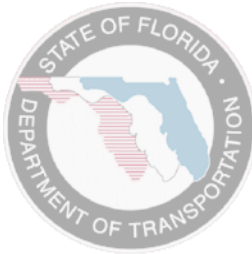


Smart Phone Application to Influence Travel Behavior (TRAC-IT Phase 3)

**FINAL REPORT
(FDOT BD 549 WO 35)**

Prepared for

**Michael Wright
Florida Department of Transportation
605 Suwannee Street, MS 30
Tallahassee FL 32399**



Prepared by

**National Center for Transit Research (NCTR) at the
Center for Urban Transportation Research (CUTR)
University of South Florida
4202 E. Fowler Ave. CUT 100
Tampa, Florida, 33620-5375**



March 2008

Disclaimer

The opinions, findings, and conclusions expressed in this publication are those of the author(s) who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the views or policies of the Florida Department of Transportation or the Research and Innovative Technologies Administration. This report does not constitute a standard, specification, or regulation.

The report is prepared in cooperation with the State of Florida Department of Transportation and the U.S. Department of Transportation.

Patents Pending

Multiple Patents Pending on TRAC-IT System and Components, USF 2008

1. Report No. BD-549-35	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle <i>Smart Phone Application to Influence Travel Behavior (TRAC-IT Phase 3)</i>		5. Report Date March 2008	
		6. Performing Organization Code	
7. Author(s) <i>Winters, Philip L., Barbeau, Sean J., and Georggi, Nevine L.</i>		8. Performing Organization Report No.	
9. Performing Organization Name and Address <i>National Center for Transit Research, Center for Urban Transportation Research, University of South Florida 4202 E. Fowler Avenue, CUT100 Tampa, FL 33620-5375</i>		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No. <i>BD 549 WO 35</i>	
12. Sponsoring Agency Name and Address <i>Research and Innovative Technologies Administration U.S. Department of Transportation, Washington, D.C. 20690 Florida Department of Transportation 605 Suwannee Street, MS 26, Tallahassee, FL 32399</i>		13. Type of Report and Period Covered Final	
		14. Sponsoring Agency Code	
15. Supplementary Notes <i>FDOT Project Manager: Michael Wright, CAP Program Manager. 605 Suwannee Street, MS 30, Tallahassee FL 32399</i>			
16. Abstract This three-phase research study focuses on using innovative technology to better understand and pattern household travel behavior for the purposes of educating, promoting, and encouraging households to utilize other alternatives to driving alone. The scope of Phase 1 called for preliminary development and testing of a portable unit consisting of a personal digital assistant (PDA), a global positioning system (GPS) device, and a wireless card, all-in-one unit nicknamed "TRAC-IT." Phase 2 investigated the impact of the travel feedback advisory system on household travel behavior. The objective of Phase 3 was to determine the capabilities of GPS-enabled mobile phones in tracking person movements across modes (car, bike, bus, etc.) and over extended time periods (e.g., weekly versus daily). A basic requirement of the system was that the mobile device selected had to be a commercially available, low cost, off-the-shelf, and a widely used device. The device also had to be equipped with embedded GPS capabilities for highly accurate position data, able to communicate this information wirelessly back to a server, and able to receive user input allowing the manual entry of survey data by participants that cannot be extracted from GPS data. Wireless communication features of the mobile phone were envisioned to offer the user with real-time information that could influence their current travel behavior. There is concern that sampling cell phone users would miss certain segments of the population that do not own mobile phones. However, an equal concern is the growing number of individuals that own only mobile phones and do not own landlines, which can potentially affect all surveys that are conducted using random calls to households with wired telephones. More research into the implications of using mobile devices as a survey tool is required. The use of GPS-enabled mobile phones and an application such as TRAC-IT presents a unique opportunity to collect high-resolution individual travel behavior data that are instantly transferred to a server for analysis by transportation professionals. As more "indoor GPS" (high-sensitivity) chips become available in the mobile phone market, it is expected that GPS data from mobile phones that are more accurate will be collected and these should be able to obtain location fixes where current phones cannot. Based on the continuous research of new innovative approaches to travel data collection using location-based approaches, the TRAC-IT research team concluded that GPS-enabled mobile phones will continue to be a vital tool in travel behavior data collection.			
17. Key Word <i>travel behavior, survey, global positioning systems, cellular phones, travel feedback advice, and travel blending</i>		18. Distribution Statement <i>Available to the public through the National Technical Information Service (NTIS), 5285 Port Royal Road, Springfield, VA 22161, (703) 487-4650, http://www.ntis.gov/, and through the NCTR web site at http://www.nctr.usf.edu/.</i>	
19. Security Classif. (of this report) <i>unclassified</i>	20. Security Classif. (of this page) <i>unclassified</i>	21. No. of Pages <i>140</i>	22. Price

Acknowledgements

This report is prepared by the National Center for Transit Research through the sponsorship of the Florida Department of Transportation (FDOT) and the U.S. Department of Transportation.

FDOT Project Manager:

Michael Wright, FDOT Public Transit Office

Project Team:

Principal Investigator: Philip L. Winters, TDM Program Director, CUTR

Co-Principal Investigators:

Sean Barbeau, Research Associate, CUTR

Nevine Labib Georggi, Research Associate, CUTR

Rafael Perez, PhD, College of Engineering, USF

Miguel Labrador, PhD, College of Engineering, USF

Computer Science and Engineering Students Team:

David Aguilar

Paola Gonzalez

Narin Persad-Maharaj

Samuel Rivera

Jeremy Weinstein

Tiffany Burrell

Dmitry Belov

Alfredo Perez

Diana Arteaga

Milena Sarmiento

The research team thanks the Sprint-Nextel Application Developer Program for their cooperation and the donation of phones and cellular service that were tested as part of this research project.

Executive Summary

This report documents Phase 3 of the “TRAC-IT” research study. The research focuses on using innovative technology to better understand and pattern household travel behavior for the purposes of educating, promoting, and encouraging households to utilize other alternatives to driving in general and to driving alone in particular. This phase of the research effort developed an application that collects household travel data using a smart cell phone. This final report is the third in a series to describe a joint collaboration among the Transportation Demand Management (TDM) Program, at the Center for Urban Transportation Research (CUTR), University of South Florida (USF), and the Computer Science Engineering (CSE) Department at the College of Engineering at USF.

The scope of Phase 1 called for preliminary development and testing of a portable unit consisting of a personal digital assistant (PDA), a global positioning system (GPS) device, and a wireless card - an all-in-one unit nicknamed “TRAC-IT.” With innovations emerging daily in this field of technology, a GPS-enabled cellular phone was briefly investigated as a possible alternative for the PDA TRAC-IT unit and showed promise. The TRAC-IT system from that phase was able to successfully upload the data to the server and provide suggestions for more efficient transportation options based on the household travel behavior. The basic feedback advisory system (also referred to as the expert system) was designed to test the ability of collecting and using the data using TRAC-IT. The preliminary expert system then generated tailored feedback for each user based on his/her household travel behavior over the entire test period. Phase 2 refined the expert system capabilities making it more comprehensive in analyzing the data collected and more sensitive to different household travel scenarios. Phase 2 also investigated the impact of that enhanced travel feedback advisory system on household travel behavior using a before and after survey analysis approach. The “before” represented initial household travel behavior for both control and experiment groups. The “after” represented survey of households in experiment group after suggestions on better travel options were presented to them. Phase 3 in this research effort was dedicated to the design, development, and testing of the GPS-enabled cellular phone as a TRAC-IT unit. Although Phase 2 and 3 were conducted simultaneously, the Phase 2 field testing portion depended on the effectiveness of TRAC-IT cell-phone software development and completion in order to test the application in measuring household travel behavior before and after feedback advice was disseminated.

This three-phase study builds on previous research conducted by CUTR for the Florida Department of Transportation (FDOT) Research Center Office, “Reducing Vehicle Trips and Vehicle Miles of Travel through Customized Travel Options” (1). The study demonstrated that the travel habits of households that were provided

with personalized advice did reduce vehicles miles of travel relative to the control group. The project offered a practical exercise that led households to re-appraise their needs and rationale for travel. Specific suggestions aimed at use of public transit service, bike paths, trip chaining, ridesharing, and e-commerce options were provided based on specific travel patterns observed in the activity diaries. A weeklong travel diary formed the basis of the personalized advice provided.

Since that CUTR/FDOT project, technology has evolved rapidly to provide a promising method to decrease the costs of collecting and analyzing travel data. At the conception of Phase 1 in 2004, the potential then existed to use PDAs and GPS add-on modules rather than paper diaries for easier and more accurate tracking of person movements, not simply vehicle movements. The GPS unit provided a means of tracking time, route, and speed, while the PDA provided a means of recording items such as mode, occupancy, and trip purpose.

Since Phase 1 was completed, mobile phones have evolved into modern mobile computers and boast many new capabilities such as text and picture messaging, internet access, and GPS. The ubiquity and low-cost of mobile phones make the possibility of using mobile phones as the primary data collection device for TRAC-IT very attractive. According to recent market research by ABI Research, the world population of GPS-enabled, location-aware services subscribers will grow from 12 million in 2006 to a projected 315 million by 2011, and North American growth is projected from 500,000 users in 2006 to 20 million users in 2011. Additionally, in 2007 global mobile phone use surpassed 3.25 billion subscribers, equivalent to half the world's population. This trend in technology sets the stage for revolutionary advances in intelligent transportation systems, especially in the area of large scale data collection and analysis of personalized travel behavior.

The objective of Phase 3 was to determine the capabilities of GPS-enabled mobile phones in tracking person movements across modes (car, bike, bus, etc.) and over extended time periods (e.g., weekly versus daily). The research methodology was to develop a mobile application that would collect travel survey data with minimal interaction from survey participants. A basic requirement of the system was that the mobile device selected had to be a commercially available, low cost, off-the-shelf, and widely used. The device also had to be equipped with embedded GPS capabilities for highly accurate position data, be able to communicate this information wirelessly back to a server, and be able to receive user input allowing the manual entry of survey data that cannot be extracted from GPS data. It was envisioned that real-time information that could influence the traveler's current travel behavior would be instantly provided via the wireless communication abilities of the mobile phone.

There are two main methods of utilizing mobile phones to obtain travel information. The first method uses anonymous cellular signaling data gathered at cell towers to estimate travel time on nearby highways. Travel time and

estimated travel speed on a specific road segment are the only items of information available from these types of systems, which may be sufficient for monitoring real-time traffic conditions. The second method, used by TRAC-IT, is the use of an actual positioning technology to gather position data for many mobile phones, with each having its position calculated individually by the phone or network using GPS, network triangulation, or a similar positioning technology. The development and deployment of highly accurate positioning technologies for cell phones in the United States has been driven by the Federal Communications Commission's (FCC) E911 mandate that requires cellular carriers to be able to locate emergency callers when 911 is dialed from a mobile phone.

TRAC-IT focused on this second method because it can yield extremely precise position data within 3-5 meters of the true position for each time point that is captured with a supported frequency of up to one fix per second. Other information, such as speed and heading, is also captured with each GPS data point; therefore, travel speeds and estimated travel times can also be extracted from these data (although values would have to be aggregated from multiple individual GPS data points to accurately reflect average speed for all traffic instead of a single vehicle's speed). Speed data derived from GPS have been shown to be accurate within .2 meters/sec in past studies. This method yields OD-type data as well as travel path information for each individual traveler and is a modern version of the traditional trip diary. Research such as TRAC-IT is still active using this method of data collection, and no commercial company has yet been established to collect travel behavior data using this method to the research team's knowledge. It is important that transportation agencies understand the difference between the two methods of data collection, as they yield very different types of data to be used for different purposes.

