IDEA Final Report

Contract Number: TRANSIT-10

Prepared for
the IDEA Program
Transportation Research Board
National Research Council

Manuel D. Rossetti, PI
Department of Systems Engineering
University of Virginia
Thornton Hall
Charlottesville, VA 22903

Submittal Date:

June 24, 1996

ACKNOWLEDGEMENTS

I would like to thank the University of Virginia's Transportation Services Department for their help during this project, especially Evan Kittredge, for his help in providing access to buses and routes. In addition, I would like to note the contributions of my 1996 Senior Design Team (Joshua Motter, Roger Weisenberg, Leena Chitre, Tom Carney, Marjorie Boursiquot, and Rob Ahlert) to the project. Their enthusiasm was exceptional. Finally, I would like to thank Tim Turitto for all his valuable effort and insights.

TABLE OF CONTENTS

Executive Summary

The main goal of this investigation was to demonstrate the feasibility of an automatic transit monitoring system for improved transit system operation based on radio frequency identification (RF/ID) of transit users. Current transit data collection systems emphasize either passenger counting or vehicle location information. The prototype developed in this project represents the integration of automatic passenger counting with automatic vehicle location to allow the automatic collection of transit system performance. The system involves the incorporation of RF/ID tags into bus passes for the purpose of improving the methods of data collection regarding transit users. Passengers carrying the tags can then be uniquely identified. In addition, RF/ID tags are placed at the bus stops to track the movement of busses along the route. This allows data such as fare information, passenger boarding and alighting event times, and transit stop identification to be captured in near real time. The large amounts of captured data are organized within a database and can be translated into the information necessary for improved decision making. Transit operators can utilize the information to improve route planning, scheduling, fare collection, capacity, and demand analysis.

The investigation represents a collaborative effort between the Transportation Research Board, the University of Virginia's Department of Systems Engineering, Civil Engineering, University Transit System, and Advanced Systems Group International (ASGI) a federally designated small business corporation. The investigation was divided into three stages covering the prototype system development life-cycle. During Stage 1, we performed a technology assessment to ascertain prototype system design criteria, a preliminary database design, and a review of literature for applicable algorithms or other required system functionality. During Stage 2, the system hardware and software specifications were completed, the database design for the prototype was finalized, and the initial software development completed. During Stage 3, the prototype was observed and tested under simulated and representative field conditions.

The results of this project include:

1. A prototype RF/ID based automatic data collection system for transit operations which has the potential to compete with current automated passenger counting and automatic vehicle location systems by integrating their functions.

2. Survey results indicating that 45% of those sampled did not believe that privacy was an important issue involving this system. In addition, more than half of those surveyed did not have a strong opinion supporting or opposing the system. Passenger acceptance increased for those passengers concerned about fare collection. Although preliminary in scope, the survey results suggest that passengers will accept the technology especially if the added convenience is explained and privacy is ensured.

3. A RF/ID based automatic database collection system for monitoring transit system performance and supporting route control analysis.

1. IDEA PRODUCT

The primary goal of this project was to produce a prototype automatic data collection system based on RF/ID tagging of transit users in order to monitor transit user movements and the operating performance of the transit system. This prototype system represents a unique combination of hardware and software which when integrated form the basis for an automatic transit monitoring system. Existing radio frequency identification (RF/ID) hardware was modified, adapted, and tested under conditions representative of transit practice for the purpose of collecting information on the movements of transit users and vehicles within the transit system. The software developed during this project enables the use of RF/ID hardware within the transportation community by specifically addressing data collection needs. In addition, algorithms and databases for the translation, storage, and retrieval of data collected from transit users were developed and tested in conjunction with the hardware. By adopting and integrating RF/ID technology into their computerized transit management systems, transportation agencies can achieve significant results in the following areas:

- real-time status information for transit users
- fare collection
- transit user convenience and marketing
- data collection regarding transit users

The experience gained with this prototype system will assist transit authorities in assessing automatic data collection of transit users based upon RF/ID.

2. CONCEPT AND INNOVATION

We address the need for innovation in the area of Transit Management Practices with an IDEA project for improved methods of data collection about or from transit users based on radio frequency identification (RF/ID) of transit users via RF/ID transit tags. Figure 1 illustrates the concept. Radio frequency receivers emitting an energy field activate the transponder within the card, receive the data stored on the card, and then transmit the data via RF links to a central data collection system. Figure 2 presents a block diagram of the hardware/software configuration. Passengers carrying the cards can then be uniquely identified. The data captured could include fare information, event times (boarding and alighting times) of the transit user, and transit stop identification. From these data, information such as origin destination pairs and transit user usage times can be derived for use in planning and control. The system includes the methods and means for collecting and organizing the data, and the data reduction techniques necessary to translate the data into usable information.

The unique innovative aspect of this investigation is the use of RF/ID to track the movements of transit users and the emphasis of the use of the data for improved decision making. The RF/ID sensors and tags modified and used in this project allow for passive collection of data and allow multiple tags to be read simultaneously within a single interrogation field. Thus, transit users can simply carry the cards and be identified by the system. Automated fare card systems are in pilot tests across the country, see reference (*1*); however, fare collection is *not* the emphasis of this project. Instead, we concentrate on how to design and operate the RF/ID automatic data collection system for the purpose of collecting transit system performance. The other component of a transit system which requires tracking is the vehicle. Advanced automated vehicle location(AVL) systems continue to be integrated into transit management systems as discussed in Labell (*1*). Our system includes a straight forward way to track the movements of vehicles by placing RF/ID tags in the roadway. With an additional antenna on the vehicle to read the embedded roadway tags, we then automatically record the location of the vehicle at its last transmission. Thus, this prototype system represents the integration of automatic vehicle location (AVL) systems and automatic passenger counting (APC) systems into a more robust and potentially beneficial integrated automatic transit monitoring system. Levinson (*2*) identified the merger of APC and AVL systems as an important goal in allowing for improved real time planning and control of transit systems.

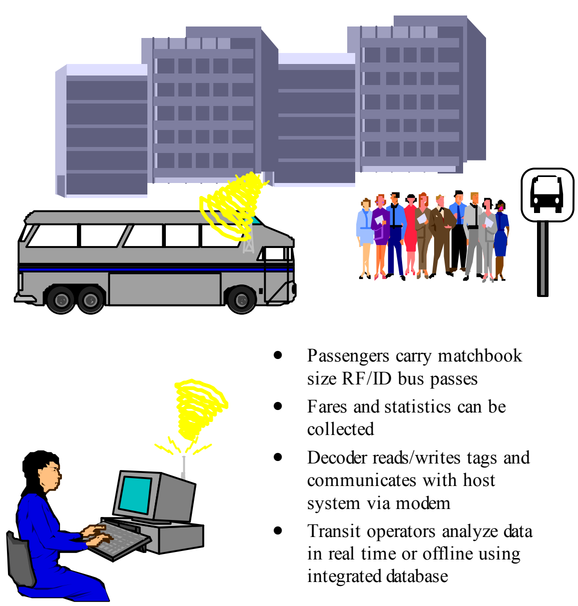

FIGURE 1 System concept

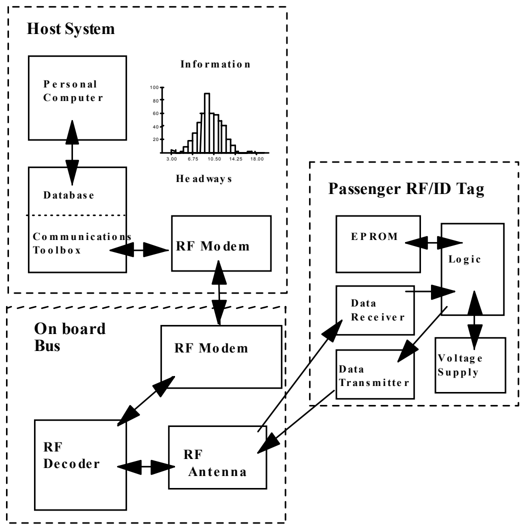

FIGURE 2 Hardware/Software interaction

3. INVESTIGATION

The main goal of this investigation was:

> To demonstrate the feasibility of implementing an automatic data collection information system for improved transit system operation based on RF/ID of transit users via RF/ID tags by developing and analyzing a prototype system.

In order to demonstrate the feasibility of using RF/ID technology in transit practice, we concentrated on the following sub-objectives:

1. To assess technological barriers and criteria in implementing RF/ID.
2. To design and evaluate a preliminary system for use at a transit property.
3. To implement and test a prototype system under field conditions.

The investigation represents a collaborative effort between the Transportation Research Board, the University of Virginia's Department of Systems Engineering, Civil Engineering, University Transit System, and Advanced Systems Group International (ASGI). ASGI is a federally designated small business corporation located in Sterling, Virginia. The company's mission is to become a leader in radio frequency technology by delivering integrated software and hardware solutions. ASGI has previous experience implementing RF/ID systems for inventory and asset tracking, security monitoring applications (e.g. parking garages, personnel access control) and employee time and attendance systems.

A prototype is a representation of a system which models the aspects of the system in terms of functionality except that it has not be fully qualified for operational use. Thus, it is important to understand that a prototype is necessarily a limited version of the final product. The development of a prototype enables design decisions and user functionality to be incorporated during the early stages of the system development life cycle and will serve as the basis for an eventual finalized product. Since a prototype is a representation of the system, the project was divided into three stages covering the prototype's system development life-cycle. During Stage 1, we performed a technology assessment to ascertain prototype system design criteria, a preliminary database design, and a review of literature for applicable algorithms or other needed system functionality. During Stage 2, the system hardware and software specifications were completed and work on the prototype began. In addition, the database design for the prototype was finalized and the software development completed. During Stage 3, the prototype was observed and tested under simulated and representative field conditions. A description of the tasks and times associated with the project stages is given below:

Stage 1: Technology Assessment

- Personnel: graduate student, principal investigator, ASGI engineers
- Duration: 2 months
- Description: Assess RF/ID design for applicability to transit systems. Assess and select representative bus route. Literature review on algorithms for data translation. Preliminary database design. Database software review and selection. Interview transit operators, define database functionality. Establish preliminary design configurations. The results should produce preliminary data on the potential utilization of RF/ID technology in transit systems.
- UVA Deliverables: TRB Progress report due 8/31/1995, TRB Stage 1 report due 9/30/1995, Preliminary database design.
- ASGI Deliverables: Conduct initial site survey, in conjunction with UVA, to determine detailed project requirements, develop system design requirements, determine software development requirements.

Stage 2: Prototype Development

- Personnel: graduate student, principal investigator, ASGI engineers
- Duration: 4 months
- Description: Complete system hardware and software specifications, design and build a prototype RF/ID system for collecting, storing, and translating

transit data. Develop database user interface procedures, develop data extraction algorithms, develop simulation program for populating and testing database functionality. Run simulation experiments, software debugging and validation, populate and test database functionality with simulated data. The results should produce a field ready system for performing trial operational testing with the University of Virginia campus transit system.
- UVA Deliverables: TRB First quarter report due 11/30/1995, TRB Stage 2 report due 2/28/1996, Prototype system database computer software program.
- ASGI Deliverables: Install interrogators/decoders and antennas on bus, install embedded controller units and interface high speed modems, install communication links between the bus rider interrogation locations and the central data collection point, provide support in developing data collection system.

Stage 3: Prototype Testing and Evaluation

- Personnel: graduate student, principal investigator, ASGI engineers
- Duration: 3 months
- Description: Perform trial testing of RF/ID system in the UVA transit system as ascertain system performance and operational viability. Automatic data collection of selected users, verification/validation of data collection through direct observations, transit user follow up survey. Examine potential coupling of RF/ID data collection with AVL systems.
- UVA Deliverables: TRB draft final report due 3/30/1996, TRB final report due 4/30/1996 Summary data from operational testing, summary data from transit user survey.
- ASGI Deliverables: Test interrogators/decoders and antennas on bus, test embedded controller units and interface high speed modems, test communication links between the bus rider interrogation locations and the central data collection point.

The rest of Section 3 discusses the need for automatic data collection in transit practice, presents the prototype design including hardware, software and implementation issues, and then gives any results/conclusions with respect to our goals and/or objectives.

3.1 SYSTEM NEED

The Transportation Cooperative Research Program project announcement for TRANSIT-IDEA(3) identified a set of innovations needed for transit systems. This project specifically addresses the following four needs:

1. "Advanced systems for transit vehicle location, identification, and management of transit systems."

2. "Improved methods of data collection about or from transit users."

3. "Automated and intelligent communication and information for route planning and for providing real-time status information for transit users."

4. "Advanced systems for fare collection and control systems."

Items 1 and 2 are specifically addressed by the development of the prototype. Items (3) and (4) are potential benefits associated with our system. An illustrative discussion highlighting item (4) will be given in Section 3.7 of this report. As with many automatic data collection(ADC) systems, the drive for automation is at first primarily concerned with the savings in time and cost associated with replacing manual data collection methods. Unfortunately, the quest for automation too often leads to data overload and then to data under utilization. The ADC system collects massive amounts of data which go unused because the system was not designed to provide information from the data.

Our initial definition of need therefore depends on the end user's information requirements. The information requirements for improved decision making that the prototype should at least meet were obtained from a study sponsored by the Transportation Research Board on bus route evaluation standards, see reference (4). This study cited five categories for bus route evaluation standards. The criteria given in Table 1 under two of the five categories were identified as pertinent to the project

TABLE 1 Key Bus Evaluation Criteria

Schedule Design Standards	Productivity Standards
maximum number of standees maximum intervals peak periods vs. off-peak periods minimum intervals duration of standee time time spent waiting at a transfer point	passengers per hour passengers per mile passengers per trip passenger miles

Other information considered important included:

- number of transfer points per passenger per trip
- bus stop to bus stop transit times
- total transit use time
- origin/destination counts
- number waiting, number of passengers in transit
- demand patterns
- number of balking users

Reference (4) also indicated that most transit systems obtain data for these criteria manually, on a yearly basis, and at great cost. For improved decision making, data must be made available frequently and in a cost effective manner. Most of the above bus evaluation criteria can be directly inferred using our database. The following criteria can not be directly inferred:

- number of transfer points per passenger per trip
- time spent waiting at a transfer point
- number of balking users

While we can not directly infer the above mentioned criteria, this should not imply that an inference procedure for these criteria can not be developed. We consider such an effort to be outside the scope of this project.

The need for ADC in transit practice has been clearly established. The use of RF/ID enables the majority of the informational requirements to be met in a timely and cost effective manner by integrating APC and AVL systems. We will discuss the extraction and reporting of these and other representative criteria in the software description section of this report.

