(myQueryString4, theBitmap1, #VTAB_SELTYPE_NEW)
     theVTab.UpdateSelection
   elseif(cpa_dChoice = WeB)then
     myqueryString5 = "[Direction] = " + dir3.quote
     theVTab.Query(myQueryString5, theBitmap1, #VTAB_SELTYPE_NEW)
     theVTab.UpdateSelection
   elseif(cpa_dChoice = EaB)then
     myqueryString6 = "[Direction] = " + dir4.quote
     theVTab.Query(myQueryString6, theBitmap1, #VTAB_SELTYPE_NEW)
     theVTab.UpdateSelection
   elseif(cpa_dChoice = InB)then
     myqueryString7 = "[Direction] = " + dir5.quote
     theVTab.Query(myQueryString7, theBitmap1, #VTAB_SELTYPE_NEW)
     theVTab.UpdateSelection
   elseif(cpa_dChoice = OtB)then
     myqueryString8 = "[Direction] = " + dir6.quote
     theVTab.Query(myQueryString8, theBitmap1, #VTAB_SELTYPE_NEW)
     theVTab.UpdateSelection
   else
     msgbox.Info ("Nothing", "")
   end

   'query schedule table for the time
   myQueryString = "[Time] >=" + lesstime.quote
   myQueryString2 = "[Time] <=" + moretime.quote
   theVTab.Query(myQueryString, theBitmap2, #VTAB_SELTYPE_NEW)
   theVTab.UpdateSelection
   theVTab.Query(myQueryString2, theBitmap2, #VTAB_SELTYPE_AND)
   theVTab.UpdateSelection

   'the schedule tables are permanently linked by me before
```

```
 'the program starts, but this is the code that is used

 "link to the bus stop theme
' theField = theVTab.FindField("Bus_stop")  'from dbf
' theTheme = theView.FindTheme("Bus Stops")  'fTheme
' aVTab = theTheme.GetFTab
' aField = aVTab.FindField("Bus_stop")
' theVTab.Link(theField, aVtab, aField)


elseif (cpa2Choice = sunday) then
 thetable3 = av.FindDoc("sunday.dbf")
 theVTab = thetable3.GetVTab
 theBitmap3 = theVTab.GetSelection

  'query schedule table for direction
  if(cpa_dChoice = NoB)then
    myQueryString3 = "[Direction] = " + dir1.quote
    theVTab.Query(myQueryString3, theBitmap1, #VTAB_SELTYPE_NEW)
    theVTab.UpdateSelection
  elseif(cpa_dChoice = SoB)then
    myqueryString4 = "[Direction] = " + dir2.quote
    theVTab.Query(myQueryString4, theBitmap1, #VTAB_SELTYPE_NEW)
    theVTab.UpdateSelection
  elseif(cpa_dChoice = WeB)then
    myqueryString5 = "[Direction] = " + dir3.quote
    theVTab.Query(myQueryString5, theBitmap1, #VTAB_SELTYPE_NEW)
    theVTab.UpdateSelection
  elseif(cpa_dChoice = EaB)then
    myqueryString6 = "[Direction] = " + dir4.quote
    theVTab.Query(myQueryString6, theBitmap1, #VTAB_SELTYPE_NEW)
    theVTab.UpdateSelection
  elseif(cpa_dChoice = InB)then
    myqueryString7 = "[Direction] = " + dir5.quote
    theVTab.Query(myQueryString7, theBitmap1, #VTAB_SELTYPE_NEW)
    theVTab.UpdateSelection
  elseif(cpa_dChoice = OtB)then
    myqueryString8 = "[Direction] = " + dir6.quote
    theVTab.Query(myQueryString8, theBitmap1, #VTAB_SELTYPE_NEW)
    theVTab.UpdateSelection
  else
    msgbox.Info ("Nothing", "")
  end

 'query schedule table for the time
 myQueryString = "[Time] >=" + lesstime.quote
 myQueryString2 = "[Time] <=" + moretime.quote
 theVTab.Query(myQueryString, theBitmap3, #VTAB_SELTYPE_NEW)
 theVTab.UpdateSelection
 theVTab.Query(myQueryString2, theBitmap3, #VTAB_SELTYPE_AND)
 theVTab.UpdateSelection

 'the schedule tables are permanently linked by me before
 'the program starts, but this is the code that is used

 "link to the bus stop theme
' theField = theVTab.FindField("Bus_stop")  'from dbf
' theTheme = theView.FindTheme("Bus Stops")  'fTheme
' aVTab = theTheme.GetFTab
' aField = aVTab.FindField("Bus_stop")
' theVTab.Link(theField, aVtab, aField)


else
 msgbox.Info ("Nothing", "")

end


'get origin user radio button selection from control panel
```

```
cpa3 = SELF.GetDialog.FindByName("origin_cpa")

if (cpa3 = nil) then
 MsgBox.Info("Can't find dialog", "")
 SELF.GetDialog.Close
end

cpa3Choice = cpa3.GetSelected

if (cpa3Choice = nil) then
 MsgBox.Info("No origin selected", "")
 SELF.GetDialog.Close
end

'get destination user radio button selection from control panel
cpa4 = SELF.GetDialog.FindByName("destination_cpa")

if (cpa4 = nil) then
 MsgBox.Info("Can't find dialog", "")
 SELF.GetDialog.Close
end

cpa4Choice = cpa4.GetSelected

if (cpa4Choice = nil) then
 MsgBox.Info("No destination selected", "")
 SELF.GetDialog.Close
end

'have the destination dialog box appear before the origin so that the origin
'can on top and active first

'if address selected go to dialog AddressSelection2—run View.Locate2
'if landmark selected go to dialog Landmark2
'if Point_Click selected go to dialog SetEnd—go to view


addressd = SELF.GetDialog.FindByName("addressd_rad")
landmarkd = SELF.GetDialog.FindByName("landmarkd_rad")
pcd = SELF.GetDialog.FindByName("PointClickd_rad")

if (cpa4Choice = addressd) then
 'av.run("View.Locate2", "")
 av.FindDialog("AddressSelection2").Open
 'av.run("AddressSelection2", "")
 'exit
elseif (cpa4Choice = landmarkd) then
 av.FindDialog("Landmark2").Open
 'av.run("Landmark", "")
 'exit
elseif (cpa4Choice = pcd) then
 SELF.GetDialog.Close

 theProject = av.GetProject
 theView = theProject.FindDoc("Route Finding")
 theThemeList=theView.getThemes

 for each i in (0..(theThemeList.count-1))
  temp=theThemeList.get(i).setActive(FALSE)
 end

 theTheme = theView.FindTheme("Transit routes.shp")
 theTheme.SetActive(TRUE)
 theTheme2 = theView.FindTheme("Bus Stops")
 theTheme2.SetActive(TRUE)

 'myAnswer = MsgBox.YesNo( "Select by Point and Click?", "Point Selection", TRUE )

 'exit
```

```
else
  cps4Choice = nil

end


'if address selected go to dialog AddressSelection---run View.Locate
'if landmark selected go to dialog Landmark
'if Point_Click selected go to dialog SetStart---go to view

addresso = SELF.GetDialog.FindByName("addresso_rad")
landmarko = SELF.GetDialog.FindByName("landmarko_rad")
pco = SELF.GetDialog.FindByName("PointClicko_rad")

if (cpa3Choice = addresso) then
  'av.run("View.Locate", "")
  av.FindDialog("AddressSelection").Open
  'av.run("AddressSelection", "")
  'exit

elseif (cpa3Choice = landmarko) then
  av.FindDialog("Landmark").Open
  'av.run("Landmark", "")
  'exit

elseif (cpa3Choice = pco) then
 SELF.GetDialog.Close

 'once the dialog is closed for point and click, make the view active
 theProject = av.GetProject
 theView = theProject.FindDoc("Route Finding")
 theThemeList=theView.getThemes

 for each i in (0..(theThemeList.count-1))
   temp=theThemeList.get(i).setActive(FALSE)
 end

 theTheme = theView.FindTheme("Transit routes.shp")
 theTheme.SetActive(TRUE)
 theTheme2 = theView.FindTheme("Bus Stops")
 theTheme2.SetActive(TRUE)

 'exit

'runs after the user interface script closes for point and click--show some instructions

message = "To select your start and end locations, please press the button with the black dot."

MsgBox.Info(message, "Point & Click")

message2 = "When you have selected your locations, please press the button with the blue diamond."

MsgBox.Info(message2, "Point & Click")

message3 = "If you need to zoom in, zoom out, or pan, please select the buttons below the blue diamond button."

MsgBox.Info(message3, "Point & Click")

else
  MsgBox.Info("Nothing", "")

end


SELF.GetDialog.Close
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        userinterface.Open
' Script Description: Opens the dialog and initializes the events.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    24-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'

'open the dialog box
theDialog = av.FindDialog("userinterface")

'move the dialog out of view but not close it right away
theDialog.Move(0, 0)


'clear textlines
'declare and update all controls in first step

a = "12"
b = "30"
timehour_txl = theDialog.FindByName("timehour_txl")
timehour_txl.SetText(a)
timehour_txl.Update

timeminute_txl = theDialog.FindByName("timeminute_txl")
timeminute_txl.SetText(b)
timeminute_txl.Update

am_rad = theDialog.FindByName("am_rad")
'am_rad.Update

pm_rad = theDialog.FindByName("pm_rad")
'pm_rad.Update

time_cpa = theDialog.FindByName("time_cpa")
'time_cpa.Update

weekday_rad = theDialog.FindByName("weekday_rad")
'weekday_rad.Update

saturday_rad = theDialog.FindByName("saturday_rad")
'saturday_rad.Update

sunday_rad = theDialog.FindByName("sunday_rad")
'sunday_rad.Update

week_cpa = theDialog.FindByName("week_cpa")
'week_cpa.Update

NB_rad = theDialog.FindByName("NB_rad")
'NB_rad.Update

SB_rad = theDialog.FindByName("SB_rad")
'SB_rad.Update

WB_rad = theDialog.FindByName("WB_rad")
'WB_rad.Update

EB_rad = theDialog.FindByName("EB_rad")
'EB_rad.Update

IB_rad = theDialog.FindByName("IB_rad")
```

```
'IB_rad.Update

OB_rad = theDialog.FindByName("OB_rad")
'OB_rad.Update

Direction_cpa = theDialog.FindByName("Direction_cpa")
'Direction_cpa.Update

addresso_rad = theDialog.FindByName("addresso_rad")
'addresso_rad.Update

landmarko_rad = theDialog.FindByName("landmarko_rad")
'landmarko_rad.Update

pointclicko_rad = theDialog.FindByName("PointClicko_rad")
'pointclicko_rad.Update

origin_cpa = theDialog.FindByName("origin_cpa")
'origin_cpa.Update

addressd_rad = theDialog.FindByName("addressd_rad")
'addressd_rad.Update

landmarkd_rad = theDialog.FindByName("landmarkd_rad")
'landmarkd_rad.Update

pointclickd_rad = theDialog.FindByName("PointClickd_rad")
'pointclickd_rad.Update

destination_cpa = theDialog.FindByName("destination_cpa")
'destination_cpa.Update

select_lbt = theDialog.FindByName("select_lbt")
'select_lbt.Update

cancel_lbt = theDialog.FindByName("reset_lbt")
'cancel_lbt.Update

'Set listeners
timehour_txl.SetListeners({select_lbt})
timeminute_txl.SetListeners({select_lbt})
time_cpa.SetListeners({select_lbt})
week_cpa.SetListeners({select_lbt})
Direction_cpa.SetListeners({select_lbt})
origin_cpa.SetListeners({select_lbt})
destination_cpa.SetListeners({select_lbt})


'make one radio button active as a default

am_rad = theDialog.FindByName("am_rad")
am_rad.Select'(true)

weekday = theDialog.FindByName("weekday_rad")
weekday.Select'(true)

NB_rad = theDialog.FindByName("NB_rad")
NB_rad.Select'(true)

addresso = theDialog.FindByName("addresso_rad")
addresso.Select'(true)

addressd = theDialog.FindByName("addressd_rad")
addressd.Select'(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       userinterface.cancel_lbt.Update
' Script Description: Enables the cancel button, sets it active.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    Jan-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always available

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        userinterface.cancel_lbt.Click
' Script Description: Closes the dialog when the cancel button is
'                  selected.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'close the dialog box

SELF.GetDialog.Close
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Geocode.Table
' Script Description: Geocodes addresses found in a table to a
'                  matchable theme. It is assumed that the
'                  addresses and theme use US style addresses.
'                  The script can be modified to support other
'                  address styles.  Geocoding results are
'                  reported and the new theme is added to the
'                  current View and drawn.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author       Date          Details
'
' ESRI         27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


' Collect information used to create the the new geocoded point theme.
' The theme to match against, the table containing addresses, and the
' name of the field containing address information are passed from the
' calling script...

'get the match source

theView     = av.GetProject.FindDoc("Route Finding")
aMatchSource = theView.FindTheme("Ottawa Street Network").GetMatchSource

' Get the address event table.

addressTable = av.GetProject.FindDoc( "Attributes of Location.shp" ).GetVTab
addressField = addressTable.FindField("Address")


addressFTab  = addressTable '.GetFTab

' Get the matchable feature source and double check that it is valid...

'if (aMatchSource = Nil) then
'  MsgBox.Error("Theme"++matchTheme.GetName++"is not matchable.","")
'  exit
'end


' Open or create index to optimize performance...

aMatchSource.OpenIndex



' Specify the output point shapefile that will be created from the
' matched addresses...

aGeoName = GeoName.Make( aMatchSource, addressFTab,
  addressField, addressField )

fnOutFile = "F:\thesis data\Chosen Locations".AsFileName

'fnOutFile = FileDialog.Put(av.GetProject.MakeFileName("Chosen Locations", "shp"),
'  "*.shp","Output Shapefile")
if (fnOutFile = nil) then
  exit
else
  aGeoName.SetFileName(fnOutFile)
end
```

```
' Create a match key based on the standardization rules for the
' match source...

aMatchKey  = MatchKey.Make(aMatchSource.GetStanRules)


' Create a new match case.  A match case is comprised of a list of
' candidate records and information describing how well the candidates
' match the key.  The MatchCase will be populated with candidates later...

aMatchCase = MatchCase.Make( aMatchSource, aMatchKey )


' Create a new match preference which will be used to access various
' geocoding preferences, such as spelling weight, minimum acceptable
' score, etc...

aMatchPref = MatchPref.Make


' Create a new theme feature table.  For every address record there
' will be a record in the FTab.  These are currently unmatched, i.e.
' the shape field is empty and the status is 'U' for unmatched.  We
' must match an address to populate the shape field and toggle the status
' to 'M' for matched, something we do in the next step...

aGeoTheme = aMatchSource.InitGeoTheme( aGeoName )


' Populate the feature table by matching addresses to the matchable
' Theme.  This places a point in the FTab's shape field for and sets
' the status to M (matched) for each match...

numrecs    = addressFTab.GetNumRecords
numMatched = 0

av.ShowMsg("Matching Addresses...")

for each i in addressFTab.GetDefBitMap
  av.SetStatus((i / numrecs) * 100)

  ' Get an address...
  aMatchKey.SetKey( addressFTab.ReturnValueString( addressField, i ))

  ' Find candidates for the address...
  numCand = aMatchSource.Search( aMatchKey, 70, aMatchCase)


  ' If there are no candidates, continue on to the next addresss.  This
  ' will be an unmatched record.  If candidates are found take the best
  ' candidate and see if it exceeds the minimum specified match score.
  ' If it does write it, otherwise skip it - it remains unmatched....

  if (numCand = 0) then   ' No candidates - skip to next address...
    continue

  else                    ' We have at least on candidate...

    aMatchCase.ScoreCandidates
    cand      = aMatchCase.GetBestCand
    candScore = cand.GetScore
    minScore  = aMatchPref.GetPrefVal( #MATCHPREF_MINMATCHSCORE )

    ' If the min required match score is exceeded - write it!
    ' If the min is not met then the cand is not written and the record
    ' will remain unmatched...

    if ( candScore >= minScore ) then
      aMatchSource.WriteMatch( i, aMatchKey, cand )
```

```
    numMatched = numMatched + 1
   end

  end
end

av.ClearMsg
av.ClearStatus
aMatchSource.EndMatch


" Report the results of the geocoding...
'
'MsgBox.Info("Total records processed:"++numrecs.AsString+NL+
'  "Total addresses matched:"++numMatched.AsString, "Geocoding Results")
'

' Add the new theme to the view and draw it...

newTheme = Theme.Make(aGeoName)
theView.AddTheme(newTheme)
newTheme.SetVisible(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Geocode
' Script Description: Runs the Geocode.Table script.
'
' Programmer:         Sarah Riley
' Date Created:       Apr. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    29-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

av.run("Geocode.Table","")
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        AddressSelection2.Update
' Script Description:
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author        Date        Details
'
' S. Riley      Jan-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always enabled

SELF.SetEnabled(true)
```

```
""""""""""""""""""""""""""""""""""""""""""""""""''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      AddressSelection2.Open
' Script Description: Opens the dialog and initializes the objects
'                within the dialog.
'
' Programmer:       Sarah Riley
' Date Created:     Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    9-Apr-2002    Initial Implementation
'
""""""""""""""""""""""""""""""""""""""""""""""""''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'open the dialog box
theDialog = av.FindDialog("AddressSelection2")

'declare and update all controls in the dialog

destinationaddress_txl = theDialog.FindByName("destinationaddress_txl")
'destinationaddress_txl.Update

ok2_lbt = theDialog.FindByName("ok_lbt")
'ok2_lbt.Update


'This script will run when the dialog is opened
'SELF refers to the dialog
'find the project and table Addresses and set editable
'for input from dialog

theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Location.shp")
'theTheme.SetActive(true)
'theView.SetEditableTheme("Location.shp")
theTable = theTheme.GetFTab
fldAddress = theTable.FindField("Address")
'theTable.SetActive(true)
theTable.SetEditable(true)

'theTheme.StopEditing(TRUE)


"delete contents first
"Table.SelectAll
'theSelection = theTable.GetSelection
'theTable.GetFTab.GetSelection.SetAll
'theTable.GetFTab.UpdateSelection
'theTable.GetSelection
'av.GetProject.SetModified(true)

theSelection = theTable.GetSelection
theSelection.SetAll

SC = theSelection.Count

if (SC > 1) then

 'Table.DeleteRecords
 'if there is more than 1 record then delete, if not
 'just make the table ready for adding records

 'theTable = av.GetActiveDoc
 theVTab = theTable '.GetFTab
 theVtab.BeginTransaction
 theVTab.RemoveRecords(theVTab.GetSelection.Clone)
 theVTab.EndTransaction
```

```
 "Table.DeleteRecordsUpdate
' theVTab = av.GetActiveDoc.GetVTab
' SELF.SetEnabled(theVTab.CanRemoveRecords and (theVTab.GetSelection.Count > 0))

else
 'theTable = av.GetActiveDoc.Get.VTab
 fldAddress = theTable.FindField("Address")
 SELF.SetServer(theTable)
end

'clears out the textline
thetextline = theDialog.FindByName("destinationaddress_txt")
thetextline.Empty
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        AddressSelection2.ok_lbt.Update
' Script Description: Enables the OK button to always be available.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always enabled

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      AddressSelection2.ok_lbt.Click
' Script Description: Runs when the OK button is selected.  Enters
'                   information into a table.
'
' Programmer:       Sarah Riley
' Date Created:     Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    12-Apr-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'SELF refers to the button

SELF.SetEnabled(true)

'gets the address entered by user
address_txl = SELF.GetDialog.FindByName("destinationaddress_txl")

'reference the attributes of address.shp vtab---table has been made
theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Location.shp")
'theView.SetEditableTheme("Location.shp")
theTable = theTheme.GetFTab
'theTable = SELF.GetServer
fldAddress = theTable.FindField("Address")

'add the new record
theTable.SetEditable(true)
recNewAddress = theTable.AddRecord
theTable.SetValue (fldAddress, recNewAddress, address_txl.GetText)
theTable.SetEditable(false)
theTable.Refresh

'close the dialog box
self.GetDialog.Close

'geocode points
av.run("Geocode", "")

'have user select red button to find route
message = "To find your root, please select the red diamond button."

MsgBox.Info(message, "Address Selection")
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        AddressSelection.Update
' Script Description: Enables and updates the dialog
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    Jan-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always enabled

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       AddressSelection.Open
' Script Description: Opens the dialog and initializes the objects
'                   within the dialog.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley   9-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'open the dialog box
theDialog = av.FindDialog("AddressSelection")

'declare and update all controls in the dialog

address_txl = theDialog.FindByName("address_txl")
'address_txl.Update

ok_lbt = theDialog.FindByName("ok_lbt")
'ok_lbt.Update


'This script will run when the dialog is opened
'SELF refers to the dialog
'find the project and table Addresses and set editable
'for input from dialog

theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Location.shp")
'theTheme.SetActive(true)
'theView.SetEditableTheme("Location.shp")
theTable = theTheme.GetFTab
fldAddress = theTable.FindField("Address")
'theTable.SetActive(true)
theTable.SetEditable(true)


'theTheme.StopEditing(TRUE)


'delete contents first
'Table.SelectAll
'for every record i delete

'theSelection = theTable.GetSelection
'theTable.GetFTab.GetSelection.SetAll
'theTable.GetFTab.UpdateSelection
'theTable.GetSelection
'av.GetProject.SetModified(true)

theSelection = theTable.GetSelection
theSelection.SetAll
SC = theSelection.Count

if (SC > 1) then

 'Table.DeleteRecords
 'if there is more than 1 record then delete, if not
 'just make the table ready for adding records

 'theTable = av.GetActiveDoc
 theVTab = theTable '.GetFTab
 theVtab.BeginTransaction
```

```
  theVTab.RemoveRecords(theVTab.GetSelection.Clone)
  theVTab.EndTransaction
  '
'  'Table.DeleteRecordsUpdate
'  theVTab = av.GetActiveDoc.GetVTab
'  SELF.SetEnabled(theVTab.CanRemoveRecords and (theVTab.GetSelection.Count > 0))

else
  'theTable = av.GetActiveDoc.Get.VTab
  fldAddress = theTable.FindField("Address")
  SELF.SetServer(theTable)
end

'clears out the textline
thetextline = theDialog.FindByName("address_txt")
thetextline.Empty
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       AddressSelection.OK_lbt.Update
' Script Description: Enables the OK button to always be available.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'always enabled, always true

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       AddressSelection.OK_lbt.Click
' Script Description: Runs when the OK button is selected.  Gets
'                  the information entered by user and inputs
'                  it into a table.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author       Date         Details
'
' S. Riley   9-Apr-2002      Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'SELF refers to the button

SELF.SetEnabled(true)

address_txl = SELF.GetDialog.FindByName("address_txl")

'reference the attributes of addresses.shp vtab--table has been made
theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Location.shp")
'theView.SetEditableTheme("Location.shp")
theTable = theTheme.GetFTab
'theTable = SELF.GetServer
fldAddress = theTable.FindField("Address")

'add the new record
theTable.SetEditable(true)
recNewAddress = theTable.AddRecord
theTable.SetValue (fldAddress, recNewAddress, address_txl.GetText)
theTable.SetEditable(false)
theTable.Refresh


'close the dialog box
self.GetDialog.Close

'stop editing the theme
'theTheme.StopEditing(TRUE)
'end
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark2.Update
' Script Description: Enables the dialog.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always enabled, always true

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark2.Themes_cbx.Update
' Script Description: Enables the themes in the view to be
'                    listed in the combo box.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    27-Mar-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'Get the view
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")

'find the document in the project
theDoc = av.GetProject.FindDoc("Route Finding")

'bring the doc to the front
if (theDoc = nil) then
 av.GetProject.GetWin.Open
elseif (theDoc.GetWin.IsOpen) then
 theDoc.GetWin.Open
end

theDoc = SELF.GetDialog.GetActiveDoc
if(theDoc.Is(View) and theDoc.GetThemes.IsEmpty.Not) then
 SELF.SetEnabled(true)
 theThemes = theDoc.GetThemes
 SELF.DefineFromList(theThemes)
 SELF.GetDialog.SetServer(theThemes.Get(0).GetFTab)
else
 MsgBox.Info( "Error", " " )
 SELF.SetEnabled(false)
end
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark2.Themes_cbx.Select
' Script Description: Sends the table information corresponding
'                     to the selected theme to the listbox.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'display current themes in the view
theTheme = SELF.GetCurrentValue
theFTab = theTheme.GetFTab
SELF.GetDialog.SetServer(theFTab)
SELF.BroadcastUpdate
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark2.ServerSelChanged
' Script Description: Runs when the selection in the VTab changes.
'                    The listbox updates itself to reflect the
'                    new selections in the combo box.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author     Date          Details
'
' S. Riley   27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

SELF.FindByName("Landmark_lbx").Update
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark2.Open
' Script Description: Opens and initializes the items within the dialog.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'open the dialog box
theDialog = av.FindDialog("Landmark2")

'center the dialog
theDialog.Move(0, 0)

'declare and update all controls
ok_lbt = theDialog.FindByName("OK_lbt")
ok_lbt.Update

cancel_lbt = theDialog.FindByName("cancel_lbt")
cancel_lbt.Update

themes_cbx = theDialog.FindByName("Themes_cbx")
themes_cbx.Update

landmark_lbx = theDialog.FindByName("Landmark_lbx")
landmark_lbx.Update

themes_cbx.SetListeners({landmark_lbx})
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark2.OK_lbt.Update
' Script Description: Enables the OK button. Sets it active.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always enabled, always true

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark2.OK_lbt.Click
' Script Description: Script runs when the OK button is selected.
'                  Gets the information entered by the user and
'                  produces a point theme to be use in finding
'                  a route.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author     Date        Details
'
' S. Riley   9-Apr-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'self refers to the dialog

viewname = "Route Finding"

theView = av.GetProject.FindDoc(viewname)
if (theView = nil) then
  MsgBox.Info("Can't find view: " + viewname.AsString, "")
  self.GetDialog.Close
  exit
end

'gets the theme selected by user
cbx_themes = self.GetDialog.FindByName("Themes_cbx")
userTheme = cbx_themes.GetCurrentValue
theTheme = userTheme

'need to activate the theme and its table
theView = av.FindDoc("Route Finding")
theThemeList=theView.GetThemes
for each i in (0..(theThemeList.count-1))
   temp=theThemeList.get(i).setActive(FALSE)
end

theTheme.setActive(TRUE)


theFTab = theTheme.GetFTab
if (theFTab = nil) then
  MsgBox.Info("No FTab", "")
  self.GetDialog.Close
  exit
end

theSelection = theFTab.GetSelection

'gets the record selected by user
lbx_record = self.GetDialog.FindByName("Landmark_lbx")
aRecord = lbx_record.GetCurrentValue

theSelection = theFTab.GetSelection

'select the choice in the table
myQueryString = "[Name] = " + aRecord.quote
theFTab.Query (myQueryString, theSelection, #VTAB_SELTYPE_AND)
theFTab.UpdateSelection

mytmpBitMap = theFTab.GetSelection.Count

if (mytmpBitMap = 0) then
  MsgBox.Info("No records found", "")
  self.GetDialog.Close
```

```
  exit
end


'The following makes the location list point theme by merging the start
'and the end locations.
'This theme is used in the network analyst routine.

theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")

themesToMerge = List.Make
themesToMerge.add(theView.FindTheme("firstlocation"))
'themesToMerge.add(theView.FindTheme("Location.shp"))
themesToMerge.add(theView.FindTheme(theTheme.AsString))   'theme corresponding to location selected by user
'themesToMerge.add(theView.FindTheme("firstlocation"))


' Themes must have matching shape types for merging.  Using the first
' active theme verify that this is the case...

checkType  = themesToMerge.Get(0).GetFtab.FindField("Shape").GetType

for each i in 1 .. (themesToMerge.Count - 1)
  t = themesToMerge.Get(i)
  if (checkType <> t.GetFTab.FindField("Shape").GetType) then
    MsgBox.Error("Theme feature type mismatch - Unable to merge.","")
    exit
  end
end


' Specify the output shapefile...

outFName = av.GetProject.MakeFileName("Chosen Locations", "shp")
'theCWD=FileName.GetCWD
'outFName=theCWD.makeTmp("Chosen Locations","shp")

' Create the list of fields used for the output theme.  The fields
' are taken from the first active theme only, it is assumed that
' other themes have an identical set of fields.  If this is not the
' case the themes will still be merged, however fields not found in
' other themes will be empty...

fieldList = List.Make

for each f in themesToMerge.Get(0).GetFTab.GetFields
  if (f.GetName = "Shape") then
    continue
  else
    fCopy = f.Clone
    fieldList.Add(fCopy)
  end
end


' Get the class of new FTab to create, create the new FTab and
' add fields that we've gathered from the input themes....

shapeType = themesToMerge.Get(0).GetFTab.FindField("Shape").GetType
outClass = POINT

mergeFTab = FTab.MakeNew( outFName, outClass )
if (fieldList.Count > 0) then
  mergeFTab.AddFields( fieldList )
end


' Populate the new FTab from the FTabs of the input themes...
```

```
for each t in themesToMerge
  av.ShowMsg( "Merging"++t.GetName )
  inFTab  = t.GetFTab
  if (inFTab.GetSelection.Count = 0) then
    theRecordsToMerge = inFTab
    numRecs = inFTab.GetNumRecords
  else
    theRecordsToMerge = inFTab.GetSelection
    numRecs = theRecordsToMerge.Count
  end

  for each rec in theRecordsToMerge
    av.SetStatus( (rec / numRecs) * 100 )
    newRec   = mergeFTab.AddRecord
    inField  = inFTab.FindField( "Shape" )
    outField = mergeFTab.FindField( "Shape" )
    mergeFTab.SetValue( outField, newrec, inFTab.ReturnValue( inField, rec ))

    if (fieldList.Count > 0) then
      for each f in fieldList
        fName    = f.GetName
        inField  = inFTab.FindField( fName )

        ' Skip field if not found in inFTab...
        if ( inField <> Nil ) then
          outField = mergeFTab.FindField( fName )
          aValue   = inFTab.ReturnValue( inField, rec )
          mergeFTab.SetValue( outField, newRec, aValue )
        end
      end  ' for each f
    end  ' if count
  end  ' for each rec
end  ' for each t

av.ClearMsg
av.ClearStatus

' add the new theme to...

mergeTheme = FTheme.Make( mergeFTab )
theView.AddTheme( mergeTheme )
mergeTheme.SetName("Chosen Locations")
mergeTheme.SetVisible(True)


'close the dialog box
self.GetDialog.Close


message = "To find  your root, please select the red diamond button."

