A TASK DECOMPOSITION MODEL FOR DISPATCHERS IN DYNAMIC SCHEDULING OF DEMAND RESPONSIVE TRANSIT SYSTEMS

Mansour Rahimi
Maged Dessouky
Ioannis Gounaris
Greg Placencia
Merrill Weidner

University of Southern California
Department of Industrial and Systems Engineering
Los Angeles, CA 90089-0193

June, 2000

METRANS Transportation Center
USC / CSULB

DISCLAIMER

ABSTRACT

Since the passage of ADA, the demand for paratransit service is steadily increasing. Paratransit companies are relying on computer automation to streamline dispatch operations, increase productivity and reduce operator stress and error. Little research has been performed to understand the task complexity of dispatchers interacting with their computer systems.

This research was conducted to provide an in-depth understanding of the task complexities and requirements for paratransit dispatchers. To achieve this, we organized, modeled, and analyzed a complete paratransit dispatching task sequence for a Los Angeles area paratransit service provider. Extensive field observations, video recordings, and expert dispatcher interviews were conducted to sketch out a dispatcher's task sequence model during a high workload period. Using Hierarchical Task Analysis (HTA), this skeleton model was further refined into a comprehensive decision hierarchy (decision tree). The HTA was further extended to capture the sequence of activities between two dispatchers, a paratransit van dispatcher and a lead dispatcher, in the same operation room. The results of this analysis have shown that the dispatchers undergo intensive and complex cognitive processes. Their task performance seems to be heavily influenced by the type of software interface they use. We found HTA to be a useful tool to model these interactions. The sequential decision tree format of HTA also shows promise for training system design. In particular, we recommend the use of this model for part-task training for entry-level dispatchers.

In addition to task modeling, we further analyzed the design of the software interfaces used in this operation from a human factors standpoint: a DOS-based screen design and a Windows-based graphical user interface design. The DOS-based design had the advantage of information simplicity. However, this design produced long information scanning and navigation time, potentially long learning curve and screen design inconsistency. The Windows-based system had the advantage of a more natural spatial and iconic representation, well-designed popup menus and cursor sensitive information display. The most significant disadvantage of this design was a very long system lag time (e.g., it sometimes took up to 30 seconds for the system to respond to an input). This problem alone was the reason why the dispatchers did not use this system during high-demand periods. Other disadvantages were high-density clutter at low zoom level, layout

inconsistency between zooms, inappropriate color-coding and no direct (active) driver interaction with the system in case of GPS failure. We believe that dispatch software developers should be encouraged to look into a more natural and quick response software interfaces. In general, we recommend a redesign of the current system based on the principles of user-centered interface design.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

DISCLOSURE

Funding source:

METRANS Transportation Center, USC/CSULB

Names of principal investigators:

Maged Dessouky
Mansour Rahimi

Total dollar amount:

$50,000

Title of project:

A Task Decomposition Model for Dispatchers in Dynamic Scheduling of Demand Responsive Transit Systems

Relationship to report

The report was written by the Principal Investigators.

Subcontract statement:

Project was funded in entirety under this contract to California Department of Transportation.

ACKNOWLEDGEMENT

This research was entirely supported by Metrans Center and its funding agencies: the Department of Transportation and California Department of Transportation. The research team greatly appreciates the continuous support of Access Services, Inc. (ASI) for providing both technical expertise and data in the first stage of this research. In particular, we thank Richard DeRock and Lance Millar from ASI in discussing with us many of the performance criteria issues in current transit systems. Also, we would not have had the opportunity for an in-depth analysis without the full cooperation of the management and technical members of their staff from our collaborating paratransit service provider (herein called "ABC" for confidentiality), and tens of hours of individual discussions with their most experienced dispatchers. We also thank Douglas Wellman, manager of USC's Television Production Facilities for the use of their post-production and recording equipment. Finally, we wish to thank Antonia Boadi, a graduate student, for aiding us in the first phase of this research.

INTRODUCTION

The growth of personalized public transit and demand responsive transit began in the late 1970s and early 1980s with large demonstration projects developed in Rochester, NY and Santa Clara County, CA. Today, the paratransit market is around a $1 billion industry. In Los Angeles County alone more than 5000 vans and 4200 cabs provide service, generating about 8 million trips per year.

The current expansion of the paratransit industry would not have been as significant if it was not for the passage of The Americans with Disabilities Act of 1990 (ADA). This act requires fixed-route transit systems to provide complementary paratransit services for disabled persons within three-quarters of a mile of a transit route. The ADA required all agencies to: (a) provide accessibility (e.g., wheelchair) to all new and renovated transit vehicles, and (b) offer complementary paratransit service to disabled individuals (Dempsey, 1991). ADA requirements combined with the funding opportunities in Intermodal Surface Transportation and Efficiency Act (ISTEA of 1991) and Transportation Equity Act for the 21st Century (TEA-21 of 1998) have created a new demand for paratransit services.

Due to the lack of advanced communication and tracking technologies, the early systems tended to operate as advanced reservation systems with some service providers requiring users to place a reservation at least one day in advance of their travel. As reported in the Transit Cooperative Research Program Report #18 (Lave et al, 1996), this mode of operation has been associated with much lower service productivity. In particular, the early systems failed to meet expectations due to low demand requests and deficiency in communication and computer technology to effectively manage such systems (Lave et al, 1996). Despite these problems, their use has been expanding since they ration capacity easily and are much less complex to implement than on-line real-time reservation systems. However, with the introduction of Intelligent Transportation System technologies (ITS) such as automatic vehicle location (AVL) devices, geographic information system (GIS), and mobile data terminals (MDTs), such a system is expected to become more complex to use and maintain.

Information Technology Implementation

An issue essential to the application of advanced information technology in transit services is the effectiveness of automation in improving the process of dispatching and scheduling of demand responsive transportation. System productivity is often measured by the ability of the dispatchers to match vehicles to trips within a very short period of time. This is a much more complex problem than taxi and airport van services due to the rideshare nature and multiple origin/destination of these trips. In this type of service, trip destinations must also be acquired by the call-takers and a more complex scheduling algorithm is needed since multiple trips must be scheduled in a logical sequence for specific vehicles.

For paratransit services we have the added complexity of scheduling each trip according to the special needs of the disabled customer. Now, a critical question is whether such a technology can be implemented efficiently and productively. Even if such a technology is easily achievable at the current technological level, can the operators use such a system within reasonable working conditions?

Paratransit Computerization

Before we address the issues related to paratransit computerization, we need to explain the elements of information flow in a dispatching environment (for a detail information flow diagram, see Figure 1, adopted from Lave et al, 1996). Computer dispatching involves the use of digital communication systems to connect the dispatching centers to the vehicle drivers. Incoming calls are answered by a group of call-takers and the pick-up addresses (and other passenger information) are inputted into the computer databases. In an automated dispatching system, a dispatch algorithm runs on the background and determines the vehicle that should service the call (usually the vehicle closest to the pick-up address in a posted zone). The information is then transmitted to the assigned vehicle (through an in-vehicle data terminal) with a precise sequence of trips assigned to this vehicle. The driver then accepts the trip and the dispatcher submits the rest of the information to the driver. The driver then proceeds to service the trips in the assigned sequence. All transactions are then kept in the computer databases for record keeping, management reporting, accounting, and performance analysis/reports.

In general, the non-scheduling aspects of paratransit operations (client certification, records and agency billing, accounting, etc.) have already been fully automated. However, the scheduling aspects of dispatching are not easily automated. Most paratransit service providers are using semi-automated or computer-assisted scheduling. In these systems, incoming calls (trip requests) are categorized into “steady” (e.g., reservations for the next day) and “ready” (e.g., real-time or within the next few hours). The call-takers check the passenger’s ADA eligibility and enter the information into the main computer. Then, the scheduling tasks begin. Usually, the night before, an experienced dispatcher develops a “skeleton” route for the next day, based solely on the steady passenger trip requests. Drivers, vehicles, and passengers are grouped together for the most efficient route. General sequence of vehicles and trips are grouped according to time of day, day of week, O/D, and special passenger needs. The vehicle path is optimized for minimum time and distance subject to vehicle capacity limitations. During the day, however, additional requests are serviced on a real-time basis. In a fully automated scheduling system, these additional trips are inserted and the schedule is re-optimized until a schedule of pick-ups and dropoffs is re-built. In a semi-automated system, this re-optimization is accomplished using the cognitive capabilities of the dispatcher on a real-time basis. The latter was the case in our analysis documented in this report. It is hoped that the advances in intelligent dispatching systems will gain increasing importance in the future.

In the semi-automated system, the interaction of the dispatcher with the driver is also important in a real-time scheduling process. For example, each driver is given the skeleton route at the beginning of his/her shift. This pre-assigned sheet includes sequence, pick-up times, and O/D addresses for each passenger. It also includes the estimated travel time and the minimum distance path for each trip. The use of ITS technologies has been expected to make the rest of this process more efficient and easier to use. The same pre-assigned route is used by the computer for the real-time "ready" trip insertions. The computer knows the exact route and the location of each vehicle in the fleet (using AVL technology). All new trips are inserted at the best point in the route sequence. The new schedule is communicated to the driver, who may begin a new trip, or deviate from the current trip and pick-up the new passenger. In practice, however, we have seen most paratransit operations use semi-automated dispatching approach. The literature seems to favor increased automation in paratransit operations. Stone et al. (1994) report that the level of satisfaction with scheduling software ranges from poor to excellent with the majority of the agencies surveyed reporting good. Also, they report that operators expect their satisfaction index to increase as the service providers fine-tune and customize their new software applications.

Paratransit Scheduling Software

The increase in automation has caused a recent shift to a more on-line reservation system. In terms of more efficient scheduling routines, Dessouky and Adam (1998) developed a real-time scheduling heuristic for paratransit services. Results of a simulation analysis using data provided by paratransit service providers in Los Angeles County show that this new heuristic outperforms dispatching rules currently used by the providers in terms of improving ridesharing and on-time performance. Despite the advantages of these algorithms, dispatchers in this rapidly growing industry lack sufficient knowledge of the elements of the complex scheduling process to perform their functions effectively in real-time. The increasing number of complex dispatching software further exacerbates this problem. Most of the current packages offer hundreds of input fields that dispatchers must understand, memorize and navigate under normal and time constrained situations. This problem is so widespread in the industry that the National Transit Institute has issued a Request for Proposal (RFP #9-03-04-01) to develop a new training system on Paratransit Scheduling and Dispatching Fundamentals. As stated in the RFP, "representatives from the paratransit industry cited chronic and widespread deficiencies in the capabilities of schedulers and dispatchers. Specifically, schedulers and dispatchers often have insufficient knowledge and insight to react effectively to situations. For example, they may not have the skills to make an efficient recovery plan when there are delays or deviations from the original operating plan." We suspect that these problems may cost hundreds of thousands of dollars for the larger consolidated regional paratransit service providers. These findings are consistent with our discussions with the paratransit representatives in Los Angeles County where similar problems were identified.

In general, two classes of employees are directly affected by these problems. The following is a sample of problems reported daily:

Dispatchers/Schedulers – not sharing trips, sending vehicles outside of desired routes, misinterpreting mobility device requirements (e.g., sending a cab to pick up a wheelchair customer).

Telephone Operators (including call-takers) – Incorrectly identifying addresses, not following routes properly, incorrect input of information into database.

These types of errors reduce productivity, cause late pick-up and dropoff, and result in excessive deadhead in mileage.

To address these problems, this research has developed a hierarchical task activity model of dispatchers in dynamic scheduling of demand responsive systems. This research has been coordinated with Access Services Inc. (ASI). As mandated by California State Senate Bill 826, ASI is the only designated consolidated transportation service agency for Los Angeles County, and its focus is to coordinate paratransit service within the county (the largest of its kind in this country). ASI has given its full support by providing access to the appropriate service providers. We conducted initial interviews with several service providers within the region. Based on these interviews and recommendation from ASI, we selected one provider with a typical (yet a technologically complex) dispatching system to conduct a complete analysis of their dispatching operations. For confidentiality purposes, the company's name will be referred to as "ABC."

PROJECT OBJECTIVES

Most of the research on scheduling demand responsive transit systems have focused on the algorithms, neglecting the human element. The objective of this project is to address this gap in research. The goal is to develop a task analytic model of a dispatcher in the scheduling process, while interacting with their computer systems. All elements of the human-computer interaction (HCI) are identified and weaknesses and strengths of the currently used system are evaluated using HCI principles. Therefore, we designed this research for a better understanding of the specific functions of dispatchers interacting with their computers, to develop a methodology to capture dispatching tasks/decisions in a formalized hierarchy (e.g., task/decision network), and to describe these tasks in terms of "bottlenecks" for future user-interface design The practical application of this research is to provide ASI and other paratransit agencies with an evaluation approach, which assists future dispatcher training and software selection.

RESEARCH METHODOLOGY

ASI has assisted the researchers in arranging the site visits to a number of companies. They have also supported this research by providing access to relevant data. In each region, paratransit service is contracted to a private operator. It is important to note that ADA paratransit in Los Angeles County is a coordinated effort of the cities. Currently, local cities continue to provide about 3.8 million annual paratransit trips throughout the county. ASI fills in the gaps and allows individuals to travel across Los Angeles County and the surrounding counties within the service area of Los Angeles County. ASI currently has 37,000 registered clients generating around 5300 trips per day. ASI is planning to install GPS receivers and mobile data terminals in all of their vehicles.

ASI divides the Los Angeles County Area into eight regions. Each region is serviced by a different provider, each of which uses one of several commercially developed computer automated dispatch (CAD) software. The pick-up location determines which provider will transport the client to the requested destination. Around 20% of the requests require delivery outside the region, and if a return trip is required, the passenger will need to make two reservations, one with each provider, since there is a strict no waiting policy after dropoff. The table below identifies the service providers currently operating in each region.

Table 1. The Current Operating Companies in Los Angeles County (1999)

Region	Name	Provider	CAD Software
A	East San Gabriel Valley	San Gabriel Transit	DDS[1]
B	San Gabriel Valley	San Gabriel Transit	DDS[1]
C,D Steady	Southern	MV Transit	PASS
C,D Ready	Southern	Independent Taxi Operations	DDS
E	Greater Los Angeles	Community Joint Venture Partners	MADS
F	San Fernando Valley	United Independent Taxi	DDS
G	Antelope Valley	Antelope Valley Transit Authority	PASS
H	Santa Clarita Valley	ATC/VANCCOM	PASS

[1]propriety with ridesharing

We remark that the service providers for regions C and D depend on the demand type. For this region, a steady demand is defined as a recurrent trip (origin/destination) at least once a week for a three months period. A ready demand is the on-demand portion of the requests.

As the above table shows, the software packages currently being used by the different providers in Los Angeles County include Trapeze PASS, Digital Dispatch System (DDS), and Motorola Automated Dispatch System (MADS). All of these provide automate billing, reservation, scheduling, and maintenance. Most of these systems have over a hundred input fields. We also note that this industry is changing very rapidly due to the changes in the marketplace and ITS implementation. Most of the packages have an interface in place to accommodate automatic vehicle location (AVL), and global positioning systems (GPS), and In-Vehicle Terminals. Each software package also offers numerous amounts of features and capabilities. These features have been grouped in seven main categories in Table 2.

Table 2. Main Features and Descriptions for a Typical Paratransit Software

User Management	Fields that are related to the overall management of the service (e.g., whether an individual is ADA eligible)
Reservation	Fields that handle the passenger pick-up reservation process (made in advance or in real-time)
Scheduling	Fields that are related to vehicle scheduling to determine passenger pick-up or dropoff times
Software Features	Fields that describe the operating characteristics of the software such as its security and networking capabilities
Vehicle/Driver Data	Fields that contain records and information regarding the drivers and vehicles
Statistical Reports	Fields that contain statistical results regarding system performance
External Modules	Fields that describe the hardware which can interact with the paratransit software

We will now explain how the following effort has provided us with a generic operator-task-activity model of the dispatchers in a dynamic scheduling system.

Identifying Dispatcher's Activities Using Hierarchical Task Decomposition

Our way of approaching the decomposition of the dispatching task is to see them as an arrangement of less complex sub-tasks, which themselves may comprise a number of interactive elements. We have adopted Hierarchical Task Analysis (HTA) (described first by Annett et al, 1971 and further refined by Shepherd, 1993) which has been used extensively in the design and evaluation of computer interactive systems. HTA is a method of considerable flexibility (for a description of different modeling efforts related to HTA, see Diaper, 1989; Kirwan and Ainsworth, 1992; Olson and Olson, 1990). We were particularly interested in describing the dispatcher's tasks in terms of their "goals" that they need to attain (we assume that the goals of the dispatchers are in congruence with those of the task assigned to them during their formal and hands-on training sessions).

We have approached this problem in two iterative layers. First, we observed about 20 hours of dispatching operation and recorded their critical tasks (see Figures 2 for a schematic of the dispatching equipment layout and Figure 3 for a picture of van dispatcher's workstation). Then, we interviewed an expert dispatcher and generated a skeleton framework for the HTA model (as described in the next section). The dispatcher's HTA was also re-described in terms of a set of goals (high level of abstraction) and subordinate goals (low level of abstraction). These subordinate goals were organized as a *plan* which governs the conditions for carrying out the constituent sub-goals (for a version of this approach applied to the design of a teach pendant, called Protocol Analysis, see Rahimi and Azevedo, 1990; for another version called Critical Decision Method, see Hoffman et al, 1998). This plan will contain information concerning the sequence in which sub-tasks are carried out and the conditions that must exist before they are undertaken. Examination of these sub-goals highlighted the feedback that the dispatchers must monitor to determine how to sequence their goal-directed actions.

The developed HTA has three distinct characteristics (see the Appendix for the complete HTA):

- A set of action sequences from high level of abstraction to low level of abstraction. The action sequences are also organized according to their sequences of priority from left to right.
- A set of decision nodes where the dispatcher must respond to (cognitively) in order to proceed to the next task. The result of each decision node is a "yes" or "no" based on the plan associated with each response.
- The lowest level of abstraction in which the decision tree reaches its "termination" node. At this point, the dispatcher returns to the higher level of the branch and continues to move to the right and down, until all the nodes are covered.

We suggest that this model is a very useful training tool for novice dispatchers and for experience dispatchers who wish to adapt their decision style to a formalized procedure. The next section explains in detail the validity of this model. The subsequent sections provide the reader with two other useful aspects of this modeling approach. One is that this model can be easily adapted for use as an interface evaluation tool. This can be accomplished by mapping the dispatchers decision paths and define their "over-used" or "under-used" cognitive resources. We have made an attempt to show this approach in the HCI section, evaluating and comparing the dispatcher's use of their DOS-based user-interface design with their use of the Windows-based system. The other is the use of our HTA as a basis for a simulation model of dispatcher's performance system. That is, the HTA model can be easily simplified into a smaller performance-based simulation model (using any general-purpose simulation software) and analyzed based on the available company data. We do not include our simulation attempt in this report.

TASK MODELING AND ANALYSIS OF DISPATCHING OPERATION

Before presenting the HTA model of ABC dispatching operations, we first present some background of their operations. ABC provides traditional taxi services as well as transit services for the disabled. The business and operation are divided into (at least) three distinct operating units: taxi operations, "Dial-a-Ride," and "Access services." Dial-a-Ride is funded by local municipalities and provides intra-city transportation for general public (disabled and able-bodied). Access services is a federally funded program that provides inter-city transportation for the disabled, within a defined service area. (ABC is one of many Access providers in Southern California, each serving an exclusive service area.) The research team made a total of about thirty hours of observations in the ABC's dispatching operations room. The subsequent discussion concerns the dispatching operations exclusively.

Under the federal Access program, ABC is provided with busses and vans owned by the program; and ABC employs drivers (approximately 75) to operate the busses and vans. The vans and busses are specially equipped to accommodate wheelchair-bound passengers. ASI compensates ABC for their services. Performance measures are usually categorized in passenger miles and utilization factors. Therefore, it is in ABC's interest to maximize person trip-miles combined with ridesharing (combining requested trips into a single route).

Under the Access program, eligible passengers are transported for a nominal fee ranging from $1.50-$4.00 (up to twice the fixed route). —which is collected by the ABC-employed driver and remitted to the Access program. The Access-provided vans may only be used to transport Access-eligible passengers. However, the Access services part of ABC's business is not as cleanly separated from the other business elements. Based on our observations, certain degree of commingling and reciprocal use seems to be a part of the operations strategy for ABC. First, ABC uses leased vans (with independent drivers) to supplement Access services capacity during peak times and off-hours. Additionally, ABC also uses its (independently operated, presumably) taxi fleet in the same way. In order to serve Access-eligible passengers, vehicle drivers must be specially trained and vehicles must meet certain standards. All of the leased vans/drivers meet these requirements.

Additionally, ABC utilizes the leased vans and taxi fleet to provide transportation to other disabled, non-Access-eligible passengers, as well, either on a cash basis or other contractual arrangement. Other "accounts" (the Access program is considered as an account) are hospitals, insurance companies and other large organizations. According to ABC, the Access program is the significant and predominant account. It is important to mention that ABC is not allowed to use Access-provided vehicles for this purpose. ABC manages and dispatches both operations—Access-eligible transit and non-Access-eligible transit for the disabled (hereinafter referred to as "Access service") in a single operation, which was the focus of our study.

Access Services Transit Vehicle Fleet

ABC's Access service fleet includes five, Access-provided "big busses," shuttle busses similar to those used for hotel and parking shuttle busses seen at most airports. The "big busses" can accommodate several wheelchairs. However, ABC uses them primarily for groups requiring transport. The fleet also includes about 75 Access-provided vans (most of the vans observed at ABC were 1998 Dodge Caravans). The vans can accommodate 2 wheelchairs plus three additional, ambulatory (non-wheelchair-bound) riders. According to ABC, these vans, on average, are driven about 400-500 miles per day. Additionally, vans are rotated so that total mileage is distributed approximately equally over the entire fleet. Access services also utilizes approximately 15 leased vans. As noted previously, these vans are used for Access-eligible and non-access-eligible disabled passengers.

Finally, the taxi services part of ABC's business includes approximately 100 taxis. Of these, approximately 40 are suitable for Access-eligible transit. It was observed that ABC utilizes the taxis for ambulatory—non-wheelchair-bound—passengers, and the vans are used for wheelchair-bound passengers. However, the vans may also be used for ambulatory passengers as well (this is discussed further subsequently). As expected, taxis transporting Access-eligible passengers are allowed to rideshare, unlike traditional, commercial taxi service.

According to ABC, approximately 60 percent of all Access services requests are handled by vans and the remaining 40 percent are served by taxis. Also according to ABC, use of leased vans will be phased out by next year. And, ABC expects future vans provided by the Access program to be fueled by liquid petroleum gas.

An important aspect of any transit fleet operation is the use of advanced technology for communication purposes. All vans (Access-provided and leased) are equipped with a GPS transponder. They are also equipped with data terminals (such as seen in taxis) for two-way data communication with the (ABC) dispatching operation. These equipment are in addition to a traditional radio system for two-way voice communication with the dispatching operation. Taxis are equipped with the same capabilities.

General Description of Access Services (Paratransit) Operation

As noted before, passengers calling ABC Access services requesting transportation may be: Access-eligible, non-Access-eligible (contract basis), or non-Access-eligible (cash basis). They may also be wheelchair-bound or ambulatory (non-wheelchair-bound). The Access component must also be disabled, requiring special accommodation or assistance. There are five fields of eligibility: unconditional, temp, trip by trip (based on destination), conditional (based on weather or other environmental factors), and denied. However, the details of what constitutes a disability, or what differentiates these from the passengers referred to ABC's taxi service, were not within the scope of this study.

General Reservation/Scheduling Criteria with Performance/Constraints Requirements

Under the terms of its contract with the Access program, ABC must provide transportation to a passenger if his/her request meets one of the following criteria. ABC is penalized (fined a monetary penalty) by the Access program if they fail to do so. Under the definitions of the Access programs, there are three types of reservations/ requests:

- Pre-scheduled (Recurrent)
- One-day advance request
- Five-hour advance request

Pre-scheduled trips are those that occur on a recurrent basis (e.g., a passenger travels to the doctor every other Tuesday at 2:00). The Access program also prescribes limits on the fraction of each type of request. For example, according to ABC, no more than 50 percent of trips may be recurrent or pre-scheduled. However, not all Access-eligible requests fall within the above criteria; i.e., some may be for "immediate" transport, or for transit within less than five hours. ABC accommodates or makes every attempt to accommodate these requests, as well. It is presumed that there are no requirements or penalties concerning non-Access-eligible, cash-basis passengers. The details and requirements of ABC's contracts with their other accounts were not within the scope of this study.

For operational reasons discussed subsequently, all transportation requests are entered into the computer database, and scheduled/dispatched, based on pick-up time. In the terminology and requirements of the Access program, a calling customer is "offered" a ride. Once the offered pick-up time is accepted by the customer, actual pick-up must occur within a window of -5 minutes (early) or +15 minutes (late), or ABC is again (fined) by the Access program. The delay fines increase with each 15-minute increment, with a maximum fine of $100 for pick-up delays over one hour.

Because of these performance requirements, extensive performance statistics are calculated and monitored by ABC. This is facilitated by extensive computerization of the dispatch and operations control processes, discussed subsequently.

Advance Route Scheduling

ABC operates on a 7-day/24-hour basis. However, the majority of requests are for rides during "daytime" hours. According to ABC, 7 a.m. to 10 a.m. and 1 p.m. to 5 p.m. are peak times. Requests for trips on "off-hours" (e.g., evenings and nights) are sparse which makes the likelihood of ridesharing very low. Additionally, the trips during off-hours are primarily assigned to outsourced drivers (i.e., taxis and leased vans), allowing ABC's company drivers to work on a "normal" shift or schedule.

Advance, skeleton routes for the Access-provided vans are constructed the night before, based on advance (24 hour) and recurrent, pre-scheduled trip requests in the computer database. According to ABC, typically, 900-1000 advance requests might be scheduled overnight. However, also according to ABC, not all advance requests for the next day are scheduled overnight. About a third of these trips may require special arrangements which would be preferable if scheduled during daytime. One consequence of this is that the unscheduled requests, mostly for later in the day, have to be dealt with on a real-time basis (i.e., dispatched) by the dispatchers. Scheduling is apparently in time-order (order of requested pick-up time). The skeleton routes (scheduled pick-up and dropoff times and locations) are transmitted electronically to the van drivers when they log into the system at the beginning of their work shift. Also, the dispatch supervisor is given a copy of the advance, skeleton routes early in the morning.

Based on our observations, the construction of the skeleton routes is primarily a manual task, and one individual is assigned to this responsibility. The task is assisted, however, by ABC's GIS software and trip management software. The latter is an elaborate system, which uses GPS to track assigned and unassigned requests, vehicles, and many details concerning vehicle and trip status on both a pro forma and real-time basis.