3.2 HARDWARE DESCRIPTION

As indicated in Figure 2, the hardware elements of the system consists of RF/ID tags, a RF decoder, RF antenna, communications equipment and a host computer. The basic hardware configuration calls for:

- (1) decoder mounted within the bus powered by 24V DC
- (2) RF modems, one attached to the serial port of the decoder on the bus, the other attached to serial port on the host computer system
- (4-6) tags per bus stop, 1 tag for each passenger
- (4) antennas per bus: 1 side read antenna for the front doorway, 2 side read antennae for the rear doorway, 1 antenna mounted underneath the bus
- a laptop computer serving as a remotely located host base station

This hardware configuration allows bus and passenger RF/ID transactions to be captured in order to meet the above mentioned informational requirements.

During the conceptual design phase, we eliminated the hardware configuration which called for a passenger reading antenna to be located at each bus stop. This alternative configuration would allow passenger waiting time at the stop to be estimated; however, we concluded that all waiting times at a bus stop were not necessary. For instance, a person may arrive at a bus stop twenty minutes before the arrival of the bus. This does not indicate a fault with the bus schedule, but merely shows the passenger's individual risk preference. Other major drawbacks in placing the passenger read antenna at the bus stop included:

- the number of antennas needed (namely one for each bus stop)
- powering the remote decoder
- the number of communication devices needed (one for each bus stop)
- the durability and maintenance of the antenna/decoder at the bus stop

This alternative configuration has the advantage of directly observing where the passenger boarded and alighted. Our final configuration solves this problem by embedding tags in the roadway at each stop and by including an antenna on the bus to read the embedded tags. Thus, we can associate a bus stop with passenger transactions and record the passenger's origin and destination.

Our final design configuration presents the following hardware technical challenges:

- Integrating four antennae with one decoder
- Mounting requirement for the antenna under the bus
- Ensuring a high read/write rate for the antenna under the bus
- Preventing extraneous read/writes to passenger tags already on the bus
- Activating and deactivating antenna fields during operations

This rest of this section discusses the hardware components of the system which includes the tags, decoders, antennas, and the RF modems.

3.2.1 Hardware Components

RF Tags

The RF tag has a dimensions of 38 x 33 x 10 mm (1.49” x 1.30” x 0.39”) and a weight of 20 grams (7 oz). It is powered by a internal lithium battery with a 5 year life expectancy depending on rate of usage. The condition of the battery is monitored during transmission. The tag has an environmental operating temperature of -20C to 80C. Insulation options are available which can extend the operating temperature range. The tag is capable of storing 115 alphanumeric characters and transmitting at approximately 1000 characters per second with parity and checksum error checking. The transmission range is 132 Khz for receive and 66 Khz for transmit.

RF Decoder

The purpose of the RF decoder is to interpret signals from the antenna and to transmit information to the tags through signals to the antenna. It converts the incoming analog radio signals from the tags to digital information for storage and transmission in ASCII format. In addition, the decoder is capable of translating digital information into analog radio signals to allow transmission of information to the tags. The decoder can be connected to a host system via a standard serial interface. The decoder consists of a central processing unit, a radio frequency unit, a communications interface board, and a power supply unit. The decoder is capable of identifying up to 8 tags simultaneously within the interrogation field.

TABLE 2 Decoder Specifications

Dimensions:	255 x 306 x 185 mm
Weight:	5.45 Kgs
Enclosure:	steel protected to IP65
Power Requirements:	240/110V AC or 24V DC
Interfaces:	RS232 (3 wire), RS422, RS485, Wiegand
Baud Rates:	up to 19,200
Data bits/parity	7 bit, odd, event, mark (no parity 2 stop bits) or space (8 bit, no parity)

RF Antenna

The RF antenna comes in different configurations and typically must be designed specifically for the application. The most common antenna is a wire encased in a decorative conduit which is designed and pre-tuned to properly generate the desired interrogation field. The antenna produces an interrogation field which will cause tags within the field to “wake up”, i.e. it turns the transponder on within the tag. Attached to the antenna is a tuning box which enables the antenna to be tuned for maximum range and to allow for adjustments. The standard configurations are 1) orientation free, 2) side read, 3) standard industrial antenna, and 4) road loop antenna. The orientation free antenna allows tags to be read regardless of their physical orientation within the interrogation field. The side read antenna is mounted vertically and gives a maximum range of 183 cm in proximity-type applications. The standard industrial antenna is oriented horizontally and is typically used as a tag encoding station. The road loop antenna is installed in slots cut into the pavement for reading tags on vehicles. Our design calls for three side read antenna to capture passenger transactions and a specially designed horizontal read antenna for reading tags embedded in the roadway.

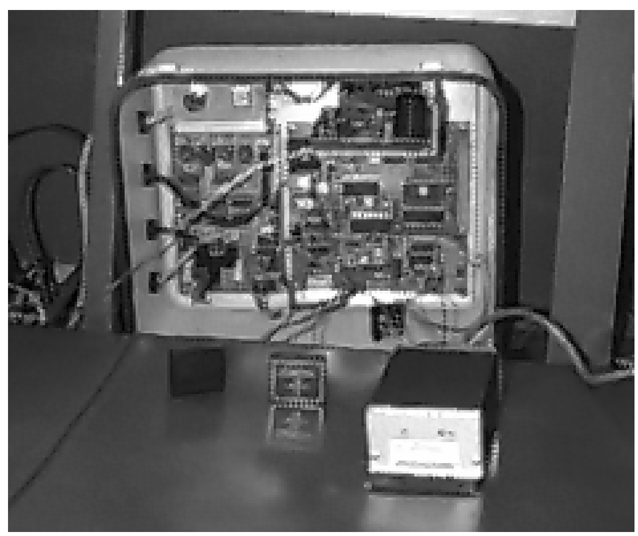

FIGURE 3 RF/ID tag, decoder and RTLU

FIGURE 4 Front RF antenna mounted to bus

FIGURE 5 Rear left RF antenna

FIGURE 6 Rear right RF antenna

RF Modem

The RF modem selected for the project was the Motorola RNET Remote LONWORKS Telemetry Unit (RLTU). The RLTU provides wireless communication and control functions via the LONTALK 7-layer communications protocol implemented in the NEURON Chip. LONWORKS technology was developed by Echelon Corporation and Motorola and is being used in process control, telecommunications, industrial and building automation, and other smart card systems. For more information concerning the capabilities of the RLTU, we refer the reader to reference (5).

Mounting

The bus model used in this application is 1981 GMC RTS with a front and rear door for passenger boarding and alighting. The front doorway width is 92 cm. The rear doorway width is 154 cm. Due to the rear doorway width, two side mounted antenna are necessary in order to properly ensure passenger data capture. The side read antenna are mounted on the stairwell panel where available. The rear door has only 1 stairwell panel which necessitates the fabrication of a substitute panel for antenna mounting purposes. Mounting the antenna on the stairwell panel reduces interference caused by metal contacting the antenna. The antenna is embedded in a adhesive tape which can be attached to the stairwell panel, see Figures 4-6. The antenna mounted underneath the bus is designed to capture tags embedding in the roadway. The layout of the antenna is shown in Figure 7.

3.2.2 Hardware Operation

The final design calls for the hardware to operate in the following manner. When the bus doors are closed, the roadway antenna is active in order to read tags embedded in the pavement for bus stop identification. When the bus doors open, the roadway antenna deactivates and the stairwell antenna activate. A relay switch polls each of the passenger reading antennae while the doors are open. The stairwell antenna and decoder captures boarding and alighting passengers. For boarding passengers, a 1 is written to the on/off field on the tag. For alighting passengers a 0 is written to the on/off field on the tag. The captured data is stored and forwarded to the base station computer using the RF modem connection. The RF modem capabilities allow near real-time data capture from any bus stop on the route. To ensure data accuracy the transactions are repeated three times within the transmission frame. An example transactional data file is given in Table 3. The data is sent via RF modem to the base station and appended to a file. The transaction file is then be processed for data validation and insertion into the database.

Valid transactions are essentially identified by the logical and temporal ordering of the data. Ideally, the data would contain a bus stop identifier followed by a door sensor input, and then all the transactions that occur at the bus stop, followed by a door sensor input, and then a bus stop identifier captured as the bus departs the stop. The next sequence of transactions would contain the next bus stop identifier followed by all the transactional data that occur at that bus stop, etc. The layout of the tags embedded in the roadway must ensure that a bus stop tag is read as the bus approaches and departs the stop. This would ensure that passenger transactions are essentially bounded by bus stop transactions. Whether or not the bus stop identifier is the first transaction depends upon whether or not the RF/ID tags associated with the bus stop are within the antenna interrogation field as the bus approaches a stop. The bus stop identifier may be read anywhere within the transactional data: at the beginning when the bus arrives at the stop, somewhere in the middle as RF/ID tag transmissions are recorded, or even at the end of the transactional data as the bus departs the bus stop. We also assume that the bus stop identifier will be read even if the bus does not stop at the bus stop. The open and close door event helps to logically bound the transactions.

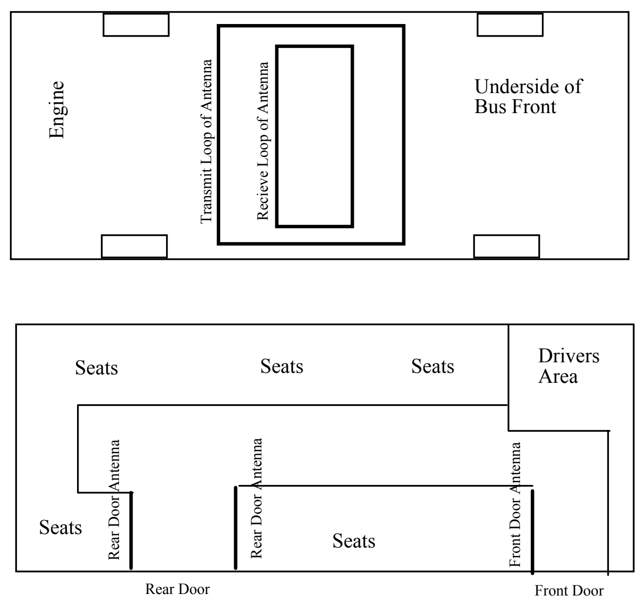

FIGURE 7 Layout of antennas on bus

The materials used in ASGI hardware and the low frequency signals used to transmit data pose no health risks to humans. ASGI equipment meets the guidelines and power density limits specified in ANSI C95.1-1532 and ANSI/UL 913-1988.

TABLE 3 Example transaction file

Data file contents	**Content description**
b001,08/21/95	bus id and date
b001,08/21/95	
b001,08/21/95	
s001,1,10:30:00	stop id, tag number, time
s001,1,10:30:00	
s001,1,10:30:00	
01	open door event id
01	
01	
p001,1,10:30:10	passenger id, on/off bit, time
p001,1,10:30:10	
p001,1,10:30:10	
p002,0,10:30:30	
p002,0,10:30:30	
p002,0,10:30:30	
00	close door event id
00	
00	
s001,2,10:30:35	stop id, tag number, time
s001,2,10:30:35	
s001,2,10:30:35	

3.3 SOFTWARE AND DATABASE DEVELOPMENT

This section describes the relational model for the software system, the data validation issues, an overview of the user interface and main functions, and the database reporting features. There are essentially three types of data models which are used in database implementations: relational, object-oriented, and network. We refer the reader to Date (6) for more information on these data models. We selected the relational model for this application because of its easy of modeling and industry acceptance. A relational database is interpreted by the users as a collection of tables. A table represents a entity which is being modeled in the system. The columns of the table represent attributes associated with an entity. Links or relationships between entities occur when entities have attribute values in common. Each entity is required to have a primary key, i.e. an attribute which allows it to be uniquely identified. If a primary key for an entity appears as an attribute for another entity, then it is called a foreign key. The basic steps of database modeling are: determining the entities to be modeled, indicating the relationships between entities, defining the primary keys for each entity, determining the attributes of each entity, and then normalizing the data model. Database modeling and development is facilitated by the use of an entity relationship diagram (ERD). This involves representing the relationships pictorially. In an ERD, an entity is represented by a rectangular box. Entities are connected to other entities via lines. Each line is labeled with a single arrowhead or multiple arrowheads depending on whether the relationship is one-to-one, one-to-many, many-to-one, or many-to-many. In the following section, we will discuss our relational model. For more information on database design, we refer the reader to references (6), (7), and (8).

3.3.1 Relational Database Model

The major entities involved in the system include passenger, bus, bus stop, and route. The passenger entity has tag transactions recorded and origin/destinations associated with a trip. The bus entity has tag transactions, bus mileage, and bus statistics recorded. In addition, the bus and a driver can be assigned to a link along a route. A route is made up of links. A link is a sequence of bus stops. Each route has a schedule associated with it which indicates the expected times that a bus should arrive to the bus stop and statistics which can be collected concerning route performance. A bus stop has transactions concerning passengers and busses which may occur and can appear in a schedule. The relational database is shown as an ERD in Figure 8 below. This relational model is able to obtain most of the performance measures indicated in the 1995 study by Howard P. Benn entitled "*Bus Route Evaluation Standards*", see reference (4). The rest of this subsection discusses the attributes and keys associated with each relation.

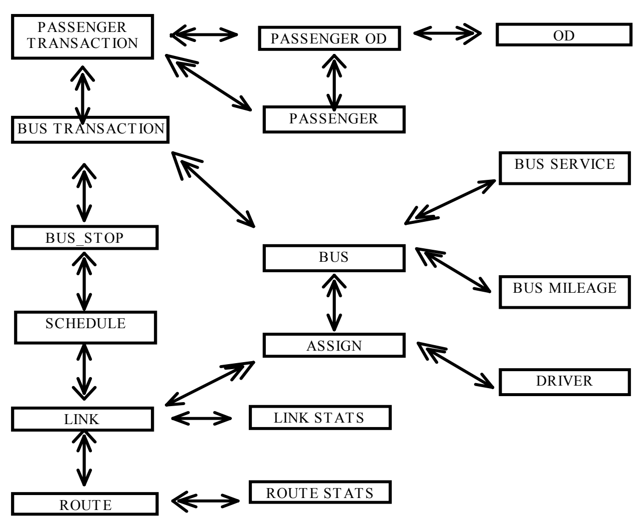

Figure 8 Entity relationship diagram

DRIVER = (<u>d-id,</u> d-name)

The DRIVER entity identifies bus drivers in the database. A unique identification number, d-id, is assigned to each driver to provide an indexing key. The bus driver's name is an attribute of the driver entity. Other attributes that may be of concern include date of birth, skill level, salary/wage, etc.

PASSENGER = (p-id)

The PASSENGER entity identifies bus riders or RF/ID tags in the database. A unique identification number, p-id, is assigned to each tag to provide an indexing key. Since this project is primarily concerned with data collection, other attributes have not been explored. For fare card applications other attributes would include amount of value on the card, date of last transaction, and other data needed for accurate billing.