MsgBox.Info(message, "Landmark")
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark2.Landmark_lbx.Update
' Script Description: Receives the theme selected from the combo
'                  box and gets table information.  Shows the
'                  appropriate field information for the user
'                  to select from.
'
' Programmer:       Sarah Riley
' Date Created:     Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley     27-Mar-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'find the document in the project
theDoc = av.GetProject.FindDoc("Route Finding")

'bring the doc to the front
if (theDoc = nil) then
 av.GetProject.GetWin.Open
elseif (theDoc.GetWin.IsOpen) then
 theDoc.GetWin.Open
end

theDoc = SELF.GetDialog.GetActiveDoc
if(theDoc.Is(View) and theDoc.GetThemes.IsEmpty.Not) then
 SELF.SetEnabled(true)
 theVTab = SELF.GetDialog.GetServer
 Field1 = theVTab.FindField("Name")
 flist = {Field1}
 numFields = flist.Count
 SELF.DefineFromVTab(theVTab, flist, false)
 SELF.FitColumns(0..(numFields - 1), false)
else
 MsgBox.Info( "Error", " " )
 SELF.SetEnabled(false)
end
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark2.DocActivate
' Script Description: Runs when the user makes a new document active.
'                     Calls the update scripts for the combo box
'                     and the listbox.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


SELF.FindByName("Themes_cbx").Update
SELF.FindByName("Landmark_lbx").Update
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark2.cancel_lbt.Update
' Script Description: Enables the cancel button to always be available.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author     Date          Details
'
' S. Riley   27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always available

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark2.cancel_lbt.Click
' Script Description: Closes the dialog when the cancel button is
'                  selected
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    27-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'close the dialog box

SELF.GetDialog.Close
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark.Update
' Script Description: Enables dialog.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    Jan-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'always enabled, always true

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark.Themes_cbx.Update
' Script Description: Shows a list of themes found in the view.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley    19-Mar-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'

'get the view a list of themes present
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")

'find the document in the project
theDoc = av.GetProject.FindDoc("Route Finding")

'bring the doc to the front
if (theDoc = nil) then
  av.GetProject.GetWin.Open
elseif (theDoc.GetWin.IsOpen) then
  theDoc.GetWin.Open
end

theDoc = SELF.GetDialog.GetActiveDoc
if(theDoc.Is(View) and theDoc.GetThemes.IsEmpty.Not) then
  SELF.SetEnabled(true)
  theThemes = theDoc.GetThemes
  SELF.DefineFromList(theThemes)
  SELF.GetDialog.SetServer(theThemes.Get(0).GetFTab)
else
  MsgBox.Info( "Error", " " )
  SELF.SetEnabled(false)
end
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      Landmark.Themes_cbx.Select
' Script Description: Broadcasts the selected theme's table
'                   information to the listbox.
'
' Programmer:       Sarah Riley
' Date Created:     Jan. 2002
' Returned Object:  none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley    19-Mar-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'display current themes in the view
theTheme = SELF.GetCurrentValue
theFTab = theTheme.GetFTab
SELF.GetDialog.SetServer(theFTab)
SELF.BroadcastUpdate
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark.ServerSelChanged
' Script Description: Runs when the selection in the VTab changes.
'                     The listbox updates itself to reflect the
'                     new selections in the combo box.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author     Date      Details
'
' S. Riley   Jan-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'

SELF.FindByName("Landmark_lbx").Update
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark.Open
' Script Description: Opens the dialog and initializes the objects.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley    22-Mar-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'open the dialog box
theDialog = av.FindDialog("Landmark")

'center the dialog
theDialog.Move(0, 0)

'declare and update all controls
ok_lbt = theDialog.FindByName("OK_lbt")
ok_lbt.Update

cancel_lbt = theDialog.FindByName("cancel_lbt")
cancel_lbt.Update

themes_cbx = theDialog.FindByName("Themes_cbx")
themes_cbx.Update

landmark_lbx = theDialog.FindByName("Landmark_lbx")
landmark_lbx.Update

themes_cbx.SetListeners({landmark_lbx})
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark.OK_lbt.Update
' Script Description: Enables the OK button to always be available.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author     Date       Details
'
' S. Riley   Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always enabled, always true

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark.OK_lbt.Click
' Script Description: Script runs when the OK button is selected.
'                     Gets the information entered by the user and
'                     produces a point theme to be use in finding
'                     a route.
'
' Programmer:         Sarah Riley
' Date Created:       Jan. 2002
' Returned Object:    none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    9-Apr-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'self refers to the dialog

viewname = "Route Finding"

theView = av.GetProject.FindDoc(viewname)
if (theView = nil) then
  MsgBox.Info("Can't find view: " + viewname.AsString, "")
  self.GetDialog.Close
  exit
end

'gets the theme selected by user
cbx_themes = self.GetDialog.FindByName("Themes_cbx")
userTheme = cbx_themes.GetCurrentValue
theTheme = userTheme

'need to activate the theme and its table
theView = av.FindDoc("Route Finding")
theThemeList=theView.GetThemes
for each i in (0..(theThemeList.count-1))
   temp=theThemeList.get(i).setActive(FALSE)
end

theTheme.setActive(TRUE)


theFTab = theTheme.GetFTab
if (theFTab = nil) then
  MsgBox.Info("No FTab", "")
  self.GetDialog.Close
  exit
end

theSelection = theFTab.GetSelection

'gets the record selected by user
lbx_record = self.GetDialog.FindByName("Landmark_lbx")
aRecord = lbx_record.GetCurrentValue

theSelection = theFTab.GetSelection

'select the choice in the table
myQueryString = "[Name] = " + aRecord.quote
theFTab.Query (myQueryString, theSelection, #VTAB_SELTYPE_AND)
theFTab.UpdateSelection

mytmpBitMap = theFTab.GetSelection.Count

if (mytmpBitMap = 0) then
  MsgBox.Info("No records found", "")
  self.GetDialog.Close
```

```
  exit
end

'The following makes the location list point theme by merging the start
'and the end locations.
'This theme is used in the network analyst routine.

'get the view and the necessary themes
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")

theFTheme = theView.FindTheme("Location.shp")
theTable = theFTheme.GetFTab
fldAddress = theTable.FindField("Address")
'theTable.SetActive(true)
theTable.SetEditable(true)

theSelection = theTable.GetSelection
theSelection.SetAll
SC = theSelection.Count

if (SC >= 2) then

  'Table.DeleteRecords
  'if there is more than 1 record then delete them
  'if not just make the table ready for adding records

  'theTable = av.GetActiveDoc
  theVTab = theTable  '.GetFTab
  theVtab.BeginTransaction
  theVTab.RemoveRecords(theVTab.GetSelection.Clone)
  theVTab.EndTransaction

else
  'theTable = av.GetActiveDoc.GetVTab
  fldAddress = theTable.FindField("Address")
  'SELF.SetServer(theTable)
end

themesToMerge = List.Make
themesToMerge.add(theView.FindTheme("Location.shp"))
themesToMerge.add(theView.FindTheme(theTheme.AsString))    'theme corresponding to location selected by user
'themesToMerge.add(theView.FindTheme("Location.shp"))
'MsgBox.List(themesToMerge, "", "")

' Themes must have matching shape types for merging.  Using the first
' active theme verify that this is the case...

checkType  = themesToMerge.Get(0).GetFtab.FindField("Shape").GetType

for each i in 1 .. (themesToMerge.Count - 1)
  t = themesToMerge.Get(i)
  if (checkType <> t.GetFTab.FindField("Shape").GetType) then
    MsgBox.Error("Theme feature type mismatch - Unable to merge.","")
    exit
  end
end


' Specify the output shapefile

outFName = av.GetProject.MakeFileName("firstlocation", "shp")
'theCWD=FileName.GetCWD
'outFName=theCWD.makeTmp("firstlocation","shp")

' Create the list of fields used for the output theme.  The fields
' are taken from the first active theme only, it is assumed that
' other themes have an identical set of fields.  If this is not the
' case the themes will still be merged, however fields not found in
' other themes will be empty.
```

```
fieldList = List.Make

for each f in themesToMerge.Get(0).GetFTab.GetFields
  if (f.GetName = "Shape") then
    continue
  else
    fCopy = f.Clone
    fieldList.Add(fCopy)
  end
end

' Get the class of new FTab to create, create the new FTab and
' add fields that we've gathered from the input themes....

shapeType = themesToMerge.Get(0).GetFTab.FindField("Shape").GetType
outClass = POINT

mergeFTab = FTab.MakeNew( outFName, outClass )
if (fieldList.Count > 0) then
  mergeFTab.AddFields( fieldList )
end

' Populate the new FTab from the FTabs of the input themes...

for each t in themesToMerge
  av.ShowMsg( "Merging"++t.GetName )
  inFTab  = t.GetFTab
  if (inFTab.GetSelection.Count = 0) then
    theRecordsToMerge = inFTab
    numRecs = inFTab.GetNumRecords
  else
    theRecordsToMerge = inFTab.GetSelection
    numRecs = theRecordsToMerge.Count
  end

  for each rec in theRecordsToMerge
    av.SetStatus( (rec / numRecs) * 100 )
    newRec   = mergeFTab.AddRecord
    inField  = inFTab.FindField( "Shape" )
    outField = mergeFTab.FindField( "Shape" )
    mergeFTab.SetValue( outField, newrec, inFTab.ReturnValue( inField, rec ))

    if (fieldList.Count > 0) then
      for each f in fieldList
        fName    = f.GetName
        inField  = inFTab.FindField( fName )

        ' Skip field if not found in inFTab...
        if ( inField <> Nil ) then
          outField = mergeFTab.FindField( fName )
          aValue   = inFTab.ReturnValue( inField, rec )
          mergeFTab.SetValue( outField, newRec, aValue )
        end
      end  ' for each f
    end  ' if count
  end  ' for each rec
end  ' for each t

av.ClearMsg
av.ClearStatus

' add the new theme to the view
mergeTheme = FTheme.Make( mergeFTab )
theView.AddTheme( mergeTheme )
mergeTheme.SetName("firstlocation")
mergeTheme.SetVisible(True)

'close the dialog box
self.GetDialog.Close
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      Landmark.Landmark_lbx.Update
' Script Description: Receives the theme selected from the combo
'                   box and gets table information.  Shows the
'                   appropriate field information for the user
'                   to select from.
'
' Programmer:       Sarah Riley
' Date Created:     Jan. 2002
' Returned Object:  none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley    22-Mar-2002  Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'find the document in the project
theDoc = av.GetProject.FindDoc("Route Finding")

'bring the doc to the front
if (theDoc = nil) then
  av.GetProject.GetWin.Open
elseif (theDoc.GetWin.IsOpen) then
  theDoc.GetWin.Open
end

theDoc = SELF.GetDialog.GetActiveDoc
if(theDoc.Is(View) and theDoc.GetThemes.IsEmpty.Not) then
  SELF.SetEnabled(true)
  theVTab = SELF.GetDialog.GetServer
  Field1 = theVTab.FindField("Name")
  flist = {Field1}
  numFields = flist.Count
  SELF.DefineFromVTab(theVTab, flist, false)
  SELF.FitColumns(0..(numFields - 1), false)
else
  MsgBox.Info( "Error", " " )
  SELF.SetEnabled(false)
end
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:        Landmark.DocActivate
' Script Description: Runs when the user makes a new document active.
'                  Calls the update scripts for the combo box
'                  and the listbox.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author       Date       Details
'
' S. Riley     Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


SELF.FindByName("Themes_cbx").Update
SELF.FindByName("Landmark_lbx").Update
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark.cancel_lbt.Update
' Script Description: Enables the cancel button to always be available.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date      Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'always true, always available

SELF.SetEnabled(true)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Landmark.cancel_lbt.Click
' Script Description: Closes the dialog when the cancel button is
'                    selected.
'
' Programmer:        Sarah Riley
' Date Created:      Jan. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date       Details
'
' S. Riley    Jan-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'close the dialog box

SELF.GetDialog.Close
```

```
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      Routing2
' Script Description: Finds an optimum route, closest bus stop,
'                   the time the bus arrives, then links to the
'                   script stats to work through Bayesian
'                   analogy to give results to user.
'                   This runs for Point and Click
'
' Programmer:       Sarah Riley
' Date Created:     Apr. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    2-May-2002   Initial Implementation
'
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'convert the graphics to a theme in order
'to solve the optimum route problem

av.run("Graphics Convert", "")

'runs the netrun script to solve best route

av.run("netrun2", "")

'runs the nearest features script to find closest
'bus stops to the locations chosen by user

av.run("nearfeatures2", "")

'runs the select by theme script

av.run("SelByTheme2", "")

'link bus stop table to the schedule table --- select from set and see how
'it works

theView = av.GetActiveDoc
thetable1 = av.FindDoc("weekday.dbf")
theVTab = thetable1.GetVTab

theBitmap1 = theVTab.GetSelection
selcount1 = theBitmap1.Count
if(selcount1 <> 0)then

  'theField = theVTab.FindField("Bus_stop")  'from dbf

  theTheme = theView.FindTheme("Bus Stops") 'fTheme
  anFTab = theTheme.GetFTab
  theBitmap2 = anFTab.GetSelection
  aField = anFTab.FindField("On_street")
  rec = theBitmap2.GetNextSet(-1)
  aField2 = anFTab.ReturnValue(aField, rec)
  'msgbox.info(aField2.AsString, "")
  myquerystring = "[On_street] = " + aField2.Quote
  theVTab.Query(myQueryString, theBitmap1, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection

else
  thetable2 = av.FindDoc("saturday.dbf")
  theVTab = thetable2.GetVTab

  theBitmap2 = theVTab.GetSelection
  selcount2 = theBitmap2.Count
  if(selcount2 <> 0)then
```

```
  'theField = theVTab.FindField("Bus_stop")  'from dbf

  theTheme = theView.FindTheme("Bus Stops") 'fTheme
  anFTab = theTheme.GetFTab
  theBitmap = anFTab.GetSelection
  aField = anFTab.FindField("On_street")
  rec = theBitmap.GetNextSet(-1)
  aField2 = anFTab.ReturnValue(aField, rec)
  myquerystring = "[On_street] = " + aField2.Quote
  theVTab.Query(myQueryString, theBitmap2, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection

  else
   thetable3 = av.FindDoc("sunday.dbf")
   theVTab = thetable3.GetVTab

   theBitmap3 = theVTab.GetSelection
   selcount3 = theBitmap3.Count
   if(selcount3 <> 0)then

    'theField = theVTab.FindField("Bus_stop")  'from dbf

    theTheme = theView.FindTheme("Bus Stops") 'fTheme
    anFTab = theTheme.GetFTab
    theBitmap2 = anFTab.GetSelection
    aField = anFTab.FindField("On_street")
    rec = theBitmap2.GetNextSet(-1)
    aField2 = anFTab.ReturnValue(aField, rec)
    myquerystring = "[On_street] = " + aField2.Quote
    theVTab.Query(myQueryString, theBitmap3, #VTAB_SELTYPE_AND)
    theVTab.UpdateSelection

   else
    'do nothing
   end
  end
end

'runs the merge script to find closest bus stop with a scheduled time

av.run("merge2", "")

'runs the time script to find the time difference between the origin
'bus stop and a bus stop with a schedule

av.run("time2", "")

'runs the stats script to solve the travel time and produce an output

av.run("stats2", "")
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      Graphics Convert
' Script Description: Converts selected graphic(s) to a shapefile
'
' Programmer:       Sarah Riley
' Date Created:     Apr. 2002
' Returned Object:   Adds the new Shapefile to the active view from
'                   which the Graphic(s) originated.
'
' Changes:
'
'Date          Details
'29-Apr-2002    Initial Implementation
'
'Author
'ESRI
'Isaac Oshima
'California Department of Fish and Game, Technical Service Branch
'February 11, 1998 (revised)
'ioshima@maphost.dfg.ca.gov
'916-323-1635
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


theProject = av.GetProject
theView = av.GetActiveDoc

'Graphics.SelectAll    Now we select all of the point graphics
theDoc = av.GetActiveDoc
theDoc.GetGraphics.SelectAll


theNewFtab = Nil

if (theView.Is(View).Not) then
  MsgBox.Error("Active Document is not a View","Active Document")
  exit
end


theGraphicsList = theView.GetGraphics
theSelectedGraphics = theGraphicsList.GetSelected

if (theSelectedGraphics.Count = 0) then
  MsgBox.Info("Please, select graphics to convert to shapefile","Select Graphics")
  exit
end

ftPoint = False
ftLine = False
ftPolygon = False
ftUnknown = False
ftKount = 0
ftList = List.Make
theClass = Nil

for each gs in theSelectedGraphics
  s = gs.GetShape
  if (s.GetDimension = 0) then
    ftPoint = True
    ftList.Add("Point")
  elseif (s.GetDimension = 1) then
    ftLine = True
    ftList.Add("Line")
  elseif (s.getDimension = 2) then
    ftPolygon = True
    ftList.Add("Polygon")
  else
    ftUnknown = True
    ftList.Add("Unknown")
```

```
    MsgBox.Info("Unknown feature option","See Isaac")
    exit
  end
end

ftList.RemoveDuplicates



if (ftList.Count <> 1) then
  MsgBox.Error("Feature Type Mismatches","Select Only One Feature Type: Point, Line or Polygon")
  Return Nil
end

theClass = ftList.Get(0).asString.Trim


'If a theme in the view is being edited, Stop Editing it before creating new theme
editThm = theView.GetEditableTheme
if (editThm <> nil) then
  doSave = MsgBox.YesNoCancel("Save edits to "+editThm.GetName+"?","Stop Editing",true)
  if (doSave = nil) then
    return nil
  end
  if (editThm.StopEditing(doSave).Not) then
      MsgBox.Info("Unable to Save Edits to "
                + editThm.GetName +
                ", please use the Save Edits As option", "")
      return nil
  else
      theView.SetEditableTheme(NIL)
  end
end

if (ftPoint = True) then
  class = Point

elseif (ftLine = True) then
  class = PolyLine

elseif (ftPolygon = True) then
  class = Polygon

else
  class = Nil
end



'def = av.GetProject.MakeFileName("Locate", "shp")
def = FileName.Make("F:\thesis data\Locate.shp")

'def = FileDialog.Put(def, "*.shp", "New Theme")

'if (def <> nil) then
  tbl = FTab.MakeNew(def, class)
   'if (tbl.HasError) then
   ' if (tbl.HasLockError) then
    '    MsgBox.Error("Unable to acquire Write Lock for file " + def.GetBaseName, "")
    'else
     '  MsgBox.Error("Unable to create " + def.GetBaseName, "")
    'end
    'return nil
  'end
  theIdField = Field.Make("ID", #FIELD_DECIMAL, 8, 0)

  theIdField.SetVisible( TRUE )
  tbl.AddFields({theIdField})
  tbl.SetEditable(False)
  theNewTheme = FTheme.Make(tbl)
```

```
  theNewFtab = theNewTheme.GetFtab
  theNewShapeField = theNewFtab.FindField("Shape")
  theView.AddTheme(theNewTheme)
  theNewTheme.SetActive(TRUE)
  theNewTheme.SetVisible(TRUE)
  theView.SetEditableTheme(theNewTheme)
  av.GetProject.SetModified(true)
'end

if (theNewFtab = Nil) then
  exit
end

theKount = 0

for each g in theSelectedGraphics
  theKount = theKount + 1

  theShape = g.GetShape


  if (theClass = "Point") then
    theNewRec = theNewFtab.AddRecord
    thePoint = theShape
    theNewFtab.SetValue(theNewShapeField,theNewRec,thePoint)
    theNewFtab.SetValueNumber(theIdField,theNewRec,(theKount))
  elseif (theClass = "Line") then
    theNewRec = theNewFtab.AddRecord
    thePolyLine = theShape.asPolyLine
    theNewFtab.SetValue(theNewShapeField,theNewRec,thePolyLine)
    theNewFtab.SetValueNumber(theIdField,theNewRec,(theKount))
  elseif (theClass = "Polygon") then
    theNewRec = theNewFtab.AddRecord
    thePolygon = theShape.asPolygon
    theNewFtab.SetValue(theNewShapeField,theNewRec,thePolygon)
    theNewFtab.SetValueNumber(theIdField,theNewRec,(theKount))
  end

end

theNewFtab.SetEditable(false)
tbl.SetEditable(false)

'Graphics.SelectAll     Now we select all of the point graphics to be deleted
theDoc = av.GetActiveDoc
theDoc.GetGraphics.SelectAll

'View.DeleteGraphics
theView = av.GetActiveDoc
theTheme = theView.GetEditableTheme

if (theView.GetGraphics.HasSelected) then
  av.GetProject.SetModified(true)
end
if (theTheme = nil) then
  theView.GetGraphics.ClearSelected
else
  theTheme.GetFtab.BeginTransaction
  theTheme.ClearSelected
  theTheme.GetFTab.EndTransaction
end
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       NetRun2
' Script Description: Finds a route between the two locations.
'
' Programmer:        Sarah Riley
' Date Created:      Nov. 2000
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    26-Apr-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'get the view and the location theme
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theTheme=theView.FindTheme("Locate.shp")
theFtab=theTheme.getFtab

theStreetFTab=theView.FindTheme("Transit routes.shp").getFtab
theNetDef=NetDef.Make(theStreetFTab)
theNetwork=Network.Make(theNetDef)

thePointList={}
thePointField=theFtab.FindField("Shape")
for each rec in theFtab
  p=theFtab.ReturnValue(thePointField, rec)
  thePointList.add(p)
end

'*************
' Set the cost for solving a routing problem
' It then solves the problem and displays
' directions for the route.

' Get a list of available costs for this line theme. Choose a cost from the list.
'
aNetCostFieldList = theNetDef.GetCostFields
aNetCostField = MsgBox.Choice(aNetCostFieldList, "Select Mintues", "Select Minutes")

' If a cost was chosen, set it.

if (aNetCostField<>nil) then
  theNetwork.SetCostField(aNetCostField)
end

'The above can be replaced by changing the hard code for Network Analyst wherein
'there will only be the chosen cost of MINUTES to select.
'****************

TravelDistance= theNetwork.FindPath(thePointList, True, False)
'msgBox.Info( TravelDistance.AsString, "Total"++reportUnitStr )

'theView.GetGraphics.add(GraphicShape.Make(theNetwork.ReturnPathShape))

aSymbol = Symbol.Make(#SYMBOL_PEN)
aSymbol.SetSize(2.0)
aSymbol.SetColor(Color.GetMagenta)

theGL=theView.GetGraphics
theRoute=GraphicShape.Make(theNetwork.ReturnPathShape)
theRoute.SetDisplay(av.GetActiveDoc.GetDisplay)
theroute.SetSymbol(aSymbol)
theGL.Add(theRoute)
'theroute.GetSymbol.SetColor(color.GetMagenta).SetSymbol(aSymbol)
theRoute.Draw
theroute.Invalidate
```

```
'set view size for public viewing

theView = av.getproject.finddoc("Route Finding")
theWin = theView.GetWin
'theWin.Resize(1025,600)
'theWin.Moveto(0,0)
theWin.Open

' Write the route result FTab, this is necessary to get directons.
'
tmpFileName = FileName.Make("F:\thesis data\Route.shp")
theNetwork.WritePath(tmpFileName)
resFTab = FTab.Make(SrcName.Make(tmpFileName.AsString))

' Show the directions then remove the temporary file.
'
'theDirections="Your route directions are as follows"+NL+NL+
'            theNetwork.GetPathDirections(resFTab)
'msgBox.Report(theDirections, "Route Information")

'add theme to view
newTheme = FTheme.Make(resFTab)
theView.AddTheme(newTheme)
newTheme.SetName("Route")
newTheme.SetVisible(True)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       nearfeatures2
' Script Description: Finds the nearest feature to points in a point theme
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:   Table with the following items: a 'From' field,
'                    a 'To' field, and a distance to nearest feature field

'
' Changes:
' Date          Details
' 29-Apr-2002   Initial Implementation
'
' Author
' Timothy J Fox
' ESRI
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Set search radius and other vars...
''''''''''''''''''''''''''''''''''''''''''''''''''''''''

shortDist = 1000000000
v = av.GetActiveDoc
tl = v.getThemes
shortTRec = 0
theCount = 0
w = "Warning"
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Get 'From' theme and field - check for 'From' selection, and field
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

f = v.FindTheme("Locate.shp")
'f = msgbox.list (tl, "Select the theme that contains" + NL + "the 'From' features." ,"'From' theme")
'if (f = nil) then
'  exit
'end
fromName = f.GetName
fromShpFld = f.GetFTab.FindField("Shape")
'fFieldList = f.GetFTab.GetFields
'ff = msgbox.multilist (fFieldList, "Select 'From' field to preserve" ,"'From' field")

Field1 = f.GetFTab.FindField("ID")
'Field2 = f.GetFTab.FindField("Address")
'fFieldList = {Field1, Field2}

'if (ff = nil) then
'  msgbox.warning("you must select a field" ,w)
'  exit
'end
if (f.GetFTab.GetSelection.Count = 0) then
   fromFeatures = f.GetFTab
   fromTotal = f.GetFTab.GetNumRecords
  else
   fromFeatures = f.GetFTab.GetSelection
   fromTotal = fromFeatures.Count
end

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'get 'To' theme and field
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

t = v.FindTheme("Bus Stops")
't = msgbox.list (tl, "Select 'To' theme", "'To' theme")
'if (t = nil) then
'  exit
'end
toName = t.GetName
toShpFld  = t.GetFTab.FindField("Shape")
'tFieldList = t.GetFTab.GetFields
'tf = msgbox.multilist (tFieldList, "Select field to preserve", "'To' field")
```

```
'if (tf = nil) then
'  msgbox.warning("you must select a field", w)
'  exit
'end

Field3 = t.GetFTab.FindField("Bus_stop")
Field4 = t.GetFTab.FindField("On_street")
Field5 = t.GetFTab.FindField("At_street")
'tFieldList = {Field3, Field4, Field5}


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'get 'To' Theme's selection
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

if (t.GetFTab.GetSelection.Count = 0) then
  toFeatures = t.GetFTab
  else
  toFeatures = t.GetFTab.GetSelection
end
''''''''''''''''''''''''''''''''''''''''''''''''''''
'create new table
''''''''''''''''''''''''''''''''''''''''''''''''''''

tempFilename = FileName.Make("F:thesis data\near.dbf")
'theFilename  = FileDialog.Put(tempFilename, "*.dbf", "Save Table where?")
'if (theFilename = nil) then
'  exit
'end
tempFilename.SetExtension("dbf")
'theFilename.SetExtension("dbf")
'newVTab = VTab.MakeNew (theFilename, dBASE
newVTab = VTab.MakeNew (tempFilename, dBASE)
'fID = ff.clone
'tID = tf.clone
f1 = Field1.clone
'f2 = Field2.clone
t1 = Field3.clone
t2 = Field4.clone
t3 = Field5.clone
distField = Field.Make("NF Dist",#FIELD_DECIMAL,8,2)
newFields = {f1, t1, t2, t3, distField}
newVTab.AddFields(newFields)
''''''''''''''''''''''''''''''''''''''''''''''''
' Find the closest feature to the point...
''''''''''''''''''''''''''''''''''''''''''''''''

av.ShowMsg("NF analysis: From"+ fromName + " To " + toName )
for each frec in fromFeatures
 fPoint = f.GetFTab.ReturnValue(fromShpFld,frec)
 theCount = theCount + 1
 for each trec in ToFeatures
  tShape = t.GetFTab.ReturnValue(toShpFld,trec)
  dist = tShape.Distance(fPoint)
  if (dist < shortDist) then
   shortDist = dist
   shortTRec = trec.Clone
  end
 end
 ''''''''''''''''''''''''''''''''''''''''''''''''
 'Write to table: 'distance','From' and 'To' fields
 ''''''''''''''''''''''''''''''''''''''''''''''''

 'tempTid = t.GetFTab.ReturnValue(tf,shortTRec)
 tempFid1 = f.GetFTab.ReturnValue(Field1,frec)
 'tempFid2 = f.GetFTab.ReturnValue(Field2,frec)
 tempTid3 = t.GetFTab.ReturnValue(Field3,shortTRec)
 tempTid4 = t.GetFTab.ReturnValue(Field4,shortTRec)
 tempTid5 = t.GetFTab.ReturnValue(Field5,shortTRec)

 newVTab.SetEditable (true)
 newRec = newVTab.AddRecord
 newVTab.SetValue(f1,newRec,tempFid1)
 'newVTab.SetValue(f2,newRec,tempFid2)
```

```
  newVTab.SetValue(t1,newRec,tempTid3)
  newVTab.SetValue(t2,newRec,tempTid4)
  newVTab.SetValue(t3,newRec,tempTid5)
  'newVTab.SetValue(tf, newRec, tempTid)
  newVTab.SetValue(distField,newRec,shortDist)
  shortDist = 1000000000
  av.SetStatus((theCount / fromTotal) * 100)
end
av.ClearStatus
av.ClearMsg
newVTab.SetEditable (false)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' create table from VTab & display table
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
newTab = table.Make (newVTab)
newTab.setName ("near.dbf")
newTabWin = newTab.GetWin
newTabWin.Resize (350, 250)
newTabWin.Open
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      SelByTheme2
' Script Description: Performs a select by theme based on the chosen
'                   bus stops and finds the stop within the
'                   schedule table.
'
' Programmer:       Sarah Riley
' Date Created:     Apr. 2002
' Returned Object:  none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley    2-May-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'Features will be selected from aTheme
'based on anotherTheme.

aView = av.GetProject.FindDoc("Route Finding")
theWin = aView.GetWin
theWin.Open

theThemeList=aView.getThemes
for each i in (0..(theThemeList.count-1))
  temp=theThemeList.get(i).setActive(FALSE)
end

aTheme = aView.FindTheme("Bus Stops")
aTheme.SetActive(TRUE)
anotherTheme = aView.FindTheme("Locate.shp")
theFTab = anotherTheme.GetFTab

'get the first selected record for merging
if(theFTab = nil)then
 MsgBox.Info("No table...", "")
 exit
end

theBitmap = Bitmap.Make(theFTab.GetNumRecords)
for each rec in (0..(theFTab.GetNumRecords-1)) by 2
 theBitmap.Set(rec)
end
theFTab.SetSelection(theBitmap)
theFTab.UpdateSelection

'msgbox.info(av.GetActiveDoc.GetDisplay.GetUnits.AsString, "")
'
aDistance = Units.Convert(1.5, #UNITS_LINEAR_KILOMETERS, av.GetActiveDoc.GetDisplay.GetUnits )


aRelType = #FTAB_RELTYPE_ISWITHINDISTANCEOF
'aDistance = 2 'make sure this is km else it would be meters

'----this can change--maybe do a for statement until a bus stop is found

aSelType = #VTAB_SELTYPE_AND

aTheme.SelectByTheme(anotherTheme, aRelType, aDistance, aSelType)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       merge2
' Script Description: Merges Chosen Locations and Bus Stop themes
'                  in order to find the closest bus stop with
'                  a scheduled time.
'
' Programmer:        Sarah Riley
' Date Created:      Nov. 2000
' Returned Object:   none
'
' Changes:
'
' Author       Date          Details
'
' S. Riley     30-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'get the theme and the location theme
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Locate.shp")
theFTab = theTheme.GetFTab

'get the first selected record for merging
if(theFTab = nil)then
  MsgBox.Info("No table...", "")
  exit
end

theBitmap = Bitmap.Make(theFTab.GetNumRecords)
for each rec in (0..(theFTab.GetNumRecords-1)) by 2
  theBitmap.Set(rec)
end
theFTab.SetSelection(theBitmap)
theFTab.UpdateSelection

themesToMerge = List.Make
themesToMerge.add(theView.FindTheme("locate.shp"))
themesToMerge.add(theView.FindTheme("Bus Stops"))
themesToMerge.add(theView.FindTheme("locate.shp"))

' Themes must have matching shape types for merging.  Using the first
' active theme verify that this is the case...

checkType  = themesToMerge.Get(0).GetFtab.FindField("Shape").GetType

for each i in 1 .. (themesToMerge.Count - 1)
 t = themesToMerge.Get(i)
 if (checkType <> t.GetFTab.FindField("Shape").GetType) then
   MsgBox.Error("Theme feature type mismatch - Unable to merge.","")
   exit
 end
end

' Specify the output shapefile...

'outFName = av.GetProject.MakeFileName("Time", "shp")
theCWD=FileName.GetCWD
outFName=theCWD.makeTmp("Time","shp")

' Create the list of fields used for the output theme.  The fields
' are taken from the first active theme only, it is assumed that
' other themes have an identical set of fields.  If this is not the
' case the themes will still be merged, however fields not found in
' other themes will be empty...

fieldList = List.Make

for each f in themesToMerge.Get(0).GetFTab.GetFields
```

```
  if (f.GetName = "Shape") then
    continue
  else
    fCopy = f.Clone
    fieldList.Add(fCopy)
  end
end

' Get the class of new FTab to create, create the new FTab and
' add fields that we've gathered from the input themes....

shapeType = themesToMerge.Get(0).GetFTab.FindField("Shape").GetType
outClass = POINT

mergeFTab = FTab.MakeNew( outFName, outClass )
if (fieldList.Count > 0) then
  mergeFTab.AddFields( fieldList )
end

' Populate the new FTab from the FTabs of the input themes...

for each t in themesToMerge
  av.ShowMsg( "Merging"++t.GetName )
  inFTab  = t.GetFTab
  if (inFTab.GetSelection.Count = 0) then
    theRecordsToMerge = inFTab
    numRecs = inFTab.GetNumRecords
  else
    theRecordsToMerge = inFTab.GetSelection
    numRecs = theRecordsToMerge.Count
  end

  for each rec in theRecordsToMerge
    av.SetStatus( (rec / numRecs) * 100 )
    newRec   = mergeFTab.AddRecord
    inField  = inFTab.FindField( "Shape" )
    outField = mergeFTab.FindField( "Shape" )
    mergeFTab.SetValue( outField, newrec, inFTab.ReturnValue( inField, rec ))

    if (fieldList.Count > 0) then
      for each f in fieldList
        fName    = f.GetName
        inField  = inFTab.FindField( fName )

        ' Skip field if not found in inFTab...
        if ( inField <> Nil ) then
          outField = mergeFTab.FindField( fName )
          aValue   = inFTab.ReturnValue( inField, rec )
          mergeFTab.SetValue( outField, newRec, aValue )
        end
      end  ' for each f
    end  ' if count
  end  ' for each rec
end  ' for each t

av.ClearMsg
av.ClearStatus

' add the new theme to the view

mergeTheme = FTheme.Make( mergeFTab )
theView.AddTheme(mergeTheme)
mergeTheme.SetName("Time")
mergeTheme.SetVisible(True)

'show the view
theView = av.GetProject.FindDoc("Route Finding")
theWin = theView.GetWin
theWin.Open
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      time2
' Script Description: Finds the time from the origin bus stop and
'                   the bus stop with a schedule.
'
' Programmer:       Sarah Riley
' Date Created:     Nov. 2000
' Returned Object:   none
'
' Changes:
'
' Author      Date      Details
'
' S. Riley    30-Apr-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Time")
theFtab = theTheme.getFtab

theStreetFTab = theView.FindTheme("Transit routes.shp").getFtab
theNetDef = NetDef.Make(theStreetFTab)
theNetwork = Network.Make(theNetDef)

thePointList={}
thePointField=theFtab.FindField("Shape")
for each rec in theFtab
 p=theFtab.ReturnValue(thePointField, rec)
 thePointList.add(p)
end

'*******************
' Set the cost for solving a routing problem
' It then solves the problem and displays
' directions for the route.