The research performed for this phase of TRAC-IT focused on the development of a software-based mobile application that could be installed on commercially-available GPS-enabled mobile phones. In addition, TRAC-IT software for a server was developed to receive and store data collected by the mobile application. The concept of the research methodology for this study was based on the findings of Phase 1 that successfully tested the TRAC-IT prototype on a PDA. Advanced capabilities of mobile devices guided the futuristic vision that the research team developed to utilize these capabilities in next generation travel surveys, real-time traveler information dissemination, and travel blending techniques.

The use of GPS-enabled mobile phones and an application such as TRAC-IT presents a unique opportunity for departments of transportation to collect high-resolution travel behavior data for individuals. These data represent an improvement over traditional OD data, since GPS data accurately represent the path of travel, and additional user input can be collected through the survey in the user interface. These data are instantly transferred to a server for analysis by transportation professionals. Additionally, field equipment costs are eliminated

since TRAC-IT is a software application that is deployable to any privately-owned, GPS-enabled mobile phone that supports the popular J2ME platform and the standardized method of accessing location data, JSR179 Location API. As more “indoor GPS” or high-sensitivity GPS chips become available in the mobile phone market in late 2008 and early 2009, DOTs can expect to collect GPS data from mobile phones that are more accurate and should be able to obtain location fixes where present phones cannot. Due to these reasons, the TRAC-IT research team believes GPS-enabled mobile phones will be an important resource for travel behavior data collection in the future.

As a cost effective tool, TRAC-IT provides real-time access to collected data for analysis. These data are instantly transferred to a server for analysis by transportation professionals, bypassing the time and cost of labor intensive data entry. Additionally, field equipment costs are eliminated since TRAC-IT is a software application that is deployable to any privately owned GPS-enabled mobile phone that supports the popular J2ME platform and the standardized method of accessing location data, JSR179 Location API. As more “indoor GPS,” or high-sensitivity chips become available in the mobile phone market in late 2008, DOTs can expect to collect more accurate GPS data and location fixes from improved and more advanced mobile phones. Due to all of these reasons, the TRAC-IT research team believes GPS-enabled mobile phones will be an important resource for travel behavior data collection in the near future.

As of early 2008, TRAC-IT can be deployed only to customers on Sprint-Nextel’s CDMA and iDEN networks. Currently, this does provide some limitations to the statistical validity of using TRAC-IT to sample the general population’s travel behavior. However, it appears that other wireless providers will be opening up their networks in the near future (end of 2008 to early 2009), which would allow TRAC-IT to be deployed to nearly everyone who owns a mobile device. Since there are more than 250 million U.S. wireless subscribers, this is a significant population to survey. Additionally, the recent FCC auction of the 700MHz spectrum to be used for future commercial services will be held to the condition of “open devices and open applications,” thus further opening up wireless carriers to application such as TRAC-IT.

There is concern that sampling cell phone users would miss certain segments of the population that do not own mobile phones. However, an equal concern is the growing number of individuals that own only mobile phones and do not own landlines, which can potentially affect all surveys that are conducted using random calls to households with wired telephones. Surprising to some, cell-phone-only households tend to have lower incomes. More research would have to be performed to analyze the implications of using data collected via a cell-phone-based application such as TRAC-IT.

Because of the many issues involved in the deployment of an location-aware application, it is recommended that U.S. DOT, along with FDOT and others DOTs, engage wireless providers in a single dialog to discuss how a large-scale test of TRAC-IT (i.e., over 200 users) could be realized. As part of this dialog, it should become apparent whether there are any policy-based or logistical issues that would prevent a large-scale deployment of TRAC-IT, including any access restrictions by cellular carriers. For wireless providers that do not currently support the TRAC-IT application, U.S. DOT should request an estimated timeline for technology deployment that would enable TRAC-IT to be used on those particular wireless providers. If there are no access restrictions or logistical barriers to deployment of TRAC-IT on a wireless provider, testing of simultaneous mobile phones should be gradual to detect and manage any unforeseen scalability issues with the system. While the design of the system has scalability in mind, there can be unknowns in the real-world implementation of technology.

Due to the many different mobile phone devices on the market, it is also recommended that DOTs create a process to test and certify specific devices with the TRAC-IT application before TRAC-IT is deployed to that particular device. This is especially important when devices are tested across multiple wireless providers, since significant differences can exist in J2ME applications across providers. Also, since the end-user device is privately owned, testing should ensure that no personal mobile phone functions will be compromised by TRAC-IT. As J2ME for mobile devices matures during the next year, including the release of MIDP 3.0 and OSGi, future development and testing for new devices should become easier, and TRAC-IT should be less likely to interfere with normal cell phone operations.

Since participants indicated that the biggest challenge they faced was remembering to start a trip via the TRAC-IT user interface, the passive recording feature of TRAC-IT should be investigated further. In this mode, the phone would always be sending location data to the TRAC-IT server unless the user selected to turn this feature off via the TRAC-IT user interface. This feature would not require the user to always remember to start a trip, since it would be recording even if no action were taken by the user. However, passive tracking could negatively impact cell phone performance, including battery life. Therefore, further analysis of the State Machine and Critical Point functionality should take place to determine if these "smart" resource management techniques can offset any negative impact on cell phone functionality when passive tracking is turned on. There are also privacy concerns when using passive tracking, since the user does not actively "opt in" for each individual trip that is recorded. Instead, the user must remember to "opt out" by using the TRAC-IT interface to turn off the passive tracking feature if the user does not want a trip to be recorded. Certain TRAC-IT features such as the establishment of user-defined "sensitive areas" or "sensitive times" where all trip recording is automatically "black-out" for particular places and times should be investigated to enhance privacy of passive tracking.

The unique methodology of TRAC-IT enables departments of transportation to explore new types of incentives in coordination with wireless service providers. For example, DOTs could potentially arrange with the wireless carrier to provide free unlimited data plans during the survey period to any participants willing to install and use TRAC-IT on their phone. The participant then would have access to new services on their mobile phone in addition to TRAC-IT, including email, internet browsing, online search, etc. Once the participant is exposed to these services, they might be much more likely to continue to pay for the service after the TRAC-IT survey period ends, thus providing additional revenue to the wireless service provider. DOTs would benefit from reduced deployment costs through cost reductions in unlimited data plans, and the wireless carrier would benefit both by the DOT paying some cost for unlimited data plans during the survey period and by the participant potentially paying full cost for the plan after the survey period ends. Similar arrangements may be reachable by providing free participant access to real-time GPS navigation applications or other software for mobile phones.

The concept of path prediction and traffic incident alerts described in this report and the TRAC-IT Phase 2 report could also be used as an incentive by itself. For example, if participants use the TRAC-IT application to record their behavior, they would automatically receive free, personalized traffic alerts based on their own travel behavior as well as their real-time location. This service could be so valuable to mobile phone users that they overcome any reservations regarding privacy and are willing to allow their travel behavior to be tracked, at least using the passive version of TRAC-IT, and used for DOT analysis.

Table of Contents

Chapter 1. Introduction	1
1.1 Project Background	1
1.2 Study Purpose and Objective	2
1.3 Research Methodology	2
1.4 Report Organization.....	3
Chapter 2. Technology Assessment.....	4
2.1 Location-Based Services Technology for Cellular Networks	4
2.1.1 <i>Device-Based Positioning Methods</i>	7
2.1.2 <i>Network-Based Methods</i>	8
2.1.3 <i>Hybrid Positioning Methods</i>	9
2.2 Software Developer Methods to Obtain Position Data.....	10
2.2.1 <i>Handset-initiated Location Requests</i>	11
Overview of J2ME Location API	12
Google Android Platform	14
JSR179 Location API 1.0	14
2.2.2 <i>Network-Initiated Location Requests</i>	17
2.2.3 <i>Current Capabilities of Cellular Providers and Cellular Devices</i>	18
Sprint-Nextel iDEN Network	22
Sprint-Nextel CDMA Network.....	26
AT&T	29
Alltel.....	30
Verizon Wireless.....	30
T-Mobile	31
Chapter 3. TRAC-IT System Design	32
3.1 Positioning Technologies	32
3.2 Form-factor.....	33
3.3 Network-Initiated vs. Handset Initiated Location Requests	33
3.4 Portability of the TRAC-IT Software.....	34
3.5 Coexistence of TRAC-IT with Cell Phone Operations	35
3.5.1 <i>Background Application Execution</i>	35
3.5.2 <i>Voice Calls</i>	36
3.5.3 <i>Data Plans</i>	36
3.5.4 <i>Energy Conservation</i>	36
3.6 Interoperability and Forwards-Compatibility of the TRAC-IT System	37
3.7 Final Selection of TRAC-IT System Components	37
Chapter 4. TRAC-IT System Architecture	39
4.1 Overview of System Architecture	39
4.2 TRAC-IT Mobile Phone Modules	44
4.2.1 <i>The User Interface</i>	45
A Sample Walkthrough of TRAC-IT User Interface.....	47
Trip Feedback	50
Route Prediction & Incident Notification	51
4.2.2 <i>The Timer</i>	51
4.2.3 <i>The LocationListener</i>	52
4.2.4 <i>The Communicator</i>	54
4.2.5 <i>On-phone Storage</i>	58
4.2.6 <i>Critical Point Detection</i>	58
4.3 The Cellular Carrier Network.....	60

4.4	The Internet	61
4.5	Server-side Modules of TRAC-IT	64
4.5.1	<i>Remote Procedure Calls - Managing the Distributed TRAC-IT Application</i>	64
	Interaction with Database Server	66
	Session and Logging Procedures	68
	Processing Location Data	68
	Defining Trip Segments	69
4.5.2	<i>Critical Point Detection</i>	70
4.5.3	<i>Automatic Mode Detection</i>	70
4.5.4	<i>Travel Advisory Feedback System</i>	72
4.5.5	<i>Route Prediction</i>	74
4.5.6	<i>Purpose Detection</i>	74
Chapter 5.	Performance Testing of GPS-enabled Mobile Phones.....	83
5.1	Evaluation of Transportation Environment Impact\Assisted GPS Estimated Accuracy	83
5.2	Impact of GPS and UDP transmissions on Battery Life of GPS-enabled Mobile Phone	85
5.3	Comparison of GPS Fix Times and Estimated Accuracies Returned By JSR179 Location API and Motorola OEM Position API on iDEN Phones	88
Chapter 6.	TRAC-IT Field Testing	90
6.1	Survey Method	90
6.2	Trip Information Feedback to Participants.....	91
6.3	Results	94
6.4	Discussion and Analysis of Results	95
6.4.1	<i>Qualitative Volunteer Feedback</i>	99
6.4.2	<i>Analysis of Technical Issues Discovered During Field Tests</i>	100
6.4.3	<i>TRAC-IT Deployment</i>	102
	Development Deployments	102
	Production Deployments	103
6.5	Status of Current Research.....	104
Chapter 7.	Recommendations for Deployment	105
7.1	Future Research	108
7.1.1	<i>Enhancements to the TRAC-IT system</i>	108
7.1.2	<i>Evaluation of Mobile Technology</i>	110
7.1.3	<i>Evaluation of Survey Protocol</i>	110
7.1.4	<i>Recruitment for Future TRAC-IT Testing</i>	111
References		112
Appendices		119

List of Tables

Table 1 - Summary of Available Positioning Methods.....	7
Table 2 - Summary of LBS Capabilities by Cellular Carrier.....	19
Table 3 - Cellular Data Networks.....	20
Table 4 - Cellular Network Technologies by Carrier	21
Table 5 - Financial Savings of Data Charges using Critical Point Algorithm	60
Table 6 - Sample DOR Codes and Their Description.....	77
Table 7 - DOR Codes and Their Relationship to TRAC-IT Purposes.....	78
Table 8 - Algorithm vs. Human for Synthetic Trips.....	81
Table 9 - Summary of Multi-Mode GPS Data.....	84
Table 10 - Battery Test Results using Sanyo 7050 Mobile Phone	86
Table 11- Sample Trip Information collected during the TRAC-IT Field Tests.....	94
Table 12 - Sample Location Data Points Recorded During Trip	95