According to ABC, the primary objective for skeleton route construction is to maximize the utilization of Access-provided vans. These vans are the only vehicles for which advance routes are constructed. As noted before, routes are constructed to maximize ridesharing while simultaneously minimizing total distance traveled. This is to say, the routes are "packed as densely as possible." ABC's data show that approximately 30 to 40 percent of trips are rideshared. Additionally, there are apparently some Access program-imposed constraints concerning ridesharing. For example, there may be limits as to the time or distance that a passenger can be routed out of the way to accommodate ridesharing. However, further details of these constraints were not discussed.

Because of contractual requirements and the compensation and incentive structures, it is in ABC's interest to maximize utilization of Access-provided vans (driven by ABC employees). However, there are also, apparently, trade-offs. Namely, ABC also feels inclined to provide work (trips) to the leased van and paratransit taxi drivers, who are compensated on a per-mile or per-trip basis. It is important for them to maintain the business relationship with other drivers and their availability for outsourcing, as contingent capacity. However, the impact of this is seen in the "now trips" (those that are assigned/dispatched on a real-time basis).

Generic "Real-time" Dispatching Operation

ABC has about 1600 total trip requests per 24 hours, with about 700 of these dispatched (the rest are pre-assigned). Despite the extensive computerization and computer capabilities, the dispatching process appeared to be significantly manual and appeared to include many artifacts and procedures from a generic taxi dispatch operation. The dispatchers were managing about 200 active or imminent trips in a short period of time. Since ABC's Access services operation has resources of at most about 170 total vehicles, it is presumed that all or most of these were in service at the time.

Several individuals were involved in the dispatching, in addition to a supervisor and a trainee. One dispatcher was dedicated to dispatching vans (called "van dispatcher"), and another was dedicated to dispatching taxis (called "taxi dispatcher") for Access service requests only. Additionally, a lead dispatcher monitored and modified dispatched assignments and assisted the dispatchers in assigning unscheduled requests (called "lead dispatcher"). The details of this, and trip assignment, are discussed subsequently. Finally, a taxi service dispatcher was also located in the same area, primarily for commercial, non-Access service requests. However, this dispatcher had no role in Access services dispatching.

Unassigned trips (as well as assigned and "active" trips) are displayed on monitors in front of the two dispatchers, one for vans and one for paratransit taxis. Trips are displayed in order of pick-up time. According to ABC, the goal is to assign unassigned ride requests to vehicles already in service at least 30 minutes before the requested pick-up time. Our observations indicate that dispatchers are able to perform this complex task with a mean accuracy of 96%. As noted before, ABC's performance in the Access program is based, in part, on utilization of the Access-provided vans. Hence, there is incentive to assign unassigned trips to these vehicles first. However, as also noted, ABC also has incentive (both in terms of maintaining contingent capacity and direct business interests) in assigning trips to the paratransit taxis and leased vans. However, in discussions with our expert dispatcher, ABC attempts to distribute trips between each on some "equitable" basis. This equitable distribution of trips appears to be a part of the dispatcher training at ABC. Due to the complexity of the "real-time" dispatching operation, no quantitative "rule" or "heuristic" is instructed for such distribution.

Finally, the dispatchers are instructed to maximize ridesharing, whether utilizing the Access-provided fleet or the leased vans and taxis. Again, this was observed to be attempted primarily by a "tacit" heuristic used by the dispatchers, as modeled in our Hierarchical Task Analysis section. The objective of maximizing ridesharing appeared to be greatly facilitated by the dispatchers' use of the combined (interfaced) GIS/GPS and trip management software, discussed subsequently. Several additional observations concerning vehicle assignment warrant mention and are discussed in the next section.

Vehicle Assignment Nuances

Before we begin detailing the dispatching operations, we present two additional nuances concerning vehicle assignment, as part of the dispatching process. First, (employee) drivers of Access-provided vans are not allowed to refuse trips assigned to them by the dispatcher. However, as independent contractors, drivers of leased vans and the paratransit taxi fleet are allowed to do so.

Additionally, as an artifact of taxi operations, the territory served by ABC's Access services is broken down into "zones." Independent drivers "book into" one or more zones based on their current location. The location of all vehicles is known and available real-time to the dispatchers. The dispatchers communicate with the independent drivers, advising them of the number of available units in their present, "booked" zones. This information may also be available to these drivers on a real-time basis via their mobile data terminals (MDTs). At all times, the dispatchers attempt to coordinate the location of these drivers, i.e., to balance present and/or anticipated needs with vehicle availability, through an "experiential" heuristic.

Dispatching Process Details

The dispatching process (and its associated sub-processes) seems to be an extremely complex and interactive undertaking. This section will attempt to describe in detail the specific tasks and interactions of the observed dispatch process. Therefore, details of the dispatching process are described in two parts: computer/software capabilities of the system, and the human/task element. We will end this section with a preliminary analysis of the dispatching tasks from a human-computer interactive (HCI) design perspective.

A. Dispatching/Trip Management Software

ABC's reservation, scheduling, and dispatch system is extensively automated using computer aided dispatch (CAD) software. For confidentiality purposes, we will not name the CAD software used by ABC. We note that most of the CAD software packages used by the industry have similar features. As a result the studied CAD software can be viewed as representative of the industry. Requests are entered into a computer database, and are coded based on the type of passenger. For example, requests are coded based on whether the passenger is wheel-chair bond (requiring a van), ambulatory (requiring either a taxi or a van), Access-eligible, and/or non-Access-eligible (prohibiting use of an Access-provided vehicle). Additionally, the number of passengers for the requested trip is inputted. Other items on the database are the origin and destination addresses, pick-up time, and fare amounts for each passenger.

The reservation database is integrated into a real-time, dispatching/trip management software system, which has been modified and updated several times since the original purchase. A full-time system administrator programs, handles, and monitors all system components. At any point in time, the dispatcher has access to a real-time text-based and DOS formatted display. This display contains information concerning "active" (in-progress) as well as pending trip requests and assigned and unassigned trip requests. The dispatcher also has similar access to the presently assigned routes (origins and destinations) of all vehicles in the system. Thus, for example, on a real-time basis, the dispatchers can examine unassigned, pending trip requests and compare these to the planned routes of all vehicles in service. The human-computer tasks will be discussed subsequently.

As noted before, all vans are equipped with GPS transponders, which broadcast the vans' present position, heading, and speed. Additionally, through manual data transmission, drivers broadcast their present status: en route to a scheduled pick-up, on-site (but not yet loaded) at a scheduled pick-up, loaded at a pick-up, en route to a destination, and/or their idle positions. This information is fed, on a real-time basis, into a GIS system and display. Thus, at any instant, the dispatchers also can view the location, heading, speed, and status of all vans in service. This information are available (real-time) on a GPS monitor. The dispatchers have access to a complex array of GIS/GPS data on a Windows-based system, including layered windows, pull-down menus, color-coded vehicle lists, locations, and maps. The differences between these two display formats will be discussed later in the Display Design section. An important remark here is that operators rarely used this system, given the significant capabilities of the GIS module.

In order to continue this discussion, we need to see the most basic capability of the software (detailed interactive tasks will follow later). To schedule an unassigned trip request, a tentative insertion is made into a van's present route schedule, by the van dispatcher. The software then calculates the pre- and post-insertion total route distance. The software can also calculate distances of individual trips or segments (all distances are based on GIS "shortest distance"). If the tentative insertion is deemed acceptable, the revised route (pick-up order) is transmitted to the affected van driver; and, all system information (trip assignment and planned van routes) is updated instantaneously. Again, the objective of the insertion process is to maximize Access-provided van utilization while simultaneously minimizing total distance traveled, subject to constraints, such as: all requests (within the time windows) must be served, van capacities, maximum (out-of-the-way) rideshare distance, and others.

Several additional points are noteworthy. First, the software seems to have an extensive data collection and analysis capabilities. This is to say, all of the van, trip, route, and other data described previously is collected and tabulated automatically. These capabilities are used to monitor the actual operation and ensure that Access program requirements and constraints are met. Second, neither the software nor the dispatchers assign specific trip driving directions to the drivers (e.g., specific links such as freeways or surface arteries). This is left to the drivers' discretion. However, the statistics tracked by the CAD software are all calculated based on the GIS "shortest distance" mileage. Moreover, ABC management reconciles actual (odometer) mileage with the "shortest distance" mileage calculated by the software. Presumably, drivers whose actual mileage significantly exceeds the theoretical "shortest distance" mileage are discouraged to do so.

We remark that ABC does not use all of the features of the CAD software, such as advance route creation and real-time dispatching or trip insertion. In fact, as described below, a router creates the skeleton route the night before (manually), and several dispatchers work together to manually develop and improve insertions.

B. Dispatching Task Elements

The dispatchers appeared to be operating at a very high stress level during the peak hours. This is evident, according to ABC, by the high turnover and "burn out" rates in this job. The aspect of the job function that was most striking was the observed and apparent need to simultaneously communicate, interface, and coordinate with multiple system interfaces, drivers, and other dispatch personnel. Moreover, a keen (short-term and long-term) memory appears to be a must prerequisite for this job (capable of remembering the locations of many vehicles, vehicle routes, pending and assigned trips, as well as other data). Those with high learning effects (i.e., high experience in both driving and dispatching) seem to handle the stressful job better during peak hours.

As described before, there were two dispatchers (one for taxis and one for vans), and a lead dispatcher, all actively engaged in the dispatching process. Additionally, a supervisor was on-hand and oversaw individual dispatch (trip insertion) decisions. Additionally, a commercial (non-paratransit) taxi dispatcher was also located in the same area; but this dispatcher had minimal, if any, interaction with the Access services dispatch operation. The dispatchers had two primary responsibilities: maintaining voice and data contact with their respective fleets (including responding to driver requests and inquiries), and assigning unassigned, pending trip requests (from a queue displayed on the monitor) to vehicles. As seen in our detailed HTA model (see the Appendix), preference was given to insertion of trips into planned Access-vehicle routes. In most instances, the dispatchers appeared to identify candidate vehicles for trip insertions based on personal judgment and geographical knowledge. Specifically, they looked at listing of either vehicle assigned routes, or of assigned trips, which included information identifying the vehicles assigned to such trips.

According to the characteristics coded in the trip information database (see the section on Human-Computer Interface Evaluation), either the van dispatcher (V) (for both the ambulatory and wheelchair van requests) or the taxi dispatcher (T) (for the cash and taxi trips) initiate the trip insertion process. In simple terms, the V is constantly checking for the unassigned trips, and becomes mentally aware of the requested pick-up time (or Estimated Time to Arrival) as well as the origin and destination points. This is an intense cognitive operation, which requires total visual and mental attention at all times. Under heavy load (or non-routine trips), they consult the lead dispatcher (L) located between the two dispatchers (see Figure 2 for physical workspace arrangement). The lead dispatcher consults a Windows-based GIS display indicating vehicle positions. Thus, the lead dispatcher is expected to have a more pictorial and comprehensive access to the critically needed information, in support of the van dispatcher's decision process. Additionally, the lead dispatcher performs the function of improving assigned (planned) routings. This task appears to be performed on a "time available" basis. The software also allowed the lead dispatcher to construct tentative trips, if needed (or if requested by the other two dispatchers).

Several, additional observations and nuances warrant mention. First, the display of assigned and pending/unassigned trips also indicated pick-ups that were late, i.e., the assigned pick-up time had passed and the driver had not yet arrived. When these were identified, they were handled on a priority basis by the dispatcher(s) and lead dispatcher working together.

Another item that creates the need for real-time trip management is the issue of "cancellation." Planned schedules and routes are adjusted on a real-time basis due to cancellations. A trip is considered a "cancellation" if the passenger cancelled a previously requested trip more than 45 minutes before the pick-up time. If the passenger called to cancel a scheduled trip less than 45 minutes before the scheduled pick-up time, it is considered a "no show." Each cancellation or no-show needs to be carefully documented in the passenger information sheet for future audit and analysis by Access and the management.

As indicated before, about 200 trips were being managed at each time by the three dispatchers. During the observed periods, much of the dispatching activity focused on the many unscheduled, imminent trips (those approaching 30 minutes of scheduled pick-up time).

<u>Specifics for Real-Time Trip Assignment Procedure</u>

The specifics of the trip insertions ("real-time") are worth exploring at this stage. We begin with the interaction between the van driver and van dispatcher and will end with a discussion of how passengers can impact this procedure as well. We use the specifics of how these tasks interact using the existing procedures indicated by expert dispatchers and trained supervisors. Again, for details of how these procedures form an HTA hierarchy and how each decision node is formed, see the Appendix at the end of this report.

A. Driver/Dispatcher interaction

When a van driver (N) has either finished the route or has a significant time gap, he/she informs the center of his/hers current location requesting a new trip. V then tries to assign N to a trip –or if possible multiple ridesharing trips- with origins close to N's actual location and destination close to N's next origin. Priority is given to the employees van drivers. If V finds one or more trips that fit the above-mentioned criteria, he/she assigns the specific trip to the van number selected, from his/her rideshare screen (S). Unless the driver does not acknowledge receiving and accepting the specific trip, V assigns the specific trip to the route of the specific van, and moves on to the next trip assignment activity.

V can also assign a trip to a van by using the S screen for any non-assigned trips (trips that their "Van #" field is blank). From all the non-assigned trips, V chooses the ones that their ETA is equal or a little greater than 30 minutes from the actual time. (This 30-minute window is important to keep the unassigned trips within the scheduled set.) V then asks (through the voice intercommunication system) for vans wishing to do the specific trip. The vans interested in the specific trip(s) send a voice request message to the center. V calls back the van showing interest based on the time sequence of the received messages. Main criteria for this selection are:

- Distance separating the van's current location with the origin point of the trip,
- Distance separating the destination of the trip(s) from the origin of the van's next trip (when N is serving another pre-assigned route
- Ability to assign multiple trips at the same van (rideshare)
- Type of contract with the van driver

The trip has to meet the above criteria based on the judgement of V, under the current circumstances. If it does, then V performs the following: moves to the S screen, inputs the van's number in the trip's "Van #" field, informs N by voice for the number of trips N has at any point in time. After this point, the driver has to acknowledge receiving the specific trip, using the MDT. Only then, N is able to access the information of the specific trip (address, name of client etc.) from the van's monitor. After V has contacted each van bidding for a specific trip and after a specific trip's assignment has been completed (by seeing the status code of S changing from A to D), V erases N's voice call-request from the problem screen (P) (by inputing the action's number). If for any reason, N does not acknowledge receiving the trip (which is done by pressing the acknowledge button on the monitor, which changes the status code from A to D), V has to contact the driver (by voice) asking for acknowledgement. If this is not possible, V repeats the trip assignment procedure.

It is important that during this decision-making process, L is not participating in the trip insertion procedure unless V requests help by verbally asking L to perform a part of the task. Also, the final decision remains with V, and cannot be questioned or denied by the van driver. Only the supervisor and the company management may question V's decision in a later time.

B. Late Passenger Pickup (caused by van driver)

Potential for a late trip is a serious problem for efficient dispatching operation. A major component of the HTA has been assigned to this portion of the dispatching activity. All dispatchers and drivers know that late trips carry a significant penalty (up to $100 per trip). As mentioned before, a large number of variables contribute to late arrivals. The following is a description of what V might be dealing with when the passenger has a possible delay.

Either V or L becomes aware of the potential late trips by constantly looking at their S screens. When such a trip is spotted, V usually waits for a few more seconds for N to respond to the call. If this waiting time does not affect the delay significantly, V contacts N one more time. Or, V may choose to move into a corrective action. This action can be re-assignment of a part of N's route to another driver, or inform the passengers of the possible delay, by calling the passenger on the phone. If the passengers (for any reason) does not accept the delay (usually this happens when the delay is more than 30 minutes), then V has to reassign the specific trip to another van. In some cases, N decides on a short cut to make up the time lost. In these cases, N will not be paid for the extra miles. The payment is only based on the pre-calculated miles of the optimum route.

C. Late Passenger Pickup (caused by passenger)

A passenger can also cause a delay problem by either being late or by calling for a change or cancellation. In the first case, N uses the voice channel to inform V of the specific situation on hand. Since the type of service is curb-to-curb, the driver may wish to knock at the door for prompt loading (not recommended by the company). A preferred approach is that V notifies the passenger using a phone call. Or, V asks L to do this as quickly as possible. If the passenger informs V or L that a few more seconds is needed, it is up to V to make the final decision based on the possibility of this delay causing problems for the other route segments.

If the passenger, on the other hand, informs V that much more time is needed, again V may decide to either wait or reassign the trip to another N. V may ask N to leave the premises and proceed to another assigned trip, or assign N another trip. (Sometimes this decision is made based on the history of the passenger being late). This action has to be followed by the entry of the specific description and circumstances of this incident in the passengers trip request form. This information is always scrutinized by the management and then reported to the ASI. Again, every attempt has to be made by V to make sure that this action is justified.

Another scenario is the passenger's cancellation of a trip, or change of ETA. If the passenger calls at least 45 minutes before the ETA, then it is handled as a cancellation. According to the company's policy the reason for canceling a trip has to be entered into the passenger's trip information and then it is removed from the S screen. The new trip request will be accepted through the call takers and the new information is entered into the system. Accordingly, V is not taking any actions unless the cancellation is causing a large time or distance gap in a pre-assigned route for a van driver. If this is the case, V will assign N another trip to fill this gap. Otherwise the driver is informed for the cancellation from the monitor, and the new trip is assigned (as mentioned before).

HUMAN-COMPUTER INTERFACE EVALUATION

Background

The following section comprises the results of our observations and evaluations. The data taking tools we used contained a set of video recording of the individual screens used by van, lead and taxi dispatchers. We also recorded the GIS screen activity used by van dispatchers. These video recordings were later integrated into a synchronized "quad-split" format using the USC's cinema/TV post-production facilities.

The purpose of this section is to add detailed human-computer interactive (HCI) information for future software design studies. The reader is cautioned that the information presented here is mostly based on the research team's observations, without the use of any experimental design or statistical sampling techniques. We mainly focused on the interactions between a dispatcher (sometimes herein referred to as "user") and computerized routing software and global positioning system. System software and hardware details will be presented first.

CAD Software:

DOS and Windows-based
Windows 95 4.00.950 Operating System

Hardware:

Processor: Pentium Computer
RAM: 16 MB
Hard Drive: 1.5 GB, 301 MB used, 1.3 MB Free
Network: Unknown, Cable appeared to be Category 5 suggesting at least 10 Mbit/s throughput.

Observation Conditions:

We observed the dispatcher operations under different workload conditions: off-peak and peak hours. During off-peak hours we observed S (names coded for worker protection). During peak hours we observed G and B.

<u>A Basic Dispatching Task Procedure</u>

In order to study the interaction of dispatchers with their software, a basic task was observed to be composed of the following elements:

1. Dispatcher monitors a "problem screen" for a “voice request” message from a driver. A “voice request” is a general request to initiate communication by the driver to the dispatcher.
2. Dispatcher responds to the “voice request” message and removes the request from the problem screen.
3. Dispatcher either handles the request himself/herself or passes the request to the lead dispatcher.
4. Dispatcher scans a “rideshare” screen, which contains a list of all requests for passenger pick-up.
5. Dispatcher searches for possible pick-ups. The dispatcher's goal is to maximize the number of passengers on each vehicle while minimizing miles traveled by each vehicle per trip.
6. Dispatcher assigns closest pick-up request to the driver and sends the updated route to the driver’s onboard Mobile Data Terminal.

<u>Software Interfaces</u>

ABC’s software consisted of two systems, one DOS-based and one Windows-based. The DOS-based software is currently the main system used for routing drivers to pick-up requests and storing information on each passenger. The DOS-based system updates the Windows-based system whenever the DOS system's information is modified. The Windows-based system did not affect the DOS-based system in any way. The DOS system consists of three primary modes (out of five possible menu selections): A problem screen, a rideshare screen, and a passenger detail screen. All screens contain the standard 24 line by 80-character text display format. There are two workspaces for each dispatcher. That is, each dispatcher uses two separate computer display devices. Both devices display screens from the DOS-based system. The two screens allow the dispatcher to simultaneously use two modes within the DOS-based system without requiring further navigation beyond the task on-screen. The dispatchers use the DOS-based system's "Problem Screen" mode and the "Rideshare" mode most frequently. Each display has its own keyboard input device. Modes can be changed on each display device by returning to a menu screen and selecting another mode. This workspace configuration is shown in Figures 1 and 2.

The lead dispatcher uses both a DOS-based screen and a Windows-based screen. The DOS-based screen is the same used by all dispatchers. It is usually set to work in rideshare mode. This screen can be changed to other modes as well. The Windows-based system consists of a screen displaying a detailed map of Los Angeles County with additional information superimposed over the map. Vehicles are tracked on this map using a Global Positioning System. This workspace configuration is shown in Figure 2 and the GPS monitor is seen on the left side of Figure 3.

ABC also operates a simulated vehicle fleet simulator for training new drivers in using the Mobile Display Terminal (MDT). The simulator uses a dummy DOS-based system router to create simulated trips for the MDT. We used a dummy routing terminal for approximately four hours to gain an in-depth understanding of the information interchange among the tasks performed by a dispatcher with this system.

A. DOS-Based System

The main tasks in the DOS-based system are layered in each of the “screens” generated by the main menu. The following evaluates each screen mode based on our observations.

The "Problem Screen" Mode. The problem screen is a driver-to-dispatcher message relay system (see Figure 4 for a sample information layout and content). The problem screen receives several different types of messages. Dispatchers typically answer "voice request" messages. The messages are queued and usually answered in the order they are received. However the dispatcher has discretion over the order in which these requests are answered. Each voice request on the problem screen includes the vehicle number, vehicle attributes, and other information about the driver. Voice requests usually consist of:

- I am clear - Do you have something for me?
- Problem with an order - I can't find the address or the customer
- Add a trip to me - I can take more people
- Complaint - Unclear, Having a Bad Day, Dispatcher is being unfair with assignments, chat, etc.

The dispatcher removes the request from his/her screen once a driver's voice request is acknowledged. When the dispatcher begins to speak to the driver, the dispatcher explicitly confirms the communication protocol by stating the vehicle number on the voice intercom.

Design Issues. Information is displayed in columns, however the columns are organized in such a way that requires a look across a line to gather information. The most important information: Message ID number, vehicle number, and car attributes are spaced throughout the message line. This arrangement violates the "five degree" limit for the viewing angle within which the information must be placed (Tullis, 1986). Inexperienced users may face difficulties (and possibly error) when using this screen under high load conditions. However this problem did not seem to affect experienced users such as G or B.

The "Rideshare Screen" Mode. The rideshare screen is the trip request mode in the DOS-based system (see Figure 5 for a detail format). There are two types of rideshares: static and dynamic. For the needs of this report, we will focus on the dynamic routing. The static routing is performed the night before (around 2 a.m.) for the next day trips. The pre-assigned trips are given to the next day drivers as a skeleton route. (This report does not address issues related to the static scheduling system). Dynamic rideshare routing is performed for trips that are not pre-assigned. The dynamic trip insertions are performed manually. Once a vehicle is assigned, updated instructions for pick-up and dropoff are sent to the assigned vehicle. The dispatcher can reassign rideshare requests as needed.

Design Issues. The only automated component of the trip insertion process is the automated information updating on assignments and pick-ups to the driver's MDT (see below). For a dispatcher to be successful in this job, an excellent memory for detail is a must. Dispatchers must memorize locations, routes, and in some cases the current number of passengers in a vehicle and a vehicles attributes, in order to dispatch quickly and efficiently. The current number of passengers in a vehicle cannot be found explicitly using the DOS-based system. Vehicle passengers and attributes can be found in the DOS-based system by using the rideshare screen and viewing a vehicle's current day history. This cannot be done in parallel to passenger rideshare data. This may force a delay in finding a vehicle that can accommodate a passenger's trip request. Even if all of a vehicle's information and attributes are known, trip information is spaced out over several screens forcing the dispatcher to typically navigate multiple screens before finding a possible pick-up for a driver. Typical navigation required 2-3 screens per vehicle assignment -- a delay of up to several seconds.

Also, significant screen navigation is required due to the simultaneous appearance of both assigned and unassigned trips on one screen. Dispatchers can only recognize unassigned trip requests by an unfilled vehicle number box. This problem mandates a dispatcher to go through both filled and unfilled requests to find a trip request for a driver. The rideshare screen also violates the "five degree" viewing limit (Tullis, 1986). The most important factors: addresses, passenger need attributes, and car number assignment are placed throughout a rideshare request line requiring a user to look across a screen to find pertinent data. In addition, keyboard commands require large movements across the keyboard to enter information.

The "Passenger Information Screen" (Pickup Detail) Mode. The detail screen displays special information on the pick-up request, including passenger name, address, telephone number, and special instructions. Dispatchers can revise this screen as needed per trip requests.

> *Design Issues*. This screen was not accessed frequently. However it was apparent that the information was not organized into any type of easily read structure. Information on each pick-up was placed throughout the screen. This required extensive visual scanning to locate a particular piece of information.

Dispatcher Message Screen. Dispatchers are able to send messages to drivers without speaking to them. The Message Screen allows dispatchers to use pre-made or "canned" messages, in addition to designing their own messages. The dispatcher then enters the vehicle number to which the message goes to and then sends the message.

> *Design Issues*. Like the Passenger Information screen above, the Dispatcher Message Screen is not organized based on any information priority groups or settings. Also, the information layout does not seem to cluster based on frequency of use or eye fixation priority.

Advantages of the DOS-based System

The double-screen (side by side) configuration helps the dispatchers to reduce their need to change screen layers in one screen. This physical “control center” arrangement allows the dispatcher to deal with information in near parallel fashion. This parallelism of physical fixed screens should be maintained in future work/system layout designs. Woods and Watts (1997) have shown that parallel information arrangements are advantageous over forced navigation in complex software interface tasks.

The tabular format in the Problem Screen and Rideshare Screen is also advantageous due to information content simplicity. A time advantage in using tabular formats has been reported by Tullis (1997). However as stated above the current arrangement violates Tullis' (1986) "five degree" viewing limit. A possible solution to this might be to examine what are the most important fields that the dispatchers need, then placing those fields in a functional cluster. In terms of the Rideshare Screen, a further improvement might be to separate the trips into assigned and unassigned components on the screen, then separating those two trip portions into trip areas. However, it is foreseeable to experience some disparity between trips in certain areas. That is, one might have many rideshare requests that must be navigated, whereas another area might have relatively few requests. Nevertheless, according to Tullis (1986, 1997), there should be an improvement in efficiency by redesigning the screen with smaller blocks to scan.

If there were a change in the DOS-based system, an optimal arrangement for all data groups would depend on the number of groups and the size of each group. Tullis (1986) found that the primary indicators of time were 1) the number of on-screen groups when those groups were smaller than five degrees, and 2) the size of the on-screen groups when the group size was larger than five degrees. Hence, any new groupings should both be smaller than five degrees and few in number. However, in trying to adhere to smaller group size and number, we should also be careful not to remove so much information such that the system becomes unusable for the expert user. That is, the efficiency of the overall dispatching task will decrease as the dispatchers gain expertise in performing the task on this type of screen.

