Title

TRACER: In-Vehicle, GPS-Based Wireless Technology for Traffic Surveillance and Management.

Authors

McNally, Michael G.
Marca, James E.
Rindt, Craig
et al.

Publication Date

2003-07-01

TRACER: In-Vehicle, GPS-Based, Wireless Technology for Traffic Surveillance and Management

UCI-ITS-AS-WP-03-2

Michael G. McNally
James E. Marca
Craig R. Rindt
Angela M. Koos

Department of Civil and Environmental Engineering and
Institute of Transportation Studies, University of California, Irvine
mmcnally@uci.edu, jmarca@uci.edu, crindt@uci.edu

July 2003

Institute of Transportation Studies
University of California, Irvine
Irvine, CA 92697-3600, U.S.A.
http://www.its.uci.edu

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

TRACER: In-vehicle, GPS-based, Wireless Technology for Traffic Surveillance and Management

M.G. McNally, J.E. Marca, C.R. Rindt, A.M. Koos
University of California, Irvine

California PATH Research Report
UCB-ITS-PRR-2003-23

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

Final Report for Task Order 4120

July 2003

ISSN 1055-1425

PATH MOU 4120

TRACER: In-vehicle, GPS-based, Wireless Technology for Traffic Surveillance and Management

FINAL REPORT

M.G. McNally, J. E. Marca, C. R. Rindt, and A. M. Koos

Institute of Transportation Studies and
Department of Civil & Environmental Engineering
University of California Irvine
Irvine, CA 92697-3600

April 2003

PATH MOU 4120

TRACER: In-vehicle, GPS-based, Wireless Technology for Traffic Surveillance and Management

M. G. McNally, J. E. Marca, C. R. Rindt, and A. M. Koos

Institute of Transportation Studies and
Department of Civil & Environmental Engineering
University of California Irvine
Irvine, CA 92697-3600

ABSTRACT

The fundamental principle of intelligent transportation systems is to match the complexity of travel demands with advanced supply-side analysis, evaluation, management, and control strategies. A fundamental limitation is the lack of basic knowledge of travel demands at the network level. Modeling and sensor technology is primarily limited to aggregate parameters or micro-simulations based on aggregate distributions of behavior. Global Positioning Systems (GPS) are one of several available technologies which allow individual vehicle trajectories to be recorded and analyzed. Potential applications of GPS which are relevant to the ATMS Testbed are implementation in probe vehicles to deliver real-time performance data to complement loop and other sensor data and implementation in vehicles from sampled households to record route choice behavior. An **E**xtensible GPS-based in-vehicle **D**ata **C**ollection **U**nit (EDCU) has been designed, tested, and applied in selected field tests. Each unit incorporates GPS, data logging capabilities, two-way wireless communications, and a user interface in an extensible system which eliminates driver interaction. Together with supporting software, this system is referred to as TRACER. The design and initial implementation tests Testbed are presented herein. This research is a contination in PATH MOU 3006; se;lected portions of the interim report for that MOU are repeated here to provide a complete overview of the research effort.

Key Words: In-vehicle data logger, GPS, CDPD, TRACER, vehicle probes

CONTENTS

1 OVERVIEW 1

1.1 Project Goals 1

1.2 Project Status 2

1.3 Report Organization 3

2 LITERATURE REVIEW 4

2.1 GPS Applications in Probe Vehicle Surveillance 4

2.2 GPS Applications in Congestion Management 5

2.3 GPS Applications in Historical Database Generation 6

2.4 GPS Applications in Fleet Management 6

2.5 GPS Applications in Travel Surveys 7

2.6 GPS Applications and Map Matching 8

2.7 GPS Applications and User Interaction 8

2.8 GPS Applications Integrated with Web-based Travel Surveys 11

3 THE TRACER SYSTEM 13

3.1 Application Requirements 13

3.1.1 Basic Survey 13
3.1.2 Probe Vehicle Study 14
3.1.3 Enhanced Survey 14
3.1.4 Route Guidance Study 15

3.2 TRACER System Specifications 15

3.3 TRACER Support Software 20

4 EVALUATING EDCU FUNCTIONALITY 21

4.1 Preliminary Map Matching of Vehicle Tracings 21

4.2 Utilization with a Probe Vehicle 21

4.3 Internet Access 24

4.4 Integration with GIS Programs 25

4.5 Software Platform Extensibility 26

4.6 Two Representative Applications 27

4.6.1 REACT! 27
4.6.2 TRICEPS 27

5 ANALYSIS OF ROUTE TRACKING DATA 32

5.1 Route Usage Behavior 32

5.2 Observing Traffic Flow Variables 36

5.3 Matching Observations to a Network Representation 38

5.4 Automatic Mapping of the Transportaiton Network 39

5.5 **Aggregating Observations over Time and Space** 42
5.6 **Summary** 44

6 TRACER MODULE TECHNICAL SPECIFICATIONS 46

6.1 **DUC Module** 47
  6.1.1 Computing Platform 48
  6.1.2 Device Hardware Interfaces 48
  6.1.3 Device API 48
  6.1.4 Application Control Program 48
  6.1.5 Power Management 49
6.2 **GPS Module** 49
  6.2.1 Satellite Signal 50
  6.2.2 Antenna Placement 50
  6.2.3 Positioning Accuracy 50
  6.2.4 Application Programming Interface 50
  6.2.5 Power Management 50
6.3 **LOG Module** 50
  6.3.1 Hardware 51
  6.3.2 Application Programming Interface 51
  6.3.3 Data Storage Capacity 51
  6.3.4 Power Management 52
6.4 **COM Module** 52
  6.4.1 Hardware 52
  6.4.2 Communications Link Reliability 53
  6.4.3 Communications Network and Protocol 53
  6.4.4 Communications Bandwidth and Speed 53
  6.4.5 Application Programming Interface 53
  6.4.6 Power Management 53
6.5 **UI Module** 54
  6.5.1 Platform 54
  6.5.2 Modes of Operation} 54
  6.5.3 Flexibility and Extensibility 55
  6.5.4 Application Programming Interface 55
  6.5.5 Power Management 55
  6.5.6 Design Recommendation 55
6.6 **BSU Module** 55
  6.6.1 Computing Platform 56
  6.6.2 Communications Network 56
6.7 **Additional Specifications** 56
  6.7.1 Operating Environment and Packaging 56
  6.7.2 Software Standardization 56
  6.7.3 Cost 56
6.8 **Summary** 56

7 TracerMap: A GIS FOR THE TRACER SYSTEM 57

7.1 **Overview** 57
7.2 **Basic Operation of TracerMap** 57
  7.2.1 The OpenMap Java Library 57
  7.2.2 TracerMap as an Application and an Applet 58

7.3 Designing a GIS for the Tracer System ... 58
  7.3.1 Data Format ... 59
  7.3.2 Data Exchange ... 60
  7.3.3 The Role of XML-RPC in TracerMap ... 61
7.4 TracerMap Functionality ... 62
  7.4.1 Using ESRI Shape Files ... 63
  7.4.2 Plotting a GPS Trace ... 63
7.5 Operational Enhancements ... 64
7.6 Future Research ... 65

8 SUMMARY AND FUTURE RESEARCH ... 67

REFERENCES ... 68

APPENDIX ... 72

1 OVERVIEW

Despite recent advances in Intelligent Transportation System (ITS) technologies, the potential for improving transportation system performance is curtailed by our limited understanding of the relationships between transportation system performance and characteristics of travel demand. While our models of supply tell us what we do under various traffic scenarios, there is little that we may confidently implement due to a fundamental lack of understanding of individual travel demand and route utilization. Any attempt to implement traffic management strategies requires assuming a demand pattern, one that cannot be subsequently modified based on the implemented supply modification. Even the real-time application of simple Advanced Transportation Management and Information Systems (ATMIS) strategies such as modification of signal timings is severely limited by the inability to generate real-time origin/destination (OD) matrices, let alone demand matrices that reflect modifications directly resulting from the implemented strategies.

The difficulty in obtaining reliable and accurate travel behavior data is one of the primary reasons individual travel behavior and route selection is so poorly understood. Travel behavior studies commonly utilize self-report measures, stated-preference surveys, or laboratory simulation to gather data but rarely involve direct measurement of actual individual behavior. When individuals omit aspects of their behavior in surveys or are limited in their behavioral options during a simulation, significant discrepancies can arise between actual behavior and "measured" behavior. These shortcomings introduce significant doubt about the reliability of data collected for model development. As we move forward with the marriage of transportation and information technology, the need to understand the motivations of the individual traveler will become even stronger. The new traveler information technologies open up a new dimension of uncertainty, raising questions about market penetration and user adoption.

1.1 Project Goals

To begin to address these shortcomings, we have attempted to merge global positioning system (GPS) technology with in-vehicle data collection for transportation research and operations applications. GPS-enhanced vehicle-based data will provide direct, objective data about travel behavior and the state of the transportation system influencing this behavior. The goals of this effort are:

1. Develop a flexible data recording platform supporting a variety of in-vehicle data recording applications.

2. Implement and evaluate core data recording applications including:

    a. GPS tracking of vehicle travel profiles;
    b. real-time transmission of vehicle position and performance data;
    c. links for future interactive in-vehicle driver surveys.

3. Investigate the use of this technology in applications such as:

   a. probe vehicles to complement conventional real-time sensors;
   b. data collection for network links in the California ATMIS Testbed, particularly as part of model calibration/validation studies;
   c. driver route choice behavior;
   d. the study of the generation and scheduling of household trips as a function of anticipated travel time and other route information.

1.2 Project Status

This project was jointly supported by PATH MOU 3006 and PATH Task Order 4120 as well as by the California Advanced ATMIS Testbed associated with the Institute of Transportation Studies (ITS) at the University of California, Irvine. The capabilities of the ATMIS Testbed that contributed to the project included the provision of historical and real-time traffic flow conditions from traffic sensors deployed throughout the Orange County, California freeway and arterial network and laboratory platforms for data collection and analysis applications.

This project was designed as a two phase study. The first phase of the project was concerned with demonstrating the feasibility of the developed GPS technologies for various data collection applications. The second phase extended the basic proof-of-concept field tests with more detailed evaluations of probe vehicle capabilities and route choice behavior. The Phase 1 (MOU 3006) project objectives were:

> Design, develop, and evaluate a portable in-vehicle data collection system and associated support software (together deemed TRACER)
>
> Investigate the use of GPS recording technology as a means of data collection, with instrumented vehicles acting as data collection probes

The Phase 2 (TO 4120) objectives were:

> Apply TRACER to assess travel time estimates using GPS survey data together with real-time loop data
>
> Investigate driver route selection behavior, in response to varying levels of recurrent and non-recurrent congestion
>
> Define applications of the technologies to increase the understanding of trip-making behavior in areas such as (a) the generation of household trips as a function of anticipated travel time and other route information, (b) trip scheduling (especially travel departure times), and trip chaining

This report summarizes both the Phase 1 (MOU 3006) and Phase 2 results.

1.3 Report Organization

This report summarizes resaearch on GPS-based in-vehicle data collection for both traffic operations and travel behavior. Section 2 provides an overview of the research literature associated with GPS applications in transportation. Section 3 provides an overview of the TRACER system.

Sections 4 and 5 document some of the initial field experience using the EDCU and provide a summary of initial tests of system functionality. Evaluation results for probe vehicle and route choice studies are presented.

Section 6 presents the technical design parameters for the extensible data collection unit (EDCU) unit itself. Section 7 provides an overview of the TRACERMap GIS developed to analyze the data from the EDCU. Finally, Section 8 provides a summary and describes some research directions.

2 LITERATURE REVIEW

In-vehicle travel time data collection techniques that have emerged with the development of advanced ITS technologies include cellular phone tracking, automatic vehicle identification (AVI), and automatic vehicle location (AVL). GPS technology, which is among the most recently applied of the AVL technologies, has become a commonly used data collection tool because of its ability to automatically collect the speed and spatial coordinates of vehicles at regular time intervals (e.g. one second).

Applications of GPS for real-time in-vehicle tracking include (1) probe vehicle surveillance, (2) congestion management, (3) historical database generation, (4) fleet management, and (5) travel diary surveys. Also presented are reviews of related research topics including (6) GPS and map matching, (7) user interaction concerns, and (8) the integration of GPS in web-based travel surveys.

2.1 GPS Applications in Probe Vehicle Surveillance

Since the 1920s, transportation professionals have been using the floating car technique to collect travel time information. Traditionally, this technique used a manual method to collect travel time data. This method required the driver to operate the test vehicle while a passenger recorded elapsed time information at predefined checkpoints using a pen and paper, audio tape recorders, or portable computers. Although the manual method is advantageous in that it requires minimal equipment costs and a low skill level, the fact that it is very labor intensive is a significant drawback. Human errors that often result from this labor intensiveness include both recording errors in the field and transcription errors as the data is put into an electronic format (Turner, *et. al.,* 1998).

As portable computers and other electronic technology improved, this manual method was automated through the use of an electronic Distance Measuring Instrument (DMI), which determines speed and distance using pulses from a sensor attached to the test vehicle's transmission (Quiroga and Bullock, 1998a). In comparison to the manual method, the amount and variety of data that can be collected is increased, while the opportunity for error is decreased. More specifically, the advantages of electronic DMIs include: only one staff member (i.e. the driver) is needed per vehicle; data reduction and analysis time are decreased due to the automatic recording of data to a portable computer; and fuel consumption and mobile source emissions can be determined from the acceleration and deceleration data that is calculated (Turner, 1996). Electronic DMIs, however, are not without their limitations. These include: DMI units are not easily shared by vehicles due to the sensor wiring that is required; frequent calibrations and the verification of factors which are unrelated to the unit itself (e.g. tire pressure) are necessary; and the overwhelming size of each data file can lead to potential disk storage space problems (Turner, *et. al.,* 1998; Benz and Ogden, 1996).

Recent advances in GPS technologies have largely overcome the data quality and quantity shortcomings of the manual and DMI methods of collecting travel time data. The use of GPS for ITS probe vehicle applications has a major advantage over other floating car techniques in that GPS can be used in conjunction with geographic information systems (GIS) since GPS provides the automatic geo-coding of speed and positional data (Quiroga and Bullock, 1998b). Although the application of GIS to transportation dates back to the 1960s, the ability of a GIS-GPS interface to display and analyze travel information has grown immensely in recent years as the computing speed and functionality of GIS has increased (Goodchild, 1999; Shadewald, 2000).

Recent traffic engineering studies utilizing a GIS-GPS interface to assist with probe surveillance include those conducted by Quiroga (1997, 1998) and Quiroga and Bullock (1998a). Quiroga has utilized GPS in a variety of traffic engineering studies with a probe vehicle and has addressed link speed estimation as well as interoperability with GIS for managing data generated by the GPS (Quiroga, 1997, 1998). Barth (1995) has used GPS to estimate traffic stream parameters for emissions assessment. Other research efforts that have used GPS probe vehicle systems for the collection of travel time data include: an examination of the use of rural buses as traffic probes (Turnbull, *et. al.,* 1998); an assessment of the feasibility of using freeway service patrol vehicles as probes for measuring level of service (Moore, *et. al.,* 2001); and an analysis of the extent to which probe vehicles equipped with dynamic in-vehicle route guidance systems are useful to drivers (Schofer, *et. al.,* 1996).

Critical issues that must be considered when using probe vehicles relate to the quality and quantity of data collected. Research efforts addressing these issues have examined such topics as the determination of the optimal level of probe vehicle deployment, the effect of sampling bias on the accuracy of probe estimates, and the result of integrating loop detector data with probe vehicle data (Van Aerde, *et. al.*, 2001; Hellinga and Fu, 1999; Sen, *et. al.,* 1997; Choi and Chung, 2001). Additionally, researchers conducting GPS studies with GIS must ensure that a good base vector map with links to a database is obtained (Quiroga and Bullock, 1998b).

2.2 GPS Applications in Congestion Management

The consequences of traffic congestion extend well beyond the personal inconveniences felt by individual travelers. Environmental quality, roadway safety, and community access are among the many quality-of-life concerns that arise in areas plagued with congestion. The 1991 Intermodal Surface Transportation Efficiency Act (ISTEA), which was re-authorized in 1998 as the Transportation Equity Act for the 21st Century (TEA21), mandates that congestion relief be considered in the selection of transportation improvement projects, and that all metropolitan areas with population exceeding 200,000 develop and implement a Congestion Management System (CMS). A CMS is a decision support tool used for selecting alternatives and strategies to better manage and operate transportation facilities in order to relieve traffic congestion and enhance mobility. As a result of the complex data management that is required for the efficient and effective assessment of any environmental, financial, social, and economic impacts of project implementation, many cities are using a GIS-GPS approach for the collection of congestion data. GIS is used to aid in the storage, retrieval and visualization of the traffic corridor and congestion information that is collected via GPS (Sutton, *et. al.,* 1994; Bullock, *et. al.,* 1996).

Analyzed GPS travel time data can be used for congestion management in order to detect incidents and to determine the location and extent of bottlenecks and queues (Bullock, *et. al.,* 1996; Jiang and Li, 2001; Gallagher, 1996). This information can in turn be used to estimate the effect of traffic congestion on vehicle emissions, wasted fuel, idling time, and traveler delay. Recent studies using GPS travel time data for congestion management have been conducted at the Remote Sensing and Image Processing Laboratory at Louisiana State University (Bullock, *et. al.,* 1996) and by the Boston Metropolitan Planning Organization (Gallagher, 1996).

2.3 GPS Applications in Historical Database Generation

Database generation is another application of GPS technology for travel time data collection. A major goal of San Antonio's recent Metropolitan Model Deployment Initiative, for example, was to develop an extensive, real-time travel speed and roadway condition database. The Texas Transportation Institute (TTI) used the following four primary sources of data in order to develop this historical database: inductance loop and acoustic detectors, traffic simulation models, AVI tags, and GPS technology. Together these sources provided data on current link travel speeds and time, as well as on historical travel behavior. The GPS component of this data collection effort involved test runs along portions of the roadway network that were not covered by AVI and other Advanced Transportation Management Systems (ATMS) technology. For each segment, TTI conducted a minimum of three test runs with GPS equipment (per 15-minute time interval) during peak conditions and under recurrent conditions (Carter, *et. al.,* 2000).

2.4 GPS Applications in Fleet Management

GPS technology can also be used to translate fleet location data into real-time reporting. When combined with technologies such as, wireless communications, GIS, and the Internet, GPS technology helps provide a cost-effective tracking and services solution for fleets of all sizes. Commercial fleets, for example, use GPS to improve efficiency and increase profits by providing real-time vehicle location and status reports, navigation assistance, driver speed and heading information, and route history collection (WebTech Wireless, 2001). When combined with on-vehicle sensors, GPS can report when and where a commercial truck trailer is connected or disconnected from its tractor. This data helps trucking firms assess delivery performance or determine if a trailer has been stolen (@Track Communications, 2001). Emergency service fleets, such as police cars, fire trucks, and ambulances, use GPS to aid in the dispatching and route guidance of the nearest vehicle to an incident (MORPC, 1999). This helps ensure that citizens receive a high level of service no matter where they live or what types of emergencies they may face. Additionally, transit fleets use GPS technology to improve scheduling, enhance dispatching capabilities, and provide real-time information to passengers (Turnbull, *et. al.,* 1998; Higgins, *et. al.,* 1995).

2.5 GPS Applications in Travel Surveys

The application of global positioning systems (GPS) to the study of individual travel behavior is a relatively recent phenomenon. The modeling and analysis of travel behavior has been based largely on records of reported trips and the socio-economic characteristics of the trip-makers, information which is typically collected via travel diary surveys, as well as transportation network and regional land use characteristics. Although early travel surveys were conducted using paper-and-pencil interview (PAPI) techniques, computer-assisted telephone interview (CATI) methods became much more common in the 1980s and 1990s. In recent years, the travel diary survey collection process has become more automated as computer-assisted-self-interview (CASI) methods, in which respondents input their travel information directly into a computer, have become widely used. Most recently, GPS technology has been used to supplement or replace other forms of travel diary survey collection. When compared to other methods of travel behavior data collection, GPS is advantageous in that respondent burden is reduced, data quality is improved, additional information can be collected, survey periods can be extended to multi-week or multi-year periods, and data analysis can be performed with greater ease (Wolf, *et. al.,* 2001; Draijer, *et. al.,* 2000; Murakami and Wagner, 1999).