List of Figures

Figure 1 - The Cell-ID Positioning Method	8
Figure 2 - Circular Lateralation	9
Figure 3 - Handset-Initiated vs. Network-Initiated Location Requests	11
Figure 4 - J2ME Architecture, including relationship to proprietary APIs	12
Figure 5 - Satellite Fixes and Cell Tower Coordinates (Point "U").....	16
Figure 6 - TRAC-IT System Architecture.....	39
Figure 7 - Main System Components of the TRAC-IT Architecture	40
Figure 8 - TRAC-IT Architecture Showing Communication Protocols	41
Figure 9 - The HTTP and UDP Wireless Transmission Methods.....	42
Figure 10 - Detailed System Architecture and Protocol of TRAC-IT Showing Relationship between Client and Server Components	43
Figure 11 -TRAC-IT Interface with J2ME Architecture.....	44
Figure 12 - Major Entities of the TRAC-IT Mobile Application	45
Figure 13 - TRAC-IT User Interface Diagram – Primary Screens.....	46
Figure 14 - TRAC-IT User Interface - QuickStop Screens	47
Figure 15 -The Recording Screen	52
Figure 16 -The State Machine Controlling Frequency of Location Updates and UDP Transmissions. Patent Pending, USF 2008.	54
Figure 17 - TRAC-IT Mobile Application Modules.....	55
Figure 18 - The TripTX Object	56
Figure 19 - The UDP Datagram Packet for TRAC-IT	57
Figure 20 - Brief Walking Trip – All Points Recorded (left) vs. Critical Points (right).....	59
Figure 21 - Car Trip with all GPS Points (left) vs. Critical Points (right).....	59
Figure 22 - Car Trip with all GPS points (left) vs. Critical Points (right).....	59
Figure 23 - Mobile Phone Network Carrier.....	61
Figure 24 - Internet Network Module.....	62
Figure 25 - Messages over the Email Server	62
Figure 26 - Wide angle View of Travel Information Displayed Using Google Earth.....	63
Figure 27 - Close-up View of Travel Data Displayed Using Google Earth	63
Figure 28 - Server-side Procedures called by Mobile Device	65
Figure 29 - TRAC-IT Phase 2 Data Schema	67
Figure 30 - UDP Transmission of Data Points	69
Figure 31 - The Automatic Mode Detection Algorithm.	71
Figure 32 - Hillsborough County Parcels/Properties GIS with Algorithm.....	75
Figure 33 - Percentages of GPS and Valid Fixes Broken Down By Mode	84
Figure 34 - Battery Tests Results using Sanyo 7050 Mobile Phone.....	86
Figure 35 - Fix Time J2ME JSR 179 Location API Versus Motorola OEM Position API	88
Figure 36 - Estimated Accuracy Uncertainty J2ME JSR179 Location vs. Motorola OEM Position	89
Figure 37 - TRAC-IT Trip in MSOutlook Meeting Invitation Format	92
Figure 38 - TRAC-IT Close-up View of Trip Data (Only Critical Points shown)	93
Figure 39 - Walking Trip Recorded by TRAC-IT Participant Shows all GPS Points Recorded.....	97
Figure 40 - Car Trip Recorded by TRAC-IT Participant Shows all GPS Points Recorded	97
Figure 41 - Bus Trip Recorded by TRAC-IT Participant - Shows all GPS Points Recorded.....	98
Figure 42 - Golf Cart Trip Recorded by TRAC-IT Participant Shows all GPS Points Recorded	98
Figure 43 - Bus Trip where GPS Path was not Recorded but O/D GPS Points were Recorded	99

List of Acronyms

<u>Acronym</u>	<u>Definition</u>
3GPP	3rd Generation Partnership Project
AFLT	Advanced Forward Link Trilateration
A-GPS	Assisted Global Positioning System
API	Application Programming Interface
BREW	Binary Runtime Environment for Wireless
CDMA	Code Division Multiple Access
CELL-ID	Cellular Base Station ID
CLDC	Connected Limited Device Configuration
DOR	Department of Revenue
ECC	Elliptical Curve Cryptography
E-OTD	Enhanced Observed Time Difference
ESTI	European Telecommunications Standards Institute
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IMEI	International Mobile Equipment Identity
IMS	IP Multimedia Subsystems
J2EE	Java 2 Platform, Enterprise Edition
J2ME	Java 2 Platform, Micro Edition
J2SE	Java 2 Platform, Standard Edition
JAX-RPC	Java API for XML-based Remote Procedure Calls
JCP	Java Community Process
JDBC	Java Database Connectivity
JSR	Java Specification Request
JSR172	Java Specification Request 172 - J2ME Web Services Specification
JSR179	Java Specification Request 179 - Location API 1.0
JSR293	Java Specification Request 293 - Location API 2.0
LCS	Longest Common Subsequence
MAM	Mobile Application Manager
MIDlet	Java application for J2ME
MIDP	Mobile Information Device Profile
MLP	Mobile Location Protocol
MVM	Multitasking Virtual Machine

OEM	Original Equipment Manufacturer
OMA	Open Mobile Alliance
OTA	Over the Air
PDA	Personal Digital Assistant
PTT	Push to Talk
QJAE	Qualcomm's Java Application Extensions
SDK	Software Development Kit
SIM-ID	Subscriber Identify Module ID
SMS	Short Message Service
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TDOA	Time Difference of Arrival
TOA	Time of Arrival
TTFF	Time To First Fix
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications Service
U-TDOA	Uplink-Time Difference of Arrival
WAP	Wireless Application Protocol

Chapter 1. Introduction

1.1 Project Background

Transportation demand management (TDM) strategies are designed to motivate people to modify their travel choices, particularly the drive-alone preference. TDM strategies and policies seek to increase transportation system efficiency and achieve specific objectives such as reduced traffic congestion, road and parking cost savings, increased safety, improved mobility for non-drivers, energy conservation, and pollution emission reductions. Marketing these strategies to the “consumer” in an effective way that results in behavioral changes necessitates a sustained understanding of how, when, where, and why people travel. Therefore, accurate, timely, and comprehensive data are vital in understanding/influencing the consumer. Data collection processes have made use of new, emerging technologies that help refine the task of monitoring/measuring the behavior to be modified. In addition, setting future mobility management goals requires the study of activity-based travel surveys.

This research focuses on using innovative technology to better understand and pattern household travel behavior for the purposes of educating, promoting, and encouraging households to utilize other alternatives to driving in general and to driving alone in particular. Phase 1 in this research series developed and pilot tested an application named TRAC-IT to track an individual’s travel behavior using a Personal Digital Assistant (PDA) platform linked with a Global Positioning Systems (GPS). The system works across various modes of transportation and automatically analyzes the data collected from the device to give personalized feedback advice, based on its server-side expert system. TRAC-IT captures travel patterns regardless of mode, automates the collection of travel characteristics (e.g., time, distance, and even mode), transmits the data to a database, conducts the analysis of the household’s travel patterns using a travel advisory feedback system, and provides advice to the individuals in the household about trip chaining using transit, biking, and walking as well as carpooling options.

The scope of Phase 1 called for preliminary development and testing of a portable unit consisting of a personal digital assistant (PDA), a global positioning system (GPS) device, and a wireless card - an all-in-one unit nicknamed “TRAC-IT” (2). Phase 2 in this research effort revamped the travel advisory feedback system making use of the initial testing conducted in Phase 1, and investigated and analyzed the impacts of the feedback advice system on household travel behavior. Phase 3 is dedicated to the design, development, and testing of the GPS-enabled cellular phone as a TRAC-IT unit. Although Phases 2 and 3 were conducted simultaneously, the Phase 2 field testing portion depended on the effectiveness of TRAC-IT cell-phone software development and completion in order to use cell phones to measure household travel behavior before and after expert system feedback was generated.

Mass production has substantially lowered the cost of “smart phones” on the cellular phone market, making them more affordable for the average consumer now carrying his or her own personalized computing devices. Due to the advancement of cell phone and wireless technologies, data functionality (such as text or multimedia messages, email, and streaming video) are now possible at a variety of broadband speeds, making the popularity of these media soar. Additionally, the Federal Communications Commission (FCC) initially mandated that all cell phones meet the E911 standards of location reporting within 50 to 300 meters by Dec. 31st, 2005, a directive that cellular providers in coordination with Public Safety Answering Points (PSAPs) are still working towards as of early 2008 (3). This mandate sets the stage for commercialized location-aware services that will be personalized to users based on their current physical location. These same services can contribute further to the TRAC-IT system.

Cell phone popularity continues to grow, as does the sophistication of the devices. It is estimated that currently over one half of the world’s population, or 3.25 billion people, owns a cell phone (4). This project expanded the PDA-based TRAC-IT by adapting the application to cell phones and integrating the technology with other databases such as 511 traffic information systems and transit AVL systems to increase the utility and effectiveness of the application. Location-aware services such as approaching transit vehicles, alternate routes, and driving directions can be calculated and delivered directly to the user based on both real-time incident and traffic conditions and their past travel behavior, allowing them to alter their mode or planned route before they encounter a problem. It also will examine modifications on the client side (e.g., within the phone) and server-side (e.g., hosted on a computer) to make the TRAC-IT system fully automated and scalable from small towns to large urban cities.

1.2 Study Purpose and Objective

The goal of this research is to influence travel behavior by mode, route, or time of day through the integration of traveler information, GPS, location-aware services and TRAC-IT’s PDA-based travel behavior advisory system into cell phone applications. The objective of this project was to determine the capabilities of GPS-enabled mobile phones in tracking person movements across modes (car, bike, bus, etc.) and over extended time periods (e.g., weekly versus daily).

1.3 Research Methodology

The research performed for this phase of TRAC-IT focused on the development of a software-based mobile application that could be installed on commercially-available, GPS-enabled mobile phones. In addition, TRAC-IT software for a server was developed to receive and store data collected by the mobile application. The concept of the research methodology for this study was based on the findings of Phase 1 that successfully tested the TRAC-IT prototype on a PDA. Advanced capabilities of mobile devices guided the futuristic vision that the research team developed to utilize these

capabilities in next generation travel surveys, real-time traveler information dissemination, and travel blending techniques.

The research methodology was to develop a mobile application that would collect travel survey data with minimal interaction from survey participants. A basic requirement of the system was that the mobile device selected had to be a commercially available, low cost, off-the-shelf, and widely used. The device also had to be equipped with embedded GPS capabilities for highly accurate position data, able to communicate this information wirelessly back to a server, and able to receive user input allowing the manual entry of survey data by survey participants that cannot be extracted from GPS data. The wireless communication features of the mobile phone could also be used to offer users real-time information to users that could influence their current travel behavior.

Initial technology assessment was conducted to determine the best hardware and software platform for TRAC-IT development that met design criteria summarized previously. TRAC-IT software for the mobile phone and server were under development and continuous testing in an iterative process for the duration of this phase and Phase 2 of this study to refine and finalize the software. Performance testing for critical TRAC-IT characteristics such as estimated accuracy and power consumption were designed and developed based on the results of these tests optimizing the TRAC-IT software application and minimizing impact on device resources. Finally, a field test was performed with participants carrying GPS-enabled mobile phones with TRAC-IT software installed to evaluate the data collection capabilities of TRAC-IT.