PASSENGER_OD = (p-id, origin, destination)

The PASSENGER_OD entity identifies passenger origin-destination pairs. The *origin* attribute indicates what stop the passenger boarded and the *destination* attribute indicates at which stop the passenger departed the bus. A bus stop is considered a destination even if it may not be the passenger's final destination. If the passenger was merely transferring to another bus, a new origin-destination pair would be recorded.

OD = (origin, destination, number)

The OD entity identifies the number of origin-destination pairs. It is similar to the PASSENGER OD entity with the exception that this keeps track of the total number of trips taken by all passengers between a origin-destination pair.

BUS = (bus-id, bus_type, num_seats, capacity)

The BUS entity identifies busses in the database. A unique identification number, bus-id, is assigned to each bus to provide an indexing key. The bus_type identifier is used to indicate model and year. The number of seats and total capacity are necessary for determining performance measures concerning the number of standees.

BUS_MILEAGE = (bus-id, month, year, miles, day-count)

The BUS_MILEAGE entity is used to create usage statistics. The unique combination (bus-id, month, year,) allows for an indexing key. The attribute *miles* gives how many miles a month a bus drives and is obtained directly from odometer readings. *Day-count* is an attribute that describes how many days out of the month the bus operated for service.

BUS_SERVICE = (service-id, bus-id, service-type, amount, cost, date)

The BUS_SERVICE entity is used to create cost statistics. The unique identifier, service-id, is an artificial attribute that allows for easier indexing. *Service-type* indicates what type of service was performed, i.e. fueling, oil, or parts servicing. *Amount* gives how many gallons or quarts for fueling and oil servicing, while *cost* indicates how much the service_type cost. This entity, in conjunction with the BUS_MILEAGE entity, will give a record of bus performance and allow for operating cost calculations. From this, monthly operating costs of the transit system due to bus maintenance and service can be calculated.

PASSENGER TRANSACTION = (p-id, bs-id, purpose, bus-id, time, date)

The PASSENGER TRANSACTION relation identifies a set of RF/ID tag transactions associated with passenger movements at a bus stop on a specific date. Time is recorded to the exact clock time the passenger boards the bus: HH:MM:SS. The tuple (p-id, time, date) provides a unique key for indexing the relation. The *purpose* attribute indicates whether or not the passenger is getting ON or getting OFF the bus. This is necessary for extracting usage time and statistics.

BUS TRANSACTION = (bus-id, date, num-on, num-off, num-on-bus, arrive_time, dwell_time, num-standing)

The BUS TRANSACTION entity identifies a set of RF/ID tag transactions associated with bus movement at a particular bus stop on a specific date. *Arrive_time* is recorded to the exact clock time the bus arrives at the stop: HH:MM:SS. The tuple (bus-id, arrive-time, date) provides a unique key for indexing the relation. The *dwell_time* captures the amount of time the bus spends at the stop. The *num-on, num-off,* and *num-on-bus* attributes indicate the number of passengers that got on and off the bus at the stop as well as the number left on the bus at the end of the transaction. *Num-standing* indicates how many passengers do not have a seat due to seating capacity constraints on the bus.

BUS_STOP = (bs-id, location)

The BUS_STOP entity identifies bus stops in the database. A unique identification number, bs-id, is assigned to each bus stop to provide an indexing key. The location attribute will identify in some geographical manner the location of the bus stop within the transit system.

LINK = (link-id, link_name, route-id, length-mi, length-time, num_of_stops)

The LINK entity identifies links on a particular route. A route can be made up of a number of links. For example, a route may have a northbound link and a southbound link, each serviced by different busses. A unique identification id, link-id, is assigned to each link to provide an indexing key. *Route-id* indicates which route the link is on while *length-mi* and *length-time* give the length of the link and the average time to traverse the link. *Num-of-stops* tells how many stops are on the link.

LINK_STATS = (link-id, bs-id, h-avg, h-var, h-nobs, h-sumsq, h-cv, rt-avg, rt-var, rt-nobs, rt-sumsq, rt-cv)

The LINK_STATS entity allows for the reporting of performance statistics. A unique identification number, link-id, is assigned to each link to provide an indexing key. The attribute *h-avg* indicates the average headway at a bus stop. The attributes *h-var, h-cv, rt-avg, rt-var, rt-cv* correspond to headway variation, coefficient of headway variation, run time average, rt variation, and run time coefficient of variation respectively. The other attributes listed above are used to assist in the calculation of these statistics.

ROUTE = (route-id, route_name, length-mi, length-time, num-of-links)

The ROUTE entity identifies routes in the database. A unique identification number, route-id, is assigned to each route to provide an indexing key. The attribute *num_of_links* indicates how many different links the route is broken up into. The *length-mi* and *length-time* attributes give the total length of the route and the average total time to traverse the route. These attributes are used in creating summary statistical reports.

ROUTE_STATS = (route-id, month, year, miles, day-count)

The ROUTE_STATS entity allows for the reporting of usage statistics. The unique combination (route-id, month, year) provides an indexing key. The attribute *miles* indicates how many miles a particular route was traveled and is obtained from manual bus odometer readings, while *day-count* shows how many days a month the route was operational.

ASSIGN = (Bus-id, d-id, link-id, date)

The ASSIGN entity represents the allocation of a driver to a bus, to a link, on a specific date. For each day the transit systems is in operation, these unique assignments will be made in order to enable specific statistics, such as number of miles driven by a driver, to be captured.

SCHEDULE = (bs-id, link-id, schedule_time_arrival, headway, sequence, period)

The SCHEDULE entity represents the allocation of a bus stop to a specific link. The set of attributes (link-id, bs-id) provide a unique key for indexing the relation. The *schedule_time_arrival* attribute gives the time the bus is scheduled to arrive at the bus stop and allows statistics concerning on-time performance to be measured. The attribute *headway* indicates the frequency a bus arrives at the particular stop for the unique link. *Sequence* indicates what number stop the bus stop is on the link, while *period* indicates what period of day the schedule is for, e.g. AM, PM.

3.3.2 Data Validation

Validation of the data involves considering two separate situations: data transmission integrity and data discrepancies. Data transmission integrity involves errors that may occur during transmission over the RF protocol, or through serial connections from the decoder to the RF modem and the RF modem to the host computer. As mentioned in the RF modem section above, the RF protocol has error checking procedures that ensure 100% accuracy in the transmission. This will correct any errors that may occur over the RF protocol; however, the EIA232 protocol for serial inter-connectivity allows for a bit error rate of 1 in 10000 bits, which means that the serial connection can cause data discrepancies. Also, given the amount of electrical equipment capable of producing "electrical noise" on the bus, errors involving bits being changed can occur over the serial data line. This problem is corrected through a combination of firmware and software. The firmware involves programming an EPROM chip that will replicate each tag read three times. The data validation software is responsible for choosing the full tag read that is identically replicated at least twice. An example is given below:

p001,1,10:30:10
p0**b**1,1,10:30:10
p001,1,10:30:10

This involves a situation where the passenger tag was read and replicated three times; however, while the data was being transferred over the serial connection, an error occurred and one of the bits in the second replication was "flipped". The software recognizes this situation and chooses the data contained in the first and third replication as being the correct tag reading.

Data discrepancies can include situations such as recording a passenger tag that boarded and never alighted. It is not technically feasible to correct every data discrepancy situation. The most viable method for dealing with data discrepancies is to indicate that they have occurred and allow the end user to determine the significance of the discrepancy. For example, in the situation of not recording a departure read for a passenger which has boarded, the discrepancy will become apparent in the passenger's origin-destination matrix. Since the transmission of the data file from the bus to the host station may occur while the bus is en-route, it is possible that the passenger has yet to get off and still remains on the bus. Hence, the recording of the passenger's departure may occur in the transmission of an ensuing data file. Given that each of the data files may be validated and input into the database separately it would involve a significant amount of processing time to parse all the data in order to ensure that the passenger's departure was captured. If a data discrepancy does occur and the passenger's departure is never captured, the data indicating that the passenger boarded is still useful for statistical purposes and should not be discarded. The passenger's origin-destination matrix will show an originating bus stop and no destination stop.

The other case the software is capable of handling involves when a bus does not stop at the bus stop. In this case, it is anticipated that the bus will at least pass over one of the tags at the stop and obtain a tag reading. The software correctly recognizes this situation and assigns a dwell time of zero for that particular stop. It may be possible that a bus misses a departure bus stop tag read as it leaves the bus stop. Another discrepancy that can occur is the bus completely missing a bus stop tag read. The software is able to correctly identify this problem by comparing each bus stop tag that was read with a known schedule of all stops along the route that the bus was servicing at that time. This known schedule is obtained from the SCHEDULE entity described in Section 3.1. The software displays a listing of all stops missed on that particular link.

The final discrepancy that can occur is an undesirable grouping of the data. The software assumes that data for each bus stop is grouped at the ends by an arrival bus stop tag read and a departure bus stop tag read. This is considered to be an optimal grouping; however, it is possible that either the arrival or the departure bus stop tag read is missed. Three cases which exist for any stop are explained below:

Case #1 - Optimal Grouping. Bus Stop 001 has an optimal grouping and Bus Stop 002 has an optimal grouping. In this case, the departure tag read of the first bus stop immediately precedes the arrival tag read of the ensuing bus stop along the route. All statistics for this stop will be correctly computed.

Example of Case # 1

s001,1,10:30:00	first stop arrival stop read
01	
p001,1,10:30:10	
00	
s001,1,10:30:20	first stop departure stop read
s002,1,10:34:00	second stop arrival stop read
01	
p002,1,10:34:10	
00	
s002,1,10:34:20	second stop departure stop read

Case #2 - Non-Optimal Grouping. Bus Stop 001 has an optimal grouping and Bus Stop 002 has missed the arrival stop read. Therefore there is no departure tag read with a arrival tag read immediately afterwards. In this case, passenger #2 (p002) will be incorrectly associated as boarding at Bus Stop 001. This will cause all the attributes for both bus stops to be incorrect with the exception of the dwell time.

Example of Case # 2

s001,1,10:30:00	first stop arrival stop read
01	
p001,1,10:30:10	
00	
s001,1,10:30:20	first stop departure stop read
01	
p002,1,10:34:10	
00	
s002,1,10:34:20	second stop departure stop read

Case #3 - Non-Optimal Grouping. Bus *i* has missed the departure stop read and Bus Stop $i + 1$ has an optimal grouping. In this case, there is no departure tag read immediately preceding an arrival tag read. This will cause the dwell time for Bus Stop 001 to be computed as a value of zero, however all other attributes will remain unaffected.

Example of Case # 3

s001,1,10:30:00	first stop arrival stop read
01	
p001,1,10:30:10	
00	
s002,1,10:30:20	second stop arrival stop read
01	
p002,1,10:34:10	
00	
s002,1,10:34:20	second stop departure stop read

Although the software can tell when one of the non-optimal grouping occurs, it can not distinguish between Case #2 and Case #3. A message is displayed for the stops that had non-optimal groupings indicating that the statistics for the stops may be incorrect.

3.3.3 Database Testing

In order to examine the behavior of the database under sample processing conditions, we developed a simulation model of the field site route. We refer the reader to Section 3.4 for a description of the field site route. Data was collected using the automated transit ridership data collection (ATRDC) software package developed by the Texas Transportation Institute and distributed through the University of Florida's Center for Microcomputers in Transportation, see reference (9). The software allows a person using a laptop computer to record boarding and alighting counts for each stop along a route while riding the bus. One bus services twenty-two stops on the route with a scheduled headway of 30 minutes. A team of fourth year systems engineering undergraduate students, collected data on forty, 30 minute loops of the bus on the route. Since the simulation was to be used to test the processing of the data, the fidelity requirements of the simulation are significantly less than those required when using the simulation to model the bus route performance. The passenger arrival distribution was assumed to be Poisson. The boarding data can then be used to set the mean of the Poisson process for each bus stop. The number of alighting passengers at a stop can be modeled as a binomial random variable which is dependent upon the number of passengers on the bus as it approaches a stop. The count data obtained from the software can then be used to set the probability that a passenger will alight at a stop along the route. Both of these assumptions are justified given previous literature on modeling bus route operations. From the time analysis report available through ATRDC, the minimum, maximum, and average running times between bus stops was available. Using this information, a triangular distribution was selected to model the running times between stops. We refer the reader to Adamski (10) for information concerning passenger service models and to Koffman (11) for information on simulating other aspects of bus operations.

The simulation model mimics the traveling of a bus along the route. At each stop, the number of boarding and alighting passengers is computed, the bus and passenger transactions are written to an ASCII file, and the bus routed to the next stop along the route. The ASCII file of bus and passenger transactions is in the same format as that expected by database validation module, see Table 3 of this report. The size of the output file was varied in order to predict the behavior of the database in terms of processing speed and storage requirements. Twelve datasets were created to represent system operation ranging from one-half day to 10 days. For each dataset, the number of events, validation time in seconds, input time in seconds, and storage size in bytes were recorded. For each dataset, the measurements were recorded starting with an empty version of the database. The number of events is the number of passenger and bus transaction events within the dataset.

The current operation of the system calls for the transactions to be recorded and sent to the host computer via a RF modem in near real-time. As transactions occur at the bus stop, the tag data is captured, stored, and forwarded to the host computer and appended to the raw data file described in the next section. At the end of the day, the data in the raw data file is then processed by the data validation and data input procedures described in the next section. Thus, we concentrated on the simulating the processing of a days worth of data. By increasing the number of simulated days, we are essentially testing how the database will react to a larger system. The information reported here is part of a study directed by the principal investigator and performed by a fourth year engineering student to meet his senior thesis requirements. For a more complete discussion of the simulation and testing results, we refer the reader to Motter (12). The results taken from Motter (12) are given in Table 4. All testing was performed on a Intel based 486dx2-66 computer running Windows 3.1.

TABLE 4 Database Testing Responses

Number of Events	Database Size (bytes)	Input Time (sec)	Validation Time (sec)
795	76800	61.63	20.71
1668	147968	154.07	44.00
2463	201984	250.52	64.32
3336	270336	372.78	87.49
5004	396544	651.03	133.25
6672	527872	1015.95	178.78
8340	653312	1448.44	223.98
10008	776448	1911.41	271.88
11676	893696	2472.3	318.90
13344	1027584	3130.76	364.48
15012	1142016	4722.39	412.60
16680	1275904	7747.46	456.57

A regression line was fit to each output variable as a function of the number of events. The resulting models were:

Database Size = 20583.2 + 75.196(Number of Events) with $R^2 = 1$

Validation Time = -3.571 + 0.027578(Number of Events) with $R^2 = 1$

ln(Input Time) = -6.16709 + 1.50427 ln(Number of Events) with $R^2 = 0.979$

As expected database size and validation time was a linear function of the number of events. For the field site route, the number of events for one day was approximately 1668. Thus, for each day of operation approximately 146 Kbytes of data will be added to the database with an input time of approximately 147 sec and a validation time of approximately 42 seconds. In addition to monitoring the above performance variables, the data stored in the database for such items as the number of boarding passengers, dwell time, etc. were compared to the numbers produced by the simulation. The results matched. This verifies that the database is computing these fundamental statistical quantities correctly.