Many of the GPS travel diary studies conducted in recently have either included the use of electronic travel diaries or have involved completely passive vehicle-based GPS devices. The first GPS-enhanced travel diary experiment in the U.S., conducted by the Lexington Metropolitan Planning Organization in 1996, falls into the first of these two general categories. This successful "proof-of-concept" study, which involved a sample of 100 households in Lexington, Kentucky, identified two distinct benefits to a survey method that includes GPS technology: first, route choice data becomes available and, second, data can be organized by highway functional class usage and travel speed (Battelle, 1997). Subsequent analyses of this study indicate that the use of automated diaries with GPS data can also provide a detailed analysis of trip chaining behavior (Yalamanchili, *et. al.,* 1999). Another recent study that has investigated the use of automated diaries with GPS was a 1999 study conducted in the Netherlands (Draijer, *et. al.,* 2000). Passive in-vehicle GPS studies include Austin's 1998 household travel survey, Quebec City's vehicle-based GPS tracking system project, and Georgia Tech's data collection effort in Atlanta (Casas and Arce, 1999; Doherty, *et. al.,* 2001; Wolf, *et. al.,* 1999).

A research effort that aimed to examine the feasibility of completely replacing the paper or electronic diary with a GPS data stream was conducted at Georgia Tech. This proof-of-concept study, which involved a sample of thirty participants, demonstrated the capability to identify trip ends within a GPS data stream, to carry out land use and address "look-ups" with a GIS, and to assign trip purposes from the derived land use and address information. When the derived travel data was compared with travel data recorded by the respondents on paper diaries, the derived data was found to match or exceed the reporting quality of the respondents (Wolf, *et. al.,* 2001).

2.6 GPS Applications and Map Matching

As an alternative to using point-to-point distance estimates, (Roden, 1999) demonstrated a map matching algorithm that could match second-by-second GPS points to a road network derived from Tiger files. The algorithm consisted of three main steps: a smoothing procedure that first removed anomalies in the raw GPS data, a procedure to identify "corners" taken en-route, and the actual matching of the points to the network and identification of links that make up the traced route. A sample of 300 trips over 58 days from one sample vehicle was used as a basis to demonstrate how a variety of travel distances and times measures could be derived with this approach.

Similar experimentation in Quebec, using differentially-corrected GPS sample data has shown that even the simplest of GIS-based algorithms that match GPS points to the nearest link using a "buffering" technique could lead to over 90 percent accuracy without further processing (Doherty, 1999). An algorithm was also developed to identify trip-ends, or "activity" nodes, based on the clustering of GPS points in time and space. All stops of length greater than a given threshold were easily detected by the algorithm. However, it was recognized that short stops, such as when dropping off passengers, would be more difficult to detect automatically, and would likely require some form of user input.

2.7 GPS Applications and User Interaction

Current applications can be categorized as using "hands-off", passive devices or using devices that require some level of user interaction. A passive vehicle-based GPS recording device has been developed at the Texas Transportation Institute and applied to 200 households who also completed a standard paper-and-pencil activity diary (Pearson, 1999). Preliminary analysis suggests that the GPS data captures more trips than the diary, especially short trips, although further analysis is needed.

The Lexington study (Murakami, et al. 1999) was one of the first to demonstrate how passively collected GPS data could be combined with hand-held computers to interactively obtain supplemental information trip purposes. Respondents were instructed to turn the unit on within their vehicle at the start of a trip, then specify the trip purposes of each person in the vehicle via prompts on the hand-held computer. These trip purpose, trip start and end times, and origin/destination addresses were then tagged to the resulting GPS point data recorded at 3-second intervals. In this way, the device was able to replicate the same type of information collected via a traditional trip diary. Subsequent analysis of 100 households focused on the improvements in the accuracy of recorded trip start and end time, travel distance calculated on a point-to-point basis. This work also highlighted many problems associated with such units, including loss of position fix and user-induced errors of omission.

A comprehensive GPS-based survey project is the work of Wolf et al. (1999) at the Georgia Institute of Technology. A multi-instrument approach was developed that builds upon and extends past approaches. Three separate surveys were planned, including:

1. a person-based hand-held electronic travel diary with a GPS receiver;
2. a vehicle-based electronic diary with GPS and an on-board engine monitoring system;
3. a passive GPS receiver and data logger to capture vehicle trips only.

The goal of these surveys is to replicate traditional activity-based diaries (albeit, more accurately), with the addition highly detailed route and link information, vehicle and engine operating conditions affecting emissions, and freeway traffic conditions derived from real-time advanced traffic monitoring system linked to the GPS data. This data is intended to support the development enhanced travel demand models capable of addressing emerging land use, travel behavior, and air quality issues.

These studies have demonstrated that GPS technology has the potential to provide a passive trace of vehicle movements over long periods of time, but can also be combined with more interactive techniques to provide a more comprehensive picture of travel behavior. Post-processing of the data can also be used to identify the links associated with a given route, and the start and end times and locations of activities. These studies have also highlighted several technical issues that require further attention:

1. keeping the equipment package as small and durable as possible;
2. maintaining a consistent power supply;
3. handling large amounts of incoming data;
4. coping with GPS signal outages; and
5. resolving the accuracy of GPS data via differential correction.

These considerations were all incorporated into the design constraints of the data collection unit described in this report.

The issue of how much user interaction is needed to conduct such surveys is also an important future concern. Indeed, user intervention is required to obtain certain attributes and may be needed in the future to correct for missing signal outages and even manually specify certain trip segments (e.g. short, drop-off passenger trips). The level of respondent burden associated with this interaction is a key concern, especially if efforts continue to focus on real-time hand-held data entry devices. However, different types of user intervention are possible, which have the potential to open up significantly new opportunities to describing and explaining observed travel patterns.

For those studies above that involved user intervention, the primary goals was to obtain supplemental information that when taken together with the GPS data, would serve to replicate and enhance existing travel diary techniques. At a minimum, this required the collection of passenger and trip purposes information. Also desirable would be multipurpose stop information.

The approach adopted thus far focuses on the use of hand-held computer devices through which supplemental information is recorded en-route. The main advantage of this approach is that the user-supplied information can be directly matched in real time with the GPS data when it is fresh in memory. However, such an approach is limited by the display capabilities of these devices, is prone to errors of omission, and is limited in the amount of detail that can be obtained before the device becomes prohibitively inconvenient for the user.

An alternative approach described by Doherty (1999) would involve "home-based" intervention with the user after-the-fact on a desktop computer. Such an approach would require that the GPS data be pre-processed and displayed back to the user in some format, so that it may be updates with supplemental information. The GPS data would essentially serve as a "memory jogger". Such an approach would greatly extend the display and data entry possibilities, and allow the user to enter information at a more convenient time of their choice. The challenges would be to develop a means for transferring and pre- processing the GPS data prior to display back to the user for intervention.

Transferring the GPS data to a central server before displaying at the home base would allow the reduction of GPS points to a series of routes (links) and trip ends (activity nodes) on a network. This information could then be displayed sequentially, spatially or temporally back to the user for further update and/or interactive experimentation. A sequential approach would be the simplest way to cycle through a series of trip ends in order to update them with additional information. Multi-purpose information could also be queried at this time. However, querying for missing trip segments may be difficult, and perhaps more easily overlooked when entering data in sequence. A spatial interface would involve the use of a GIS to display the routes and trip ends overlaid on a road network (plus any land-use, landmark, etc. layers that can be accommodated), followed by subsequent prompts for information at each node displayed on screen. In this way, missing nodes, multipurpose nodes, and even missing trip segments could be added. The spatial interface would also open up many new opportunities for interactive queries and experiments concerning route choice decision processes. The main challenge would be to overcome potential difficulties in a user's ability to understand and/or navigate through such a display. A temporal interface would involve displaying trips and trips ends as a series of activity nodes and trips along a time dimension. This activity-based approach may be more intuitive than a spatial display, mimicking how activities appear in a typical day-planner. Such an approach would allow exploration of various aspects of how these activities were decided upon over time (this is explored in more depth in the following section). A spatial display may still be needed if investigation of route choice is desired.

All three of these approaches are worthy of further exploration as alternative means to actively update the largely passive data collection opportunities provided via GPS. However, if the goal is to move beyond observed patterns towards an understanding of the underlying reasons and decision processes that led the observed patterns, then a home-based approach offers significantly new opportunities.

2.8 GPS Applications Integrated with Web-based Travel Surveys

In recent years there has been a growing interest in experimenting with new approaches for household activity/travel surveys. These experiments can be divided into two general approaches. The first involves the application of new technologies, such as GPS and handheld computers, to obtain high resolution personal travel data (Battelle, 1997; Guensler and Wolf, 1998). Such data has promise to advance existing travel models and may even assist in a paradigm shift for travel forecasting. These data, however, are outcomes of the decision process, often termed activity scheduling, that determines when, where, with whom, and for how long to engage in various activities. Axhausen and Gärling (1992) stressed the importance of this process by arguing that it is at the core of travel behavior changes. Effects of transportation policies such as tolling, congestion pricing, and travel demand management measures depend on how people would adjust their daily activity and travel pattern to changes to their everyday lives. They also argued that the process is "largely unknown" and new methods should be developed to conduct in-depth study of the process. The second group of data collection experiments concerns the development of innovative approaches focusing on the process of activity scheduling.

The REACT! software is based on the Computerized Household Activity Scheduling Elicitor (CHASE) program developed by Doherty and Miller (2000). CHASE was unique in that it collected data on the household activity scheduling process for a week long period. Members of sample households would run the program daily to record activities from their initial plans to final actions. CHASE was tested in several experimental surveys and proved to be an efficient data collection tool for study of household activity scheduling. Several critical areas for improvement were identified prior to the development of REACT! (Lee and McNally, 2001), a program aimed at extending the scope of CHASE. In addition to hardware and software enhancement, significant advancement were made in terms of tracing decisions involved in the scheduling process. REACT!, and a modified version of that software, are being used in the current project.

3 THE TRACER SYSTEM

The in-vehicle data collection system, deemed TRACER, has been developed for a variety of applications. It combines a GPS with an automated data collection system and a wireless modem which allows data to be accessed (and the vehicle to be tracked) any time the vehicle is under power. A brief review of the design of this system in this section is followed by basic system evaluation in Section 4 and a comprehensive presentation of system design in Section 6.

3.1 Application Requirements

While the potential uses of GPS-enhanced vehicle-based data collection vary significantly, a set of generic applications can be defined that span functional requirements for various uses. Four data collection applications are envisioned:

1. *Basic*: basic multi-day survey of vehicle trajectories;
2. *Probe*: obtaining real-time traffic stream conditions via probe vehicle,
3. *Enhanced*: enhanced multi-day trajectory survey with behavioral logging;
4. *Route Guidance*: routing behavior under real-time route guidance

The most critical for initial application were the first three applications (the fourth, while not scheduled for immediate application nevertheless played a significant role in defining system functionality). Basic system design parameters for each of these data collection applications are summarized below.

3.1.1 Basic Survey

In the basic survey application, the data collection system records a vehicle's trajectory for later analysis. A prototype application compares basic survey data with traffic flow data collected from loop sensors on local freeways. Parameters for this application included:

1. The system must collect sufficient data to enable subsequent matching of vehicle trajectories to a transportation network. It was assumed that such matching would require positional fixes taken in as frequently as 1-second intervals at differential GPS accuracy while the vehicle is in motion
2. Drivers will average a maximum of 4-hours per day over a 7-day period that requires storage of positional fixes for as many as 28-hours total.
3. The survey participant will have minimal access to the data collection device during the 7 day data collection period, and the device must be able to operate without maintenance for 7 days. The ideal system could be stowed in the trunk of a vehicle and retrieved after the data collection period.

Post-analysis of collected data will vary with different research projects. This required that data be accessible in a flexible, non-proprietary format that could be analyzed on a variety of computing platforms. For instance, work during phase 1 of this project compared basic survey data with traffic flow data collected from other sensors in the California ATMIS Testbed (e.g., loop detectors). Analysis of collected data will involve both commercial and in-house software running on platforms such as Solaris-based Sun workstations, Linux-based PCs, and Windows/NT-based PCs.

3.1.2 Probe Vehicle Study

In probe vehicle study applications, the data collection unit collects data from a vehicle traveling in the traffic stream. The data is logged by the on-board unit and transmitted in near real-time to a base station over a wireless communications network. Parameters for this application include:

1. The studies will take place in urban and suburban settings.
2. Data collected for logging and remote transmission include the position and speed of the vehicle, point-to-point travel times, and acceleration data for pollution studies (the collection of acceleration data would require advanced technologies such as micro-gyroscopes not incorporated into the system).
3. Collection of some types of data may require transmission of data as often as once per second, even when the vehicle is stopped; however, as in the basic survey application, stopped data need not be logged.

Post-analysis of collected data will vary with different research projects. This requires data be accessible in a flexible, non-proprietary format that can be analyzed on a variety of computing platforms.

3.1.3 Enhanced Survey

The enhanced survey application extends the basic survey to include additional logs of respondent activity during the data collection process. For instance, log data might be the results of questions asked of drivers about the purpose, characteristics, expectations, etc. of a trip before or after it is made. A prototype solution extends the data collection device designed for the basic survey by connecting a user interface for gathering the additional information. Parameters for this application include:

1. The data collection device is isolated from the survey participants as with the basic survey except for the user interface for survey data collection.
2. The user interface must be sufficiently compact so as not to interfere with the driver and normal use of the vehicle.
3. It is likely that some survey respondents will not have extensive experience using computing devices, thus, the input device for survey data collection must be implemented on a platform that allows user-friendly interfaces.

The capabilities necessary for conducting the enhanced survey were not required in the final design of the units, but that design must allow for and facilitate the later addition of these capabilities.

3.1.4 Route Guidance Study

In the route guidance study drivers are provided with descriptive information about network conditions and/or prescriptive guidance about the best routes to take via real-time wireless communication with the vehicle. In addition to the capabilities given for the probe vehicle application, this application requires:

1. Two-way communication between a base station and the vehicle (for the transmission of descriptive and/or prescriptive guidance from the base station to the vehicle and positional fixes of the vehicle back to the base station).
2. A user interface, such as in the enhanced survey, that is flexible enough to display route guidance information to the driver, such as current position, desired destination, and suggested routing instructions.

The route guidance study application will ultimately require substantial software in the TMC to drive the route guidance application. In the current design, the system provides support for remote wireless communication and a software API for two-way data transmittal.

3.2 TRACER System Specifications

The TRACER system incorporates a prototype extensible data collection unit (EDCU) meeting the application requirements above with a suite of base station processing software (for real-time and post-processing). These self-contained units are based on a power-efficient x686-class, 133MHz microprocessor running a Linux-based embedded operating system. The unit uses flash-RAM as its primary storage, and controls both a GPS receiver and a cellular packet data (CDPD) modem. The operating system runs programs to control the various EDCU applications outlined.

Figures 1 and 2 show the prototype unit in both outside and inside views; figures 3, 4, and 5 show the final EDCU unit design (the diskette is included for scale). Figures 3 and 4 shows the final exterior design (with a slightly larger packaging), while Figure 5 shows the inside of the unit with input/output ports, flash ram, and the CDPD modem. A third generation unit in design places an 802.11b card parallel to the CDPD card enabling local wireless communications with a laptop or PDA for system configuration or user interface.

Under normal operation, the units are designed to operate by tapping a vehicle's power supply via an auxiliary power port (e.g. cigarette lighter). With the addition of a battery pack, they can also be isolated from the vehicle's power for short duration applications. The units are enclosed in a weather resistant case that is suitable for the relatively harsh environment found in an automobile.

Figure 1. Prototype EDCU in R100 Case (exterior view)

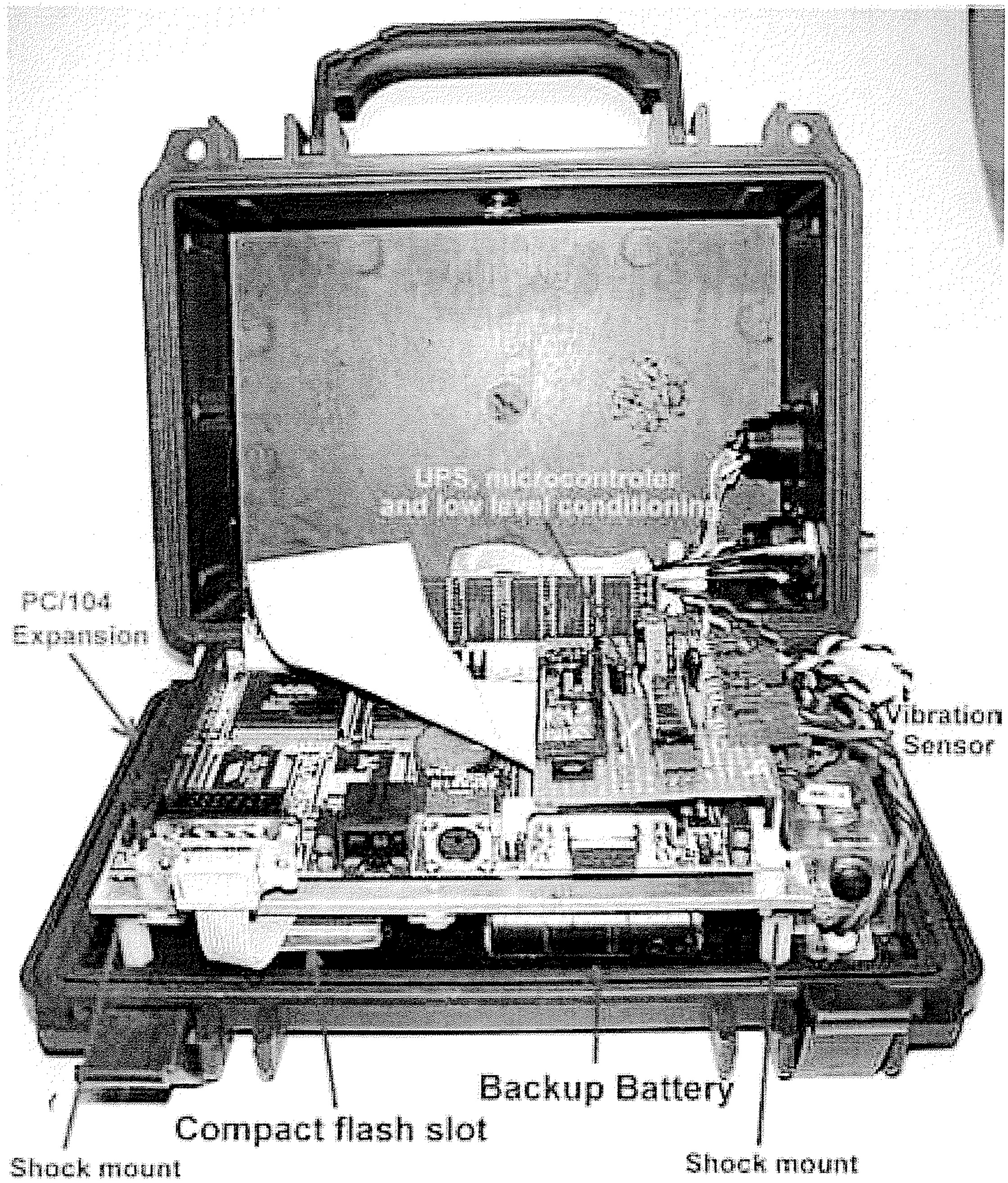

Figure 2. Prototype EDCU in R100 Case (interior view)

Figure 3. Final EDCU Design in R120 Case (exterior view)

Figure 4. Final EDCU Design in R120 Case (exterior side view)