' Get a list of available costs for this line theme. Choose a cost from the list.
'
aNetCostFieldList = theNetDef.GetCostFields
aNetCostField = MsgBox.Choice(aNetCostFieldList, "Select Minutes", "Select Minutes")

' If a cost was chosen, set it.
if (aNetCostField<>nil) then
 theNetwork.SetCostField(aNetCostField)
end

'The above can be replaced by changing the hard code for Network Analyst wherein
'there will only be the chosen cost of MINUTES to select.
'****************


TravelDistance= theNetwork.FindPath(thePointList, True, False)
'msgBox.Info( TravelDistance.AsString, "Total"++reportUnitStr )

'theView.GetGraphics.add(GraphicShape.Make(theNetwork.ReturnPathShape))

'aSymbol = Symbol.Make(#SYMBOL_PEN)
'aSymbol.SetSize(2.0)
'aSymbol.SetColor(Color.GetMagenta)
'
'theGL=theView.GetGraphics
'theRoute=GraphicShape.Make(theNetwork.ReturnPathShape)
'theRoute.SetDisplay(av.GetActiveDoc.GetDisplay)
'theroute.SetSymbol(aSymbol)
'theGL.Add(theRoute)
''theroute.GetSymbol.SetColor(color.GetMagenta).SetSymbol(aSymbol)
'theRoute.Draw
'theroute.Invalidate
'
```

```
'set view size for public viewing

theView = av.getproject.FindDoc("Route Finding")
theWin = theView.GetWin
'theWin.Resize(1025,600)
'theWin.Moveto(0,0)
theWin.Open

' Write the route result FTab, this is necessary to get directons.
'
tmpFileName = FileName.Make("F:\thesis data\route.shp")
theNetwork.WritePath(tmpFileName)
resFTab = FTab.Make(SrcName.Make(tmpFileName.AsString))
'
' Show the directions then remove the temporary file.
'
'theDirections="Your route directions are as follows"+NL+NL+
'            theNetwork.GetPathDirections(resFTab)
'msgBox.Report(theDirections, "Route Information")

'gets the view, sets the time invisible
theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Time")
theTheme.SetActive(FALSE)
theTheme.SetVisible(FALSE)

'add theme to view
newTheme = FTheme.Make(resFTab)
theView.AddTheme(newTheme)
newTheme.SetName("Route2")
newTheme.SetVisible(True)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      stats2
' Script Description: Performs the Bayesian statistics, reports
'                 the results in a report message box and
'                 zooms to the route in the view.
'
' Programmer:       Sarah Riley
' Date Created:     Apr. 2002
' Returned Object:  none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley   15-Jun-2002      Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'get the view and the route theme
theView = av.GetActiveDoc
theTheme = theView.FindTheme("Route")
theTable = theTheme.GetFTab

'set up table for selection
thebitmap = theTable.GetSelection
expr = "[F_cost] = 0"
theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
theTable.UpdateSelection ' highlight found records

theField = theTable.FindField("T_cost")

rec = theBitMap.GetNextSet(-1) 'first record in bitmap
theField2 = theTable.ReturnValue(theField, rec) 'fill values


'msgbox.info(theField2.AsString, " ")

'aBitmap.ClearAll

'now perform the statistics on this value
'perform the statistics on theField2.AsString


TravelTime = theField2

T1 = TravelTime - 1
T2 = TravelTime
T3 = TravelTime + 3

'all in minutes

'route = route number from table

' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
' since each table is cleared at the end, only one table should have records highlighted

dbTable1 = av.FindDoc("weekday.dbf")
theVTab1 = dbTable1.GetVTab
theSelection1 = theVTab1.GetSelection

selCount1 = theSelection1.Count

if (selCount1 <> 0) then

'call on the route number from the field "Route_no"

theField1 = theVTab1.FindField("Route_no")
rec = theSelection1.GetNextSet(-1) 'first record in the selection
aField = theVTab1.ReturnValue(theField1, rec)
```

```
if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

 'use transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.255
    P2 = 0.72
    P3 = 0.025
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.165
      P2 = 0.705
      P3 = 0.13
    else
      P1 = 0.20
      P2 = 0.705
      P3 = 0.095
    end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.26
   P2 = 0.64
   P3 = 0.10
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.25
     P2 = 0.655
     P3 = 0.095
   else
     P1 = 0.25
     P2 = 0.635
     P3 = 0.115
   end
 elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
   P1 = 0.21
   P2 = 0.645
   P3 = 0.145
 else
   P1 = 0.24
   P2 = 0.645
   P3 = 0.115  'all day value
 end

else

'use non-transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
   P1 = 0.225
   P2 = 0.71
   P3 = 0.065
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
   if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
     P1 = 0.195
     P2 = 0.555
     P3 = 0.25
   else
     P1 = 0.24
     P2 = 0.60
     P3 = 0.16
   end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.34
   P2 = 0.585
   P3 = 0.075
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.34
     P2 = 0.525
```

```
      P3 = 0.135
    else
      P1 = 0.405
      P2 = 0.49
      P3 = 0.105
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.305
    P2 = 0.565
    P3 = 0.13
  else
    P1 = 0.325
    P2 = 0.56
    P3 = 0.115  'all day values
  end
end

' read the results from the schedule tables

theTable2 = av.FindDoc("weekday.dbf")
theVTab2 = theTable2.GetVTab
theBitmap2 = theVTab2.GetSelection

theVTab2.UpdateSelection
'theVTab2.PromoteSelection
aField4 = theVTab2.FindField("Route_no")
aField5 = theVTab2.FindField("Time")
rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

dbTable2 = av.FindDoc("saturday.dbf")
theVTab2 = dbTable2.GetVTab
theSelection2 = theVTab2.GetSelection

selCount2 = theSelection2.GetSelection

if (selCount2 <> 0) then
 'call on the route number from the field "Route_no"

 theField2 = theVTab2.FindField("Route_no")
 rec = theSelection2.GetNextSet(-1) 'first record in bitmap
 aField = theVTab2.ReturnValue(theField2, rec) 'fill values

 if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

  'use transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.245
    P2 = 0.745
    P3 = 0.01
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.35
    P2 = 0.615
    P3 = 0.035
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.26
    P2 = 0.68
    P3 = 0.06
  elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
    P1 = 0.23
    P2 = 0.645
    P3 = 0.125
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.22
```

```
      P2 = 0.685
      P3 = 0.095
    else
      P1 = 0.26
      P2 = 0.66
      P3 = 0.08  'all day values
    end

  else

    'use non transitway stats
    if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
      P1 = 0.37
      P2 = 0.50
      P3 = 0.13
    elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
      P1 = 0.215
      P2 = 0.655
      P3 = 0.13
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
      P1 = 0.23
      P2 = 0.64
      P3 = 0.13
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.24
      P2 = 0.51
      P3 = 0.25
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.32
      P2 = 0.60
      P3 = 0.08
    else
      P1 = 0.265
      P2 = 0.595
      P3 = 0.14  'all day values
    end
  end

  ' read the results from the schedule tables

  theTable3 = av.FindDoc("saturday.dbf")
  theVTab3 = theTable3.GetVTab
  theBitmap3 = theVTab3.GetSelection

  theVTab3.UpdateSelection
  'theVTab3.PromoteSelection
  aField4 = theVTab3.FindField("Route_no")
  aField5 = theVTab3.FindField("Time")
  rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
  dbTable3 = av.FindDoc("sunday.dbf")
  theVTab3 = dbTable3.GetVTab
  theSelection3 = theVTab3.GetSelection

  selCount3 = theSelection3.GetSelection

  if(selCount3 <> 0) then
    'call on the route number from the field "Route_no"

    theField3 = theVTab3.FindField("Route_no")
    rec = theSelection3.GetNextSet(-1)  'first record in the selection
    aField = theVTab3.ReturnValue(theField3, rec)

    if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
      or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then
```

```
  'use transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.51
    P2 = 0.365
    P3 = 0.125
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.27
    P2 = 0.68
    P3 = 0.05
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.155
    P2 = 0.74
    P3 = 0.105
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.13
    P2 = 0.715
    P3 = 0.155
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.19
    P2 = 0.75
    P3 = 0.06
  else
    P1 = 0.18
    P2 = 0.715
    P3 = 0.105  'all day values
  end

 else
  'use non transitway stats

  if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.32
    P2 = 0.605
    P3 = 0.075
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.21
    P2 = 0.665
    P3 = 0.125
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.30
    P2 = 0.62
    P3 = 0.08
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.35
    P2 = 0.565
    P3 = 0.085
  else
    P1 = 0.285
    P2 = 0.615
    P3 = 0.10  'all day values
  end
 end

' read the results from the schedule tables

theTable4 = av.FindDoc("sunday.dbf")
theVTab4 = theTable4.GetVTab
theBitmap4 = theVTab4.GetSelection

theVTab4.UpdateSelection
'theVTab4.PromoteSelection
aField4 = theVTab4.FindField("Route_no")
aField5 = theVTab4.FindField("Time")
rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values
```

```
    else
    'MsgBox.Info("No tables have a selection....", "")
    end
  end
end


'the new expected time is calculated

ExTime = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

'work through Bayesian Probabilities
'discete values are given
'therefore use discrete case
'

'check for advisory

aTheme = theView.FindTheme("Transit routes.shp")
anFTab = aTheme.GetFTab

'aBitmap = anFTab.GetSelection
'
'aCount = aBitmap.Count
'
'if (aCount <> 0) then
'  MsgBox.Info("Selection", "")
'else
'  aBitmap.SetAll
'end

aBitmap = Bitmap.Make(anFTab.GetNumRecords)
for each rec in (0..(anFTab.GetNumRecords-1)) by 2033
  aBitmap.Set(rec)
end
anFTab.SetSelection(aBitmap)
anFTab.UpdateSelection

AdvisField = anFTab.FindField("Advisory")

rec = aBitMap.GetNextSet(-1) 'first record in bitmap
AdvField = anFTab.ReturnValue(AdvisField, rec) 'fill values


if (AdvField = 0)then  'no advisory
  Time = ExTime

else  'advisory

'conditional probabilities

'P(early|early)
EE = 0.7

'P(early|ontime)
EO = 0.25

'P(early|late)
EL = 0.05

'P(ontime|early)
OE = 0.15

'P(ontime|ontime)
OO = 0.7

'P(ontime|late)
OL = 0.15
```

```
'P(late|early)
LE = 0.05

'P(late|ontime)
LO = 0.25

'P(late||late)
LL = 0.7

'calculate the denominators for the Bayesian equation

early = (P1*EE) + (P2*OE) + (P3*LE)
ontime = (P1*EO) + (P2*OO) + (P3*LO)
late = (P1*EL) + (P2*OL) +(P3*LL)

'need to find out what condition the buses are running
'these will be posterior probabilities
'OC TRANSPO MUST ENTER A CODE TO STATE HOW BUSES ARE RUNNING ie. R1, R2, R3
'R1 = EARLY
'R2 = ONTIME
'R3 = LATE

aTheme = theView.FindTheme("Transit routes.shp")
anFTab = aTheme.GetFTab

'aBitmap = anFTab.GetSelection
'
'aCount = aBitmap.Count
'
'if (aCount <> 0) then
'  MsgBox.Info("Selection", "")
'else
'  aBitmap.SetAll
'end

aBitmap = Bitmap.Make(anFTab.GetNumRecords)
for each rec in (0..(anFTab.GetNumRecords-1)) by 2033
  aBitmap.Set(rec)
end
anFTab.SetSelection(aBitmap)
anFTab.UpdateSelection

ConField = anFTab.FindField("Condition")

rec = aBitMap.GetNextSet(-1) 'first record in bitmap
condField = anFTab.ReturnValue(ConField, rec) 'fill values

if (condField.AsString = "R2") then 'ontime

   'P(ontime|ontime)
   time1 = (P2*OO)/ontime

   'P(early|ontime)
   time2 = (P1*EO)/ontime

   'P(ontime|ontime)
   time3 = (P3*LO)/ontime

   Time = (time1*T2) + (time2*T1) + (time3*T3)


   'rerun routing to determine if there is another route
   av.run("netrun3","")

   'get the view and the route theme
   theView = av.GetActiveDoc
   theTheme = theView.FindTheme("Route1")
   theTable = theTheme.GetFTab

   'set up table for selection
```

```
thebitmap = theTable.GetSelection
expr = "[F_cost] = 0"
theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
theTable.UpdateSelection ' highlight found records

theField = theTable.FindField("T_cost")

rec = theBitMap.GetNextSet(-1) 'first record in bitmap
theField2 = theTable.ReturnValue(theField, rec) 'fill values


'msgbox.info(theField2.AsString, " ")

'aBitmap.ClearAll

'now perform the statistics on this value
'perform the statistics on theField2.AsString


TravelTime = theField2

T1 = TravelTime - 1
T2 = TravelTime
T3 = TravelTime + 3

'all in minutes

'route = route number from table

' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
' since each table is cleared at the end, only one table should have records highlighted

dbTable1 = av.FindDoc("weekday.dbf")
theVTab1 = dbTable1.GetVTab
theSelection1 = theVTab1.GetSelection

selCount1 = theSelection1.Count

if (selCount1 <> 0) then

'call on the route number from the field "Route_no"

theField1 = theVTab1.FindField("Route_no")
rec = theSelection1.GetNextSet(-1)  'first record in the selection
aField = theVTab1.ReturnValue(theField1, rec)

if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

'use transitway stats

if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
   P1 = 0.255
   P2 = 0.72
   P3 = 0.025
elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
   if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
     P1 = 0.165
     P2 = 0.705
     P3 = 0.13
   else
     P1 = 0.20
     P2 = 0.705
     P3 = 0.095
   end
elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
  P1 = 0.26
  P2 = 0.64
  P3 = 0.10
elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
```

```
    if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
      P1 = 0.25
      P2 = 0.655
      P3 = 0.095
    else
      P1 = 0.25
      P2 = 0.635
      P3 = 0.115
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.21
    P2 = 0.645
    P3 = 0.145
  else
    P1 = 0.24
    P2 = 0.645
    P3 = 0.115  'all day value
  end

else

'use non-transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.225
    P2 = 0.71
    P3 = 0.065
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.195
      P2 = 0.555
      P3 = 0.25
    else
      P1 = 0.24
      P2 = 0.60
      P3 = 0.16
    end
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.34
    P2 = 0.585
    P3 = 0.075
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
      P1 = 0.34
      P2 = 0.525
      P3 = 0.135
    else
      P1 = 0.405
      P2 = 0.49
      P3 = 0.105
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.305
      P2 = 0.565
      P3 = 0.13
  else
      P1 = 0.325
      P2 = 0.56
      P3 = 0.115  'all day values
  end
end

' read the results from the schedule tables

theTable2 = av.FindDoc("weekday.dbf")
theVTab2 = theTable2.GetVTab
theBitmap2 = theVTab2.GetSelection

theVTab2.UpdateSelection
'theVTab2.PromoteSelection
```

```
aField4 = theVTab2.FindField("Route_no")
aField5 = theVTab2.FindField("Time")
rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

dbTable2 = av.FindDoc("saturday.dbf")
theVTab2 = dbTable2.GetVTab
theSelection2 = theVTab2.GetSelection

selCount2 = theSelection2.GetSelection

if (selCount2 <> 0) then
 'call on the route number from the field "Route_no"

 theField2 = theVTab2.FindField("Route_no")
 rec = theSelection2.GetNextSet(-1) 'first record in bitmap
 aField = theVTab2.ReturnValue(theField2, rec) 'fill values

 if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

  'use transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.245
    P2 = 0.745
    P3 = 0.01
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.35
    P2 = 0.615
    P3 = 0.035
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.26
    P2 = 0.68
    P3 = 0.06
  elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
    P1 = 0.23
    P2 = 0.645
    P3 = 0.125
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.22
    P2 = 0.685
    P3 = 0.095
  else
    P1 = 0.26
    P2 = 0.66
    P3 = 0.08  'all day values
  end

 else

  'use non transitway stats
  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.37
    P2 = 0.50
    P3 = 0.13
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.215
    P2 = 0.655
    P3 = 0.13
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.23
    P2 = 0.64
    P3 = 0.13
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.24
```

```
    P2 = 0.51
    P3 = 0.25
   elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.32
    P2 = 0.60
    P3 = 0.08
   else
    P1 = 0.265
    P2 = 0.595
    P3 = 0.14  'all day values
   end
  end

  ' read the results from the schedule tables

  theTable3 = av.FindDoc("saturday.dbf")
  theVTab3 = theTable3.GetVTab
  theBitmap3 = theVTab3.GetSelection

  theVTab3.UpdateSelection
  'theVTab3.PromoteSelection
  aField4 = theVTab3.FindField("Route_no")
  aField5 = theVTab3.FindField("Time")
  rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
  dbTable3 = av.FindDoc("sunday.dbf")
  theVTab3 = dbTable3.GetVTab
  theSelection3 = theVTab3.GetSelection

  selCount3 = theSelection3.GetSelection

  if(selCount3 <> 0) then
   'call on the route number from the field "Route_no"

   theField3 = theVTab3.FindField("Route_no")
   rec = theSelection3.GetNextSet(-1)  'first record in the selection
   aField = theVTab3.ReturnValue(theField3, rec)

   if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
     or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

    'use transitway stats

    if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
     P1 = 0.51
     P2 = 0.365
     P3 = 0.125
    elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
     P1 = 0.27
     P2 = 0.68
     P3 = 0.05
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
     P1 = 0.155
     P2 = 0.74
     P3 = 0.105
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
     P1 = 0.13
     P2 = 0.715
     P3 = 0.155
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
     P1 = 0.19
     P2 = 0.75
     P3 = 0.06
    else
     P1 = 0.18
```

```
        P2 = 0.715
        P3 = 0.105  'all day values
      end

    else
     'use non transitway stats

      if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
        P1 = 0.32
        P2 = 0.605
        P3 = 0.075
      elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
        P1 = 0.21
        P2 = 0.665
        P3 = 0.125
      elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
        P1 = 0.30
        P2 = 0.62
        P3 = 0.08
      elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
        P1 = 0.35
        P2 = 0.565
        P3 = 0.085
      else
        P1 = 0.285
        P2 = 0.615
        P3 = 0.10  'all day values
      end
    end

  ' read the results from the schedule tables

  theTable4 = av.FindDoc("sunday.dbf")
  theVTab4 = theTable4.GetVTab
  theBitmap4 = theVTab4.GetSelection

  theVTab4.UpdateSelection
  'theVTab4.PromoteSelection
  aField4 = theVTab4.FindField("Route_no")
  aField5 = theVTab4.FindField("Time")
  rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
  result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values


  else
  'MsgBox.Info("No tables have a selection....", "")
  end
 end
end


'the new expected time is calculated

ExTime2 = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

if (Time < ExTime2)then
 TrvlTime = Time

elseif(Time > ExTime2)then
 TrvlTime = ExTime2
else
 TrvlTime = Time
end


elseif(condField.AsString = "R1")then  'early

 'P(ontime|early)
```

```
time1 = (P2*OE)/early

'P(early|early)
time2 = (P1*EE)/early

'P(late|early)
time3 = (P3*LE)/early

ExTime = (time1*T2) + (time2*T1) + (time3*T3)
'ExTime = ExTime1.Round

time = ExTime

'rerun routing to determine if there is another route
 av.run("netrun3","")

 'get the view and the route theme
 theView = av.GetActiveDoc
 theTheme = theView.FindTheme("Route1")
 theTable = theTheme.GetFTab

 'set up table for selection
 thebitmap = theTable.GetSelection
 expr = "[F_cost] = 0"
 theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
 theTable.UpdateSelection ' highlight found records

 theField = theTable.FindField("T_cost")

 rec = theBitMap.GetNextSet(-1) 'first record in bitmap
 theField2 = theTable.ReturnValue(theField, rec) 'fill values


 'msgbox.info(theField2.AsString, " ")

 'aBitmap.ClearAll

 'now perform the statistics on this value
 'perform the statistics on theField2.AsString


 TravelTime = theField2

 T1 = TravelTime - 1
 T2 = TravelTime
 T3 = TravelTime + 3

 'all in minutes

 'route = route number from table

 ' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
 ' since each table is cleared at the end, only one table should have records highlighted

 dbTable1 = av.FindDoc("weekday.dbf")
 theVTab1 = dbTable1.GetVTab
 theSelection1 = theVTab1.GetSelection

 selCount1 = theSelection1.Count

 if (selCount1 <> 0) then

 'call on the route number from the field "Route_no"

 theField1 = theVTab1.FindField("Route_no")
 rec = theSelection1.GetNextSet(-1)  'first record in the selection
 aField = theVTab1.ReturnValue(theField1, rec)

if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then
```

```
'use transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.255
    P2 = 0.72
    P3 = 0.025
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.165
      P2 = 0.705
      P3 = 0.13
    else
      P1 = 0.20
      P2 = 0.705
      P3 = 0.095
    end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.26
   P2 = 0.64
   P3 = 0.10
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.25
     P2 = 0.655
     P3 = 0.095
   else
     P1 = 0.25
     P2 = 0.635
     P3 = 0.115
   end
 elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
   P1 = 0.21
   P2 = 0.645
   P3 = 0.145
 else
   P1 = 0.24
   P2 = 0.645
   P3 = 0.115  'all day value
 end

else

'use non-transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
   P1 = 0.225
   P2 = 0.71
   P3 = 0.065
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
   if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
     P1 = 0.195
     P2 = 0.555
     P3 = 0.25
   else
     P1 = 0.24
     P2 = 0.60
     P3 = 0.16
   end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.34
   P2 = 0.585
   P3 = 0.075
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.34
     P2 = 0.525
     P3 = 0.135
   else
     P1 = 0.405
```

```
      P2 = 0.49
      P3 = 0.105
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.305
    P2 = 0.565
    P3 = 0.13
  else
    P1 = 0.325
    P2 = 0.56
    P3 = 0.115  'all day values
  end
end

' read the results from the schedule tables

theTable2 = av.FindDoc("weekday.dbf")
theVTab2 = theTable2.GetVTab
theBitmap2 = theVTab2.GetSelection

theVTab2.UpdateSelection
'theVTab2.PromoteSelection
aField4 = theVTab2.FindField("Route_no")
aField5 = theVTab2.FindField("Time")
rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

dbTable2 = av.FindDoc("saturday.dbf")
theVTab2 = dbTable2.GetVTab
theSelection2 = theVTab2.GetSelection

selCount2 = theSelection2.GetSelection

if (selCount2 <> 0) then
 'call on the route number from the field "Route_no"

 theField2 = theVTab2.FindField("Route_no")
 rec = theSelection2.GetNextSet(-1) 'first record in bitmap
 aField = theVTab2.ReturnValue(theField2, rec) 'fill values

 if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

  'use transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.245
    P2 = 0.745
    P3 = 0.01
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.35
    P2 = 0.615
    P3 = 0.035
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.26
    P2 = 0.68
    P3 = 0.06
  elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
    P1 = 0.23
    P2 = 0.645
    P3 = 0.125
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.22
    P2 = 0.685
    P3 = 0.095
  else
```

```
      P1 = 0.26
      P2 = 0.66
      P3 = 0.08  'all day values
    end

  else

    'use non transitway stats
    if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
      P1 = 0.37
      P2 = 0.50
      P3 = 0.13
    elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
      P1 = 0.215
      P2 = 0.655
      P3 = 0.13
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
      P1 = 0.23
      P2 = 0.64
      P3 = 0.13
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.24
      P2 = 0.51
      P3 = 0.25
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.32
      P2 = 0.60
      P3 = 0.08
    else
      P1 = 0.265
      P2 = 0.595
      P3 = 0.14  'all day values
    end
  end

  ' read the results from the schedule tables

  theTable3 = av.FindDoc("saturday.dbf")
  theVTab3 = theTable3.GetVTab
  theBitmap3 = theVTab3.GetSelection

  theVTab3.UpdateSelection
  'theVTab3.PromoteSelection
  aField4 = theVTab3.FindField("Route_no")
  aField5 = theVTab3.FindField("Time")
  rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
  dbTable3 = av.FindDoc("sunday.dbf")
  theVTab3 = dbTable3.GetVTab
  theSelection3 = theVTab3.GetSelection

  selCount3 = theSelection3.GetSelection

  if(selCount3 <> 0) then
    'call on the route number from the field "Route_no"

    theField3 = theVTab3.FindField("Route_no")
    rec = theSelection3.GetNextSet(-1)  'first record in the selection
    aField = theVTab3.ReturnValue(theField3, rec)

    if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
      or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

      'use transitway stats
```

```
  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.51
    P2 = 0.365
    P3 = 0.125
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.27
    P2 = 0.68
    P3 = 0.05
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.155
    P2 = 0.74
    P3 = 0.105
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.13
    P2 = 0.715
    P3 = 0.155
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.19
    P2 = 0.75
    P3 = 0.06
  else
    P1 = 0.18
    P2 = 0.715
    P3 = 0.105  'all day values
  end

 else
  'use non transitway stats

  if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.32
    P2 = 0.605
    P3 = 0.075
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.21
    P2 = 0.665
    P3 = 0.125
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.30
    P2 = 0.62
    P3 = 0.08
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.35
    P2 = 0.565
    P3 = 0.085
  else
    P1 = 0.285
    P2 = 0.615
    P3 = 0.10  'all day values
  end
 end

' read the results from the schedule tables

theTable4 = av.FindDoc("sunday.dbf")
theVTab4 = theTable4.GetVTab
theBitmap4 = theVTab4.GetSelection

theVTab4.UpdateSelection
'theVTab4.PromoteSelection
aField4 = theVTab4.FindField("Route_no")
aField5 = theVTab4.FindField("Time")
rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values


else
'MsgBox.Info("No tables have a selection....", "")
```

```
  end
 end
end


'the new expected time is calculated

ExTime3 = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

 'P(ontime|early)
 time1 = (P2*OE)/early

 'P(early|early)
 time2 = (P1*EE)/early

 'P(late|early)
 time3 = (P3*LE)/early

 ExTime4 = (time1*T2) + (time2*T1) + (time3*T3)
 'ExTime = ExTime1.Round


if (time < ExTime4)then
 TrvlTime = Time

elseif(time > ExTime4)then
 TrvlTime = ExTime4
else
 TrvlTime = time
end



elseif(condField.AsString = "R3")then  'late

 'P(ontime|late)
 time1 = (P2*OL)/late

 'P(early|late)
 time2 = (P1*EL)/late

 'P(late|late)
 time3 = (P3*LL)/late

 ExTime = (time1*T2) + (time2*T1) + (time3*T3)
 'ExTime = ExTime1.Round

 time = ExTime

 'rerun routing to determine if there is another route
  av.run("netrun3","")

  'get the view and the route theme
  theView = av.GetActiveDoc
  theTheme = theView.FindTheme("Route1")
  theTable = theTheme.GetFTab

  'set up table for selection
  thebitmap = theTable.GetSelection
  expr = "[F_cost] = 0"
  theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
  theTable.UpdateSelection ' highlight found records

  theField = theTable.FindField("T_cost")

  rec = theBitMap.GetNextSet(-1) 'first record in bitmap
  theField2 = theTable.ReturnValue(theField, rec) 'fill values


  'msgbox.info(theField2.AsString, " ")
```

```
'aBitmap.ClearAll

'now perform the statistics on this value
'perform the statistics on theField2.AsString


TravelTime = theField2

T1 = TravelTime - 1
T2 = TravelTime
T3 = TravelTime + 3

'all in minutes

'route = route number from table

' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
' since each table is cleared at the end, only one table should have records highlighted

dbTable1 = av.FindDoc("weekday.dbf")
theVTab1 = dbTable1.GetVTab
theSelection1 = theVTab1.GetSelection

selCount1 = theSelection1.Count

if (selCount1 <> 0) then

'call on the route number from the field "Route_no"

theField1 = theVTab1.FindField("Route_no")
rec = theSelection1.GetNextSet(-1)  'first record in the selection
aField = theVTab1.ReturnValue(theField1, rec)

if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

'use transitway stats

if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.255
    P2 = 0.72
    P3 = 0.025
elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.165
      P2 = 0.705
      P3 = 0.13
    else
      P1 = 0.20
      P2 = 0.705
      P3 = 0.095
    end
elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.26
   P2 = 0.64
   P3 = 0.10
elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.25
     P2 = 0.655
     P3 = 0.095
   else
     P1 = 0.25
     P2 = 0.635
     P3 = 0.115
   end
elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
   P1 = 0.21
   P2 = 0.645
```

```
    P3 = 0.145
  else
    P1 = 0.24
    P2 = 0.645
    P3 = 0.115  'all day value
  end

else

'use non-transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.225
    P2 = 0.71
    P3 = 0.065
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.195
      P2 = 0.555
      P3 = 0.25
    else
      P1 = 0.24
      P2 = 0.60
      P3 = 0.16
    end
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.34
    P2 = 0.585
    P3 = 0.075
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
      P1 = 0.34
      P2 = 0.525
      P3 = 0.135
    else
      P1 = 0.405
      P2 = 0.49
      P3 = 0.105
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.305
      P2 = 0.565
      P3 = 0.13
  else
      P1 = 0.325
      P2 = 0.56
      P3 = 0.115  'all day values
  end
end

' read the results from the schedule tables

theTable2 = av.FindDoc("weekday.dbf")
theVTab2 = theTable2.GetVTab
theBitmap2 = theVTab2.GetSelection

theVTab2.UpdateSelection
'theVTab2.PromoteSelection
aField4 = theVTab2.FindField("Route_no")
aField5 = theVTab2.FindField("Time")
rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

dbTable2 = av.FindDoc("saturday.dbf")
theVTab2 = dbTable2.GetVTab
theSelection2 = theVTab2.GetSelection
```

```
selCount2 = theSelection2.GetSelection

if (selCount2 <> 0) then
 'call on the route number from the field "Route_no"

 theField2 = theVTab2.FindField("Route_no")
 rec = theSelection2.GetNextSet(-1) 'first record in bitmap
 aField = theVTab2.ReturnValue(theField2, rec) 'fill values

 if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

  'use transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.245
    P2 = 0.745
    P3 = 0.01
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.35
    P2 = 0.615
    P3 = 0.035
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.26
    P2 = 0.68
    P3 = 0.06
  elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
    P1 = 0.23
    P2 = 0.645
    P3 = 0.125
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.22
    P2 = 0.685
    P3 = 0.095
  else
    P1 = 0.26
    P2 = 0.66
    P3 = 0.08  'all day values
  end

 else

  'use non transitway stats
  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.37
    P2 = 0.50
    P3 = 0.13
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.215
    P2 = 0.655
    P3 = 0.13
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.23
    P2 = 0.64
    P3 = 0.13
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.24
    P2 = 0.51
    P3 = 0.25
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.32
    P2 = 0.60
    P3 = 0.08
  else
    P1 = 0.265
    P2 = 0.595
    P3 = 0.14  'all day values
  end
 end
```

```
' read the results from the schedule tables

theTable3 = av.FindDoc("saturday.dbf")
theVTab3 = theTable3.GetVTab
theBitmap3 = theVTab3.GetSelection

theVTab3.UpdateSelection
'theVTab3.PromoteSelection
aField4 = theVTab3.FindField("Route_no")
aField5 = theVTab3.FindField("Time")
rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
 dbTable3 = av.FindDoc("sunday.dbf")
 theVTab3 = dbTable3.GetVTab
 theSelection3 = theVTab3.GetSelection

 selCount3 = theSelection3.GetSelection

 if(selCount3 <> 0) then
  'call on the route number from the field "Route_no"

  theField3 = theVTab3.FindField("Route_no")
  rec = theSelection3.GetNextSet(-1)  'first record in the selection
  aField = theVTab3.ReturnValue(theField3, rec)

  if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
    or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