Two important conclusions can be drawn from these results. First, for a larger system with many more events the input time can take hours of processing time if all transactions are stored and processed at one time. For a full scale implementation, we would recommend near real-time processing of the transactions in order to spread the computation out over the entire operating day. Validation can take more time than shown if checking for system exceptions and transmission errors is necessary. The possibility of improved hardware and communication protocols should be explored to reduce the need for validation checking. Secondly, for a larger system the requirements for hard drive storage can quickly exceed current disk storage capacity. The storage of passenger and bus transactions is the significant component of the database size. For a full scale system, we recommend developing policies which

- summarize bus and passenger transactions into periodic (e.g. weekly, monthly) statistical relations
- backup to tape and then purge bus and passenger transactions from the database

While suggestive in nature, these results are clearly preliminary. We therefore recommend a more complete analysis of the database operating policies be explored before full implementation. This would also include the timing of statistical reporting capabilities. In the next section, we discuss the software application procedures and operating functionality.

3.4 SOFTWARE APPLICATION

The goal of the software developed for this project was to collect and organize the automatically captured RF/ID data in order to display bus system performance data. This data can then be used to aid decision making concerning the optimization of the bus system. The software consists of the following modules:

External Modules

1. Raw Data File
   a) An ASCII file containing the data collected from the passengers and bus stop tags.

Clarion for Windows Modules

1. Data Validation Module
   a) A Clarion for Windows module which takes a raw data file and validates the replicated data. This module is responsible for choosing the single record out of the replicated records to be entered into the database. It places each validated record into a valid data input file. This file is then entered into the Data Input Module for processing into the database.

2. Data Input Module
   a) This is a module inside of Clarion for Windows that processes the file containing records obtained from the Data Validation Module into the database. This module is also responsible for computing certain statistics such as dwell time, number standing, number on/off etc. It then outputs these statistics into the appropriate database files within the database. This module modifies the following database files: PASSENGER TRANSACTION, BUS TRANSACTION, and PASSENGER OD, OD, and LINK STATS entities.

3. Browse Database Files
   a) The database files are the entities and relations of the relational database that were described in the last section. This module allows these entities and their attributes to be browsed by the user. It also provides basic query abilities. These abilities will be described in Section 3.4.2 of this report. This module also allows the user to input/update information into the database.

4. Reporting Module
   a) This module displays information concerning the performance of buses, routes, drivers, etc. Example reports include: on-time performance, boarding/alighting patterns, cost and usage statistics, etc.. The functionality of this module is discussed in Section 3.4.3 of this report.

Figure 9 illustrates how the modules interact. The raw data input module captures the data transmitted during the operation of the automatic RF/ID system for storage and retrieval into the database. The data validation module parses through the raw captured data and eliminates data discrepancies. The input module places the data into the proper fields in the database. The reporting module allows standard statistical reports to be generated based on queries into the database. The user interface module allows attribute data entry through input forms for data required by the database. The next section will present an overview of the software functionality.

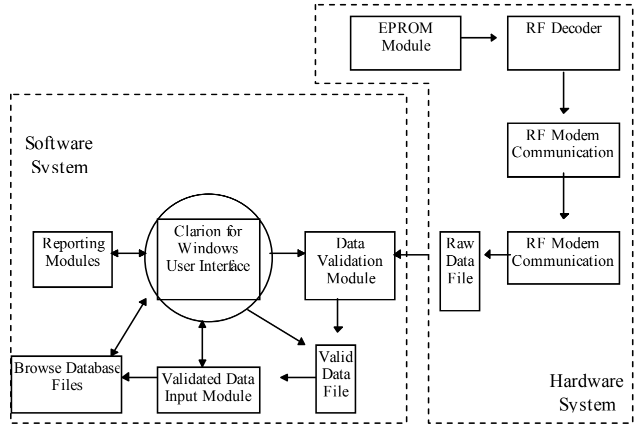

FIGURE 9 Software/Hardware subsystems

3.4.1 Interface Description

The user sees a menu bar from which he may choose to either browse the database or validate a raw data file. The menu bar is a permanent fixture in the background and may be selected at anytime or place within the application. If the user chooses to validate a raw data file, he can then input the filename of the file to be validated or search for a file to validate. Once the file has been located, the validation process begins with the push of a button. After the validation process has been completed, the user is automatically transferred into the Input Data Module where he may view the validated file for errors. If the file has been satisfactorily validated, it is input into the database by the selection of a button. Once the data has been validated and input into the database, the user may then browse the database.

Each entity or relation as described in the Relational Database Model section is considered a file by Clarion. The user may choose to browse any of these files from the menu bar. Browse procedures for each of the files provide the user certain functionality depending on which file is being browsed and its relationships within the relational model. Each browse procedure consists of a browse window with buttons and/or tabs. The browse window is responsible for displaying the file. Standard buttons in all browse procedures are the *edit* buttons which include *insert, change,* and *delete* buttons to allow the user to either input data, change existing data, or delete data from the file. Other buttons provide links to either parent or child files. A parent file has a one-to-many relationship with another file (single arrow to double arrow in the database diagram) and is linked to the child through a button. A child file has a many-to-one relationship with another file (double arrow to single arrow). Child files have a set of tabs across the top of the browse window. Each tab is named according to one of the indexing keys of the file that is being browsed as described above in Section 3.3.2 of this report. Associated with all the tabs is a single button that links to a parent file. This button allows the user to select a record from the parent file and view all associated records in the file currently being viewed. Other buttons provide relational links to child files. This allows the user to obtain information about a particular entity in the file currently being viewed. Pressing a button is functionally equivalent to a basic SQL query. A more detailed explanation is given below in each of the browse procedures.

3.4.2 Browse Procedure Descriptions

In each of the procedure descriptions, the word "*file*" is equivalent to the word "*entity*" used in the Section 3.3.2 of this report. A record is defined as the complete set of attributes associated with an entity. The variable *x* corresponds to the attribute of the record highlighted by the user.

BUS PROCEDURE

This procedure allows the bus file and its attributes (<u>bus-id,</u> bus_type, num_seats, capacity) to be viewed within a browse window. The *edit* buttons allow the user to input a new bus and its attributes into the database, change the attributes of an existing bus, or delete a bus from the database.

Since the BUS entity is a parent file with four child files (PASSENGER TRANSACTION, BUS_MILEAGE, BUS_SERVICE, ASSIGN), this procedure contains four buttons that link to each of the individual child files. Selecting a bus record from the browse window and pressing one of the four buttons of the child files will show all the attributes of the child file associated with the selected record in the browse window. For instance, if the user selected a bus record and pressed the button linking to the BUS_MILEAGE child file, the user would then see all the attributes (<u>bus-id</u>, <u>month</u>, <u>year</u>, miles, day-count) for the selected bus. The same occurs the other three child files. This corresponds to the basic SQL query mentioned above. Each button and its associated query are given below.

- BUS_MILEAGE "Display how many days each month BUS *x* operated."
- PASSENGER TRANSACTION "Display all transaction records associated with BUS *x*."
- BUS_SERVICE "Display what type of service BUS *x* received each month."
- ASSIGN "Display all assignments that BUS *x* was given."

BUS STOP PROCEDURE

This procedure allows the BUS STOP file and all its attributes (<u>bs-id</u>, location) to be viewed within a browse window. This particular file is both a parent file to the PASSENGER TRANSACTION file and a child file to the SCHEDULE file. The standard *edit* buttons are included. Two buttons, a SCHEDULE button and a PASSENGER TRANSACTION button, provide relational links to the files. Pressing the SCHEDULE button brings up a browse window containing the SCHEDULE file and all its attributes (<u>bs-id</u>, <u>link-id,</u> schedule_time_arrival, headway, sequence, period). The user can then select which link he would like to see given an associated bus stop. Pressing the PASSENGER TRANSACTION button brings up all associated transaction records for the particular bus that the user has highlighted in the browse window. Each button and its associated query is given below.

- PASSENGER TRANSACTION "Display all transaction records associated with BUS STOP *x*."
- SCHEDULE "Display all bus stops on LINK *x*."

PASSENGER PROCEDURE

This procedure allows the PASSENGER file and all its attribute (<u>p-id</u>) to be viewed in the browse window. *Edit* buttons are also included. This file is a parent file to the PASSENGER TRANSACTION and PASSENGER OD files. Two buttons to these files are provided for the relational links. The buttons and the associated SQL query are given below.

- PASSENGER TRANSACTION "Display all transactions associated with PASSENGER *x*."
- PASSENGER OD "Display all origin-destinations for PASSENGER *x*."

ROUTE PROCEDURE

This procedure allows the ROUTE file and its attributes (<u>route-id</u>, route_name, length-mi, length-time, num-of-links) to be viewed in the browse window. Standard *edit* buttons are provided. This file is a parent file of the LINK and ROUTE_STATS files. The buttons and the associated query are given below.

- LINK "Display all links on ROUTE *x*."
- ROUTE_STATS "Display how many days for each month ROUTE *x* operated."

LINK PROCEDURE

This procedure allows the LINK file and its attributes (<u>link-id</u>, link_name, route-id, length-mi, length-time, num_of_stops) to be viewed. The standard *edit* buttons are provided. This file is a child file of the ROUTE and SCHEDULE files. Instead of two buttons for each file, the browse window contains tabs that contain the indexing keys of the parent file. Both tabs are associated with one button. The one button links to both files independently. Selecting a tab changes the file associated with the button. As in the BUS STOP procedure, when the button is pressed an associated parent browse window is displayed. Once a record is selected in the parent browse window, the associated records in the child file are displayed. Again, selecting a tab and pressing the button corresponds to one of the SQL queries below.

- SCHEDULE TAB "Display all links that BUS STOP *x*. is on."
- ROUTE TAB "Display all links that are on ROUTE *x*"

DRIVER PROCEDURE

This procedure allows the DRIVER file and its attributes (<u>d-id,</u> d-name) to be viewed. The standard *edit* buttons are provided. This file is a parent file of the ASSIGN file. One button is provided for the relational link. The SQL query corresponding to the button is given below.

- ASSIGN "Display all assignments DRIVER *x* was given."

ASSIGN PROCEDURE

This procedure allows the ASSIGN file and its attributes (<u>Bus-id</u>, <u>d-id</u>, <u>link-id</u>, <u>date</u>)
to be viewed in the browse window. The standard *edit* buttons are provided. This file is the child file of the BUS, DRIVER, and LINK files. Three buttons are provided that link to each file. The associated SQL queries are given below.

- BUS "Display BUS information on BUS *x*."
- DRIVER "Display DRIVER information on DRIVER *x*."
- LINK "Display LINK information on LINK *x*."

As in the LINK procedure, tabs are provided along the top of the browse window. Each tab contains an indexing key into one of the three parent files. All the tabs correspond to one particular button. The file that the button is linked to, and the corresponding SQL query, changes according to which tab is selected. When a tab is selected and the button is pushed, the corresponding browse window for the parent file is displayed. The user then selects a record out of the parent window which causes all associated records in the child to be displayed. The SQL queries associated with this action is given below.

- BUS TAB "Display all assignments given to BUS *x*."
- DRIVER TAB "Display all assignments given to DRIVER *x*."
- LINK TAB "Display all assignments given to LINK *x*."

SCHEDULE

This procedure displays the SCHEDULE file and its attributes (<u>bs-id</u>, <u>link-id,</u> schedule_time_arrival, headway, sequence, period). The standard *edit* buttons are provided. This file is the parent of the BUS STOP and LINK files. Two buttons correspond to the relational links for these files. The SQL queries associated with these buttons are given below.

- BUS STOP "Display information for BUS STOP *x*."
- LINK "Display information for LINK *x*."

PASSENGER TRANSACTION PROCEDURE

This procedure displays the PASSENGER TRANSACTION file and its attributes (<u>p-id,</u> bs-id, purpose, bus-id, <u>time</u>, <u>date</u>) in the browse window. The standard *edit* buttons are provided. This file is a child file to the BUS, PASSENGER, and BUS STOP files. Tabs associated with the indexing keys into each of the parent files are along the top of the browse window. These tabs are all associated with one button that contains the relational link depending on which tab is selected. The SQL queries for the selected tabs are given below.

- BUS STOP TAB "Display all transactions associated with BUS STOP *x*."
- PASSENGER TAB "Display all transactions associated with PASSENGER *x*."
- BUS TAB "Display all transactions associated with BUS *x*."
- BUS TRANSACTIONS TAB "Display all passengers who boarded at a particular stop on a particular day."

In addition, individual buttons to each of the parent files are provided to correspond to different SQL queries. The buttons and the associated queries are given below.

- BUS TRANSACTIONS "Display BUS TRANSACTIONS for PASSENGER TRANSACTION *x*."
- BUS STOP "Display BUS STOP information on BUS STOP *x*."
- PASSENGER "Display PASSENGER information on PASSENGER *x*."
- BUS "Display BUS information on BUS *x*."

BUS TRANSACTIONS PROCEDURE

This procedure displays the file BUS TRANSACTIONS and its attributes (p-id, bs-id, purpose, bus-id, time, date). The standard *edit* buttons are provided. This file is the child file of the PASSENGER TRANSACTION file and therefore has a button related to the parent file. The SQL query associated with this button is given below.

- PASSENGER TRANSACTION "Display all passengers who boarded a particular bus at a particular bus stop, on a particular day."

BUS_MILEAGE PROCEDURE

This procedure displays the file BUS_MILEAGE and its attributes (bus-id, month, year, miles, day-count). The standard *edit* buttons are provided. This file is the child file of the BUS file. A button relates this file to its parent file. The SQL query associated with this button is given below.

- BUS "Display information for BUS *x*."

BUS_SERVICE PROCEDURE

This procedure displays the file BUS_SERVICE and its attributes (service-id, bus-id, service-type, amount, cost, date). The standard *edit* buttons are provided. This file is the child file of the BUS file. A button relates this file to its parent file. The SQL query associated with this button is given below.

- BUS "Display information for BUS x."

STATE PROCEDURE

The STATE file is not part of the relational database. It is used to keep track of the number on board a bus between the transmissions of data files from the bus. This allows the statistics for the bus to remain accurate even though it may transmit data over several individual transmissions of data files. When the INPUT Module is called, this file is checked to obtain the previous state of a bus for that particular day in order to get the number on board the bus.