The problem of maintaining the novice versus expert dispatcher's performance efficiency is a difficult issue frequently addressed in the literature. For example, novice versus expert use of displays with sparse data fields has been studied by Potter and Woods (1994, referenced in Woods and Watts, 1997) and Reiersen et al (1988, referenced in Woods and Watts, 1997). In these studies, "status at a glance" systems failed to provide enough system information for expert users.

Problems with the DOS-based System

Learning Curve. At ABC, there seems to be a slow learning curve with minimal use of computer aids to understanding how the system is operated. In our observations, several times the screen unexpectedly froze because the operator unintentionally hit the <clear> key on the keyboard while trying to use certain commands. In this system, key commands are extremely esoteric; and the on-screen help were not readily useful for most of these commands. Even less helpful were key command sequences to perform special tasks. This problem is compounded by the fact that the keyboard is non-standard and appears to be custom made for the DOS-based system. In our observations, the key command shortcuts were also difficult to learn for the novice user.

Unintelligible Help. Help screens were hard to access and difficult to understand. The help descriptions were not easily discernable. One way of evaluating the usefulness of a help system is its ability to map the keys or key sequences with program functions. We found it difficult to understand how keys mapped to program functions from the given help summaries.

Lack of Parallelism. For each monitor, the system has no provisions for using parallel information screens. To find essential trip information, the user has to navigate between screens rapidly. This rapid navigation seems difficult and error-prone. For example, if one is using the problem screen, and wants to find a rideshare, one needs to go back to the menu screen, then to the rideshare screen, then back to the problem screen to check for the next voice request. Furthermore, each screen, has its own set of commands, making each screen very mode centered. This would place a memory burden on the user to observe and remember how to use each screen.[1] Woods and Watts (1997) report that users must maintain a constant memory burden when using such a linear navigational system.

Inconsistency in Screen Design. The Problem Screen and Rideshare Screen were designed as a series of ordered columns. This ordered columnar design made reading and scanning the screen relatively easy and efficient. Tullis (1997) reports similar ease of use for screens with ordered columns. Nevertheless, when using passenger information screens for route creation, the user experiences information disorientation. The individual

[1] Most commands are mode-specific. Only a few commands have system-wide application.

data elements were scattered throughout the screen. Furthermore, entry errors resulted in cryptic error messages. This made valid data entry difficult and tedious. Shneiderman (1997) reported similar problems for an information intensive task.

B. Windows-Based GPS Vehicle Location System

The GPS Windows-based system is currently used exclusively as a vehicle location device. The dispatchers use this system when they wish to find the travel direction of the vehicles. This system was particularly useful in cases where the drivers do not respond to communications from dispatchers. Based on our observations, this system was not used as a routing system.

The Windows-based system consists of a Map of Los Angeles County with layers of additional information superimposed on the map. Additional information includes:

- Vehicle data: car number, travel direction, status (stopped for pick-up, moving, etc),
- Passenger data: addresses
- Routes: simplified vector depictions of trips

We observed the map screen being used during about 20-30 percent of the dispatching and routing requests. The Windows-based system was used during non-peak hours significantly more than peak hours. Dispatcher preferred using their own memory and asking drivers for additional information, rather than using the map screen.

Testing Conditions:

The Windows-based portion of the CAD software was tested over a three-hour period. The software was relatively easy to use and understand. We estimate the learning curve for this system to be about two hours for the initial use. There were no manuals or help files available. From information obtained from the supervisors the CAD software training is mostly an on-the-job training.

The Windows-based System Overview:

The Windows-based system is primarily a GPS locator and tracker. It shows a limited capacity to act as a routing software. The GPS aspect of the CAD software was readily apparent as the map and filters were highly visible. Help popup tags appeared when the arrow key was over the on-screen button, and greatly assisted the learning process. The primary screen is a scalable map of Los Angeles County (Figure 6). Additional screen information is superimposed onto the map. The Windows-based system allows the user to display both vehicle and passenger data in relation to the map. Each vehicle and pick-up/dropoff point is represented on the screen by an icon (Figure 7). Vehicles are typically displayed as arrows pointing in the direction they are traveling[2], with a tag displaying the vehicle number and speed[3]. Pick-up sites are represented as red squares and dropoff sites are represented as red circles.

[2] Vehicles had multiple icons. The arrows were used for vehicles in transit. Icons also were color-coded based on the vehicle status.

[3] Vehicles tags appear as: #car; speed

Observing ABC users, the normal screen configuration consisted of the Map Window and Vehicle Detail Window used in parallel. This configuration can be seen in Figure 8. Also, the Windows-based system is capable of filtering information on vehicles and passengers by using checklist dialog boxes (Figures 9 and 10). Information is mainly displayed graphically with popup menus providing layered or cascading popup menus (Figures 11 and 12). Details can also be displayed in separate windows in columnar fashion for vehicles (Figure 13) and pick-up/dropoff locations (Figure 14). This screen is very similar to the DOS-based rideshare screen in terms of format and columnar data presentation. When needed, Trip Routes can also be displayed as simplified vector depictions (Figure 16).

Unlike the DOS-based system's Rideshare Screen, the Windows-based system has a limited capacity to assign vehicles to a rideshare request (see Figures 15 and 16 respectively). Furthermore while the Windows-based system can assign vehicles to trips, there appears to be no provision to allow any explicit summary mode for vehicles, including what trips they have been assigned. There was also no observable capability equivalent to the Problem Screen or Message Screens in the DOS-based system.

Advantages of Windows-based System

The overall impression of the Windows-based system was that it was designed to be an interactive map system, due to its graphical format. This graphical design allows the user to understand the spatial relationship among the various vehicles, pick-up/dropoff sites, streets, etc. which is an advantage over the DOS-based system. In theory, this should reduce user memory load and should allow the dispatcher to focus on other critical tasks.

One major advantage of this system is that it is icon driven. On-screen icons represent a number of different objects (e.g., vehicles, locations). Colors, shapes, and information "tags" are used to present object details. This arrangement increases the screen’s information density. After a period of training, it is possible to quickly identify a vehicle's number, status, direction of travel, and speed. (Figs. 7 and 8). Popup menus further supplement the information presentation with certain degree of redundancy (for an example, see Figure 11). This arrangement is repeated for trip information (Figure 12). Popup menus are easily accessible by clicking on an icon.

The literature supports this type of information presentation in iconic form. When conveying complex information in small screen space, icons have been shown effective (Hemenway, 1982; Rohr and Keppel, 1984; both cited by Tullis, 1997). In addition to icons, the Windows-based system has icon buttons that mimic most menu commands. These buttons are explained by small descriptors that appear when the cursor is held over the icon for a short period of time. We feel that these aids are helpful in both learning and working with the system.

Problems with the Windows-based System

In general, the Windows-based system was cumbersome to use. Other generic problems associated with this system were navigational problems similar to those stated by Woods and Watts (1997) and high-density information display problems as noted by Tullis (1997). The following section will explain specific problems associated with this system.

Lag Time. Perhaps the most prominent problem in using this system is a delayed response or *Lag Time.* Shneiderman (1997) defines one measure of Lag Time as "Response time": *the number of seconds it takes from the moment the users initiate an activity (usually by pressing an ENTER key or mouse button) until the computer begins to present results on the display or printer.* Shneiderman further refines this measure to include the *user think time* (the number of seconds during which users think before entering the next action), additional time delays (e.g. messages, information feedback, etc.) and *user planning time* (the time user plans a task).

All these factors appear to be present in dispatchers interaction with the Windows-based system. However the primary problem areas appear to be the response time and the update (system update) time, defined here as *the time between the end of the response time and the time the computer is ready to accept a new command.*

Our (non-statistical) samples showed a system response time of about three seconds between a mouse click and system response onscreen. Update time after display response time for the system was highly variable. During testing, some update delays were as short as two seconds, whereas, other delays recorded more than 30 seconds.

The mean system response time was measured around 15 seconds. This update rate is disconcerting and forces the user to spend more time waiting to initiate a new command even after the user has planned for the next command. Even though users have been shown to adapt to various response times (Grossberg et. al. 1976, cited in Shneiderman, 1997), system delays of this magnitude may be unacceptable during high workload periods. It is mainly for this reason that the dispatchers are not willing to spend time on the Windows system during peak hours. On the other hand, the DOS-based system has an extremely short response time (less than 5 seconds). The dispatchers are able to perform assignment and routing tasks quickly compared to the Windows-based system.

In addition, Shneiderman (1997) asserts that delay time can be a predictor of operator error rates in a computer interactive task. He cites a number of studies in which extremely short and long response times resulted in increased system errors[4]. Other studies on system response uncertainty and its effects on user performance appear to indicate that highly variable lag times would cause confusion to the user (Sarter et al., 1997; Woods, 1993 Obradovich and Woods, 1996; all cited in Woods and Watts, 1997). At the very least it would seem that this delay would cause frustration and perhaps an increase in errors. The effect of delay time on dispatcher performance should be considered for a future study.

Onscreen Complexity. The Windows-based system has the capability to show routes onscreen. However using this feature may lead to the problem of high-density

[4] A test of circuit-layout clerks who assigned telephone equipment in response to service request found that delays of 4 seconds caused higher error rates than for 12-second delays. Error rates again increased after 12 seconds, with delays of 18 seconds and longer causing the highest error rates.

information display (Figures 6 and 16). Each line in Figure 16 represents a trip. When the volume of trips is high at any given time a large number of lines will appear on screen. As can be seen in Figure 16, even a modest number of trips creates a highly complex mapping space, which the user must navigate through (we observed this interaction during peak hours). Instead, the dispatchers usually used two windows in parallel: the regular map with vehicle overlays and the vehicle detail window. In essence, the dispatchers preset (tailored) a usable display, removing or not using other screens that were built into the system (as noted in *Onscreen Scaling Factors* below). This observation is similar to that noted by Potter and Woods (1994) in the prototype thermal bus software. This problem was also noted in Cook and Woods (1996, cited in Woods and Watts, 1997) of a highly flexible surgical ward software package. In both cases users constrained flexible dynamic views within the system to eliminate the need to navigate among screens, especially during high workloads. In cases when the task load intensifies, cognitive resources are not available for navigational tasks – forcing the user to limit the system use through previously preset screens.

Another explanation is that the Windows-based system may be too flexible for high workload environments. Even with a small fleet of vehicles, the system tends to jumble together at lower screen zoom format. For larger fleets, this effect would most likely increase clutter and render the system useless for vehicle identification. In order to change the zoom effect, the user must select a new window – with a system response time of about 30 seconds. Then, the user must type the vehicle number each time a vehicle information is requested. One way of avoiding this delay is to use the Vehicle Details Window (Figure 13). The Vehicle Details Window allows the user to double click on a vehicle list and be sent directly to a centered, enlarged view of the vehicle icon in its present location and state.

However, the problem remains that the user must navigate throughout the display space to find the vehicle and select it in the Vehicle Details Window. In contrast, the DOS-based system is static, allowing the user to touch the keyboard and move between screens. Furthermore, because the DOS-based system places all the information in predictable rows and columns, there is very little need to look all over a screen to locate the information (i.e. to find an item, the user just scans a column and a row).

Onscreen Scaling Factors. As mentioned above, the Windows-based system provides a zoom function as part of the map navigation task. However as observed by Woods and Watts (1997), zooming may disorient the user. This appears to be the case with the Windows-based system zooming. We found it somewhat difficult to find a map scale that optimally shows a vehicle, its trip route, and other necessary data such as a vehicle number, and location.

Another problem was the inconsistency in-between screen zoom changes and geographical feature resolutions. At one level, the map displays a city, while at another level, we see a city block - with no discernable way to tell the difference in resolution. This created a keyhole shot of the entire fleet of vehicles and trips. Only a small part of the fleet could be seen at anytime with any reasonable resolution of a single vehicle or city. Potter and Woods (1994) reported a similar problem in navigating the prototype Space Station Freedom thermal bus program. There, like the Windows-based system, one screen replaced another as the scale was changed. In that study, the operator used the multiple window capacity in the software to see all the scale views at once.[5] However, while there was a capability to use multiple maps in the Windows-based system, they were not used in parallel as they were in the Potter and Wood (1994) study. A large monitor size (e.g., larger than 19 inches), gives the operator a better screen area for multiple windows.

The end result was that a user is forced to not only navigate between zoom magnification but also through menus to find a usable screen. This significant amount of navigation was observed to increase the task time. Furthermore Woods (1984, cited in Woods and Watts, 1997) has shown that there is cognitive cost associated with display transitions.[6] In this case, the cost burden appeared to be the act of having to navigate the menus and selecting an appropriate filter to display the information needed. Furthermore this problem might diminish depending on the skill level of the user. This might be considered as a factor for a further study.

Another related factor is that mapping from a 2D view into a 3D driver view might cause problems in dispatching. While the Windows-based system does allow the dispatcher to locate an address, it does not provide any details for the driver who is ultimately responsible to find the address. Trip detail menus allow the dispatcher to access passenger information without changing screens. However, there does not appear to be any true facility to add remarks (as is in the DOS-based program's Passenger Detail Screen). Hence when a driver is looking for an address, it may be hard to locate unless additional information is given.

Information Sparseness. In the Windows-based system, information dense iconic screens provide only a sketch view of the routing process. Also, the screens seem to provide very little information on a vehicle's current or past trips. There is also no detail on rideshares or destinations. Furthermore, much of the bookkeeping functionality present in the DOS-based system is absent in the Windows-based system. It seems that the Windows-based system screen design primarily focuses on current tasks, making it difficult to quickly recognize a vehicle’s load status.[7]

GPS System Failure. While it is advantageous to have real-time updates of vehicles, any failure (breakdown) in the GPS system also creates a breakdown in the update capability. It seems to us that the system data on routing would be unusable without GPS updates. However as indicated above, there is no feature that allows vehicles to directly interact with the dispatcher in the Windows-based system as in the DOS-based system. Therefore how data would be gathered in such a failure is unclear. We also observed that when the trip data is unavailable, the system functions as a purely GPS program.

Driver-Dispatcher Passive Interaction (Lack of Communication Mode). The Windows-based system makes the vehicle monitoring essentially passive. It is therefore possible that a dispatcher could assign a pick-up without ever talking to the driver. The dispatcher

---

[5] Henderson and Card (1986) called this configuration room.
[6] We did not calculate this cost function.
[7] It should be noted that the Windows-based system can show the vehicle/trip assignment, however it does not appear that these trip details can be filtered to show a vehicle with trips assigned to it.

could see a vehicle on the map and see that there is a pick-up request on the driver's route. The dispatcher looks up the vehicle on the map and sees that the vehicle has the required attributes and space available. The dispatcher then assigns the pick-up to the driver. The message is then relayed to the driver's MDT. Theoretically, with the current system, there is no interaction except rideshare assignments. As mentioned in the "failure" section above, a driver would never know that anything went wrong if there was a breakdown of the GPS system (i.e., we did not experience any capability to send or receive messages in the Windows-based system). However, the DOS-based system does not seem to suffer from this problem because it has the capacity to send and receive messages both verbally and via the driver's MDT. Unless the whole DOS system goes down, the system allows for active communication.

Vehicle Detail Windows Color Block. The Windows-based system allows the user to color entire lines on the vehicle details screen to correspond to a vehicle's status rather than just icons. As can be seen in Figure 15, the color-coding makes the lines more difficult to read. This is due to the nature of the color-coding scheme, which reduces the contrast between lines. Furthermore the icon colors and background seem to have the same brightness level making it more difficult to read the characters. Fowler and Stanwick (1995, p. 331, cited in Tullis, 1997) suggest that contrast is more important to onscreen legibility than is color.

C. In-Vehicle MDT (Mobile Display Terminal)

Testing Conditions:

We used the company's vehicle fleet simulator to test the Mobile Data Terminals. This simulator is designed to train new drivers on MDT. The MDT is a LCD display, 4 lines x 40 characters. The MDT has a set of specialized function keys, a number keypad, arrows to navigate, and a row of lights to show that the driver is in communication with the dispatcher (Figure 19).

The MDT displays "just enough" information to a driver. "Just enough" means that information is restricted, giving a driver a "bare bones" description of the task they perform.

An example for this information restriction is given based on the normal procedure where a driver accepts a rideshare request. A rideshare request is sent to the driver. This is just a numerical value. The driver must then either ACCEPT the request or do nothing. If the driver accepts by pressing the ACCEPT button on the MDT, the driver is given the basic information of the trip: passenger's name, trip pick-up address, trip destination address, and a route. When the driver arrives at the pick-up site, they must push the LOAD button on the MDT then enter the trip number.[8] When the driver delivers the passenger to the destination, the driver then presses the FNC button, followed by the UNLOAD button and enters the trip number again to release the driver from the task. No information other than what the driver should do is given on the driver's MDT.

[8] This process is done twice, once to show that the driver is on-site and the second to show that a pick-up actually occurred.

MDT Advantages

Because the MDT offers "just enough" information, the driver is given relatively little discretion in how they perform a task. The driver acts as a node that performs a highly specific task. From an efficiency standpoint, this is advantageous because the driver does technically have enough information to perform the task and communicate with the dispatcher regarding the task. This limited information prevents the driver to take any action autonomously.

Problems with MDT

Frame Problem. MDT provides a limited amount of information on the display at any time. While this may be advantageous in that it limits driver actions as explained above, it also makes finding other necessary information difficult. We know that the previous messages are stored on the MDT until cleared. However these are stored linearly. To find information on a trip, a driver is forced to navigate through several message screens until they find the message they need. This is why drivers write down trip information when they receive a trip assignment – a dangerous task while driving.

Memory Burdens. A large memory burden is placed on the MDT user. While a user is able to recall previous information, a significant amount of navigation is required. Therefore, the MDT is seldom used to recall information. Consequently, a MDT user must memorize several keypad commands, all of which cannot be looked up, and most are forgotten outside the short-term memory demand. It is also difficult to use the MDT without knowing keypad commands – a training nightmare for the level of driver education we experienced. Nevertheless, experienced drivers had very little problem using the system.

FUTURE RESEARCH EXTENSIONS

The recommendations given here are based on the specifics of the currently used dual-system application. For a general research direction, see the Conclusion section.

Further study should be directed toward understanding how the interaction between the DOS-based system and the Windows-based system affects the human user. The DOS-based system was exclusively used to fill pick-up requests. This information was relayed to the map screen – not visa-versa. Therefore it is unclear how this interaction could have affected the dispatchers in this task.

As stated above, the Windows-based system mimics a large part of the DOS-based system. However, when the Windows-based and the DOS-based systems overlap in functionality, the Windows-based system is frequently not used. Some factors such as slow speed can be removed with better hardware. However, we cannot predict how the Windows-based system would be received if it had the same response time as the current DOS-based system. It is also unclear whether it would be feasible or even advisable to configure the Windows-based system to have all the functionality of the DOS-based system, or to leave a marginal amount of functionality in a text-based system. It is conceivable that to shift all functions from the current DOS-based system to a Windowed system would decrease user efficiency.

This means that GUIs are not foolproof means to replacing text-based system designs. Also, the strengths and weaknesses presented here are based on our observations only. We need to further study these activities based on a more rigorous scientific and statistical methodology.

As a final comment, what was most striking about the dispatching operation (as well as advance skeleton route creation) was the manual way in which the tasks were performed. While it was clear that the decisions were aimed at the objectives of maximizing utilization and minimizing total mileage, many significant questions are unanswered:

- What tacit rules and neural logic do the dispatchers follow in choosing assignments between outsourced services and Access-provided vans?

- What tacit rules and neural logic do the dispatchers follow in choosing between taxis and leased vans when trips are outsourced?

- If more than one vehicle of the desired type (i.e., taxi, leased van, etc.) is available and suitable, what rules and neural logic is used in choosing the specific vehicle?

- Is there a way to increase the percentage of the advanced (skeleton) routes? Does this eliminate much of the workload of the dispatchers, especially during peak periods?

Finally, the skills and abilities of the dispatchers to identify insertions (and/or choose to outsource the trip) were truly astonishing. During the HTA development (shown in the Appendix), we attempted to capture high level hierarchical progression of rules and decisions that the dispatchers are following in a generic dispatching process. However, capturing the tacit rules and heuristics, and the "fuzzy" decision criteria of the dispatchers requires extensive microanalysis tools and is beyond the scope of this study. Moreover, the dispatchers decisions clearly were aimed at certain obvious and sometimes tacit objectives, including those noted previously. But, it was equally obvious that other constraints and requirements (such as limits imposed by the Access program on ridesharing) did not appear to be considered in the manual process. We do know, however, that the performance versus such requirements is tracked using the data/statistical capabilities of the software, on a post-hoc basis (i.e., deviations and violations would presumably be detected after the fact.)

Another interesting question is whether humans can do a better job (as it is done here) compared to a more automated trip scheduling system. It is speculated that this particular operation includes certain nuances (such as the leased vans and paratransit taxi fleet, and ABC's desire to utilize them) that could not be readily accommodated by the software. Moreover, all vehicles would be required to possess all of the data and location (GPS) capabilities. And, it is not known to what extent available software can accommodate the multitude of requirements and constraints that exist, such as emergency situation, last minute cancellations, and traffic/model conditions. It is speculated that, due to software limitations and the associated needs and limitations of interfacing necessary hardware with the software, there exists no software system that could fully and totally automate an operation such as ABC's.

Further study should be directed toward understanding how the interaction between the DOS text screens and the Windows map screen affects the human user. Text screens were used exclusively to fill pick-up requests. This information was then displayed on the Map Screen. The reverse of this never occurred; i.e. the map screen was never used to fill a pick-up request. Therefore it is unclear how this interaction affects the human user.

The Windows' GUI system (as stated above) appears to be capable of displaying all information in the DOS system. As stated above, the system was slow and hard to navigate, which theoretically would be a frustrating disincentive for high-load periods. Furthermore, it appears that to have the GPS system to show the equivalent amount of data as was shown in parallel on the DOS screens, too much of the screen real estate would be taken. Would this create further crowded, complicated design that would easily confuse and disorient the user? If this is the case, does this mean that the GUI used in this GPS system cannot replicate the text-based system with the same or better efficiency? Why the Windows implementation of the system fails and how the DOS system succeeds should help in redesigning and understanding what components are essential to a successful interface for the dispatching tasks.

IMPLEMENTATION

Potential Benefits:

We envision three main benefits from this research.

The HTA model that we produced can be used as a training tool. We have been told by one of the expert dispatchers that "You can sit next to a dispatcher for one month or study this process map for one hour." In particular, we designed this model in a generic decision hierarchy format which means that it does not reflect only an expert's viewpoint – it represents all the dispatchers should do in order to complete their tasks successfully. However, the training system produced using this process model should have the form of "part-task" training. That is, each section of the decision tree should be selected according to the weaknesses and strengths of the dispatcher trainee. The section that most benefits the trainee should be learned as a part of the overall task performance. As the trainee gains familiarity with each section, other sections must be introduced and their linkages must be learned, accordingly. Eventually, the trainee must be exposed to the whole tree, for a complete and comprehensive understanding of all the subtasks and the sub-goals of the tree.

The second use of this model is considered for software interface evaluation. As described in the body of this report, any evaluation of the usefulness of the dispatching software must consider the critical tasks and decisions that dispatchers must make – often in a very short period of time – to perform their assigned performance goals. Most software designers are neither formally educated nor knowledgeable about the complex evaluation and design issues in the field of human-computer interaction. We have made an initial attempt in this project to expose the readers to an elementary evaluation of the interfaces used in this particular company. A more rigorous study of HCI for this type of interface is beyond the scope of this project.

The third use of this model is its ability to be easily transformed into a simulation tool. For example, one of the graduate students in this project used the model to produce a network of decision nodes in a general-purpose simulation software, called AweSim (Pritsker and O'Reilly, 1999). The simulation model was reduced to 25 nodes (compared to the HTA nodes of about 200). Then, AweSim was run on this model with a large number of sampled data from the ABC Company. According to the student team performing this modeling and analysis, the HTA model was beneficial and gave them a significant head-start in their ability to model the critical dispatching tasks, efficiently.

Practical Application of Research:

The most practical application of this research is the use of the HTA model for training purposes. As explained above, the model format and language is simple yet powerful enough to train the incoming dispatchers with any level of formal education. This is a major advantage for any paratransit company, since the turnaround rate for dispatchers is very high (around 1.5 years in our company). Therefore, potential for gains in efficiency and productivity is apparent with the HTA style training system versus the traditional lecture format that the company is using now. We suggest that the formal training sessions should include a complete demonstration of the task elements of the HTA tree using the standard audiovisual equipment. Given more resources, the company can even add the quad-split videotapes of the dispatching screens to demonstrate each step of the decision tree. However, this step requires accurate and detail matching of the HTA decision nodes with the display screen of each action sequence.

Procedure and Methods for Implementation:

In order for this model to be used industry-wide, each company has to develop its own HTA, based on its set of equipment, management procedure, software and hardware systems, and dispatch training systems. With the experiences of this research team, building an HTA for any dispatch operation should not be a difficult nor time-consuming task. Two features are in our favor: one is that the dispatching task in this research is one of the most complex ones we could find in Los Angeles. Second is the use of a decision tree modeling software called "Inspiration" that made the task of building the decision tree very easy. Any employee with technical or computer skills can perform or modify our tree with little effort.

The Extent of Additional Work for Suitable Implementation:

The only additional work for suitable implementation of this project is to build the model into the current training procedures of our company. For other organizations, as we mentioned before, our HTA model can be easily modified for their particular operation. Then, this model should be incorporated into the dispatch training sessions as discussed previously.

CONCLUSIONS

The paratransit industry seems to be confident that ADA, ISTEA, and TEA-21 have created a steady increase in demand for paratransit services. To meet this demand, computer automation has been embraced as a means of improving capacity and increasing system efficiency and productivity. Unfortunately, open literature contains very little scientific studies, evaluating the application of computer automation to this industry. From a limited amount of information we have (e.g., Lave et al, 1996), it seems that every aspect of paratransit service benefits from computerization. Traditionally, the non-scheduling components of this operation has benefited the most, such as agency billing, accounting, record-keeping, data archiving and system performance data tracking and analysis. On the other hand, routing, scheduling and dispatching components seem to have a mix result, especially when used in a "real-time" demand-response mode. We, therefore, support the notion that companies should consider introducing these technologies (including current and future ITS) in a gradual and systematic way.

In order to implement this systematic approach, companies need to view their technology implementation schemes similar to the current trends in "supply chain management." By this we mean that the paratransit companies need to provide detail information about their specific operational needs and requirements to the software developers. And, both need to communicate their system development strategies with their local and regional governmental service providers. Two problems have prevented this from happening in this particular industry. One is the way software has been introduced into the paratransit operation: essentially grew out of the traditional taxi and emergency vehicle dispatch operation, and slowly "force-fitted" into paratransit and dial-a-ride operations. Our case company is a clear example of this approach where taxi-based software modules are being modified constantly by the company's system engineer. We believe that this approach leads to system inefficiency and user training problems. It is interesting to note that in the last stage of our research, the company management decided to split the van and taxi services into two separate operational control units with independent software control systems.