   'use transitway stats

   if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.51
    P2 = 0.365
    P3 = 0.125
   elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.27
    P2 = 0.68
    P3 = 0.05
   elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.155
    P2 = 0.74
    P3 = 0.105
   elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.13
    P2 = 0.715
    P3 = 0.155
   elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.19
    P2 = 0.75
    P3 = 0.06
   else
    P1 = 0.18
    P2 = 0.715
    P3 = 0.105  'all day values
   end

  else
   'use non transitway stats

   if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.32
    P2 = 0.605
    P3 = 0.075
   elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
```

```
      P1 = 0.21
      P2 = 0.665
      P3 = 0.125
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.30
      P2 = 0.62
      P3 = 0.08
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.35
      P2 = 0.565
      P3 = 0.085
    else
      P1 = 0.285
      P2 = 0.615
      P3 = 0.10  'all day values
    end
  end

  ' read the results from the schedule tables

  theTable4 = av.FindDoc("sunday.dbf")
  theVTab4 = theTable4.GetVTab
  theBitmap4 = theVTab4.GetSelection

  theVTab4.UpdateSelection
  'theVTab4.PromoteSelection
  aField4 = theVTab4.FindField("Route_no")
  aField5 = theVTab4.FindField("Time")
  rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
  result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values


  else
  'MsgBox.Info("No tables have a selection....", "")
  end
 end
end


'the new expected time is calculated

ExTime3 = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

 'P(ontime|early)
 time1 = (P2*OE)/early

 'P(early|early)
 time2 = (P1*EE)/early

 'P(late|early)
 time3 = (P3*LE)/early

 ExTime4 = (time1*T2) + (time2*T1) + (time3*T3)
 'ExTime = ExTime1.Round


if (time < ExTime4)then
 TrvlTime = Time

elseif(time > ExTime4)then
 TrvlTime = ExTime4
else
 TrvlTime = time
end


else
 'MsgBox.Info("Did nothing", "")
```

```
end
end

'read the results from the table created by Nearest Feature
'called near.dbf

theTable1 = av.FindDoc("near.dbf")
theVTab1 = theTable1.GetVTab
theBitmap1 = theVTab1.GetSelection
theBitmap1.SetAll
theVTab1.UpdateSelection

aField1 = theVTab1.FindField("Bus_stop")
aField2 = theVTab1.FindField("On_street")
aField3 = theVTab1.FindField("At_street")

rec1 = theBitMap1.GetNextSet(-1) 'first record in bitmap
result1 = theVTab1.ReturnValue(aField1, rec1) 'fill values
rec2 = theBitMap1.GetNextSet(-1) 'first record in bitmap
result2 = theVTab1.ReturnValue(aField2, rec2) 'fill values
rec3 = theBitMap1.GetNextSet(-1) 'first record in bitmap
result3 = theVTab1.ReturnValue(aField3, rec3) 'fill values

'result5 needs the T_cost added to it---time distance between the bus stop
'with a time and the bus stop without a time---if the distance is zero then
' result5 = "Time"

theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Route2")
theTable = theTheme.GetFTab
theBitmap = theTable.GetSelection

QueryString = "[F_cost] = 0"
theTable.Query(QueryString, theBitmap, #VTAB_SELTYPE_NEW)
theTable.UpdateSelection
theField = theTable.FindField("T_cost")
aRec = theBitmap.GetNextSet(-1)   'first record in bitmap
Tvalue = theTable.ReturnValue(theField, aRec)  'fill values

ArrivalTime = result5.AsNumber + Tvalue.Round

'get view and zoom to route
theView = av.GetActiveDoc
theTheme = theView.FindTheme("Locate.shp")
theTheme.SetActive(TRUE)

'View.ZoomToTheme

av.GetProject.SetModified(true)
theView = av.GetActiveDoc
theThemes = theView.GetActiveThemes
r = Rect.MakeEmpty
for each t in theThemes
  r = r.UnionWith(t.ReturnExtent)
end

if (r.IsEmpty) then
  return nil
elseif ( r.ReturnSize = (0@0) ) then
  theView.GetDisplay.PanTo(r.ReturnOrigin)
else
  theView.GetDisplay.SetExtent(r.Scale(1.1))
end

theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Bus Stops")
theTheme.SetVisible(FALSE)

theTheme2 = theView.FindTheme("Locate.shp")
theFTab = theTheme2.GetFTab
```

```
theBitmap = theFTab.GetSelection
theBitmap.ClearAll

theWin = theView.GetWin
theWin.Resize(1000,800)
theWin.Moveto(0,0)
theWin.Open

'report results

ArrTime = ArrivalTime.AsString
rightTime = ":" + ArrTime.Right(2)
' need to find out how many characters are in ArrivalTime
lftTime = ArrivalTime.AsString
lftTime.Count
if(lftTime.Count < 4)then
  leftTime = lftTime.Left(1)
else
  leftTime = lftTime.Left(2)
end

NewTime = leftTime + rightTime

'set decimal places for travel time
TvlTime = TrvlTime.SetFormat("dd.dd")

mymessage = "The route information is as follows:" + NL + NL +
"The route:  " + result4.AsString + NL + "The bus stop:  " +
result1.AsString + NL + "Location:  " + "On Street:  " +
result2.AsString + "  " + "At:  " + result3.AsString + NL +
NL + "Bus arrival at approximately:  " + NewTime.AsString + NL + NL +
"Approximate Travel Time:  " + TvlTime.AsString + "min" + NL + NL +
"If you need more information, please contact OCTranspo."

MsgBox.Report(mymessage, "Results")

'print the results
aSEd = SEd.MakeFromSource(mymessage, "tempscript")
aSEd.Print
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       Routing
' Script Description: Finds an optimum route, closest bus stop,
'                   the time the bus arrives, then links to the
'                   script stats to work through Bayesian
'                   analogy to give results to user
'                   This runs for Address and Location selections.
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    2-May-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'runs the netrun script to solve best route
av.run("netrun", "")

'runs the nearest features script to find closest
'bus stops to the locations chosen by user

av.run("nearfeatures", "")


'runs the select by theme script

av.run("SelByTheme", "")

'from the select by theme above--find the bus stop in the dbf table

theView = av.GetActiveDoc
thetable1 = av.FindDoc("weekday.dbf")
theVTab = thetable1.GetVTab

theBitmap1 = theVTab.GetSelection
selcount1 = theBitmap1.Count
if(selcount1 <> 0)then
 'theField = theVTab.FindField("Bus_stop")  'from dbf

 theTheme = theView.FindTheme("Bus Stops") 'fTheme
 anFTab = theTheme.GetFTab
 theBitmap2 = anFTab.GetSelection
 aField = anFTab.FindField("On_street")
 rec = theBitmap2.GetNextSet(-1)
 aField2 = anFTab.ReturnValue(aField, rec)
 'msgbox.info(aField2.AsString, "")
 myquerystring = "[On_street] = " + aField2.Quote
 theVTab.Query(myQueryString, theBitmap1, #VTAB_SELTYPE_AND)
 theVTab.UpdateSelection

else
 thetable2 = av.FindDoc("saturday.dbf")
 theVTab = thetable2.GetVTab

 theBitmap2 = theVTab.GetSelection
 selcount2 = theBitmap2.Count
 if(selcount2 <> 0)then
  'theField = theVTab.FindField("Bus_stop")  'from dbf

 theTheme = theView.FindTheme("Bus Stops") 'fTheme
 anFTab = theTheme.GetFTab
 theBitmap = anFTab.GetSelection
 aField = anFTab.FindField("On_street")
 rec = theBitmap.GetNextSet(-1)
 aField2 = anFTab.ReturnValue(aField, rec)
```

```
  myquerystring = "[On_street] = " + aField2.Quote
  theVTab.Query(myQueryString, theBitmap2, #VTAB_SELTYPE_AND)
  theVTab.UpdateSelection

  else
   thetable3 = av.FindDoc("sunday.dbf")
   theVTab = thetable3.GetVTab

   theBitmap3 = theVTab.GetSelection
   selcount3 = theBitmap3.Count
   if(selcount3 <> 0)then
    'theField = theVTab.FindField("Bus_stop")  'from dbf

    theTheme = theView.FindTheme("Bus Stops") 'fTheme
    anFTab = theTheme.GetFTab
    theBitmap2 = anFTab.GetSelection
    aField = anFTab.FindField("On_street")
    rec = theBitmap2.GetNextSet(-1)
    aField2 = anFTab.ReturnValue(aField, rec)
    myquerystring = "[On_street] = " + aField2.Quote
    theVTab.Query(myQueryString, theBitmap3, #VTAB_SELTYPE_AND)
    theVTab.UpdateSelection

   else
    'do nothing
   end
  end
end

'runs the merge script to find closest bus stop with a scheduled time

av.run("merge", "")

'runs the time script to find the time difference between the origin
'bus stop and a bus stop with a schedule

av.run("time", "")

'runs the stats script to solve the travel time and produce an output

av.run("stats", "")
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       NetRun
' Script Description: Finds a route between the chosen locations.
'
' Programmer:        Sarah Riley
' Date Created:      Nov. 2000
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    30-Apr-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'get the view and the location theme
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theTheme=theView.FindTheme("Chosen Locations")
theFtab=theTheme.getFtab

theStreetFTab=theView.FindTheme("Transit routes.shp").getFtab
theNetDef=NetDef.Make(theStreetFTab)
theNetwork=Network.Make(theNetDef)

thePointList={}
thePointField=theFtab.FindField("Shape")
for each rec in theFtab
  p=theFtab.ReturnValue(thePointField, rec)
  thePointList.add(p)
end

'*************
' Set the cost for solving a routing problem
' It then solves the problem and displays
' directions for the route.

' Get a list of available costs for this line theme. Choose a cost from the list.
'
aNetCostFieldList = theNetDef.GetCostFields
aNetCostField = MsgBox.Choice(aNetCostFieldList, "Select Minutes", "Select Minutes")

' If a cost was chosen, set it.

if (aNetCostField<>nil) then
  theNetwork.SetCostField(aNetCostField)
end

'The above can be replaced by changing the hard code for Network Analyst wherein
'there will only be the chosen cost of MINUTES to select.
'****************


TravelDistance= theNetwork.FindPath(thePointList, True, False)
'msgBox.Info( TravelDistance.AsString, "Total"++reportUnitStr )

'theView.GetGraphics.add(GraphicShape.Make(theNetwork.ReturnPathShape))

aSymbol = Symbol.Make(#SYMBOL_PEN)
aSymbol.SetSize(2.0)
aSymbol.SetColor(Color.GetMagenta)

theGL=theView.GetGraphics
theRoute=GraphicShape.Make(theNetwork.ReturnPathShape)
theRoute.SetDisplay(av.GetActiveDoc.GetDisplay)
theroute.SetSymbol(aSymbol)
theGL.Add(theRoute)
'theroute.GetSymbol.SetColor(color.GetMagenta).SetSymbol(aSymbol)
theRoute.Draw
theroute.Invalidate
```

```
'set view size for public viewing

theView = av.getproject.finddoc("Route Finding")
theWin = theView.GetWin
'theWin.Resize(1025,600)
'theWin.Moveto(0,0)
theWin.Open

' Write the route result FTab, this is necessary to get directons.
'
tmpFileName = FileName.Make("F:\thesis data\Route.shp")
theNetwork.WritePath(tmpFileName)
resFTab = FTab.Make(SrcName.Make(tmpFileName.AsString))

' Show the directions then remove the temporary file.
'
'theDirections="Your route directions are as follows"+NL+NL+
'            theNetwork.GetPathDirections(resFTab)
'msgBox.Report(theDirections, "Route Information")

'add theme to view
newTheme = FTheme.Make(resFTab)
theView.AddTheme(newTheme)
newTheme.SetName("Route")
newTheme.SetVisible(True)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       nearfeatures
' Script Description: Finds the nearest feature to points in a point theme
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:   Table with the following items: a 'From' field,
'                    a 'To' field, and a distance to nearest feature field

'
' Changes:
' Date           Details
' 29-Apr-2002    Initial Implementation
'
' Author
' Timothy J Fox
' ESRI
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


''''''''''''''''''''''''''''''''
' Set search radius and other vars...
''''''''''''''''''''''''''''''''

shortDist = 1000000000
v = av.GetActiveDoc
tl = v.getThemes
shortTRec = 0
theCount = 0
w = "Warning"
''''''''''''''''''''''''''''''''''''''''''''''
'Get 'From' theme and field - check for 'From' selection, and field
''''''''''''''''''''''''''''''''''''''''''''''

f = v.FindTheme("Chosen Locations")
'f = msgbox.list (tl, "Select the theme that contains" + NL + "the 'From' features." ,"'From' theme")
'if (f = nil) then
'  exit
'end
fromName = f.GetName
fromShpFld = f.GetFTab.FindField("Shape")
'fFieldList = f.GetFTab.GetFields
'ff = msgbox.multilist (fFieldList, "Select 'From' field to preserve" ,"'From' field")

Field1 = f.GetFTab.FindField("ID")
Field2 = f.GetFTab.FindField("Address")
'fFieldList = {Field1, Field2}

'if (ff = nil) then
'  msgbox.warning("you must select a field" ,w)
'  exit
'end
if (f.GetFTab.GetSelection.Count = 0) then
   fromFeatures = f.GetFTab
   fromTotal = f.GetFTab.GetNumRecords
  else
   fromFeatures = f.GetFTab.GetSelection
   fromTotal = fromFeatures.Count
end

''''''''''''''''''''''''''''''''''''''
'get 'To' theme and field
''''''''''''''''''''''''''''''''''''''


t = v.FindTheme("Bus Stops")
't = msgbox.list (tl, "Select 'To' theme", "'To' theme")
'if (t = nil) then
'  exit
'end
toName = t.GetName
toShpFld  = t.GetFTab.FindField("Shape")
'tFieldList = t.GetFTab.GetFields
```

```
'tf = msgbox.multilist (tFieldList, "Select field to preserve", "'To' field")
'if (tf = nil) then
'  msgbox.warning("you must select a field", w)
'  exit
'end

 Field3 = t.GetFTab.FindField("Bus_stop")
 Field4 = t.GetFTab.FindField("On_street")
 Field5 = t.GetFTab.FindField("At_street")
'tFieldList = {Field3, Field4, Field5}


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'get 'To' Theme's selection
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
if (t.GetFTab.GetSelection.Count = 0) then
  toFeatures = t.GetFTab
  else
  toFeatures = t.GetFTab.GetSelection
end
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'create new table
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
tempFilename = FileName.Make("F:thesis data\near.dbf")
'theFilename  = FileDialog.Put(tempFilename, "*.dbf", "Save Table where?")
'if (theFilename = nil) then
'  exit
'end
tempFilename.SetExtension("dbf")
'theFilename.SetExtension("dbf")
'newVTab = VTab.MakeNew (theFilename, dBASE
newVTab = VTab.MakeNew (tempFilename, dBASE)
'fID = ff.clone
'tID = tf.clone
f1 = Field1.clone
f2 = Field2.clone
t1 = Field3.clone
t2 = Field4.clone
t3 = Field5.clone
distField = Field.Make("NF Dist",#FIELD_DECIMAL,8,2)
newFields = {f1, f2, t1, t2, t3, distField}
newVTab.AddFields(newFields)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Find the closest feature to the point...
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
av.ShowMsg("NF analysis: From"+ fromName + " To " + toName )
for each frec in fromFeatures
  fPoint = f.GetFTab.ReturnValue(fromShpFld,frec)
  theCount = theCount + 1
  for each trec in ToFeatures
    tShape = t.GetFTab.ReturnValue(toShpFld,trec)
    dist = tShape.Distance(fPoint)
    if (dist < shortDist) then
      shortDist = dist
      shortTRec = trec.Clone
    end
  end
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''
  'Write to table: 'distance','From' and 'To' fields
  ''''''''''''''''''''''''''''''''''''''''''''''''''''''

  'tempTid = t.GetFTab.ReturnValue(tf,shortTRec)
  tempFid1 = f.GetFTab.ReturnValue(Field1,frec)
  tempFid2 = f.GetFTab.ReturnValue(Field2,frec)
  tempTid3 = t.GetFTab.ReturnValue(Field3,shortTRec)
  tempTid4 = t.GetFTab.ReturnValue(Field4,shortTRec)
  tempTid5 = t.GetFTab.ReturnValue(Field5,shortTRec)

  newVTab.SetEditable (true)
  newRec = newVTab.AddRecord
  newVTab.SetValue(f1,newRec,tempFid1)
```

```
 newVTab.SetValue(f2,newRec,tempFid2)
 newVTab.SetValue(t1,newRec,tempTid3)
 newVTab.SetValue(t2,newRec,tempTid4)
 newVTab.SetValue(t3,newRec,tempTid5)
 'newVTab.SetValue(tf, newRec, tempTid)
 newVTab.SetValue(distField,newRec,shortDist)
 shortDist = 1000000000
 av.SetStatus((theCount / fromTotal) * 100)
end
av.ClearStatus
av.ClearMsg
newVTab.SetEditable (false)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' create table from VTab & display table
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
newTab = table.Make (newVTab)
newTab.setName ("near.dbf")
newTabWin = newTab.GetWin
newTabWin.Resize (350, 250)
newTabWin.Open
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      SelByTheme
' Script Description: Performs a select by theme based on the chosen
'                  bus stops and finds the stop within the
'                  schedule table.
'
' Programmer:        Sarah Riley
' Date Created:      Apr. 2002
' Returned Object:   none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley    2-May-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'run the following to find closest bus stop to the
'origin bus stop in order to get a time of the
'bus arrival

'It can be done using this request.
'Features will be selected from aTheme
'based on anotherTheme.

aView = av.GetProject.FindDoc("Route Finding")
theWin = aView.GetWin
theWin.Open

theThemeList=aView.getThemes
for each i in (0..(theThemeList.count-1))
  temp=theThemeList.get(i).setActive(FALSE)
end

aTheme = aView.FindTheme("Bus Stops")
aTheme.SetActive(TRUE)
anotherTheme = aView.FindTheme("Chosen Locations")
theFTab = anotherTheme.GetFTab

'get the first selected record for merging
if(theFTab = nil)then
 MsgBox.Info("No table...", "")
 exit
end

theBitmap = Bitmap.Make(theFTab.GetNumRecords)
for each rec in (0..(theFTab.GetNumRecords-1)) by 2
 theBitmap.Set(rec)
end
theFTab.SetSelection(theBitmap)
theFTab.UpdateSelection


'msgbox.info(av.GetActiveDoc.GetDisplay.GetUnits.AsString, "")
'
aDistance = Units.Convert(1.5, #UNITS_LINEAR_KILOMETERS, av.GetActiveDoc.GetDisplay.GetUnits )


aRelType = #FTAB_RELTYPE_ISWITHINDISTANCEOF
'aDistance = 1 'make sure this is km else it would be meters

'---this can change--maybe do a for statement until a bus stop is found

aSelType = #VTAB_SELTYPE_AND

aTheme.SelectByTheme(anotherTheme, aRelType, aDistance, aSelType)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       merge
' Script Description: Merges Chosen Locations and Bus Stop themes
'                  in order to find the closest bus stop with
'                  a scheduled time.
'
' Programmer:        Sarah Riley
' Date Created:      Nov. 2000
' Returned Object:   none
'
' Changes:
'
' Author      Date         Details
'
' S. Riley    30-Apr-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'get the view and the location theme
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Chosen Locations")
theFTab = theTheme.GetFTab

'get the first selected record for merging
if(theFTab = nil)then
  MsgBox.Info("No table...", "")
  exit
end

theBitmap = Bitmap.Make(theFTab.GetNumRecords)
for each rec in (0..(theFTab.GetNumRecords-1)) by 2
  theBitmap.Set(rec)
end
theFTab.SetSelection(theBitmap)
theFTab.UpdateSelection

themesToMerge = List.Make
themesToMerge.add(theView.FindTheme("Chosen Locations"))
themesToMerge.add(theView.FindTheme("Bus Stops"))
themesToMerge.add(theView.FindTheme("Chosen Locations"))


' Themes must have matching shape types for merging.  Using the first
' active theme verify that this is the case...

checkType  = themesToMerge.Get(0).GetFtab.FindField("Shape").GetType

for each i in 1 .. (themesToMerge.Count - 1)
  t = themesToMerge.Get(i)
  if (checkType <> t.GetFTab.FindField("Shape").GetType) then
    MsgBox.Error("Theme feature type mismatch - Unable to merge.","")
    exit
  end
end

' Specify the output shapefile...

'outFName = av.GetProject.MakeFileName("Time", "shp")
theCWD=FileName.GetCWD
outFName=theCWD.makeTmp("Time","shp")

' Create the list of fields used for the output theme.  The fields
' are taken from the first active theme only, it is assumed that
' other themes have an identical set of fields.  If this is not the
' case the themes will still be merged, however fields not found in
' other themes will be empty...

fieldList = List.Make
```

```
for each f in themesToMerge.Get(0).GetFTab.GetFields
  if (f.GetName = "Shape") then
    continue
  else
    fCopy = f.Clone
    fieldList.Add(fCopy)
  end
end

' Get the class of new FTab to create, create the new FTab and
' add fields that we've gathered from the input themes....

shapeType = themesToMerge.Get(0).GetFTab.FindField("Shape").GetType
outClass = POINT

mergeFTab = FTab.MakeNew( outFName, outClass )
if (fieldList.Count > 0) then
  mergeFTab.AddFields( fieldList )
end

' Populate the new FTab from the FTabs of the input themes...

for each t in themesToMerge
  av.ShowMsg( "Merging"++t.GetName )
  inFTab  = t.GetFTab
  if (inFTab.GetSelection.Count = 0) then
    theRecordsToMerge = inFTab
    numRecs = inFTab.GetNumRecords
  else
    theRecordsToMerge = inFTab.GetSelection
    numRecs = theRecordsToMerge.Count
  end

  for each rec in theRecordsToMerge
    av.SetStatus( (rec / numRecs) * 100 )
    newRec   = mergeFTab.AddRecord
    inField  = inFTab.FindField( "Shape" )
    outField = mergeFTab.FindField( "Shape" )
    mergeFTab.SetValue( outField, newrec, inFTab.ReturnValue( inField, rec ))

    if (fieldList.Count > 0) then
      for each f in fieldList
        fName    = f.GetName
        inField  = inFTab.FindField( fName )

        ' Skip field if not found in inFTab...
        if ( inField <> Nil ) then
          outField = mergeFTab.FindField( fName )
          aValue   = inFTab.ReturnValue( inField, rec )
          mergeFTab.SetValue( outField, newRec, aValue )
        end
      end  ' for each f
    end  ' if count
  end  ' for each rec
end  ' for each t

av.ClearMsg
av.ClearStatus

' add the new theme to the view

mergeTheme = FTheme.Make( mergeFTab )
theView.AddTheme(mergeTheme)
mergeTheme.SetName("Time")
mergeTheme.SetVisible(True)

'show the view
theView = av.GetProject.FindDoc("Route Finding")
theWin = theView.GetWin
theWin.Open
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       time
' Script Description: Finds the time from the origin bus stop and
'                   the bus stop with a schedule.
'
' Programmer:        Sarah Riley
' Date Created:      Nov. 2000
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    30-Apr-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'get the view and the time theme
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Time")
theFtab = theTheme.getFtab

theStreetFTab = theView.FindTheme("Transit routes.shp").getFtab
theNetDef = NetDef.Make(theStreetFTab)
theNetwork = Network.Make(theNetDef)

thePointList={}
thePointField=theFtab.FindField("Shape")
for each rec in theFtab
  p=theFtab.ReturnValue(thePointField, rec)
  thePointList.add(p)
end

'*************
' Set the cost for solving a routing problem
' It then solves the problem and displays directions for the route.

' Get a list of available costs for this line theme. Choose a cost from the list.
'
aNetCostFieldList = theNetDef.GetCostFields
aNetCostField = MsgBox.Choice(aNetCostFieldList, "Select Minutes", "Select Minutes")

' If a cost was chosen, set it.

if (aNetCostField<>nil) then
  theNetwork.SetCostField(aNetCostField)
end

'The above can be replaced by changing the hard code for Network Analyst wherein
'there will only be the chosen cost of MINUTES to select.
'****************
TravelDistance= theNetwork.FindPath(thePointList, True, False)
'msgBox.Info( TravelDistance.AsString, "Total"++reportUnitStr )

'theView.GetGraphics.add(GraphicShape.Make(theNetwork.ReturnPathShape))

'aSymbol = Symbol.Make(#SYMBOL_PEN)
'aSymbol.SetSize(2.0)
'aSymbol.SetColor(Color.GetMagenta)
'
'theGL=theView.GetGraphics
'theRoute=GraphicShape.Make(theNetwork.ReturnPathShape)
'theRoute.SetDisplay(av.GetActiveDoc.GetDisplay)
'theroute.SetSymbol(aSymbol)
'theGL.Add(theRoute)
''theroute.GetSymbol.SetColor(color.GetMagenta).SetSymbol(aSymbol)
'theRoute.Draw
'theroute.Invalidate
```

```
''set view size for public viewing
'
theView = av.getproject.finddoc("Route Finding")
theWin = theView.GetWin
'theWin.Resize(1025,600)
'theWin.Moveto(0,0)
theWin.Open

' Write the route result FTab, this is necessary to get directons.
'
tmpFileName = FileName.Make("F:\thesis data\route.shp")
theNetwork.WritePath(tmpFileName)
resFTab = FTab.Make(SrcName.Make(tmpFileName.AsString))
'
'' Show the directions then remove the temporary file.
''
'theDirections="Your route directions are as follows"+NL+NL+
'            theNetwork.GetPathDirections(resFTab)
'msgBox.Report(theDirections, "Route Information")

'make the Time theme invisible
theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Time")
'theTheme.SetActive(FALSE)
theTheme.SetVisible(FALSE)

'add theme to view
newTheme = FTheme.Make(resFTab)
theView.AddTheme(newTheme)
newTheme.SetName("Route2")
newTheme.SetVisible(True)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:      stats
' Script Description: Performs the Bayesian statistics, reports
'                   the results in a report message box and
'                   zooms to the route in the view.
'
' Programmer:       Sarah Riley
' Date Created:     Apr. 2002
' Returned Object:  none
'
' Changes:
'
' Author      Date          Details
'
' S. Riley   15-Jun-2002     Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'get the view and the route theme
theView = av.GetActiveDoc
theTheme = theView.FindTheme("Route")
theTable = theTheme.GetFTab

'set up table for selection
thebitmap = theTable.GetSelection
expr = "[F_cost] = 0"
theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
theTable.UpdateSelection ' highlight found records

theField = theTable.FindField("T_cost")

rec = theBitMap.GetNextSet(-1) 'first record in bitmap
theField2 = theTable.ReturnValue(theField, rec) 'fill values


'msgbox.info(theField2.AsString, " ")

'aBitmap.ClearAll

'now perform the statistics on this value
'perform the statistics on theField2.AsString


TravelTime = theField2

T1 = TravelTime - 1
T2 = TravelTime
T3 = TravelTime + 3

'all in minutes

'route = route number from table

' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
' since each table is cleared at the end, only one table should have records highlighted

dbTable1 = av.FindDoc("weekday.dbf")
theVTab1 = dbTable1.GetVTab
theSelection1 = theVTab1.GetSelection

selCount1 = theSelection1.Count

if (selCount1 <> 0) then

'call on the route number from the field "Route_no"

theField1 = theVTab1.FindField("Route_no")
rec = theSelection1.GetNextSet(-1)  'first record in the selection
aField = theVTab1.ReturnValue(theField1, rec)
```

```
if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

 'use transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.255
    P2 = 0.72
    P3 = 0.025
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.165
      P2 = 0.705
      P3 = 0.13
    else
      P1 = 0.20
      P2 = 0.705
      P3 = 0.095
    end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.26
   P2 = 0.64
   P3 = 0.10
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.25
     P2 = 0.655
     P3 = 0.095
   else
     P1 = 0.25
     P2 = 0.635
     P3 = 0.115
   end
 elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
   P1 = 0.21
   P2 = 0.645
   P3 = 0.145
 else
   P1 = 0.24
   P2 = 0.645
   P3 = 0.115  'all day value
 end

else

'use non-transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
   P1 = 0.225
   P2 = 0.71
   P3 = 0.065
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
   if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
     P1 = 0.195
     P2 = 0.555
     P3 = 0.25
   else
     P1 = 0.24
     P2 = 0.60
     P3 = 0.16
   end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.34
   P2 = 0.585
   P3 = 0.075
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.34
     P2 = 0.525
```

```
      P3 = 0.135
    else
      P1 = 0.405
      P2 = 0.49
      P3 = 0.105
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.305
    P2 = 0.565
    P3 = 0.13
  else
    P1 = 0.325
    P2 = 0.56
    P3 = 0.115  'all day values
  end
end

' read the results from the schedule tables

theTable2 = av.FindDoc("weekday.dbf")
theVTab2 = theTable2.GetVTab
theBitmap2 = theVTab2.GetSelection

theVTab2.UpdateSelection
'theVTab2.PromoteSelection
aField4 = theVTab2.FindField("Route_no")
aField5 = theVTab2.FindField("Time")
rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

dbTable2 = av.FindDoc("saturday.dbf")
theVTab2 = dbTable2.GetVTab
theSelection2 = theVTab2.GetSelection

selCount2 = theSelection2.GetSelection

if (selCount2 <> 0) then
 'call on the route number from the field "Route_no"

 theField2 = theVTab2.FindField("Route_no")
 rec = theSelection2.GetNextSet(-1) 'first record in bitmap
 aField = theVTab2.ReturnValue(theField2, rec) 'fill values

 if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

  'use transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.245
    P2 = 0.745
    P3 = 0.01
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.35
    P2 = 0.615
    P3 = 0.035
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.26
    P2 = 0.68
    P3 = 0.06
  elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
    P1 = 0.23
    P2 = 0.645
    P3 = 0.125
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.22
```

```
      P2 = 0.685
      P3 = 0.095
    else
      P1 = 0.26
      P2 = 0.66
      P3 = 0.08  'all day values
    end

  else

   'use non transitway stats
    if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
      P1 = 0.37
      P2 = 0.50
      P3 = 0.13
    elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
      P1 = 0.215
      P2 = 0.655
      P3 = 0.13
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
      P1 = 0.23
      P2 = 0.64
      P3 = 0.13
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.24
      P2 = 0.51
      P3 = 0.25
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.32
      P2 = 0.60
      P3 = 0.08
    else
      P1 = 0.265
      P2 = 0.595
      P3 = 0.14  'all day values
    end
  end

  ' read the results from the schedule tables

  theTable3 = av.FindDoc("saturday.dbf")
  theVTab3 = theTable3.GetVTab
  theBitmap3 = theVTab3.GetSelection

  theVTab3.UpdateSelection
  'theVTab3.PromoteSelection
  aField4 = theVTab3.FindField("Route_no")
  aField5 = theVTab3.FindField("Time")
  rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
  dbTable3 = av.FindDoc("sunday.dbf")
  theVTab3 = dbTable3.GetVTab
  theSelection3 = theVTab3.GetSelection

  selCount3 = theSelection3.GetSelection

  if(selCount3 <> 0) then
   'call on the route number from the field "Route_no"

   theField3 = theVTab3.FindField("Route_no")
   rec = theSelection3.GetNextSet(-1)  'first record in the selection
   aField = theVTab3.ReturnValue(theField3, rec)

   if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
     or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then
```

```
  'use transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.51
    P2 = 0.365
    P3 = 0.125
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.27
    P2 = 0.68
    P3 = 0.05
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.155
    P2 = 0.74
    P3 = 0.105
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.13
    P2 = 0.715
    P3 = 0.155
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.19
    P2 = 0.75
    P3 = 0.06
  else
    P1 = 0.18
    P2 = 0.715
    P3 = 0.105  'all day values
  end

 else
  'use non transitway stats

  if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.32
    P2 = 0.605
    P3 = 0.075
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.21
    P2 = 0.665
    P3 = 0.125
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.30
    P2 = 0.62
    P3 = 0.08
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.35
    P2 = 0.565
    P3 = 0.085
  else
    P1 = 0.285
    P2 = 0.615
    P3 = 0.10  'all day values
  end
 end

' read the results from the schedule tables

theTable4 = av.FindDoc("sunday.dbf")
theVTab4 = theTable4.GetVTab
theBitmap4 = theVTab4.GetSelection

theVTab4.UpdateSelection
'theVTab4.PromoteSelection
aField4 = theVTab4.FindField("Route_no")
aField5 = theVTab4.FindField("Time")
rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values
```