1.4 Report Organization

The next chapter presents a technology assessment for Location-Based Services (LBS) and cell phones that the research team completed by examining the current capabilities of cell phones and those of different providers as well as software developer methods. Chapter 3 describes the design of the TRAC-IT system based on the technology assessment findings, while Chapter 4 describes the TRAC-IT system architecture. Chapter 5 discusses the preliminary testing conducted using the GPS-mobile phone and the challenges that emerged during the deployment phase. Chapter 6 reports how the TRAC-IT cell phone application was used in tracking travel behavior in surveys conducted by the research team. Findings, recommendations, and future research are summarized in last chapter of this report.

Chapter 2. Technology Assessment

The research team conducted an extensive analysis of mobile phone technology to evaluate the feasibility of using mobile phones to record travel behavior at an individual level. This investigation included an analysis of positioning methods available for cellular networks and devices, methods of software development to gain access to position data, software standardization and compatibilities for mobile phones, and the current status of positioning techniques.

2.1 Location-Based Services Technology for Cellular Networks

LBS are an exciting area of emerging technologies in the telecommunications industry. Common applications of LBS include real-time navigation systems, while newer location-aware applications include location-based messaging, social networking, and photography. With the recent advancements in mobile phone and network technology, LBS are growing at an astounding rate. According to recent market research, the world population of GPS-enabled, location-aware services subscribers will grow from 12 million in 2006 to a projected 315 million by 2011, and North American growth is projected from 500,000 users in 2006 to 20 million users in 2011 (5) (6). The widespread nature of GPS-enabled mobile phones provides the ideal environment for new types of LBS to be established. Since LBS are well suited for transportation applications, new types of location-aware transportation applications can be developed and deployed on a large scale.

Before discussing LBS technology in depth, it is important to differentiate between the two main methods of utilizing mobile phones to obtain travel information. The first method uses cellular signaling data to estimate average speed or travel time on road segments. The second method uses positioning technologies such as GPS to determine the precise location of each individual mobile phone that is being examined. It is important that transportation professionals who collect and analyze travel surveys appreciate the difference between the two methods of data collection as they yield very different types of data that can be used for different purposes.

The first method uses anonymous cellular signaling data gathered at cell towers to estimate travel time on nearby highways. This technique works by looking at where and when signal handoffs occur when the cell phone travels from one tower's coverage area to another; proprietary algorithms then attempt to determine road segment speeds allowing estimates of travel times from this data. It is assumed that the mobile phone user's permission is not required to use this signaling data, although privacy advocates are pushing for more control over consumer records. Travel time and estimated travel speed on a specific road segment are the only items of information available from these types of systems, which may be sufficient for traffic operations. However, travel path, individual travel behavior, and origin and destination information are not available from this method of data collection. Companies operating these

systems in the U.S. include AirSage, Cellint, Delcan, and IntelliOne. They have been evaluated by various DOTs with varying results of system accuracies (7), (8), (9). General conclusions seem to be that the systems work well at free flow speeds, but accuracies vary when traffic is traveling at lower speeds. These results may occur because there are not as many handoffs between towers occurring when phones are not moving a large geographic distance, so it is harder to extrapolate road speed data when the frequency of handoffs is low.

The second method is the use of GPS-enabled mobile phones to gather continuous position data for each mobile phone, including speed and heading data, calculated by the phone or network using Global Positioning Systems (GPS), network triangulation, or a similar positioning technology. The TRAC-IT research project focuses on this particular method. GPS is highly accurate, with most position requests being accurate within 3 to 30 meters and speeds shown to be accurate within 0.2 meters/sec in past studies (10). GPS fixes can be collected with a frequency of up to once per second for each mobile phone, thereby generating a wealth of information of an individual's travel behavior. This method yields origin-destination (O-D) data as well as travel path information for each individual traveler and is a modern version of the traditional trip diary. Travel speeds and estimated travel times can also be extracted from these data (although values would have to be aggregated from multiple individual GPS data points to accurately reflect average speed instead of a single vehicle's speed).

GPS has been studied in the past as a method for replacing travel diaries and has exhibited excellent results (11). The advantage of using GPS-enabled mobile phones is that the general public uses them, thus eliminating equipment cost for survey deployment. A software application must be distributed to the device to enable it as a data collection tool. Another advantage is that GPS reporting is continuous across all modes of transportation, and represents the user's movements, whether driving, walking, or biking. The one vital requirement for tracking personal travel behavior is that the individual carry the phone on all the trips.

Even though utilizing GPS-enabled mobile phones in data collection has many benefits and offers a wealth of information, there remain challenges to overcome when deploying such a method. First, a software infrastructure that spans both the mobile device and a server must be developed. There can be adverse effects on handset performance, such as battery life or the ability to receive incoming phone calls if the position requests are not efficiently managed by the software running on the phone. Therefore, the software developer must have advanced knowledge of software information systems for mobile phone. Additionally, users must opt-in to install the software application on their phone, so there should be some kind of incentive for the participants to do so. This could be either allowing the user access to some kind of service, such as real-time traffic information alerts or providing traditional monetary incentives. The TRAC-IT project examines this method of data collection and evaluates a prototype architecture that implements such a method.

Location-aware applications require access to geographic data that describes the real-time position of the device. There are three common classes of positioning techniques that can be utilized to obtain geographic information:

- ♦ Device-based methods, where the end-user device performs the necessary calculations in order to determine the position of the device.
- ♦ Cellular network-based methods, which determine the device position by analyzing the wireless communication with the device.
- ♦ Hybrid methods, used primarily to improve the Time-To-First-Fix (TTFF) and the accuracy of the position estimation. Hybrids utilize end-user devices to perform measurements and send the information to the network where position calculations take place: “device-assisted network-based” or “network-assisted device-based.”

The Federal Communication Commission’s E911 mandate requiring cellular providers to identify the position of 911 wireless callers has been the driving force behind the implementation of LBS in the United States (3). Since many different technologies can be used to implement any of the above positioning methods, Phase 2 of the E911 mandate has left the implementation of the positioning system technology up to each carrier and has specified only accuracy requirements. Device-based solutions must be within 50 meters for 67 percent of calls, and 150 meters for 95 percent of calls. In the case of network-based solutions, 67 percent of calls must be within 100 meters, and 95 percent within 300 meters (12). Table 1 includes a summary of all positioning methods discussed in the following sections and the U.S. carriers utilizing them. The current trend for U.S. cellular carriers is to provide assisted Global Positioning Systems (A-GPS) solutions for handsets, regardless of the solutions implemented to meet the E911 mandate requirements. This trend is due to the fact that most location-aware commercial products such as real-time navigation require the high precision positioning data that only A-GPS is capable of providing. Therefore, if the cellular carrier is interested in recovering the cost of implementing LBS on its network and maintaining a steady income based on commercial LBS, A-GPS must be available in devices on the network.

All information shared in this report is publicly-available information collected from a variety of sources, including the wireless service providers’ and cell phone manufacturers’ websites and documentation hosted on these websites. Information such as software compatibilities and mobile phone performance data was gathered by empirical testing of GPS-enabled cell phones using software created by the research team.

Table 1 - Summary of Available Positioning Methods

Location Technologies for Cellular Phones			
Technology	Type	Accuracy	U.S. Carrier
Cell-ID	Network	100m-3km (Depends on the size of the cell)	All
Cell-ID + Timing Advance (TA)	Network	Band size configurable. Default is 500m	AT&T
Enhanced Forward Link Triangulation (EFLT)	Network	250-350m	None
Angle of Arrival (AOA)	Network	100-200m	None
Advanced Forward Link Trilateration (AFLT)	Network	50-200m	Sprint and Verizon
Enhanced Observed Time Difference (EOTD)	Device	50-200m	None
Uplink Time Difference of Arrival (U-TDOA)	Network	40-60m	T-Mobile, AT&T
Assisted Global Positioning System (A-GPS)	Hybrid	5-30m	Sprint, Nextel, Alltel, Verizon, and AT&T

Source: Federal Communication Commission Official Website link to 911 services
<http://www.fcc.gov/911/enhanced/>

2.1.1 Device-Based Positioning Methods

The most common device-based positioning method is based on GPS technology. GPS utilizes hardware and software that resides in the end-user device to determine the position of the device. Many cell phones and certain PDAs are equipped with embedded GPS receivers that receive the signals from satellites. The GPS system is a constellation of 24 satellites that broadcast data that can be processed by end-user devices to calculate their position using the Time of Arrival (TOA) method and applying circular lateration in combination with timing measurements. The constellation consists of six orbits spaced 60 degrees apart, with four satellites per orbit. This guarantees that a GPS receiver is under the coverage of at least four satellites at any given point in the earth's surface.

The main advantage of GPS is that very high-accuracy location data can be obtained within 5 meters of accuracy in ideal scenarios. This extreme level of accuracy allows the development of real-time applications that interact with the users based on their current position, for example, vehicle navigation systems that provide the driver with real-time driving directions utilize GPS. However, some disadvantages also exist. First, since the GPS transmissions from the satellites are very weak, the device must have a clear view of the sky to receive the transmissions used to calculate its position. This means that pure GPS positioning solutions do not work indoors or in situations where radio signals may be interrupted, such as during severe weather or in "urban canyons" (areas surrounded by many tall buildings).

Second, GPS devices can take a significant amount of TTFF - up to 2 minutes or more - when the GPS hardware is first turned on. This scenario, referred to as a "cold-start," results from the GPS hardware having to scan many radio channels to determine what satellites may be in view. Subsequent fixes, referred to "warm" or "hot" starts,

depending on the idle period, can be obtained at a faster rate of approximately one fix per second.

2.1.2 Network-Based Methods

Many network-based positioning mechanisms are available in wireless cellular networks. In the United States, these mechanisms have been introduced mainly because of the E911 mandate that requires carriers to provide positioning information for emergency calls. The FCC executed the E911 mandate in stages, requiring the end device position to be determined within a "cell" (the area of cellular network coverage) by 1998, and more accurate position information by the end of 2005. The Cell-ID positioning method, (Figure 1), which simply returns the position of the center of the cell with which the device is currently communicating, varies in accuracy based on the size of the cell (anywhere from 100m in urban areas with high-density cellular coverage to 20,000m in rural areas with sparse coverage).

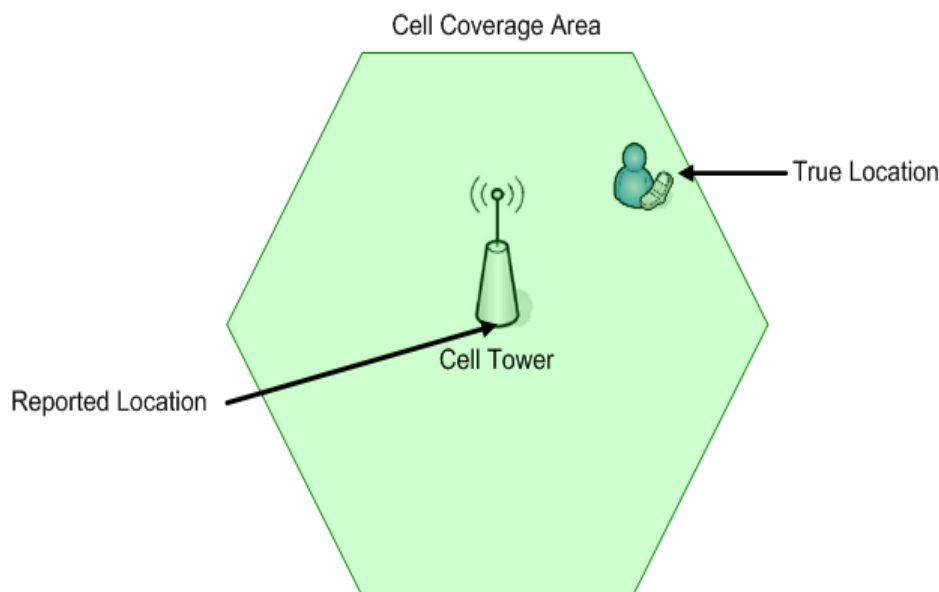


Figure 1 - The Cell-ID Positioning Method

To meet the more stringent requirements of E911 Phase II, more advanced mechanisms have been developed, such as the Enhanced Observed Time Difference (E-OTD), Time Difference of Arrival (TDOA), Advanced Forward-Link Trilateration (AFLT), and Uplink Time Difference of Arrival (U-TDOA). To calculate the user's position, most of these mechanisms utilize the Hyperbolic Lateralation mechanism, which is similar to the well-known circular lateralation mechanism, shown in Figure 2. In these mechanisms, cellular base stations are utilized as fixed position references similar to the satellites utilized in the GPS system. In other methods, positioning calculations are performed by the network after collecting enough information from the end-user device and close base stations. A helpful overview of positioning systems for wireless cellular networks and location-based services was published in recent editions of *Dispatch Monthly Magazine* and *FCC E911 – Wireless Services* (13), (14).