Even if the first issue is resolved, we still face the second and perhaps a more critical problem facing this industry: a clear lack of design philosophy based on user-centered design principles. According to Norman (1988, 1999), a user-centered design should:

- Make it easy to determine what actions are possible at any moment.
- Make things visible, including the conceptual model of the system, the alternative actions, and the results of actions.
- Make it easy to evaluate the current state of the system.
- Follow natural mappings between intentions and the required actions; between actions and the resulting effect; and between the information that is visible and the interpretation of the system-state at any screen level.

In other words, a good design should make sure that (a) the user can figure out what to do, and (b) the user can tell what is going on. Based on our observations, the DDS system in our case company did not satisfy most of these design requirements. Only the most experienced dispatchers were able to reach certain degree of maturity with respect to these design principles.

We therefore recommend a complete redesign of computer dispatch software (and its interfaces) based on the following notion. We think the most important activities for developing usable systems are those early in the design process, before substantial resources have been committed to any particular design. These include activities related to the difficult decision of what functions to support, and not just the decision of how to present already-decided-upon functions. Methods that aid understanding the work context and how the new technology might influence it are becoming more important as the HCI community recognizes that usability is more than just screen layout. Understanding users means more than cataloging what information is (or believed to be) important. It requires involving the dispatchers (at different levels of training and expertise) perhaps in new and creative ways as collaborators in the design process. We suggest that the cognitive modeling approach in this research is an important step for such a collaborative and iterative design process.

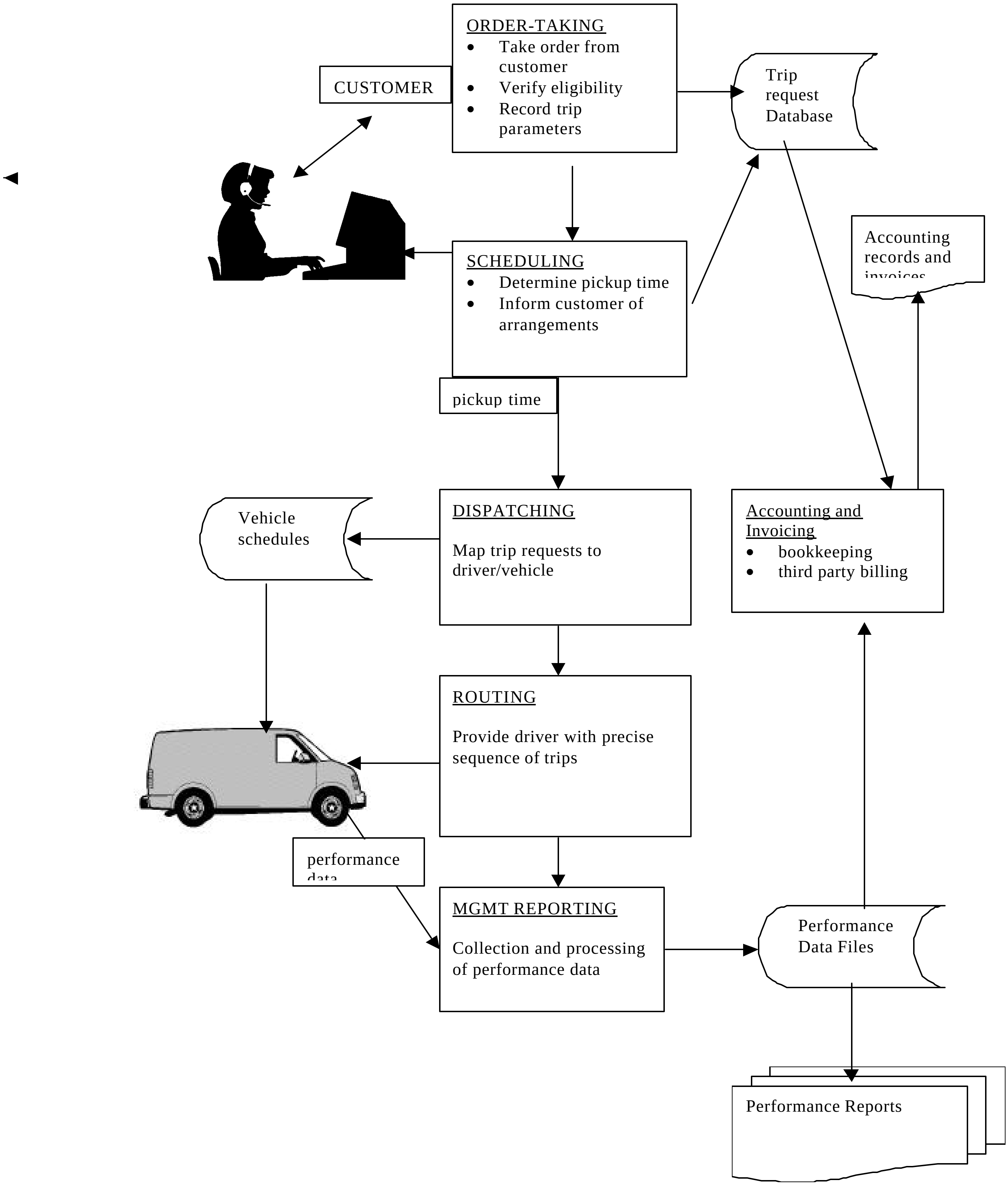


Figure 1. A schematic diagram for software modules and information flow in a computerized dispatch operation.

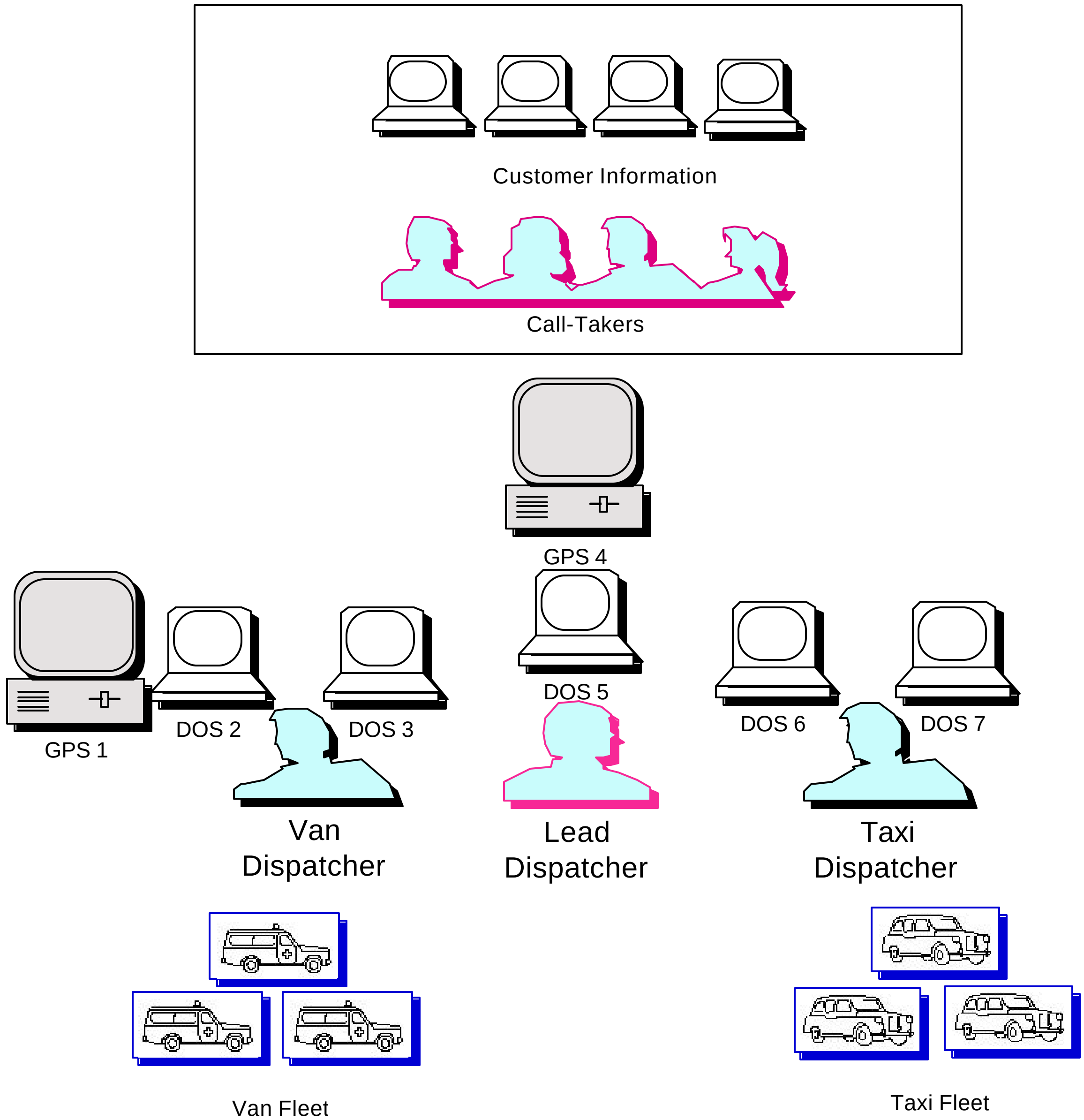


Figure 2. A schematic diagram for the entire dispatch operation and work site. This research deals with the van and lead dispatchers operation only. The GPS display contains all Windows-based information and DOS displays contain mostly tabular information.

Figure 3. The van dispatcher with his Windows-based and DOS-based displays, two keyboards (one for each display), and his phones communicating with other dispatchers, passengers, and the van drivers.

ID	NO	Street / Message Text	UNIT	ATTR	TYPE	P	ZONE	TIME	C	CAR	TYPE	OTHER
Queue ID #		Message (Voice Request)		Vehicle Attributes	Cash / Non-Cash		Current Zone #	Time Message Received		Car ID #		

Figure 4. DOS “Problem Screen” format with tabulated column headings and critical heading description.

ID	TIME	ATTR	CAR	S	#	T	ADDRESS	TO_ADDRESS
Pickup Request #	Pickup Request Time to Pickup	Attributes needed to fill Pickup request	Car # Currently Assigned to Pickup Request	Status {D = Dispatched, N = There, A = Being Assigned, U = Undispatched}			Pickup Address	Where Car Goes to Deliver the Rider

Figure 5. DOS “Rideshare Request Screen” format with tabulated column headings and critical heading description.

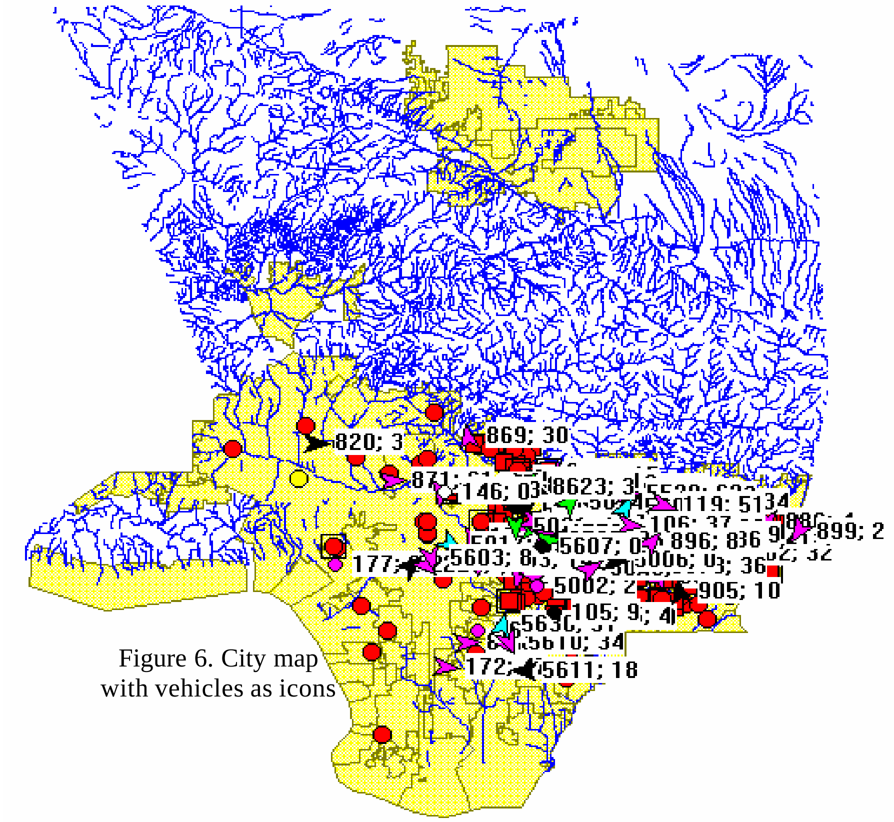


Figure 6. City map
with vehicles as icons

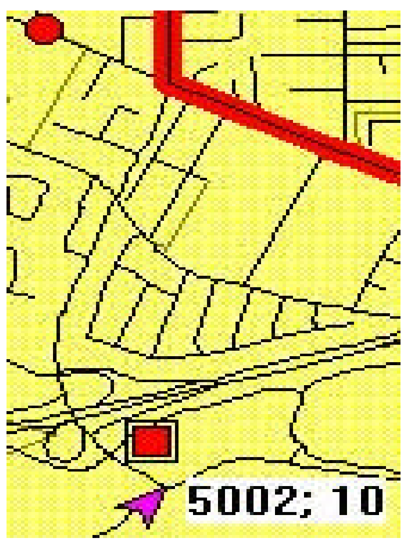


Figure 7. Map icons
(Zoomed-in)

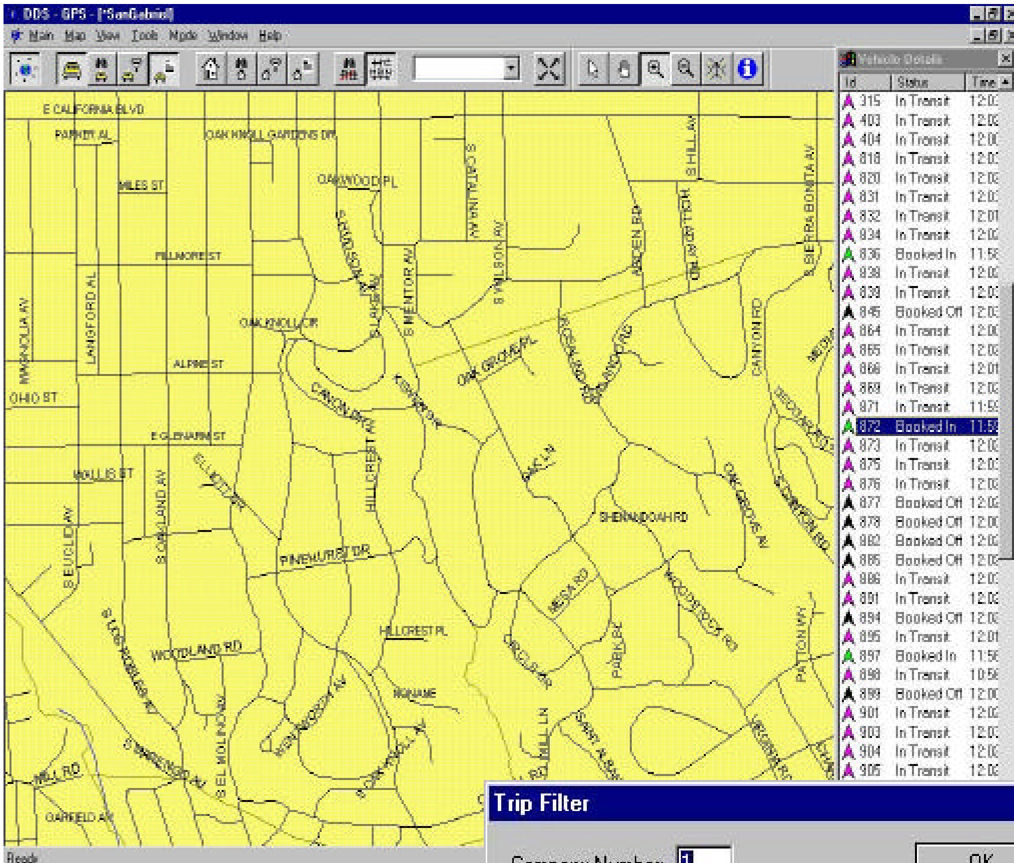


Figure 8. The ma
confi

Figure 9. Vehicle State Filter Window

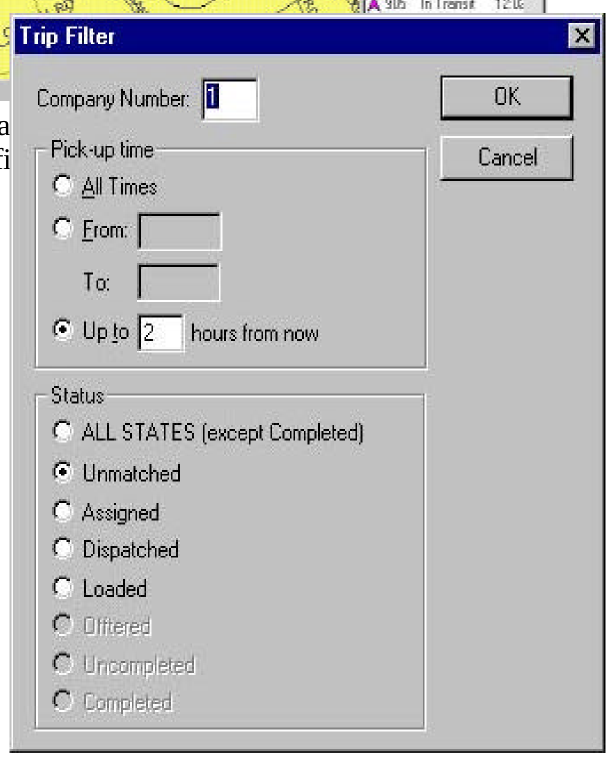


Figure 10
Trip Filter Window

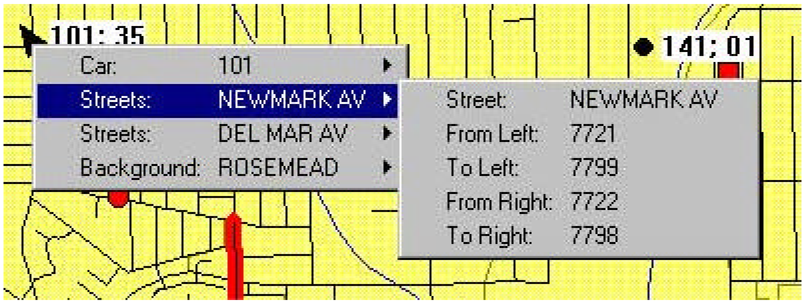


Figure 11. Vehicle detail pop-up menu

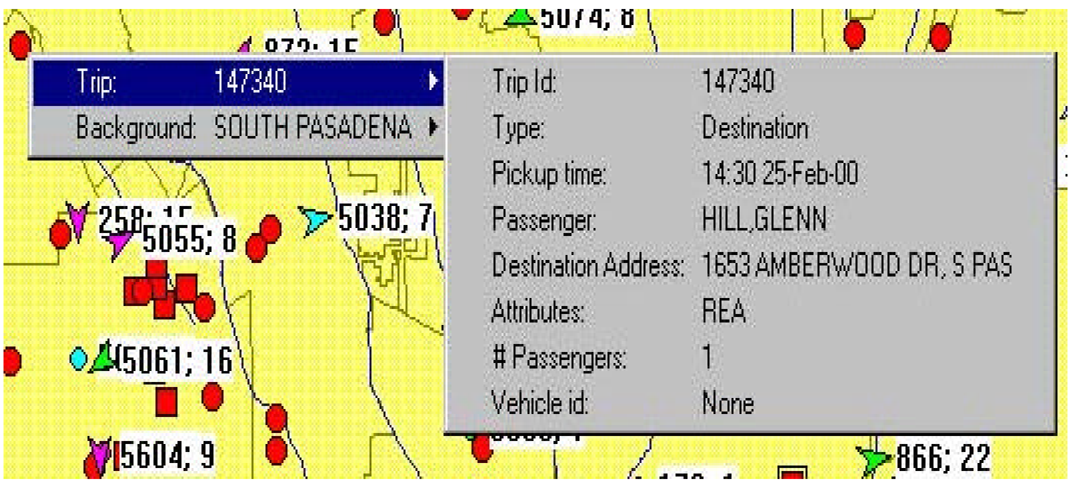
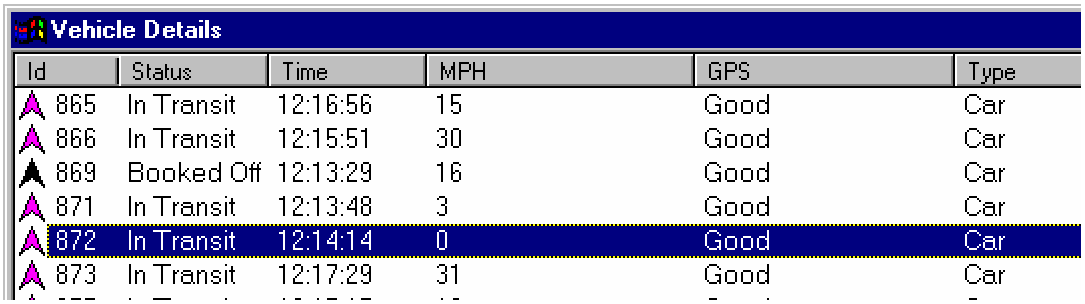


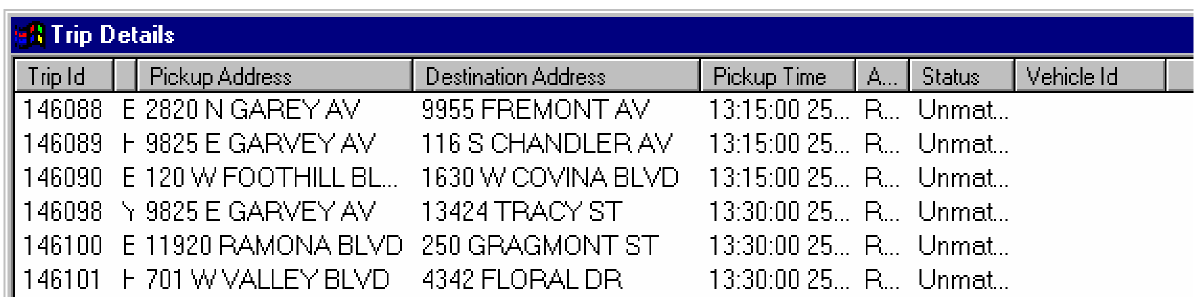
Figure 12. Trip detail information with pup-up menu



Vehicle Details

Id	Status	Time	MPH	GPS	Type
865	In Transit	12:16:56	15	Good	Car
866	In Transit	12:15:51	30	Good	Car
869	Booked Off	12:13:29	16	Good	Car
871	In Transit	12:13:48	3	Good	Car
872	In Transit	12:14:14	0	Good	Car
873	In Transit	12:17:29	31	Good	Car

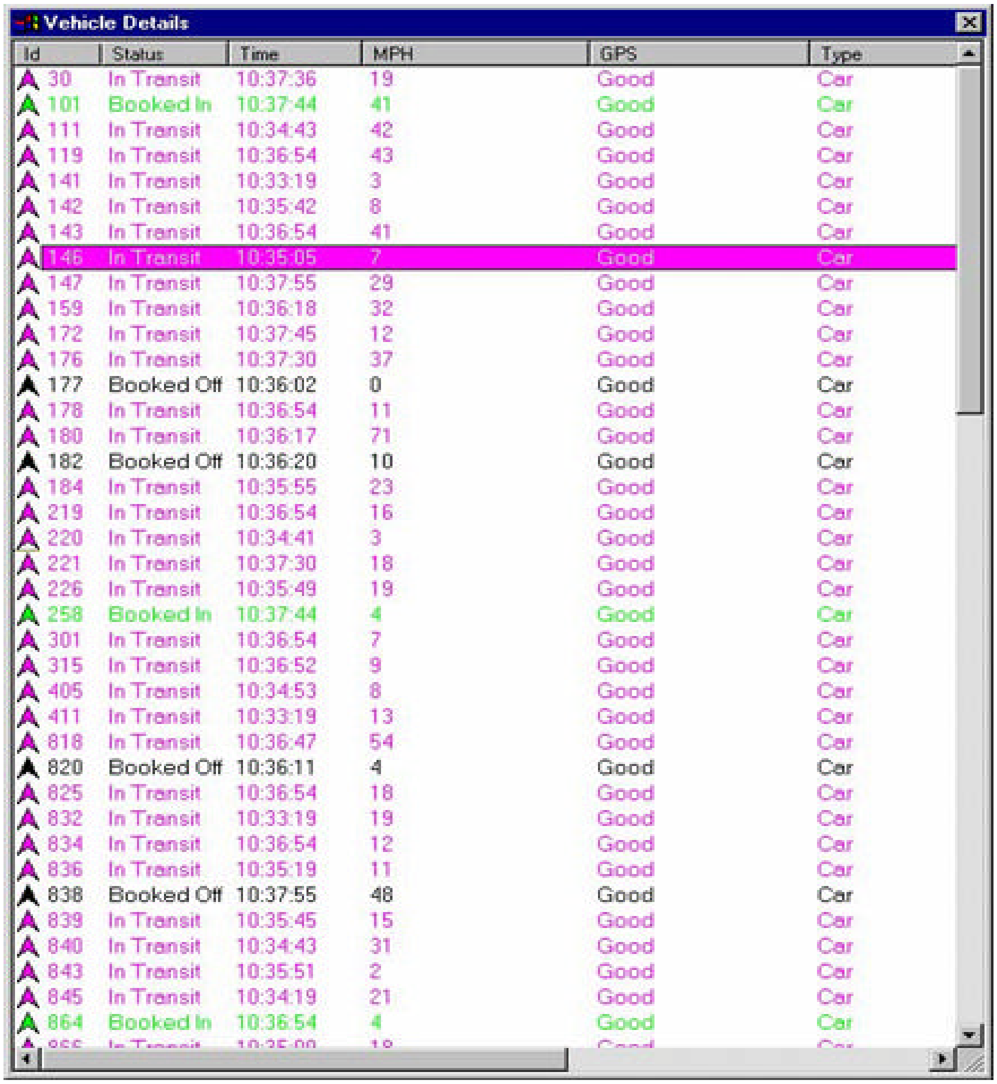
Figure 13. Vehicle details window



Trip Details

Trip Id		Pickup Address	Destination Address	Pickup Time	A...	Status	Vehicle Id
146088	E	2820 N GAREY AV	9955 FREMONT AV	13:15:00 25...	R...	Unmat...	
146089	H	9825 E GARVEY AV	116 S CHANDLER AV	13:15:00 25...	R...	Unmat...	
146090	E	120 W FOOTHILL BL...	1630 W COVINA BLVD	13:15:00 25...	R...	Unmat...	
146098	Y	9825 E GARVEY AV	13424 TRACY ST	13:30:00 25...	R...	Unmat...	
146100	E	11920 RAMONA BLVD	250 GRAGMONT ST	13:30:00 25...	R...	Unmat...	
146101	H	701 W VALLEY BLVD	4342 FLORAL DR	13:30:00 25...	R...	Unmat...	