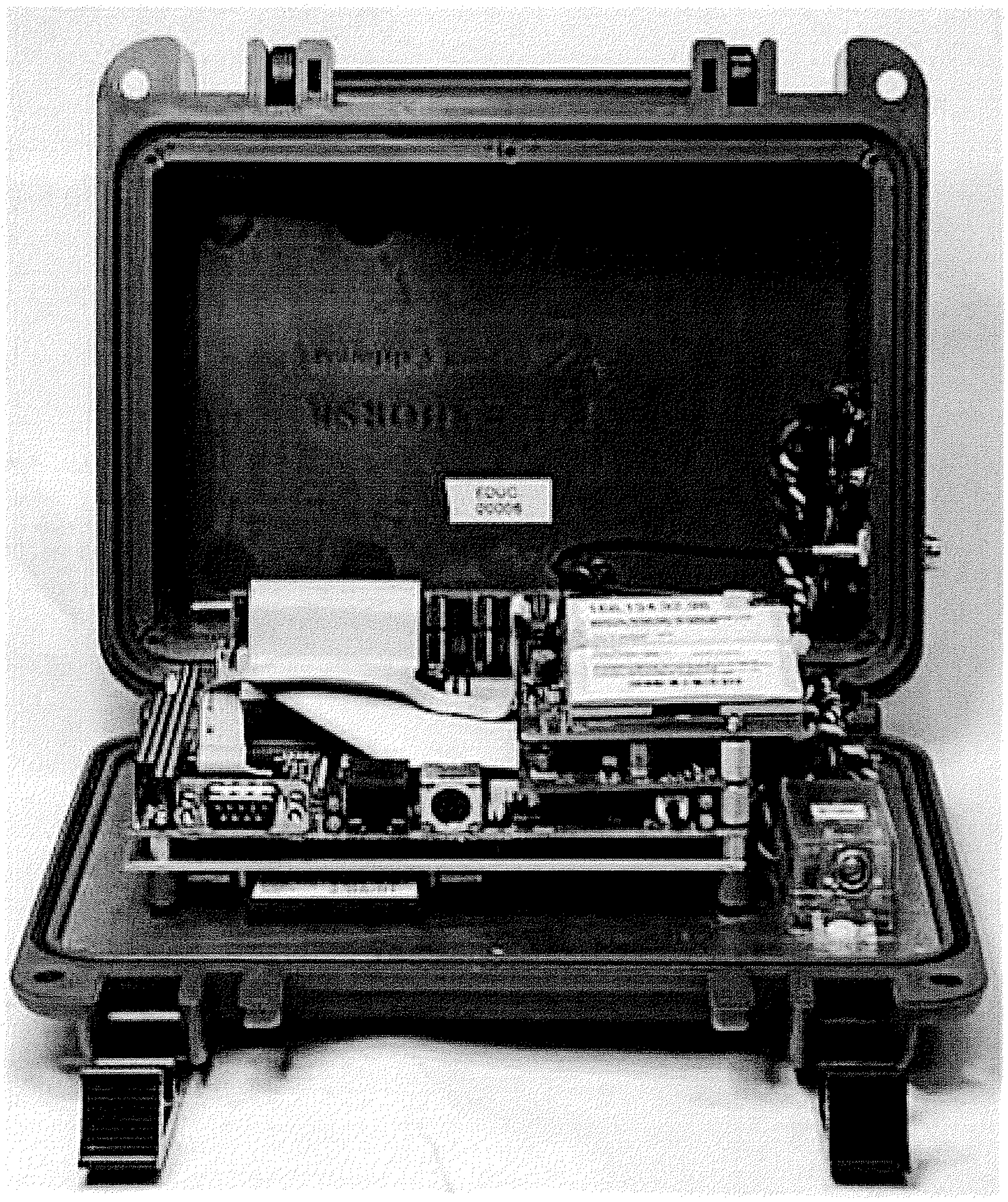

Figure 5. Final EDCU Design in R120 Case (interior view)

Because the equipment was intended to fill multiple data collection roles with significantly different requirements, versatility was a primary design goal. The data collection system therefore employed a modular design to ensure such flexibility. The device consists of the following modules:

1. a data collection unit controller (DUC),
2. a global positioning system module (GPS),
3. a on-board logging module (LOG),
4. a communications module (COMM),
5. a user interface module (UI), and
6. a base station unit (BSU).

The prototype units contain the DUC, GPS, COMM, and LOG modules. The DUC and the LOG modules form the heart of the extensible data collection unit (EDCU), with the DUC responsible for the operation of the data collection unit and the LOG responsible for saving the collected data. The DUC is implemented as a computer on a board, with a 16 MB flash RAM module to use as system memory. The flash RAM has been partitioned such that there is adequate space for data logging. To the software applications, the LOG unit exists as a distinct mount point in the Linux file system. Saving data to the LOG is as simple as writing to a file.

The COMM unit is handled as a communications device implementing a TCP/IP stack -- in other words, it is not much different at the applications level from a modem connected to the Internet. Sending data to a base station unit (BSU) is accomplished by writing data to a data socket connected to the BSU. Receiving incoming instructions from the BSU is accomplished by listening to an open socket. In practice, accessing data from the unit is accomplished through the usual programs such as FTP, telnet, or through a web browser (or by reading the flash RAM directly). Since the Linux operating system was selected along with standard CDPD modems, the BSU module effectively becomes any computer connected to the Internet. In practice, the BSU role is filled by a dedicated server to ensure the integrity and security of the collected data.

3.3 TRACER Support Software

The GPS-enabled travel survey data must be integrated with geographic information systems (GIS) for real-time and post-processing. The power of knowing where a vehicle is during a survey period is increased when one can merge that information quickly with a GIS program. BBN Technologies' OpenMap (http://openmap.bbn.com) GIS is written in Java and was ideally suited to be Internet-based. The software can probe a EDCU over the Internet and posts that information on a map. An EDCU in a survey vehicle can send tracking data to the driver's personal computer for subsequent consideration in REACT!. Initial work on OpenMap software development is documented in Section 7.

4 EVALUATING EDCU FUNCTIONALITY

The key components of the TRACER system, the Extensible Data Collection Units (EDCU), were manufactured by a sub-contractor (AeroData) in July 2000. Understanding the capabilities and limitations of the prototype EDCU is crucial to determining its range of application to ongoing research, and how future versions of the EDCU can be improved. A battery of functional tests was planned for the EDCU; preliminary experience with the EDCU, and insights on its extensible nature, are presented here in.

4.1 Preliminary Map Matching of Vehicle Tracings

Figure 6 depicts the tracing of a test drive in the Irvine area, layered (with no map-matching algorithms) over a map generated from TIGER-Line files. Separate traverses were made by the test vehicle in each lane if the I-5 freeway; Figure 7 zooms on a section of this traverse and depicts individual lane tracings. A third layer is a mapping of Caltrans District 12 freeway loop detector locations. The loop detectors provide vehicle counts and occupancies; from these occupancies and assumptions on the mix of vehicle lengths, speed estimates may be derived (see next section). The uncorrected map layer does not coincide well with the GPS vehicle tracing or loop detector location layers, illustrating the necessity of properly syncing map datums. Digital maps can be corrected or commercially-available maps can be acquired. The OpenMap software being developed in association with this research effort reflects common datum for various map layers.

4.2 Utilization with a Probe Vehicle

Using the same data points presented in Figure 6, a comparison was made between EDCU-recorded GPS travel speeds and speed estimates generated by the freeway loop detectors. The travel occurred the evening peak on a weekday when flow conditions exhibited moderate levels of service. The UCI Testbed provided 30-second loop detector data on the freeways in question. The position information was then compared to the loop detectors in front of and behind the recorded positions, both in time and space. The results for the I-5 Freeway are plotted in Figure 8 with selected points for other freeways summarized in Table 1. The plot provides a iso-line which depict congruity of speed estimates. While a general level of consistency was observed, the EDCU units provided evidence that the variability of loop detector speed estimates over short time intervals are significant, thus, the use of these speeds in traveler information systems may be unreliable over such time intervals. Note that the speed variation was greater in the southbound peak direction. A summary of speed comparisons between the TRACER-equipped probe vehicles and freeway loop detectors is provided in Table 2.

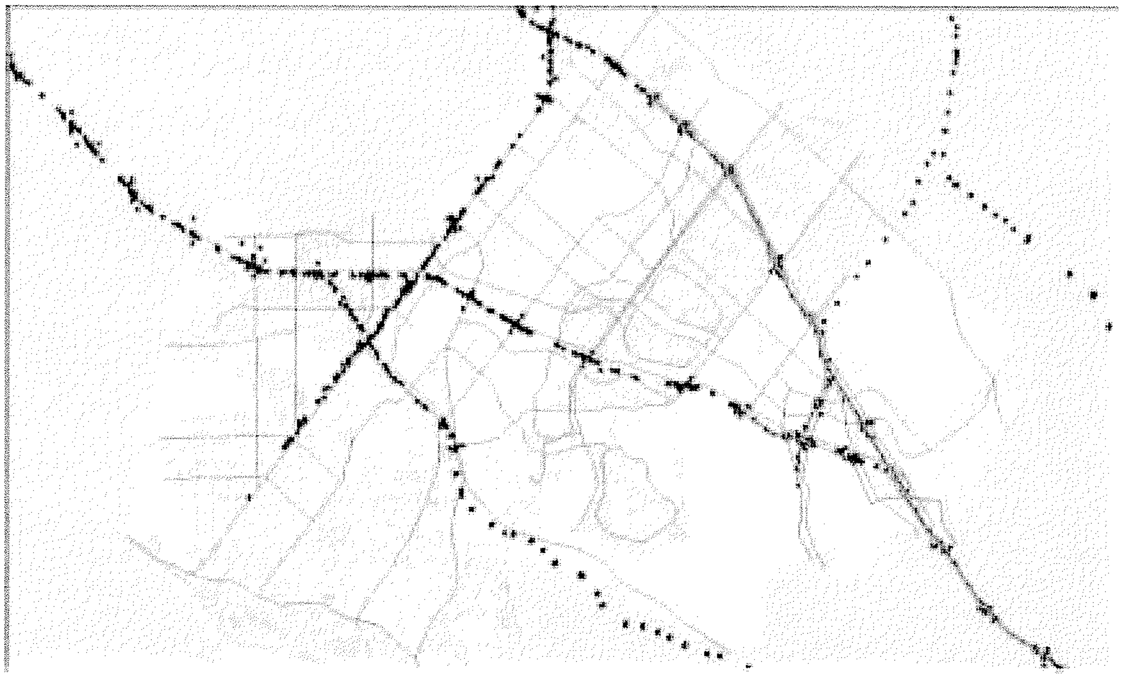

Figure 6. Sample Tracings (red dots are Caltrans loop detector stations)

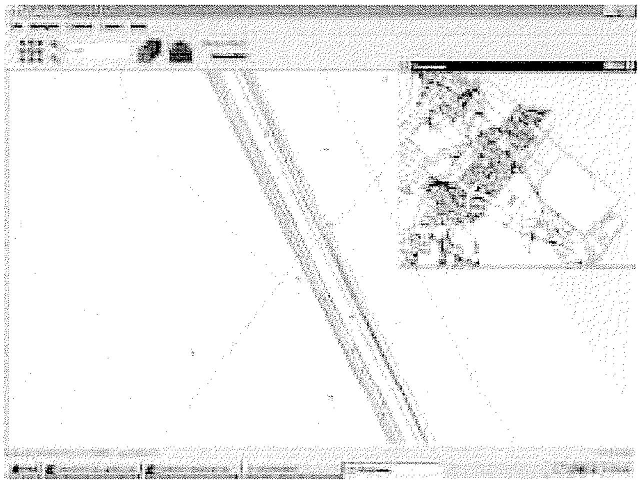

Figure 7. Sample Tracings Zoomed to I-5 freeway Section

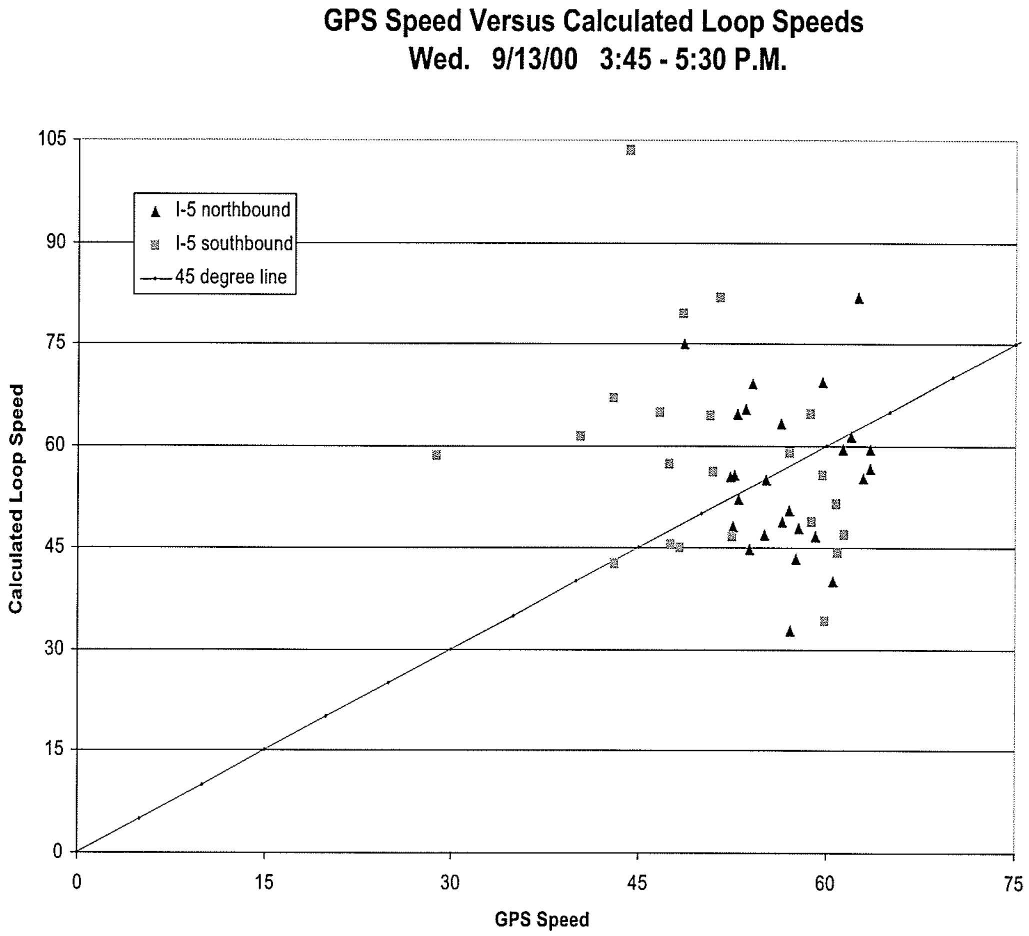

Figure 8. Comparison of GPS and Loop Speed Estimates

Table 1. Comparison of loop estimated speeds versus GPS-recorded speeds

Freeway	Cross Facility	Detector Station	Time (pm)	Speed (mph)	
				Loop	GPS *
I-5 NB	SR-133	1204808	8:58:29	52.1	57.6
	Jeffrey Rd	1204924	8:59:06	52.06	58.9
	Culver Dr	1204982	9:00:36	61.4	57.8
	Culver Dr	1205012	9:01:21	59.2	60.3
SR-55 SB	McFadden	1203239	9:04:20	56.0	54.4
	Dyer Rd	1203161	9:07:30	59.7	53.4
I-405 SB	Bristol Str	1201453	9:09:20	48.0	50.1

4.3 Internet Access

An important design element for the EDCU is the ability to communicate with the unit over the Internet via its wireless CDPD modem. Initial operational tests with the modem showed that the EDCU operated flawlessly when interfaced with the Internet via its ethernet port. The decision to use a version of Linux designed for embedded systems was instrumental in facilitating the Internet connection. The EDCU is in effect a miniature web-server, providing a simple user interface over hypertext transfer protocol (HTTP) links, access to its flash ram files over file transfer protocol (FTP) links, and access to the raw NMEA output from the GPS unit via Telnet. At any time an EDCU is powered, remote access to the EDCU is available to download data or upload data or instructions.

Preliminary questions regarding EDCU logging and memory were addressed by placing the EDCU prototype on the Internet, allowing access to the unit's data. The initial logging script, written in Perl, polls the GPS unit every 10 seconds, and stores the date, time, latitude, longitude, and speed. That daily file, uncompressed, takes up approximately 440KB. At the end of the day, the files are compressed using gzip, resulting in a file size of about 55KB. If an EDCU was to be used in an extended travel survey function, a day's data could be zipped and uploaded in a very short amount of time, even if the connection with the cellular data modem was poor. Further, a two-week survey would only require about a megabyte of memory, if 13 days are compressed and the final day is left in its original, uncompressed state. If the EDCU were to be placed in a traffic probe vehicle uploading data in a near continuous fashion, the data stream of 1 second measurements would only require a total of 4400KB in 24 hours, which can easily be handled by even the spottiest modem connection. Under actual operational conditions, additional information than time, position, and speed could be transmitted. For example, information about the satellites is important if one wishes to apply differential correction to the signals received. Also, the small footprint of the positional data means that there will be some space on the CDPD connection to transmit other information over the link, such as new driving instructions. Figures 9 and 10 depict some of the EDCU internet access screens for setting parameters and displaying data.

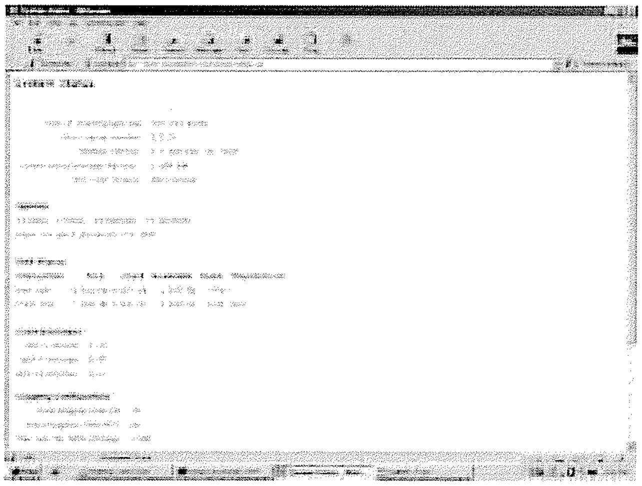

Figure 9. Web EDCU Interface: Parameter Page

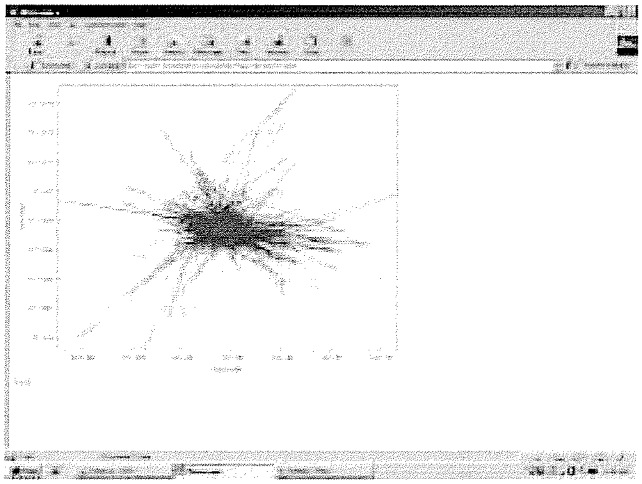

Figure 10. Web EDCU Interface: Trace Plotting

4.4 Integration with GIS Programs

Another area of interest for GPS-enabled travel survey data is the integration of that data with existing Geographic Information Systems (GIS). The power of knowing where a vehicle is during a survey period is increased when one can merge that information quickly with a GIS program. To test the functionality of the EDCU in this regard, the available offerings of Internet-enabled mapping software were reviewed. Most existing programs carry substantial licensing fees, which is undesirable in an academic research setting. The exception to this is BBN Technologies' OpenMap (http://openmap.bbn.com). The license is an open source variant that provides full access to the source code, and free use for non-commercial purposes. The OpenMap GIS is written in Java, and therefore is ideally suited to Internet-based mapping problems.

4.5 Software Platform Extensibility

The choice of an embedded system running the Linux operating system, while important for enabling easy Internet access, is perhaps more important due to the ability to change the EDCU's programmed behavior. While the operating environment is fairly spartan, with initially 16MB and eventually 32MB of total system operating and storage space, the programs that control the behavior are small themselves.