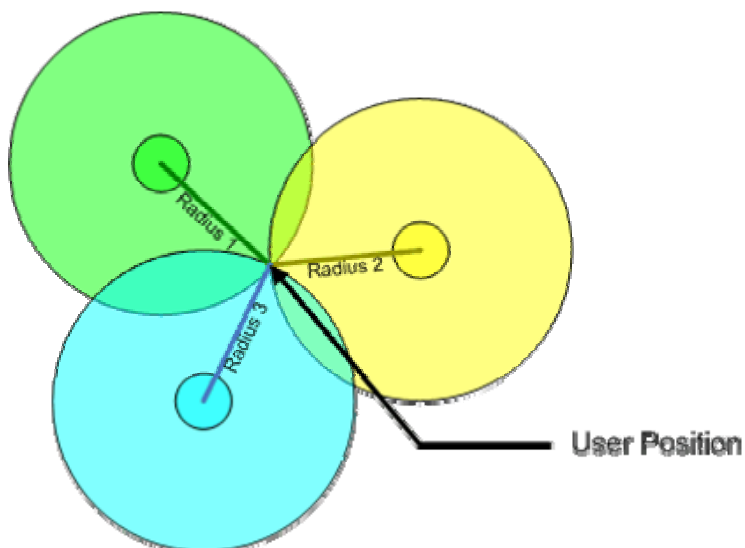


Figure 2 - Circular Trilateration

Network-based techniques have the advantage of using cellular signals in order to determine position information. Therefore, network-based methods are able to determine the position of a device wherever it can receive cellular signals, including indoors and within urban canyons. However, network-based solutions are not able to supply the extreme levels of accuracy associated with GPS. These solutions can accurately supply data up to approximately 50 meters in ideal conditions, which limits their use for certain positioning applications. To date, the U-TDOA method is perceived as the best network-based positioning method to meet E911 accuracy requirements.

2.1.3 Hybrid Positioning Methods

Due to a desire to combine the advantages of both device and network-based solutions, some “hybrid” positioning technologies such as Assisted GPS (A-GPS) have been developed. A-GPS uses standard GPS signals in coordination with network-based data or techniques, such as Advanced Forward Link Trilateration (AFLT), to obtain highly accurate positioning information and surpass the limitations of pure GPS. For example, the TTFF can be significantly reduced if the GPS device is supplied with an estimate of where it may be located and which satellites are in view. The cellular network can provide a rough location of the cellular device, allowing it to narrow the search space of what GPS satellites may be in view. New hardware architectures for embedded A-GPS chips in mobile phones also are increasing the sensitivity of the GPS receiver, allowing positioning indoors and in other environments where previous GPS-enabled mobile phones were unable to obtain a fix (15).

Hybrid methods also provide a network-positioning backup for scenarios where A-GPS may be not available. For example, if GPS hardware cannot obtain an accurate fix within a given timeout period, the device can provide AFLT or Cell-ID information.

Even though the Cell-ID position accuracy can vary depending on the size of the serving cell, at least some coarse location information is available to the application (16).

2.2 Software Developer Methods to Obtain Position Data

Although different technologies are used to calculate the user's position, software development should not be further complicated by the need to differentiate the details of one type of technology from another when using position data in a software application. As a third party software, the goal of location Application Programming Interfaces (APIs) is to provide developers access to position data while hiding some of the implementation details from the software developer. There are many APIs available for this purpose, including those defined for software that executes on handsets as well as those defined for use by network applications. Handset-based APIs are accessed by software running on the mobile device through "handset-initiated" location requests, while network-based APIs are accessed by web applications that send "network-initiated" location requests to a server in a cellular network (Figure 3).

There are advantages and disadvantages to each approach, which are discussed in detail in the following sections. It should be noted that methods by which a third-party software obtains location information (i.e., "handset-initiated" or "network-initiated" location requests) are separate from methods by which position information is calculated. Therefore, a handset-initiated location request could return location data determined by a network-based positioning method, and similarly a network-initiated request could return location data determined by a device-based positioning method. The API abstracts the process of position calculation and transfer of the resulting data from the software application requesting the information. It should also be noted that access to position data via an API and the implementation of E911 services are completely separate entities. Therefore, all cell phones provide some level of E911 service to callers by using some type of positioning technology. However, not all cell phones or cell phone carriers support and/or allow access to location information by third party software. Both privacy concerns and technological barriers contribute to the complexity of cell phone applications.

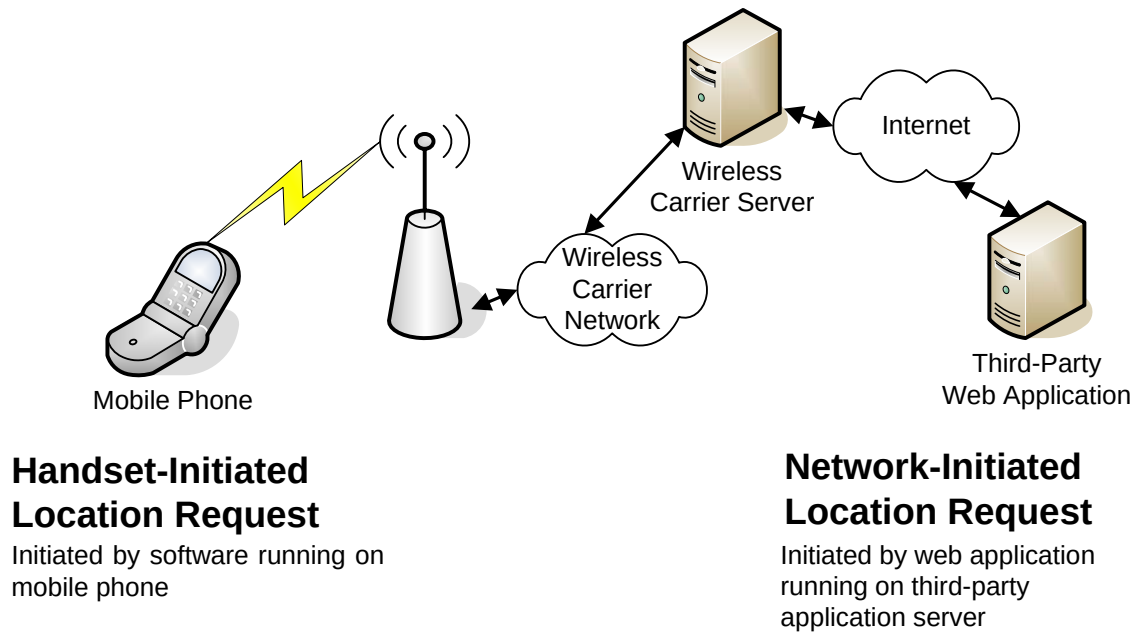


Figure 3 - Handset-Initiated vs. Network-Initiated Location Requests

2.2.1 Handset-initiated Location Requests

The large number of different mobile device manufacturers and cellular carriers make software development for mobile devices rather complex. There are many different programming languages available, many of which have functions proprietary to a particular chipset or operating system. Therefore, while an application developed for a proprietary system may operate on one mobile phone, it would have to be completely re-written to execute on a device from a different manufacturer. One programming language that has emerged as a platform-independent means of software development for mobile phones is Java 2 Micro Edition (J2ME) (17). J2ME, like Java 2 Standard Edition (J2SE) and Java 2 Enterprise Edition (J2EE) for desktop and server computers, is able to execute on devices and chipsets from different manufacturers. Additionally, J2ME has been designed to work across different cell phone networks independent of the underlying cellular network implementation, such as Code Division Multiple Access (CDMA) or Global System for Mobile Communications (GSM). As a result, a J2ME application designed to meet the basic J2ME standard that utilizes no proprietary libraries should be transferable from one J2ME-enabled device to another with few or no changes in the software. The J2ME Architecture is shown in Figure 4 and is discussed in detail in section describing Client-side Modules of TRAC-IT.

J2ME Architecture, including relationship to proprietary APIs

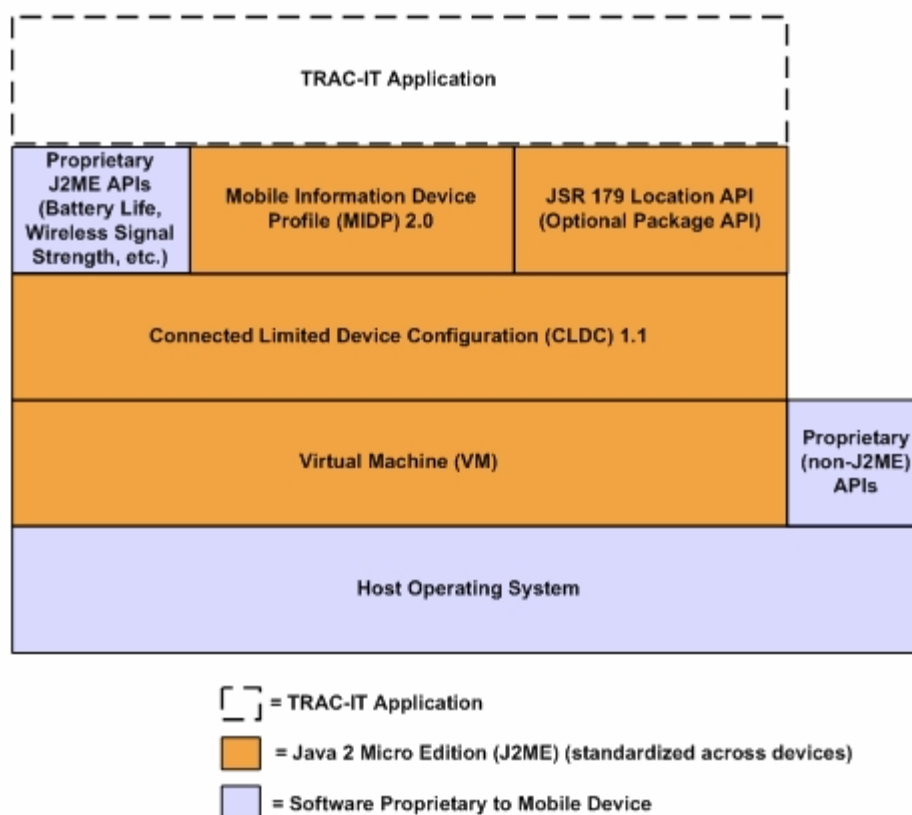


Figure 4 - J2ME Architecture, including relationship to proprietary APIs

As A-GPS capabilities became available in mobile phones, vendors began to expose this functionality through proprietary APIs designed for handset-initiated location requests. Using these APIs, third party software applications running on the cell phone could utilize A-GPS data to provide various types of LBS. These proprietary APIs can be divided into two main categories: proprietary APIs for proprietary operating systems, and proprietary APIs for Java 2 Micro Edition (J2ME).