Figure 14. Trip detail windows



Vehicle Details

Id	Status	Time	MPH	GPS	Type
30	In Transit	10:37:36	19	Good	Car
101	Booked In	10:37:44	41	Good	Car
111	In Transit	10:34:43	42	Good	Car
119	In Transit	10:36:54	43	Good	Car
141	In Transit	10:33:19	3	Good	Car
142	In Transit	10:35:42	8	Good	Car
143	In Transit	10:36:54	41	Good	Car
146	In Transit	10:35:05	7	Good	Car
147	In Transit	10:37:55	29	Good	Car
159	In Transit	10:36:18	32	Good	Car
172	In Transit	10:37:45	12	Good	Car
176	In Transit	10:37:30	37	Good	Car
177	Booked Off	10:36:02	0	Good	Car
178	In Transit	10:36:54	11	Good	Car
180	In Transit	10:36:17	71	Good	Car
182	Booked Off	10:36:20	10	Good	Car
184	In Transit	10:35:55	23	Good	Car
219	In Transit	10:36:54	16	Good	Car
220	In Transit	10:34:41	3	Good	Car
221	In Transit	10:37:30	18	Good	Car
226	In Transit	10:35:49	19	Good	Car
258	Booked In	10:37:44	4	Good	Car
301	In Transit	10:36:54	7	Good	Car
315	In Transit	10:36:52	9	Good	Car
405	In Transit	10:34:53	8	Good	Car
411	In Transit	10:33:19	13	Good	Car
818	In Transit	10:36:47	54	Good	Car
820	Booked Off	10:36:11	4	Good	Car
825	In Transit	10:36:54	18	Good	Car
832	In Transit	10:33:19	19	Good	Car
834	In Transit	10:36:54	12	Good	Car
836	In Transit	10:35:19	11	Good	Car
838	Booked Off	10:37:55	48	Good	Car
839	In Transit	10:35:45	15	Good	Car
840	In Transit	10:34:43	31	Good	Car
843	In Transit	10:35:51	2	Good	Car
845	In Transit	10:34:19	21	Good	Car
864	Booked In	10:36:54	4	Good	Car

Figure 15. Color-coded vehicle details window with highlighted line

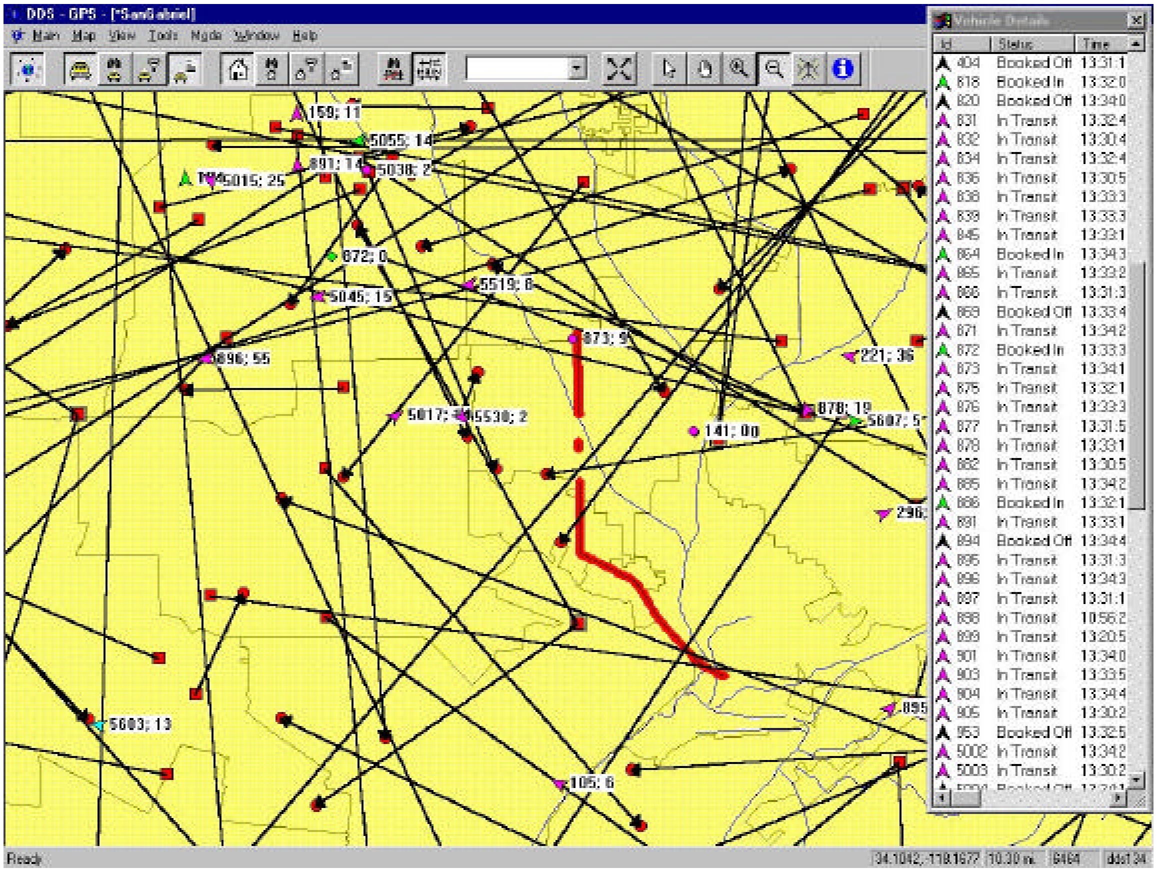


Figure 16. Map with routing details (red line is a street selected after assignment)

Figure. 17. Trip assignment menu

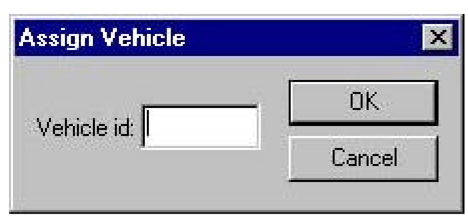


Figure 18. Trip assignment dialog box

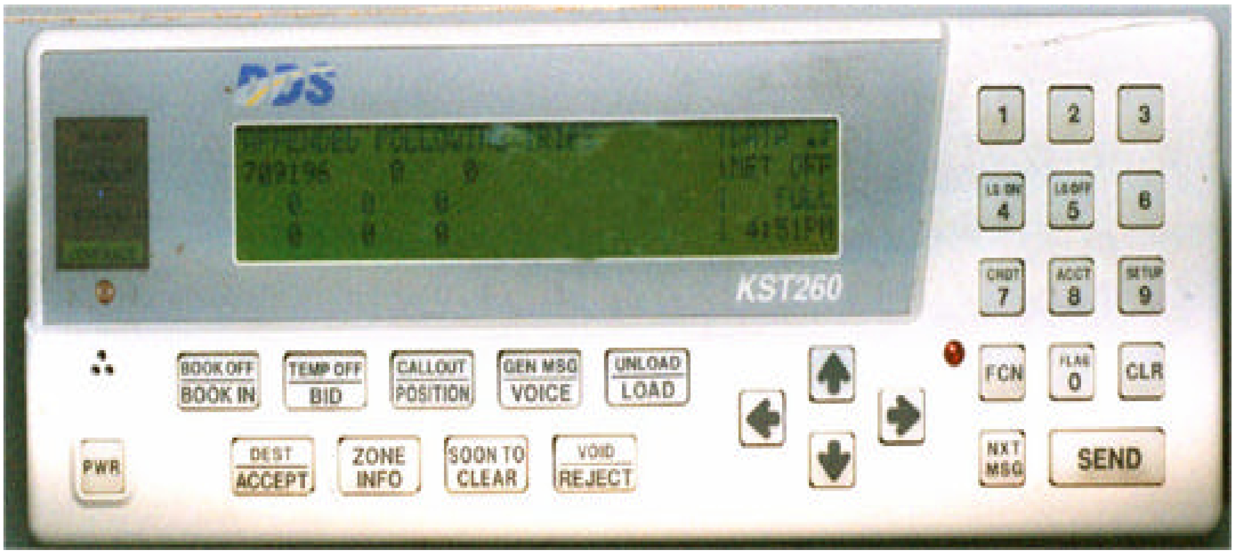


Figure 19. In-vehicle mobile data terminal (MDT)

APPENDIX

A Detailed Description of HTA Network Elements

Introduction

In this section, we describe in detail, the cognitive activities of an expert van dispatcher, as the focus of our HTA analysis. We are also examine and add the van dispatcher's cooperation with the lead dispatcher, as both dispatchers become heavily interactive during peak demand periods. As mentioned in the body of the report, a detailed HTA diagram has been generated and presented in the next section of this Appendix. This network is created as depth first. This means that we develop the nodes from top to bottom until we reach an END node, then we go up until we find the first IF-THEN node, and then follow its other branches (for node symbols and explanation, see Node Symbol Description below). When we have covered the entire IF-THEN nodes, we go to the top node and we move from left to right to the next node. We follow this approach in order to give an impression of time sequencing in the network. Therefore, the action at the bottom node follows the top and the right node follows the left.

Van Dispatcher's Node Description:

Each van dispatcher needs about 10 minutes, from the time he/she enters the room up to the moment that he/she is starts their shift. This time is required for the dispatcher to mentally prepare for the current workload, current trip assignments in the network, and potential difficulties encountered by the previous dispatcher. During this adjustment period, the dispatcher stands closely behind the one who is about to leave, and watches both his/her actions and the screens that he/she is working on and checks for any specific problems or any trips running late. After taking his/her position, he/she is primarily interested in three aspects of the operation: 1) late and unassigned trips, 2) problem entries and 3) delayed trips. Therefore, we have modeled the cognitive interactions of the van dispatchers with their system according to these three elements. Note that the HTA network that we generated captures only the "typical" or "generic" activities. Other variations on this network are possible, due to the complexity of the system interactions.

Late and Unassigned Trips

This is perhaps the highest priority for a dispatcher in his/her web of cognitive activities. The dispatcher attends to this need, by first checking the rideshare screen for any trips with empty vehicle number field next to the trip number which means this particular trip has not been assigned yet to any vehicle. This becomes even a more significant problem when the due pick-up time is less than 30 minutes and the trip still remains unassigned. Initially he/she is checking his/her GPS screen in order to locate the pick-up address and then visually searches to see if there are any vehicles available in close proximity. Their availability status shows from the symbol with which they appear on the GPS screen (see the GPS map screen layout in the report for a pictorial representation). If he/she finds an available vehicle close by, he/she has to first communicate with the van driver to inform them of the trip being assigned. Then, the van driver acknowledges the trip. The dispatcher has to perform a sequence of actions at his/her rideshare screen, in order to dispatch the trip to the specific vehicle. If the van driver acknowledges receipt, then the rideshare screen shows the vehicle's number listed in the specific field next to the trip number.

If the van dispatcher does not receive any response back from the van driver, he/she starts considering the ridesharing option. He/she first checks the rideshare screen for any vehicles whose departure and destination of the current trip matches (or closely matches) with the departure and destination of the trip that he/she is trying to assign. If he/she finds any vehicle matching the above criteria then the van dispatcher has to first retrieve the trip's data, then contact the van driver by voice, assign him/her the specific trip and receive the van driver's voice confirmation.

In the case that the van dispatcher cannot locate any vehicle matching the distance criteria, the assignment algorithm of the system will take over. This is designed into the system using the rule of the pick-up time being less than 30 minutes. The system will either assign the trip to a vehicle that according to the software is the most appropriate to serve the specific trip, or leaves it unmatched. There is also the possibility that even if the system assigns the trip, the driver may not accept the trip. This is unlikely for the company drivers (if they do so, they must explain their decision to the management). In this case, if the trip is still unmatched, the van dispatcher can either ask the taxi dispatcher to try to accommodate this trip (by an Access certified taxi) and contacts the passenger to buy more time. Or, in the worst case, the dispatcher may contact another provider to accommodate this trip. However, the current practice seems to favor the dispatchers to repeat the whole procedure if they can buy enough time from the passenger.

Problem Entries

The van dispatcher is checking the problem screen for any problem entries that have not already been answered by the previous van dispatcher. Usually, there are multiple messages appearing on the screen. Some of them have high priority; the van dispatcher ignores what he/she believes to be low and medium priority. The low priority messages are the ones that are usually produced by the software itself, e.g., "late meter on." The decision rule is that the messages that do not apply to the current operating conditions are left for later consideration. Among all the problem messages, the highest priority goes to the "No Show" message. ("No Show" is used from Access to indicate that either a passenger didn't show up at the pick-up place at the pick-up time or that he/she called less than 45 minutes before his/her due pick-up time to cancel/change his/her trip. This does not leave adequate time for the dispatcher to inform the van driver who might be at the pick-up location anyway.) This is due to the fact that the driver declares to the dispatcher that either the passenger (who ordered the trip) is not at the pick-up location or the passenger is not ready for the actual pick-up.

A passenger no-show is a serious problem because of its potential impact on all the following trips in a specific vehicle route. Therefore, the van dispatcher has to do a sequence of actions to check if the "No Show" is factual and begins to gather information to respond accordingly. The dispatcher first locates the van at the GPS screen by typing its vehicle number and presses <Enter>. Then, at the information screen, he/she retrieves the trip's details. Based on these two pieces of information, he/she moves to the next decision point. If the van is at the expected pick-up location, then the van dispatcher calls the passenger to verify a "No Show" condition. Talking to the passenger generates the following decision node: If the passenger denies the "No Show", this means that the passenger still wants to make the specific trip. In this situation, the van dispatcher denies the "No Show" to the van driver and erases the problem from his/her problem screen. The dispatcher then informs the van driver at the same time. If the dispatcher confirms the "No Show", he/she adds the "No Show" in the passenger's data at the comments field, and he/she grants the " No Show" to the van driver. If the passenger is not answering the phone, then the van dispatcher considers the van driver's claim as factual. The dispatcher adds the "No Show" to the passenger data under the comments field, and grants the "No Show" to the van driver.

In terms of the medium priority problem entries, we characterize entries based on the following requests from the van driver:

A. Free and available for a trip,
B. Cannot find the pick-up address,
C. Cannot locate the passenger,
D. To be relieved of a pre-assigned trip and
E. Additional trip while serving one currently.

The van dispatcher answers the calls based on the sequence that they were received (First-In-First-Out). The dispatcher enters the problem number that corresponds to the driver request and then calls him/her on the voice channel for detailed information. For each of the above requests, the dispatcher is expected to perform a series of actions to respond to the specifics of the situation. Here is the detailed explanation for each request and the corresponding dispatcher interactions:

A. Frequently, a van driver requests to be assigned to a trip because he/she is idle between trips of his/her daily (pre-assigned) route. After such a request, the van dispatcher checks the time availability of the driver for another trip. The dispatcher does this by checking the driver's current location and where his/her next pick-up address is located. If a time slack is available, then the dispatcher attempts to assign a trip that is either still unassigned or has been refused by another van driver for whatever reason. But this will only be done if the new (added) trip does not take the van driver away from his/her current route direction. Such a decision is based on driving time, direction, next pick-up address, and possible traffic congestion. An expert dispatcher develops this mental picture in a few seconds, while retrieving the routing data for the specific van.

The actual steps the dispatcher takes are as follows. He/she enters the vehicle's number at the rideshare screen and presses F8. In the same time he/she locates the van at the GPS screen to check its current position. Based on the above information, if the van dispatcher finds an appropriate trip, he/she performs a sequence of actions to assign the specific trip(s) to the van that was idle (and requested additional trips). Then, he/she informs the van driver (verbally) that the trip is assigned. If the van dispatcher cannot find any appropriate trip, he/she then asks the van driver to "sign in." By this, he/she is letting the driver know that a trip is not currently available. If the driver remains signed in, he/she is on the queue for another available trip, or in rare occasions, the system may assign the driver a trip. Otherwise the driver continues to be idle until his/her next pick-up.

B. If a van driver informs the van dispatcher that he/she has a problem locating the pick-up address, then the van dispatcher has to perform a sequence of actions (a) to retrieve the driver's trip information and (b) to guide the driver to his/her current pick-up address. The van dispatcher initially retrieves the specific trip information by entering the trip number at the rideshare screen and presses F8. Knowing the address, he/she then pinpoints the vehicle location using the GPS screen. Depending on how far away from the pick-up address the van driver is currently located, the dispatcher may need to do a single or multiple zoom-out of the GPS screen in order to have both the van symbol and the pick-up address on the same screen. The van dispatcher is then able to give directions in order to aid the driver to proceed as quickly as possible to the pick-up address. The dispatcher then waits for the driver to acknowledge his/her new (or corrected) direction.

C. Sometimes, the driver is located at the pick-up address at the right time, yet, the driver still has a problem finding the passenger. The van dispatcher has to follow a specific procedure to find out if the driver is at the wrong address, or if the information is inputted incorrectly into the system. The van dispatcher starts by retrieving the trip data by entering the trip number at the rideshare screen and presses F8. Then, he/she locates the pick-up address at the GPS screen. After retrieving the data, he/she checks the trip details in order to make sure that the driver is aware of the factual data, and proceeds as before.

Some of the data in the passenger information screen that help to clear the error may be: a name at the front door, apartment number or building number, etc. If the van driver confirms that he/she is aware of and followed any special remarks/requirements, then the van dispatcher has to call the passenger to check if this is actually a "No-Show" situation. If the passenger does not answer the phone, then the dispatcher enters into the Comments field of the passenger's data: contacted passenger, no answer. Then, he/she will grant a no-show to the van driver. If on the other hand the passenger answers the phone, then he/she can inform the van dispatcher if he/she is willing to make the trip. Under this case, the van dispatcher erases the problem from his/her problem screen and informs the van driver to stand by for the passenger pick-up. If the customer informs the van dispatcher that he/she is not willing to do the trip, then the van dispatcher updates the passenger's data and grants the "No-Show" to the van driver. The dispatcher also informs the driver of this no-show by the voice channel, simultaneously.

This final voice communication is necessary since in many situations, the driver may have skipped or is unaware of some of the specific trip instructions. This communication informs the driver again of these instructions. If the van driver (after following these instructions) can locate the passenger, he/she proceeds to pick up the passenger and starts the trip. If not, then the van dispatcher has to call the passenger in order to inform them of the problem and find out if he/she is still interested in doing the specific trip or not. This elaborate and duplicate procedure is mandated by Access Services in order to claim a no-show. A detailed documentation of this activity is kept in the passenger's data file for future inspection and audit.

D. A justifiable reason needs to be present for a driver to request a relief from a pre-assigned trip. It is up to the dispatcher to make this judgement. The dispatcher has to find out and record the reason(s) for such a relief. In such a case, the dispatcher retrieves the vehicle's routing data at the rideshare screen by entering the vehicle's number and presses F8. Having retrieved both the system's information and the van driver's explanation, the dispatcher makes a decision granting or denying the van driver's request. If the van dispatcher accepts the request, he/she then deletes the specific trip from the van driver's route. Immediately, this trip becomes available for assignment to any other vehicle as it was described before. If the van dispatcher does not agree with the van driver, then he/she asks the van driver to cover this trip anyway (against the wishes of the driver). The driver is allowed to complain about the dispatcher's decision (after the shift) to the operations manager.

E. Sometimes a van driver requests to be assigned another trip, while serving the current trip. In this case, the van dispatcher has to check the unassigned trips in order to see if he/she can rideshare a trip that he/she has difficulty assigning to other idle vans. In order to do this task, he/she has to first check the current trip details of the requesting van driver. The dispatcher performs this task by entering the vehicle number at the rideshare screen and presses F8. At the same time, he/she locates the specific van at the GPS screen in order to have a good geographical picture of the current position. If the van dispatcher finds a suitable trip that he/she can assign to the van driver, he/she selects the trip number from the rideshare screen, presses TAB, enters the vehicle number and finishes the assignment procedure by pressing F17. At that moment, the new trip details are available to the van driver. The van dispatcher considers the action completed after the van driver has acknowledged the trip acceptance.

Delayed Trips

The last stage of the van dispatcher's activity is to check for the delayed trips on the rideshare screen. Again, this is a critical task for the dispatchers to perform due to the potential for penalties assigned to unserviced trips. During the peak demand times, the van dispatcher normally does not have the opportunity to deal with severely delayed trips. Thus, the van dispatcher requests the lead dispatcher to take control of these trips. The following section describes the lead dispatcher's attempts to handle this type of request.

Like any other dispatcher, the lead dispatcher's shift begins with a 5 – 10 minute period familiarizing with the existing trip request conditions and tracing the potential problems. From the time he/she sits down and starts the shift, he/she is continuously monitoring both the rideshare screen and the GPS screen looking for vans running late from their pre-assigned pick-ups. His/her main function is to inform the van dispatcher of potential problems, rather than taking independent actions. For example, experience may tell the van dispatcher that, according to the current traffic conditions (reported by other van drivers), an assigned van may not have adequate time to reach his/her next pick-up location on time. The van dispatcher then asks the lead dispatcher to take care of this situation, while he/she is attending to other requests. The lead dispatcher then locates the trip at the rideshare screen in order to see the pick-up time and location and also any details that may indicate passenger flexibility in pick-up time. If, for example, the passenger has to go to a hospital or an airport, the passenger time flexibility may not exist for delayed pick-up. This situation requires an immediate and concentrated attention of the lead dispatcher.

In such a case, the lead dispatcher locates the late van at the GPS screen, and develops a mental estimate of the time the driver needs to reach the pick-up location. If he/she sees that the van has reached the previous pick-up location and he/she is already moving toward the dropoff location, then the driver has forgotten to "load" the trip, or the main system was not updated (showing the van still on the way to the pick-up location). In this situation, the lead dispatcher is using the message window from the rideshare screen to send a message to the van driver to load the trip. The driver has to perform this task once to let the system know that he/she has reached the pick-up location, and then again, to actually board the passenger. If the van driver follows the lead dispatcher's request, then the symbol at the "status" field of the rideshare screen will change from N (indicating "vehicle on site") to L (indicating "passenger picked up") which is a logical conclusion for the lead dispatcher. Then he/she informs the van dispatcher verbally that the delayed trip problem has been solved and the passenger was picked up.

If, on the other hand, the lead dispatcher checks the GPS screen and he/she finds the van dispatcher at the pick-up location (perhaps sitting idle), it means that the van driver forgot to load the trip and the main system was not updated. The lead dispatcher follows the same procedure as mentioned before. The only exception is that the lead dispatcher waits to see the change of the status (indicating that the passenger has been picked up) in order to consider this request completed. Then he/she informs the van dispatcher verbally that the van driver has picked up the passenger.

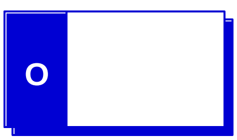
In the worst scenario, however, the lead dispatcher (by checking the GPS screen) finds that the van is not close to the pick-up location. In this case, he/she uses the message function of the rideshare screen to send a message to the van driver requesting the estimated time to arrival at the pick-up address. Then he/she informs the van dispatcher of the van's current location and distance from the pick-up address, and informs them that he/she is waiting for a response from the van driver. If the van driver replies that the estimated time of arrival at his/her next pick-up location is less than 10 minutes, then the lead dispatcher allows the van driver to proceed to his/her pick-up location and he/she continues to monitor the delay situation. This continuous monitoring is necessary to ensure that there will be no additional delays for this trip. If the van driver replies that the estimated time of arrival is more than 10 minutes, the company's regulation does not allow them to take any further action. However, he/she has to inform the van dispatcher of this situation. The van dispatcher is now responsible to handle the situation accordingly.

Node Symbol Description

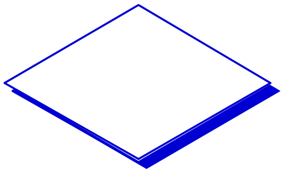
Insert Node indicates that an activity needs to continue somewhere else in the network, possibly due to a lack of space or is being duplicated with an Exit Node with the same number (see below)

Exit Node indicates that an activity is being followed (or duplicated) from another point on the network. Note: there must be at least one Insert Node with the same number as the one on this Exit Node.

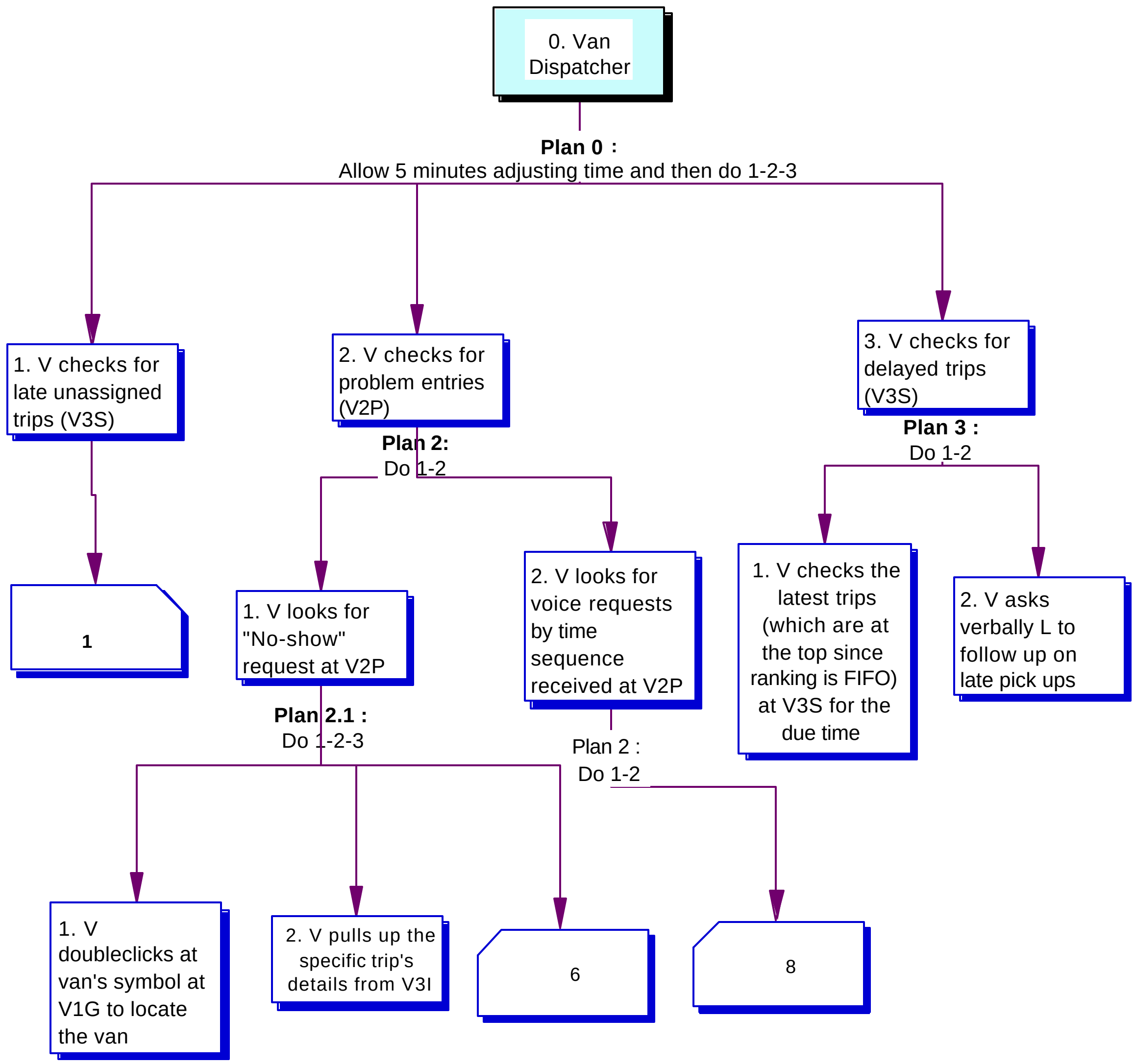
Termination Node indicates that the procedure (following depth first rule) has been completed at this point.