PASSENGER ORIGIN-DESTINATION

This procedure displays the file PASSENGER OD and its attributes (p-id, origin, destination).
It provides the location (bus stop) associated with a passenger boarding a bus and the location (bus stop) associated with the passenger alighting a bus. The standard *edit* buttons are provided. This file is the child file of the PASSENGER file and therefore has a button related to the parent file. The SQL query associated with this button is given below.

- PASSENGER "Display information for PASSENGER *x*."

OVERALL ORIGIN-DESTINATION

This procedure displays the file OD and it attributes (origin, destination, number). It is used for statistical purposes. This file is an ORIGIN-DESTINATION matrix that keeps track of the number of trips between two bus stops. Each time a passenger origin-destination is completed this file is updated to indicate that another trip between two bus stops has occurred. It is useful in showing the flow of passengers through the transit system.

LINK STATS

This procedure displays the file LINK STATS and the attributes (link-id, bs-id, h-avg, h-var, h-cv, rt-avg, rt-var, rt-cv). It is also used for statistical purposes. This file tracks the performance of buses along a particular link. It is useful in showing how a particular route is adhering to its scheduled headway.

INPUTFILE

This file is not part of the relational database. The Validation Module places validated data into this file for incorporation into the database. Standard *edit* buttons are provided to allow the user to ensure the data was validated correctly by the Validation Module.

3.4.3 Report Functionality

A report is a customized presentation of the data within the database for a specific purpose. Reports are typically the result of a set of queries into the database. Each of the queries given in the above section are examples of the preliminary reporting capabilities of the software. Reports should be available for common sets of queries which occur on a periodic basis. The reports can then be used to analyze the performance of the system. In addition, the reports can be used as a basis for developing reports required by state and federal funding agencies. The reporting capabilities of the database include:

- bus statistics
- route statistics
- schedule adherence

The following subsections, will present an overview of each of the above reporting components.

3.4.3.1 Bus Statistics Reports

The purpose of the bus statistical report is to report on the costs and service effectiveness of operating the busses within the system. During bus operations busses incur costs associated with fuel, oil, and maintenance. The relations BUS_MILEAGE and BUS_SERVICE support queries associated with bus operations. In order to support reporting functionality, the following additional report files were created:

BMRPT = (bus-id, current_month_mileage, month_lastyear_mileage, current_year_mileage, last_year_mileage)

FUEL_REPORT = (bus-id, current_month_fuel_cost, month_lastyear_fuel_cost, current_year_fuel_cost, last_year_fuel_cost)

OIL_REPORT = (bus-id, current_month_oil_cost, month_lastyear_oil_cost, current_year_oil_cost, last_year_oil_cost)

SERVICE_REPORT = (bus-id, current_month_m_cost, month_lastyear_m_cost, current_year_m_cost, last_year_m_cost)

TTL_COST_BUS = (bus-id, current_month_tot_cost, month_lastyear_tot_cost, current_year_tot_cost, last_year_tot_cost)

BCOST_MILE = (bus-id, current_month_cpm, month_lastyear_cpm, current_year_cpm, last_year_cpm)

BMRPT contains the current month's mileage, the mileage for the same month last year, the current year's total mileage, and last year's total mileage for each bus. The FUEL_REPORT contains fuel costs associated with a bus for the current month this year and last year as well as the totals for this year and last. The OIL_REPORT contains the total oil costs associated with a bus for the current month this year and last year as well as the total oil costs for this year and last. The SERVICE_REPORT contains total costs for other maintenance services associated with a bus for the current month this year and last year as well as the total service costs for this year and last. The TTL_COST_BUS contains the total of all costs associated with a bus for the current month this year and last year as well as the totals for this year and last. The BCOST_MILE report contains the cost effectiveness of operating each bus based on the statistics cost/mile for the current month this year and last year as well as the totals for this year and last.

In order to compute the values associated with the report files one must cycle through the appropriate database relations in order to extract the appropriate information. For example, to compute the BMRPT report file after the user has selected an appropriate month and year, one must loop through the BUS relation to extract each bus identifier, bus-id. Then, for each bus-id the BUS_MILEAGE file must be scanned for matching bus-id. For each matching bus-id, if the month and year match then the appropriate mileage counter is incremented. The bus-id and mileage values can then be added to the BMRPT file. The logic to compute the costs associated with fuel, oil, and service is very similar except that after finding the bus-id from the BUS relation one needs to check for matching bus-id, month, and years in the BUS_SERVICE relation and sum the appropriate costs. The cost per mile is then computed using the values in the TTL_COST_BUS and BMRPT files for each matching bus-id.

3.4.3.2 Route Statistics Reports

The purpose of the route statistical report is to report on the service effectiveness of routes within the system. The primary factors of interest are the passengers, operating days, and route mileage. Service effectiveness can be measured as the number of passengers per mile for the route and the number of passengers served per day on the route. The relations RTE-MILEAGE and RTE-OPER support queries associated with route operations. In order to support reporting functionality, the following additional report files were created:

OPER_DAYRPT = (route-id, current_month, month_lastyear, current_year, last_year)

RMILERPT = (route-id, current_month, month_lastyear, current_year, last_year)

RMILE_DAYRPT = (route-id, current_month, month_lastyear, current_year, last_year)

PASS_COUNTRPT = (route-id, current_month, month_lastyear, current_year, last_year)

OPER_DAYRPT contains the current month's operating day totals, the totals for the same month last year, the current year's totals, and last year's totals for each route. The RMILERPT contains a summary of the number of miles covered on a route for the current month this year and last year as well as the totals for this year and last. The RMILE_DAYRPT contains a summary of the average miles per day covered on a route for the current month this year and last year as well as the totals for this year and last. The PASS_COUNTRPT contains passenger count totals for the current month this year and last year as well as the total counts for this year and last.

Given the above report files, the appropriate data must be extracted from the database, any additional computations performed and then the values displayed in a report format. The logic for the extraction of the values for OPER_DAYRPT and RMILERPT is similar to that discussed for the BMRPT report file except now we are concerned with a route and use the RTE_MILEAGE and RTE_OPER relations. For each matched route-id, we extract the appropriate data by month and year. The passenger counting logic requires the use of the PASSENGER TRANSACTION, ASSIGN, ROUTE, and LINK relations. The ROUTE relation is examined to get each route-id. For each record in the PASSENGER TRANSACTION file get the bus-id and date if the date matches the specified month and year. Then the appropriate link is found in the ASSIGN file for matching bus-id and date. For the matching link, get the route-id. This indicates that a passenger transaction occurred on the appropriate route during the appropriate month and year. The passenger transaction can then be counted into the total that occurred for the given month and year. For a more detailed, pseudo-code description of this logic with refer the reader to Chitre (13). Given the data in PASS_COUNTRPT, OPER_DAYRPT, and RMILERPT, the service effectiveness factors can be computed. For example, to get the passengers served per mile, one divides the values in the PASS_COUNTRPT by the values in the RMILERPT for each appropriate route-id.

3.4.3.2 Schedule Adherence Report

The schedule adherence report is used to measure the on-time performance of the busses along a route in the transit system. The report by Levinson(2) suggests that 65% of transit managers consider on-time performance as essential, critical, highly important, or very important. The schedule adherence report shows, see page 39 of Levinson (2), for any bus stop along the route for a specific time period during the day

1. the number of trips passing the stop
2. the number and percentage of trips that are
   a) on time
   b) ahead/behind schedule by 0-2 minutes, 2-5 minutes, 5-9 minutes, more than 10 minutes.

In order to support reporting functionality, the following additional report file was created:

ADHERE = (bs-id, num_total_trips, num_5+min_early, num_2-5min_early, num_0-2min_early, num_on_time, num_0-2min_late, num_2-5min_late, num_5-10min_late, num_10+min_late)

An example of the schedule adherence report is as follows:

	Stop 1	Stop2
Total Trips	12	14
5+ min. early	1	0
2-5 min. early	1	1
0-2 min. early	1	2
On Time	5	5
0-2 min. late	2	3
2-5 min. late	2	3
5-10 min. late	0	0
10+ min. late	0	0

This allows a transit manager to easily determine the bus stops which have consistent problems meeting the schedule.

The formation of this report is dependent on processing the BUS TRANSACTION, SCHEDULE, and ASSIGN relations. Given the user input of (route-id, date, starting time of period, ending time of period), search the ASSIGN table for the entered date and route and store the corresponding bus-id. From the bus transaction relation, BUS TRANSACTION, process all transactions occurring on the entered date for the specified time period for the current bus-id. Within the SCHEDULE relation find the scheduled time of arrival for the current bus stop and compare this to the actual time of arrival in BUS TRANSACTION. The difference is the schedule deviation which then can be categorized according to the criteria in the ADHERE file.

3.4.3.3 Additional Reports

Since the scope of this project was limited to developing a prototype, we consider the above reports sufficient to demonstrate feasibility of the use of the database in transit practice. There are additional reports which can easily be supported by the database. A potential list for future development is given in Table 24 on page 34 of Levinson(2). The current reporting functionality of the prototype does not take full advantage of the automatic data collection process. With additional sensors mileage, fuel consumption, and other bus operating status could be automatically captured. This, for example, would alleviate the need for user input in the BUS_MILEAGE relation. These additional sensors may allow maintenance to be scheduled more efficiently which has become an important issue given the increasing average age of bus fleets in transit practice. In addition, the reporting functionality was tailored to the specific route selected for field trials. A full implementation would require making the software able to better handle a general transit system. The next section discusses the preparation and operation of the prototype along the selected field route.

3.5 FIELD SITE PREPARATION AND TESTING

This section of the report discusses the recruitment of test subjects, the preparation of the field site for testing, and the operational testing of the prototype.

3.5.1 Subject Recruitment

The University of Virginia's University Transportation Services (UTS) identified the Hereford College Dormitory-Hospital Complex route for use during field tests. UTS serves the student and staff population at the University of Virginia with 4 distinct routes within a 4 square mile area. The Hereford College Dormitory-Hospital Complex Route was selected for this study. Figure 10 indicates the selected route within the context of the University. There are 22 bus stops on the route serviced by 1 bus between the hours of 7:30 a.m. to 5:30 p.m. The route takes approximately 30 minutes per loop with 2 loops scheduled per hour. The stop located at the Hereford College Dormitory has been added this year. The route runs along Jefferson Park Avenue to the Hospital Complex. Jefferson Park Avenue(JPA) is a main thoroughfare within the city/university area. Traffic conditions on JPA are representative of typical transit conditions. Ridership surveys were administered in Stage 1 of the project and potential test subject volunteers were identified. These subjects attended an information session in order to explain the project and field site study plan. The data collection period for the test subjects was February 1, 1996 to April 30, 1996. The test subject's participation involved:

1. Carrying the RF/ID tag on their person (within a pocket, purse, book bag, etc.) while using the University Transit bus route identified as the Hereford Special.

2. Filling out a pre-study survey and a usage survey involving their perceptions of the study and the technology. The pre-study survey was completed during the information session. The usage survey was to be completed and returned by February 22, 1996.

To ease the concerns of the participants, the information gathered during the study via survey and through the RF/ID data collection system was handled anonymously. A unique identification number was assigned to each tag; however, no records were kept to identify which tag was assigned to each participant. This is similar to the tag acting as a debit card based bus pass. Twenty-six participants volunteered for the study. The participants were paid $10 for participating in the study. In addition to the test subject surveys, we conducted surveys in order to ascertain passenger acceptance of the technology. The results of these surveys are discussed in Section 3.6 in this report.

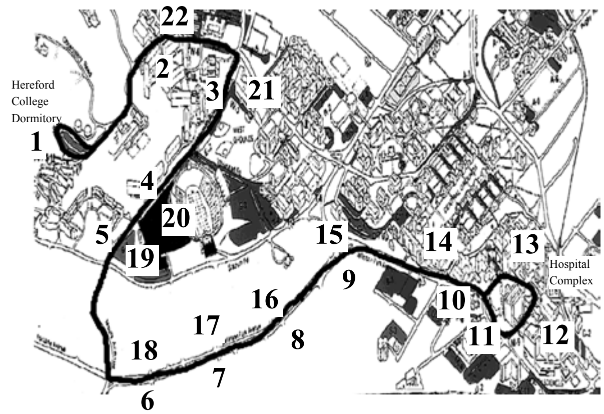

Figure 10: Hereford College Hospital Route

3.5.2 Field Study Preparation

From 11/6/1995 to 11/27/1995 the Automated Transit Ridership Data Collection Software (ATRDCS) available through the University of Florida's McTrans Transportation Research Center was used to collect basic ridership data on this route. Using a laptop computer, a team of undergraduate engineering students from the University of Virginia observed and recorded passenger counts and bus running times while riding on the bus. Approximately, 886 passengers boarded the bus during a total time of 1200 minutes. Thus, the bus is serving approximately 44 passengers per hour. This data was used to develop a simulation program which generates data for testing and evaluating the database functionality. The results of the database testing were given in Section 3.3.1 of this report.

Table 5 indicates the distances between stops and the number of embedded tags to be located at each stop. The bus route runs on both University roads and on City of Charlottesville roads. The City of Charlottesville's Department of Public Works was contacted and permission secured for the embedding of tags in city streets. Using a saw designed to cut pavement, the pavement was sliced to a depth of 5-8 cm (2-3 inches). No digging permits were necessary because the tags are only embedded 2-3 inches under the road surface. The tag was then placed in the slot and an epoxy road sealer was used to seal the cut. This process was successfully tested with a tag and a decoder attached to a portable computer.

TABLE 5 Bus Route Characteristics

Bus Stop Number	Distance to Next Stop (meters)	Cumulative Distance (meters)	# embedded tags
1	482.7	482.7	4
2	482.7	965.4	4
3	482.7	1448.1	4
4	160.9	1609	3
5	643.6	2252.6	4
6	160.9	2413.5	4
7	321.8	2735.3	4
8	241.35	2976.65	4
9	402.25	3378.9	4
10	321.8	3700.7	4
11	643.6	4344.3	4
12	160.9	4505.2	4
13	643.6	5148.8	5
14	321.8	5470.6	4
15	321.8	5792.4	4
16	160.9	5953.3	4
17	482.7	6436	4
18	482.7	6918.7	4
19	160.9	7079.6	4
20	482.7	7562.3	3
21	482.7	8045	4
22	643.6	8688.6	4

One tag is positioned near the bus stop sign at the place most often covered by the bus during a dwell at a bus stop. Another tag is placed closer to the middle of the roadway in order to ensure a tag read when a bus does not stop at the bus stop. The basic tag layout for a stop is given in Figure 11. There is some minor variation for each stop because the configuration of each stop is different. For example, some stops on the route have an off road loading/unloading zone, but other stops simply have the bus stop alongside of the road. Tag 3 in the picture is positioned near the middle of the lane so that the tag will pass midway between the wheels of the bus. Tag 1 is intended to capture a bus that pulls up short of the stop due to traffic or other conditions. Tag 2 is intended to capture the normal stopping zone of the bus and tag 4 is intended to capture the bus departing from the stop. Bus stop 13 has five tags with the fifth tag place 6.42m to the west of tag 1. Stops 4 and 20 each have 3 tags. Tag 3 was unnecessary due to the width of the road.