```
    else
    'MsgBox.Info("No tables have a selection....", "")
    end
  end
end


'the new expected time is calculated

ExTime = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

'work through Bayesian Probabilities
'discete values are given
'therefore use discrete case
'

'check for advisory

aTheme = theView.FindTheme("Transit routes.shp")
anFTab = aTheme.GetFTab

'aBitmap = anFTab.GetSelection
'
'aCount = aBitmap.Count
'
'if (aCount <> 0) then
'  MsgBox.Info("Selection", "")
'else
'  aBitmap.SetAll
'end

aBitmap = Bitmap.Make(anFTab.GetNumRecords)
for each rec in (0..(anFTab.GetNumRecords-1)) by 2033
  aBitmap.Set(rec)
end
anFTab.SetSelection(aBitmap)
anFTab.UpdateSelection

AdvisField = anFTab.FindField("Advisory")

rec = aBitMap.GetNextSet(-1) 'first record in bitmap
AdvField = anFTab.ReturnValue(AdvisField, rec) 'fill values


if (AdvField = 0)then  'no advisory
  Time = ExTime

else  'advisory

'conditional probabilities

'P(early|early)
EE = 0.7

'P(early|ontime)
EO = 0.25

'P(early|late)
EL = 0.05

'P(ontime|early)
OE = 0.15

'P(ontime|ontime)
OO = 0.7

'P(ontime|late)
OL = 0.15
```

```
'P(late|early)
LE = 0.05

'P(late|ontime)
LO = 0.25

'P(late|late)
LL = 0.7

'calculate the denominators for the Bayesian equation

early = (P1*EE) + (P2*OE) + (P3*LE)
ontime = (P1*EO) + (P2*OO) + (P3*LO)
late = (P1*EL) + (P2*OL) +(P3*LL)

'need to find out what condition the buses are running
'these will be posterior probabilities
'OC TRANSPO MUST ENTER A CODE TO STATE HOW BUSES ARE RUNNING ie. R1, R2, R3
'R1 = EARLY
'R2 = ONTIME
'R3 = LATE

aTheme = theView.FindTheme("Transit routes.shp")
anFTab = aTheme.GetFTab

'aBitmap = anFTab.GetSelection
'
'aCount = aBitmap.Count
'
'if (aCount <> 0) then
' MsgBox.Info("Selection", "")
'else
' aBitmap.SetAll
'end

aBitmap = Bitmap.Make(anFTab.GetNumRecords)
for each rec in (0..(anFTab.GetNumRecords-1)) by 2033
  aBitmap.Set(rec)
end
anFTab.SetSelection(aBitmap)
anFTab.UpdateSelection

ConField = anFTab.FindField("Condition")

rec = aBitMap.GetNextSet(-1) 'first record in bitmap
condField = anFTab.ReturnValue(ConField, rec) 'fill values

if (condField.AsString = "R2") then 'ontime

   'P(ontime|ontime)
   time1 = (P2*OO)/ontime

   'P(early|ontime)
   time2 = (P1*EO)/ontime

   'P(ontime|ontime)
   time3 = (P3*LO)/ontime

   Time = (time1*T2) + (time2*T1) + (time3*T3)


   'rerun routing to determine if there is another route
   av.run("netrun3","")

   'get the view and the route theme
   theView = av.GetActiveDoc
   theTheme = theView.FindTheme("Route1")
   theTable = theTheme.GetFTab

   'set up table for selection
```

```
thebitmap = theTable.GetSelection
expr = "[F_cost] = 0"
theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
theTable.UpdateSelection ' highlight found records

theField = theTable.FindField("T_cost")

rec = theBitMap.GetNextSet(-1) 'first record in bitmap
theField2 = theTable.ReturnValue(theField, rec) 'fill values


'msgbox.info(theField2.AsString, " ")

'aBitmap.ClearAll

'now perform the statistics on this value
'perform the statistics on theField2.AsString


TravelTime = theField2

T1 = TravelTime - 1
T2 = TravelTime
T3 = TravelTime + 3

'all in minutes

'route = route number from table

' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
' since each table is cleared at the end, only one table should have records highlighted

dbTable1 = av.FindDoc("weekday.dbf")
theVTab1 = dbTable1.GetVTab
theSelection1 = theVTab1.GetSelection

selCount1 = theSelection1.Count

if (selCount1 <> 0) then

'call on the route number from the field "Route_no"

theField1 = theVTab1.FindField("Route_no")
rec = theSelection1.GetNextSet(-1)  'first record in the selection
aField = theVTab1.ReturnValue(theField1, rec)

if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

'use transitway stats

if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
   P1 = 0.255
   P2 = 0.72
   P3 = 0.025
elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
   if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
     P1 = 0.165
     P2 = 0.705
     P3 = 0.13
   else
     P1 = 0.20
     P2 = 0.705
     P3 = 0.095
   end
elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
  P1 = 0.26
  P2 = 0.64
  P3 = 0.10
elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
```

```
    if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
      P1 = 0.25
      P2 = 0.655
      P3 = 0.095
    else
      P1 = 0.25
      P2 = 0.635
      P3 = 0.115
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.21
    P2 = 0.645
    P3 = 0.145
  else
    P1 = 0.24
    P2 = 0.645
    P3 = 0.115  'all day value
  end

else

'use non-transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.225
    P2 = 0.71
    P3 = 0.065
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.195
      P2 = 0.555
      P3 = 0.25
    else
      P1 = 0.24
      P2 = 0.60
      P3 = 0.16
    end
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.34
    P2 = 0.585
    P3 = 0.075
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
      P1 = 0.34
      P2 = 0.525
      P3 = 0.135
    else
      P1 = 0.405
      P2 = 0.49
      P3 = 0.105
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
     P1 = 0.305
     P2 = 0.565
     P3 = 0.13
  else
     P1 = 0.325
     P2 = 0.56
     P3 = 0.115  'all day values
  end
end

' read the results from the schedule tables

theTable2 = av.FindDoc("weekday.dbf")
theVTab2 = theTable2.GetVTab
theBitmap2 = theVTab2.GetSelection

theVTab2.UpdateSelection
'theVTab2.PromoteSelection
```

```
  aField4 = theVTab2.FindField("Route_no")
  aField5 = theVTab2.FindField("Time")
  rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

  dbTable2 = av.FindDoc("saturday.dbf")
  theVTab2 = dbTable2.GetVTab
  theSelection2 = theVTab2.GetSelection

  selCount2 = theSelection2.GetSelection

  if (selCount2 <> 0) then
   'call on the route number from the field "Route_no"

   theField2 = theVTab2.FindField("Route_no")
   rec = theSelection2.GetNextSet(-1) 'first record in bitmap
   aField = theVTab2.ReturnValue(theField2, rec) 'fill values

   if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
    or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

     'use transitway stats

      if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
        P1 = 0.245
        P2 = 0.745
        P3 = 0.01
      elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
        P1 = 0.35
        P2 = 0.615
        P3 = 0.035
      elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
        P1 = 0.26
        P2 = 0.68
        P3 = 0.06
      elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
        P1 = 0.23
        P2 = 0.645
        P3 = 0.125
      elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
        P1 = 0.22
        P2 = 0.685
        P3 = 0.095
      else
        P1 = 0.26
        P2 = 0.66
        P3 = 0.08  'all day values
      end

   else

     'use non transitway stats
      if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
        P1 = 0.37
        P2 = 0.50
        P3 = 0.13
      elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
        P1 = 0.215
        P2 = 0.655
        P3 = 0.13
      elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
        P1 = 0.23
        P2 = 0.64
        P3 = 0.13
      elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
        P1 = 0.24
```

```
      P2 = 0.51
      P3 = 0.25
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.32
      P2 = 0.60
      P3 = 0.08
    else
      P1 = 0.265
      P2 = 0.595
      P3 = 0.14  'all day values
    end
  end

  ' read the results from the schedule tables

  theTable3 = av.FindDoc("saturday.dbf")
  theVTab3 = theTable3.GetVTab
  theBitmap3 = theVTab3.GetSelection

  theVTab3.UpdateSelection
  'theVTab3.PromoteSelection
  aField4 = theVTab3.FindField("Route_no")
  aField5 = theVTab3.FindField("Time")
  rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
  dbTable3 = av.FindDoc("sunday.dbf")
  theVTab3 = dbTable3.GetVTab
  theSelection3 = theVTab3.GetSelection

  selCount3 = theSelection3.GetSelection

  if(selCount3 <> 0) then
   'call on the route number from the field "Route_no"

   theField3 = theVTab3.FindField("Route_no")
   rec = theSelection3.GetNextSet(-1)  'first record in the selection
   aField = theVTab3.ReturnValue(theField3, rec)

   if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
     or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

    'use transitway stats

    if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
      P1 = 0.51
      P2 = 0.365
      P3 = 0.125
    elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
      P1 = 0.27
      P2 = 0.68
      P3 = 0.05
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
      P1 = 0.155
      P2 = 0.74
      P3 = 0.105
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.13
      P2 = 0.715
      P3 = 0.155
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.19
      P2 = 0.75
      P3 = 0.06
    else
      P1 = 0.18
```

```
        P2 = 0.715
        P3 = 0.105  'all day values
      end

    else
     'use non transitway stats

      if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
        P1 = 0.32
        P2 = 0.605
        P3 = 0.075
      elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
        P1 = 0.21
        P2 = 0.665
        P3 = 0.125
      elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
        P1 = 0.30
        P2 = 0.62
        P3 = 0.08
      elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
        P1 = 0.35
        P2 = 0.565
        P3 = 0.085
      else
        P1 = 0.285
        P2 = 0.615
        P3 = 0.10  'all day values
      end
    end

   ' read the results from the schedule tables

   theTable4 = av.FindDoc("sunday.dbf")
   theVTab4 = theTable4.GetVTab
   theBitmap4 = theVTab4.GetSelection

   theVTab4.UpdateSelection
   'theVTab4.PromoteSelection
   aField4 = theVTab4.FindField("Route_no")
   aField5 = theVTab4.FindField("Time")
   rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
   result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
   rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
   result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values


   else
   'MsgBox.Info("No tables have a selection....", "")
   end
  end
end


'the new expected time is calculated

ExTime2 = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

if (Time < ExTime2)then
  TrvlTime = Time

elseif(Time > ExTime2)then
  TrvlTime = ExTime2
else
  TrvlTime = Time
end


elseif(condField.AsString = "R1")then  'early

  'P(ontime|early)
```

```
time1 = (P2*OE)/early

'P(early|early)
time2 = (P1*EE)/early

'P(late|early)
time3 = (P3*LE)/early

ExTime = (time1*T2) + (time2*T1) + (time3*T3)
'ExTime = ExTime1.Round

time = ExTime

'rerun routing to determine if there is another route
 av.run("netrun3","")

 'get the view and the route theme
 theView = av.GetActiveDoc
 theTheme = theView.FindTheme("Route1")
 theTable = theTheme.GetFTab

 'set up table for selection
 thebitmap = theTable.GetSelection
 expr = "[F_cost] = 0"
 theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
 theTable.UpdateSelection ' highlight found records

 theField = theTable.FindField("T_cost")

 rec = theBitMap.GetNextSet(-1) 'first record in bitmap
 theField2 = theTable.ReturnValue(theField, rec) 'fill values


 'msgbox.info(theField2.AsString, " ")

 'aBitmap.ClearAll

 'now perform the statistics on this value
 'perform the statistics on theField2.AsString


 TravelTime = theField2

 T1 = TravelTime - 1
 T2 = TravelTime
 T3 = TravelTime + 3

 'all in minutes

 'route = route number from table

 ' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
 ' since each table is cleared at the end, only one table should have records highlighted

 dbTable1 = av.FindDoc("weekday.dbf")
 theVTab1 = dbTable1.GetVTab
 theSelection1 = theVTab1.GetSelection

 selCount1 = theSelection1.Count

 if (selCount1 <> 0) then

 'call on the route number from the field "Route_no"

 theField1 = theVTab1.FindField("Route_no")
 rec = theSelection1.GetNextSet(-1)  'first record in the selection
 aField = theVTab1.ReturnValue(theField1, rec)

if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then
```

```
 'use transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.255
    P2 = 0.72
    P3 = 0.025
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.165
      P2 = 0.705
      P3 = 0.13
    else
      P1 = 0.20
      P2 = 0.705
      P3 = 0.095
    end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.26
   P2 = 0.64
   P3 = 0.10
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.25
     P2 = 0.655
     P3 = 0.095
   else
     P1 = 0.25
     P2 = 0.635
     P3 = 0.115
   end
 elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
   P1 = 0.21
   P2 = 0.645
   P3 = 0.145
 else
   P1 = 0.24
   P2 = 0.645
   P3 = 0.115  'all day value
 end

else

'use non-transitway stats

 if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
   P1 = 0.225
   P2 = 0.71
   P3 = 0.065
 elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
   if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
     P1 = 0.195
     P2 = 0.555
     P3 = 0.25
   else
     P1 = 0.24
     P2 = 0.60
     P3 = 0.16
   end
 elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
   P1 = 0.34
   P2 = 0.585
   P3 = 0.075
 elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
   if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
     P1 = 0.34
     P2 = 0.525
     P3 = 0.135
   else
     P1 = 0.405
```

```
        P2 = 0.49
        P3 = 0.105
      end
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.305
      P2 = 0.565
      P3 = 0.13
    else
      P1 = 0.325
      P2 = 0.56
      P3 = 0.115  'all day values
    end
  end

  ' read the results from the schedule tables

  theTable2 = av.FindDoc("weekday.dbf")
  theVTab2 = theTable2.GetVTab
  theBitmap2 = theVTab2.GetSelection

  theVTab2.UpdateSelection
  'theVTab2.PromoteSelection
  aField4 = theVTab2.FindField("Route_no")
  aField5 = theVTab2.FindField("Time")
  rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

  dbTable2 = av.FindDoc("saturday.dbf")
  theVTab2 = dbTable2.GetVTab
  theSelection2 = theVTab2.GetSelection

  selCount2 = theSelection2.GetSelection

  if (selCount2 <> 0) then
   'call on the route number from the field "Route_no"

   theField2 = theVTab2.FindField("Route_no")
   rec = theSelection2.GetNextSet(-1) 'first record in bitmap
   aField = theVTab2.ReturnValue(theField2, rec) 'fill values

   if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
    or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

    'use transitway stats

     if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
       P1 = 0.245
       P2 = 0.745
       P3 = 0.01
     elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
       P1 = 0.35
       P2 = 0.615
       P3 = 0.035
     elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
       P1 = 0.26
       P2 = 0.68
       P3 = 0.06
     elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
       P1 = 0.23
       P2 = 0.645
       P3 = 0.125
     elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
       P1 = 0.22
       P2 = 0.685
       P3 = 0.095
     else
```

```
      P1 = 0.26
      P2 = 0.66
      P3 = 0.08  'all day values
    end

  else

    'use non transitway stats
    if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
      P1 = 0.37
      P2 = 0.50
      P3 = 0.13
    elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
      P1 = 0.215
      P2 = 0.655
      P3 = 0.13
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
      P1 = 0.23
      P2 = 0.64
      P3 = 0.13
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.24
      P2 = 0.51
      P3 = 0.25
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.32
      P2 = 0.60
      P3 = 0.08
    else
      P1 = 0.265
      P2 = 0.595
      P3 = 0.14  'all day values
    end
  end

  ' read the results from the schedule tables

  theTable3 = av.FindDoc("saturday.dbf")
  theVTab3 = theTable3.GetVTab
  theBitmap3 = theVTab3.GetSelection

  theVTab3.UpdateSelection
  'theVTab3.PromoteSelection
  aField4 = theVTab3.FindField("Route_no")
  aField5 = theVTab3.FindField("Time")
  rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
  result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
  dbTable3 = av.FindDoc("sunday.dbf")
  theVTab3 = dbTable3.GetVTab
  theSelection3 = theVTab3.GetSelection

  selCount3 = theSelection3.GetSelection

  if(selCount3 <> 0) then
   'call on the route number from the field "Route_no"

   theField3 = theVTab3.FindField("Route_no")
   rec = theSelection3.GetNextSet(-1)  'first record in the selection
   aField = theVTab3.ReturnValue(theField3, rec)

   if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
     or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

    'use transitway stats
```

```
    if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
      P1 = 0.51
      P2 = 0.365
      P3 = 0.125
    elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
      P1 = 0.27
      P2 = 0.68
      P3 = 0.05
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
      P1 = 0.155
      P2 = 0.74
      P3 = 0.105
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.13
      P2 = 0.715
      P3 = 0.155
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.19
      P2 = 0.75
      P3 = 0.06
    else
      P1 = 0.18
      P2 = 0.715
      P3 = 0.105  'all day values
    end

  else
    'use non transitway stats

    if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
      P1 = 0.32
      P2 = 0.605
      P3 = 0.075
    elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
      P1 = 0.21
      P2 = 0.665
      P3 = 0.125
    elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
      P1 = 0.30
      P2 = 0.62
      P3 = 0.08
    elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.35
      P2 = 0.565
      P3 = 0.085
    else
      P1 = 0.285
      P2 = 0.615
      P3 = 0.10  'all day values
    end
  end

' read the results from the schedule tables

theTable4 = av.FindDoc("sunday.dbf")
theVTab4 = theTable4.GetVTab
theBitmap4 = theVTab4.GetSelection

theVTab4.UpdateSelection
'theVTab4.PromoteSelection
aField4 = theVTab4.FindField("Route_no")
aField5 = theVTab4.FindField("Time")
rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values


else
'MsgBox.Info("No tables have a selection....", "")
```

```
    end
   end
  end

'the new expected time is calculated

ExTime3 = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

 'P(ontime|early)
  time1 = (P2*OE)/early

  'P(early|early)
  time2 = (P1*EE)/early

  'P(late|early)
  time3 = (P3*LE)/early

  ExTime4 = (time1*T2) + (time2*T1) + (time3*T3)
  'ExTime = ExTime1.Round


if (time < ExTime4)then
  TrvlTime = Time

elseif(time > ExTime4)then
  TrvlTime = ExTime4
else
  TrvlTime = time
end



elseif(condField.AsString = "R3")then  'late

  'P(ontime|late)
  time1 = (P2*OL)/late

  'P(early|late)
  time2 = (P1*EL)/late

  'P(late|late)
  time3 = (P3*LL)/late

  ExTime = (time1*T2) + (time2*T1) + (time3*T3)
  'ExTime = ExTime1.Round

  time = ExTime

  'rerun routing to determine if there is another route
    av.run("netrun3","")

    'get the view and the route theme
    theView = av.GetActiveDoc
    theTheme = theView.FindTheme("Route1")
    theTable = theTheme.GetFTab

    'set up table for selection
    thebitmap = theTable.GetSelection
    expr = "[F_cost] = 0"
    theTable.Query(expr, theBitMap, #VTAB_SELTYPE_NEW) 'do query
    theTable.UpdateSelection ' highlight found records

    theField = theTable.FindField("T_cost")

    rec = theBitMap.GetNextSet(-1) 'first record in bitmap
    theField2 = theTable.ReturnValue(theField, rec) 'fill values


    'msgbox.info(theField2.AsString, " ")
```

```
'aBitmap.ClearAll

'now perform the statistics on this value
'perform the statistics on theField2.AsString


TravelTime = theField2

T1 = TravelTime - 1
T2 = TravelTime
T3 = TravelTime + 3

'all in minutes

'route = route number from table

' perform an if statement to determine which schedule table was used ie. weekday, saturday, sunday
' since each table is cleared at the end, only one table should have records highlighted

dbTable1 = av.FindDoc("weekday.dbf")
theVTab1 = dbTable1.GetVTab
theSelection1 = theVTab1.GetSelection

selCount1 = theSelection1.Count

if (selCount1 <> 0) then

'call on the route number from the field "Route_no"

theField1 = theVTab1.FindField("Route_no")
rec = theSelection1.GetNextSet(-1)  'first record in the selection
aField = theVTab1.ReturnValue(theField1, rec)

if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

'use transitway stats

if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
   P1 = 0.255
   P2 = 0.72
   P3 = 0.025
elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
   if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
     P1 = 0.165
     P2 = 0.705
     P3 = 0.13
   else
     P1 = 0.20
     P2 = 0.705
     P3 = 0.095
   end
elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
  P1 = 0.26
  P2 = 0.64
  P3 = 0.10
elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
  if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
    P1 = 0.25
    P2 = 0.655
    P3 = 0.095
  else
    P1 = 0.25
    P2 = 0.635
    P3 = 0.115
  end
elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
  P1 = 0.21
  P2 = 0.645
```

```
    P3 = 0.145
  else
    P1 = 0.24
    P2 = 0.645
    P3 = 0.115  'all day value
  end

else

'use non-transitway stats

  if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.225
    P2 = 0.71
    P3 = 0.065
  elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    if((theField2.AsString >= "731") and (theField2.AsString <= "830"))then
      P1 = 0.195
      P2 = 0.555
      P3 = 0.25
    else
      P1 = 0.24
      P2 = 0.60
      P3 = 0.16
    end
  elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.34
    P2 = 0.585
    P3 = 0.075
  elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    if((theField2.AsString >= "1531") and (theField2.AsString <= "1630"))then
      P1 = 0.34
      P2 = 0.525
      P3 = 0.135
    else
      P1 = 0.405
      P2 = 0.49
      P3 = 0.105
    end
  elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
      P1 = 0.305
      P2 = 0.565
      P3 = 0.13
  else
      P1 = 0.325
      P2 = 0.56
      P3 = 0.115  'all day values
  end
end

' read the results from the schedule tables

theTable2 = av.FindDoc("weekday.dbf")
theVTab2 = theTable2.GetVTab
theBitmap2 = theVTab2.GetSelection

theVTab2.UpdateSelection
'theVTab2.PromoteSelection
aField4 = theVTab2.FindField("Route_no")
aField5 = theVTab2.FindField("Time")
rec4 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result4 = theVTab2.ReturnValue(aField4, rec4) 'fill values
rec5 = theBitMap2.GetNextSet(-1) 'first record in bitmap
result5 = theVTab2.ReturnValue(aField5, rec5) 'fill values

else

dbTable2 = av.FindDoc("saturday.dbf")
theVTab2 = dbTable2.GetVTab
theSelection2 = theVTab2.GetSelection
```

```
selCount2 = theSelection2.GetSelection

if (selCount2 <> 0) then
 'call on the route number from the field "Route_no"

 theField2 = theVTab2.FindField("Route_no")
 rec = theSelection2.GetNextSet(-1) 'first record in bitmap
 aField = theVTab2.ReturnValue(theField2, rec) 'fill values

 if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
  or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

  'use transitway stats

   if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
     P1 = 0.245
     P2 = 0.745
     P3 = 0.01
   elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
     P1 = 0.35
     P2 = 0.615
     P3 = 0.035
   elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
     P1 = 0.26
     P2 = 0.68
     P3 = 0.06
   elseif ((theField2.AsString >= "1531") and (theField2.AsString < = "1830"))then
     P1 = 0.23
     P2 = 0.645
     P3 = 0.125
   elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
     P1 = 0.22
     P2 = 0.685
     P3 = 0.095
   else
     P1 = 0.26
     P2 = 0.66
     P3 = 0.08  'all day values
   end

 else

  'use non transitway stats
   if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
     P1 = 0.37
     P2 = 0.50
     P3 = 0.13
   elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
     P1 = 0.215
     P2 = 0.655
     P3 = 0.13
   elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
     P1 = 0.23
     P2 = 0.64
     P3 = 0.13
   elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
     P1 = 0.24
     P2 = 0.51
     P3 = 0.25
   elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
     P1 = 0.32
     P2 = 0.60
     P3 = 0.08
   else
     P1 = 0.265
     P2 = 0.595
     P3 = 0.14  'all day values
   end
end
```

```
 ' read the results from the schedule tables

 theTable3 = av.FindDoc("saturday.dbf")
 theVTab3 = theTable3.GetVTab
 theBitmap3 = theVTab3.GetSelection

 theVTab3.UpdateSelection
 'theVTab3.PromoteSelection
 aField4 = theVTab3.FindField("Route_no")
 aField5 = theVTab3.FindField("Time")
 rec4 = theBitMap3.GetNextSet(-1) 'first record in bitmap
 result4 = theVTab3.ReturnValue(aField4, rec4) 'fill values
 rec5 = theBitMap3.GetNextSet(-1) 'first record in bitmap
 result5 = theVTab3.ReturnValue(aField5, rec5) 'fill values


else
 dbTable3 = av.FindDoc("sunday.dbf")
 theVTab3 = dbTable3.GetVTab
 theSelection3 = theVTab3.GetSelection

 selCount3 = theSelection3.GetSelection

 if(selCount3 <> 0) then
  'call on the route number from the field "Route_no"

  theField3 = theVTab3.FindField("Route_no")
  rec = theSelection3.GetNextSet(-1)  'first record in the selection
  aField = theVTab3.ReturnValue(theField3, rec)

  if ((aField.AsString = "97") or (aField.AsString = "95") or (aField.AsString = "86")
    or (aField.AsString = "77") or (aField.AsString = "66") or (aField.AsString = "35")) then

   'use transitway stats

   if ((theField2.AsString >= "301") and (theField2.AsString <= "630"))then
    P1 = 0.51
    P2 = 0.365
    P3 = 0.125
   elseif ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.27
    P2 = 0.68
    P3 = 0.05
   elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
    P1 = 0.155
    P2 = 0.74
    P3 = 0.105
   elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
    P1 = 0.13
    P2 = 0.715
    P3 = 0.155
   elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
    P1 = 0.19
    P2 = 0.75
    P3 = 0.06
   else
    P1 = 0.18
    P2 = 0.715
    P3 = 0.105  'all day values
   end

  else
   'use non transitway stats

   if ((theField2.AsString >= "631") and (theField2.AsString <= "930"))then
    P1 = 0.32
    P2 = 0.605
    P3 = 0.075
   elseif ((theField2.AsString >= "931") and (theField2.AsString <= "1530"))then
```

```
        P1 = 0.21
        P2 = 0.665
        P3 = 0.125
      elseif ((theField2.AsString >= "1531") and (theField2.AsString <= "1830"))then
        P1 = 0.30
        P2 = 0.62
        P3 = 0.08
      elseif ((theField2.AsString >= "1831") and (theField2.AsString <= "300"))then
        P1 = 0.35
        P2 = 0.565
        P3 = 0.085
      else
        P1 = 0.285
        P2 = 0.615
        P3 = 0.10  'all day values
      end
    end

  ' read the results from the schedule tables

  theTable4 = av.FindDoc("sunday.dbf")
  theVTab4 = theTable4.GetVTab
  theBitmap4 = theVTab4.GetSelection

  theVTab4.UpdateSelection
  'theVTab4.PromoteSelection
  aField4 = theVTab4.FindField("Route_no")
  aField5 = theVTab4.FindField("Time")
  rec4 = theBitMap4.GetNextSet(-1) 'first record in bitmap
  result4 = theVTab4.ReturnValue(aField4, rec4) 'fill values
  rec5 = theBitMap4.GetNextSet(-1)  'first record in bitmap
  result5 = theVTab4.ReturnValue(aField5, rec5) 'fill values


  else
  'MsgBox.Info("No tables have a selection....", "")
  end
 end
end


'the new expected time is calculated

ExTime3 = (P1*T1) + (P2*T2) + (P3*T3) ' prior probability

 'P(ontime|early)
 time1 = (P2*OE)/early

 'P(early|early)
 time2 = (P1*EE)/early

 'P(late|early)
 time3 = (P3*LE)/early

 ExTime4 = (time1*T2) + (time2*T1) + (time3*T3)
 'ExTime = ExTime1.Round


if (time < ExTime4)then
 TrvlTime = Time

elseif(time > ExTime4)then
 TrvlTime = ExTime4
else
 TrvlTime = time
end


else
 'MsgBox.Info("Did nothing", "")
```

```
end
end

'read the results from the table created by Nearest Feature
'called near.dbf

theTable1 = av.FindDoc("near.dbf")
theVTab1 = theTable1.GetVTab
theBitmap1 = theVTab1.GetSelection
theBitmap1.SetAll
theVTab1.UpdateSelection

aField1 = theVTab1.FindField("Bus_stop")
aField2 = theVTab1.FindField("On_street")
aField3 = theVTab1.FindField("At_street")

rec1 = theBitMap1.GetNextSet(-1) 'first record in bitmap
result1 = theVTab1.ReturnValue(aField1, rec1) 'fill values
rec2 = theBitMap1.GetNextSet(-1) 'first record in bitmap
result2 = theVTab1.ReturnValue(aField2, rec2) 'fill values
rec3 = theBitMap1.GetNextSet(-1) 'first record in bitmap
result3 = theVTab1.ReturnValue(aField3, rec3) 'fill values

'result5 needs the T_cost added to it---time distance between the bus stop
'with a time and the bus stop without a time---if the distance is zero then
' result5 = "Time"

theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Route2")
theTable = theTheme.GetFTab
theBitmap = theTable.GetSelection

QueryString = "[F_cost] = 0"
theTable.Query(QueryString, theBitmap, #VTAB_SELTYPE_NEW)
theTable.UpdateSelection
theField = theTable.FindField("T_cost")
aRec = theBitmap.GetNextSet(-1)   'first record in bitmap
Tvalue = theTable.ReturnValue(theField, aRec)  'fill values

ArrivalTime = result5.AsNumber + Tvalue.Round

'get view and zoom to route
theView = av.GetActiveDoc
theTheme = theView.FindTheme("Locate.shp")
theTheme.SetActive(TRUE)

'View.ZoomToTheme

av.GetProject.SetModified(true)
theView = av.GetActiveDoc
theThemes = theView.GetActiveThemes
r = Rect.MakeEmpty
for each t in theThemes
  r = r.UnionWith(t.ReturnExtent)
end

if (r.IsEmpty) then
  return nil
elseif ( r.ReturnSize = (0@0) ) then
  theView.GetDisplay.PanTo(r.ReturnOrigin)
else
  theView.GetDisplay.SetExtent(r.Scale(1.1))
end

theView = av.GetProject.FindDoc("Route Finding")
theTheme = theView.FindTheme("Bus Stops")
theTheme.SetVisible(FALSE)

theTheme2 = theView.FindTheme("Locate.shp")
theFTab = theTheme2.GetFTab
```

```
theBitmap = theFTab.GetSelection
theBitmap.ClearAll

theWin = theView.GetWin
theWin.Resize(1000,800)
theWin.Moveto(0,0)
theWin.Open

'report results

ArrTime = ArrivalTime.AsString
rightTime = ":" + ArrTime.Right(2)
' need to find out how many characters are in ArrivalTime
lftTime = ArrivalTime.AsString
lftTime.Count
if(lftTime.Count < 4)then
  leftTime = lftTime.Left(1)
else
  leftTime = lftTime.Left(2)
end

NewTime = leftTime + rightTime

'set decimal places for travel time
TvlTime = TrvlTime.SetFormat("dd.dd")

mymessage = "The route information is as follows:" + NL + NL +
"The route:  " + result4.AsString + NL + "The bus stop:  " +
result1.AsString + NL + "Location:  " + "On Street:  " +
result2.AsString + "  " + "At:  " + result3.AsString + NL +
NL + "Bus arrival at approximately:  " + NewTime.AsString + NL + NL +
"Approximate Travel Time:  " + TvlTime.AsString + "min" + NL + NL +
"If you need more information, please contact OCTranspo."

MsgBox.Report(mymessage, "Results")

'print the results
aSEd = SEd.MakeFromSource(mymessage, "tempscript")
aSEd.Print
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Script Name:       NetRun3
' Script Description: Finds a route between the two locations.
'
' Programmer:        Sarah Riley
' Date Created:      Nov. 2000
' Returned Object:   none
'
' Changes:
'
' Author      Date        Details
'
' S. Riley    15-Jun-2002    Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


'get the view and the location theme
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theTheme=theView.FindTheme("Locate.shp")
theFtab=theTheme.getFtab

theStreetFTab=theView.FindTheme("Transit routes.shp").getFtab
theNetDef=NetDef.Make(theStreetFTab)
theNetwork=Network.Make(theNetDef)

thePointList={}
thePointField=theFtab.FindField("Shape")
for each rec in theFtab
  p=theFtab.ReturnValue(thePointField, rec)
  thePointList.add(p)
end

'**************
' Set the cost for solving a routing problem
' It then solves the problem and displays
' directions for the route.