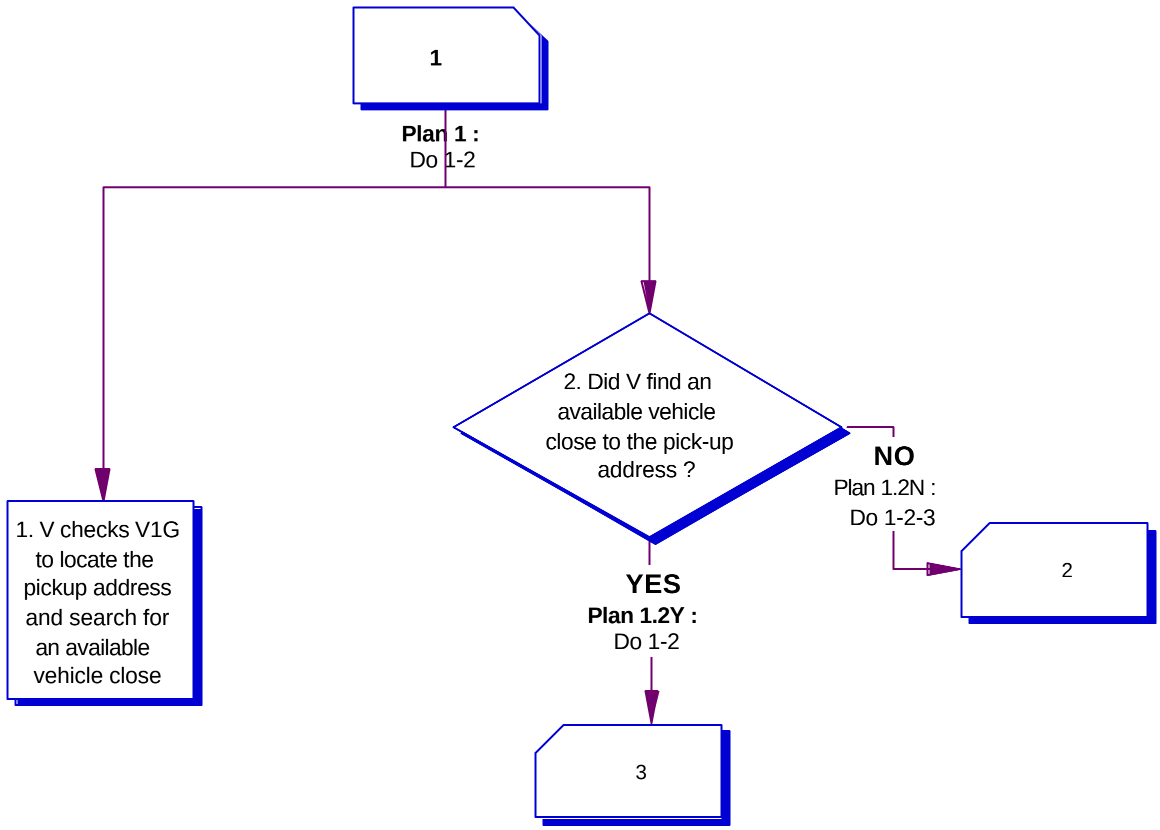


Either-Or Node indicates that at this level (point in time) only one node can be chosen and followed.

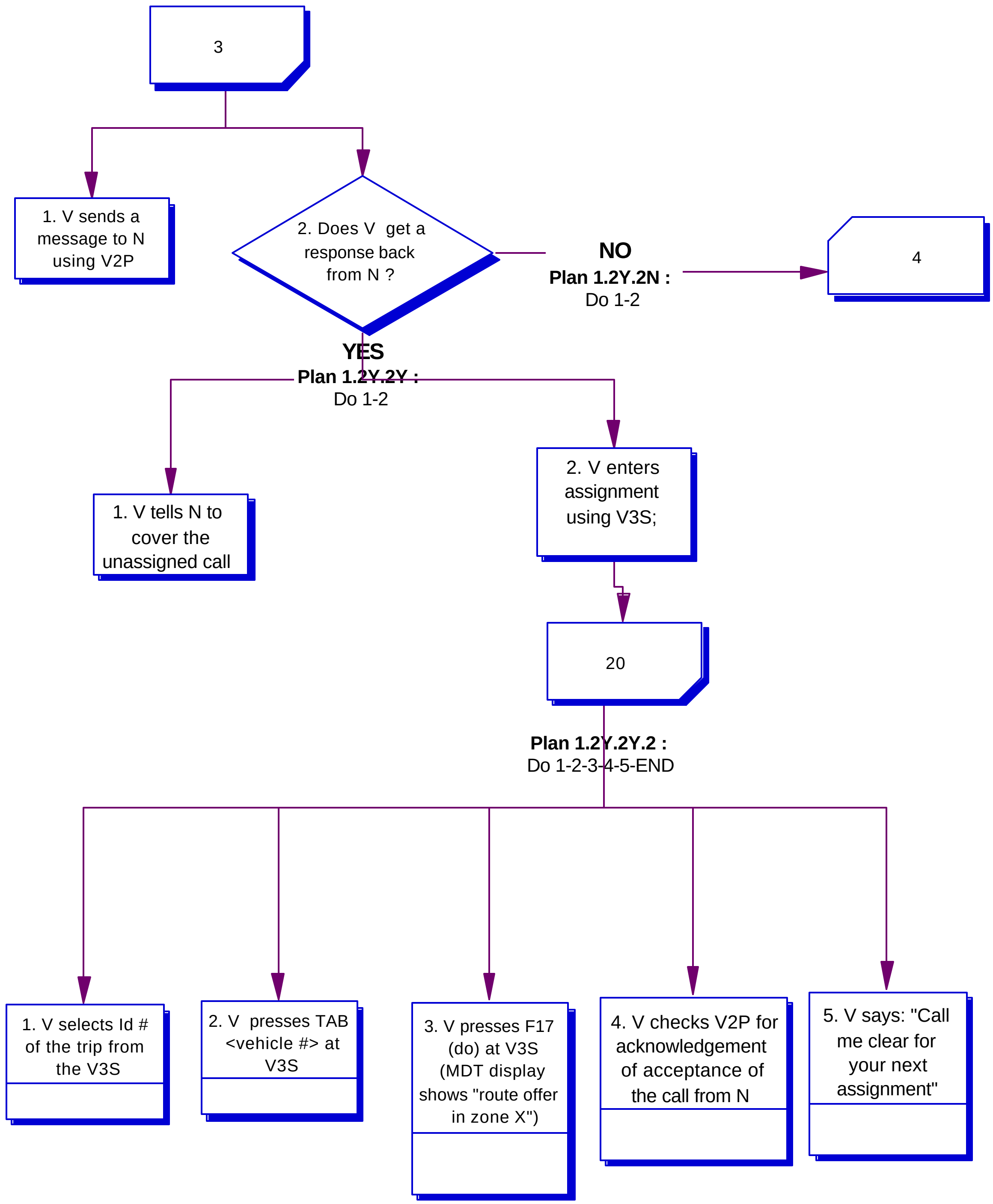


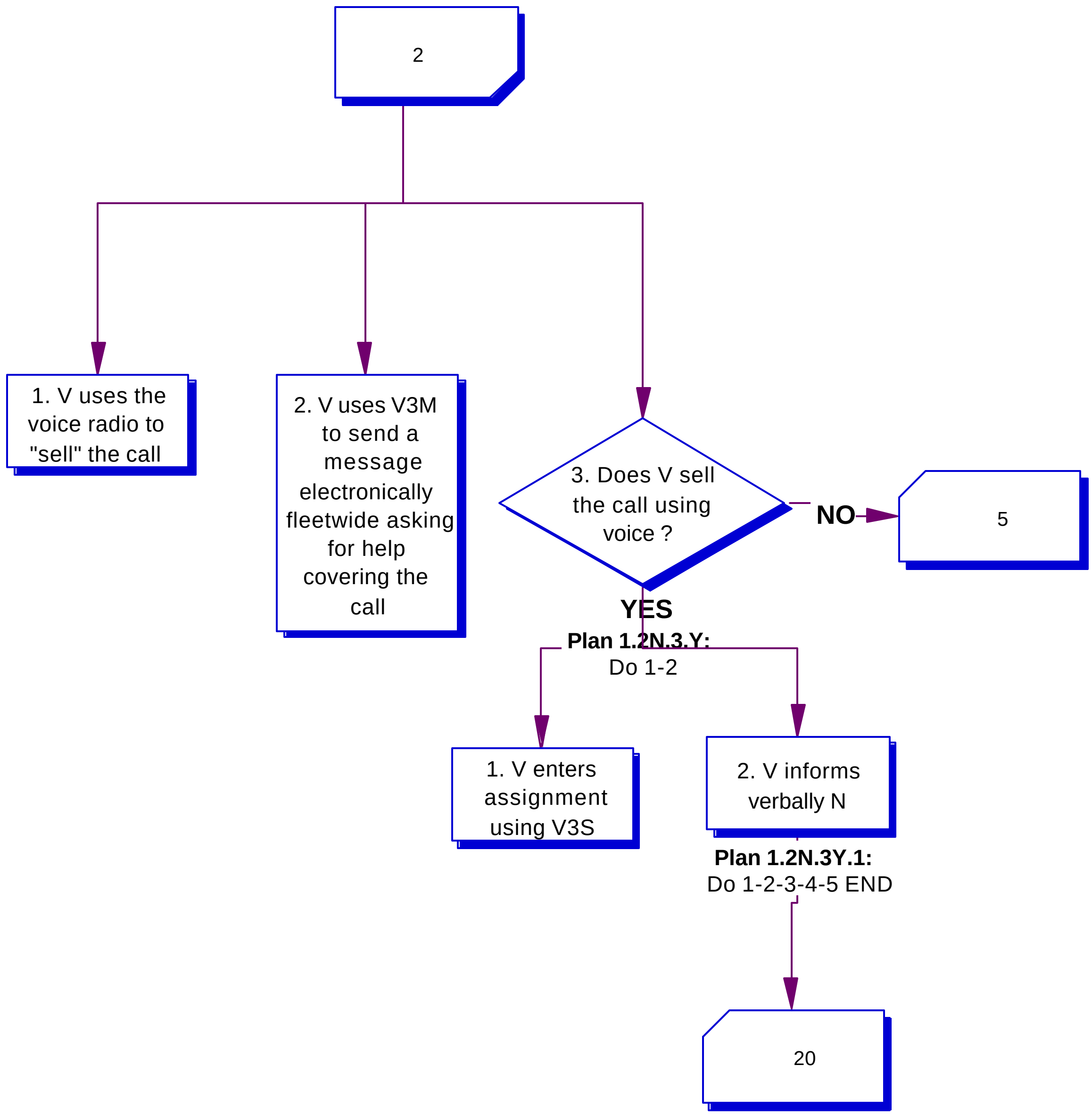
If-Then Node indicates that the path will be chosen depended on the answer to a question included in the node. The selection in this node is either Yes or No.

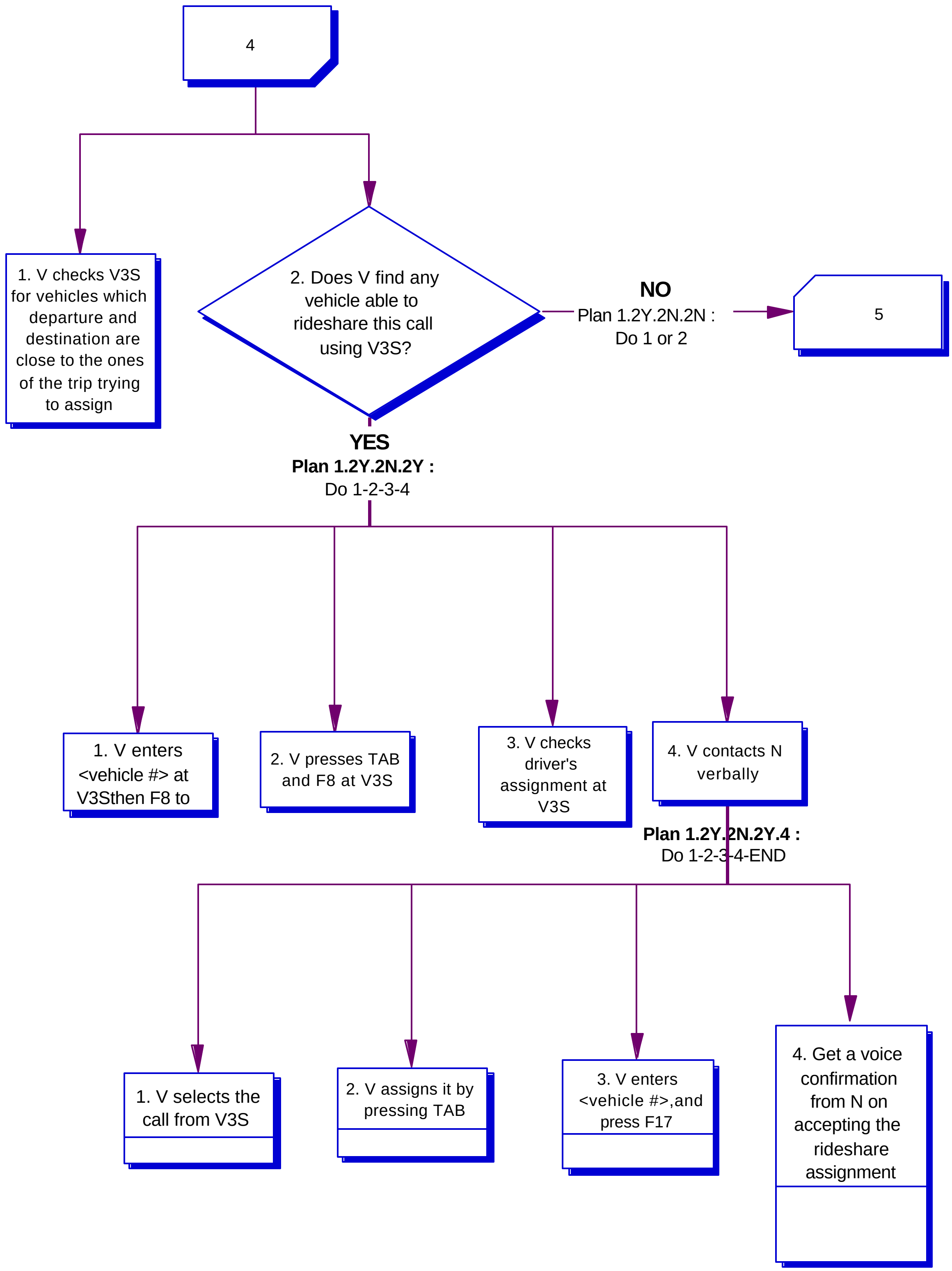

0. Van Dispatcher
Plan 0 :
Allow 5 minutes adjusting time and then do 1-2-3
1. V checks for late unassigned trips (V3S)
2. V checks for problem entries (V2P)
3. V checks for delayed trips (V3S)
Plan 2:
Do 1-2
Plan 3 :
Do 1-2
1
1. V looks for "No-show" request at V2P
2. V looks for voice requests by time sequence received at V2P
1. V checks the latest trips (which are at the top since ranking is FIFO) at V3S for the due time
2. V asks verbally L to follow up on late pick ups
Plan 2.1 :
Do 1-2-3
Plan 2 :
Do 1-2
1. V doubleclicks at van's symbol at V1G to locate the van
2. V pulls up the specific trip's details from V3I
6
8

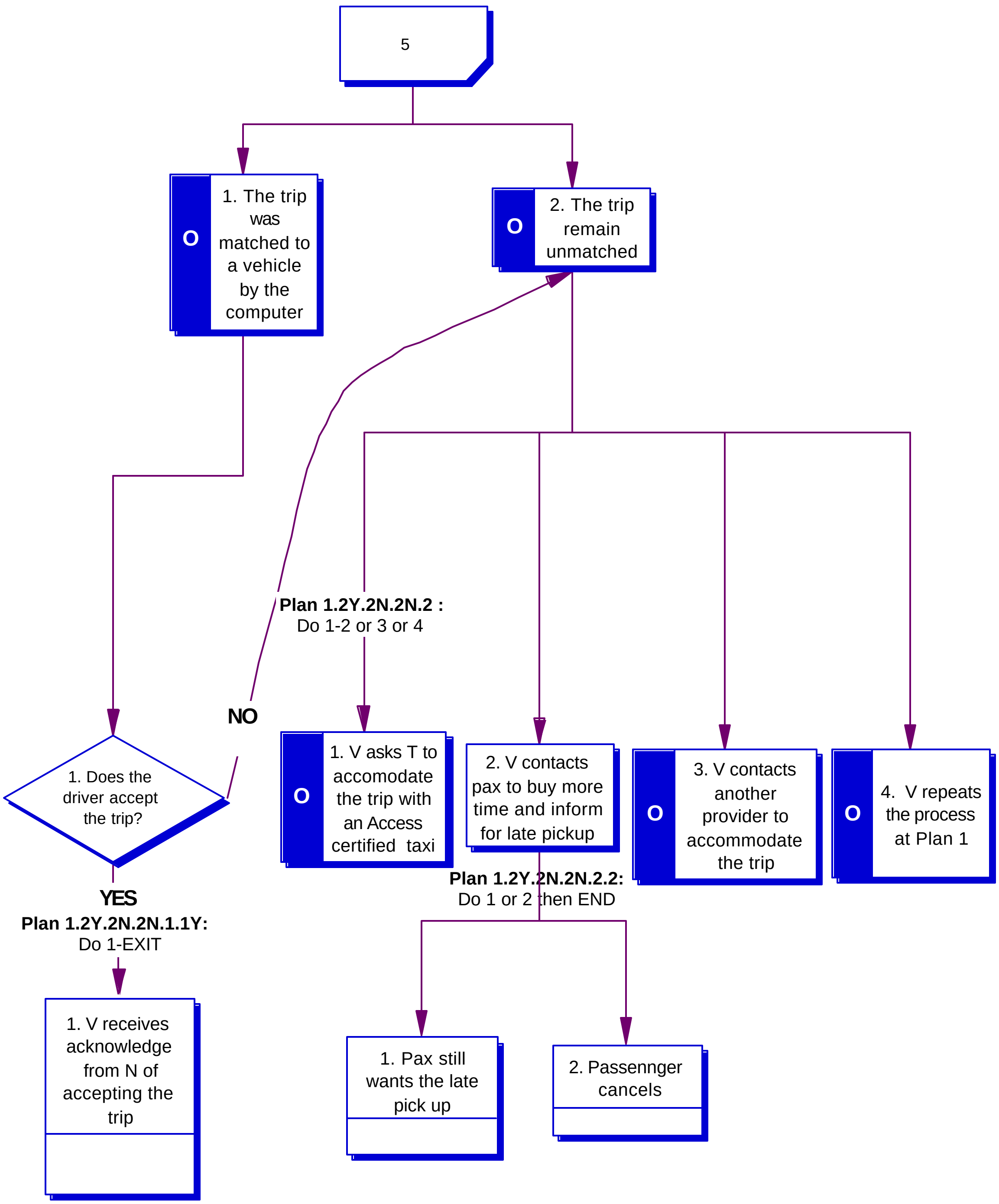


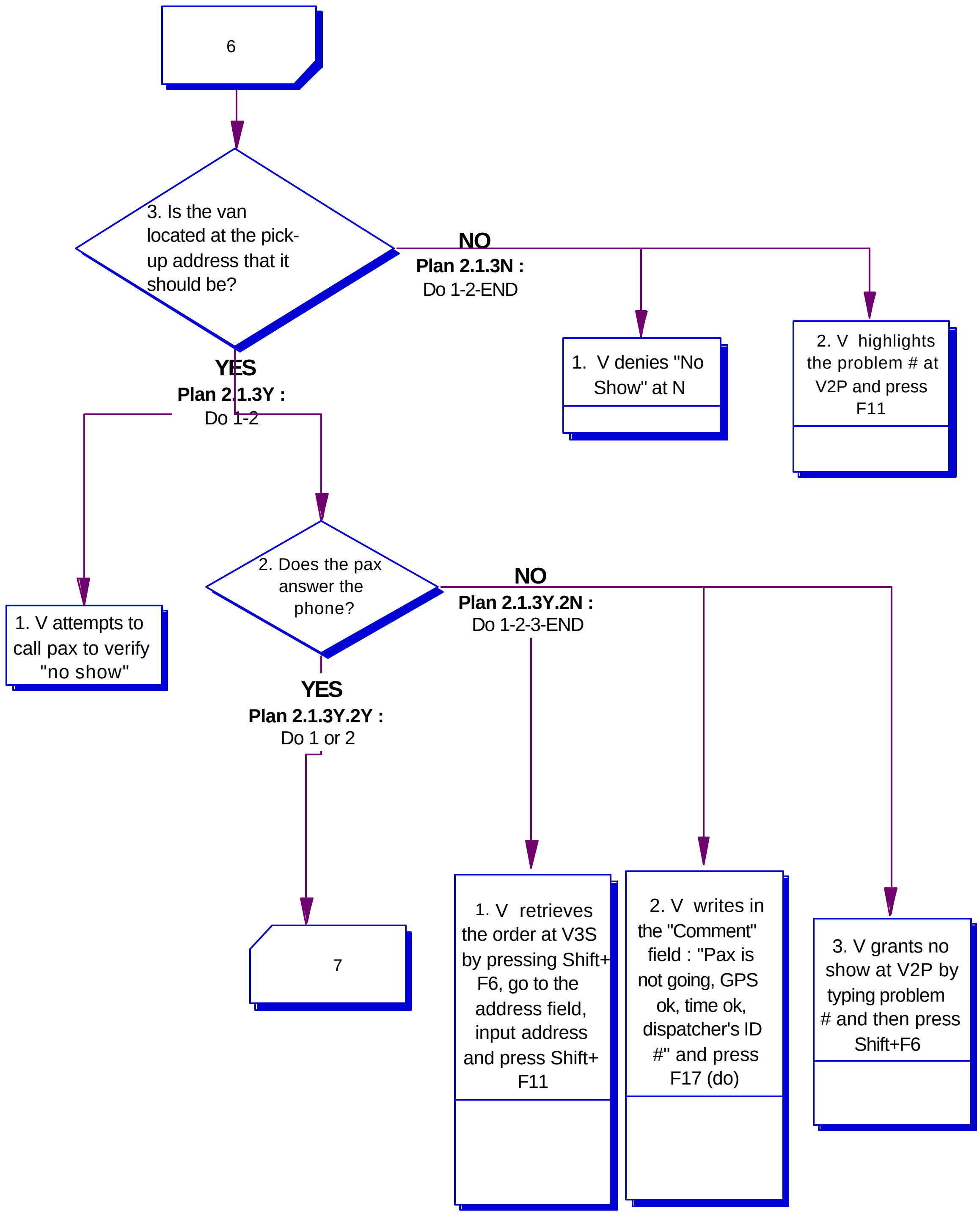
1
Plan 1 :
Do 1-2
2. Did V find an available vehicle close to the pick-up address ?
NO
Plan 1.2N :
Do 1-2-3
2
1. V checks V1G to locate the pickup address and search for an available vehicle close
YES
Plan 1.2Y :
Do 1-2
3


3
1. V sends a message to N using V2P
2. Does V get a response back from N ?
NO
Plan 1.2Y.2N :
Do 1-2
4
YES
Plan 1.2Y.2Y :
Do 1-2
1. V tells N to cover the unassigned call
2. V enters assignment using V3S;
20
Plan 1.2Y.2Y.2 :
Do 1-2-3-4-5-END
1. V selects Id # of the trip from the V3S
2. V presses TAB <vehicle #> at V3S
3. V presses F17 (do) at V3S (MDT display shows "route offer in zone X")
4. V checks V2P for acknowledgement of acceptance of the call from N
5. V says: "Call me clear for your next assignment"

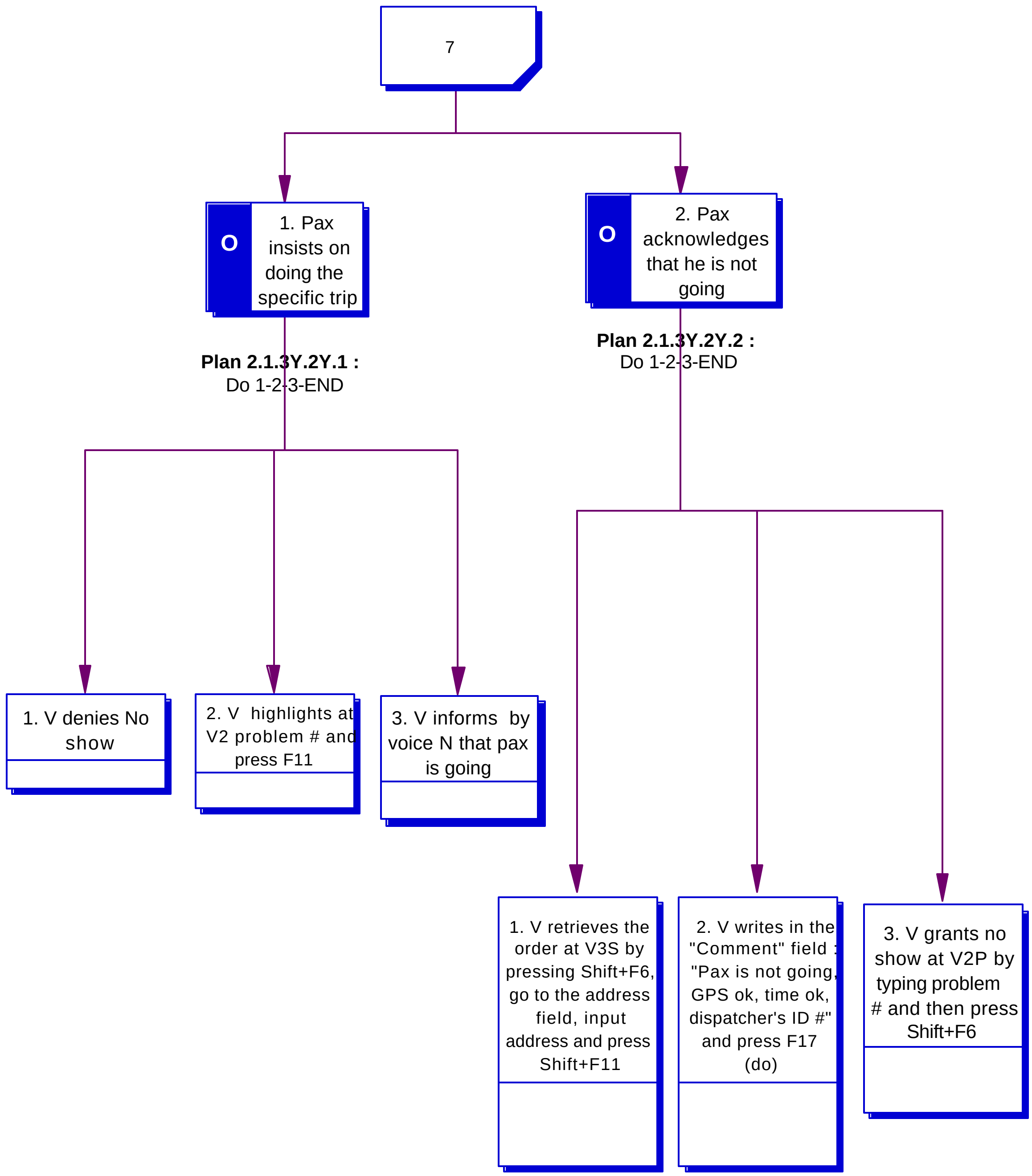

2
1. V uses the voice radio to "sell" the call
2. V uses V3M to send a message electronically fleetwide asking for help covering the call
3. Does V sell the call using voice ?
NO
5
YES
Plan 1.2N.3.Y:
Do 1-2
1. V enters assignment using V3S
2. V informs verbally N
Plan 1.2N.3Y.1:
Do 1-2-3-4-5 END
20