To illustrate this point, we decided to explore generating XML documents containing the output of the GPS unit's NMEA sentences. Extensible Markup Language (XML) has the ability to transform itself into different kinds of web pages, tailored for example to computers or cellular telephones. For the transportation engineering community, XML is likely to be much more important for its ability to mark up data of any kind and make it highly portable. The XML alphabet soup is even thicker than that of the Intelligent Transportation Systems (ITS) world, and so a full description of our work would take up too much space. In sum, we developed a description of how to mark up GPS NMEA sentence data, and then added a small routine to Linux's GPS controller daemon to output using this format when requested to do so. The XML markup codes embed in the data words that define each data element. For example, the latitude might be represented by <LAT>37.106712</LAT>. Readers familiar with other markup languages, such as HTML, might recognize the format, but not the tag. That is because we define the <lat> tag in our definition document. The extra characters required by the XML markup increase tremendously the amount of space required to store the GPS position data. For example, a 24 hour period of 10 second observations generates 440KB of data. Adding the extra characters required to include the XML tags inflates the file size up to 1,400KB! However, XML is important for communication, not for storage. It is simple enough to store the data without the tags, and then to add them on when sending the information to a requesting client.

To illustrate this concept, a Java program was developed to load the raw EDCU log files, process them, and then generate XML output. Similar functionality was programmed in C into the actual GPS Linux driver. The processing step, in addition to adding the XML tags, also checked whether the GPS unit was stationary (according to the GPS speed). This information can be passed on to any other program that understands GPS Trace XML documents, such as a mapping program or a Web-based survey tool like our own REACT! program, making collected travel data transparent to other developers.

4.6 **Two Representative Applications**

We presently require GPS-enhanced vehicle-based data for two research efforts: the REACT! travel behavior survey project (Lee et al., 2000) and the Testbed Real-time Integrated Control and Evaluation Prototype System (TRICEPS) traffic management project in the California ATMS Testbed in Orange County, CA (McNally, 2000). The data requirements for these two projects are behavioral on one hand (REACT!), and operational on the other (TRICEPS). Together they span a broad range of operating requirements for in-vehicle data collection in transportation research. Our goal is the development of a data collection tool capable of supporting both research efforts. Such flexibility will not only reduce the development effort for the individual projects, but will ease the integration of results between the projects which could lead to better understanding of the relationship between travel demand and performance.

4.6.1 REACT!

The REACT! survey program provides an efficient means to collect detailed information on observed activity-travel patterns within a household over a week long period. It does so by providing a graphical interface depicting a person's schedule on a computer (similar to a typical day-timer) on which a person may interactively add, modify, and delete activities as they get planned and subsequently executed. A REACT! survey participant logs in at least once a day to keep their schedule up to date over a week long period. REACT! is an Internet-based survey tool, incorporating features such as nightly uploads of survey data, to complete operation of the survey over the Internet when the connection bandwidth is sufficient. REACT! incorporates in its design a spatial GIS interface for interactive location choice (Lee, 2000).

The next step for REACT! is to combine its operation with vehicle-based travel data. The GPS data collection device will provide accurate traces of routes and travel choices of survey participants. Since the REACT! program is built around Internet technologies, it should be possible to transmit this travel information from the GPS data collection device over a wireless modem to a base station computer, process that data, and then send it back over the Internet to the REACT! respondent's computer. Even without a wireless data connection, in-vehicle travel data could be used to either initialize REACT! and its questions for the respondent, or else to design follow up questions or validate REACT! responses.

4.6.2 TRICEPS

TRICEPS is a development platform that facilitates the implementation and evaluation of a wide range of algorithms for traffic control and Advanced Transportation Management Systems (ATMS) using both simulated and real world data (McNally *et al.*, 2001). It is a component of the California ATMS Testbed, a multi-agency transportation operations environment covering two contiguous sub-areas in Orange County, California that include major decision points for freeway travelers in the region as well as a significant portion of the surface street network. TRICEPS supports research activities in the Testbed by providing:

1. consistent interfaces for transportation management hardware and software components, both simulated and real-world implementations of these components, and

2. a set of core transportation management applications which include automatic incident detection (AID) algorithms based on artificial neural network technology and algorithms for integrated traffic control.

TRICEPS provides a virtual interface to real-world transportation system data collection components. This interface is implemented using the Common Object Request Broker Architecture (CORBA) to manage the communication link with the California Department of Transportation (Caltrans) District 12 (Orange County) data server. This link provides access to data from all data collection and control hardware in the system including loop detectors, ramp meters, Changeable Message Signs (CMS), and video camera data.

A GPS-enhanced vehicle-based data collection device will enable the creation of probe vehicles. A probe vehicle travels in the traffic stream with other vehicles, and transmits highly accurate data about the conditions faced by moving vehicles. The addition of probe vehicle data to the TRICEPS platform will provide a unique source of data to transportation management applications operating within TRICEPS. Furthermore, TRICEPS provides real-time loop data from the Orange County freeway system and from arterials in selected cities in the Testbed. This data can be processed along with the GPS-based trajectories to evaluate travel times and congestion levels on the primary traffic routes as well as on alternate routes.

Table 2. Speeds of TRACER-equipped Probe Vehicles versus Loop Detectors

Post Mile	**GPS Time (GMT)**	**GPS Latitude**	**GPS Longitude**	**GPS Speed**	**Fwy**	**Lane**	**VDS_ID**	**Loop Time (PST)**	**Loop Count**	**Loop Occ**	**Loop Speed**	**Est. Speed**	**Loop Status**	**Speed diff (%)**	**Est Spd dif(%)**
22.1	0:17:52	33.654783	-117.743	53.8	5 N	1	1204731	16:17:34	0	0	70		3		
	0:18:00	33.656728	-117.7445	55.1	5 N	1	1204750	16:18:19	5	31	59.4	59.4	1	-7.2%	**-7.2%**
	0:18:20	33.66154	-117.7483	53.2	5 N	1	1204766	16:18:19	0	0	70		3		
	0:18:54	33.669173	-117.7544	51.0	5 N	1	1204808	16:19:04	0	0	0		5		
	0:19:00	33.670678	-117.7551	50.3	5 N	1	1204825	16:19:04	20	154	47.8	47.8	1	5.1%	**5.1%**
	0:19:36	33.678963	-117.7593	45.7	5 N	1	1204861	16:19:49	22	168	48.2	48.2	1	-5.3%	**-5.3%**
	0:20:28	33.69022	-117.7683	45.5	5 N	1	1204924	16:20:34	17	113	55.4	55.4	1	-17.9%	**-17.9%**
25.15	0:20:42	33.693328	-117.7703	47.7	5 N	1	1204937	16:20:34	0	0	0		5		
26.35	0:23:34	33.708195	-117.7808	44.3	5 S	HOV	1204965	16:23:40	9	59	68.4	56.2	1	-35.3%	**-21.2%**
	0:24:12	33.700305	-117.7753	49.6	5 S	HOV	1204942	16:24:25	4	25	70	58.9	1		**-15.9%**
	0:25:02	33.688392	-117.7674	51.0	5 S	HOV	1204902	16:25:10	10	57	70	64.6	1		**-21.1%**
	0:25:24	33.682978	-117.7638	53.8	5 S	HOV	1204884	16:25:09	0	0	70		3		
	0:26:02	33.675617	-117.7566	35.0	5 S	HOV	1204839	16:25:54	9	54	70	61.4	1		**-42.9%**
	0:26:28	33.670828	-117.7553	37.3	5 S	HOV	1204814	16:26:39	4	22	70	66.9	1		**-44.3%**
22.75	0:27:20	33.660958	-117.7482	44.7	5 S	HOV	1204759	16:27:24	2	9	70	81.8	1		**-45.4%**
22.1	0:41:10	33.65451	-117.7428	54.3	5 N	HOV	1204729	16:41:30	2	9	70	81.8	1		**-33.6%**
	0:41:18	33.65645	-117.7444	54.7	5 N	HOV	1204748	16:41:30	0	0	70		3		
	0:41:38	33.661282	-117.7481	53.8	5 N	HOV	1204757	16:41:30	3	18	70	61.4	1		**-12.3%**
	0:42:10	33.66901	-117.7543	55.1	5 N	HOV	1204806	16:42:15	2	13	68.3	56.6	1	-19.3%	**-2.7%**
	0:42:18	33.67116	-117.7553	53.3	5 N	HOV	1204823	16:42:15	10	62	70	59.4	1		**-10.2%**
	0:42:48	33.678822	-117.7592	54.7	5 N	HOV	1204859	16:43:00	12	80	66.8	55.2	1	-18.1%	**-1.0%**
	0:43:08	33.68311	-117.7638	55.0	5 N	HOV	1204876	16:43:00	0	0	70		3		
	0:43:36	33.690047	-117.7683	52.6	5 N	HOV	1204922	16:43:45	5	46	48.1	40.0	1	9.4%	**31.5%**
	0:44:20	33.700435	-117.7751	51.8	5 N	HOV	1204956	16:44:30	16	85	70	69.3	1		**-25.2%**
	0:44:58	33.709513	-117.7816	46.1	5 N	HOV	1204980	16:45:15	19	134	63.4	52.2	1	-27.3%	**-11.7%**
26.81	0:45:16	33.712407	-117.7848	42.3	5 N	HOV	1205010	16:45:15	10	49	70	75.1	1		**-43.8%**

Table 2. (continued)

Post Mile	GPS Time (GMT)	GPS Latitude	GPS Longitude	GPS Speed	Fwy	Lane	VDS_ID	Loop Time (PST)	Loop Count	Loop Occ	Loop Speed	Est. Speed	Loop Status	Speed diff (%)	Est Spd dif(%)
28.1	0:47:06	33.72637	-117.8035	25.0	5 S	5	1205105	16:46:52	23	145	70	58.4	1		**-57.1%**
	0:49:28	33.712083	-117.7849	38.5	5 S	5	1204997	16:49:06	9	32	70	103.6	1		**-62.8%**
	0:49:52	33.70812	-117.7808	42.2	5 S	5	1204967	16:49:51	11	51	70	79.4	1		**-46.9%**
	0:50:32	33.700518	-117.7756	41.2	5 S	5	1204944	16:50:36	14	90	70	57.3	1		**-28.0%**
	0:51:32	33.688608	-117.7678	40.6	5 S	5	1204904	16:51:21	9	51	70	65.0	1		**-37.5%**
	0:52:00	33.682932	-117.764	45.6	5 S	5	1204886	16:52:05	9	71	64.3	46.7	1	-29.1%	**-2.4%**
	0:52:22	33.678903	-117.7598	45.5	5 S	5	1204869	16:52:05	0	0	70		3		
	0:52:40	33.675628	-117.7569	37.5	5 S	5	1204841	16:52:50	15	129	59.3	42.8	1	-36.8%	**-12.5%**
23.5	0:53:02	33.670857	-117.7555	44.1	5 S	5	1204816	16:52:50	14	80	70	64.4	1		**-31.6%**
22.1	1:05:50	33.655133	-117.7431	46.8	5 N	5	1204731	17:06:11	0	0	0		3		
	1:05:58	33.656825	-117.7444	48.0	5 N	5	1204750	17:06:12	16	107	70	55.1	1		**-12.8%**
23.5	1:07:02	33.670723	-117.7549	47.0	5 N	5	1204816	17:07:03	9	48	70	69.0	1		**-31.9%**
23.5	1:19:56	33.671023	-117.7549	46.0	5 N	5	1204825	17:19:40	16	91	70	64.7	1		**-28.9%**
	1:20:38	33.679075	-117.7592	45.7	5 N	5	1204861	17:20:25	16	106	70	55.6	1		**-17.7%**
	1:21:00	33.682965	-117.7634	46.5	5 N	5	1204878	17:21:10	8	45	70	65.5	1		**-28.9%**
	1:21:36	33.690335	-117.7683	43.5	5 N	5	1204924	17:21:55	0	0	0		1		
25.15	1:21:50	33.693032	-117.77	40.6	5 N	5	1204937	17:21:55	0	0	0		1		

Table 2. (continued)

Post Mile	GPS Time (GMT)	GPS Latitude	GPS Longitude	GPS Speed	Fwy	Lane	VDS_ID	Loop Time (PST)	Loop Count	Loop Occ	Loop Speed	Est. Speed	Loop Status	Speed diff (%)	Est Spd diff(%)
22.14	23:43:12	33.654705	-117.7429	49.0	5 N	2	1204737	15:43:15	11	64	63.3	63.3	1	-22.5%	**-22.5%**
	23:43:22	33.656885	-117.7446	49.1	5 N	2	1204750	15:43:08	20	151	48.8	48.8	1	0.7%	**0.7%**
	23:43:42	33.661165	-117.7479	46.8	5 N	2	1204761	15:44:00	10	82	44.9	44.9	1	4.2%	**4.2%**
	23:44:18	33.668988	-117.7542	51.4	5 N	2	1204808	15:44:38	19	150	46.6	46.6	1	10.3%	**10.2%**
	23:44:26	33.671022	-117.7551	50.1	5 N	2	1204825	15:44:38	23	195	43.4	43.4	1	15.4%	**15.3%**
	23:45:02	33.679147	-117.7595	47.9	5 N	1	1204861	15:45:23	23	180	47	47.0	1	1.9%	**1.8%**
	23:45:22	33.683142	-117.7637	49.7	5 N	1	1204878	15:45:23	10	112	32.9	32.9	1	50.9%	**51.0%**
	23:45:52	33.690102	-117.7683	49.6	5 N	1	1204924	15:46:08	17	124	50.5	50.5	1	-1.9%	**-1.8%**
25.15	23:46:06	33.69324	-117.7703	46.5	5 N	1	1204937	15:46:08	0	0	0		5		
26.35	23:50:14	33.708137	-117.7807	42.0	5 S	1	1204967	15:50:00	11	90	45	45.0	1	-6.7%	**-6.7%**
	23:50:48	33.700575	-117.7754	53.4	5 S	1	1204944	15:50:44	19	149	46.9	46.9	1	13.8%	**13.7%**
	23:51:38	33.688273	-117.7674	51.8	5 S	1	1204904	15:51:29	16	106	55.6	55.6	1	-6.8%	**-6.7%**
	23:52:00	33.682905	-117.7638	52.1	5 S	1	1204886	15:52:14	9	97	33.8	34.2	1	54.1%	**52.5%**
	23:52:18	33.679057	-117.7597	52.9	5 S	1	1204869	15:52:14	13	108	44.3	44.3	1	19.3%	**19.3%**
	23:52:34	33.675458	-117.7566	52.8	5 S	1	1204841	15:52:14	12	86	50.8	51.4	1	3.9%	**2.7%**
	23:52:50	33.671003	-117.7554	51.1	5 S	1	1204825	15:52:52	20	151	48.8	48.8	1	4.8%	**4.8%**
22.75	23:53:34	33.661175	-117.7485	41.3	5 S	1	1204761	15:53:43	15	121	45.6	45.6	1	-9.4%	**-9.5%**

5 ANALYSIS OF ROUTE TRACKING DATA

This section describes the experience to date with the Extensible Data Collection Unit (EDCU) for studying traffic. The primary source of data presented in this report are GPS readings taken from from several units which have been fielded for varying lengths of time. Three units have been in near continuous use for almost a year, while the other three units have seen less continuous usage. The remaining EDCUs have been used for specific and short periods of time, thus, the associated data is not generalizable and is not analyzed here. It is important to note at the outset that the data collected is inappropriate for drawing conclusions about general traffic flow properties. However, it is perfectly valid to use this data to determine what kinds of questions the EDCU can answer.

5.1 Route Usage Behavior

The primary focus of the work on this project has been to enable the wireless connection between the EDCUs and the Testbed database, and to create web-based interfaces to allow various kinds of limited, authorized access to that data. The real-time data transmission system is relatively stable, having undergone one major revision after the initial design and implementation. At this time, the system is ready to be used for formal experiments.

During the development of the wireless communication system, and over the course of several small pilot studies, hundreds of thousands of GPS records have been collected. We have managed to collect 1,143,665 total GPS points, almost entirely over the wireless data connection. Of these, 693,252 have a non-zero entry for speed, meaning that the vehicle was moving. These at-speed records are grouped into 3,350 distinct trips, although this number overestimates the actual number of trips -- true origin to true destination -- observed. The initial implementation of the wireless system, since corrected, would periodically drop 5 minute chunks of data if the signal was interrupted. We have discovered that once the wireless system is in place, it is much easier to leave the unit in the vehicle and rely exclusively upon wireless data collection. This contrasts with our initial intent of using the wireless collection as a backup to the physical storage on a flashram card. We estimate that the 16MB flashram card being used in the EDCUs can store approximately a month of very heavy travel without any problems. However, we have three units that have been fielded for much longer periods, with the longest duration extending for over 9 months of continuous data collection without accessing the flashram.

The collected data points extend throughout southern California, from Santa Barbara in the north, to the San Diego in the south, and from the coast to Riverside, San Bernardino, and the Palmdale area. A truncated map, covering only the Orange County area, is shown in Figure 11.

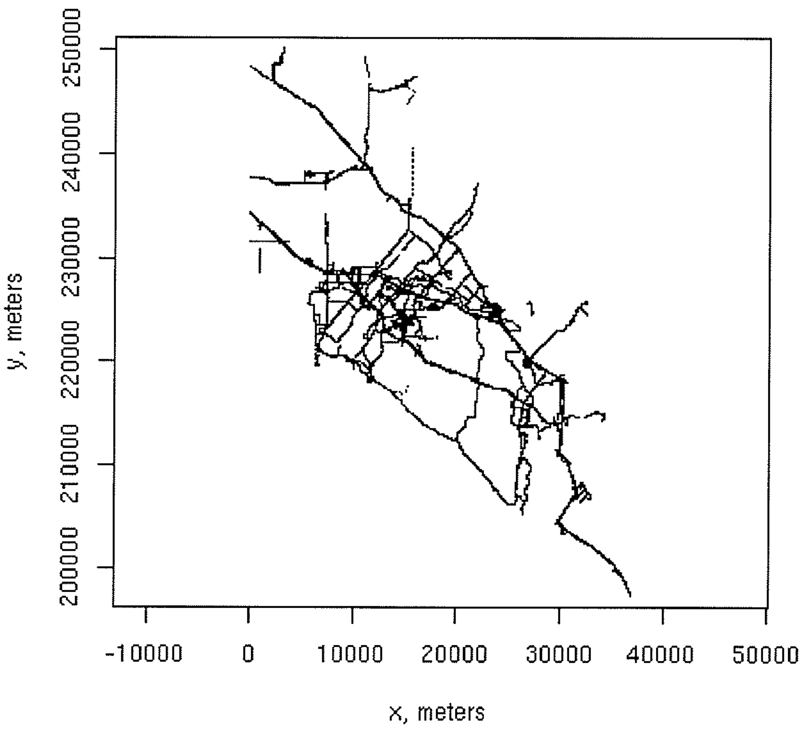

Figure 11. Central Orange County Represented by GPS Data Points

The detail available within this data set is remarkable. Figure 12 shows a detailed section of the larger map, zoomed in to the Newport Beach, Costa Mesa, and Irvine areas. Figure 13 zooms in tighter to show just those point in the immediate vicinity of UC Irvine. Finally, Figure 14 shows the detail available at the street level, where individual GPS points can be resolved and the 10 meter RMS error of the GPS measurement is noticeable. Note that direct GPS data was utilized without differential correction. The EDCUs are currently being enhanced with WAAS GPS which promise differential GPS accuracy without the need to apply the corrections.