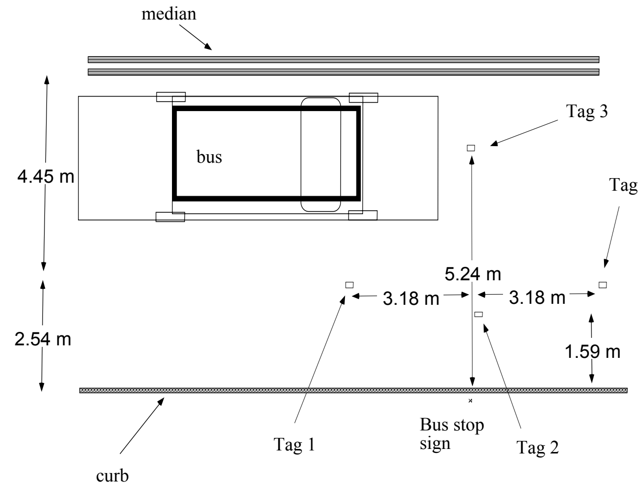

FIGURE 11: Basic tag layout

The major difficulty in determining the placement of the tags is the fact that a bus can stop anywhere at the bus stop in relation to the bus stop sign. A normal stop is considered to be one where the bus stops so the front doors are aligned with the bus stop sign; however, the bus may stop at the sign, some distance before the sign, or some distance after the sign. Most of these variations from the norm occur when there is not a dedicated pull in lane at the bus stop for the bus. The bus stopping location at the bus stop can vary by driver or may be dependent on a number of other factors. One of these other factors is curbside parking. If cars are allowed to park immediately before a stop, this can affect the ability of the bus to pull up to the curb in time to stop at the bus stop sign. This generally will cause the driver to stop farther down from the bus stop sign in order to allow the rear wheels of the bus to clear the parked cars. For bus stops such as this, the general cross design was shifted down the road to the right of the bus stop sign (relative to Figure 11). At other stops, cars may be allowed to park immediately after the bus stop sign. This would affect the driver's ability to pull the bus out of the bus stop; therefore, the bus would generally stop short of the bus stop sign. In this case, the general cross design was shifted up the road to the left of the bus stop sign (relative to Figure 11). In some cases, parking was allowed immediately before and after the bus stop not allowing the bus to pull close to the curb at all. For these stops, the general cross design was pushed out from the curb closer to the center of the road. For all of these cases, the possibility also existed that at any particular time no cars may be parked thus allowing the bus to pull normally up to the stop.

Other factors that were considered when placing the tags at the stop. For example, one must consider a driver's tendency to sometimes stop where a crowd was waiting and not actually where the bus stop sign was located. Secondary roads and driveways close to the stop were considered as some drivers did not like to pull up to a stop and block the road or driveway. Timed stops also presented a problem as a number of buses may be at the stop waiting. This problem was remedied by embedding one more tag farther up from Tag 1 (to the left in Figure 11) and placing another tag farther down (to the right in Figure 11) from Tag 4.

Proper location of the tags at the stops also depends on the speed of the bus and the strength of the antenna located underneath the bus. We attempted to find a starting configuration which minimized the number of tags at the stop and could be easily adapted at future installations. Because this was a feasibility study it was deemed more appropriate to find the lower bound on the number of tags needed at a stop. Additional tags can only improve the operating characteristics of the system.

3.5.3 Results of Operational Testing

This section discusses the results of operational testing. The design is then reviewed in light of the results and possible design modifications suggested which can improve the reliability of the system before full implementation is considered.

3.5.3.1 Review of Hardware Components

Analysis of Antenna Placement

One of the properties of the RF antenna is that interrogation field generated by the antenna is omni-directional and cannot be directed. The current installation of the door antennas is such that panels attached to the antennas are near seating areas of the bus. This can allow a passenger who is carrying a tag and sitting in one of these seating areas to be unintentionally read when the doors are opened. In this case, the passenger is neither boarding nor alighting but merely sitting in a seat that is within the antenna interrogation field. Two solutions to this problem exist: an alternate placement of the door antennas or some form of shielding that can reduce the area of the interrogation field.

An alternate placement of the door antennas would involve changing from a side reading antenna panel to a antenna that was embedded around the inside of the bus doors. The tradeoff here is that the door of the bus contains metal. One of the weaknesses of the RF technology is the degradation of the field strength as the antenna gets closer to metal. It was for this reason that this placement was not considered in the original design. Further research into a suitable method of embedding the antenna into the door without experiencing the signal degradation from the metal might be beneficial.

Another solution might be to use the weakness described above as an advantage. It might be possible to shield the back of the antenna with a thin film of metal that keeps the interrogation field from extending in the wrong direction (i.e. in the direction of the seats on the bus).

Analysis of Tag Placement

The operational testing of the prototype also gave some insight into the placement of the tags located at the bus stop. The results indicated that a portion of the bus stop tags were not being read, approximately 4-6 stops out of 22. There are two explanations for this: not enough tags at the stop or degradation of the field strength of the underneath antenna.

The original design called for approximately four tags to be placed at each stop. Due to the previously discussed factors which can determine the actual stopping location of the bus, our conclusion is that neither the number, nor the previously discussed cross design described is sufficient to ensure a tag read at the stop regardless of whether or not the bus comes to a halt at the bus stop. Given the low cost of installing tags and the low cost of a tag, a better design might be 6-9 tags arranged in a line some distance before the bus stop sign, after the bus stop sign, and perhaps at the bus stop sign itself. Not only would this increase the likelihood of obtaining a bus stop tag read, but it would also increase the likelihood of obtaining the optimal grouping that was described in Section 3.3.2.

Analysis of Data Transmissions

This section describes the issues involved in the transmission of data from the bus to the remote computer for processing and translation. This particular implementation of the prototype involved a real time transmission of tag reads from the bus to the host computer via RF modem. While the RF modem has a fairly substantial range, approximately 5 miles, the modem is limited by an inability to transmit through areas populated with tall buildings. Operational tests indicated that transmissions were blocked by the Hospital Buildings along the selected test route. A more robust system of communication is needed. In actuality, the optimal method of data transmission would be end user specific. For instance, in the case that the system was being used as an automated fare system in conjunction with it's statistical capabilities, then real-time transmission of data would not be necessary. A storage device could be placed on board the bus which could record the transactions which occur during the day then the data could be downloaded at the end of the day into the database. Statistical information concerning the passengers and bus vehicle movements would still be available but not in real-time. If real-time route control, fare collection, and passenger/bus status information is required, then we recommend the use of a more reliable communication device, such as a cellular modem, as opposed to the RF modem. The tradeoff in the cost involved would be offset by the increase in the security and reliability of the transmission of data by the cellular modem.

Analysis of Antenna Reliability

The following tests were performed to examine the read reliability of the antennas which capture both passenger and bus transactions. The first set of tests involved the passing of tags through the interrogation field of both the front and rear door antenna in the form of a simulated rider entering or exiting the bus. The second set of tests involved both stationary testing and in-route testing of the antenna used to capture tags embedded in the roadway. The stationary tests involved placing tags under the bus within the interrogation field. The in-route tests involved driving the bus along the route to record tags embedded within the roadway. Two types of in-route tests were performed. The first involved the bus simulating the operation of the route by moving from bus stop to bus stop along the route. At each bus stop along the route the bus stopped, opened its doors, closed its doors, and the proceeded to the next stop. The second type of in-route test involved the bus simply driving by the bus stop at a normal operating speed.

1. Test 1 Passenger Reading Antenna
   a) Front Antenna
      i) Number of trials = 100
      ii) Percentage of correct reads = 100%
   b) Rear Antenna
      i) Number of trials = 100
      ii) Percentage of correct reads = 100%
2. Test 2 Road Tag Reading Antenna
   a) Stationary Test
      i) Number of trials = 100
      ii) Percentage of correct reads = 100%
   b) In-route Test 1
      i) Number of possible bus stop reads = 60
      ii) Percentage of correct bus stop reads = 95%
   c) In-route Test 2
      i) Average vehicle speed = 25 mph
         a) Number of possible bus stop reads = 40
         b) Percentage of correct bus stop reads = 50%
      ii) Average vehicle speed = 18 mph
         a) Number of possible bus stop reads = 20
         b) Percentage of correct bus stop reads = 55%

The rear antenna tests indicated that the field strength of the antenna was not sufficient to make accurate reads more than 12 inches from either the left or the right side antenna, but within the interogation field 100% accuracy was obtained. The results of road tag testing clearly indicate a need to improve the design of this antenna for those cases where the bus does not stop at the bus stop. The lack of reliability can be attributed to a need for improved tuning of the antenna and the need for additional tags within the roadway to capture a by-passing bus.

3.5.3.1 Review of Software Components

This section reviews the software components of the prototype and offers recommendations for further research that might improve the effectiveness of the software.

The chief cause of concern regarding the software components of the prototype is the translation procedure and it's inability to correctly calculate some of the statistics when a non-optimal grouping of the bus stop tag occurs. Ideally, the software would be able to recognize which of the cases described in Section 3.3.2 has occurred and then make the necessary adjustments to automatically ensure the statistics for all the stops are accurate. In the current design, this is not technically possible. In order for the software to be able to correctly recognize which of the cases has occurred it must be able to distinguish between arrival tag reads and departure tag reads. The greatest obstacle in being able to accomplish this task is that a bus may sit at a bus stop and open and close it's doors a few times before leaving. An example of this is given below:

s001,1,10:30:00	first stop arrival stop read
01	
p001,1,10:30:10	
00	close door id
01	open door id
p002,1,10:31:10	
00	
s002,1,10:30:00	possible second stop arrival read
	or possible departure read ????

In this particular situation, it is not clear what has occurred at stop 001. A number of possibilities exist.

The bus may have remained at stop 001 and merely closed it's door and reopened. In this case, the tag read of stop 002 is an arrival read, or the bus has left the stop and missed an arrival stop read at stop 002. The tag read of 002 would then be a departure tag read. In order to be able to correctly identify what has transpired, the following addition to the hardware would be necessary. For each opening and closing door ID, a odometer reading would be obtained. This would enable the software to distinguish between when a bus has remained at a stop and opened and closed it's doors and when it's moved to the next stop. An example of this modification is given below:

s001,1,10:30:00	first stop arrival stop read
01, 3500.00	open door id and odometer read
p001,1,10:30:10	
00, 3500.00	close door id and odometer read
01, 3500.50	open door id and odometer read
p002,1,10:31:10	
00, 3500.50	close door id and odometer read
s002,1,10:30:00	second stop departure stop read

With the addition of the odometer reading it becomes obvious that the bus has moved along the route and has not remained at the stop and reopened and closed it's doors. The software could use this information in order to distinguish what has exactly transpired along the route. In addition, the tags placed at the stop can be coded as to their location relative to the bus stop sign. This information can be used to distinguish between arrival and departure reads at a stop.

In addition to the translation procedure, the relational database model itself may be able to be improved with further research. The current implementation of a transit bus schedule into the SCHEDULE and LINK entities of the database might seem somewhat cumbersome. Further research into other forms of data models, such as an object-oriented model, might yield a model that represents the relationships of a bus transit schedule more intuitively.

3.6 OTHER IMPLEMENTATION ISSUES

Two other implementation issues explored in this study were passenger acceptance and cost. Passenger acceptance was measured via survey. The results reported here are part of a study directed by the principal investigator and performed by a fourth year engineering student to meet his senior thesis requirements within the Systems Engineering Department at the University of Virginia.

3.6.1 Passenger Acceptance

In order to measure the passenger acceptance of RF/ID technology, a team of undergraduate systems engineering students distributed a series of surveys to bus passengers at the University of Virginia and its surrounding community. Five surveys were used in the study: the Hereford Route Transit Survey, Passenger Acceptance Survey (Bookstore distribution), Passenger Acceptance Survey (Bus distribution), Automated Transit Pass Pre-Survey, and Automated Transit Pass Usage Survey. The survey results suggest that most transit passengers will not oppose the implementation of the radio frequency tags as bus passes, as long as the system meets the needs of the individual transit agency's passengers. We refer the reader to Weisenberg (15) for a more detailed discussion of these results.

3.6.1.1 Hereford Route Transit Survey

The purpose of the Hereford Route Transit Survey was to obtain the preliminary views of passengers concerning the radio frequency transit passes. Undergraduate systems engineering students distributed the survey to 131 passengers along the Hereford Special route of University Transit Service during November 1995. The preliminary survey indicated that most transit passengers would not oppose the automated transit database system. Only 26.5% responded with objections to the system. The most common problems stated were forgetting tags and privacy concerns associated with monitoring passenger trips. This survey served to provide possible questions for the remaining surveys and provided some initial support for passenger acceptance.

3.6.1.2 Passenger Acceptance Survey (Bookstore distribution)

The main objective of the Passenger Acceptance Survey was to forecast the likelihood of passengers accepting the automated system and to identify any widespread objections. The undergraduate engineering students distributed 455 surveys at the University of Virginia Bookstore on January 15-17, 1996. Subjects completed a one-page questionnaire while waiting in line to buy books. A total of 368 respondents were recorded. The results suggest that most individuals do not hold strong opinions supporting or opposing the automated transit database. As shown in Figure 12, three quarters of the sample think their use of transit services would not change with the implementation of this system, while roughly equal proportions of the remaining quarter predicted their transit use would increase or decrease. Figure 13, shows the predicted acceptance of a passenger rated on a 0-10 scale. This illustrates that the overall system acceptance follows a distribution skewed slightly toward the higher ratings.

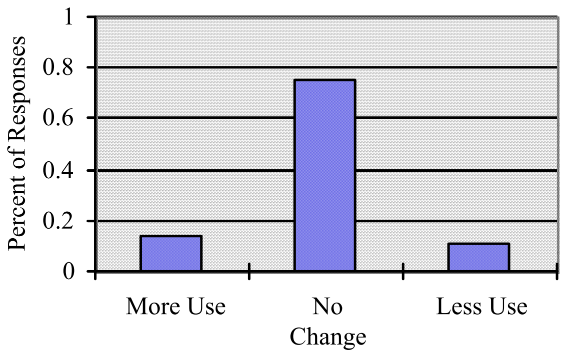

Figure 12 Change in transit use

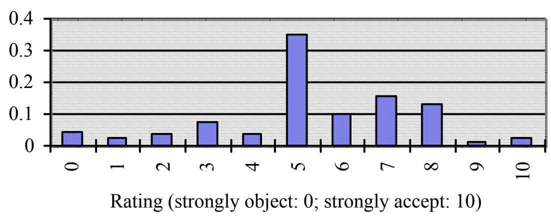

Figure 13 Passenger acceptance distribution

A C++ program was written to input responses from the three Passenger Acceptance Survey questions to predict whether a passenger would likely accept the system, may accept the system, or oppose the system. Figure 14 illustrates that for over half of the sample, acceptance cannot be determined. A larger proportion of surveyed individuals are likely to accept the system than reject it. The small percentage of people opposing the system is encouraging.