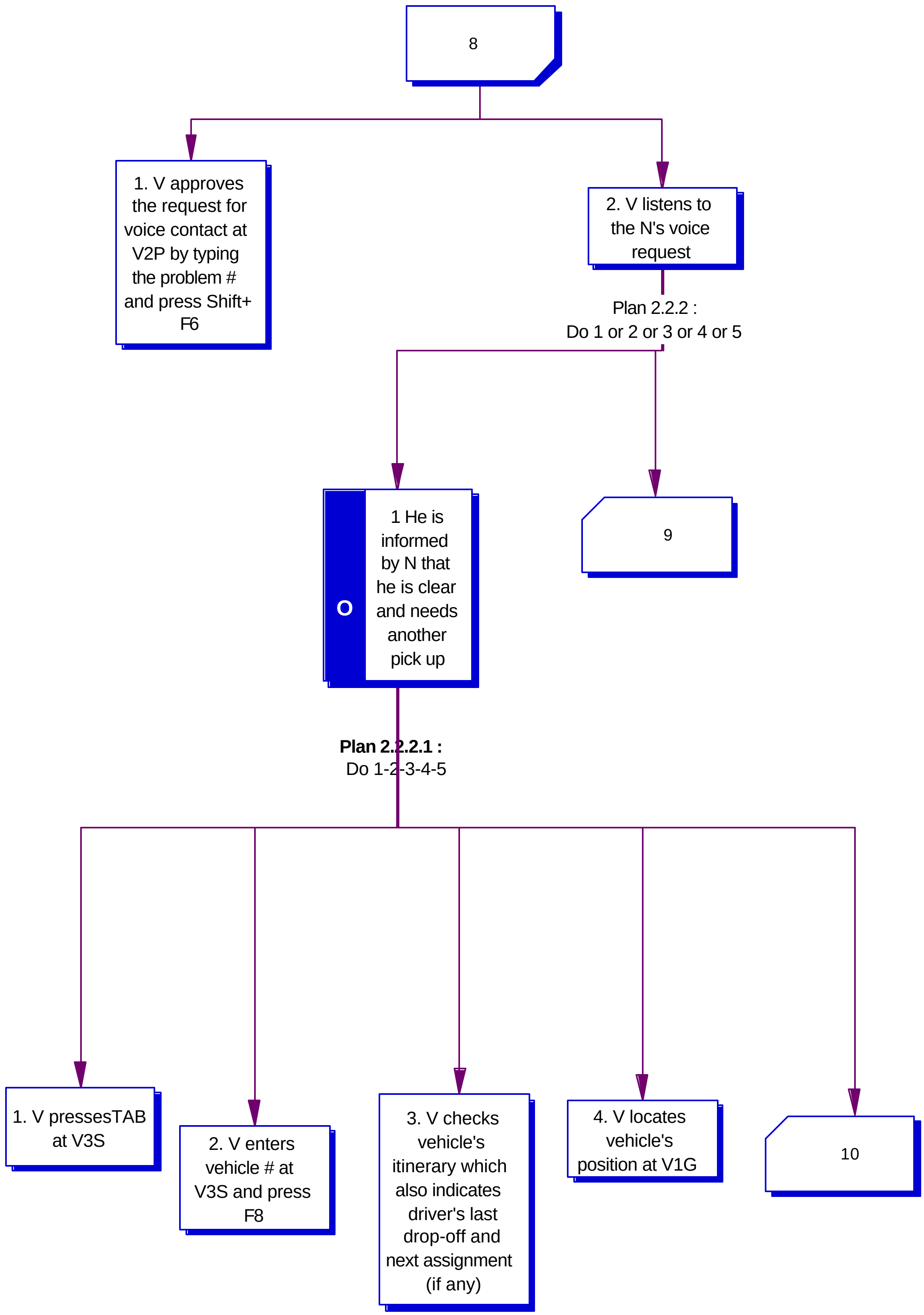

4
1. V checks V3S for vehicles which departure and destination are close to the ones of the trip trying to assign
2. Does V find any vehicle able to rideshare this call using V3S?
NO
Plan 1.2Y.2N.2N : Do 1 or 2
5
YES
Plan 1.2Y.2N.2Y : Do 1-2-3-4
1. V enters <vehicle #> at V3Sthen F8 to
2. V presses TAB and F8 at V3S
3. V checks driver's assignment at V3S
4. V contacts N verbally
Plan 1.2Y.2N.2Y.4 : Do 1-2-3-4-END
1. V selects the call from V3S
2. V assigns it by pressing TAB
3. V enters <vehicle #>,and press F17
4. Get a voice confirmation from N on accepting the rideshare assignment

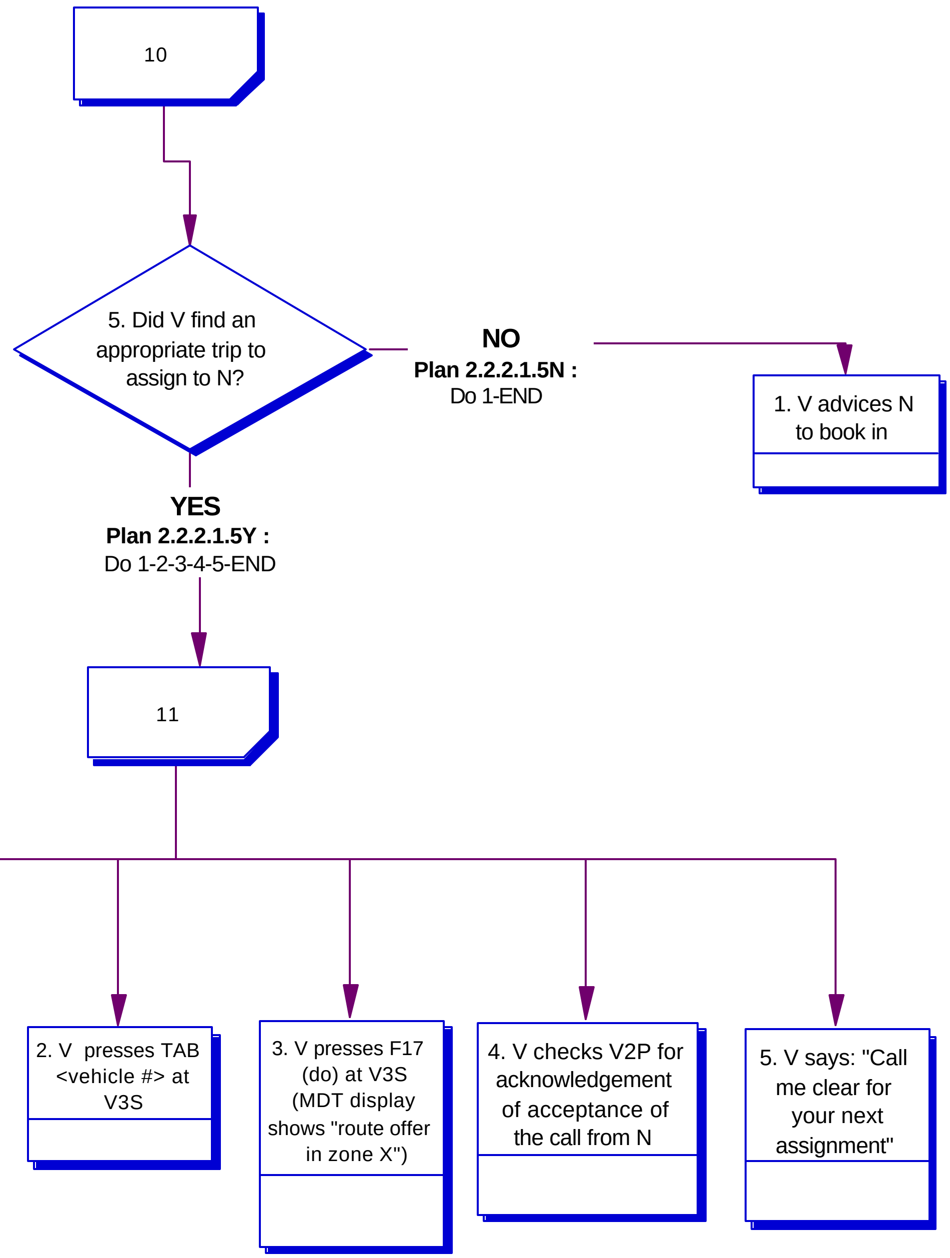

5
O
1. The trip was matched to a vehicle by the computer
O
2. The trip remain unmatched
1. Does the driver accept the trip?
NO
YES
Plan 1.2Y.2N.2N.1.1Y:
Do 1-EXIT
1. V receives acknowledge from N of accepting the trip
Plan 1.2Y.2N.2N.2 :
Do 1-2 or 3 or 4
O
1. V asks T to accomodate the trip with an Access certified taxi
2. V contacts pax to buy more time and inform for late pickup
O
3. V contacts another provider to accommodate the trip
O
4. V repeats the process at Plan 1
Plan 1.2Y.2N.2N.2.2:
Do 1 or 2 then END
1. Pax still wants the late pick up
2. Passennger cancels

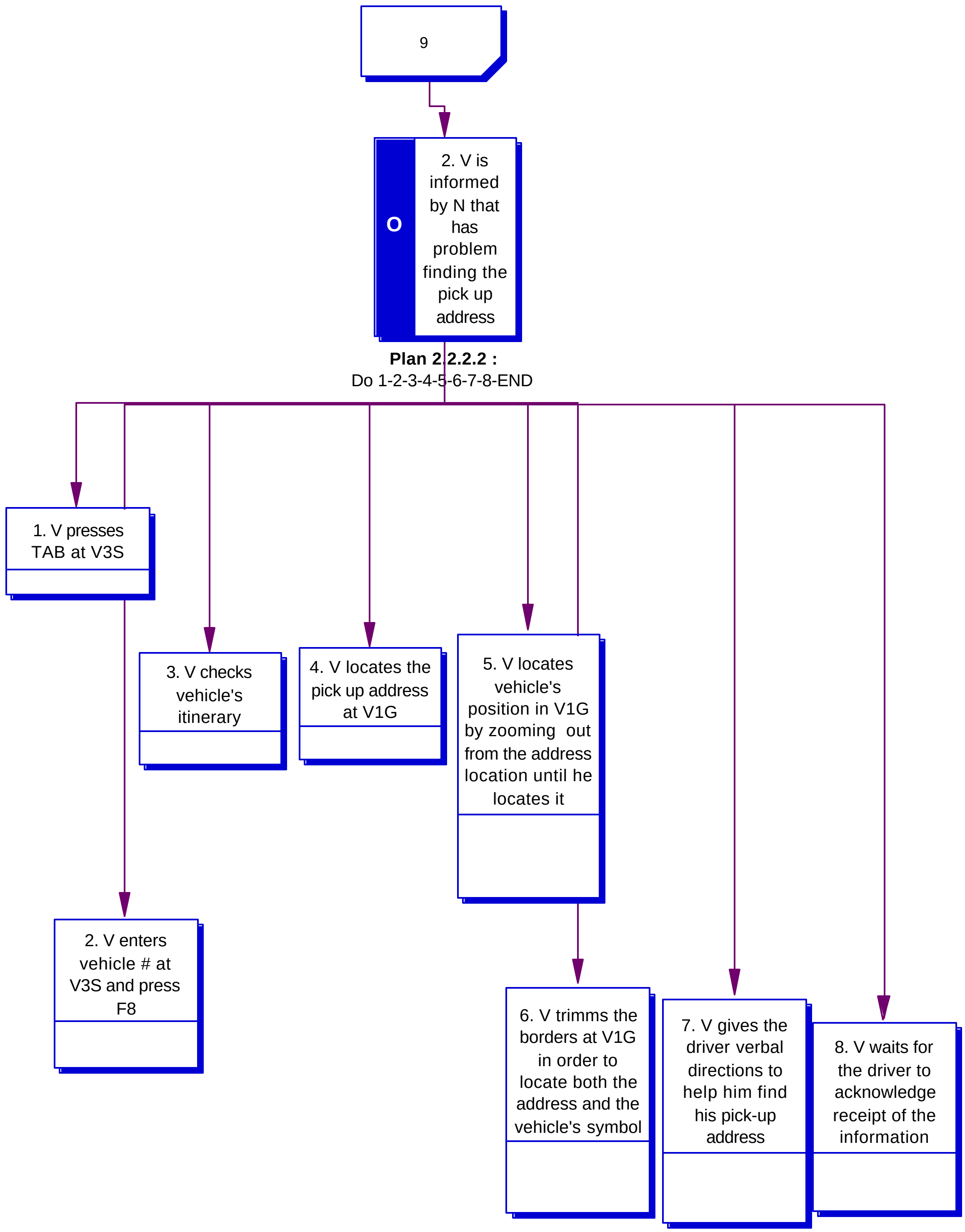


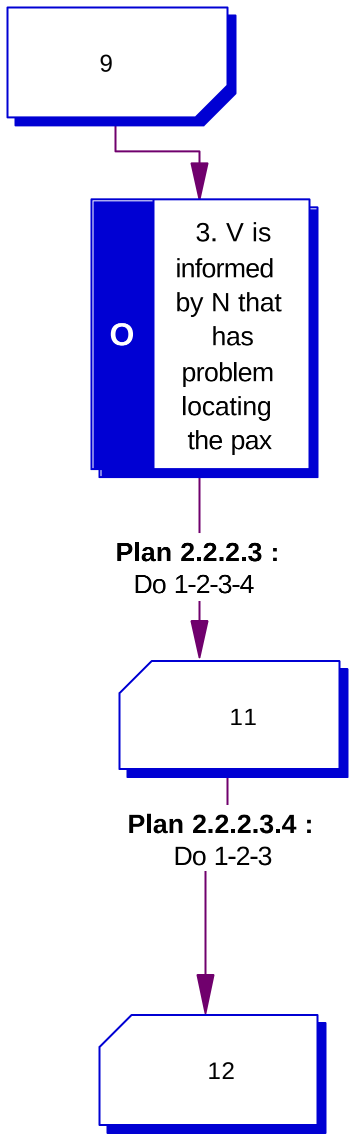
6
3. Is the van located at the pick-up address that it should be?
NO
Plan 2.1.3N :
Do 1-2-END
1. V denies "No Show" at N
2. V highlights the problem # at V2P and press F11
YES
Plan 2.1.3Y :
Do 1-2
1. V attempts to call pax to verify "no show"
2. Does the pax answer the phone?
NO
Plan 2.1.3Y.2N :
Do 1-2-3-END
YES
Plan 2.1.3Y.2Y :
Do 1 or 2
7
1. V retrieves the order at V3S by pressing Shift+ F6, go to the address field, input address and press Shift+ F11
2. V writes in the "Comment" field : "Pax is not going, GPS ok, time ok, dispatcher's ID #" and press F17 (do)
3. V grants no show at V2P by typing problem # and then press Shift+F6

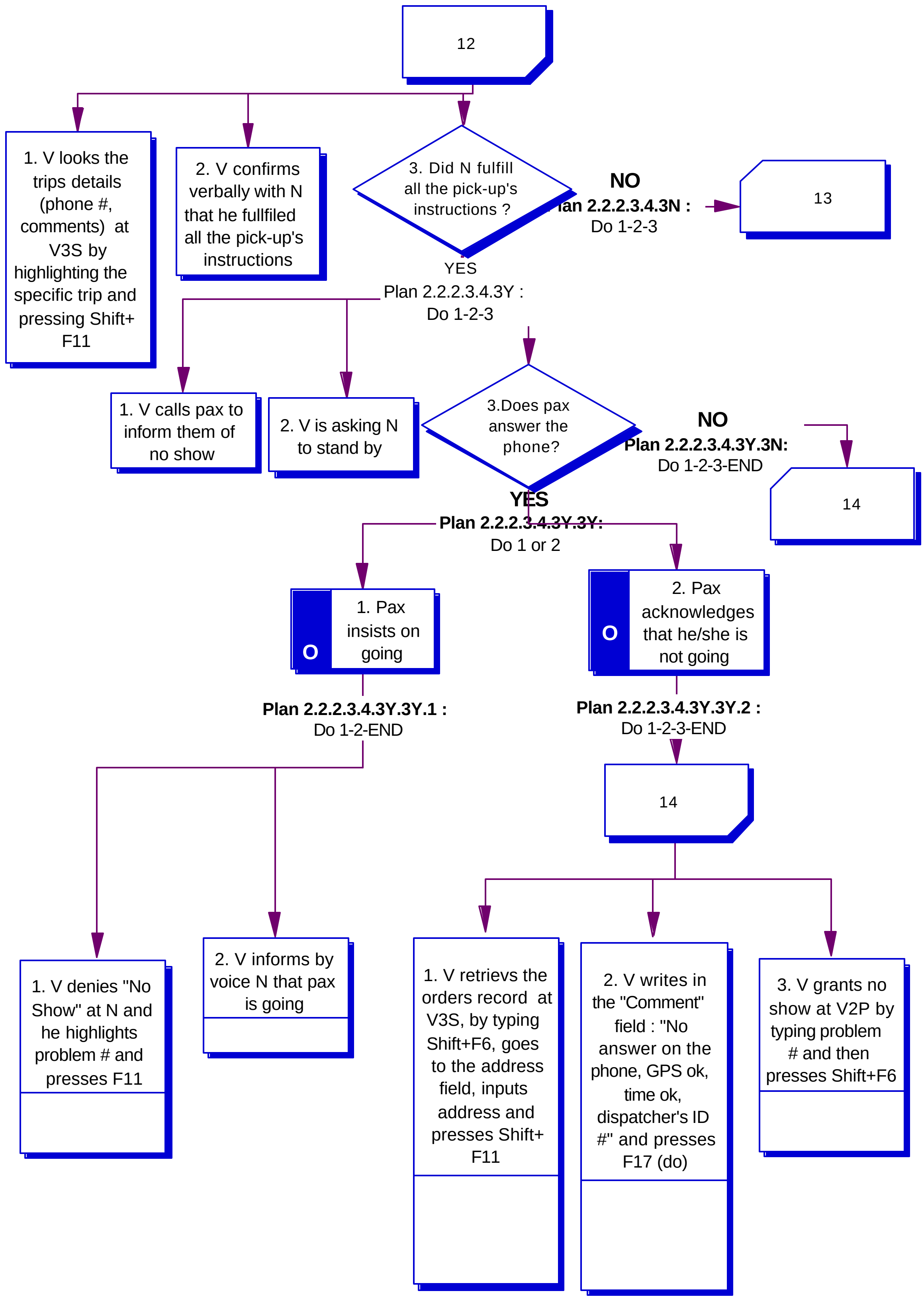

7
O 1. Pax insists on doing the specific trip
O 2. Pax acknowledges that he is not going
Plan 2.1.3Y.2Y.1 : Do 1-2-3-END
Plan 2.1.3Y.2Y.2 : Do 1-2-3-END
1. V denies No show
2. V highlights at V2 problem # and press F11
3. V informs by voice N that pax is going
1. V retrieves the order at V3S by pressing Shift+F6, go to the address field, input address and press Shift+F11
2. V writes in the "Comment" field "Pax is not going, GPS ok, time ok, dispatcher's ID #" and press F17 (do)
3. V grants no show at V2P by typing problem # and then press Shift+F6

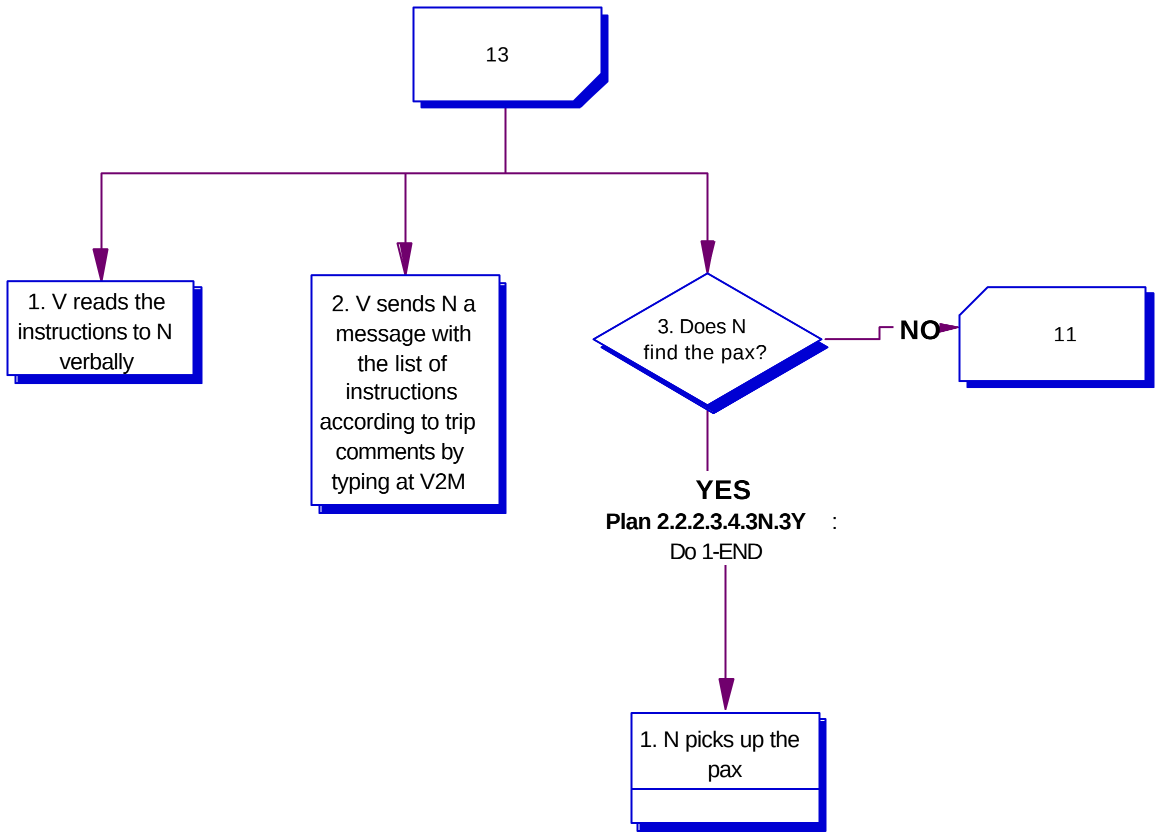

8
1. V approves the request for voice contact at V2P by typing the problem # and press Shift+ F6
2. V listens to the N's voice request
Plan 2.2.2 :
Do 1 or 2 or 3 or 4 or 5
O
1 He is informed by N that he is clear and needs another pick up
9
Plan 2.2.2.1 :
Do 1-2-3-4-5
1. V pressesTAB at V3S
2. V enters vehicle # at V3S and press F8
3. V checks vehicle's itinerary which also indicates driver's last drop-off and next assignment (if any)
4. V locates vehicle's position at V1G
10

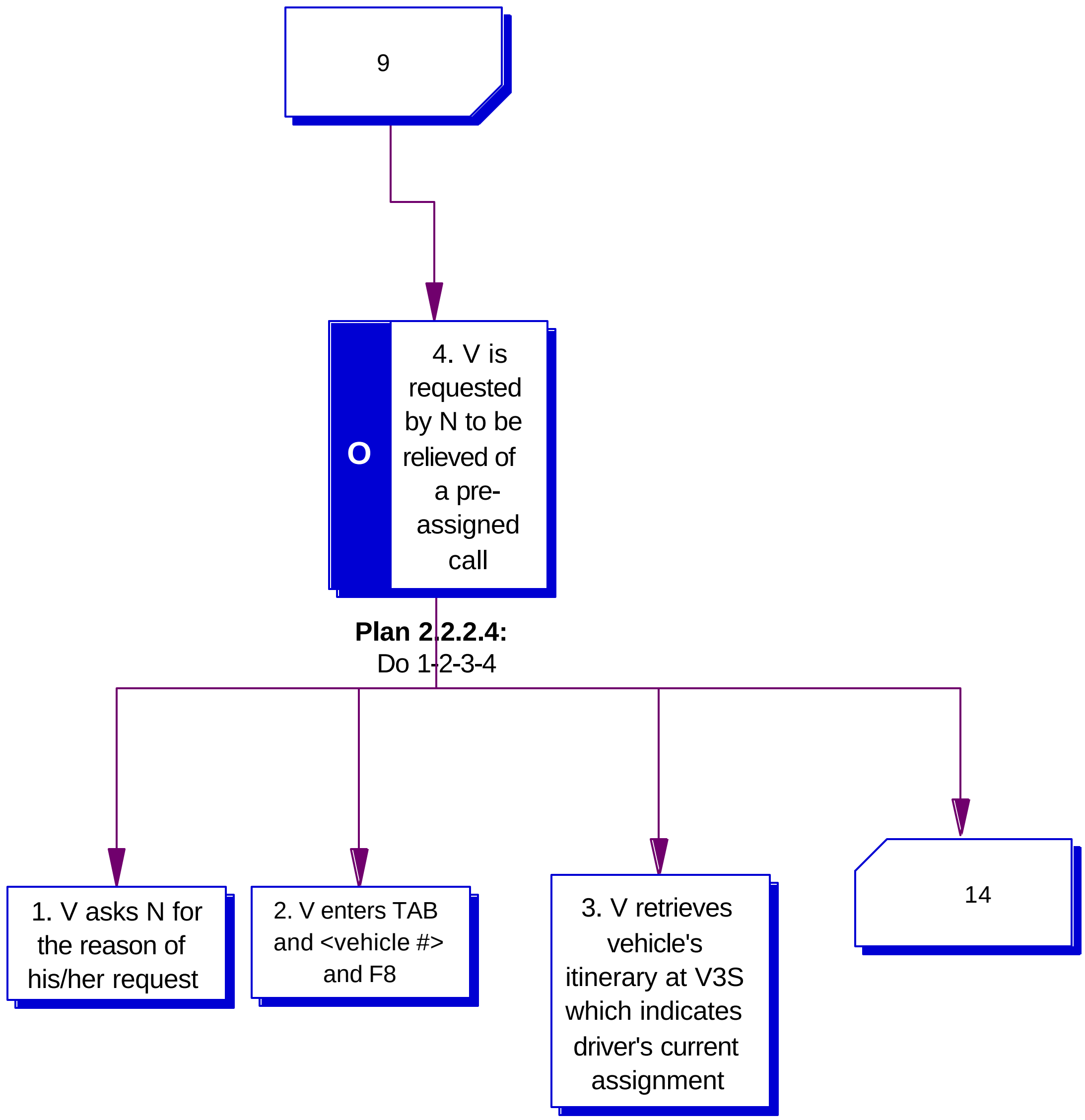

10
5. Did V find an appropriate trip to assign to N?
NO
Plan 2.2.2.1.5N :
Do 1-END
1. V advices N to book in
YES
Plan 2.2.2.1.5Y :
Do 1-2-3-4-5-END
11
1. V selects Id # of the trip from the V3S
2. V presses TAB <vehicle #> at V3S
3. V presses F17 (do) at V3S (MDT display shows "route offer in zone X")
4. V checks V2P for acknowledgement of acceptance of the call from N
5. V says: "Call me clear for your next assignment"