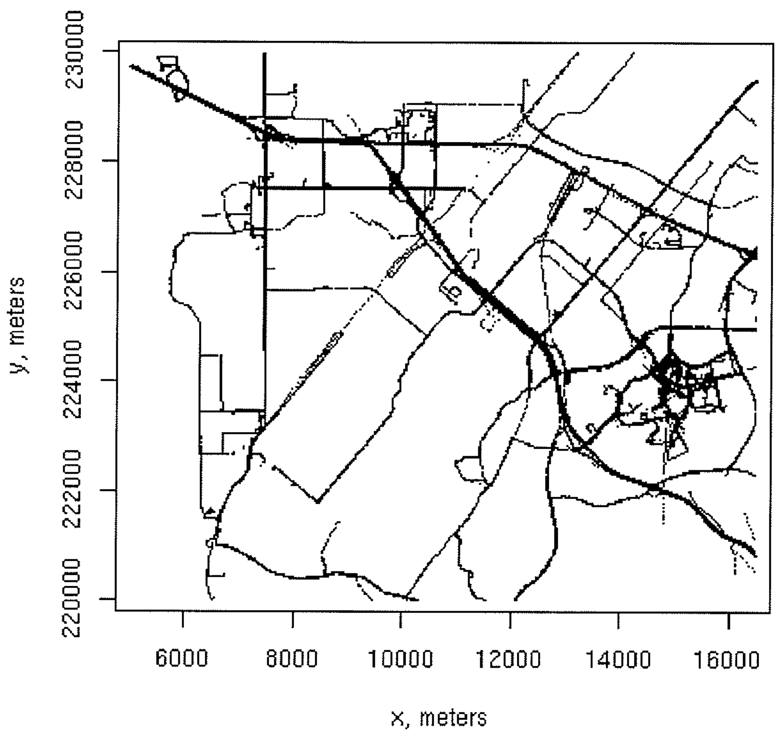

Figure 12. Image from Figure 11 Zoomed onto Irvine Area.

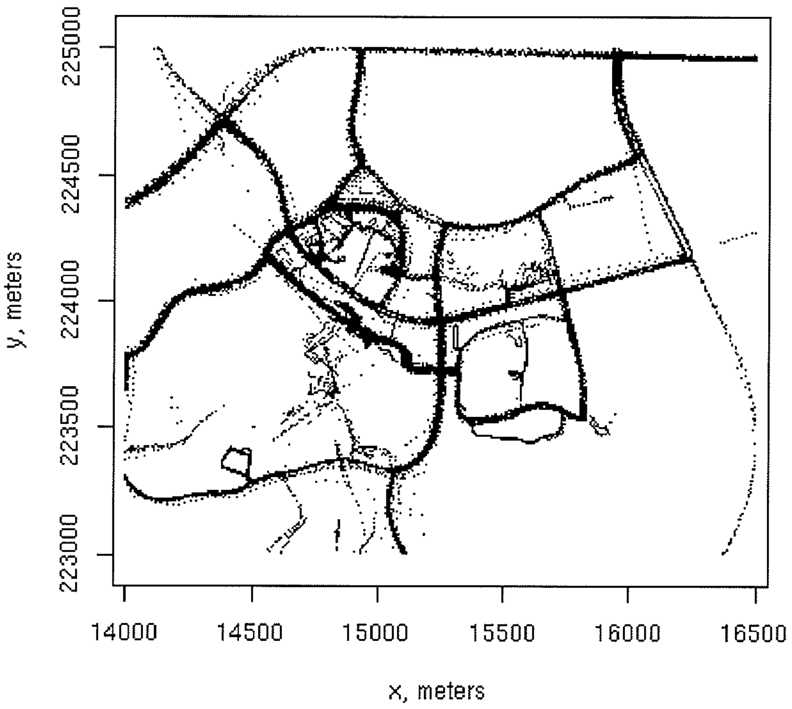

Figure 13. Image from Figure 12 Zoomed into UC Irvine Area

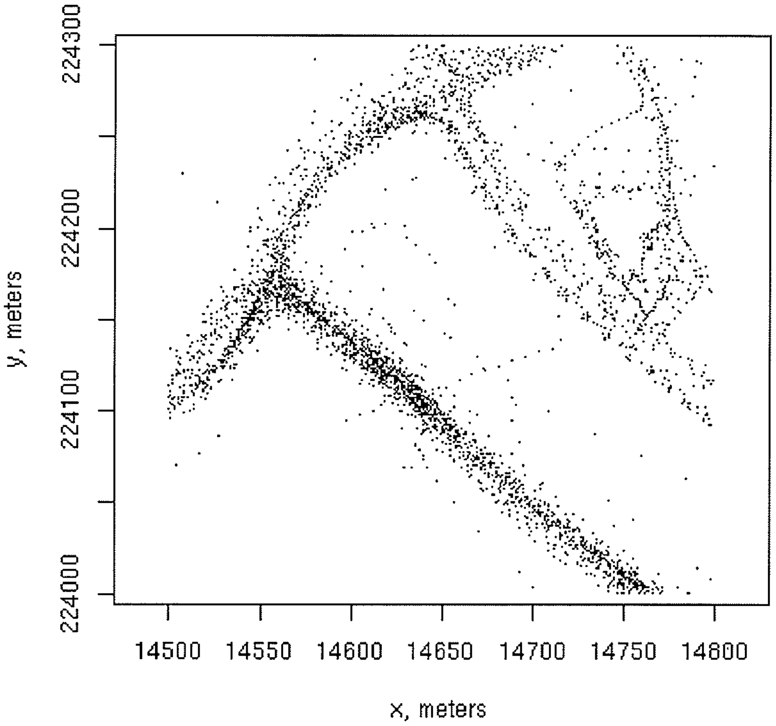

Figure 14. Image from Figure 13 Zoomed into UCI Intersections

The amount of data and the level of detail are remarkably easy to collect. All that is required is to place the EDCU in a vehicle. The only caveat is that the data are biased to show only where drivers travel. While a broad, representative sample of the population of licensed drivers will most likely capture 90% of all major streets and freeways, one can never be certain of collecting data on every street and intersection unless trained probe-vehicle drivers are specifically instructed to methodically cover every street.

5.2 Observing Traffic Flow Variables

The EDCU offers two methods for examining traffic flow. First, using the cellular data connection, we can field a fleet of vehicles and coordinate all of their data streams to develop a real-time picutre of the traffic conditions for the network as a whole. Second, individual vehicles can be operated independently of each other to provide unique snapshots of the drivers' perspective of the transportation network. We have not attempted to field a fleet of vehicles simultaneously. Instead all of our data collection to date has been with devices placed with independent drivers who merely record their daily travel patterns. This section will first discuss some of the lessons learned from following individual vehicles, and then present some thoughts on the implementation of a probe fleet.

One of the immediately obvious results of monitoring a vehicle for an extended period of time is that repeated travel patterns are observed for the routes that the vehicle uses. A daily trip to and from work, for example, typically uses a handful of routes and a small slice of time. The variablility of speeds experiences in these times is fairly narrow.

However, the speeds, acceleration profiles, and routes that are chosen are clearly a function of the driver of the vehicle. Out of four different EDCUs that were fielded for more than a month with drivers coming to and from the UC Irvine campus, all four vehicles used different routes onto campus, and different roads or freeways getting to campus. Even when traveling on the same road segments, the drivers traveled at different times of the day in different traffic conditions. Therefore it would be difficult to draw general conclusions about the features of a traffic stream from these kinds of measurements.

If a fleet of vehicles is fielded for the express purpose of monitoring traffic, the current EDCU design will allow all fielded devices to communicate with the central database at the same time. At the moment the capacity of the system is not limited by the server hardware, since the wireless cellular digital packet data (CDPD) modems on the EDCUs allow only about a 6 to 9 Kbps bandwidth. Ideally, the movements of the fleet should be coordinated by the central location to provide complete coverage of interesting traffic flow features.

In theory, the more vehicles that are so-equipped, the better the resulting "movie" of network conditions should be at any given time. But by the same token, trying to monitor the flow on a large proportion of the network at all times may not be the most appropriate use of a probe vehicle fleet, given the large body of data already available from loop detectors (El-Araby *et al.*, 2001). It is difficult to predict how many vehicles would be needed to provide adequate coverage, or even what adequate means in this case. Rather than complete characterization of instantaneous traffic conditions, it makes more sense focus on gathering new information, rather than revisiting old information. At first this will mean gathering flow information through bottlenecks and incidents. But over time those incidents will become well known, and the probe vehicle fleet will have to be turned towards new problems.

We foresee three different research projects that can and should be performed using our EDCU devices. First, we should explore the parameters related to fielding a sparse fleet of probe vehicles which attempts to characterize specific traffic events. Using the 30 devices, we should be able to obtain nearly complete coverage the Irvine testbed freeway and arterial segments during morning and evening rush hour periods. This will allow us to decide the measurement precision of a probe fleet per lane-kilometer of roadway, the theoretical limits of measurement precision, the kinds of traffic events which yield well to this type of measurement, and so on.

Second, we also need to study algorithms to assign probe vehicles efficiently when trying to measure different events. For example, rush hour congestion may require a completely different deployment of probe vehicles than special event management. Furthermore, once several examples of a particular kind of traffic flow event have been monitored and measured in great detail by the probe vehicle fleet, one would expect future measurements of similar events can concentrate on refining our understanding of the details of the event rather than re-establishing the known average characteristics, if any.

Finally, the third project that should be carried out is a simulation study of the potential for up to 100% of the traffic vehicle stream to be equipped with broadcasting GPS devices. If in the future all vehicles have this capability, the a probe vehicle fleet will rapidly become an unnecessary expense. On the other hand, in order for us to make sense of the mountains of data and in order to give the individual drivers the tools to process and use the data they are collecting, we will need to have algorithms in place -- ones which can only developed through experimenting with a probe fleet.

5.3 Matching Observations to a Network Representation

The first step in methodically tackling the traffic flow measurement problem is to accurately translate, in real time, the position of a vehicle in time and space to a network abstraction. While there are many commercial applications which clearly have solved this problem, their implementations are closed and expensive. We feel there is academic value to be gained from revisiting this problem. The recent published literature on this problem focuses on obtaining very fast matches between a measured position and a known network. For example, Zewang *et al.* (2002) use a 1-km moving window to limit the number of links that are considered for matching, and then apply Dempster-Shafer evidence reasoning to fuse the evidence supporting multiple matches between sequences of points and the known links. Pyo *et al.* (2001) take a similar approach, using the multiple hypothesis technique to carry forward multiple possible matches to a sequence of points. They add a Kalman filter step to remove bias in the digital road map data. Their method takes a computational hit due to the need to generate pseudo-measurements for each potential match hypothesis. Joshi (2001) proposes a more efficient matching technique based on a rotational variation metric. A rotational variation metric measurement of zero means that two shapes are identical aside from rotation and translation. This approach recognizes that translation and rotation are shape and size preserving transformations. Even if a measurement and a road digitization both contain errors, good matches between the two can be found by simple rotation and translation of the more interesting portions. The result is a computationally efficient method of map matching. However, the method has a critical dependence upon detailed descriptions of the shape of a link, and upon those shapes being interesting and distinct. It may have difficulty matching against sparsely characterized networks (such as the publicly available census TIGER/Line data) or against highly regular street grids

Our interests lie in generating a two-way comparison, one which includes the instantaneous vehicle to network matching problem, as well as in creating algorithms for incremental improvements to the underlying network representation (enhancing the shape points, and so on). Preliminary work in this area has focused on the idea that the map matching problem is really an instance of the Error Correcting Subgraph Isomorphism problem, a well studied problem which is known to be NP-complete. We have not found any published references which formally establish this equivalence, and so we hope this research will shed new light on the general map matching problem. This work is ongoing, and the formal proof of the equivalency between the problems is not yet complete. While the general class of problem is likely to be NP-complete, we hope to use various properties of the map matching problem to augment algorithms that have been developed for the error correcting subgraph isomorphism problem. Specifically, the Euclidean distances and sparse connectivity of transportation networks may allow very good approximate solutions to be computed in polynomial time. These algorithms will hopefully form a significant contribution to the map matching literature.

5.4 Automatic Mapping of the Transportaiton Network

The solution approach discussed above to the map matching problem can also be used to generate the network representation itself. As we mentioned above, most commercial map matching applications strictly map vehicle positions onto a fixed, known network. If the network changes, then presumably the in-vehicle device will have to be updated with a new map. In addition, methods like the rotational variation metric will improve if the digital representation of the network is highly detailed.

We think a feasible, general solution is to have the vehicles generate their own "ground truth" about the parts of the network on which they travel every day. Actual movements can be used to augment the shape points of the underlying network, as well as to modify links when they change. Related work is presented in Li and McDonald (2002) in which the authors focus on the estimation of travel speed on a link, exclusive of other characteristics.

For an example of the need for automatic map generation capabilities, consider Figures 15 and 16. Between October 2002 and February 2003, the offramp alignment was modified to accomodate a widening of the on and off ramps in conjuction with the construction of a new shopping mall. The offramp alignment will change again once the construction is completed. Small changes like this occur frequently, and the regular users of these facilities can generate automatic corrections to the standard maps.

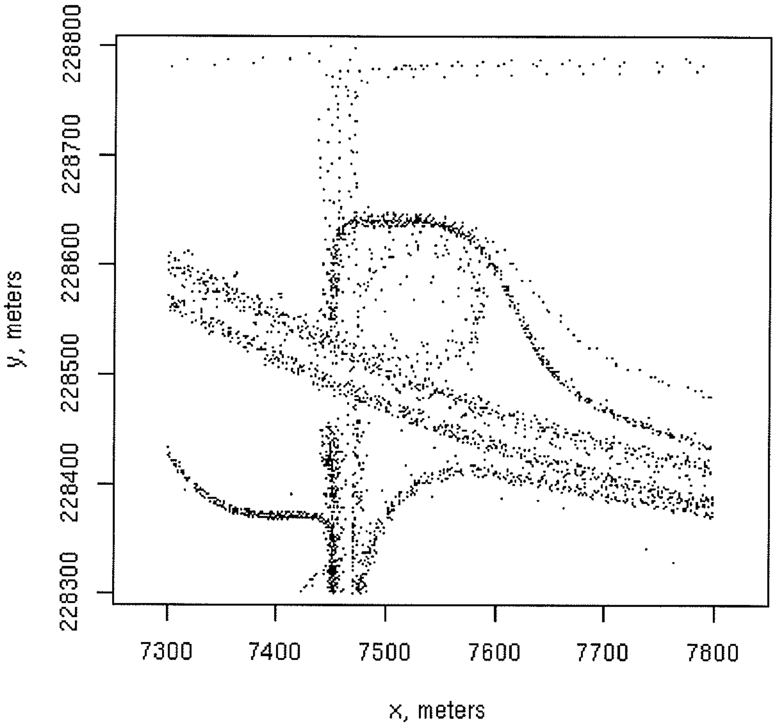

Figure 15. GPS Data Points at I-405 and Harbor Blvd Interchange

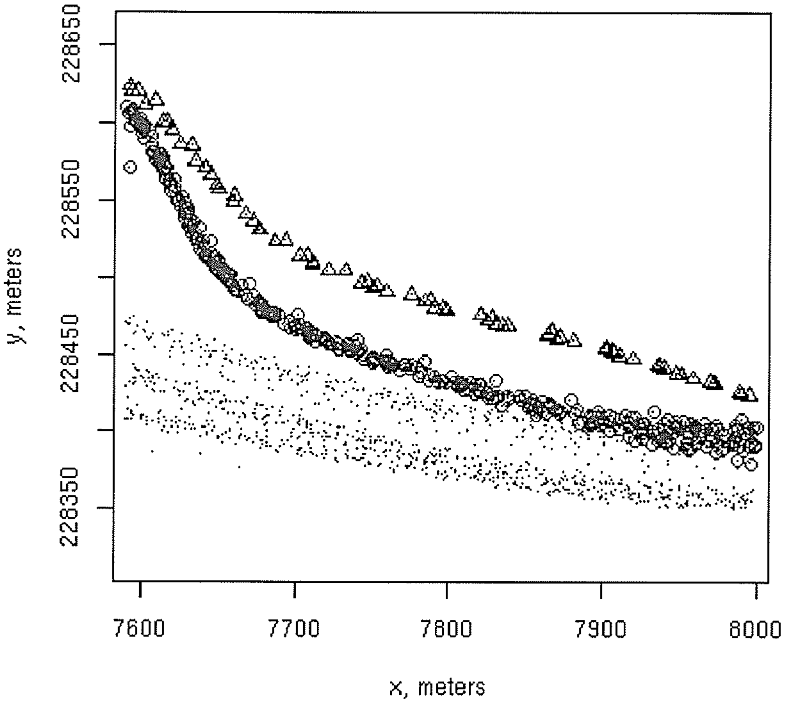

Figure 16. Figure 15 Zoomed to Ramps I-405 NB to Harbor Blvd

As a more ambitious extension, it should be possible to build up a reasonably accurate, driver-centric transportation network representation using only the data gathered from extended data collection. The entire traffic network will not be traversed by a single driver through repeated driving. However, if each driver with an in-vehicle GPS device can create their own image of the traffic network, then the combination of these driver centric networks should have very good coverage of the entire traffic network. There are many problems that must be solved before this is possible in a general sense. Some of these problems are:

> Detecting the location of stop lines and intersections. Stops can occur well back from the stop line, and also can occur due to traffic rather than signals. Other strategies include looking for two or more paths joining or splitting and looking for right angle turns.
>
> Determining roadway type (two-lane, four-lane, etc), approximate speed limits.

We have a small project using the EDCU devices to track UC Irvine campus shuttles on their daily routes. We will use this project to develop some initial algorithms, and explore the potential to generate a simple network abstraction. If this project goes well, we will expand the work.

5.5 Aggregating Observations over Time and Space

Another related problem is how to aggregate individual measurements over time and space. Matching a geo-coded point to a link on a network serves as a de facto spatial aggregation technique. However, this is not necessarily a useful approach to take in all circumstances. For example, the raw data may reveal consistently different speed profiles before and after some problem point in a link. Examining the raw data might show slow speeds at the bottleneck, followed by faster speed downstream from the bottleneck. Abstracting to the link level will obscure this with an overall link average speed.

One would expect that repeated measurements can generate very good aggregate estimates of the actual road parameters, according to the central limit theorem. \cite{michler:01statis} explores the application of the central limit theorem to speed estimation using floating car data under different, typical traffic patterns. However, spatial aggregation of GPS records must respect the strong correlation between space (longitude, latitude, elevation), time, and the measurement of that space, speed, and time by the GPS device. GPS antennas are remarkable devices, but their 10 meter RMS accuracy depends critically upon being able to see four satellites well separated in the sky. If the satellites are clustered, or low on the horizon, then the GPS measurements will be more inaccurate. Further, all measurements taken for a particular trip are likely to rely upon the same constellation of GPS satellites. Finally, any areas with bad coverage due to trees or buildings are likely to always have bad coverage. The measurement of position and speed in those places is likely to have completely different distribution of errors than other places.

A further complication is the fact that vehicle positions are not free to vary over the entire 2-D plane. Instead they are confined to roads and parking lots. The layout of roads differs from point to point and region to region. Without correcting for the layout of roads, the analysis of the positions is bound to be flawed. Analyzing the observed GPS points with an image processing routine to extract the road features, for example, will result in errors because the discontinuities at the edges of the roads will not be handled properly.

As an example, consider again the changing offramp geometry shown in Figure 16 and repeated in Figure 17. A visual inspection of the points in the figure resolves into a main freeway segment and two separate offramps. If one looks carefully, there are two distinct streams of traffic (northbound and southbound) on the main freeway segment, and a third onramp at the bottom of the image for the southbound stream (see also Figure 15). One would want to have an automatic data processing routine detect at least 3 but preferably 5 different groups of points, one for each feature that a person viewing the points can detect.