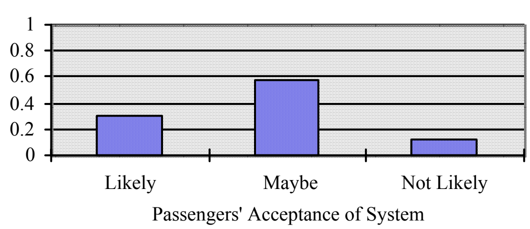

Figure 14 Passenger acceptance

The surveyed population indicated that the benefits of improved routing and scheduling and eliminating the need of physically displaying a bus pass were important, with over 40% replying that such benefits were “extremely important”. On the other hand, almost 60% viewed automated fare collection as “not important”. This could be attributed to the fact that the survey was distributed on the Grounds of the University of Virginia. The University Transit Service does not charge fares. Those not charged a fare would obviously not have a need for an automated fare collection system. When asked their concerns with the system, over half of the sample viewed the cost of the bus passes as “extremely important”. Approximately 45% of the sample indicated that they did not believe that privacy was an important issue with the system.

The main limitation in the surveys distributed at the University of Virginia Bookstore was that the majority of those surveyed were University of Virginia students, so the sample group was mostly of the same age and educational level. In order to gather a more unbiased sample, surveys were distributed to bus passengers in the Charlottesville community.

3.6.1.3 Passenger Acceptance Survey (Bus distribution)

This survey also aimed at gathering the public’s views on various issues. The student data collectors distributed 62 surveys at bus stops of Charlottesville Transit Service (Barracks Road Shopping Center) and University Transit Service (Scott Stadium), and on Charlottesville Transit Service buses (Route 7). These surveys, distributed in February 1996, included two additional questions to identify the passenger’s age and educational level. It was also noted whether the respondent was a UTS or CTS passenger. The results of this survey are limited because of non-response. Only 34 of the 62 distributed surveys were fully completed.

The overall results of the bus/bus stop surveys supported the results of the bookstore surveys. The results indicated that passengers consider improved routing and scheduling and the cost of a bus pass as important issues. Similarly, the results of both surveys indicate that privacy issues and the size or shape of the pass are not significant issues for passengers. Approximately three quarters of the sample stated that they do not anticipate the system changing their use of transit services. One of the most significant differences between UTS and CTS passenger responses involves the importance of automated fare collection, as displayed in Figure 15.

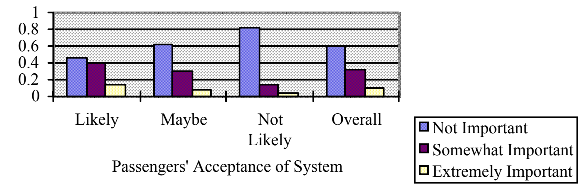

Figure 15 Importance of Automated Fare Collection

The graph illustrates that CTS passengers have a much higher need for an automated fare collection system. Since CTS passengers currently have to pay fares, they would benefit from an automated system that would eliminate the need for carrying the exact change necessary for a given trip each day. UTS passengers, on the other hand, do not need to pay fares and so would not gain any benefits from such a system. CTS passengers also worried more about boarding with no money left on an automated bus fare card. This would suggest a need for a system that could handle a credit or debit card. Those worried about not having money on their fare card could obtain a credit fare card while those who wished to pay for bus fares in advanced could obtain a debit fare card.

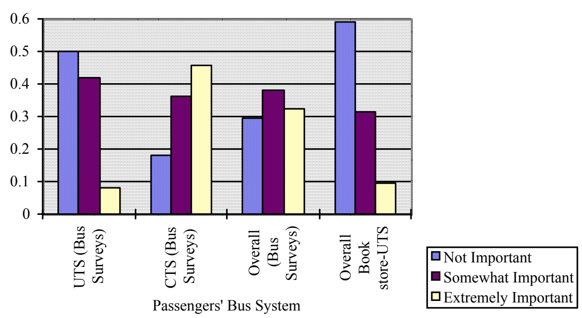

Figure 16 Predicted Passenger Acceptance UTS/CTS Surveys

Figure 16 displays various passenger groups' predicted acceptance of the system. The figure shows acceptance significantly higher for the bus survey sample as compared to the bookstore sample. For both UTS and CTS passengers, only a small percentage of the people would likely oppose the system. Unfortunately, more data would be required to draw significant conclusions regarding differences in passenger acceptance by age, educational level, and bus system use.

3.6.1.4 Automated Transit Pass Pre-Survey and Usage Survey

The main objective of the final two surveys was to determine if opinions of the system changed after carrying the radio frequency tags on a daily basis. Those University of Virginia students participating in the prototype database test completed the Pre-Survey upon receiving their tags. The participants agreed to complete and return the Usage Survey during the third week of February 1996.

The Automated Transit Pass Surveys reveal that passenger views of the automated bus passes do not change significantly after possessing and carrying the tags for several weeks. This supports the responses to the Passenger Acceptance Survey. The Usage Survey indicated that over 90% of the respondents accepted the system. This is about the same as previous responses. In addition, the Usage Survey confirmed the Passenger Acceptance Survey's conclusion that most passengers do not actively support or oppose the system, as no one responded that changes in their use of the Hereford Special bus were due to the automated passes. Over 90% of the group responding to the Usage Survey carried their automated pass all of the time. A large fraction of the sampled passengers were concerned with the inconvenience of replacing a broken pass and system failure. Thus, it would be important for a system to be capable of sending tags to people through the mail, or some other effortless method of distribution. Carrying the radio frequency devices allowed testers to notice the potential for automated fare collection. The importance of automated fare collection rose in the second survey. After having tested the bus passes, passengers seemed less willing to reveal personal attributes for an automated database. A large portion of the sample responded as preferring a key ring bus pass or credit card-shaped bus pass to the current shape of the tag. When asked to rate their preferences for contactless bus passes as fare cards compared with cash, tokens, magnetic stripe cards, and contact smart cards, passengers rated the contactless passes highest as displayed in Figure 17.

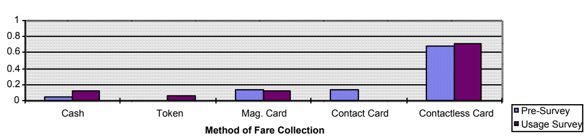

Figure 17 Preferred Method of Fare Collection

In response to open-ended questions, passengers liked the system's efficiency in that it eliminates the need to carry exact change for a bus fare. Passengers anticipated quicker boarding with the implementation of the system, and would appreciate cheaper fares since transit managers can minimize costs with optimized routes and schedules. Some listed drawbacks of the system were forgetting the pass, losing the pass, system failure, privacy, and potential for running out of money if the automated pass acts as a debit card. After testing the tags, people indicated that they liked not having to physically display the tag. People worried about the cost of the tag and whether cash could still be used to pay fares for people unfamiliar with a community's automated bus system.

3.6.1.5 Summary of Survey Results

The results of surveys suggest that no widespread opposition exists that would hinder implementation of the system. Most transit passengers do not have strong opinions supporting or opposing the system, and the largest concern with the system involved the cost of the bus pass. The survey analysis stressed the importance of understanding the needs of a transit company's passengers before implementing an automated data collection system. For example, users of the Charlottesville Transit Service, who pay fares, have a much higher demand for automated fare collection than users of University Transit Service, who do not need to pay or show identification to board. An automated transit database system could be beneficial to passengers if the implementation of the system were directed at meeting the needs of the individual transit agency's current passengers.

3.6.2 COST CONSIDERATIONS

Transit operator acceptance is dependent primarily on the value and convenience of automatically collected data in conjunction with the cost of collecting the data.

Current passenger data collection methods range from manual methods which use the bus driver to press a counter as passengers enter the bus to fully automated passenger counting (APC) systems, see for example Levinson (2). Automated or semi-automated fare collection systems also offer possibilities for collecting ridership data. A typical system will collect fares with an electronic fare box on each bus either through fare cards or coins. The fare collection device will count the number of passengers that ride based on the amount of fares collected. The fare collection device must be removed nightly and the data downloaded by a probe. Ridership reports can then be generated. These systems simply count the number of riders per day and can not associate passengers with a particular origin or destination. Thus, these systems do not approach the functionality available in an RF/ID based system. The costs associated with enhanced APC systems were reported in Levinson (2) and are repeated in Table 6 for comparative purposes.

TABLE 6 Estimated Cost Considerations in APC and RF/ID

APC/AVL Item	Cost	RF/ID Item	Cost
Counting Sensors	$1200-1700 per bus	ASGI Tag	$17-$25 per tag
Micro-processor	$1500 per bus	Decoder	$1500 per bus
Data retrieval unit	$4000-5000 per unit	Host computer	$3000
Systems display unit	$1500 per unit	High Speed modem	$600 per bus
Signposts	$300 per unit	RF antennas	$600 per bus
Signpost receiver	$1000 per unit	Installation	
Signpost receiving antenna	$200 per bus	Tag	$12 per tag
Installation		Hardware	24 hr per bus
Micro-processor	24 hr per bus		
Signposts	$300 per unit		

The functionality and costs as reported by Levinson (2) of these upgraded APC systems is very comparable to the functionality of the prototype developed in this project. The approximate costs on a per bus basis are $2,900 for the APC/AVL type system and $2,700 for the RF/ID system. Both systems require a method for retrieving and storing the data with similar costs if additional disk storage is considered for our system. The paper by Martin, Steane, and Mauceri (14) further suggests that APC's require a network of signpost transmitters spaced approximately 3-7 km apart along the route in order to monitor schedule adherence. According to Benn (4), transit operators report bus stop spacing in the non-central business district of 6-8 stops per mile (1.61 km). Assuming one signpost is required approximately every 3-7 km, this translates into approximately 12-35 stops per signpost. For a system the size of our test route 1-2 signposts would be required. If only 1 signpost is required the break even cost per tag including installation at $12/tag and 4 tags per stop is $6.18 which is infeasible at current manufacturing costs of approximately $13/tag; however, we feel that 12-15 stops per signpost is more realistic. With 12-15 stops per signpost the break even cost is $21.33-$14.67 which is feasible at current tag costs. Tag manufacturing costs will continue to decrease as higher volume demand occurs. In addition, better installation methods should be considered to cut the installation costs. The primary requirement for tags embedded in the roadway is to track bus movements. Other tracking technology could be considered such as global positioning systems (GPS); however, with these systems mechanisms would be still be necessary in order to bound the events that occur at a bus stop. For example, a sensor on board the bus would be necessary to detect that the bus was not in motion. A GPS reading could then taken to determine the time and location of the bus and then the doors could open to process passengers via RF/ID. With the reduction in costs of GPS, this scenario should be investigated.

TABLE 7 Break Even Cost Per Tag vs 1 Signpost @ $1600

	4 tags/stop		6 tags/stop	
# stops	not installed	installed	not installed	installed
12	$33.33	$21.33	$22.22	$10.22
15	$26.67	$14.67	$17.77	$5.77
22	$18.18	$6.18	$12.12	$0.12
27	$14.81	$2.81	$9.88	$-2.12
35	$11.42	$-0.58	$7.62	$-4.38

Although encouraging, these results are clearly preliminary. Implementation of RF/ID on a larger scale within a urban transit system would enable a better understanding of the cost/benefit tradeoff associated with RF/ID over a projected system life-cycle.

3.7 EXAMPLE USE IN ROUTE MANAGEMENT

Because the system captures the location of the bus, our system has the potential to provide near real-time information on the status of the entire transit system(busses and riders). In order to illustrate how the integration APC and AVL based on RF/ID can be exploited, we examined the use of the information stored in the database to control bus headway. The information reported here is part of a study directed by the principal investigator and performed by a graduate engineering student to meet his Masters thesis requirements, see reference (*16*).

3.7.1 Threshold Based Control Strategies

Headway is defined as the time or distance, from a fixed point, between the departure of one vehicle and the arrival of the next (*17*). The scheduled headway along a route indicates to passengers how often a bus arrives at a stop (i.e. every 10 minutes). Variations in observed headway along a route can cause buses servicing the route to "bunch" which can decrease system performance. Bunching can contribute to increases in overall passenger waiting times at a stop in addition to increasing overall passenger waiting times along the route. These increases lessen the reliability of the service in the eyes of the transit user and generally lead to ill will between the transit user and operator. This can result in decreases in ridership. Reductions in headway variation can be translated directly into decreased passenger waiting and transit times. In order to reduce headway variation along a route, methods such as threshold based control can be used. A threshold-based strategy involves identifying a certain value (x_0), known as the threshold value, at a particular bus stop along the route which acts as a control point. If the observed headway between an incoming bus and the previous bus is less than (x_0), then the incoming bus is held up to the threshold value. If the observed headway is greater than the threshold value the bus is not held (*18*).

In order to implement this strategy, an optimal control stop and optimal threshold value must be identified. We refer the interested reader to references (*19*) and (*20*) for more information on determining the threshold value and control point. The optimal control point is usually found to be located at stops along the route immediately before stops where a large number of passengers board. This is due to the fact that stopping at the control point reduces headway variation, which as was mentioned above, reduces passenger waiting times. By locating the control point prior to where the largest number of passengers board, more passengers are allowed to appreciate the reduction in headway variation. Choosing the optimal threshold value involves a tradeoff between delaying passengers at a control point and delaying passengers at stops downstream of the control point. As the threshold value approaches the scheduled headway, the delay to passengers downstream of the control point decreases while the delay to passengers at the control point increases. This is because as the threshold gets closer to the scheduled headway, the reduction in headway variation is greater. This allows a reduction in the waiting time of the passengers downstream of the control point; however, because the threshold value is increasing, buses are waiting longer at the control point which causes longer delays to passengers onboard the bus.

Research has also determined the route characteristics that warrant threshold-based holding strategies. Routes that have sufficiently short headways, less than 10 minutes, have shown performance improvements from threshold-based holding strategies (*20*), whereas Turnquist(*21*) showed routes that had sufficiently large headways benefited more from other holding strategies such as schedule-based holding strategies. Passenger profiles are also relevant in determining which routes may benefit from threshold-based holding strategies. Routes that have either passengers boarding along the middle of the route and alighting at the end or boarding and alighting uniformly along the route have shown the most significant reductions in headway variation and passenger waiting times (19).