Nokia's "Location Acquisition API" for the Symbian S60 platform is an example of one such proprietary API that falls into the first category (18). Applications developed in C++ using Nokia's API will execute properly on that specific operating system, but the same application is not transferable to other manufacturer's devices. Qualcomm's Binary Runtime Environment for Wireless (BREW) platform also falls into this first category, as it exposes access to position information via an "IPosdet" (Position Determination) interface (19). However, BREW applications will execute only on devices that utilize Qualcomm's chipset and BREW platform.

Overview of J2ME Location API

As previously mentioned, J2ME applications are more portable than those developed for specific platforms such as Symbian and BREW. Since Symbian and BREW provide

the means to run J2ME applications on top of each of these platforms, it is preferable to create J2ME applications because of their portability to any platform that is "J2ME-enabled." However, manufacturers can still choose to expose proprietary APIs in J2ME that are accessible by J2ME applications on that particular device. J2ME applications that use these proprietary APIs will be able to use that particular application functionality only on devices that support the specific proprietary Java API. An application that utilizes a proprietary J2ME API differs from applications created for proprietary operating systems in that the main functionality of the J2ME application is still portable to a different J2ME-enabled device and only the proprietary J2ME API for a specific function (e.g., location) cannot be utilized on other devices. Applications created for proprietary operating systems must be completely recreated for a new platform, since none of the application will be portable. Examples of proprietary J2ME location APIs are Motorola's OEM Position API for integrated Digital Enhanced Network (iDEN) devices as well as Qualcomm's Java Application Extensions (QJAE) API for devices utilizing Qualcomm chipsets (20), (21). Performance of these various APIs can differ in terms of response time and accuracy of position data, so software developers should be aware of the particular characteristics of the API they are planning on using, as well as the other options for location APIs on the same device, before developing LBS using a particular API (22).

When J2ME became more prevalent in the mobile device market and location capabilities of mobile phones became more common, it was evident that a standardized API was required on the J2ME platform that would allow location-aware software to be deployed to many different devices without the use of various proprietary APIs. Fortunately, the Java Community Process (JCP) allows new standards to be proposed for Java platforms, including J2ME. In 2002 "Java Specification Request (JSR) 179: Location API for J2ME" was proposed and accepted as a necessary standard for J2ME (23). An expert group including members such as IBM, Nokia, Symbian, Intel, Motorola, Sony-Ericsson, ESRI, and Sun Microsystems, Inc. was created to develop the specification for this standard and in September 2003 the final standard was released. This standard allows a location-aware J2ME application to be developed that will execute on any J2ME platform that supports the optional JSR179 package. In an effort to improve JSR179 and add additional location-aware functionality, "JSR293: Location API 2.0" was proposed in early 2006 (24). An expert group consisting of Nokia, Motorola, Sprint-Nextel, GPS hardware manufacturer SiRF, Navteq, Samsung, Cingular, Sun Microsystems, Inc., Sony Ericsson, European cellular providers Orange and Telecom Italia, China Mobile Communications Co. Ltd, and USF has been deliberating on this standard since this time, with the public review stage taking place in mid 2007. JSR293 is expected to be finalized in mid to late 2008 and should appear in commercially-available cell phones shortly after this time. As of early 2008, it is understood that J2ME-enabled cell phones that support a "standardized" location API support JSR179. Even though there may be performance differences between JSR179 and other proprietary location APIs on the same device, portability is one of the most important factors in mobile applications. Unless the performance

differences are drastic, JSR179 is currently the preferred location API for J2ME software development as of late 2007. Similarly, when JSR293 becomes available, JSR293 will likely become the preferred location API for J2ME.

Google Android Platform

Google's Android platform for mobile phones is another proprietary mobile phone platform that deserves mention (25). Android is an open-source software stack based on Linux that will span several mobile device and chipset manufacturers similar in concept to J2ME, although not nearly as widespread or as generally accepted as a development platform compared to J2ME. Software developers will write code for Android applications in the Java programming language, although the accessible APIs for Android applications are proprietary in nature and do not conform to J2ME standards, including a proprietary location API that is very similar in functionality to JSR179 but does not conform to the specification. Android devices are expected to contain an implementation of the J2ME platform, which will allow J2ME applications to execute on top of the Android platform similarly to how J2ME applications execute on BREW and Symbian platforms. As of late 2007, Android was not available on any hardware phones accessible to the general public but is available for execution on software emulators. It is debatable how widespread and accepted Android will become and a variety of technical and political hurdles will decide whether it is a worthwhile platform for mobile application development. Currently, J2ME remains the accepted standardized platform for Java development on mobile devices, and JSR179 remains the only standardized location API until JSR293 becomes available.

JSR179 Location API 1.0

JSR179 Location API 1.0 exposes two main functions to access location data on the cell phone through a LocationProvider. The principal function of the LocationProvider is to select a device-based, network-based, or hybrid method by which the device's location will be determined, based on criteria submitted by the application that defines the desired accuracy of the location fix as well as a timeout value of how long the function should search for a fix before it returns. After obtaining an initial LocationProvider reference from the JSR179 implementation, the application can request position data through a simple call to a function "getLocation()," which will return the current location of the device (if the location can be determined). This "one-shot" functionality is ideal for applications that need to request location information once, such as a geo-tagging a photograph or text-message. When a high-accuracy fix is first requested (i.e. a GPS "cold start") from a phone with Assisted GPS technology, the phone will attempt to utilize assistance information from the network that lessens the TTFF. If the GPS hardware cannot obtain a fix that meets the provided accuracy criteria within a given timeout period, then the J2ME Location API will usually provide Cell-ID information for the cellular area that is currently serving the device along with a note that a high accuracy GPS fix was not attainable. If a GPS fix can be obtained, it will be returned to the user along with other information, including the estimated accuracy of

the GPS fix based on the number of available satellites as well as other positioning data. Subsequent “hot” fixes, or fixes requested soon after a previous fix should return new location fixes rapidly unless the GPS signal is obstructed. If a slight delay between fixes occurs, the subsequent fixes are referred to as “warm” fixes, which take slightly longer to acquire a fix than a “hot” fix. If the application does not query the J2ME Location API for an extended amount of time, the GPS hardware will have to begin from the “cold start” state again since the GPS chip will have powered down.

The second way an application can request position data is to register a `LocationListener` with the `LocationProvider`. This `LocationListener` is a mechanism that monitors the device’s location over an extended period of time and provides location updates to the application at requested intervals (e.g., every 10 seconds). This method is ideal for applications that require continuous updates on the position of the device, such as tracking or navigation applications. Since the update interval is defined by the application, location updates can be provided as frequently as once per second or as frequently as the underlying hardware and operating system can allow, whichever is less frequent. If the device is informed that the application is going to require positioning data over an extended period of time, it can optimize power consumption and data communications for this scenario, thereby making use of the `LocationListener` more efficiently than repeated calls to the `getLocation()` function. The device can maintain either hot, warm, or cold states underlying the API depending on what combination is most efficient for the interval registered by the application and the characteristics of the GPS hardware. Additionally, the implementation of the JSR179 `LocationListener` should efficiently handle the multithreaded environment required for this functionality, which reduces the burden of managing these threads by the third party application. Both the “one-shot” location requests and the `LocationListener` provide software application running on mobile devices with immediate low-latency access to location information, which is ideal for real-time location-aware applications that require detailed position information or a high level of control over the positioning process.

It should be noted that, although the JSR179 Location API is a standardized specification for input/output parameters and behaviors of defined functions, there is no standard implementation of JSR179. That leads device manufacturers to interpret the specification differently in certain areas that are somewhat vague resulting in different behaviors of the software that implements JSR179 on different devices. While the JSR179 specification is generally well-designed, there are a few areas where behaviors of JSR179 may differ. Certain aspects of JSR293 focus on reducing the ambiguity in the JSR179 specification by clarifying expected implementation behavior and further defining objects and functions to support this expected behavior. Further information on these areas of improvement in JSR179 and a comparison between JSR179 and JSR293 can be found in the March 2008 special issue of *Computer Communications* on “Advanced Location Based Services” (26).

The most popular positioning method used to satisfy handset-initiated location requests from applications with very high accuracy requirements is assisted GPS, and most often Cell-ID is used for coarse location information when a GPS fix cannot be obtained or the precision of GPS is not required. A relationship can be shown between GPS data collected in Tampa, Florida and the Cell-ID location fix labeled "U" that was returned when GPS data was not available (Figure 5).

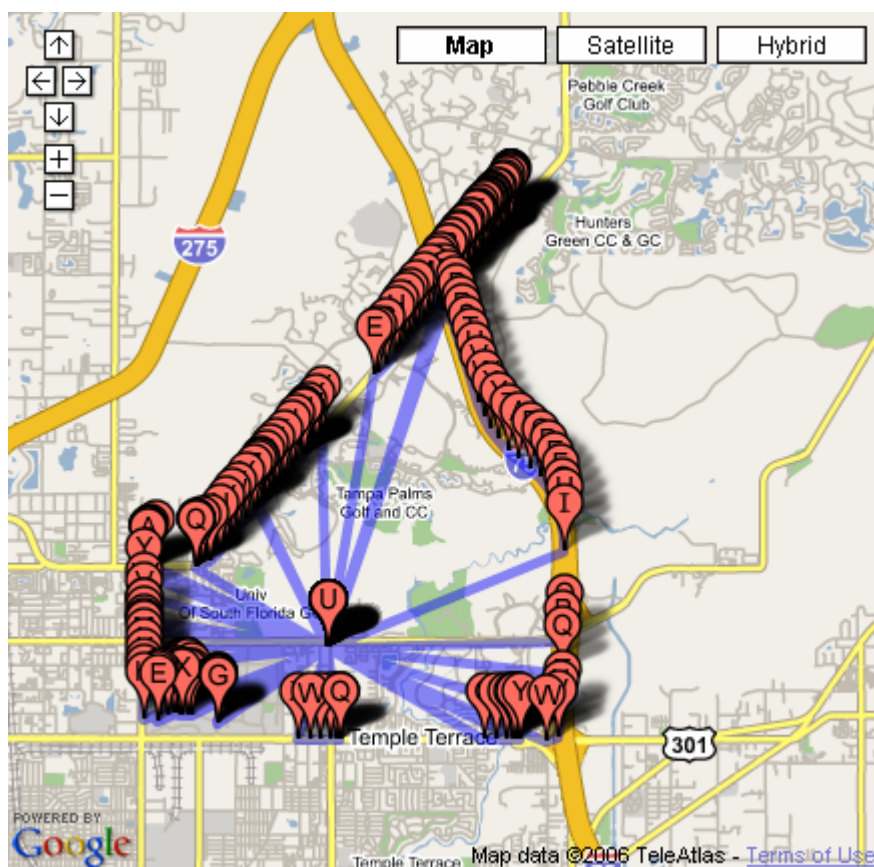


Figure 5 - Satellite Fixes and Cell Tower Coordinates (Point "U")

Network trilateration techniques, such as AFLT, also function on certain networks. However, frequency of use is limited by cellular providers to a location fix approximately once every 15 minutes due to the extra load AFLT fixes place on network resources. Network trilateration positioning and GPS have the highest impacts on battery consumption due to energy expended during wireless communication and GPS hardware activation, while Cell-ID has minimal energy impact. Dead-reckoning is a type of positioning technology that utilizes the last known location in combination with internal sensors, such as accelerometers, to estimate current location during GPS failures. Currently, JSR179 implementations do not support dead-reckoning. Consequently, any JSR179 implementation that exposes dead-reckoning will do so via a proprietary API. Since support for dead-reckoning is included in JSR293 and since devices including accelerometers are becoming more common, it is expected that a

basic form of dead-reckoning technology will be available on some cell phones implementing the JSR293 specification.