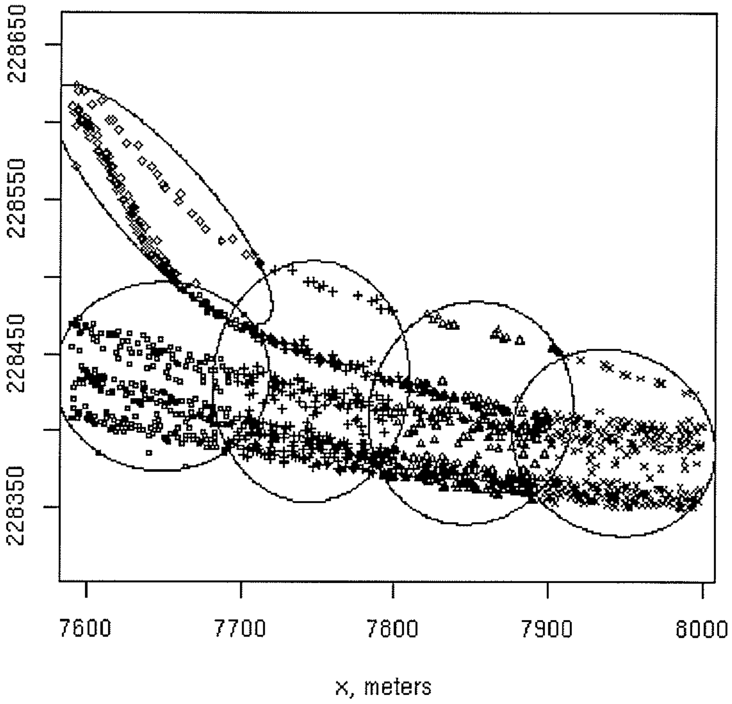

Figure 17. K-means clustering of Figure 16 Data Points

Figure 17 shows the five clusters generated by a standard K-means cluster algorithm. Each cluster is plotted with a different symbol, and the five groups have been circled with ellipsoids. Not only are the groups not the desired lanes and offramps, they are clustered perpendicular to the desired direction. The reason for this is that the default K-means cluster approach is to add new points based on the closest average distance to all points in a cluster so far. Changing this algorithm to add points that are closest to any point in a cluster might result in better cluster shapes. The results of just such an agglomerative algorithm is shown in Figure 18. This approach does captures the right direction of the desired clustering. However, this clustering requires nearly 40 groups to separate the new offramp from all other observations, and almost 60 groups to separate the tail end of the original offramp from the main flow of the freeway. The initial clustering step must be followed by a second clustering step to properly group the new offramp, old offramp, and freeway segments. This is not feasible for a general application.

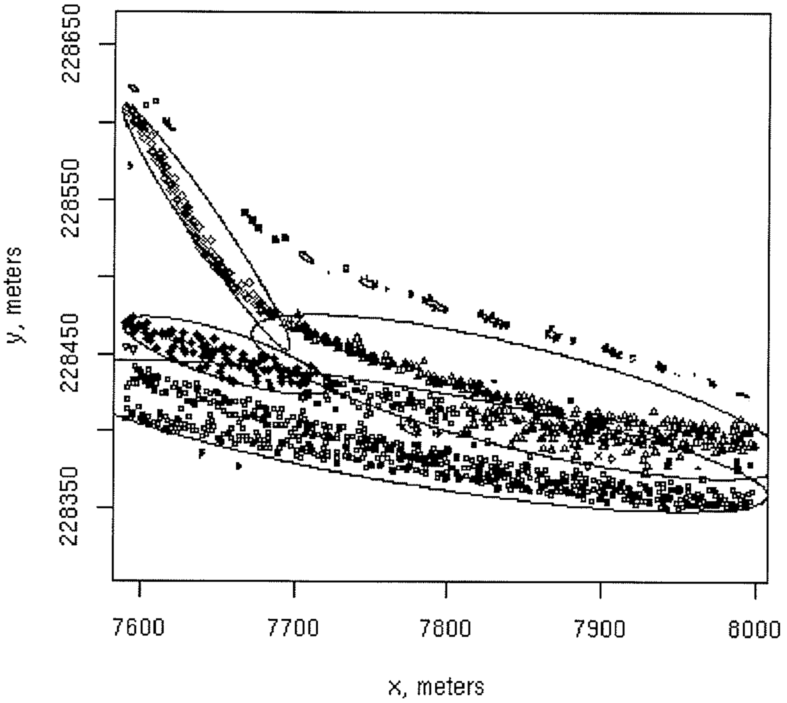

Figure 18. Agglomerative Clustering of Figure 16 Data Points

In short, standard clustering routines and image processing techniques are inadequate for aggregating vehicle-based GPS observations. What is needed is an algorithm which first matches observations to a network representation, and then computes local statistics based on the directionality of movement and the knowledge of the road network. The direction of movement can be taken either from the heading reported by the GPS, or by processing prior points in the data stream. The local road geometry at a minimum will include the knowledge that lanes are typically 12 feet wide. If more detailed geometry is available after matching the observations of interest to the network representation, then that geometric information should also be used. The aggregation of spatial statistics should then be performed in such a way that the correlated errors within each trip (with the same satellite constellation) and between trips (with the same buildings and trees blocking the satellites) are incorporated correctly.

5.6 Summary

This report has presented the some preliminary experience with gathering and processing travel data with the Extensible Data Collection Units. In the course of testing the EDCUs and fielding small pilot surveys of various kinds, we have managed to collect 1,143,665 total GPS points, almost entirely over the wireless data connection. Of these, 693,252 have a non-zero entry for speed, meaning that the vehicle was moving. These at-speed records are grouped into 3,350 distinct trips, although some early problems with the wireless data collection system mean that this number overestimates the actual number of origin to destination trips observed. We have discovered that once the wireless system is in place, it is much easier to leave the unit in the vehicle and rely exclusively upon wireless data collection. This contrasts with our initial intent of using the wireless collection as a backup to the physical storage on a flashram card.

Given the informal nature of the data collected so far, it is difficult to justify any theoretical results. However, we have used the collected data to get an idea what kinds of data can be collected, and what kinds of research should be pursued. The research directions include projects that make use of the current capabilities of the EDCU, as well as projects which extend those capabilities.

The most important lesson learned is that it is extremely easy to collect vast amounts of data. Therefore it is vital to be able to store and process that information efficiently. Future research will include development of these kinds of data processing and data mining algorithms.

6 TRACER MODULE TECHNICAL SPECIFICATIONS

Because the Tracer equipment is intended to fill multiple data collection roles with significantly different requirements, versatility and portability were primary design goals. The data collection system therefore employed a modular design to ensure such flexibility. The final design comprised the following modules and implemented the required functionality as an embedded system:

1. DUC: Core Data Collection Unit
2. GPS: Global Positioning System Module
3. LOG: Data Logger/Storage Module
4. COM: Communication Module
5. UI: User Interface
6. BSU: Base Station Unit

For the remainder of the document, both "data collection system" and "system" refer to the complete set of modules, and "data collection unit" and "unit" refer to the modules that are placed in the vehicle (i.e., all modules except for BSU), and "device" refers to specific pieces of hardware used within a module (e.g., a Garmin GPS device was used to implement the GPS module).

Basic system functionality requires the DUC, GPS, and LOG modules; the COM, UI, and BSU modules represent extensible system components which were not necessary for the probe vehicle study. Table 2 shows the modules required by each data collection application.

Table 3. System Components by Data Collection Application

System Component	Basic Survey	Probe Study	Enhanced Survey	Routing Study
DUC	Yes	Yes	Yes	Yes
GPS	Yes	Yes	Yes	Yes
LOG	Yes	Yes	Yes	Optional
COM	No	Yes	Optional	Yes
UI	No	No	Yes	Yes
BSU	No	Yes	Optional	Yes
Yes = required for application; No = not required for application				

6.1 DUC Module

The core data collection unit (DUC) is responsible for controlling the operation of the data collection unit. This required a device that interfaces with all system modules and executed control programs that manage the unit during data collection applications, designed as a programmable embedded system. Specific design requirements for the DUC included: (1) the computing platform (hardware architecture, on-board memory, and operating system), (2) device hardware interfaces (e.g., serial ports, and power connections necessary for interfacing with and supporting other modules), (3) dev-API (APIs encapsulating device functionality for application programming), (4) application control program (the controlling logic and app-API for a data collection application), and (5) power management. Requirements 1 and 2 were fundamental design elements that defined unit versatility. Requirements 3 and 4 reflect the APIs that define device functionality.

The EDCU was conceptualized as a fully programmable embedded system. The control program runs on top of an OS using the software drivers that present the dev-API to access the modules required by a given application for data collection (e.g., GPS), and data storage (e.g., LOG). The control program presents the app-API on a terminal device and/or over the network.

Table 4. DUC Module System Requirements

Module Feature		Design Requirement
Platform	Architecture	Intel Pentium (PC104)
	Memory	16 Mb on-board memory for OS & control program
	OS	LINUX for real-time, device and network management (includingTCP/IP)
Hardware	Interfaces	Slots for all modules Serial port for VT100 diagnostic access Slot for network hardware
Software	Device	Software drivers for all modules Implementation details behind dev-API interface
	Application	Control programs for managing each application App-API for each data collection application
Power	General	Power source for all other unit modules
	Battery	Battery backup and power management
	External	Externally powered through vehicle auxiliary port (e.g., cigarette lighter); auto-off time-out

6.1.1 Computing Platform

The choice of computing platform was the most important factor affecting the versatility of the data collection unit. The DUC utilize a Pentium 2 class processor with processing power sufficient to support all applications of the system as envisioned. The DUC contains sufficient on-board memory to support the operating system and execute the controlling programs for all defined applications. While several operating systems (OS) were considered, Linux provided the required features, including:

real-time operation;
device management;
network support (including TCP/IP networking);
support for cross-platform development; and
known CORBA implementation

Power-management functionality that may place special requirements on the operating system or initialization of the system from ROM was considered.

6.1.2 Device Hardware Interfaces

The DUC provides hardware interfaces for connection of other modules to the system. These interfaces support connection of all defined system modules. The DUC provides a serial port for connection of a VT100 terminal and also supports a network card, elements that are required for programming, debugging, testing, and field preparation of the data collection unit.

6.1.3 Device API

Each module supports a fixed API which defines the functional capabilities of the module (for further information see the API section for each module). The underlying implementation of these interfaces is hidden behind the API. The collection of APIs provided by each module connected to the DUC comprise the dev-API of the DUC. The dev-API defines the data collection functionality of the complete data collection unit and ensures that the on-board application control program accessing the module via its API will not need any direct knowledge of the actual hardware implemented for the different modules in the data collection unit.

6.1.4 Application Control Program

The data collection unit will be used for a number of different data collection tasks and the control logic for each of these applications are expected to vary, therefore, this logic is implemented by a control program operating in the DUC.

The control program performs two functions. First, it controls execution of the data collection application by coordinating operation of connected modules to achieve the functionality required for the application. To maximize modularity in the data collection unit, the control program utilizes the dev-API and avoids low-level device operations that rely on knowledge of underlying implementation details. Second, the control program provides an app-API as a means for interacting with the data collection application. This interaction varies with the application, but minimally includes setting application parameters (e.g., data storage rates), downloading logged data, and real-time interactivity. The app-API is accessible from all external system interfaces including serially connected terminal devices and network connections (e.g., ethernet or via the COM module). Each application is managed by a stand alone control program

6.1.5 Power Management

The DUC provides and manages power for all modules. The vehicle itself is the primary power source (via an auxiliary port such as a cigarette lighter) and a rechargeable battery back-up is also integrated. The DUC is activated by a motion sensor and requires power from the vehicle (or an auxiliary battery back) to continue activation. The power management design ensures minimal impact on vehicle operation and the DUC can not drain the vehicle's battery when the vehicle is not operating. The battery back-up provides for a graceful shut-down in the event of a sudden loss of primary power. Such sudden power "failures" might occur each time the user turns off the engine. The DUC controls power delivery to any and all externally connected modules, as applicable. Thus, a UI module which is not integrated into the data collection unit will have power as well as data passed from the DUC.

6.2 GPS Module

The GPS module is responsible only for obtaining positional fixes from satellites. Primary design concerns include: (1) maintaining satellite signal lock, (2) antenna placement, (3) positioning accuracy, (4) APIs, and (5) power management requirements. These requirements are summarized in Table 4 and are detailed in the following sections.

Table 5. GPS Module System Requirements

Module Feature		Design Requirement
Signal	Satellite Fix	Less than 30 sec hot reacquisition
	Antenna	Independent GPS with external connection to EDCU; antenna may be externally (magnetic or trunk lip) or internally (front or back dash) mounted
	Accuracy	100m with Selective Availability (SA) degradation
Interface	Protocol	Supports full NEMA-0183 sentences [including GPS fix data (GGA); geographic position, latitude and longitude (GLL)]
Power	Main	Power from DUC module
	Battery	Battery backup from DUC Module

6.2.1 Satellite Signal

The GPS module is external to the EDCU. The Garmin GPS selected is a self-contained unit (antenna and hardware in a mouse-sized unit available in a variety of mounting options) connected to the DUC via a co-axial cable. The GPS module continuously determines vehicle location via a "3-D" triangulation. Current GPS technology does not guarantee constant contact with 4 or more satellites, thus, occasional errors due to loss of satellite contact will occur.

6.2.2 Antenna Placement

The GPS module selected for initial deployment (Garmin Trakpak 35) provides a range of mounting options to be non-intrusive while securing mounting inside or outside the subject vehicle. The antenna does not require any mounting or remounting by drivers. Preliminary tests suggest interior or exterior placement works equally well in flat suburban environments such as the Irvine area. Time for acquisition of signal upon hot reboot (a GPS receiver assumes that a receiver has all positional information stored in memory so that it can quickly reacquire satellite fixes) is less than 30-seconds.

The mounting mechanism used in all trials was a soft magnetic pad that screws to the GPS unit. No operational problems were observed, nor has the mounting marred, dented, scratched, or otherwise damaged test vehicles. The subject vehicles are often the property of survey volunteers, and as such need to be treated with the utmost care and consideration.

6.2.3 Positioning Accuracy

The GPS module provides longitude/latitude positions with an error no larger than 100-m when SA is active.

6.2.4 Application Programming Interface

All modules define an API for software control of the component. Where appropriate, software drivers for devices are provided to support the API. The API for the GPS module supports all portions of the NMEA-0183 interface standard (although only selected portions are currently utilized).

6.2.5 Power Management

The GPS module supports power conservation function implemented in the DUC. In addition, all power necessary to operate the GPS is provided by the DUC. When in standby mode, the GPS module maintains all information necessary to obtain a rapid satellite fix upon return to normal operation.

6.3 LOG Module

The LOG module is responsible for storing any data produced by the unit's modules over the course of the data collection period. This data includes but is not limited to: NMEA sentences produced by the GPS module, survey responses received from the UI module, and data uploaded from the BSU via the COM module. Basic LOG module parameters include:

> The maximum survey duration is at least 7 days.
> The vehicle will not be in motion for more than an average of 4 hours per day over the course of a survey.
> The module can logging position each second while the vehicle is in motion.
> The module is capable of writing data out to the DUC upon request, both on a record by record basis, and in small (approximately 1 kb) blocks. The module can store any data sent to it by the DUC (limited by the maximum capacity).

Conceptually, this module is a storage medium for the DUC module and may be implemented as such. The main design concerns are: (1) hardware, (2) API, (3) data storage capacity, and (4) power management. Requirements in each of these areas are summarized in Table 5 and are detailed in the following sections.

Table 6. LOG Module System Requirements

Module Feature		Design Requirement
Hardware		16-32 Mb Flash RAM
Software	System	Stored on flash RAM then boots to RAM
Interface	I/O Protocol	Standard access
Storage		16-32 Mb Flash allocated to LOG and software
Power	Main	Power from DUC module
	Battery	Battery backup from DUC Module

6.3.1 Hardware

The hardware implementation of the LOG interface is constrained only by size and power limitations. As implemented, 16 Mb flash RAM cards are utilized, but these are expandable. The primary role of the LOG module is to collect and store GPS position data for subsequent processing.

6.3.2 Application Programming Interface

Each module defines an API for software control of the component. The LOG module accepts and logs NMEA-0183 sentences dumped by the GPS module and passed through the DUC. This logged data is retrievable via a simple interface (e.g., a read memory command from the DUC). For future extensions and general maintenance, the LOG module was designed to mimic the behavior of a data storage device on a computer, with most of the data processing and addressing functions being handled by the (programmable) DUC.

6.3.3 Data Storage Capacity

Based on estimations of maximum data demands over all planned applications, the LOG module was designed with removable 16 Mb Flash RAM cards.

6.3.4 Power Management

The LOG module receive power from the DUC during normal operation. Battery back up is required to ensure data is not lost during standby and power-off periods. The DUC provides sufficient power at the start of standby and power-off periods to ensure a graceful transition, with no data lost.

6.4 COM Module

The COM module is responsible for providing communications links for transmission of data between the BSU and the data collection unit in near real-time. Assumptions regarding the COM module include:

> The unit will only be used in developed areas with full access to a wireless communications network
> Real-time is defined as continuous communication of data to the BSU.
> Near real-time is defined as communication of position information that occurs no less frequently than one successful data transfer per minute of all positional information since the last successful data transfer.

The main design concerns are: (1) hardware, (2) communications link reliability, (3) communications network and protocol, (4) bandwidth, (5) APIs, and (6) power requirements. Requirements in each of these areas are summarized in Table 5 and are detailed in the following sections.

Table 7: COM Module System Requirements

Module Feature		Design Requirement
Hardware		Commercial Cellular Digital Packet Data (CDPD) modem mounted inside DUC module (EDCU)
Signal	Antenna	External and internal (whip or other) mag mounts
	Bandwidth	Nominal 19.2 kbps
Power	Main	Power from DUC module

6.4.1 Hardware

The design of the COM module focused on the hardware and software necessary to communicate a stream of data from the data collection unit to the BSU over a wireless communications network. While no restrictions were placed on the wireless communications technology, preference was given to designs that utilize commercial wireless communications networks, such as cellular digital packet data (CDPD) networks, or digital cellular networks. Although CDPD coverage was not available in the Irvine area during the design phase, it was subsequently expanded throughout southern California during deployment. Initial testing of the units with CDPD modems was done in other locations, followed by full testing in Orange County (and several other metropolitan areas since). The COM module is an internal component of the data collection unit linked to the DUC, and this link also provides power to the COM module.

6.4.2 Communications Link Reliability

The COM module minimally provides one successful two-way data transfer per minute between the data collection unit and the BSU, while operating in test environments. The current implementation transfers are positional data in real-time to a web site where vehicle tracings can be displayed. All data is logged on a base server. No testing of Route Guidance applications, and the necessary BSU to COM communication links has yet occurred.

Two antenna options exist for the COM module, both external to the unit with suitably non-intrusive and secure mountings. These include a standard cellular "whip" antenna, also external to the vehicle (a magnetic mount, although other semi-permanent mountings are available) and a magnetic-mounted block antenna (similar in size to the GPS unit) which could also be mounted inside the subject vehicle. As with the GOPS antenna, the COM antenna does not need to be mounted or remounted each time the vehicle is used by a survey participant.

6.4.3 Communications Network and Protocol

A third-party communications network was selected, initially AT&T Wireless. [http://www.attws.com/]

6.4.4 Communications Bandwidth and Speed

NMEA-0183 protocol requires a 4800 baud transmission rate. Other applications (e.g., route guidance mode) may require more bandwidth. A quick perusal of available digital cellular modems revealed that the minimum nominal data transfer rate was 19.2 kbps. This is met by the selected service provider.

6.4.5 Application Programming Interface

Each module defines an API for software control of the component and the software drivers are provided to support the API.

6.4.6 Power Management

The addition of the COM module does not significantly affect the power management functions of the DUC. When the DUC decides to put the data collection unit into standby mode, the COM module also shifts into standby mode.

6.5 UI Module

The UI module is required for the functions of the enhanced survey and the route guidance study, and as such, the basic design does not include this component. The current units do allow for and facilitate the future implementation of the UI module, reflecting the following basic functionality:

field programming of the COMM, GPS, and LOG modules;
an interface for interactive survey applications; and
displaying graphics, sound, and/or text to the vehicle driver.

The main design concerns were (1) platform, (2) modes of operation, (3) flexibility and extensibility, (4) APIs, and (5) power management.

6.5.1 Platform

The UI might consist of a graphical display screen and an input device. A simple solution would be to pair a liquid crystal display with a small keyboard, and use the DUC to render the graphics and process the input as necessary. Another solution would be to use a programmable PDA, perhaps coupled with off the shelf programs, such as mapping and GIS software to present the user with current position. If the UI is not designed into the data collection unit, then a live (powered) data connection should be provided part of the data collection unit.

6.5.2 Modes of Operation}

Each of the three expected uses for the UI module requires different capabilities. The setup function requires the ability to enter simple instructions that are sent to the DUC in the basic data collection unit. These instructions are envisioned as being survey program initialization commands, such as entering the local position, specifying the desired GPS module polling rate, or simply instructing the survey program to start.