' Get a list of available costs for this line theme. Choose a cost from the list.
'

aNetCostFieldList = theNetDef.GetCostFields
aNetCostField = MsgBox.Choice(aNetCostFieldList, "Select Minutes", "Select Minutes")

' If a cost was chosen, set it.

if (aNetCostField<>nil) then
  theNetwork.SetCostField(aNetCostField)
end

'The above can be replaced by changing the hard code for Network Analyst wherein
'there will only be the chosen cost of MINUTES to select.
'*****************

TravelDistance= theNetwork.FindPath(thePointList, True, False)
'msgBox.Info( TravelDistance.AsString, "Total"++reportUnitStr )

'theView.GetGraphics.add(GraphicShape.Make(theNetwork.ReturnPathShape))

aSymbol = Symbol.Make(#SYMBOL_PEN)
aSymbol.SetSize(2.0)
aSymbol.SetColor(Color.GetMagenta)

theGL=theView.GetGraphics
theRoute=GraphicShape.Make(theNetwork.ReturnPathShape)
theRoute.SetDisplay(av.GetActiveDoc.GetDisplay)
theroute.SetSymbol(aSymbol)
theGL.Add(theRoute)
'theroute.GetSymbol.SetColor(color.GetMagenta).SetSymbol(aSymbol)
theRoute.Draw
```

```
theroute.Invalidate

'set view size for public viewing

theView = av.getproject.finddoc("Route Finding")
theWin = theView.GetWin
'theWin.Resize(1025,600)
'theWin.Moveto(0,0)
theWin.Open

' Write the route result FTab, this is necessary to get directons.
'
tmpFileName = FileName.Make("F:\thesis data\Route.shp")
theNetwork.WritePath(tmpFileName)
resFTab = FTab.Make(SrcName.Make(tmpFileName.AsString))

' Show the directions then remove the temporary file.
'
'theDirections="Your route directions are as follows"+NL+NL+
'            theNetwork.GetPathDirections(resFTab)
'msgBox.Report(theDirections, "Route Information")

'add theme to view
newTheme = FTheme.Make(resFTab)
theView.AddTheme(newTheme)
newTheme.SetName("Route1")
newTheme.SetVisible(True)
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'View.Hotlink
' Script Name:       Theme.HotLink
' Script Description: Opens OC Transpo webpage linked to shapfile field "HotLink" of URL record
' Usage:             As Tool under View Menu.  Theme Properties linked to "HotLink" field of shapefile
' Author:            Sarah Riley
' Created:           15-Mar-2002
' Changes:
'
' Author      Date          Modifications
' S. Riley    15-Mar-2002   Initial Implementation
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' default project path should be "c:\school"

' default window header

Title = "HotLink to the Internet"

' Test Micro$oft Operating system type, setup DLL's for appropriate system.

if (System.GetOSVariant = #SYSTEM_OSVARIANT_MSWNT) then
  dllShell32 = DLL.Make("C:\winnt\system32\shell32.dll".AsFileName)
  dllUser32  = DLL.Make("C:\winnt\system32\user32.dll".AsFileName)
elseif (System.GetOSVariant = #SYSTEM_OSVARIANT_MSW95) then
  dllShell32 = DLL.Make("C:\windows\system\shell32.dll".AsFileName)
  dllUser32  = DLL.Make("C:\windows\system\user32.dll".AsFileName)
else
  MsgBox.Info("Windows is not being used.", Title)
  exit
end
if ((dllShell32 = nil) or (dllUser32 = nil)) then
  MsgBox.Error("Can't find required DLL's.  Check your setup.", Title)
  exit
end

' setup Win32API to talk with Avenue

dpGetActiveWindow = DLLProc.Make(dllUser32, "GetActiveWindow",#DLLPROC_TYPE_INT32, {})

dpShellExecute = DLLProc.Make
            (
             dllShell32, "ShellExecuteA",
             #DLLPROC_TYPE_INT32,
             {
              #DLLPROC_TYPE_INT32,
              #DLLPROC_TYPE_STR,
              #DLLPROC_TYPE_STR,
              #DLLPROC_TYPE_STR,
              #DLLPROC_TYPE_STR,
              #DLLPROC_TYPE_INT32
             }
            )

' Get window handle of ArcView window

activeWin = DLL.GetAVWindowHandle

' Get the URL off clicked area of theme

hotLink = SELF

' Send info to default browser

hotLinkToBrowser = dpShellExecute.Call({ActiveWin, "Open", hotLink, Title, FileName.GetCWD.AsString, 1})

if (hotLinkToBrowser <= 32) then
  MsgBox.Warning("Hotlink failed", Title)
end
```

```
'View.Locate

'ESRI SCRIPT
'-----------------------------------------------------------
'Get the project, get all themes and inactivate them except for ottawa streets
'
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theThemeList=theView.getThemes
for each i in (0..(theThemeList.count-1))
  temp=theThemeList.get(i).setActive(FALSE)
end
theTheme=theView.FindTheme("Ottawa Street Network")
theTheme.setActive(TRUE)

'-----------------------------------------------------------

theView = av.GetActiveDoc
for each t in theView.GetActiveThemes
 if ((t.Is( FTHEME )) and (t.GetMatchSource <> NIL)) then
    p = LocateDialog.Show(t)
    if (p <> NIL) then
      if (p.IsNull) then
        MsgBox.Info("Cannot locate address.", "Locate")
      else
        proj = theView.GetProjection
        pp = p.ReturnProjected(proj)
        s = BasicMarker.Make
        s.SetStyle(#BASICMARKER_STYLE_PATTERN)
        s.SetFont(Font.Make("ESRI GEOMETRIC SYMBOLS", "normal"))
        s.SetCharacter(34)
        s.SetColor(Color.GetGreen)
        g = theView.GetGraphics
        thePoint = GraphicShape.Make(pp)
        thePoint.SetDisplay(av.GetActiveDoc.GetDisplay)
        thePoint.SetSymbol(s)
        g.Add(thePoint)
        'ext = theView.GetDisplay.ReturnExtent
        if (not(pp.IsContainedIn(theView.GetDisplay.ReturnExtent))) then
          theView.GetDisplay.PanTo(pp)
        end
        return nil
      end
    end
 end
end
```

```
'View.Locate2

'ESRI SCRIPT
'-----------------------------------------------------------
'Get the project, get all themes and inactivate them except for ottawa streets
'
theProject = av.GetProject
theView = theProject.FindDoc("Route Finding")
theThemeList=theView.getThemes
for each i in (0..(theThemeList.count-1))
  temp=theThemeList.get(i).setActive(FALSE)
end
theTheme=theView.FindTheme("Ottawa Street Network")
theTheme.setActive(TRUE)