The J2ME security model prevents malicious code from being installed and executed on a Java-enabled mobile device. Part of this security model includes the protection of "sensitive" APIs from unauthorized access by applications. JSR179 is considered one of these sensitive APIs since it exposes the geographic location of the user, which is private information. For an application utilizing JSR179 to successfully be installed on a mobile device, it must be signed using a digital certificate from an authorizing party. In the United States, since the wireless carrier controls access to its wireless network and devices are often locked to one network, it follows that the authorizing parties are the cellular carriers. Usually, business agreements must be reached and a certification process must be completed before an application will be signed by a wireless carrier. Development certificates may be used to sign applications

2.2.2 Network-Initiated Location Requests

As an alternative to handset-initiated location requests from software running on a cell phone, cellular carriers can make device location information available to web applications that query a carrier's location server. In contrast to handset-initiated location requests, network-initiated location-requests do not require third party software to be installed on the device. Instead, a third party web application submits a query to a cellular carrier's location server defining a request for a particular mobile device. If the web application has proper permissions, the location server then determines the location of the device through either a device-based, network-based, or hybrid positioning method and then passes the result of this positioning attempt to the web application. Several standards define how location is calculated for handsets and handled internally for a cellular providers network including the European Telecommunications Standards Institute (ETSI)'s "GSM 03.71: Location Services (LCS)" and the 3rd Generation Partnership Project (3GPP)'s "TS 22.071: Location Services (LCS)" and "TS 23.171: Functional stage 2 description of location services in UMTS" (27). These standards are more important to telecommunications professionals than to typical software developers. Standards important to software developers interested in obtaining device location information through web applications, are the Open Mobile Alliance (OMA) Location Working Group's Mobile Location Protocol (MLP) as well as a set of telecommunication web services known as Parlay X (28), (29). Each of these standards specifies a format that network-initiated requests should follow, as well as the expected format of the response from the location server. While network-initiated location requests have the advantage of not requiring a third party application to be installed on the mobile device, they are subject to limitations and access restrictions by the cellular carriers. These limitations can include access restrictions to cellular carrier industry partners, restrictions on the frequency of location update requests (i.e., only one position update every 10 minutes per phone), restrictions on the number of devices queried simultaneously (i.e. only 20 phones can be queried for their position at

a time), and a limitation of the amount of location data returned (i.e. provides only latitude and longitude, but not altitude, heading, or speed).

2.2.3 Current Capabilities of Cellular Providers and Cellular Devices

The following sections discuss device and network characteristics for developing and deploying LBS applications for mobile phones and the capabilities of major U.S. cellular carriers to support these applications. Table 2 provides a summary of the extensive technology assessment conducted by the research team on these carriers and their LBS capabilities.

Table 2 - Summary of LBS Capabilities by Cellular Carrier

Cellular Carrier	Handset-Initiated Location Request				Network-Initiated Location Request	
	Support HI?	Restricts Access HI?	Supports J2ME?	Embedded GPS exposed?	Supports NI?	Restricts Access NI?
Sprint (CDMA)	Yes	Yes	Yes	Yes	Yes	Yes
	<u>Note:</u> Quickly maturing LBS platform for HI and NI. All devices released after early 2007 support JSR179 for embedded GPS. Developer website: http://developer.sprint.com					
Nextel (iDEN)	Yes	No	Yes	Yes	Yes	Yes
	<u>Note:</u> Mature LBS platform for HI and NI. All devices released after 2004 capable of JSR179 support for embedded GPS. Developer website: http://developer.sprint.com					
AT&T (formerly Cingular)	Yes	Yes	Yes	No*	Yes	Yes
	<u>Note:</u> LBS still emerging and still closed to 3rd party developers. Only devices supporting JSR179 with embedded GPS are PDAs Developer website: http://developer.att.com/					
Verizon	Yes	Yes	No	Yes	No	N/A
	<u>Note:</u> LBS still closed to 3rd party developers. BREW Only for applications. Only RIM Blackberry, Palm Treo, and Windows Mobile devices w/ external Bluetooth support 3rd party LBS applications. Developer website: http://www.vzwdevelopers.com					
Alltel	Yes	Yes	No	Yes	No	N/A
	<u>Note:</u> Only BREW LBS applications or RIM Blackberry devices are supported for 3rd party software development Developer website: http://developer.alltel.com					
T-Mobile	No	N/A	Yes	No*	No	N/A
	<u>Note:</u> LBS still closed to 3rd party developers. No direct developer support. Only RIM Blackberry 8800 has embedded GPS Developer website: N/A					

HI = Handset-initiated location requests by 3rd party applications

NI = Network-initiated location requests by 3rd party applications

(*) Unlocked GSM devices w/ embedded autonomous GPS may be available through other vendors outside of the U.S. market

All information current as of 11/16/2007

Capabilities of cellular providers have progressed rapidly over the past few years. A general evaluation of cellular networks' features is shown in Table 3. Each U.S. cellular carrier is at a different stage of network deployment. A comparison of the various cellular network technologies deployed by the major U.S. cellular carriers is shown in Table 4.

Table 3 - Cellular Data Networks

	1G	2G	2.5G	3G	3.5G	4G AND BEYOND
Technology	AMPS	GSM, CDMA, iDEN	GPRS, 1xRTT, EDGE	UMTS, 1xEV-DO	HSDPA (upgrade for UMTS), 1xEV-DV	WiMax
Speeds	n/a	Less than 20Kbps	30Kbps to 90Kbps	144Kbps to 2Mbps	384Kbps to 14.4Mbps	100Mbps to 1Gbps
Features for single user	Analog (voice only)	Voice; SMS; conference calls; caller ID; push to talk	MMS; image; Web browsing; short audio/video clips; games, applications, and ring tone downloads	Full-motion video; streaming music, 3D gaming; faster Web browsing	On-demand video; video conferencing	High-quality streaming video; high-quality video conferencing; Voice-over-IP telephony

When discussing handset-initiated location requests and software running on handheld devices, the focus is on traditional mobile phones owned by the majority of consumer wireless subscribers. Larger PDA devices such as Palm Treos, RIM Blackberries, and devices running Windows Mobile platforms are excluded from this group of devices because they represent a smaller segment of the wireless device market and are much more expensive than the widely-used cell phones. Proprietary development environments are also necessary for Palm, RIM, and Windows Mobile platforms (30), (31). RIM Blackberry and Palm devices support J2ME applications through some modifications in their respective proprietary IDEs. Currently, Windows Mobile does not support the J2ME platform. This research project focused primarily on devices that support the J2ME platform and can easily run J2ME applications developed using open IDEs for Java such as Netbeans or Eclipse (32), (33).

To facilitate development of J2ME applications, Sun Microsystems provides a generic software J2ME emulator known as the "Sun Java Wireless Toolkit for CLDC" that runs MIDP and CLDC-compliant applications in a desktop environment (34). Support for additional APIs defined by JSRs is included in the emulators allowing software developers to develop and test J2ME applications without necessarily installing them on an actual device. There is no substitution for field testing any application, particularly those with LBS. Emulators do provide a simple method of testing the basic functionality of a J2ME application (also known as a "MIDlet").

Table 4 - Cellular Network Technologies by Carrier

Cellular Data Networks for Mobile Phones						
Cellular Carrier	Network Type	Network Classification	Avg. Download Speed	Burst Download Speed	Avg. Upload Speed	Burst Upload Speed
Alltel	CDMA (1xRTT)	2G	40-70Kbps	144Kbps	~40-70Kbps	~144Kbps
	http://www.alltel.com/business/enhanced/mobilelink.html					
	EV-DO	3G	400-700Kbps	2.4MB	~40-60Kbps	100Kbps
	http://www.alltel.com/business/enhanced/mobilelink.html					
AT&T Wireless (formerly Cingular)	GPRS	2.5G	~25-35Kbps	~90Kbps	~25-35Kbps	~90Kbps
	http://www.cingular.com/sbusiness/data_connect					
	EDGE	2.5G	70-135Kbps	200Kbps	~70-135Kbps	~200Kbps
	http://www.cingular.com/sbusiness/data_connect					
	UMTS/HSDPA (in deployment)	3G	400-700Kbps	~1.8Mbps	~220-320Kbps	~384Kbps
Nextel	iDEN	2G	56Kbps	56Kbps	56Kbps	56Kbps
	http://coverage.sprintpcs.com/IMPACT.jsp?PCode=vanity:coverage					
Sprint (now Sprint-Nextel)	CDMA (1xRTT)	2G	60-80Kbps	144Kbps	~60-80Kbps	~144Kbps
	http://coverage.sprintpcs.com/IMPACT.jsp?PCode=vanity:coverage					
	EV-DO	3G	400-700Kbps	2MB	40-70Kbps	~100Kbps
	EV-DO Rev. A	3G	600Kbps-1.4Mbps	~3.1Mbps	350Kbps-500Kbps	~1.8Mbps
	http://coverage.sprintpcs.com/IMPACT.jsp?PCode=vanity:coverage					
T-Mobile	GPRS	2.5G	20-35Kbps	~56Kbps	~20-35Kbps	~56Kbps
	http://www.t-mobile.com/plans/default.asp?tab=internet					
	UMTS/HSDPA (status unknown)	3G	~400-700Kbps	~1.8Mbps	~220-320Kbps	~384Kbps
Verizon Wireless	http://www.t-mobile.com/plans/default.asp?tab=internet					
	CDMA (1xRTT)	2G	60-80Kbps	144Kbps	~60-80Kbps	~144Kbps
	http://b2b.vzw.com/broadband/coveragearea.html					
	EV-DO	3G	400-700Kbps	2MB	40-60Kbps	100Kbps
	EV-DO Rev. A	3G	600-1.4Mbps	~3.1Mbps	500-800Kbps	~1.8Mbps
Verizon Wireless	http://b2b.vzw.com/broadband/coveragearea.html					

~ = Approximate speed based on network technology, actual speed was not documented by carrier

Sprint-Nextel iDEN Network

The Sprint-Nextel iDEN network (formerly Nextel iDEN network) was the first commercial cellular network in the U.S. to expose location-based services capabilities to third party application developers. Additionally, as of November 2007, Sprint-Nextel's iDEN network was the only cellular network to allow free and open access for handset-initiated location requests on devices with embedded GPS (Note: this does not include international GSM devices with autonomous GPS that do not require network resources, since these devices can be used on U.S. GSM networks without carrier knowledge). Any software developer can create and test J2ME applications compliant with MIDP 2.0 and CLDC 1.1 specifications on Motorola iDEN handsets without restriction or required permission. Since the release of the Motorola i88, all Motorola devices that use the iDEN network for both voice and data services are capable of running J2ME mobile applications that can request position information from the device. Older Motorola iDEN models prior to the i860 that came out in 2004 only support the Motorola OEM Position API accessed through J2ME, while all Motorola iDEN models released from the i860 up to and including the most recent i880 iDEN model also support the standardized JSR179 Location API. Custom software emulators that execute on a PC and simulate behavior for that particular model are available from Motorola (<http://developer.motorola.com/>). These emulators also allow a text file of GPS positions to be loaded into SDK that will then be returned by the JSR179 when the application makes a function call to this API inside the emulator. This enables limited testing for location-aware applications. However, there is no substitution for testing with an actual phone since emulator behavior often differs from the actual device in subtle but important ways. For example, only on the real phone are certain security restrictions enforced, which means that certain applications that work correctly on emulators may fail on actual devices due to incorrectly configured security settings. Also, an application only has to deal with the uncertainty of real GPS positioning on the real device. This uncertainty refers both to the availability of GPS in different environments as well as the accuracy of a fix returned at a particular time and place under particular environmental conditions. Developing an application that works reliably under these conditions is a difficult task that requires testing in the real world with real devices.