Conducting an enhanced survey requires the ability to present the user with questions relating to travel decisions. In addition, the UI shall be capable of recording both preset and open-ended responses. For both of these, the use of a keyboard is preferred over the use of a stylus, since a keyboard is better suited to entering sentences. Finally, a route guidance study requires the ability to present the driver with recommended routes and/or current travel information. Thus the UI shall include a graphical display and the associated hardware necessary for rendering small, readable maps. It is expected that these maps may push the envelope of both processing power and readability, however, the display should present:

the local area, centered on the vehicle, updated every second;
the vehicle's planned route;
at least one alternative route to the planned route;
short alpha-numeric labels for optional routes to indicate expected times

Finally, the UI, when operating in this mode, shall also allow for user input, such as a tap on a keypad to record the driver's decision to use an alternate route.

6.5.3 Flexibility and Extensibility

The UI module shall support application design in a high-level language such as C, C++, Java, etc so that customized interfaces can be designed in the future.

6.5.4 Application Programming Interface

All modules shall define an API for software control of the component. Where appropriate, software drivers for devices shall be provided to support the API. A programmable interface for rendering graphics and posing questions is desired. The input device should be easily accessible to first time users, but a PDA stylus is acceptable, but a small keyboard would be ideal.

6.5.5 Power Management

As with all other modules, power shall be provided and controlled by the DUC. If the UI is powered by an independent battery, as with most PDAs, then the module shall be capable of a standby mode, and shall enter this mode when the DUC decides to power down. The unit shall not place any power drain on the DUC-provided power during standby mode.

6.5.6 Design Recommendation

Because the operational needs of the UI are stringent, it is likely that a PDA (such as the Palm Pilot series) with its own memory, processor, and operating system might be the best choice for filling the role of the UI module. Therefore, the EDCU design provides a data port that will support connection to a PDA.

6.6 BSU Module

One of the main technologies available in the California ATMIS Testbed is laboratory access to traffic control and monitoring devices in the field which include traffic signals, loop detectors, ramp meters, and changeable message signs. Data access and control field devices in the Testbed is provided via an implementation of the CORBA operating over dedicated fiber links to traffic management centers in Orange County, California.

The BSU module should enable the integration of the GPS data collection system as another monitoring device available on the Testbed. To accomplish this, the BSU module must provide a remote interface to the data collection unit. Conceptually, this interface is simply software for the base station (e.g., the TMC that implements the unit's API by making remote calls to the unit over the communications network. This remote API can easily be encapsulated as a CORBA object and integrated seamlessly into the Testbed.

The main design concerns for the BSU module include the computing platform and the communications network.

6.6.1 Computing Platform

The BSU module runs on a PC workstations running Linux, but operations on multiple platforms are possible.

6.6.2 Communications Network

Since the selected COM module provides data feeds to an internet site, the BSU module has default access to this data.

6.7 Additional Specifications

6.7.1 Operating Environment and Packaging

The data collection unit is intended for use in personal and work vehicles and thus was designed to withstand shock typical of urban and suburban road conditions, temperatures ranging from -20 deg F to 140 deg F, and moderate exposure to moisture. The data collection units are packaged in enclosed, durable (and lockable) cases. Current external module connection ports (power, GPS, and CDPD) are conveniently located and clearly labeled. The unit was intended for discrete use inside of personal vehicles and was therefore sized to be conveniently stowed under vehicle seats, in the trunk, or in other discrete locations inside the vehicle.

6.7.2 Software Standardization

Software developed for the system has been implemented using high-level languages such as C, C++, perl, and Java.

6.7.3 Cost

The total cost of designing and manufacturing thirty EDCU units, including all required hardware and software for the DUC, GPS, LOG, and COM modules, was approximately $56,000, or an average of $1860 per unit (including R&D and tooling costs). Anticipated costs of additional units range from $1200 to $1400.

Cost of CDPD service is currently $49 per month for unlimited use. A variety of byte-based rates are available, as well as a sleeper mode which de-activates service but maintains the modem IP address and provides for immediate re-activation (at $1.50 per month). CDPD service is being phased out in 2004, to be replaced by GPRS (the data equivalent of GSM). Service providers are accommodating transfer of service and modems.

6.8 Summary

The overriding design goal was to produce a vehicle trajectory recording system which can serve in many different projects over several years. A design that met this goal of robustness and extensibility was achieved.

7 TRACERMAP: A GIS FOR THE TRACER SYSTEM

7.1 Overview

This section summarizes the initial development of the Geographic Information System (GIS) portion of the Tracer system, entitled TracerMap. The GIS is implemented using BBN Corporation's OpenMap Java library (BBNT Solutions, 2001). The current functionality of TracerMap is to read and display collected GPS data. This section outlines the work completed to achieve this basic functionality and some selected enhancements, then concludes with a discussion of how TracerMap could be taken forward.

7.2 Basic Operation of TracerMap

TracerMap can be used in two modes -- as a standalone Java application, and within a web browser as a Java applet. The two modes are virtually identical from the user's point of view, and aside from some startup differences, are identical under the hood as well. This section will refer generically to TracerMap functionality, unless there is a specific difference in the applet or the application version. Section 6.2.1 gives a brief overview of BBN Corp.'s OpenMap Java class library. Section 6.2.2 then gives a description of the TracerMap application based on the OpenMap library.

7.2.1 The OpenMap Java Library

OpenMap is an open source geographic information system (GIS) library implemented in Java. OpenMap conceptually organizes the map into layers. For example, one layer might be a solid, filled in polygon representing the shape of a county, while another layer might contain labeled points identifying locations of a particular activity-type within the county.

Each layer is a specific implementation of an appropriate Java class. For example the class openmap.layer.shape is used to display ESRI shape files. Other classes are tailored to display other kinds of geographical data. An application consists primarily of several Java classes each controlling the display of a single map layer. These classes are controlled by other specialized Java classes, in particular the MapBean class.

The OpenMap library was chosen for development primarily because the license is an open source type of license, essentially allowing free development and distribution of GIS code. More importantly, the library offers a high degree of flexibility in producing GIS-enhanced applications.

The Java language was attractive because of its relatively pure object oriented features, the large toolkit of communications classes available, and its ability to run on any platform and within a web browser. In short, using the OpenMap product enabled the creation of a GIS tailored to the exact needs of the project using commercial quality code. Finally, the fact that our code and the underlying library code are free greatly increases the likelihood that others will experiment with and perhaps improve our system.

Since the OpenMap library has an open source style license, our derivative code necessarily has similar licensing restrictions. First and foremost is the requirement that the code that we develop for the OpenMap platform shall also be released to the public under an open license. To satisfy the spirit of this requirement, this report documents the basic design decisions that underlie the first generation of TracerMap. The intent is to convey to future users of our code the reasons why we designed the system the way we did. By documenting our process and having the source code openly available, we hope that others will contribute to our development efforts, or use our code as the beginning of something bigger and better.

7.2.2 TracerMap as an Application and an Applet

The primary goal of the TracerMap GIS is to display the data collected from the global positioning system (GPS) antennae contained in the extensible data collection units (EDCUs). The UC Irvine ATMS Testbed already has a complete traffic network coded into Paramics, and so the ability to perform traffic engineering analyses was not a primary design goal. However, the TracerMap application was seen as playing a useful role in aggregating GPS traces, and linking together other data sources such as location and status of embedded inductive loop detectors.

In addition to running as an application, Java can also be run within a web browser as an applet, as long as the program follows certain strict rules pertaining to file access and communication with remote clients. The second design goal of the TracerMap package was to display travel over the internet via an applet. One application of this might be to display a day's travel information to actual travelers participating in a Tracer-based travel behavior study. The conversion of the initial development version of TracerMap to run as an applet was not trivial, but the conversion was a one-time operation. All future development on TracerMap should work equally well in both the application and applet environments.

7.3 Designing a GIS for the Tracer System

This section provides the rationale for the design choices made when programming TracerMap. The design incorporated both the existing and anticipated hardware capabilities of Tracer, as well as the various applications of the Tracer system. This material should be particularly interesting to anyone wishing to modify the TracerMap codebase for their own applications.

The basic hardware of the Tracer system consists of the mobile extensible data collection units (EDCUs), and server computers located in our labs. The servers have a secure connection to data gathered from Caltrans and other local traffic agencies, so the Tracer system can be seen as connecting actual travelers with the gigabytes of real-time traffic operations and control data that are available to traffic engineers.

7.3.1 Data Format

The foundation of the Tracer system consists of the EDCUs, and therefore much of the design of TracerMap caters to their capabilities. The EDCUs have real-time communication capability via their wireless data modems. They can also be used in the field as passive recording devices, with the data being uploaded to the server at the end of the survey. The units are extensible, as their name implies. While they currently collect very basic GPS records consisting of latitude, longitude, speed, and time, they could also record much more from the GPS antenna. Further, the future extensions might include recording other hardware input, such as gyroscope readings or emissions rates, or human input, such as notations about trip purposes.

Clearly the data format for communication with the EDCUs needs to be flexible and extensible. We chose to use extensible markup language (XML) to define the data. A document type definition (DTD) was composed to describe GPS data. DTDs were originally designed for documents, despite their widespread use in defining data in XML. This has given rise to the proposed XML Schema standard. When XML Schema is finalized, this DTD will be converted into a Schema. The current official DTD is located at http://www.its.uci.edu/~jmarca/DTD. The GPS namespace is officially defined as:

`<GPS:Trace xmlns:GPS="http://www.its.uci.edu/gpsxml/">.`

Although it looks like a standard URL, this namespace does not currently point to anything. The XML Namespace specification states that namespaces should be unique, and that they do not have to point to anything. A vocal opposition in the XML developer community feels that the namespace should point to something useful. In the future, we may place XML-GPS resources at the end of this namespace pointer.

The DTD allows for many variations in the complexity of the GPS data taken from the GPS antenna, from simple latitude and longitude, to the full description of all satellites in view. The XML-aware application (TracerMap) expecting data that conform to this DTD knows how to parse all of the elements defined in the DTD, and so it can extract the elements it wants and ignore the rest.

In order to change or extend the data that the EDCUs are sending to the outside world, the DTD must be modified, and the EDCUs must send data that conforms to the new DTD. Without going into the details, it is not necessary for the EDCUs to send all of the elements defined in the DTD. Many elements are optional. A valid conforming GPS:Trace “document” contains at least one GPS:Record. A valid GPS:Record at a minimum contains a GPS:Time element, and a

GPS:Position element. Any number of optional elements can be tacked onto these documents, and existing applications can still transmit valid documents as long as these minimal requirements are not changed. Therefore, future upgrades and enhancements can be made while still supporting legacy equipment and programs that adhere to the current GPS DTD.

With a small field unit such as the EDCU, two important design constraints are the memory available for storing data, and the bandwidth available for transmitting that data. Adding extra text to each GPS reading so as to encapsulate it within the XML document definition seems to fly in the face of limited memory and transmission bandwidth. However, if the collected GPS information cannot be understood by another application, then the data is useless. We designed the GPS:Trace DTD in order to standardize our GPS data messages, regardless of their originating hardware. We hope that by using and publishing this DTD, others might adopt it as their standard, or propose their own modifications and improvements.

However the current state of wireless transmission technology here in Orange County, California is such that the maximum transmission bandwidth one can reasonably expect is about 9.6 kbps. This is a very slow rate, and so we may for the short term transmit raw GPS data rather than XML-ified data, in order to make the most of the bandwidth available. The XML encoding will instead be performed by the server receiving the data from the EDCUs. As wireless bandwidth improves, the need for this short term fix will be eliminated. The TracerMap GIS expects data to conform to the GPS:Trace DTD. The program is not particular about where the data is located, whether being transmitted in real time or retrieved from a file, so long as it can be parsed and validated by the built-in XML parser. As a fall-back option, the current version of TracerMap will attempt to read data as if it were the raw output from an EDCU if the XML document validation fails. This was implemented in order to support the short term, low-bandwidth hack documented in the preceding paragraph. This will not be supported in future versions of TracerMap, since the output from the EDCU will most likely be different depending upon the application.

7.3.2 Data Exchange

Once GPS data is collected in the EDCU, it can either be transmitted immediately to the Tracer server via the cellular data modem, or saved onto a flash RAM memory card for later retrieval. The data are transmitted between applications as XML files that conform to the GPS:Trace DTD, as discussed above. The prototype EDCU does not have a working data modem, and so all testing was performed off-line. The files were uploaded to the server and saved, both as XML files, and as raw data.

In order to support the TracerMap program and to enable it to run as an applet, several server programs were written in Java to handle loading data into TracerMap. These programs run in the background independent of TracerMap.

Since they are running on the web-server computer, they have permission to read and write files on that computer. An applet also running on that server computer, in this case TracerMap, can send messages to these helper programs in order to read and write files.

These background Java servers were written for loading the EDCU data, and the shape files that define the roads and political boundaries of the map. While not strictly necessary for running TracerMap as a standalone application, these server programs must be present for TracerMap to be run as an applet, and so the necessary functionality was removed from TracerMap and placed permanently in the supporting programs. One advantage to removing the file-loading functions from TracerMap into helper programs is that these programs can change to reflect advances in related hardware and databases, without having to directly modify the TracerMap program.

7.3.3 The Role of XML-RPC in TracerMap

The XML-RPC (XML-Remote Procedure Call) protocol (UserLand Software, 2001) is used to enable the importation of GPS:Trace data into the TracerMap program, and to facilitate all message passing between concurrently running programs. The use of XML in XML-RPC should not be confused with the XML protocol used to store the GPS trace data. In XML-RPC, the XML part is used to facilitate the exchange of functional information between programs. Its use is transparent to the user. The user of an XML-RPC client-server system send and receive simple data structures, such as strings, numeric types, vectors of simple types (including vectors of vectors), and so on. For an in depth discussion of XML-RPC, there are some very useful books (McLaughlin, 2000) and websites (Kidd, 2001; Warner, 2001). This section serves only to give a flavor of how XML-RPC is used in TracerMap.

RPC stands for remote procedure call. In very simple terms, RPC describes the case when a program calls procedures within completely independent, concurrently running programs, usually running on another computer. The idea in some form has been around as long as the client-server model of computer networks, but many of the implementation details were tricky to work out. XML-RPC combines the well-defined textual representation possible in XML with the concept of remote procedure calls. The use of XML allows the rigid description of what constitutes a valid RPC message. The fact that XML is text based provides a great deal of transparency to the user, which further aids development.

XML-RPC allows the definition of several distinct data types, the exact members being dependent upon the particular XML-RPC implementation being used. These data types are wrapped up within XML. The XML document identifies itself as an XML-RPC message, and then presents the data types to the client and server in a well-formed manner.

In practice this works as follows. Say the TracerMap program wants to get a particular block of GPS:Trace data, called TodayTrace, from a known source, say MobileEDCU. The TracerMap program knows that MobileEDCU is a GPS data handler, and has a publicly registered a procedure called WriteXMLFile that accepts zero or more parameters and returns an XML character stream. This procedure has been registered with the XML-RPC server called serverX. The TracerMap program tells the XML-RPC server to go get the GPS data from the named procedure handler, and waits for the response. The calling code is as follows.

```
response=(TraceParse) serverX.execute(“MobileEDCU.WriteXMLFile”,params);
```

The XML-RPC server called serverX here gets the execute command, strips off the MobileEDCU part and uses that to identify the XML-RPC procedure handler being requested. It wraps up the command in the required XML containing stream, and passes along the request for the procedure being requested, or "execute(“WriteXML”,params)" to the handler MobileEDCU. This process is called marshalling the message.

The XML-RPC handler gets this execute message as a stream of XML text. The handler un-marshalls the stream of data (decodes the enclosing XML commands), and extracts the requested procedure and the passed parameters (which may consist of several different data types). Then it proceeds to execute WriteXML with the passed parameters "params".

The usual case is for the XML-RPC procedure handler to wrap up the resulting data in an XML stream and return that to the XML-RPC server. In other words, the reply data is marshalled using XML, and sent to back to the caller, where it is unmarshalled. The TracerMap program is then passed the contents of the response, which in this case is a chunk of XML encoded GPS position data, as requested, conforming to the GPS:Trace DTD.

While this example is rather complicated, it was included because it is exactly what is being done within the Tracer environment. And as complicated as it is, the messages being passed are simply XML text documents, wrapped up in another layer of XML to correctly move the data packets from computer to computer over the network. These text files are easy to pass around via http. This exchange from map program to XML-RPC server to XML-RPC handler and back can allow multiple, simultaneous GPS data handlers to be registered with the XML-RPC server. As long as the TracerMap program knows the names of the different data handlers and their particular calling commands, then each can be called in turn to build up a map containing all mobile EDCU devices.

7.4 TracerMap Functionality

The minimum functionality required of TracerMap was to display collected GPS traces on a map of Orange County. As mentioned in the preceding discussion, this functionality required communicating with various other parts of the Tracer system, both as an application and as an applet. In terms of the OpenMap Java library, displaying a GPS trace on top of a map of places and streets requires the use of different objects. This section describes the particulars of this implementation.

7.4.1 Using ESRI Shape Files

A widely used format for GIS data are those used by ESRI in its ArcGIS products. OpenMap can read in these files through its ShapeLayer class. We had available a map of Orange County, including its streets and highways, in ESRI format. This map was originally based on Census Tiger line files. These files were loaded successfully in preliminary versions of TracerMap using the ShapeLayer class.

Due to the way in which ESRI shape files are interpreted by the ShapeLayer class, this class cannot be run by an applet. Without going into too much detail, the implementation of ShapeLayer creates a temporary random access file, which violates the applet security model. Therefore, following the advice of the OpenMap help files, we revised TracerMap to use the LinkLayer class, in which layer information is obtained by querying an external link. The link is to a program that is run locally on the server, which means that it does have permission to create random access files. As far as the applet is concerned, it merely contacts an external program asking for shape data, and it gets back shape data, so the applet security model is happy. There were some further refinements that were necessary in order to get the LinkLayer to work with an applet. As these modifications are fairly technical and do not add to the understanding of how TracerMap works, the interested reader is referred to the well commented source code, which is freely available. The end result is that prior to running TracerMap, one must first fire up the ShapeServer auxiliary programs.

To simplify the codebase, each ESRI layer requires its own ShapeServer program to be running. To conform to the Java security model, these must be running with the same URL base as the calling applet, although a standalone application can query any URL. A unique port number is assigned to each server program. The TracerMap program is informed, via its startup properties file, the URL and port number of the ShapeServer program, as well as the properties file to examine for the colors and other attributes of that layer.

7.4.2 Plotting a GPS Trace

There are several classes that might be suitable for displaying GPS trace data. Preliminary implementations focused on the fact that an EDCU in the field would be actively adding points to the plot as the vehicle moved. The initial solution was therefore to create an active web socket connected to the EDCU device. The map would then query the socket for periodic updates. In practice, this proved to be too slow, since each point required too much computational overhead.

A much faster implementation was written by loading up the GPS points, connecting them with lines, and plotting the entire trace as a single object within the map. In order to allow for dynamic changes to the GPS data, access to the data was enabled through a server class, similar to the method used for the ESRI shape files. The user of TracerMap can specify at runtime the name of the file desired. This file is then passed to the GPS trace server, which replies with the necessary data. Currently this has only been tested with static files. When the production version EDCUs are received with working data modems, the program will be tested against a dynamic file. The file specified is parsed by the GPS trace server as a URL, so that one could specify a file located on a remote (mobile) server just as easily as one on the local file system.

After extensive testing of the GPS trace layer, it was noticed that occasionally there would be gaps in the collected GPS data, which showed up in the rendered map as rather large line segments. Some digging revealed that all of these apparent jumps occurred with zero speed recorded. In fact they were due to normal effects such as losing a satellite fix for a period of time, or turning off the EDCU, moving the unit, and then turning it back on again. In order to highlight these segments without breaking the visual linkage of the file, all rendered links with detectable position change at zero speed are drawn in a different color than the rest of the line segment. This color is user definable, but an example would be setting these gaps to be colored in pink, while the rest of the GPS trace is colored in blue. An operator would immediately see where the data might be suspect by noting the color differences.