'-----------------------------------------------------------

theView = av.GetActiveDoc
for each t in theView.GetActiveThemes
 if ((t.Is( FTHEME )) and (t.GetMatchSource <> NIL)) then
    p = LocateDialog.Show(t)
    if (p <> NIL) then
      if (p.IsNull) then
        MsgBox.Info("Cannot locate address.", "Locate")
      else
        proj = theView.GetProjection
        pp = p.ReturnProjected(proj)
        s = BasicMarker.Make
        s.SetStyle(#BASICMARKER_STYLE_PATTERN)
        s.SetFont(Font.Make("ESRI GEOMETRIC SYMBOLS", "normal"))
        s.SetCharacter(34)
        s.SetColor(Color.GetRed)
        g = theView.GetGraphics
        thePoint = GraphicShape.Make(pp)
        thePoint.SetDisplay(av.GetActiveDoc.GetDisplay)
        thePoint.SetSymbol(s)
        g.Add(thePoint)
        'ext = theView.GetDisplay.ReturnExtent
        if (not(pp.IsContainedIn(theView.GetDisplay.ReturnExtent))) then
          theView.GetDisplay.PanTo(pp)
        end
        return nil
      end
    end
 end
end
```

Appendix D
Attribute Tables

List of Tables

Table D.1: Sample of Ottawa Street Network

FNODE_	TNODE_	LPOLY_	RPOLY_	LENGTH	GSNF505R_	GSNF505R_I	NAME	TYPE	DIR	ADDR_FM_LE	ADDR_TO_LE	ADDR_FM_RG	ADDR_TO_RG	ADD_IMPUT	ARC_ID	SOURCE
4820	4799	1564	1564	109.049	6141	812034	WAREBERG			3085	3085	-1	-1	0	812034	1
4839	4820	1564	1605	88.769	6166	900746	SENTIER BOISE	LN		3067	3073	3060	3070	0	900746	1
4829	4839	1564	1613	32.479	6167	900747	MORIN	RD		1077	1077	-1	-1	0	900747	1
4820	4865	1564	1605	150.633	6209	625225	ARMSTRONG	RD		1085	1111	-1	-1	4	625225	1
4882	4871	1617	1617	66.388	6228	625228	LEO	LN		2985	2995	2990	2996	0	625228	1
4884	4882	1617	1617	45.849	6230	625229	LEO	LN		2975	2981	2984	2988	3	625229	1
4887	4865	1605	1564	91.236	6232	812036	PHILIP	RD		3071	3071	3070	3070	0	812036	1
4839	4887	1605	1613	145.823	6233	900748	MORIN	ST		1091	1105	-1	-2	0	900748	1
4865	4917	1564	1564	145.061	6276	625232	ARMSTRONG	RD		1113	1139	-1	-1	8	625232	1
4918	4925	1617	1617	25.370	6283	900749	CAMERON	ST		895	899	904	906	15	900749	1
4882	4944	1617	1617	164.895	6308	625237	EAST SHORE	LN		-1	-1	-1	-1	0	625237	1
4945	4939	1617	1617	93.847	6309	812038	MANSE	RD		2517	2535	-1	-1	0	812038	1
4925	4945	1617	1617	50.344	6310	900750	CAMERON	ST		901	911	908	916	15	900750	1
4945	4948	1617	1617	6.397	6317	900751	CAMERON	ST		913	913	918	918	15	900751	1
4944	4950	1617	1644	191.672	6322	625238	HIGHWAY 17	HY		-1	-1	-2	-1	0	625238	1
4887	4956	1564	1613	199.509	6329	812039	MORIN	ST		1117	1137	1126	1126	0	812039	1
4950	4956	1613	1644	83.930	6330	625240	HIGHWAY 17	HY		-1	-1	-1	-2	0	625240	1
4966	4955	1617	1617	30.975	6343	625243	WEST	ST		-1	-1	-2	-1	0	625243	1
4956	4973	1564	1644	499.733	6351	625247	HIGHWAY 17	HY		-1	-2	-2	-1	0	625247	1
4966	4975	1655	1617	48.677	6352	625244	THIRD	ST		-1	-1	-1	-2	0	625244	1
4976	4944	1617	1644	1350.030	6353	625246	HIGHWAY 17	HY		-1	2639	-1	-2	0	625246	1
4976	4940	1617	1617	122.379	6354	812040	EAST	ST		981	2621	-1	-1	0	812040	1
4978	4925	1617	1617	286.457	6356	827441	FERRY ROAD	LN		2457	2481	2466	2478	0	827441	1
4948	4980	1617	1617	176.537	6361	625248	MANSE	RD		-1	-1	-1	-1	0	625248	1

4975	4982	1655	1617	51.206	6364	625249	THIRD	ST		-1	-1	-2	-1	0	625249	1	
4948	4982	1617	1617	87.897	6365	812041	CAMERON	ST		915	921	920	954	1	812041	1	
4986	4976	1617	1660	146.333	6368	625252	HIGHWAY 17	HY		-2	-1	-1	-1	0	625252	1	
4975	4989	1655	1655	42.761	6374	625253	DUKE	ST		-1	-1	-1	-1	0	625253	1	
4973	4999	1640	1644	62.327	6389	216028				-1	-1	-1	-1	0	216028	1	
4999	4983	1662	1644	135.912	6390	216029	OLD MONTREAL	RD		-2	-1	-1	-1	0	216029	1	
5000	4983	1644	1658	430.178	6391	812042	QUEEN	ST		2961	3055	2992	-1	14	812042	1	
5002	5000	1644	1669	299.312	6392	812043	QUEEN	ST		2895	2959	2920	2990	15	812043	1	
5006	4986	1617	1663	324.307	6398	625255	HIGHWAY 17	HY		2559	-2	-1	-1	0	625255	1	
5006	4966	1655	1617	95.510	6399	625256	WEST	ST		982	-1	-1	-2	8	625256	1	
5010	4976	1660	1644	78.520	6407	812044	EAST	ST		3	979	-2	-1	12	812044	1	
5011	5006	1655	1674	103.479	6408	625257	HIGHWAY 17	HY		-1	-1	-1	-1	0	625257	1	
4982	5011	1655	1617	71.116	6409	900755	CAMERON	ST		945	965	956	982	2	900755	1	
5014	5010	1660	1677	155.463	6412	625259	SPARKS	ST		2597	2615	2534	-1	2	625259	1	
4986	5014	1660	1663	74.310	6413	625260	FAUBERT	ST		-1	-1	996	996	0	625260	1	
5017	5014	1663	1680	76.819	6418	625261	SPARKS	ST		2577	2589	-1	-1	8	625261	1	
5021	5017	1663	1683	79.341	6424	625262	SPARKS	ST		2565	2575	-1	-1	12	625262	1	
5024	5002	1644	1671	1004.588	6428	812045	QUEEN	ST		2671	2893	2678	2918	15	812045	1	
5032	5024	1644	1671	208.512	6440	625267	QUEEN	ST		2625	2669	2628	2676	15	625267	1	
5034	5021	1663	1684	148.337	6443	625265	SPARKS	ST		2539	2563	2530	2530	6	625265	1	
5034	5006	1674	1663	88.324	6444	625266	WEST	ST		980	980	-1	-1	0	625266	1	
5035	5032	1644	1692	23.494	6445	625268	QUEEN	ST		2619	2623	2622	2626	15	625268	1	
5035	5010	1677	1644	74.556	6446	216219	EAST	ST		-1	1	-1	-2	4	216219	1	
5043	5035	1677	1692	165.728	6456	625269	QUEEN	ST		2591	2617	2582	2620	3	625269	1	
5014	5043	1677	1680	74.310	6457	625270	FAUBERT	ST		-1	-1	998	998	0	625270	1	
5046	5043	1680	1692	36.778	6461	625271	QUEEN	ST		2581	2589	2574	2580	15	625271	1	
5049	5046	1680	1702	45.100	6464	625272	QUEEN	ST		2569	2579	2564	2572	15	625272	1	
5049	5017	1683	1680	76.428	6465	827443	MONK	ST		-1	-1	997	1001	0	827443	1	
5052	5049	1683	1702	23.387	6468	625274	QUEEN	ST		2561	2567	2558	2562	15	625274	1	
5000	5053	1658	1669	126.828	6469	812046	PIERRETTE			-1	-2	-1	-1	0	812046	1	
5056	5034	1674	1684	103.045	6472	625275	SPARKS	ST		2511	2531	2530	2530	1	625275	1	

Table D.2: Sample of Transit Routes

FNODE_	TNODE_	LPOLY_	RPOLY_	NAME	TYPE	ADD_F_L	ADD_T_L	ADD_F_R	ADD_T_R	ARC_ID	CFCC	SPEED_LIMI	ONE WAY	LENGTH	CONDITION	F_ELEV	T_ELEV	MINS
7349	7309	2705	2476	RAMP		0	0	0	0	718879	A63	25	FT	41.514	R1	0	0	0.629
7388	7386	2705	2738	RAMP		0	0	0	0	718884	A63	25	FT	38.361	R1	0	0	0.624
7309	7388	2705	2636	HIGHWAY 17	HY	-1	-1	-2	-2	718885	A25	55		115.173	R1	0	0	0.645
7404	7349	2723	2484	RAMP		0	0	0	0	718886	A63	25	FT	46.273	R1	0	0	0.636
7386	7404	2723	2740	RAMP		0	0	0	0	718887	A63	25	FT	47.187	R1	0	0	0.637
7388	7522	2738	2636	HIGHWAY 17	HY	-1	-1	-2	-1	718891	A25	55		237.769	R1	0	0	0.728
7562	7539	2714	2805	PRINCESS LOUISE	DR	685	689	682	688	218026	A41	25	TF	89.069	R1	0	0	0.450
7539	7563	2606	2805	PRINCESS LOUISE	DR	693	697	690	696	786315	A41	25	TF	76.949	R1	0	0	0.115
7522	7588	2795	2681	HIGHWAY 17	HY	-1	-2	-1	-1	718897	A25	55		115.303	R1	0	0	0.078
7563	7610	2712	2810	PRINCESS LOUISE	DR	781	781	700	-2	900154	A41	25	TF	81.243	R1	0	0	0.438
7612	7562	2606	2805	PRINCESS LOUISE	DR	679	679	672	680	786318	A41	25	TF	105.045	R1	0	0	0.474
7404	7647	2484	2740	RAMP		0	0	0	0	718901	A63	25	FT	229.026	R1	0	0	0.909
7648	7612	2606	2810	PRINCESS LOUISE	DR	655	677	634	670	218093	A41	25	TF	68.446	R1	0	0	0.102
7588	7665	2795	2783	HIGHWAY 17	HY	-2	-2	-1	-1	718902	A25	55		144.935	R1	0	0	0.098
7693	7648	2710	2841	PRINCESS LOUISE	DR	635	653	628	632	218127	A41	25	TF	57.601	R1	0	0	0.403
7610	7717	2781	2810	PRINCESS LOUISE	DR	709	713	-2	-2	786324	A41	25	TF	114.255	R1	0	0	0.487
7751	7693	2797	2841	PRINCESS LOUISE	DR	621	633	620	626	218174	A41	25	TF	101.515	R1	0	0	0.468
7785	7751	2797	2894	PRINCESS LOUISE	DR	605	619	598	610	218204	A41	25	TF	102.346	R1	0	0	0.470
7788	7647	2740	2840	ST JOSEPH	BV	3327	3395	3376	3448	718912	A01	30	TF	278.077	R1	0	0	0.913
7717	7812	2850	2810	PRINCESS LOUISE	DR	761	761	-2	694	718914	A41	25	TF	83.844	R1	0	0	0.125
7858	7785	2710	2909	PRINCESS LOUISE	DR	569	589	578	588	218263	A41	25	TF	149.979	R1	0	0	0.791
7812	7901	2850	2918	PRINCESS LOUISE	DR	761	761	700	-2	718923	A41	25	TF	94.726	R1	0	0	0.141
7665	7904	2795	2763	HIGHWAY 17	HY	-2	-1	-1	-2	718921	A25	55		465.589	R1	0	0	0.883
7907	7858	2710	2909	PRINCESS LOUISE	DR	529	567	572	576	218315	A41	25	TF	90.061	R1	0	0	0.701
7935	7907	2949	2909	PRINCESS LOUISE	DR	523	527	568	570	218328	A41	25	TF	39.582	R1	0	0	0.626
7901	7958	2809	2918	PRINCESS LOUISE	DR	763	-1	-2	724	718927	A41	25	TF	73.723	R1	0	0	0.110

7904	7978	2964	2763	HIGHWAY 17	HY	-1	-2	-2	-1	718929	A25	55		194.221	R1	0	0	0.699
8007	7935	2949	2975	PRINCESS LOUISE	DR	507	521	550	566	218403	A41	25	TF	102.087	R1	0	0	0.719
7958	8016	2985	2810	PRINCESS LOUISE	DR	-1	-1	726	-2	218254	A41	25	TF	92.817	R1	0	0	0.138
8031	7788	2795	2840	ST JOSEPH	BV	3199	3325	3240	3374	718935	A01	30	TF	513.457	R1	0	0	1.205
8022	8032	2896	2896	VARENNES	BV	-1	-1	-1	-1	718938	A41	25	TF	70.527	R1	0	0	0.422
8016	8045	3020	2810	PRINCESS LOUISE	DR	-1	-1	-2	-1	218277	A41	25	TF	42.644	R1	0	0	0.064
8072	8007	3010	2977	PRINCESS LOUISE	DR	499	499	538	548	718943	A41	25	TF	66.148	R1	0	0	0.666
7647	8084	2710	2840	TENTH LINE	RD	-1	-1	-1	-2	218507	A01	35	TF	450.045	R1	0	0	1.046
8045	8107	3020	2895	PRINCESS LOUISE	DR	-1	-2	-1	-1	218317	A41	25	TF	57.338	R1	0	0	0.086
8107	8123	3020	3047	PRINCESS LOUISE	DR	-2	-1	-1	-2	718951	A41	25	TF	25.726	R1	0	0	0.038
8138	8031	3017	2840	ST JOSEPH	BV	3155	3197	3194	3238	765530	A01	30	TF	174.756	R1	0	0	0.784
8074	8138	3017	3043	PRESTONE	DR	1151	1169	1162	1174	218749	A41	25		92.151	R1	0	0	0.704
8141	8072	3042	2977	PRINCESS LOUISE	DR	489	497	524	536	218524	A41	25	TF	73.571	R1	0	0	0.677
8144	8136	2840	3064	PRESTONE	DR	1267	1269	1274	1304	218637	A41	25		118.883	R1	0	0	0.177
8150	8022	2809	2896	WATTERS	RD	-1	-2	814	842	718957	A41	25	TF	133.940	R1	0	0	0.767
8123	8164	3062	3047	PRINCESS LOUISE	DR	-1	-2	-2	-1	718961	A41	25	TF	51.518	R1	0	0	0.644
8174	8144	3015	3064	PRESTONE	DR	1249	1265	1256	1272	218716	A41	25		79.136	R1	0	0	0.118
7978	8178	2964	2968	HIGHWAY 17	HY	-2	-1	-1	-1	718962	A25	55		455.518	R1	0	0	0.876
8179	8141	3051	3075	PRINCESS LOUISE	DR	475	487	494	522	677350	A41	25	TF	52.608	R1	0	0	0.645
8198	8074	3009	3043	CENTRUM	RD	-1	-2	-2	-1	218889	A41	25		227.592	R1	0	0	0.906
8164	8200	3062	2895	PRINCESS LOUISE	DR	-2	-1	-1	-2	718970	A41	25	TF	45.857	R1	0	0	0.635
8032	8204	2994	3023	VARENNES	BV	-1	-1	-1	-2	718975	A41	25	TF	224.904	R1	0	0	0.652
8178	8210	2964	3092	CHAMPLAIN	ST	1103	1113	1020	1022	219044	A41	25		40.490	R1	0	0	0.627
8138	8174	2840	3072	PRESTONE	DR	1171	1247	-4	-4	718978	A41	25		353.427	R1	0	0	1.094
8224	8179	3051	3086	PRINCESS LOUISE	DR	455	473	446	462	218593	A41	25	TF	81.785	R1	0	0	0.689
8204	8231	2896	3023	VARENNES	BV	-1	-2	-2	-1	718982	A41	25	TF	39.392	R1	0	0	0.059
8238	8119	2964	3060	PLACE D'ORLEANS	DR	-1	-1	-1	-2	718980	A41	25		287.064	R1	0	0	0.995
8210	8238	2964	3106	CHAMPLAIN	ST	1115	1127	1024	1024	718981	A41	25		40.795	R1	0	0	0.628
8255	8150	3058	2896	WATTERS	RD	-1	-1	778	812	718987	A41	25	TF	151.834	R1	0	0	0.793
8136	8265	2967	3070	PRESTONE	DR	1301	1317	1306	1316	218745	A41	25		202.256	R1	0	0	0.869

Table D.3: Sample of Bus Stops

BUS_STOP	ON_STREET	AT_STREET	LATITUDE	LONGITUDE	O_C-DOOR	DWELL_TIME
WD564	417 RAMP	ACRES	45.3448	-75.8163	4	30
AF965	ABBEY	1A	45.4068	-75.6648	4	60
AF970	ABBEY	2A	45.4068	-75.6645	4	60
WJ620	ABBEYHILL	CASTLEFRANK FS	45.2944	-75.8818	4	15
WJ650	ABBEYHILL	BANNING NS	45.2975	-75.8794	4	15
WJ660	ABBEYHILL	CHURCH M	45.2969	-75.8796	4	15
WJ670	ABBEYHILL	HAZEL MALL ENT	45.3000	-75.8766	4	15
WD558	ACRES	RICHMOND	45.3429	-75.8093	4	15
WD562	ACRES	QUEENSWAY	45.3444	-75.8139	4	15
RE900	AIRPORT	1A	45.3201	-75.6645	4	60
RE410	AIRPORT PARKWAY	BRIDGE NS	45.3287	-75.6541	4	15
RE420	AIRPORT PARKWAY	OVERPASS FS	45.3247	-75.6602	4	15
RE430	AIRPORT PARKWAY	DEPARTURE	45.3204	-75.6654	4	15
CA920	ALBERT	KENT	45.4187	-75.7034	4	60
CA930	ALBERT	BAY	45.4169	-75.7076	4	60
CB900	ALBERT	METCALFE	45.4217	-75.6961	4	60
CB910	ALBERT	BANK	45.4201	-75.7001	4	60
CJ075	ALBERT	TRANSITWAY FS	45.4136	-75.7117	4	60
CJ080	ALBERT	BOOTH FS	45.4125	-75.7142	4	60
CJ085	ALBERT	BOOTH NS	45.4127	-75.7135	4	60
CJ087	ALBERT	PERKINS	45.4133	-75.7121	4	60
CJ090	ALBERT	ROCHESTER FS	45.4124	-75.7141	4	60
AA035	ALEXANDER	HOMAS	45.4432	-75.6930	4	15
AA050	ALEXANDER	CHARLES NS	45.4428	-75.6921	4	15
AB300	ALFRED	MONTREAL FS	45.4391	-75.6549	4	15
AF090	ALTA VISTA	POST OFFICE M	45.4128	-75.6562	4	30
AF105	ALTA VISTA	ROLLAND FS	45.4116	-75.6560	4	15
AF110	ALTA VISTA	CALEDON FS	45.4102	-75.6573	4	15
AF120	ALTA VISTA	CALEDON NS	45.4101	-75.6571	4	15
SD250	AMBERWOOD	# 82 FS	45.3206	-75.7073	4	15
SD260	AMBERWOOD	86 NS	45.3204	-75.7073	4	15
SD270	AMBERWOOD	LACEWOOD NS	45.3205	-75.7104	4	15
SD280	AMBERWOOD	CALAIS NS	45.3210	-75.7106	4	15
SD290	AMBERWOOD	#43 NS	45.3214	-75.7132	4	15
SD300	AMBERWOOD	40 FS	45.3211	-75.7130	4	15
SD310	AMBERWOOD	WEATHERWD #1 F	45.3203	-75.7143	4	15
SD320	AMBERWOOD	ERINLEA FS	45.3194	-75.7136	4	15
SD330	AMBERWOOD	WEATHERWD #2 F	45.3187	-75.7134	4	15
SD340	AMBERWOOD	MERIVALE FS E	45.3173	-75.7148	4	15
EM580	AMIENS	PRESTWICK NS	45.4739	-75.4990	4	15
SD640	ANTARES	AURIGA NS	45.3298	-75.7083	4	15
SD660	ANTARES	56 M	45.3314	-75.7096	4	15

SD670	ANTARES	39 NS	45.3315	-75.7094	4	15
SD680	ANTARES	CAPELLA NS	45.3340	-75.7081	4	15
SD690	ANTARES	CAPELLA FS	45.3339	-75.7078	4	15
SD700	ANTARES	HUNT CLUB FS	45.3350	-75.7057	4	15
SD710	ANTARES	HUNT CLUB NS	45.3351	-75.7048	4	15
SD620	AURIGA	EPSILON NS	45.3310	-75.7035	4	15
SD635	AURIGA	46 MID	45.3304	-75.7052	4	15
SD645	AURIGA	43 MID	45.3301	-75.7058	4	15
SD650	AURIGA	ANTARES NS	45.3299	-75.7075	4	15
EA700	AVIATION MUSEUM	PLANTERS FS	45.4584	-75.6463	4	15
CA080	BANK	QUEEN NS	45.4207	-75.7011	4	30
CA090	BANK	SLATER NS	45.4193	-75.6999	4	30
CA100	BANK	SLATER NS	45.4191	-75.6997	4	30
CA110	BANK	SPARKS FS	45.4211	-75.7013	4	30
CF490	BANK	RIVERDALE FS	45.3914	-75.6802	4	30
CF500	BANK	CHESLEY FS	45.3912	-75.6802	4	30
CF510	BANK	OSSINGTON M	45.3923	-75.6813	4	30
CF520	BANK	GROVE FS	45.3925	-75.6819	4	30
CF530	BANK	BELMONT NS	45.3929	-75.6821	4	30
CF540	BANK	HOPEWELL NS	45.3941	-75.6836	4	30
CF550	BANK	SUNNYSIDE FS	45.3948	-75.6838	4	30
CF560	BANK	EUCLID NS	45.3953	-75.6844	4	30
CF570	BANK	AYLMER OPP	45.3957	-75.6844	4	30
CF580	BANK	WILTON NS	45.3979	-75.6858	4	30
CF590	BANK	WILTON FS	45.3981	-75.6856	4	30
CF600	BANK	COLLISEUM M OP	45.3986	-75.6862	4	30
CF610	BANK	COLLISEUM M	45.3990	-75.6862	4	30
CF620	BANK	HOLMWOOD FS	45.3991	-75.6865	4	30
CF630	BANK	HOLMWOOD FS	45.3998	-75.6866	4	30
CF640	BANK	FIFTH NS	45.4013	-75.6877	4	30
CF650	BANK	FIFTH NS	45.4008	-75.6872	4	30
CF660	BANK	THIRD NS	45.4026	-75.6885	4	30
CF670	BANK	THIRD FS	45.4027	-75.6883	4	30
CF680	BANK	FIRST NS	45.4039	-75.6892	4	30
CF690	BANK	FIRST NS	45.4035	-75.6887	4	30
CF700	BANK	GLEBE NS	45.4047	-75.6896	4	30
CF710	BANK	GLEBE NS	45.4041	-75.6891	4	30
CF720	BANK	STRATHCONA OPP	45.4071	-75.6911	4	30
CF730	BANK	PATTERSON FS	45.4062	-75.6902	4	30
CF740	BANK	CHAMBERLAIN FS	45.4078	-75.6914	4	30
CF750	BANK	ISABELLA NS	45.4084	-75.6915	4	30
CH010	BANK	LAURIER FS	45.4183	-75.6991	4	30
CH020	BANK	NEPEAN FS	45.4176	-75.6984	4	30
CH030	BANK	LISGAR NS	45.4169	-75.6978	4	30
CH040	BANK	SOMERSET FS	45.4157	-75.6967	4	30
CH060	BANK	LEWIS NS	45.4135	-75.6947	4	30

Table D.4: Churches

NAME	ADDRESS
All Saints (Sandy Hill)	317 Chapel
All Saints (Westboro)	347 Richmond Rd
Cathedral of the Annunciation	289 Spencer
Anglican Catholic Church of Canada	190 Colonnade
Epiphany	1290 Ogilvie Rd
St. Richard's	8 Withrow
St. John the Evangelist	154 Somerset
Christ Church Anglican Bells Corner	3861 Richmond Rd
St. George's	152 Metcalfe
St. Mark's	1606 Fisher Ave
Hawthorne United	2244 Russell Rd
Barrhaven United	3013 Jockvale
Bell Street United	384 Arlington
Bell's Corners United	3955 Richmond Rd
Britannia United	985 Pinecrest
Calvary United	1681 Cyrville Rd
Carleton Memorial United	740 Melta Cr
City View United	6 Epworth Ave
Dominion Chalners United	355 Cooper St
Emmanuel United	691 Smyth
Fellowship United	119 SteepleHill
Glen Cairn United	140 Abbeyhill
Woodroffe United	207 Woodroffe
Bilberry Creek	480 Charlemagne Bl
Arabic Evangelical Baptist Church o	887 Canterbury
Bethany Baptist	380 Centrepoint Dr
Bromley Road Baptist	1900 Bromley Rd
Calvary Baptist	107 Main St
Eglise Baptiste Evangelique D'Ottaw	284 King Edward
Fellowship Baptist	1078 Klondike
Haven Baptist	4000 Strandherd
Nepean Baptist	135 Woodfield
Our Lady of Fatima	153 Woodroffe Ave
Annunciation - St. Nicholas Orthodo	55 Clarey Ave
Divine Infant Roman Catholic	6658 Bilberry Dr
Holy Redeemer	44 Rothsay
Our Lady of Perpetual Help	15 Lebreton
St. Patrick's Basilica	281 Nepean St
St. Anthony's	427 Booth St
St. Augustine's	1060 Baseline Rd
St. Barnabas	70 James
St. John the Apostle	2340 Baseline Rd

St. John the Divine	105 Slack Rd
St. Joseph	151 Laurier
St. Maurice	4 Perry St
St. Patrick's - Fallowfield Barrhav	41 Weybridge
St. Thomas the Apostle	2345 Alta Vista Dr
Abiding Word Evangelical Lutheran	1675 Belcourt Blvd
All Saints	1061 Pinecrest
Christ Risen Lutheran	85 Leacock Dr
Divine Word Evangelical Lutheran	1081 Greenbank Rd
Faith Evangelical Lutheran	43 Meadowlands
Trinity Presbyterian	110 McCurdy
Erskine Presbyterian	343 Bronson
Grace Presbyterian	1220 Tenth Line Rd
Knox Presbyterian	120 Lisgar
St. Andrew's	82 Kent St

Table D.5: Cinemas

NAME	ADDRESS
ByTowne Cinema	325 Rideau St
Coliseum	3090 Carling Ave
Famous Players	50 Rideau St
IMAX/OMNIMAX	100 Laurier
Mayfair Theatre	1074 Bank St
World Exchange	111 Albert St
Orlean Town Centre	250 Centrum Blvd
South Keys	2240 Bank St
Silvercity	2385 City Park Dr

Table D.6: Emergency

ID	NAME	ADDRESS
36	Barrhaven Base	1073 Greenbank Road
38	Industrial Base	911 Industrial Avenue
40	Land Ambulance Services - Leased	530 Tremblay Road
41	Manotick Base	5669 Main Street
42	March Base	1040 Riddell Drive
43	Metcalfe Base	8011 Victoria Street
45	Orleans Base	1439 Youville Drive
47	Ottawa 01 Base	738 Gladstone Avenue
48	Ottawa 02 Base	360 Hunt Club Road
49	Ottawa Ambulance (CACC)	2380 St Laurent Boulevard
50	Queensway Base	3045 Baseline Road
51	Richmond Base	6179 Perth Street
52	Stittsville Base	1631 Main Street
54	Cumberland Fire Station 1	500 Charlemagne Boulevard
58	Elections/Beaver Boxing	1447 Carling Avenue
59	Gloucester Fire Station 1	1700 Blair Road
60	Gloucester Fire Station 2	6213 Jeanne D'Arc Boulevard
61	Gloucester Fire Station 3	3202 Leitrim Road
62	Gloucester Fire Station 4	3080 Old Innes Road
63	Goulbourn Fire Station 1 and Ambulance Base - Stittsville	1631 Main Street
64	Goulbourn Fire Station 2 - Richmond	6280 Perth Street
65	Kanata Fire Station 1	1021 Teron Road
66	Kanata Fire Station 2	1040 Riddell Drive
67	Kanata Fire Station 3	380 Eagleson Road
68	Nepean Fire Station 1 - Bells Corners	3845 Richmond Road
69	Nepean Fire Station 2 - Viewmount	230 Viewmount Drive
70	Nepean Fire Station 3 - Barhaven	1075 Greenbank Road
71	Nepean Fire Station 4 - Knoxdale	60 Knoxdale Road
72	Osgoode Fire Station 1 - Metcalfe	8011 Victoria Road
74	Osgoode Fire Station 3 - Greely	6891 Parkway
76	Ottawa Fire Station 1	1443 Carling Avenue
77	Ottawa Fire Station 2	135 Preston Street
78	Ottawa Fire Station 3	530 King Edward Avenue
79	Ottawa Fire Station 4	900 Industrial Avenue
80	Ottawa Fire Station 5	900 Montreal Road
81	Ottawa Fire Station 6	220 Beechwood Avenue
82	Ottawa Fire Station 7	700 Brookfield Road
83	Ottawa Fire Station 8	2355 Altavista Drive
84	Ottawa Fire Station 9	3336 McCarthy Road
85	Ottawa Fire Station 10	635 O'Connor Street

86	Ottawa Fire Station 11	1300 Woodroffe Avenue
87	Ottawa Fire Station 12	1397 Richmond Road
88	Ottawa Fire Station 13	275 Coventry Road
89	Ottawa Fire Station 14	3255 Conroy Road
90	Ottawa MacDonald-Cartier International Airport Fire Station	302 Campbell Road
92	Rideau Fire Station 2 - Manotick	5669 Main Street
99	Barrhaven Community Police Centre/ Barrhaven poste de police communautaire	101 Malvern Drive
100	Bayshore Community Police Centre / Bayshore poste de police communautaire	98 Woodridge Crescent
101	Bells Corners Community Police Centre / Bells Corners poste de police communautaire	2E Lovell Lane
102	Carlington Community Police Centre/ Carlington poste de police communautaire	1309 Carling Ave
103	Central Police Headquarters / Poste de police centrale	474 Elgin Street
105	Cumberland Community Police Centre / Cumberland poste de police communautaire	260 Tompkins Road
106	Cyrville Community Police Centre / Cyrville poste de police communautaire	1980 Ogilvie Road
107	East Division Headquarters / Div. est poste de police	4561 Highway 31
109	Kanata Community Police Centre / Kanata poste de police communautaire	95 Abbeyhill Drive
110	Lowertown Foot Patrol / Patrouille pΘdestre de la Basse-Ville	368 Dalhousie Street
111	Manotick Community Police Centre/ Manotick poste de police communautaire	5542 Ann Street
112	Metcalfe Community Police Centre/ Metcalfe poste de police communautaire	8253B Victoria Street
113	Orleans Community Police Centre / Poste de police communautaire d'OrlΘans	5929 Jeanne D'Arc Boulevard
115	Parkwood Hills Community Police Centre / Parkwood Hills poste de police communautaire	1343 Meadowlands Drive
116	Police Station - Leased	1329 Wellington Street
117	Police Station - Leased	21 Concourse Gate
118	Professional Development Centre/ Centre de perfectionnement professionnel	1385 Woodroffe Avenue
119	Regional Office / Bureau de la Region	111 Lisgar Street
120	Richmond Community Police Centre/ Richmond poste de police communautaire	6179 Perth Street
121	Rideau Street Community Police Centre / rue Rideau poste de police communautaire	362 Rideau Street
122	Rockcliffe Park Community Police Centre / Rockcliffe Park poste de police communautaire	360 Springfield Road
123	Somerset Community Police Centre / Somerset poste de police communautaire	393 Somerset Street West
124	South Ottawa Community Police Centre / Ottawa-Sud poste de police communautaire	2870 Cedarwood Drive
125	Stittsville Community Police Centre / Stittsville poste de police communautaire	1532 Main Street

126	Vanier Community Police Centre / Vanier poste de police communautaire	252 McArthur Road
128	West Division Headquarters and Community Police Centre / Div. ouest et poste de police communautaire	245 Greenbank Road
129	Westboro Community Police Centre / Westboro poste de police communautaire	418A Richmond Road
130	Youth Gym / Gymnase des jeunes	1463 Prince of Wales Drive

Table D.7: Grocery Stores

NAME	ADDRESS
A & P/Superfresh McCarth	3310 McCarthy Rd
A & P/Superfresh Heron	1670 Heron Rd
Bulk Barn	2210 Bank St
Costco	1900 Cyrville Rd
Farm Boy Fresh Merivale	1642 Merivale Rd
Farm Boy Fresh Montreal	585 Montreal Rd
Farm Boy Fresh Hazeldean	457 Hazeldean Rd
Farm Boy Fresh Centrum	240 Centrum Rd
Food Basics Kirkwood	667 Kirkwood Ave
Food Basics Charlemagne	470 Charlemagne
Food Basics Katimavik	150 Katimavik
IGA Barrhaven	900 Greenbank Rd
Loblaws Ogilvie	1980 Ogilvie Rd
Loblaws Orleans	1226 Orleans Blvd
Loblaws Eagleson	700 Eagleson Rd
Loblaws Merivale	1460 Merivale Rd
Loblaws Robertson	59 Robertson Rd
Loblaws Strandherd	3777 Strandherd
Loblaws Bank	2210 Bank St
Loblaws Carling	2085 Carling Ave
Loblaws Iris	2685 Iris
Loblaws Isabella	64 Isabella
Loblaws Rideau	363 Rideau St
Loblaws Riverside	2241 Riverside Dr
Loblaws St. Laurent	1910 St Laurent Bl
Loblaws McArthur	100 McArthur Ave
Loeb Merivale	1512 Merivale Rd
Loeb Booth	314 Booth St
Loeb Bank	754 Bank St
Loeb Rideau	245 Rideau St
Loeb Walkey	2261 Walkley Rd
Loeb Carling	4048 Carling Ave
Loeb Richmond	3655 Richmond Rd
Loeb Carleton	345 Carleton
Loeb South Keys	2515 Bank St
Loeb Gladwin	2212 Gladwin Cr
Loeb Beechwood	50 Beechwood Ave
Loeb Montreal	1930 Montreal Rd
Loeb St.Laurent	1021 St. Laurent
Loeb McArthur	320 McArthur Ave
Loeb Innes	4510 Innes

Loeb 1360 Richmond	1360 Richmond
Loeb Greenbank	250 Greenbank Rd
Loeb Hazeldean	420 Hazeldean
Loeb Jeanne D'Arc	6509 Jeanne d'Arc
Loeb Tenth Line	1675 Tenth Line
Market Fresh St. Laurent	1200 St. Laurent B
Market Fresh Selkirk	29 Selkirk
Sobeys	700 Terry Fox Dr
YIG Carling	2148 Carling Ave
YIG Laurier	2737 Laurier Ave
YIG Eagleson	300 Eagleson Rd
YIG Bank	296 Bank St
YIG Grant Carmen	200 Grant Carmen
YIG Orleans	1619 Orleans Blvd
YIG Alta Vista	2681 Alta Vista Dr
YIG 2210 Bank	2210 Bank St

Table D.8: Hospitals

ID	NAME	ADDRESS
1	Royal Ottawa Hospital	1145 Carling Avenue
2	Ottawa Hospital - Civic Campus	1053 Carling Avenue
3	Ottawa Hospital - Riverside Campus	1967 Riverside Drive
4	Children's Hospital of Eastern Ontario	401 Smyth Road
5	Ottawa Hosptial - General Campus	501 Smyth Road
6	Montfort Hospital	713 Montreal Road
8	Queensway-Carleton Hospital	3045 Baseline Road
9	Rehabilitation Centre	505 Smyth Road
10	Saint Vincent Hospital	60 Cambridge Street North
11	Elizabeth Bruyere Hospital	43 Bruyere Street

Table D.9: Hotels - Motels

NAME	ADDRESS
Chateau Laurier	1 Rideau St.
Crowne Plaza	101 Lyon St.
Delta Ottawa Hotels and Suites	361 Queen St.
Quality Hotel	290 Rideau St.
Travelodge	402 Queen St.
Westin Ottawa	11 Colonel By Drive
WelcomINNS	1220 Michael St.
Travelodge and Convention Cent	1376 Carling Ave
McKellar Park Apt/Hotel	1983 Carling Ave
Adam's Airport Inn	2721 Highway 31
Monterey Motor Inn	2259 Highway 16
Southway Inn	2431 Bank St

Table D.10: Hotlink

ID	HOTLINK
0	http://www.octranspo.com

Table D.11: Libraries

ID	NAME	ADDRESS
1	Main Library	120 Metcalfe St
2	Alta Vista	2516 Alta Vista Dr
3	Carlingwood	281 Woodroffe Av
4	Elmvale Acres	1910 St. Laurent Bl
5	Rideau	377 Rideau St
6	Sunnyside	1049 Bank St
7	St. Laurent	515 Cote St
8	Rosemount	18 Rosemount Av
9	Beaverbrook	2500 Campeau Drive
10	Hazeldean	50 Castlefrank Road
11	Nepean Central	101 Centrepointe Dr
12	Centenial	3870 Richmond Road
13	Emerald Plaza	1547 Merivale Road
14	Ruth E. Dickinson	101 Malvern Dr
15	Cumberland	1599 Tenth Line Road
16	Blossom Park	2950 Highway 31
17	North Gloucester	2036 Ogilvie Road
18	Orleans	1705 Orleans Bl
19	Blackburn Hamlet	199 Glen Park Dr
20	Greely	7008 Parkway Rd.
21	Metcalfe	2782 Albert St.
24	Manotick	5499 South River Dr.
26	Rockcliffe Park	380 Springfield Road
28	Stittsville	1637 Main Street
30	Richmond	6240 Perth Street

Table D.12: Location

ID	ADDRESS	NAME
1		
2		

Table D.13: Museums

NAME	ADDRESS
Canada Aviation Museum	11 Aviation
Science and Technolgy	1867 St Laurent
Museum of Nature	240 McLeod
Billings Estate	2100 Cabot
Museum of Contemporary Photography	1 Rideau
Canadian War Museum	330 Sussex
Carleton University Art Gallery	1125 Colonel By
Currency Museum of the Bank of Cana	245 Sparks street
Royal Canadian Mint	320 Sussex
National Gallery of Canada	380 Sussex

Table D.14: Recreation Centers

NAME	ADDRESS
RA Centre	2451 Riverside Dr
Blackstar Community & Resource Cent	1800 Bank St
Coliseum Sports & Recreation Centre	1015 Bank St
Dovercourt Recreation Centre	411 Dovercourt
Hunt Club - Riverside Community Cen	3320 Paul Anka Dr
Kanata Leisure Centre	70 Aird Pl
James Patrick Sports Centre	350 Hunt Club Rd
Nepean Sportsplex	1701 Woodroffe Ave
Ottawa Boys & Girls Club	1463 Prince of Wal
Overbrook Community Recreation Asso	33 Quill
Walter Baker Sports Centre	100 Malvern Dr

Table D.15: Schools

NAME	STREET
A. Y. Jackson Secondary	150 Abbeyhill Dr
Adult High School	300 Rochester St
Agincourt Public School	1250 Agincourt Rd
Albert Street Secondary Alternative	440 Albert St
Alexander Montessori	1002 Karen Way
Alta Vista Public School	1349 Randall Av
Arc-en-ciel (école)	1830 Portobello Bl
Arch Street Public School	2129 Arch St
Ashbury College	362 Mariposa Av
Assumption Catholic E.S.	330 Lajoie St
Barrhaven Public School	80 Larkin Dr
Bayshore Catholic E.S.	50 Bayshore Dr
Bayshore Public School	145 Woodridge Cr
Bayview Public School	3071 Riverside Dr
Bell High School	40 Cedarview Rd
Bell's Corners Public School	3770 Richmond Rd
Bishop Hamilton Chris. Montess.	2199 Regency Tr
Blessed Kateri Tekakwitha Catholic E.S.	6400 Beausejour Dr
Blossom Park Public School	3810 Sixth St
Briargreen Public School	19 Parkfield Cr
Bridlewood Community E.S.	63 Bluegrass Dr
Broadview Public School	590 Broadview Av
Brookfield High School	824 Brookfield Rd
Cadieux (école)	345 St Denis St
Cairine Wilson Secondary School	975 Orleans Bl
Cambridge St. Comm. Public School	250 Cambridge St, N
Canadian Montessori Academy	2 Peter St
Canterbury High School	900 Canterbury Av
Carleton Heights Public School	1660 Prince Of Wales Dr
Carson Grove Elementary School	1401 Matheson Rd
Castlefrank Elementary School	55 McCurdy Dr
Cedarview Middle School	2760 Cedarview Rd
Centennial Public School	376 Gloucester St
Centre De Jour	140 Genest St
Centre Jules Leger	281 Lanark St
Century Public School	8 Redpine Dr
Chapel Hill Catholic E.S.	1534 Forest Valley Dr
Charles H. Hulse Public School	2605 Alta Vista Dr
Christie Public School	2625 Draper Av
Churchill School	345 Ravenhill Av
Clifford Bowey Public School	1300 Kitchener Av

Colonel By Secondary School	2381 Ogilvie Rd
Connaught Public School	1149 Gladstone Av
Convent Glen Catholic E.S.	6212 Jeanne D'arc Bl
Convent Glen Elementary School	1708 Grey Nuns Dr
Corpus Christi Catholic E.S.	157 Fourth Av
Counterpoint Children's Academy	1151 River Rd
Crystal Bay Centre for Special Ed.	31 Moodie Dr
D. A. Moodie Intermediate School	595 Moodie Dr
D. Roy Kennedy Public School	919 Woodroffe Av
Des Pins (école)	1487 Ridgebrook Dr
Des Pionniers (école)	720 Merkley Dr
Des Villageois (école)	1550 Verchere St
Des Voyageurs (école)	6030 Voyageur Dr
Devonshire Comm. Public School	100 Breezehill Av
Divine Infant Catholic E.S.	8100 Jeanne D'arc Bl
Dr. F. J. McDonald Catholic E.S.	2860 Ahearn Av
Dunlop Public School	1310 Pebble Rd
Dunning-Foubert Elementary School	1610 Prestwick Dr
É.é.p. Charlotte-Lemieux	2093 Bel Air Dr
É.é.p. Francojeunesse	119 Osgoode St
É.é.p. Franco-Nations	1085 Grenon Av
É.é.p. Gabrielle-Roy	1000 Brookfield Dr
É.é.p. Jeanne-Sauvé	1917 Gardenway Dr
É.é.p. Le Trillium	135 Alice St
É.é.p. L'Odyssée	1770 Grey Nuns Dr
É.é.p. Marie-Curie	860 Colson Av
É.é.p. Séraphin-Marion	22 Steel St
É.p. Le Transit	140 Genest St
É.s.p. De La Salle	501 St Patrick St
É.s.p. de l'Ouest d'Ottawa	1303 Fellows Rd
É.s.p. Louis-Riel	1655 Bearbrook Rd
Earl Of March Secondary School	948 Teron Rd
Éducation permanente	623 Smyth Rd
Éducation sans Frontières	181 Donald St
Elgin Street Public School	310 Elgin St
Élisabeth-Bruyère (école)	100 Stonehaven Dr
Elizabeth Park Public School	15 De Niverville Dr
Elizabeth Wyn Wood Site	20 Rossland Av
Elmdale Public School	49 Iona St
Elmridge Catholic E.S.	1923. Elmridge Dr
Elmwood School	261 Buena Vista Rd
Emily Carr Middle School	2681 Innes Rd
Fallingbrook Community E.S.	679 Deancourt Cr
Featherston Drive Public School	1801 Featherston Dr
Fernhill School	1050 Naughton St
Fielding Drive Public School	777 Fielding Dr

First Avenue Public School	73 First Av
Fisher Park / Summit Public	250 Holland Av
Forest Valley Elementary School	1570 Forest Valley Dr
Franco-Cité (école secondaire cath.)	623 Smyth Rd
Franco-ouest (Collège cath.)	411 Seyton Dr
Frank Ryan Catholic I.S.	128 Chesterton Dr
Frederick Banting Site	115 Terrence Matthews Cr
Gameau (école secondaire cath.)	6588 Carriere St
General Vanier Public School	1025 Harkness Av
Georges Vanier Catholic E.S.	40 Varley Dr
Georges-Étienne-Cartier (école)	880 Thorndale Dr
Glashan Public School	28 Arlington Av
Glebe Collegiate Institute	212 Glebe Av
Glen Cairn Public School	182 Morrena Dr
Glen Ogilvie Public School	46 Centre Park Dr
Gloucester High School	2060 Ogilvie Rd
Good Shepherd Catholic E.S.	101 Bearbrook Rd
Grant School	2720 Richmond Rd
Greenbank Middle School	168 Greenbank Rd
Hawthorne Public School	2158 St Laurent Blvd
Head Start Academy (1st fl)	88 Main St
Henry Larsen Elementary School	1750 Sunview Dr
Henry Munro Middle School	2105 Kender Av
Heritage Academy	2360 Virginia St
Hillcrest High School	1900 Dauphin Rd
Hillel Academy	881 Broadview Av
Hilson Avenue Public School	407 Hilson Av
Holy Cross Catholic E.S.	2820 Springland Dr
Holy Family Catholic E.S.	245 Owl Dr
Holy Redeemer Catholic E.S.	75 Mccurdy Dr
Holy Trinity Catholic H.S.	180 Katimavik Rd
Hopewell Avenue Public School	17 Hopewell Av
Immaculata Catholic I.S. and H.S.	140 Main St
Immaculate Heart of Mary Catholic E.S.	445 Pleasant Park Rd
J. H. Putman Public School	2051 Bel Air Dr
J. S. Woodsworth Secondary School	159 Chesterton Dr
Jean Vanier Catholic I.S.	320 Lajoie St
Joan Of Arc Academy	209 Bayswater Av
Jockvale Elementary School	101 Malvern Dr
John McCrae Secondary School	103 Malvern Dr
John Young Elementary School	5 Morton Dr
Kanata Academy	209 Bearverbrook Rd
Katimavik Elementary School	64 Chimo Dr
Knoxdale Public School	170 Greenbank Rd
La Vérendrye (école)	614 Eastvale Dr
Lady Evelyn Public School	63 Evelyn Av

Lakeview Public School	35 Corkstown Rd
L'Alternative	307 Montgomery St
Lamira Dow Billings E.S.	2147 Loyola Av
Lamoureux (école)	2540 Kaladar Ave
Laurentian High School	1357 Baseline Rd
Laurier-Carrière (école)	14 Four Seasons Dr
Le Carrefour	2315 St Joseph Bl
Le Petit Prince (école)	349 Olmstead St
Le Phare Elementary School	1965 Naskapi Dr
Leitrim Montessori Preschool	4561 Highway 31
Léo-D.-Côte (école intermédiaire)	6664 Carriere St
Leslie Park Public School	20 Harrison St
Lester B. Pearson Catholic H.S.	2072 Jasmine Cr
Life Christian Academy	2214 Innes Rd
Lisgar Collegiate Institute	29 Lisgar St
L'Odyssee College International	839 Shefford St
Lycee Claudel (ecole)	1635 Riverside Dr
MacDonald Cartier Academy	270 Crichton St
Machon Sarah High School	831 Broadview Av
Maimonides	25 Esquimault Av
Manor Park Public School	100 Braemar St
Manordale Public School	16 Carola St
Maple Ridge E.S.	1000 Valin St
Marius Barbeau (école)	1345 Nottinghill Av
Mary Honeywell E.S.	54 Kennevale Dr
McArthur High School	485 Donald St
McGregor Easson Public School	991 Dynes Rd
McMaster Catholic E.S.	1760 Mcmaster Av
Meadowlands Public School	10 Fieldrow St
Merivale High School	1755 Merivale Rd
Merivale Public School	63 Slack Rd
Montfort (école)	641 Sladen Av
Mother Teresa Catholic H.S.	440 Longfields Dr
Mutchmor Public School	185 Fifth Av
Nepean High School	574 Broadview Av
Norman Johnston Site	2401 Cleroux Cr
Notre Dame Catholic H.S.	710 Broadview Av
Notre-Dame-des-Champs (école)	3349 Navan Rd
Nouveaux-Horizons (école)	3681 Innes Rd
Orleans Wood Elementary School	7859 Decarie Dr
Ottawa Christian Elemen. School	2191 Benjamin Av
Ottawa Islamic Elementary	10 Coral St
Ottawa Montessori School	335 Lindsay St
Ottawa Seventh Day Adventist	2200 Benjamin Av
Ottawa Torah Institute High	151 Chapel St
Our Lady Of Fatima Catholic E.S.	2135 Knightsbridge Rd

Our Lady Of Mount Carmel Catholic E.S.	675 Gardenvale Rd
Our Lady Of Peace Catholic E.S.	3877 Richmond Rd
Our Lady Of Victory Catholic E.S.	1175 Soderlind St
Our Lady Of Wisdom Catholic E.S.	1565 St George St
Overbrook Public School	149 King George St
Parkwood Hills Public School	60 Tiverton Dr
Parsifal (ecole)	630 Island Park Dr
Pauline-Vanier (école intermédiaire)	1837 Stonehenge Cr
Pinecrest Public School	1281 Pinecrest Rd
Pineview Catholic E.S.	1500 Beaverpond Dr
Pleasant Park Public School	564 Pleasant Park Rd
Pope John XXIII Catholic E.S.	165 Knoxdale Rd
Préseault (école)	1080 St Pierre
Prince Of Peace Catholic E.S.	1620 Heatherington Rd
Queen Elizabeth Public School	689 St-Laurent Bl
Queen Mary Street Public School	557 Queen Mary St
Queen of the Angels Adult H.S.	1461 Heron Rd
Queenswood Public School	1445 Duford Dr
R. Byms Curry Public School	185 Owl Dr
Redeemer Christian High School	82 Colonnade Rd
Regina Public School	2599 Regina St
Reine-des-Bois (école)	1450 Duford St
Richmond Public School	3499 Mcbean St
Rideau High School	815 St-Laurent Bl
Ridgemont High School	2597 Alta Vista Dr
Riverview Public School (Ottawa)	260 Knox Cr
Robert Bateman Public School	1250 Blohm Dr
Robert E. Wilson Public School	373 Mcarthur Av
Robert Hopkins Elementary School	2011 Glenfern Av
Roberta Bondar Public School	159 Lorry Greenberg Dr
Rockcliffe Park Public School	350 Buena Vista Rd
Roger-Saint-Denis (école)	186 Barrow Cr
Roland Michener Public School	100 Penfield Dr
Sainte-Anne (école)	340 York St
Sainte-Bernadette (école)	3781 Sixth St
Sainte-Geneviève (école)	2198 Arch St
Sainte-Marie (école)	2599 Innes Rd
Saint-François-d'Assise (école)	35 Melrose Av
Saint-Luc (école)	2485 Dwight Cr
Samuel-Genest (Collège cath.)	704 Carson Rd
Sawmill Creek Elem. School	3400 Daoust Av
Severn Avenue Public School	2553 Severn Av
Sir Guy Carleton Secondary School	55 Centrepoint Dr
Sir Robert Borden High School	131 Greenbank Rd
Sir Wilfrid Laurier Secondary School	1515 Tenth Line Rd
Sir Winston Churchill Public School	49 Mulvagh Av

South Carleton High School	3673 Mcbean St
St. Andrew Adult H.S	1119 Lazard St
St. Anne School	500 Stonehaven Dr
St. Anthony Catholic E.S.	391 Booth St
St. Augustine Catholic E.S.	1009 Arnot Rd
St. Bernard Catholic E.S.	1722 St Bernard St
St. Brigid Catholic E.S.	200 Springfield Rd
St. Clare Catholic E.S.	2133 Gardenway Dr
St. Daniel Catholic E.S.	1160 Maitland Av
St. Elizabeth Ann Seton Catholic E.S.	41 Weybridge Dr
St. Elizabeth Catholic E.S.	893 Admiral Av
St. Francis of Assisi Catholic E.S.	795 Watters Rd
St. George Catholic E.S.	401 Picadilly Av
St. Gregory Catholic E.S.	70 Fieldrow St
St. Isidore Catholic E.S.	1105March Rd
St. James Catholic E.S.	50 Stonehaven Dr
St. John the Apostle Catholic E.S.	30 Costello Av
St. Joseph Adult H.S.	20 Graham St
St. Joseph Catholic I.S.	130 Keyworth Av
St. Jude's Academy (Home Study)	112 Gardenview Private
St. Luke Catholic E.S. (Nepean)	60 Mountshannon Dr
St. Luke Catholic E.S. (Ottawa)	2485 Dwight Cr
St. Margaret Mary Catholic E.S.	88 Bellwood Av
St. Marguerite d'Youville Catholic E.S.	89 Lorry Greenberg Dr
St. Martin De Porres Catholic E.S.	20 Mckitrick Dr
St. Mary Catholic E.S. (Gloucester)	5536 Highway 31
St. Mary Catholic E.S. (Ottawa)	175 Beech St
St. Matthew Catholic H.S.	6550 Bilberry Dr
St. Michael Catholic E.S. (Ottawa)	741 Bernard St
St. Monica Catholic E.S.	2000 Merivale Rd
St. Nicholas Adult H.S.	130 Lotta Av
St. Patrick Adult H.S.	290 Nepean St
St. Patrick Catholic E.S. (Nepean)	68 Larkin Dr
St. Patrick Catholic H.S.	2525 Alta Vista Dr
St. Patrick Catholic I.S.	1485 Heron Rd
St. Paul Catholic H.S.	2611 Draper Av
St. Peter Catholic H.S.	750 Charlemagne Bl
St. Philip Catholic E.S.	79 Maitland Av
St. Pius X Catholic H.S.	1481 Fisher Av
St. Rita Catholic E.S.	1 Inverness Rd
St. Thomas Catholic E.S.	9 Leeming Dr
St. Thomas More Catholic E.S.	1620 Blohm Dr
Stephen Leacock Public School	25 Leacock Dr
Terre-des-Jeunes (pav. Gaston-Vincent)	1313 Field St
Terre-des-Jeunes (pav. St-Bonaventure)	1366 Coldrey Av
Terry Fox Elementary School	6400 Jeanne D'arc Bl

Thomas D'Arcy McGee Catholic E.S.	635 Laverendrye Dr
Torah Academy of Ottawa	1231 Prince of Wales Dr
Trillium Elementary School	1515 Varennes Bl
Turnbul Learning Center	1132 Fisher Av
Uplands Catholic E.S.	17 De Niverville Dr
Vincent Massey Public School	745 Smyth Rd
Viscount Alexander Public School	65 Mann Av
Vision Jeunesse (école)	235 Mcarthur Ave
W. E. Gowling School	250 Anna Av
W. Erskine Johnston Public School	50 Varley Dr
W. O. Mitchell Elementary School	80 Steeple Chase Dr
Westboro Academy	130 Keyworth St
Woodroffe Avenue Public School	235 Woodroffe Av
Woodroffe High School	2410 Georgina Dr
Yitzhak Rabin High School	1755 Merivale Rd
York Street Public School	310 York St

Table D.16: Shopping Centers

NAME	ADDRESS
Rideau Centre	73 Rideau St.
Bayshore Shopping Centre	100 Bayshore Dr.
St. Laurent Shopping Centre	1200 St. Laurent Bv
Place d'Orleans	110 Place d'Orleans
Merivale Mall	1642 Merivale Rd.
Heron Gate Mall	1670 Heron Rd.
Carlingwood Mall	2121 Carling Ave
Westgate Shopping Centre	1309 Carling Ave

Table D.17: Tourist Attractions

SITE	ADDRESS
Parliament Hill	Wellington St.
Supreme Court of Canada	Wellington St.
National Archives & National Library of Canada	395 Wellington S
Currency Museum	245 Sparks St.
National Arts Centre	53 Elgin St.
National War Memorial	Elgin St.
Museum of Contemporary Photography	1 Rideau St.
Ottawa Congress Centre	55 Colonel By Dr
Peacekeeping monument	390 Sussex Dr.
National Gallery of Canada	380 Sussex Dr.
Canadian War Museum	330 Sussex Dr.
Royal Canadian Mint	320 Sussex Dr.
Canadian Museum of Nature	240 McLeod St.
Canadian Aggriculture Museum	1 Experimental F
Prime Minister's House	24 Sussex Dr.
Rideau Hall	1 Sussex Dr.
Carleton University Art Gallery (St. Patrick Bld)	1125 Colonel By
Billings Estate	210 Cabot St.
Bytown Museum	1 Rideau St.
Aviation Museum	11 Aviation Park
Museum of Science and Technology	1867 St. Laurent
Laurier House	334 Laurier Ave
Vimy House	221 Champagne Av
Civic Centre/Lansdowne Park	1015 Bank St.
Corel Centre	1000 Palladium D
St. Patrick's Basilica	261 Nepean St.
Notre Dame Cathedral	60 Guigues St.

Appendix E
Calalp.ave

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
 Name:  View.CalculateFeatureGeometry
'
' Title:  Calculates feature geometry values
'
' Topics:  GeoData
'
' Description:  Calculates area and perimeter for polygon themes and length
' for line themes. If the View has been projected the calculations are in
' projected meters. Otherwise the calculations are in 'native' map units.
' Modify the script to provide calculation in the current report units of
' the View. The script processes the list of active themes to calculate
' area and perimeter, or length, depending on the theme type.
'
' The script will add the fields: Area and Perimeter to polygon themes, Length
' to line themes if they do not exist. If the fields exist their values will
' be recalculated. Rerun the script if you change the projection of the view.
'
' Requires:  A View with at least one active theme.  You must have write access
' to the active theme(s).
'
' Self:
'
' Returns:
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' Get the view and its projection if any.
'
theView = av.GetActiveDoc
thePrj = theView.GetProjection
if (thePrj.IsNull) then
  hasPrj = false
else
  hasPrj = true
end

' Get the list of active themes. if there aren't any, let the user know
' and exit.
'
theActivethemeList = theView.GetActivethemes
if (theActivethemeList.Count = 0) then
  MsgBox.Error("No active themes.","")
  Exit
end

' Loop through the list of active themes. if you can't edit the theme
' inform the user.
'
For Each thetheme in theActivethemeList
  theFTab = thetheme.GetFTab
  if (theFTab.CanEdit.Not) then
    MsgBox.Info("Cannot edit table for theme:"++thetheme.AsString,"")
    Continue
  end
  '
  ' Make the FTAB editable, and find out which type of feature it is.
```

```
'
theFTab.SetEditable(TRUE)
theType = theFTab.FindField("shape").GetType
if (theType = #FIELD_SHAPEPOLY) then
 '
 ' if it's polygonal check for the existence of the fields "Area" and
 ' Perimeter. if they do not exist, create them.
 '
 if (theFTab.FindField("Area") = nil) then
  theAreaField = Field.Make("Area",#FIELD_DOUBLE,16,3)
  theFTab.AddFields({theAreaField})
 else
  ok = MsgBox.YesNo("Update Area?", "Calculate", true)
  if (ok.Not) then
   continue
  end

  theAreaField = theFTab.FindField("Area")
 end

 if (theFTab.FindField("Perimeter") = nil) then
  thePerimeterField = Field.Make("Perimeter",#FIELD_DOUBLE,16,3)
  theFTab.AddFields({thePerimeterField})
 else
  ok = MsgBox.YesNo("Update Perimeter?", "Calculate", true)
  if (ok.Not) then
   continue
  end

  thePerimeterField = theFTab.FindField("Perimeter")
 end

 '
 ' Loop through the FTAB and find the projected area and perimeter of each
 ' shape and set the field values appropriately.
 '
 theShape = theFTab.ReturnValue(theFTab.FindField("shape"),0)
 For Each rec in theFTab
  theFTab.QueryShape(rec,thePrj,theShape)

  theArea = theShape.ReturnArea
  thePerimeter = theShape.ReturnLength

  theFTab.SetValue(theAreaField,rec,theArea)
  theFTab.SetValue(thePerimeterField,rec,thePerimeter)
 end

elseif (theType = #FIELD_SHAPELINE) then
 '
 ' if the data source is linear, check for the existence of the
 ' field "Length". if it doesn't exist, create it.
 '
 if (theFTab.FindField("Length") = nil) then
  theLengthField = Field.Make("Length",#FIELD_DOUBLE,16,3)
  theFTab.AddFields({theLengthField})
```

```
    else
      ok = MsgBox.YesNo("Update Length?", "Calculate", true)
      if (ok.Not) then
        continue
      end

      theLengthField = theFTab.FindField("Length")
    end

    '
    ' Loop through the FTAB and find the projected length of each shape and set
    ' the field values appropriately.
    '
    theShape = theFTab.ReturnValue(theFTab.FindField("shape"),0)
    For Each rec in theFTab
      theFTab.QueryShape(rec,thePrj,theShape)

      theLength = theShape.ReturnLength

      theFTab.SetValue(theLengthField,rec,theLength)
    end

  end

  theFTab.SetEditable(FALSE)
end
```