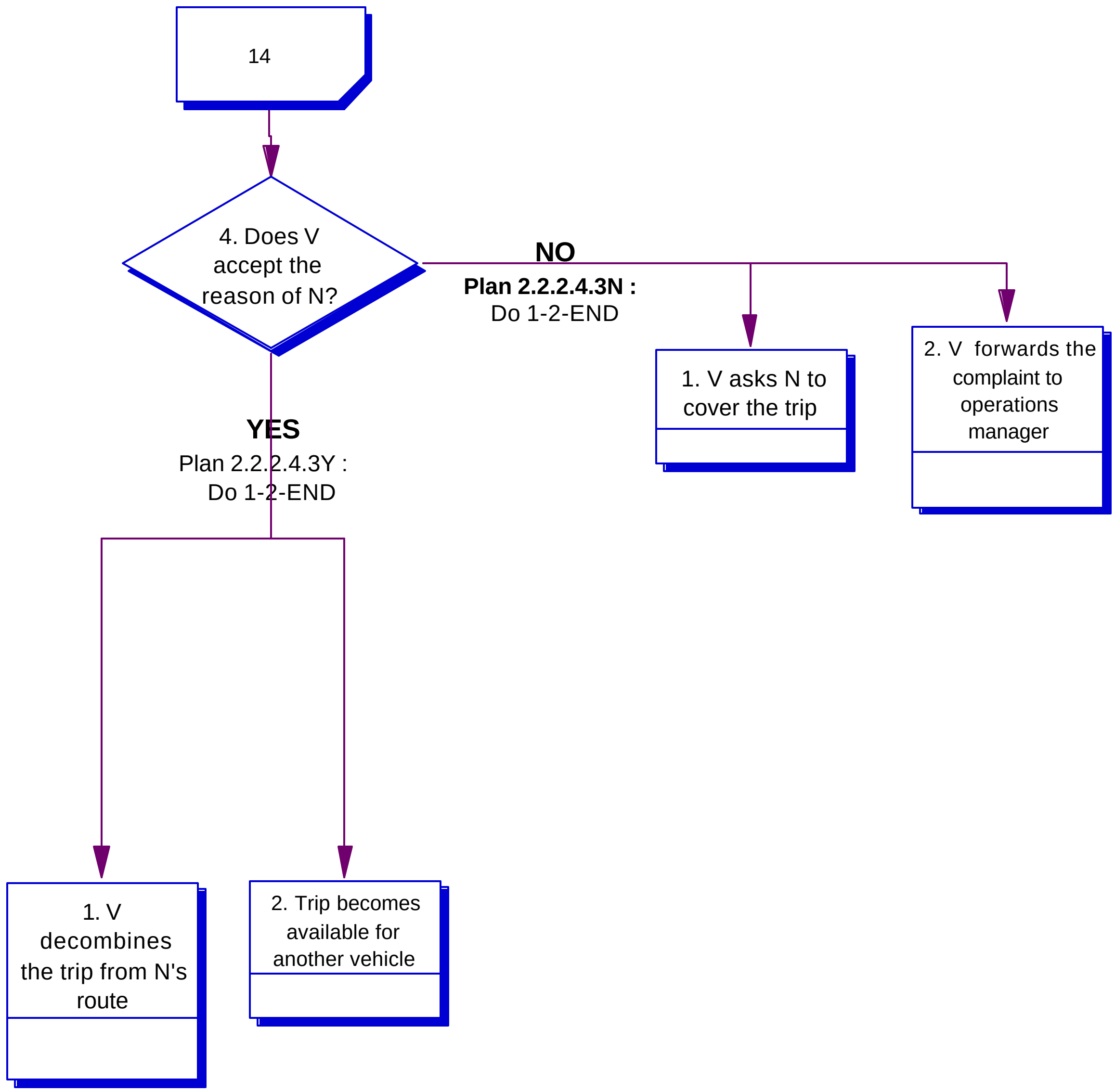

9
O
2. V is informed by N that has problem finding the pick up address
Plan 2.2.2.2 :
Do 1-2-3-4-5-6-7-8-END
1. V presses TAB at V3S
2. V enters vehicle # at V3S and press F8
3. V checks vehicle's itinerary
4. V locates the pick up address at V1G
5. V locates vehicle's position in V1G by zooming out from the address location until he locates it
6. V trimms the borders at V1G in order to locate both the address and the vehicle's symbol
7. V gives the driver verbal directions to help him find his pick-up address
8. V waits for the driver to acknowledge receipt of the information

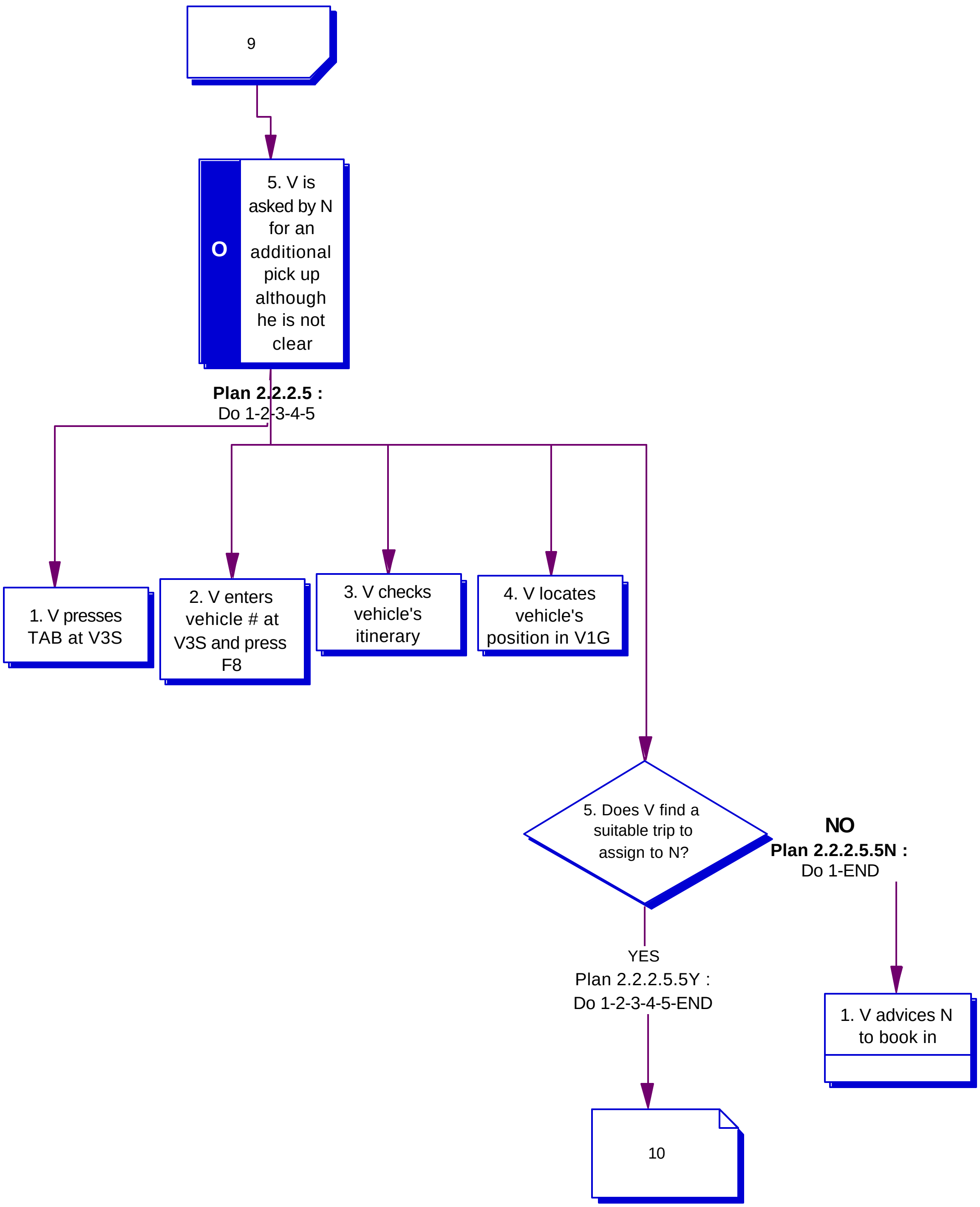

9
O
3. V is informed by N that has problem locating the pax
Plan 2.2.2.3 :
Do 1-2-3-4
11
Plan 2.2.2.3.4 :
Do 1-2-3
12

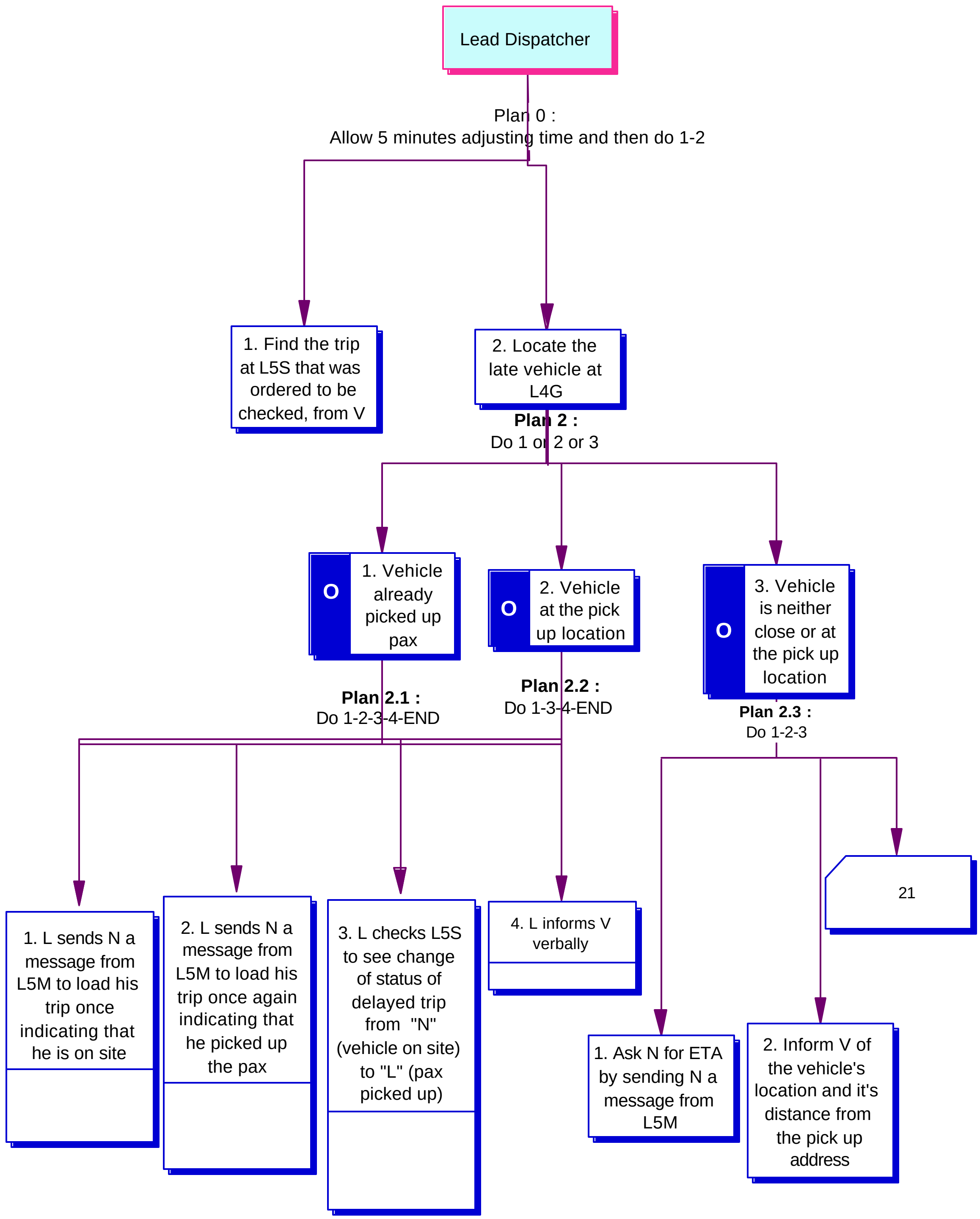

12
1. V looks the trips details (phone #, comments) at V3S by highlighting the specific trip and pressing Shift+ F11
2. V confirms verbally with N that he fullfiled all the pick-up's instructions
3. Did N fulfill all the pick-up's instructions ?
NO
Plan 2.2.2.3.4.3N :
Do 1-2-3
13
YES
Plan 2.2.2.3.4.3Y :
Do 1-2-3
1. V calls pax to inform them of no show
2. V is asking N to stand by
3.Does pax answer the phone?
NO
Plan 2.2.2.3.4.3Y.3N:
Do 1-2-3-END
14
YES
Plan 2.2.2.3.4.3Y.3Y:
Do 1 or 2
O
1. Pax insists on going
O
2. Pax acknowledges that he/she is not going
Plan 2.2.2.3.4.3Y.3Y.1 :
Do 1-2-END
Plan 2.2.2.3.4.3Y.3Y.2 :
Do 1-2-3-END
14
1. V denies "No Show" at N and he highlights problem # and presses F11
2. V informs by voice N that pax is going
1. V retrievs the orders record at V3S, by typing Shift+F6, goes to the address field, inputs address and presses Shift+ F11
2. V writes in the "Comment" field : "No answer on the phone, GPS ok, time ok, dispatcher's ID #" and presses F17 (do)
3. V grants no show at V2P by typing problem # and then presses Shift+F6

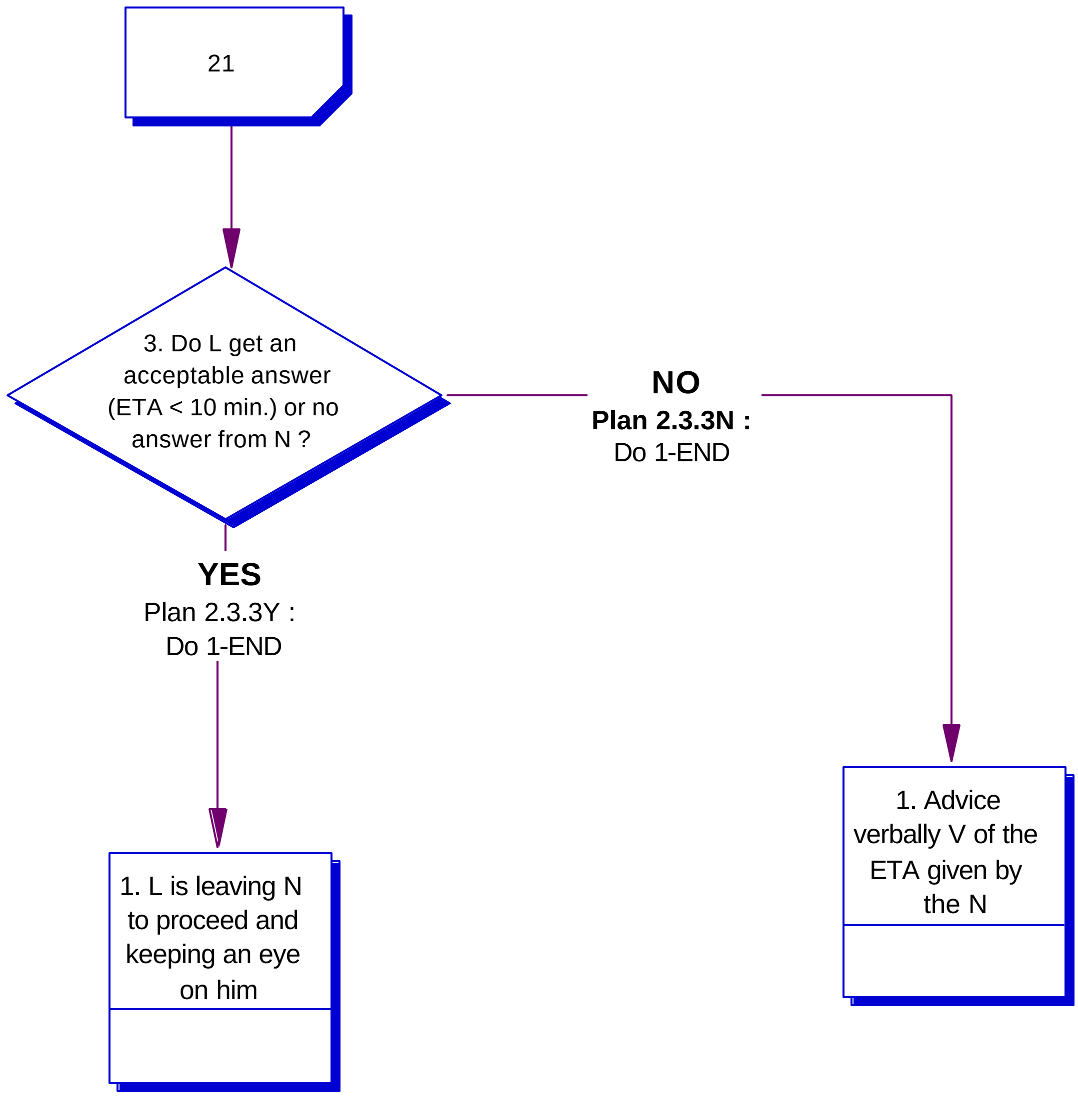

13
1. V reads the instructions to N verbally
2. V sends N a message with the list of instructions according to trip comments by typing at V2M
3. Does N find the pax?
NO
11
YES
Plan 2.2.2.3.4.3N.3Y :
Do 1-END
1. N picks up the pax


9
O
4. V is requested by N to be relieved of a pre-assigned call
Plan 2.2.2.4:
Do 1-2-3-4
1. V asks N for the reason of his/her request
2. V enters TAB and <vehicle #> and F8
3. V retrieves vehicle's itinerary at V3S which indicates driver's current assignment
14


14
4. Does V accept the reason of N?
NO
Plan 2.2.2.4.3N :
Do 1-2-END
1. V asks N to cover the trip
2. V forwards the complaint to operations manager
YES
Plan 2.2.2.4.3Y :
Do 1-2-END
1. V decombines the trip from N's route
2. Trip becomes available for another vehicle


9
O
5. V is asked by N for an additional pick up although he is not clear
Plan 2.2.2.5 :
Do 1-2-3-4-5
1. V presses TAB at V3S
2. V enters vehicle # at V3S and press F8
3. V checks vehicle's itinerary
4. V locates vehicle's position in V1G
5. Does V find a suitable trip to assign to N?
NO
Plan 2.2.2.5.5N :
Do 1-END
1. V advices N to book in
YES
Plan 2.2.2.5.5Y :
Do 1-2-3-4-5-END
10


Lead Dispatcher
Plan 0 :
Allow 5 minutes adjusting time and then do 1-2
1. Find the trip at L5S that was ordered to be checked, from V
2. Locate the late vehicle at L4G
Plan 2 :
Do 1 or 2 or 3
O
1. Vehicle already picked up pax
O
2. Vehicle at the pick up location
O
3. Vehicle is neither close or at the pick up location
Plan 2.1 :
Do 1-2-3-4-END
Plan 2.2 :
Do 1-3-4-END
Plan 2.3 :
Do 1-2-3
21
1. L sends N a message from L5M to load his trip once indicating that he is on site
2. L sends N a message from L5M to load his trip once again indicating that he picked up the pax
3. L checks L5S to see change of status of delayed trip from "N" (vehicle on site) to "L" (pax picked up)
4. L informs V verbally
1. Ask N for ETA by sending N a message from L5M
2. Inform V of the vehicle's location and it's distance from the pick up address


21
3. Do L get an acceptable answer (ETA < 10 min.) or no answer from N ?
NO
Plan 2.3.3N :
Do 1-END
YES
Plan 2.3.3Y :
Do 1-END
1. L is leaving N to proceed and keeping an eye on him
1. Advice verbally V of the ETA given by the N

LEGEND:
Code attributes: <PERSON, DEVICE, SCREEN>

PERSON

V : Van Dispatcher
T : Taxi Dispatcher
L : Lead Dispatcher
N: Van Driver
X : Taxi Driver

DEVICE

1: Van's GPS Monitor
2: Van's left DOS
3: Van's right DOS
4: Lead's GPS Monitor
5: Lead's DOS
6: Taxi's left DOS
7: Taxi's right DOS

SCREEN

S : Share-ride screen
P : Problem screen
I : Trip detail
J : Trip summary
O : Order-taker screen
R : Routing screen
G : GPS screen
M : Message screen

REFERENCES

Annett, J., Duncan, K.D., Stammers, R.B., and Gary, M.J. (1971). *Task Analysis*. London: HMSO Publishers.

Cook, R. I. and Woods, D. D. (1996). Adapting to new technology in the operating room. *Human Factors*, 38(4): 593.

Dempsey, P.S. (1991). The Civil Rights of the handicapped in transportation: The Americans with Disability Act and related legislation. *Transportation Law Journal*, 19(2): 309-333.

Dessouky, M. M., and Adam, S. (1998). Real-time scheduling for demand responsive transit systems. *Proceedings of the 1998 Conference on IEEE Systems, Man, and Cybernetics*, San Diego, CA.

Diaper, D., (Ed.) (1989). *Task Analysis for Human-Computer Interaction*. Chichester: Ellis Horwood.

Grossberg, M., Wiesen, R. A., and Yntema, D. B. (1976). An experiment on problem solving with delayed computer responses. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(3): 219-222.

Hemenway, K. (1982). Psychological issues in the use of icons in command menus. *Proceedings: Human Factors In Computing Systems* (Gaithersburg, MD), 20-25. New York: Association of Computing Machinery.

Hoffman, R.R., Crandall, B., and Shadbolt, N. (1998). Use of the critical decision method to elicit expert knowledge: A case study in the methodology of cognitive task analysis. *Human Factors*, 40(2): 254-276.

Kirwan, B., and Ainsworth, L.K. (Eds.) (1992). *A Guide to Task Analysis*. London: Taylor and Francis.

Lave, R. E., Teal, R., and Piras, P. (1996). A Handbook for Acquiring Demand-Responsive Transit Software," Transit Cooperative Research Program Report #18, Transportation Research Board, Washington, D. C.

Norman, D. A. (1988). *The Design of Everyday Things*. New York, Basic Books.

Norman, D. A. (1999). *Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex and Information Appliances Are the Solution*. Boston, MA: MIT Press.

Obradovich, J. H. and Woods, D. D. (1996). Users as designers: How people cope with poor HCI design in computer-based medical devices. *Human Factors*, 38(4): 574.

Olson, J.R. and Olson, G. M. (1990). The growth of cognitive modeling in human computer interaction since GOMS. *Human Computer Interaction,* 5: 221-266.

Pritsker, A. A. B. and O'Reilly, J. J. (1999). *Simulation with Visual SLAM and AweSim,* second edition Edition, New York: John Wiley & Sons.
Potter, S. S. and Woods, D. D. (1994). Breading Down Barriers in Cooperative Fault Management: Temporal and Functional Information Displays. Cognitive Systems Engineering Laboratory Report, CSEL 94-TR-02, Columbus, OH: The Ohio State University.

Rahimi, M. and Azevedo, G. (1990). A task analysis of industrial robot teach programming. *Proceedings, 2nd International Conference on Human Aspects of Advanced Manufacturing and Hybrid Automation*. Honolulu, Hawaii.

Reierson, C.S.., Marshall, E., and Baker, S.M. (1988). An experimental evaluation of an advanced alarm system for nuclear power plants. In J. Patrick and K. Duncan (Eds.), *Training, Human Decision Making and Control*. North-Holland: Elsevier Science Publishers.

Rohr G. and Keppel, E. (1984). Iconic interfaces: Where to use and how to construct. In *Human Factors in Organizational Design and Management*, 269-275. North Holland: Elsevier Science Publishers.

Sarter, N. B., Woods, D. D. and Billings, C. (1997). Automation surprises. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics,* Second Edition, New York: Wiley.

Shepherd, A. (1993). An approach to information requirement specification for process control tasks. *Ergonomics,* 31(36): 805-817.

Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, (Third Edition). New York: Addison Wesley.

Stone, J.R., Nalevanko, A. M., and Gilbert, G. (1994). Computer dispatch and scheduling for paratransit: An application of advanced public transportation systems. *Transportation Quarterly*, 48(2): 173-184.

Tullis, T.S. (1986). Optimizing the usability of computer-generated displays. *Proceedings of HCI'86 Conference on People and Computers: Designing for Usability* (York, England). London: British Computer Society.

Tullis, T.S. (1997). Screen Design. In M. G. Helander, T.K. Landauer and P. V. Prabhu (Eds.): *Handbook of Human-Computer Interaction,* Second, completely revised edition, (Chapter 23). Amsterdam: Elsevier Science B.V.

Woods, D. D. (1993). The price of flexibility in intelligent interfaces. *Knowledge/Based Systems,* 6: 1-8.

Woods D.D. and Watts J.C. (1997). How not to have to navigate through too many displays. In M. G. Helander, T.K. Landauer and P. V. Prabhu (Eds.): *Handbook of Human-Computer Interaction,* Second, completely revised edition, (Chapter 26). Amsterdam: Elsevier Science B.V.