7.5 Operational Enhancements

The initial goal of rendering GPS trace data on top of a recognizable map of Orange County was achieved, and has been tested to render properly multiple large GPS trace files simultaneously on the map. The user interface, while not a polished as a commercial application, is quite usable, relying on the Java Swing GUI library. However, we quickly developed operational needs for the TracerMap program that went beyond simply rendering maps and traces.

The first application for which TracerMap was used required matching several GPS trace plots against speed and volume data collected from embedded inductive loop detectors. It was a simple matter to render the loop detector layer using OpenMap's PointLayer class. Some small extensions were added to the class to enable the display of each loop's coordinates and other identifying information.

Next, the analysis required matching loops with the closest line segment in the GPS trace. Discussions with the programmers of OpenMap at BBN Corp. revealed that there was no existing way for one map layer to query another. Therefore, we implemented a simple mouse interface for pairing loops and trip segments. When a loop or a segment is selected with the mouse, a dialog pops up that allows the operator to dump the relevant data to a file. While a little crude, this allows the operator to eyeball the closest loops and links, and pair them off in a separate file for further analysis. A better solution will be implemented for this problem, but its execution has been postponed until the integration of Tracer with a database is completed, since all or most of the matching algorithms might be handled at the data source, rather than at the client-side interface.

Another operational enhancement was to write a user interface which allows the addition of new GPS trace layers to the map. To do this, the operator selects the layer menu, and then selects the palette icon from any GPS trace layer. The palette icon leads to a user interface which allows the operator to either replace the data in the existing layer with different data, or to create an entirely new layer from different data. The dialog takes as input the name of the new layer, and the location of the data. Currently the name is in the format of a local file, but the internal workings of the program treat that filename as if it were a URL. The next generation of this interface will add checkboxes which will indicate whether the file identifier is a local file, or a remote URL. While URLs can point to local files, the syntax is a bit different from simply typing the path to the file. Rather than simply requiring the odd URL format for a local file, the checkbox will give the operator the option to type a full URL or just the local file name and path.

7.6 Future Research

The next step in the development of TracerMap is to fully integrate the mapping functions with the dynamic data sources that are available. That means making sure that the mobile EDCUs connected via data modems can be queried at any time, as well as programming a database layer that can automatically generate queries to the ATMIS Testbed database.

In December 2000, BBN released an upgraded version of the OpenMap libraries. OpenMap has a large and active community of users and contributors, in addition to the engineers at BBN Corp. This translates into a fairly rapid cycle of product updates and improvements, which is another benefit of OpenMap's open source license. We are currently actively reviewing the enhancements to the layer classes. We will update our TracerMap extensions to these classes if necessary. We will take this update opportunity to review our own code, and rationalize the integration of layers within TracerMap, and the integration of TracerMap into the larger Tracer and ATMIS Testbed context.

The most important function that is missing is a way for layers interact. Specifically, TracerMap should be able to easily compare collected speed data with the data streaming in real time from other traffic sensors. There are several ways to accomplish this, including communicating entirely within the TracerMap program, and sending messages to external programs or databases to generate a comparison report and perhaps an evaluation map layer.

Another application is to embed the Java applet version of TracerMap into a full web page. The initial application is to use the applet to generate relevant context for an internet survey of travelers. The participants would be given an EDCU, and then would be asked to log in to the survey website each evening. The website would ask questions about that day's travel, including routing choices and awareness of traffic conditions. TracerMap would present the portion of that day's travel that is relevant to the current question, so as to prompt the respondent about the context of the question.

On the backend of the application, we plan to integrate the collection of data from the mobile EDCUs into a centralized database. While we still need to test the wireless data connections, it is anticipated that the average connection speed will be around 9.6 kbps. At this speed, multiple users requesting the same data from a single EDCU will hopelessly clog the communication pipeline. A more efficient implementation would be to centralize the data collection activities in a single server machine, which accepts multiple external requests for data, and services those requests by either grabbing historical data from a database, or else sending a single request for information to the mobile EDCUs that are equipped with CDPD modems.

7 SUMMARY AND FUTURE RESEARCH

This report has documented the design and construction of TRACER, an vehicle tracking system, comprising an extensible data collection unit (EDCU) and support software for tracking and post-processing. The design of the extensible data collection system was primarily motivated by the need for flexibility and portability given a wide range of application projects needing mobile GPS data. Previous PATH-supported research which developed the TRICEPS ATMS and the CARTESIUS decision support system identified the need for accurate speed and travel time data. This work also identified the growing need for path data in the development, calibration, and validation of simulation models as well as in the management and control of transportation systems. Together with research supported by University of California Transportation Center to develop the REACT! Internet-based travel survey software, the development of a GPS-based tracking device to provide basic probe data, and which could be integrated with REACT! for path surveys, held significant promise to improve the effectiveness of Intelligent Transportation Systems.

Starting with the data needs of existing research projects, and after reviewing the literature and various hardware vendors for other solutions, the TRACER system documented in this report was developed. The preliminary experiences with the system have been quite promising in both planned applications and in concurrent projects where the availability of the system facilitated the completion of or extended the definition of associated tasks. The presence of a robust, extensible GPS data collection tool has prompted many new project ideas and has led to collaborations with other researchers. The most promising aspect of the device is its wireless connection to the Internet, an aspect that may well prove invaluable for designing and testing future traveler information systems.

The rapid advancement of technology has created the opportunity for applying new, powerful tools to transportation engineering problems, but often the very speed of technological change hinders the adoption of these tools in a research environment. This paper documents the development of an extensible data collection unit (EDCU). The unit combines a standard GPS unit, a cellular data modem, and an embedded processor running the Linux operating system. Some preliminary uses and applications of the EDCU are presented as well.

The EDCU satisfies multiple functional requirements, due to the flexibility of its modular components and its full-powered operating system. The EDCU will serve the in-vehicle data collection needs of travel demand modelers and ITS researchers for the foreseeable future.

8 REFERENCES

@Track Communications (2001) "TrackWare FAQ," accessed on-line on 12 June 2001 at http://www.attrackcommuniations-cgi.com/trackware_faq.html.

Battelle (1997). *Global Positioning Systems for Personal Travel Surveys: Lexington Area Travel Data Collection Test,* Final Report, FHWA.

BBNT Solutions (2001). OpenMap tm , http://openmap.bbn.com/.

Benz, R.J. and Ogden M.A. (1996) "Development and Benefits of Computer-Aided Travel Time Data Collection," *Transportation Research Record 1551,* Transportation Research Board, Washington, D.C., pp. 1-7.

Bullock, D., Quiroga, C. and Kamath, N. (1996) "Data Collection and Reporting for Congestion Management Systems," In: *National Traffic Data Acquisition Conference (NATDAC '96) Proceedings, Volume I,* Report No. NM-NATDAC-96, Alliance for Transportation Research, Albuquerque, New Mexico, pp. 135-146.

Carter, M., *et. al.* (2000) "Metropolitan Model Deployment Initiative – San Antonio Evaluation Report," Report No. FHWA-OP-00-017, Science Applications International Corporation.

Casas, J. and Arce, C. (1999) "Trip Reporting in Household Travel Diaries: A Comparison to GPS-Collected Data," Presentation: 79th Annual Meeting of the Transportation Research Board, Washington, D.C.

Choi, K. and Chung Y-S. (2001) "Travel time estimation algorithm using GPS probe and loop detector data fusion," Presentation: 80th Annual Meeting of the Transportation Research Board, Washington, D.C.

Doherty, S.T., Noel, N., Lee-Gosselin, M., Sirois, C., Ueno, M. and Theberge, F. (2001) "Moving Beyond Observed Outcomes: Integrating Global Positioning Systems and Interactive Computer-Based Travel Behaviour Surveys," In: *Transportation Research Circular E-C026,* Proceedings of the Transportation Research Board Conference on Personal Travel: The Long and Short of It, Washington, D.C., pp. 449-466.

Draijer, G., Kalfs, N. and Perdok, J. (2000) "Global Positioning System as Data Collection Method for Travel Research," *Transportation Research Record 1719,* Transportation Research Board, Washington D.C., pp. 147-153.

El-Araby,K, Kates,R, Keller,H, and Huber,W (2001). "Trade-offs between robustness, accuracy, and resolution for floating car and loop detector data fusion", Proceedings: 8th World Congress on ITS, Sydney.

Gallagher, J. (1996) "Travel Time Data Collection Using GPS," In: *National Traffic Data Acquisition Conference (NATDAC '96) Proceedings, Volume I,* Report No. NM-NATDAC-96, Alliance for Transportation Research, Albuquerque, New Mexico, pp. 147-161.

Goodchild, M.F. (1999) "GIS and Transportation: Status and Challenges," Keynote Address, International Workshop on GIS-T and ITS, Chinese University of Hong Kong.

Hellinga, B. and Fu, L. (1999) "Assessing Expected Accuracy of Probe Vehicle Travel Time Reports," *Journal of Transportation Engineering,* vol. 125, no. 6, pp. 524-530.

Higgins, L., Vaidya, N. and Turnbull, K.F. (1995) "Assessment of METROlift Automatic Vehicle Location System," Report No. TTI/ITS RCE-95/02, Texas A&M ITS Research Center of Excellence, Texas Transportation Institute, College Station, Texas.

Jiang, Y. and Li, S. (2001) "Measuring and Analyzing Vehicle Position and Speed Data at Work Zones Using Global Positioning System

Joshi,RR (2001). "A new approach to map matching for in-vehicle navigation systems: the rotational variation metric", IEEE 4th International Conference on Intelligent Transportation Systems, pp.33-38.

Kidd, E. (2001). XML-RPC HOWTO, http://xmlrpc-c.sourceforge.net/xmlrpc-howto/xmlrpc-howto.html, eric.kidd@pobox.com.

Li, Y and McDonald, M. (2002). "Link travel time estimation using single GPS equipped probe vehicle", Proceedings: The IEEE 5th International Conference on Intelligent Transportation Systems, 932-937.

McLaughlin, B. (2000). Java and XML, O'Reilly & Associates, Inc., 101 Morris St., Sebastopol, CA 95472.

Michler, E. (2001). "Statistical analysis of floating car data systems", 8th World Congress on ITS Proceedings, Sydney.

Mid-Ohio Regional Planning Commission (MORPC) (1999) "ITS Integration Strategy for Central Ohio", http://www.morpc.org/trans/its/itsplan.htm (6/12/01).

Moore, J.E., Cho, S., Basu, A. and Mezger, D.B. (2001) "Use of Los Angeles Freeway Service Patrol Vehicles as Probe Vehicles," Report No. UCB-ITS-PRR-2001-5, California PATH Program, Institute of Transportation Studies, University of California, Berkeley.

Murakami, E. and Wagner, D.P. (1999) "Can Using Global Positioning System (GPS) Improve Trip Reporting?" *Transportation Research Part C,* vol. 7, pp. 149-165

Pyo,J-S, Shin,D-H, and Sung,T-K (2001). "Development of a map matching method using the multiple hypothesis technique", Proceedings: The IEEE 4th International Conference on Intelligent Transportation Systems, pp. 23--27.

Quiroga, C.A. and Bullock, D. (1998a) "A Data Model for Managing GPS Travel Time Data," In: *Applications of Advanced Technologies in Transportation: Proceedings of the 5th International Conference* (eds: Chris T. Hendrickson and Stephen G. Ritchie), American Society of Civil Engineers, pp. 403-410.

Quiroga, C.A. and Bullock, D. (1998b) "Travel Time Studies With Global Positioning and Geographic Information Systems: An Integrated Methodology," *Transportation Research Part C,* vol. 6, pp. 101-127.

Schofer, J.L., Koppelman, F.S., Webster, R.G., Berka, S. and Peng, T-S.(1996) "Field Test of the Effectiveness of *ADVANCE* Dynamic Route Guidance on a Suburban Arterial Street Network: Evaluation Report," *ADVANCE* Project Document #8463.ADV.01, Northwestern University Transportation Center.

Sen, A., Thakuriah, P., Zhu, X. and Karr, A.F. (1997) "Frequency of Probe Reports and Variance of Travel Time Estimates," *Journal of Transportation Engineering,* vol. 123, no. 4, pp. 290-297.

Shadewald, J.K. (2000) "GIS Based Model Interfacing: Incorporating Existing Software and New Techniques into a Streamlined Interface Package," In: *Mid-Continent Transportation Symposium Proceedings,* Iowa State University, Ames, IA, pp. 169-173.

Sutton, J., O'Packi, P. and Harris, B. (1994) "Conceptual Design of a GIS-T Congestion Management System," *URISA 1994 Annual Conference Proceedings,* Urban and Regional Information Systems Association, Washington, D.C., pp. 349-361.

Turnbull, K.F., Poe, C.M. and Brydia, R.E. (1998) "Rural and Small Community Transit AVL Research," Presentation: 1998 Rural ITS Conference, Pennsylvania State University.

Turner, S.M. (1996) "Advanced Techniques for Travel Time Data Collection," *Transportation Research Record 1551,* Transportation Research Board, Washington, D.C., pp. 51-58.

Turner, S.M., Eisele, W.L., Benz, R.J., and Holdener, D.J. (1998), *Travel Time Data Collection Handbook*, Report No. FHWA-PL-98-035, Texas Transportation Institute.

UserLand Software (2001). XML-RPC Home Page, http://www.xmlrpc.com/.

Van Aerde, M., Hellinga, B., Yu, L. and Rakha, H. (1993) "Vehicle Probes as Real-Time ATMS Sources of Dynamic O-D and Travel Time Data," *Large Urban Systems – Proceedings of the ATMS Conference,* St. Petersburg, FL, pp. 207-230.

Warner, D. (2001). XML-RPC: It Works Both Ways, http://www.oreillynet.com/pub/a/python/2001/01/17/xmlrpcserver.html, davewarner98@yahoo.com.

WebTech Wireless (2001) "Experience the Power of Location Services in Your Fleet," accessed on-line at http://www.webtechwireless.com/fleets.htm on 12 June 2001.

Wolf, J. Guensler, R. and Bachman, W. (2001) "Elimination of the Travel Diary: An Experiement to Derive Trip Purpose from GPS Travel Data," Presentation: 80th Annual Research Board, Washington, D.C.

Wolf, J., Hallmark, S., Oliveira, M., Guensler, R. and Sarasua, W. (1999) "Accuracy Issues with Route Choice Data Collection by Using Global Positioning System," *Transportation Research Record 1660,* Transportation Research Board, pp. 66-74.

Yalamanchili, L., Pendyala, R.M., Prabaharan, N. and Chakravarthy, P. (1999) "Analysis of Global Positioning System-Based Data Collection Methods for Capturing Multistop Trip-Chaining Behavior," *Transportation Research Record 1660,* Transportation Research Board, pp. 58-65.

Zewang,C, Yongrong,S, and Xing, Y (2002). "Development of an algorithm for car navigation system based on Dempster-Shafer evidence reasoning", Proceedings: The IEEE 5th International Conference on Intelligent Transportation Systems, pp.534-537.

APPENDIX

A Preliminary GPS Data Document Type Definition

This appendix lists the GPS data transmission DTD. Note that the element GPS:Trace is composed of several GPS:Record elements. The most recent version of this DTD is located at: http://www.its.uci.edu/ jmarca/DTD. Finally, note that by convention the GPS namespace is defined as:

<GPS:Trace xmlns:GPS="http://www.its.uci.edu/gpsxml/">.

This URL does not point to anything, which is in accordance with the definition of a namespace, which is a confusing point of the XML namespace definition causing much angst amongst the developers.

```
<!-- ========================================================= -->
<!-- draft GPS data DTD -->
<!-- ========================================================= -->
<!-- GPS Trace Data -->
<!-- Sequence of gps points from a single gps recorder -->
<!ELEMENT GPS:Trace (GPS:Record)+ >
<!-- xmlns namespace of GPS definitions id Unique identifier -->
<!ATTLIST GPS:Trace xmlns:GPS CDATA #REQUIRED id ID #IMPLIED >
<!-- ========================================================= -->
<!-- Single GPS Record -->
<!-- A single recorded gps point -->
<!ELEMENT GPS:Record (GPS:Time, GPS:Pos, GPS:Qual?, GPS:Sat?) >
<!-- ========================================================= -->
<!-- Time of measurement -->
<!ELEMENT GPS:Time (GPS:UTC, (GPS:yr, GPS:mon, GPS:day)?, >
<!-- (GPS:hr, GPS:min, GPS:sec)?) >
<!-- Universal Time Code -->
<!-- format mm/dd/yy hh:mm:ss -->
<!ELEMENT GPS:UTC (#PCDATA) >
<!-- year -->
<!ELEMENT GPS:yr (#PCDATA) >
<!-- month -->
<!ELEMENT GPS:mon (#PCDATA) >
<!-- day -->
<!ELEMENT GPS:day (#PCDATA) >
<!-- hour -->
<!ELEMENT GPS:hr (#PCDATA) >
<!-- minute -->
<!ELEMENT GPS:min (#PCDATA) >
<!-- second -->
<!ELEMENT GPS:sec (#PCDATA) >
<!-- ========================================================= -->
<!-- Position data -->
<!-- at least lat,lon. check quality for fix or not -->
<!ELEMENT GPS:Pos (GPS:Lat, GPS:Lon, GPS:Alt?, GPS:Speed?, GPS: Track?) >
<!-- Latitude -->
<!ELEMENT GPS:Lat (#PCDATA) >
<!-- Longitude -->
<!ELEMENT GPS:Lon (#PCDATA) >
<!-- Altitude -->
<!ELEMENT GPS:Alt (#PCDATA) >
<!-- Speed from GPS unit, knots -->
<!ELEMENT GPS:Speed (#PCDATA) >
<!-- Track made good, degrees -->
<!-- Need to examine gpsd code. Could be course over ground, NMEA$GPRMC(8) -->
<!ELEMENT GPS:Track (#PCDATA) >
<!-- ========================================================= -->
<!-- Solution quality -->
<!-- Quality varies with satellite fix, atmospheric conditions -->
<!ELEMENT GPS:Qual (GPS:pdop?, GPS:hdop?, GPS:vdop?) >
<!-- Status 0=no fix, 1=non-differential gps fix, 2=differential gps fix >
```

```
<!-- Mode 1=no fix, 2=2D fix, 3=3D fix -->
<!ATTLIST GPS:Qual Status (0|1|2) #IMPLIED Mode (1|2|3) #IMPLIED >
<!-- Position dilution of precision -->
<!-- value range from 0.5 to 99.9 -->
<!ELEMENT GPS:pdop (#PCDATA) >
<!-- Horizontal dilution of precision -->
<!-- value range from 0.5 to 99.9 -->
<!ELEMENT GPS:hdop (#PCDATA) >
<!-- Vertical dilution of precision -->
<!-- value range from 0.5 to 99.9 -->
<!ELEMENT GPS:vdop (#PCDATA) >
<!-- ======================================================== -->
<!-- Satellite information -->
<!-- Information on the satellites used to determine the position solution -->
<!ELEMENT GPS:Sat (GPS:SatSol?, GPS:inview, GPS:SatData*) >
<!-- Satellites used in solution -->
<!-- should be less than inview -->
<!ELEMENT GPS:SatSol (#PCDATA) >
<!-- number of satellites in view -->
<!-- zero to 12 satellites possible -->
<!ELEMENT GPS:inview (#PCDATA) >
<!-- ======================================================== -->
<!-- Satellite position data -->
<!-- At most twelve satellites -->
<!ELEMENT GPS:SatData (GPS:elev, GPS:azi, GPS:ss?) >
<!-- PRN Satellite PRN number, 1 to 32 -->
<!ATTLIST GPS:SatData PRN (1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18| >
<! 19|20|21|22|23|24|25|26|27|28|29|30|31|32) #REQUIRED >
<!-- Satellite elevation -->
<!-- 00 to 90 degrees -->
<!ELEMENT GPS:elev (#PCDATA) >
<!-- Satellite azimuth -->
<!-- 000 to 359 degrees -->
<!ELEMENT GPS:azi (#PCDATA) >
<!-- Satellite signal to noise ratio (signal strength) -->
<!-- 00 to 99 dB when tracking, else null -->
<!ELEMENT GPS:ss (#PCDATA) >
<!-- ======================================================== -->
<!-- end draft GPS data DTD -->
<!-- ======================================================== -->
```