3.7.2 Prototype Support for Threshold Based Control

Abkowitz and Engelstein (22) gave the following methodology to determine which control strategy for headway variation control should be applied to any given route:

1. Determination of mean running time
2. Determination of running-time variation
3. Determination of headway variation
4. Determination of passenger wait time
5. Determination of optimal control strategy

The prototype is able to calculate values for the mean running times, running time variations, and headway variation from actual observed data captured along the route. This should be contrasted with the use of empirical or simulation based regression equations to determine these parameters as discussed in reference (21). RF/ID enables the actual collection of the necessary data in a timely manner. The necessary values can all be found in the STAT relation of the relational database. In order to complete step 4 of the methodology, passenger waiting times are required. Some researchers, see Barnett (24), have assumed purely random passenger arrivals, while others, Turnquist (25). considered random and non-random passenger arrivals. Other studies, Abkowitz and Tozzi (18) have indicated that the passenger random arrival assumption is valid only for those routes with short headways, around ten minutes or less. Research assuming random passenger arrivals has yielded the following equation for the expected waiting time for a passenger until a bus arrives to a stop, see reference (23):

$$E[W] = \frac{E[H]}{2} + \frac{Var[H]}{2E[H]}$$

where

E[W] = expected wait time
Var(H) = headway variation
E[H] = expected headway

Thus, under the assumption of random arrivals the passenger waiting time at a stop is dependent upon the expected headway and the headway variation which are easily computed from the database. In addition to passenger waiting times step 4 requires data on the number of passengers on the bus and boarding/alighting along the route, all of which can be computed from the database. The relevant piece of data that is needed to determine threshold values and control points is an accurate description of the flow of passengers through the route. The prototype is able to obtain accurate descriptions of the passenger boarding/alighting patterns along the route and can also show how they might change over different time periods. The boarding/alighting patterns can be generated from the OD relation of the relational database and from the BUS TRANSACTION relation. The OD relation tracks the number of passengers that board at a specific location and alight at another location. The BUS TRANSACTION relation shows how many passengers board or alight at a specific bus stop. It also shows how many passengers are on board the bus at each stop. With this information, a more accurate boarding/alighting pattern can be generated by the prototype which would give a transit manager more input in determining where possible locations for a control point may be for a certain route.

3.7.3 Prototype Support For Management Techniques

In order to analyze the effects of threshold values and control points, a simulation model of the route is extremely useful. In order to develop a simulation model the information listed below is needed:

1. run time data
   a) run time averages
   b) run time variation
   c) coefficient of variation
2. passenger boarding and alighting pattern
   a) number of passengers boarding at each stop
   b) number of passengers alighting at each stop
3. number of passengers traveling along the route
4. number of buses along the route

Since the prototype obtains average run times and run time variation from stop to stop, this information could be used to develop a more accurate distribution that drives the buses along the route. A separate distribution can be generated for the run times at each particular stop. The number of buses used along the route is also needed. Both the number of buses and the run time information can be directly obtained from the relational database. The transit manager would merely have to indicate which route to simulate and the database could process the necessary queries to obtain the number of buses and run time information. The passenger arrival process can be generated from general passenger count data that again can be obtained directly from the database. In fact, the database supports the ability to generate different arrival processes for different time periods. This passenger information can all be obtained from the PASSENGER TRANSACTION entity of the database, while all the stops located on the route chosen by the transit manager can be obtained from the SCHEDULE entity. The boarding and alighting pattern can be generated from the OD and BUS TRANSACTION entities which could be used in generating a more accurate arrival process. With extra effort the entire simulation process could be automated. In addition, for those routes where threshold based control strategies are inappropriate, the database can support other scheduling and design changes.

4. PLANS FOR IMPLEMENTATION

The prototype indicates the feasibility of using RF/ID in transit; however, additional enhancements and functionality would be required to make the system fully functional in a transit environment. Based on the results contained in this report, we consider the following enhancements as necessary to justify full implementation in practice:

1. Hardware modifications to improve antenna read reliability when the bus does not stop at the bus stop.
2. Investigation of integrating global positioning systems instead of embedded road tags.
3. Integration of automated fare collection
   a) This is necessary because the system can not be cost justified if used solely as a data collection tool. Other studies have indicated that RF/ID is feasible for fare collection, see Labell (1). The prototype integrates APC and AVL information. Thus, because transit service providers must already collect fares, they would receive the additional benefits associated with the APC/AVL information collected. The best way to integrate fare collection would be to partner with a company which already specializes in this arena.
4. Client-server database implementation
   a) Fare collection would require design changes to the database to support fare transactions, disputes, billing, etc. In addition, for larger implementations, we suggest a (Structured Query Language) SQL database server such as Sybase or Oracle with a client front end built using Clarion, PowerBuilder or some other rapid application development tool. The current design can be easily adapted to the client-server model of computing.
5. Improved data communication between bus and host system
   a) To gain the full benefits of the information collected, we feel that real-time data collection is necessary. As previously discussed, current RF modems do not appear to be sufficient for the data transmission reliability requirements within an urban environment. Cellular modems should be investigated and integrated into the client-server model proposed above.

Although not required for deployment in transit practice, the following additional functionality would be important in distinguishing the system from competitors:

1. Enhanced on-line and off-line reporting as suggested in Levinson (2). For example, many additional management reports can easily be developed based on the current database model, e.g. a Section 15 report. In addition, a real-time display of bus and passenger status in the system would be beneficial.
2. Integration of additional automatic vehicle maintenance and monitoring functions.
3. Integration of automated statistical and simulation analysis tools. These tools would use the database to perform trend analysis and enable "what if" analysis to be performed for predicting the performance of various service changes and examining route control strategies.

In addition, we recommend a more detailed examination of the cost/benefits of RF/ID technology when considered over a projected system life-cycle. This is a difficult task because the dollar benefits gained due to the additional information from our system can not be easily assessed in advance. In other words, the information system enables systematic transit service changes which may significantly reduce operating costs; however, these systematic changes in transit service can not be assumed in advance for various transit properties. This cost analysis would also enable a better understanding of how to reduce the cost of the system, e.g. tag cost.

ASGI has identified potential transit authorities with an interest in perfecting more cost effective, near real-time methods of tracking transit users. ASGI intends to further enhance and then market the final product(s) through their existing 14 distributors and possible 1300 dealers selling RF/ID products. Negotiations are currently in progress to form a partnership between ASGI, the University of Virginia, a transit property, a transit equipment provider, and potential governmental agencies to perform a follow up study which incorporates the above mentioned enhancements and additional functional capabilities. This partnership would take the system from the prototype stage to the operational product stage while performing tests in an operational setting.

5. CONCLUSIONS

The integration of automatic passenger counting and automatic vehicle location systems is a necessary step in meeting the goals set out by the U. S. Department of Transportation's Operation Time Saver. Radio frequency identification of passengers via RF/ID bus passes will enable the integration of APC and AVL to achieve:

- real-time status information for transit users
- automated fare collection
- transit user convenience and marketing
- data collection regarding transit users

Because of the real time nature of the data collection, the potential exists for large volumes of data to be collected. This prototype was specifically designed to facilitate the extraction of information from the volumes of data collected. The prototype has been tested under field site conditions within the University Transit System. This project represents a contribution to transit practice by exploiting the integration of APC and AVL. The results of the project include:

1. A prototype RF/ID based automatic data collection system for transit operations which has the potential to compete with current automated passenger counting and automatic vehicle location systems by integrating their functions.

2. Survey results indicating that 45% of those sampled did not believe that privacy was an important issue involving this system. In addition, more than half of those surveyed did not have a strong opinion supporting or opposing the system. Passenger acceptance increased for those passengers concerned about fare collection. Although preliminary in scope, the survey results suggest that passengers will accept the technology especially if the added convenience is explained and privacy is ensured.

3. A RF/ID based automatic database collection software prototype system for monitoring transit system performance and supporting route control analysis.

In order to continue to exploit the integration of APC and AVL, we recommend that the prototype be enhanced to include improved hardware reliability, automated fare collection, and client-server database operations.

6. INVESTIGATOR PROFILE

Manuel D. Rossetti, Assistant Professor
Department of Systems Engineering
University of Virginia
Thornton Hall
Charlottesville, VA 22903

Phone: (804) 924-4438
Fax: (804) 982-2972
E-mail: mr2m@virginia.edu

EDUCATION: Dr. Rossetti received his MS and Ph.D. degrees in Industrial and Systems Engineering from The Ohio State University in 1988 and 1992 respectively, and his BS in Industrial Engineering from the University of Cincinnati in 1985.

Experience: Dr. Rossetti joined the faculty of the Department of Systems Engineering in 1993. He is responsible for research and teaching in the areas of automatic data collection systems, manufacturing systems, transportation systems, discrete event simulation, stochastic modeling and vehicle routing and scheduling.

SELECTED PUBLICATIONS:

1. Rossetti M. D. and Clark G. M.(1995) “Estimating the Expected Sojourn Times for an Exiting Markov Renewal Process,” *Communications in Statistics: Theory and Methods*, Vol. 24, No. 2, pp. 553-579.

2. Benton W. C. and Rossetti M. D.(1992) “The Vehicle Scheduling Problem with Intermittent Customer Demands,” *Computers and Operations Research*, Vol. 19, No. 6, pp. 521-531.

3. Rossetti, M. D. and Delaney, P. J. (1995) “Control of Initialization Bias in Queueing Simulations Using Queueing Approximations,” in *The Proceedings of the 1995 Winter Simulation Conference*.

4. Rossetti M. D. and Stanford, J. A. (1995) “Group Sequencing a PCB Assembly System Via an Expected Sequence Dependent Heuristic”, Dept. of Systems Engineering, University of Virginia, Working Paper, SYS-MDR-1-1995, submitted for publication in *International Journal of Flexible Manufacturing Systems.*

5. Rossetti M. D. and Clark G. M.(1994) "Estimating Operation Times From Arrival and Departure Events", Dept. of Systems Engineering, University of Virginia, Working Paper, SYS-MDR-4-1994, submitted for publication in *International Journal of Flexible Manufacturing Systems*.

7. REFERENCES

1. Labell, Lawerence N., Carol P. Schweiger, and Mary Kihl, Advanced Public Transportation Systems: The State of the Art Update '92, (U.S. Department of Transportation, Research Special Programs Administration, John A. Volpe National Transportation Systems Center, Kendall Square, Cambridge MA 02142, April 1992) Report No. DOT-VNTSC-FTA-92-3.

2. Levinson, H. S. *Supervision Strategies for Improved Reliability of Bus Routes*, Transit Cooperative Research Program, Synthesis of Transit Practice 15, Transportation Research Board, National Research Council, 2101 Constitution Avenue, N. W., Washington D.C. 20418, 1991.

3. Transportation Research Board Project Announcement TRANSIT-IDEA, Transit Cooperative Research Program (TCRP) Innovations Deserving Exploratory Analysis (IDEA). May 1993.

4. Benn, H. *Bus Route Evaluation Standards*, Transit Cooperative Research Program, Synthesis of Transit Practice 10, Transportation Research Board, National Research Council, 2101 Constitution Avenue, N. W., Washington D.C. 20418, 1995.

5. Motorola, Inc. "Advance Information: RTLU Remote LONWORKS Telemetry Unit", November 1995.

6. Date, C. J. *An Introduction to Database Systems*, Fourth Edition, Addison Wesley, 1986.

7. Hoffer, J. A. and McFadden, F. R. *Modern Database Management*, Benjamin/Cummings Publishing, New York, 1994.

8. Burch, J. G. *Systems Analysis, Design, and Implementation*, Boyd & Fraser Publishing Company, South-Western Publishing Company, Boston, MA, 02116, 1992.

9. Barnes, K. E. and Urbanik, T. "Automated Transit Ridership Data Collection Software Development and User's Manual", Texas Transportation Institute, Technical Report 1087-3F, September 1990.

10. Adamski, A. (1992) “Probabilistic Models of Passenger Service Processes at Bus Stops”, *Transportation Research - B*, Vol. 26B, No. 4, pp. 253-259.

11. Koffman, R. "A Simulation Study of Alternative Real-Time Bus Headway Control Strategies", Transportation Research Board, *Transportation Research Record 663*, pp. 41-46, 1978.

12. Motter, J. "Database Validation Through Simulation", B.Sc. Thesis, Systems Engineering, School of Engineering and Applied Science, University of Virginia, 1996.

13. Chitre, L. "Automatic Data Collection Information System for Transit System Operations", B.Sc. Thesis, Systems Engineering, School of Engineering and Applied Science, University of Virginia, 1996.

14. Martin, H. A., Steane, A. H., Mauceri, V. C. "The Use of Automatic Passenger Counting Data to Verify Schedule Adherence", in Computer-Aided Transit Scheduling, *Proceedings of the Fifth International Workshop on Computer-Aided Scheduling of Public Transport*, Montreal, Canada, August 19-23, M. Desrochers and J.-M. Rousseau, Editors, 1990.

15. Weisenberg, R. "Passenger Acceptance of an Automated Transit Information System", B.Sc. Thesis, Systems Engineering, School of Engineering and Applied Science, University of Virginia, 1996.

16. Turitto, T. " ", M.Sc. Thesis, Systems Engineering, School of Engineering and Applied Science, University of Virginia, 1996.

17. Haefner, J. Introduction to Transportation Systems, Holt, Rinehart and Winston, New York 1986.

18. Abkowitz, M., and T. Tozzi “Research Contributions to Managing Transit Service Reliability” *Journal. of Advanced Transportation*., 21(1), pp. 47-65, 1987.

19. Abkowitz, M., and T. Tozzi “Transit Route Characteristics and Headway-Based Reliability Control” *Transportation Research Record*, No. 1078, pp. 11-16, 1986.

20. Abkowitz, M., and M. Lepofsky “Implementing Headway-Based Reliability Control on Transit Routes,” American Society of Civil Engineers, *Journal of Transportation. Engineering,* Vol. 116, No. 7, pp. 49-63, 1990.

21. Turnquist, M.A. “Strategies for Improving Reliability of Bus Transit Service,” *Transportation Research Record,* No. 818, pp. 25-29, 1981.

22. Abkowitz, M., and I. Engelstein “Methods for Maintaining Transit Service Regularity.” *Transportation. Research Record*, No. 961, pp. 1-8, 1984.

23. Abkowitz, M., Eiger A., and I. Engelstein "Optimal Control of Headway Variation on Transit Routes." *Journal of Advanced Transportation*., 20(1), 73-88, 1986.

24. Barnett, A. "On Controlling Randomness in Transit Operations." *Transportation Science*. 8,2, pp. 102-116, 1974.

25. Turnquist, M.A. "A Model for Investigation the Effects of Service Frequency and Reliability on Bus Passenger Waiting Time." *Transportation Research Record.* 663, pp. 70-73, 1978.