Appendix F
Cfcc.dbf

CFCC	DESCRIPTION	SPEED_	DISPLAY	FCC
A00	Road, major and minor categories unknown	40	4	A00
A01	Road, unseparated	40	4	A01
A02	Road, unseparated, in tunnel	40	4	A02
A03	Road, unseparated, underpassing	40	4	A03
A04	Road, unseparated, with rail line in center	40	4	A04
A05	Road, separated	40	4	A05
A06	Road, separated, in tunnel	40	4	A06
A07	Road, separated, underpassing	40	4	A07
A08	Road, separated, with rail line in center	40	4	A08
A10	Primary road with limited access or interstate highway	65	1	A10
A11	Primary road with limited access or interstate highway, unseparated	65	1	A11
A12	Primary road with limited access or interstate highway, unseparated, i	65	1	A12
A13	Primary road with limited access or interstate highway, unseparated, u	65	1	A13
A14	Primary road with limited access or interstate highway, unseparated, v	65	1	A14
A15	Primary road with limited access or interstate highway, separated	70	1	A15
A16	Primary road with limited access or interstate highway, separated, in t	70	1	A16
A17	Primary road with limited access or interstate highway, separated, unc	70	1	A17
A18	Primary road with limited access or interstate highway, separated, witl	70	1	A18
A20	Primary road without limited access, U.S. and State highway	55	2	A20
A21	Primary road without limited access, U.S. and State highways, unsepa	55	2	A21
A22	Primary road without limited access, U.S. and State highways, unsepa	55	2	A22
A23	Primary road without limited access, U.S. and State highways, unsepa	55	2	A23
A24	Primary road without limited access, U.S. and State highways, unsepa	55	2	A24
A25	Primary road without limited access, U.S. and State highways, separa	60	2	A25
A26	Primary road without limited access, U.S. and State highways, separa	60	2	A26
A27	Primary road without limited access, U.S. and State highways, separa	60	2	A27
A28	Primary road without limited access, U.S. and State highways, separa	60	2	A28
A30	Secondary and connecting road, State and county highways	45	3	A30
A31	Secondary and connecting road, State and county highways, unsepar	45	3	A31
A32	Secondary and connecting road, State and county highways, unsepar	45	3	A32
A33	Secondary and connecting road, State and county highways, unsepar	45	3	A33
A34	Secondary and connecting road, State and county highways, unsepar	45	3	A34
A35	Secondary and connecting road, State and county highways, separate	50	3	A35
A36	Secondary and connecting road, State and county highways, separate	50	3	A36
A37	Secondary and connecting road, State and county highways, separate	50	3	A37
A38	Secondary and connecting road, State and county highway, separatec	50	3	A38
A40	Local, neighborhood, and rural road, city street	25	5	A40
A41	Local, neighborhood, and rural road, city street, unseparated	25	5	A41
A42	Local, neighborhood, and rural road, city street, unseparated, in tunne	25	5	A42
A43	Local, neighborhood, and rural road, city street, unseparated, underpa	25	5	A43
A44	Local, neighborhood, and rural road, city street, unseparated, with rail	25	5	A44
A45	Local, neighborhood, and rural road, city street, separated	35	5	A45
A46	Local, neighborhood, and rural road, city street, separated, in tunnel	35	5	A46
A47	Local, neighborhood, and rural road, city street, separated, underpass	35	5	A47
A48	Local, neighborhood, and rural road, city street, separated, with rail lin	35	5	A48
A50	Vehicular trail, road passable only by four-wheel drive (4WD) vehicle	15	6	A50
A51	Vehicular trail, road passable only by 4WD vehicle, unseparated	15	6	A51
A52	Vehicular trail, road passable only by 4WD vehicle, unseparated, in tu	15	6	A52
A53	Vehicular trail, road passable only by 4WD vehicle, unseparated, und	15	6	A53

A60	Road with characteristic unspecified	40	4	A60
A61	Cul-de-sac, the closed end of a road that forms a loop or turn aro	25	5	A61
A62	Traffic circle, the portion of a road or intersection of roads that foı	25	5	A62
A63	Access ramp, the portion of a road that forms a cloverleaf or limit	55	5	A63
A64	Service drive	25	5	A64
A65	Ferry crossing	5	7	A65
A70	Other thoroughfare	5	7	A70
A71	Walkway, nearly level road for pedestrians, usually unnamed	1	7	A71
A72	Stairway, stepped road for pedestrians, usually unnamed	1	7	A72
A73	Alley, road for service vehicles, usually unnamed, located at the ı	25	5	A73
B00	Railroad, major and minor categories unknown	0	0	B00
B01	Railroad track, not in tunnel or underpassing	0	0	B01
B02	Railroad track, in tunnel	0	0	B02
B03	Railroad track, underpassing	0	0	B03
B10	Railroad main track	0	0	B10
B11	Railroad main track, not in tunnel or underpassing	0	0	B11
B12	Railroad main track, in tunnel	0	0	B12
B13	Railroad main track, underpassing	0	0	B13
B20	Railroad spur track	0	0	B20
B21	Railroad spur track, not in tunnel or underpassing	0	0	B21
B22	Railroad spur track, in tunnel	0	0	B22
B23	Railroad spur track, underpassing	0	0	B23
B30	Railroad yard track	0	0	B30
B31	Railroad yard track, not in tunnel or underpassing	0	0	B31
B32	Railroad yard track, in tunnel	0	0	B32
B33	Railroad yard track, underpassing	0	0	B33
B40	Railroad ferry crossing	0	0	B40
B50	Other rail line	0	0	B50
B51	Carline, a track for street cars, trolleys, and other mass transit rai	0	0	B51
B52	Cog railroad, incline railway, or logging tram	0	0	B52
C00	Miscellaneous ground transportation, not road or railroad	0	0	C00
C10	Pipeline, major category used alone	0	0	C10
C20	Power transmission line, major category used alone	0	0	C20
C30	Other ground transportation that is not a pipeline or a power trans	0	0	C30
C31	Aerial tramway, monorail, or ski lift	0	0	C31
D00	Landmark, major and minor categories unknown	0	0	D00
D10	Military installation or reservation	0	0	D10
D20	Multihousehold or transient quarters	0	0	D20
D21	Apartment building or complex	0	0	D21
D22	Rooming or boarding house	0	0	D22
D23	Trailer court or mobile home park	0	0	D23
D24	Marina	0	0	D24
D25	Crew of vessel	0	0	D25
D26	Housing facility for workers	0	0	D26
D27	Hotel, motel, resort, spa, YMCA, or YWCA	0	0	D27
D28	Campground	0	0	D28
D29	Shelter or mission	0	0	D29
D30	Custodial facility	0	0	D30
D31	Hospital	0	0	D31

D32	Halfway house	0	0	D32
D33	Nursing home, retirement home, or home for the aged	0	0	D33
D34	County home or poor farm	0	0	D34
D35	Orphanage	0	0	D35
D36	Jail or detention center	0	0	D36
D37	Federal penitentiary, State prison, or prison farm	0	0	D37
D40	Educational or religious institution	0	0	D40
D41	Sorority or fraternity	0	0	D41
D42	Convent or monastery	0	0	D42
D43	Educational institution, including academy, school, college, and ur	0	0	D43
D44	Religious institution, including church, synagogue, seminary, temp	0	0	D44
D50	Transportation terminal	0	0	D50
D51	Airport or airfield	0	0	D51
D52	Train station	0	0	D52
D53	Bus terminal	0	0	D53
D54	Marine terminal	0	0	D54
D55	Seaplane anchorage	0	0	D55
D60	Employment center	0	0	D60
D61	Shopping center or major retail center	0	0	D61
D62	Industrial building or industrial park	0	0	D62
D63	Office building or office park	0	0	D63
D64	Amusement center	0	0	D64
D65	Government center	0	0	D65
D66	Other employment center	0	0	D66
D70	Tower	0	0	D70
D71	Lookout tower	0	0	D71
D80	Open space	0	0	D80
D81	Golf course	0	0	D81
D82	Cemetery	0	0	D82
D83	National park or forest	0	0	D83
D84	Other Federal land	0	0	D84
D85	State or local park or forest	0	0	D85
D90	Special purpose landmark	0	0	D90
D91	Post office box ZIP Code(R)	0	0	D91
E00	Physical feature, tangible but not transportation or hydrographic.	0	0	E00
E10	Fence line locating a visible and permanent fence between separa	0	0	E10
E20	Topographic feature	0	0	E20
E21	Ridge line, the line of highest elevation of a linear mountain	0	0	E21
E22	Mountain peak, the point of highest elevation of a mountain	0	0	E22
F00	Nonvisible boundary, major and minor categories unknown	0	0	F00
F10	Nonvisible jurisdictional boundary of a legal or administrative entit	0	0	F10
F11	Offset boundary of a legal or administrative entity	0	0	F11
F12	Corridor boundary of a legal or administrative entity	0	0	F12
F13	Interpolated boundary of a legal or administrative entity used for c	0	0	F13
F14	Superseded legal or administrative boundary	0	0	F14
F15	Superseded legal or administrative boundary, corrected through p	0	0	F15
F20	Nonvisible feature for data base topology	0	0	F20
F21	Automated feature extension to lengthen existing physical feature	0	0	F21
F22	Irregular feature extension, determined manually, to lengthen exis	0	0	F22

F23	Closure extension to complete data base topological closure betwee	0	0	F23
F24	Nonvisible separation line used with offset and corridor boundaries	0	0	F24
F25	Nonvisible centerline of area enclosed by corridor boundary	0	0	F25
F30	Point-to-point line, follows a line of sight and should not cross any vi	0	0	F30
F40	Property line, nonvisible boundary of either public or private lands, e	0	0	F40
F50	ZIP Code(R)boundary, reserved for future use in delineating ZIP Co	0	0	F50
F60	Map edge, now removed, used during data base creation	0	0	F60
F70	Statistical boundary	0	0	F70
F71	1980 statistical boundary	0	0	F71
F72	1990 statistical boundary, used to hold collection and tabulation cen	0	0	F72
F73	1990 statistical boundary and extent of land use, it is not classifiable	0	0	F73
F74	1990 statistical boundary, used to hold a tabulation census block bo	0	0	F74
F80	Nonvisible other tabulation boundary, major category used when the	0	0	F80
F81	School district tabulation boundary	0	0	F81
F82	Special census tabulation boundary	0	0	F82
H00	Water feature	0	0	H00
H01	Shoreline of perennial water feature	0	0	H01
H02	Shoreline of intermittent water feature	0	0	H02
H10	Stream	0	0	H10
H11	Perennial stream or river	0	0	H11
H12	Intermittent stream, river, or wash	0	0	H12
H13	Braided stream or river	0	0	H13
H20	Canal, ditch, or aqueduct	0	0	H20
H21	Perennial canal, ditch, or aqueduct	0	0	H21
H22	Intermittent canal, ditch, or aqueduct	0	0	H22
H30	Lake or pond	0	0	H30
H31	Perennial lake or pond	0	0	H31
H32	Intermittent lake or pond	0	0	H32
H40	Reservoir	0	0	H40
H41	Perennial reservoir	0	0	H41
H42	Intermittent reservoir	0	0	H42
H50	Bay, estuary, gulf, sound, sea, or ocean	0	0	H50
H51	Bay, estuary, gulf, or sound	0	0	H51
H53	Sea or ocean	0	0	H53
H60	Gravel pit or quarry filled with water	0	0	H60
H70	Nonvisible water area definition boundary	0	0	H70
H71	USGS closure line, used as maritime shoreline	0	0	H71
H72	Census water center line, computed to use as median positional bou	0	0	H72
H73	Census water boundary, international in waterways or at 12-mile limi	0	0	H73
H74	Census water boundary, separates inland from coastal or Great Lak	0	0	H74
H75	Census water boundary, separates coastal from territorial at 3-mile l	0	0	H75
H80	Special water feature, major category used when the minor category	0	0	H80
H81	Glacier	0	0	H81
X00	Feature not yet classified	0	0	X00

Appendix G
OC Transpo Operating Statistics

As of December 31, 2000

Urban Transit Area	368 sq. km
Population served	695,000
Passengers	80.1 million
Average weekday ridership	325,000

Service Facts

Passenger km carried	689.5 million
Service km operated	50.2 million
(includes charters and extras, excludes special event)	

Vehicle service hours	2.1 million
Average passenger trip length in km	10
Total km of routes	5,335
Number of routes (includes school routes)	204
Employees	2,127
Bus Stops	5,500
Bus stops with map and schedule display boxes	2,970
Bus stops with shelters	1,064
Bus stops with benches	1,050
Telidon schedule display units	25
Telephone Centres calls (annual)	655,000

"560" schedule information calls (annual)	10,000,000
Web site visits (annual)	1,020,000

Transitway

Weekday passenger volume	200,000
Peak hour one-way passenger volume	10,000
Buses per peak hour, through central area	190
Number of stations	34
Bike & Ride locations	20

Appendix H
Schedule Tables

List of Tables

Table H.1: Weekday Schedule Sample

ROUTE_NO	DIR	RUN_NO	BUS_STOP	ON_STREET	AT_STREET	TIME
3	SB	301	AF920	HURDMAN	1C	625
3	SB	301	AF920	HURDMAN	1C	923
3	SB	301	AF920	HURDMAN	1C	1204
3	SB	301	AF920	HURDMAN	1C	1440
3	NB	301	AF920	HURDMAN	1C	607
3	NB	301	AF920	HURDMAN	1C	902
3	NB	301	AF920	HURDMAN	1C	1150
3	NB	301	AF920	HURDMAN	1C	1430
3	NB	301	AF920	HURDMAN	1C	1738
3	SB	302	AF920	HURDMAN	1C	645
3	SB	302	AF920	HURDMAN	1C	943
3	SB	302	AF920	HURDMAN	1C	1224
3	SB	302	AF920	HURDMAN	1C	1500
3	SB	302	AF920	HURDMAN	1C	1809
3	SB	302	AF920	HURDMAN	1C	2040
3	SB	302	AF920	HURDMAN	1C	2310
3	NB	302	AF920	HURDMAN	1C	636
3	NB	302	AF920	HURDMAN	1C	917
3	NB	302	AF920	HURDMAN	1C	1210
3	NB	302	AF920	HURDMAN	1C	1450
3	NB	302	AF920	HURDMAN	1C	1756
3	NB	302	AF920	HURDMAN	1C	2023
3	NB	302	AF920	HURDMAN	1C	2250
3	SB	303	AF920	HURDMAN	1C	702
3	SB	303	AF920	HURDMAN	1C	1003
3	SB	303	AF920	HURDMAN	1C	1244
3	SB	303	AF920	HURDMAN	1C	1520
3	SB	303	AF920	HURDMAN	1C	1840
3	SB	303	AF920	HURDMAN	1C	2110
3	NB	303	AF920	HURDMAN	1C	654
3	NB	303	AF920	HURDMAN	1C	935
3	NB	303	AF920	HURDMAN	1C	1231
3	NB	303	AF920	HURDMAN	1C	1510
3	NB	303	AF920	HURDMAN	1C	1816
3	NB	303	AF920	HURDMAN	1C	2053
3	NB	303	AF920	HURDMAN	1C	2320
3	SB	304	AF920	HURDMAN	1C	742
3	SB	304	AF920	HURDMAN	1C	1023
3	SB	304	AF920	HURDMAN	1C	1304
3	SB	304	AF920	HURDMAN	1C	1600
3	NB	304	AF920	HURDMAN	1C	715
3	NB	304	AF920	HURDMAN	1C	1009
3	NB	304	AF920	HURDMAN	1C	1251
3	NB	304	AF920	HURDMAN	1C	1535

3	NB	304	AF920	HURDMAN	1C	1834
3	SB	305	AF920	HURDMAN	1C	802
3	SB	305	AF920	HURDMAN	1C	1043
3	SB	305	AF920	HURDMAN	1C	1324
3	SB	305	AF920	HURDMAN	1C	1620
3	SB	305	AF920	HURDMAN	1C	1911
3	SB	305	AF920	HURDMAN	1C	2140
3	NB	305	AF920	HURDMAN	1C	737
3	NB	305	AF920	HURDMAN	1C	1029
3	NB	305	AF920	HURDMAN	1C	1311
3	NB	305	AF920	HURDMAN	1C	1557
3	NB	305	AF920	HURDMAN	1C	1858
3	NB	305	AF920	HURDMAN	1C	2122
3	SB	306	AF920	HURDMAN	1C	822
3	SB	306	AF920	HURDMAN	1C	1103
3	SB	306	AF920	HURDMAN	1C	1344
3	SB	306	AF920	HURDMAN	1C	1640
3	SB	306	AF920	HURDMAN	1C	1940
3	SB	306	AF920	HURDMAN	1C	2210
3	NB	306	AF920	HURDMAN	1C	800
3	NB	306	AF920	HURDMAN	1C	1049
3	NB	306	AF920	HURDMAN	1C	1331
3	NB	306	AF920	HURDMAN	1C	1617
3	NB	306	AF920	HURDMAN	1C	1926
3	NB	306	AF920	HURDMAN	1C	2152
3	SB	307	AF920	HURDMAN	1C	842
3	SB	307	AF920	HURDMAN	1C	1123
3	SB	307	AF920	HURDMAN	1C	1404
3	SB	307	AF920	HURDMAN	1C	1720
3	SB	307	AF920	HURDMAN	1C	2010
3	NB	307	AF920	HURDMAN	1C	820
3	NB	307	AF920	HURDMAN	1C	1109
3	NB	307	AF920	HURDMAN	1C	1350
3	NB	307	AF920	HURDMAN	1C	1658
3	NB	307	AF920	HURDMAN	1C	1953
3	NB	307	AF920	HURDMAN	1C	2221
3	SB	308	AF920	HURDMAN	1C	603
3	SB	308	AF920	HURDMAN	1C	903
3	SB	308	AF920	HURDMAN	1C	1143
3	SB	308	AF920	HURDMAN	1C	1421
3	SB	308	AF920	HURDMAN	1C	1740
3	NB	308	AF920	HURDMAN	1C	841
3	NB	308	AF920	HURDMAN	1C	1129
3	NB	308	AF920	HURDMAN	1C	1410
3	NB	308	AF920	HURDMAN	1C	1718
3	SB	350	AF920	HURDMAN	1C	722
3	NB	350	AF920	HURDMAN	1C	948
3	SB	E-483	AF920	HURDMAN	1C	1700
3	NB	E-491	AF920	HURDMAN	1C	1637

Table H.2: Saturday Schedule Sample

ROUTE_NO	DIR	RUN_NO	BUS_STOP	ON_STREET	AT_STREET	TIME
3	SB	301	AF920	HURDMAN	1C	640
3	SB	302	AF920	HURDMAN	1C	740
3	SB	303	AF920	HURDMAN	1C	810
3	SB	304	AF920	HURDMAN	1C	840
3	SB	301	AF920	HURDMAN	1C	910
3	SB	305	AF920	HURDMAN	1C	940
3	SB	302	AF920	HURDMAN	1C	1010
3	SB	303	AF920	HURDMAN	1C	1040
3	SB	304	AF920	HURDMAN	1C	1110
3	SB	301	AF920	HURDMAN	1C	1140
3	SB	305	AF920	HURDMAN	1C	1210
3	SB	302	AF920	HURDMAN	1C	1240
3	SB	303	AF920	HURDMAN	1C	1310
3	SB	304	AF920	HURDMAN	1C	1340
3	SB	301	AF920	HURDMAN	1C	1410
3	SB	305	AF920	HURDMAN	1C	1440
3	SB	302	AF920	HURDMAN	1C	1510
3	SB	303	AF920	HURDMAN	1C	1540
3	SB	304	AF920	HURDMAN	1C	1610
3	SB	301	AF920	HURDMAN	1C	1640
3	SB	305	AF920	HURDMAN	1C	1710
3	SB	302	AF920	HURDMAN	1C	1740
3	SB	303	AF920	HURDMAN	1C	1810
3	SB	304	AF920	HURDMAN	1C	1840
3	SB	301	AF920	HURDMAN	1C	1910
3	SB	305	AF920	HURDMAN	1C	1940
3	SB	404	AF920	HURDMAN	1C	2010
3	SB	403	AF920	HURDMAN	1C	2040
3	SB	401	AF920	HURDMAN	1C	2110
3	SB	17501	AF920	HURDMAN	1C	2140
3	SB	303	AF920	HURDMAN	1C	2210
3	SB	304	AF920	HURDMAN	1C	2240
3	SB	301	AF920	HURDMAN	1C	2310
3	SB	305	AF920	HURDMAN	1C	2340
3	NB	302	AF920	HURDMAN	1C	714
3	NB	303	AF920	HURDMAN	1C	744
3	NB	304	AF920	HURDMAN	1C	814
3	NB	301	AF920	HURDMAN	1C	844
3	NB	305	AF920	HURDMAN	1C	917
3	NB	302	AF920	HURDMAN	1C	947
3	NB	303	AF920	HURDMAN	1C	1018
3	NB	304	AF920	HURDMAN	1C	1048
3	NB	301	AF920	HURDMAN	1C	1123
3	NB	305	AF920	HURDMAN	1C	1153

3	NB	302	AF920	HURDMAN	1C	1223
3	NB	303	AF920	HURDMAN	1C	1253
3	NB	304	AF920	HURDMAN	1C	1323
3	NB	301	AF920	HURDMAN	1C	1353
3	NB	305	AF920	HURDMAN	1C	1425
3	NB	302	AF920	HURDMAN	1C	1455
3	NB	303	AF920	HURDMAN	1C	1525
3	NB	304	AF920	HURDMAN	1C	1555
3	NB	301	AF920	HURDMAN	1C	1625
3	NB	305	AF920	HURDMAN	1C	1655
3	NB	302	AF920	HURDMAN	1C	1725
3	NB	303	AF920	HURDMAN	1C	1755
3	NB	304	AF920	HURDMAN	1C	1825
3	NB	301	AF920	HURDMAN	1C	1855
3	NB	305	AF920	HURDMAN	1C	1919
3	NB	302	AF920	HURDMAN	1C	1949
3	NB	303	AF920	HURDMAN	1C	2009
3	NB	304	AF920	HURDMAN	1C	2039
3	NB	301	AF920	HURDMAN	1C	2109
3	NB	305	AF920	HURDMAN	1C	2139
3	NB	404	AF920	HURDMAN	1C	2205
3	NB	403	AF920	HURDMAN	1C	2235
3	NB	401	AF920	HURDMAN	1C	2305
3	NB	303	AF920	HURDMAN	1C	005
3	SB	301	AA170	CRICHTON	BEECHWOOD NS	650
3	SB	302	AA170	CRICHTON	BEECHWOOD NS	750
3	SB	303	AA170	CRICHTON	BEECHWOOD NS	820
3	SB	304	AA170	CRICHTON	BEECHWOOD NS	850
3	SB	301	AA170	CRICHTON	BEECHWOOD NS	920
3	SB	305	AA170	CRICHTON	BEECHWOOD NS	950
3	SB	302	AA170	CRICHTON	BEECHWOOD NS	1020
3	SB	303	AA170	CRICHTON	BEECHWOOD NS	1050
3	SB	304	AA170	CRICHTON	BEECHWOOD NS	1122
3	SB	301	AA170	CRICHTON	BEECHWOOD NS	1152
3	SB	305	AA170	CRICHTON	BEECHWOOD NS	1222
3	SB	302	AA170	CRICHTON	BEECHWOOD NS	1252
3	SB	303	AA170	CRICHTON	BEECHWOOD NS	1322
3	SB	304	AA170	CRICHTON	BEECHWOOD NS	1352
3	SB	301	AA170	CRICHTON	BEECHWOOD NS	1422
3	SB	305	AA170	CRICHTON	BEECHWOOD NS	1452
3	SB	302	AA170	CRICHTON	BEECHWOOD NS	1522
3	SB	303	AA170	CRICHTON	BEECHWOOD NS	1552
3	SB	304	AA170	CRICHTON	BEECHWOOD NS	1622
3	SB	301	AA170	CRICHTON	BEECHWOOD NS	1652
3	SB	305	AA170	CRICHTON	BEECHWOOD NS	1722
3	SB	302	AA170	CRICHTON	BEECHWOOD NS	1752
3	SB	303	AA170	CRICHTON	BEECHWOOD NS	1821
3	SB	304	AA170	CRICHTON	BEECHWOOD NS	1851
3	SB	301	AA170	CRICHTON	BEECHWOOD NS	1919

Table H.3: Sunday Schedule Sample

ROUTE_NO	DIR	RUN_NO	BUS_STOP	ON_STREET	AT_STREET	TIME
3	SB	308	AF920	HURDMAN	1C	823
3	SB	302	AF920	HURDMAN	1C	853
3	SB	307	AF920	HURDMAN	1C	923
3	SB	304	AF920	HURDMAN	1C	953
3	SB	305	AF920	HURDMAN	1C	1023
3	SB	306	AF920	HURDMAN	1C	1053
3	SB	301	AF920	HURDMAN	1C	1123
3	SB	303	AF920	HURDMAN	1C	1153
3	SB	308	AF920	HURDMAN	1C	1223
3	SB	302	AF920	HURDMAN	1C	1253
3	SB	307	AF920	HURDMAN	1C	1323
3	SB	304	AF920	HURDMAN	1C	1353
3	SB	305	AF920	HURDMAN	1C	1423
3	SB	306	AF920	HURDMAN	1C	1453
3	SB	301	AF920	HURDMAN	1C	1523
3	SB	303	AF920	HURDMAN	1C	1553
3	SB	308	AF920	HURDMAN	1C	1623
3	SB	302	AF920	HURDMAN	1C	1653
3	SB	307	AF920	HURDMAN	1C	1723
3	SB	304	AF920	HURDMAN	1C	1753
3	SB	305	AF920	HURDMAN	1C	1823
3	SB	306	AF920	HURDMAN	1C	1853
3	SB	301	AF920	HURDMAN	1C	1923
3	SB	303	AF920	HURDMAN	1C	1953
3	SB	308	AF920	HURDMAN	1C	2023
3	SB	307	AF920	HURDMAN	1C	2123
3	SB	305	AF920	HURDMAN	1C	2223
3	NB	302	AF920	HURDMAN	1C	841
3	NB	303	AF920	HURDMAN	1C	952
3	NB	308	AF920	HURDMAN	1C	1022
3	NB	302	AF920	HURDMAN	1C	1052
3	NB	307	AF920	HURDMAN	1C	1122
3	NB	304	AF920	HURDMAN	1C	1152
3	NB	305	AF920	HURDMAN	1C	1222
3	NB	306	AF920	HURDMAN	1C	1255
3	NB	301	AF920	HURDMAN	1C	1325
3	NB	303	AF920	HURDMAN	1C	1355
3	NB	308	AF920	HURDMAN	1C	1425
3	NB	302	AF920	HURDMAN	1C	1455
3	NB	307	AF920	HURDMAN	1C	1525
3	NB	304	AF920	HURDMAN	1C	1555
3	NB	305	AF920	HURDMAN	1C	1625
3	NB	306	AF920	HURDMAN	1C	1655
3	NB	301	AF920	HURDMAN	1C	1725

3	NB	303	AF920	HURDMAN	1C	1755
3	NB	308	AF920	HURDMAN	1C	1825
3	NB	302	AF920	HURDMAN	1C	1853
3	NB	307	AF920	HURDMAN	1C	1923
3	NB	304	AF920	HURDMAN	1C	1953
3	NB	305	AF920	HURDMAN	1C	2023
3	NB	306	AF920	HURDMAN	1C	2053
3	NB	301	AF920	HURDMAN	1C	2123
3	NB	303	AF920	HURDMAN	1C	2153
3	NB	308	AF920	HURDMAN	1C	2223
3	NB	307	AF920	HURDMAN	1C	2323
86	EB	8601	AF920	HURDMAN	1C	733
86	EB	8603	AF920	HURDMAN	1C	833
86	EB	8602	AF920	HURDMAN	1C	903
86	EB	8605	AF920	HURDMAN	1C	933
86	EB	8601	AF920	HURDMAN	1C	1003
86	EB	8604	AF920	HURDMAN	1C	1033
86	EB	8603	AF920	HURDMAN	1C	1105
86	EB	8602	AF920	HURDMAN	1C	1135
86	EB	8605	AF920	HURDMAN	1C	1205
86	EB	8601	AF920	HURDMAN	1C	1235
86	EB	8604	AF920	HURDMAN	1C	1305
86	EB	8603	AF920	HURDMAN	1C	1335
86	EB	8602	AF920	HURDMAN	1C	1405
86	EB	8605	AF920	HURDMAN	1C	1435
86	EB	8601	AF920	HURDMAN	1C	1505
86	EB	8604	AF920	HURDMAN	1C	1535
86	EB	8603	AF920	HURDMAN	1C	1605
86	EB	8602	AF920	HURDMAN	1C	1635
86	EB	8605	AF920	HURDMAN	1C	1705
86	EB	8601	AF920	HURDMAN	1C	1735
86	EB	8604	AF920	HURDMAN	1C	1805
86	EB	8603	AF920	HURDMAN	1C	1835
86	EB	8602	AF920	HURDMAN	1C	1905
86	EB	8605	AF920	HURDMAN	1C	1935
86	EB	8601	AF920	HURDMAN	1C	2002
86	EB	8604	AF920	HURDMAN	1C	2032
86	EB	8603	AF920	HURDMAN	1C	2102
86	EB	8602	AF920	HURDMAN	1C	2132
86	EB	8605	AF920	HURDMAN	1C	2202
86	EB	8601	AF920	HURDMAN	1C	2232
86	EB	1804	AF920	HURDMAN	1C	2332
86	WB	8602	AF920	HURDMAN	1C	712
86	WB	8601	AF920	HURDMAN	1C	812
86	WB	8603	AF920	HURDMAN	1C	912
86	WB	8602	AF920	HURDMAN	1C	942
86	WB	8605	AF920	HURDMAN	1C	1012