Nextel phones utilize an A-GPS method as well as a Cell-ID method for position determination (35). Either method can be requested specifically by the application through input to the JSR179 API. Under normal operation, if a highly-accurate A-GPS fix is requested but a position cannot be determined using GPS, a Cell-ID location information will be returned instead. For the Nextel iDEN network, the actual physical geographic location of the cell phone tower is returned instead of the position of the center of the serving cell, which can potentially differ depending on the area of coverage in relation to the tower position. Obtaining GPS without assistance information, or autonomous GPS, is also possible through direct input to the API, although there is no automatic failover to autonomous GPS if assistance data is not

obtainable from the network (i.e., the JSR179 implementation will return Cell-ID information if assistance data for A-GPS is not obtainable, so the application itself must specifically request autonomous GPS if assisted GPS is not possible.) Positioning is possible when the iDEN network is not available by using autonomous GPS.

Networking support through Java is also important for location-aware applications so they can transfer location and other data back to a server. While the J2ME MIDP 2.0 specification requires only HTTP to be supported so that a device is certified as MIDP 2.0-compliant, iDEN phones support Transmission Control Protocol (TCP), Secure Sockets Layer (SSL) (secure version of TCP), User Datagram Protocol (UDP), Hypertext Transfer Protocol (HTTP), and Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) through J2ME, thus giving software developers many options to choose the protocol that best fits their application (36), (37), (38). A compatible SSL digital certificate must be purchased from Verisign for an annual fee of \$499 to enable secure communications between the cell phone and server via SSL or HTTPS. For data communication initiated by a server (i.e., “push” communication to the phone), server sockets are available on the phone for applications to listen to incoming communications on a particular port. Many iDEN phones released since the i870 support the JSR172 J2ME Web Services Specification API, which allows the device to directly access and utilize web services using Simple Object Access Protocol and XML to enable distributed applications (39), (40). The Sprint-Nextel iDEN network exhibits data speeds for upload and download at approximately 56Kbps. A high-speed version called “WiDEN,” available on certain model iDEN phones, is used by the phone when extra data channels are available and can potentially quadruple packet data speeds (41). As with most cellular providers that require the full use of the communication channel for voice when a call is active (i.e., Voice-Over-IP is not yet utilized), a data and voice session cannot exist simultaneously. Therefore, data will not be sent to the server if the user is on a phone call. Also, if data is being sent very frequently to the server (i.e., a location fix every four seconds), incoming calls to the cell phone may be blocked and may go directly to voicemail. Therefore, the application executing on the cell phone must intelligently manage GPS data transmissions to servers in order to avoid interfering with normal cell phone voice communications. In the near future, as cell phone carriers move towards IP Multimedia Subsystems (IMS) and IP-based voice communications, this limitation will disappear enabling voice and data sessions to co-exist.

For position data to be transferred from the cell phone to a server, a data plan is required. An unlimited data plan is preferred so there is an upper limit on the cost incurred by the subscriber. An unlimited data plan can be added to any Nextel iDEN plan for \$10 per month.

Applications using handset-initiated location requests also provide the end-user ultimate control over their privacy. When users exit the J2ME application running on their phone, they are no longer tracked because the application will not be able to access any Location APIs or transfer or store any location data. This allows users to

limit tracking near sensitive locations or at sensitive times while still retaining full use of their mobile phone's other features.

Although LBS applications can be openly developed for Motorola iDEN handsets, there exist deployment limitations. For example, applications utilizing protected APIs such as the JSR179 Location API require that the application be signed in order to run on the actual phone. The Motorola iDEN Software Development Kits (SDKs) can be used to temporarily sign the application, but these applications signed with test certificates will only last 48 hours before the application expires and cannot be started again. In order for an application to work indefinitely after installation, it must be signed with a Motorola production certificate, which is only available after the application has been submitted to and approved by Motorola and Sprint-Nextel. This process can be started by submitting an application on the Sprint-Nextel Developer Website, <http://developer.sprint.com>. Approval is usually reserved for Sprint-Nextel commercial partners.

Another limitation is that applications cannot be distributed to iDEN phones via Over the Air (OTA) installation. OTA is commonly used to download a J2ME application, which takes the form of two files (.jar and .jad files), from a web page to a cell phone. On phones that are OTA-compatible, the installation process will automatically launch and the application will be installed to the phone. However, Motorola iDEN phones do not support OTA, so alternate means must be utilized in order to download an application to a phone. During development, a USB cable can be physically attached to the phone and development PC. The application can be loaded to the phone using "WebJAL," a Motorola application. For remote installations, Sprint-Nextel provides a Mobile Application Manager (MAM) website that allows customers to download applications to cell phones on their own subscription plans. However, special permission is required from Sprint-Nextel as a partner status in order to make applications downloadable on cell phones that are not on the developer's subscription plan.

One of the major limitations to utilizing location-aware applications on the Sprint-Nextel iDEN network is a handset manufacturing shift to utilize hybrid iDEN/CDMA phones instead of iDEN-only phones. iDEN-only phones (the last models being the Motorola i880 released in the last quarter of 2006 and Motorola i335 released in the summer of 2007) utilize the iDEN network for voice, data, and push-to-talk (a.k.a. PTT, "Direct Connect," or "walkie-talkie") services. After the Sprint-Nextel merger, the Nextel iDEN network has been slowly rebranded and phased out for voice and data services in favor of the faster Sprint CDMA network. However, since the Nextel iDEN network was originally optimized for short, bursty traffic, it handles PTT traffic well. Therefore, the term "Nextel Direct Connect" has recently been changed to represent Sprint's product portfolio of PTT services regardless of the network or platform, instead of the iDEN network to which it previously referred (42). As a result, all new "Nextel Direct Connect" phones released after the i880 (with the single exception of the i335) are hybrid phones that contain a CDMA chipset used for voice and data services and an

iDEN chipset used only for PTT. These models include the Motorola ic502, ic602, and ic902. New capabilities, discussed in detail in the following Sprint CDMA Network section, will be possible for applications on the CDMA network that previously were not available on the iDEN network. However, this transition creates issues for application development. Since the Qualcomm CDMA chipset governs the operating system, J2ME virtual machine, and application environment for these phones instead of the Motorola iDEN chipset, they are not compatible with any Motorola iDEN proprietary APIs, including the Motorola OEM Position API.

A bigger issue for location-aware software development on these phones is that they are now subject to the same application development restrictions as traditional CDMA phones on the Sprint CDMA network. These restrictions are discussed in detail in the following section as well. If the current trend continues, Motorola pure iDEN handsets will only be supported as legacy devices, and application developers will focus new application development towards the Sprint CDMA platform. Sprint-Nextel has only committed to maintaining the Nextel iDEN network until 2010, so it is possible that all voice, data, and PTT services will eventually be transitioned to the Sprint CDMA network so Sprint-Nextel can focus on maintaining a single CDMA network instead of the iDEN network in addition to the CDMA network. On January 30, 2008, Sprint announced a commitment to release additional handsets that “utilize the (iDEN) network’s unique capabilities” (43). It remains to be seen whether this means the release of additional pure iDEN handsets or hybrid iDEN/CDMA handsets that use iDEN only for PTT.

Sprint-Nextel provides means for network-initiated location requests to devices on the Nextel iDEN network referred to as the “Wireless GPS Platform” (44). However, access is limited to software developers who are Sprint-Nextel partners as handsets must be provisioned to work with the service. Additionally, this service is mostly directed towards enterprise customers that wish to track their own employees, not towards commercial applications that would track customers on their own personal subscription. Commercial subscribers can sign up for applications via the Wireless GPS Platform at the sprint.com website and must specifically opt-in for the application. The technical implementation accessing the Wireless GPS platform is based on the Open Mobile Alliance (OMA) Location Work Group’s Mobile Location Protocol (MLP) (28). Upon a network-initiated location request by a web application, the network’s location server will first try to obtain a GPS fix from the handset and pass this back to the web application. If GPS is not available, Cell-ID information will be passed back to the web application. Frequency is limited to one location request every 10 minutes per handset, which limits the usefulness of network-initiated requests for real-time applications such as navigation or high-precision tracking. Also, if the device does not have iDEN network coverage, location data will not be returned to the web application. Another significant limitation for privacy concerns is that there is no way the user can opt-out of being tracked temporarily from the handset, as they must visit the Sprint.com website and opt-out of the service in order to prevent being tracked. This characteristic of the network-initiated platform does not allow the user to specifically

control when and where they are being tracked unless the user turns off the mobile phone entirely. Another limitation to using the network-initiated platform is amount of data that is returned in each location fix. Only a timestamp, latitude, longitude, and an estimated accuracy of the location fix are returned to the web application. Additional information that may be important to some applications such as speed and heading are not available through this server-side API.

Despite the many limitations to the Wireless GPS Platform, several advantages over handset-initiated location requests are noted. For certain applications that require infrequent location updates, occasional latitude and longitude position information from these network-initiated location requests may be adequate. Since no third party J2ME application is required to be running on the handset and frequency of position requests is limited, there is a minimal impact on battery life for this service. There is also no distribution and maintenance of handset-based software since no J2ME application is required on the handset. Cost for the Wireless GPS Platform is approximately \$10 per month per handset and would be bundled into the monthly fee for the application paid by the end-user.

Sprint-Nextel CDMA Network

Sprint-Nextel's CDMA network (formerly Sprint PCS CDMA network) is second in the U.S. only to the iDEN network in terms of longevity of LBS for consumers. In the past, Sprint CDMA devices have not had certain advanced software capabilities such as support for JSR179 and multitasking Java virtual machines that were common on iDEN devices. However, handsets exhibiting these features released in early 2007 for the Sprint CDMA network promise to push LBS to the next level on this particular platform and even surpass iDEN LBS capabilities.

Sprint-Nextel also provides a variety of resources that support J2ME application development. The "Sprint Mobility IDE" is available as a J2ME development environment, which is Sprint-branded version of the Netbeans IDE with some Sprint-specific sample applications added. Sprint also provides a "Sprint Wireless Toolkit," which is a Sprint-branded version of the Sun Wireless Toolkit that includes emulators for devices on the Sprint network as well as support for J2ME APIs proprietary to Sprint devices called "Sprint Extensions." Both the Sprint Mobile IDE and Sprint Wireless Toolkit are available to download at <http://developer.sprint.com>. This website also includes a list of all Sprint devices and their ability to support particular features and APIs. Before an application utilizing JSR179 can be installed and tested on a device on the Sprint network, it must be signed using an authorized code-signing digital certificate. A Java Class 3 code-signing certificate from Verisign can be purchased with an annual cost of \$499. In addition, the phone that will execute the application must have its developer root enabled, which can be accomplished via a process on Sprint Developer's website (45). Applications to be installed on mobile phones that do not have the developer root enabled (i.e., the general populations' cell phones) must be

signed by Sprint's own digital certificate, which can only be accomplished through a business relationship with Sprint and an application certification process.

Since the Sprint CDMA network utilizes devices from multiple manufacturers such as Sanyo, Motorola, Samsung, and LG, it is more difficult to give generalizations of development characteristics these devices have than those on the iDEN network, which are from the single vendor Motorola. Therefore, it is best to consult the Sprint developer website to confirm the specifications of particular devices. However, the following features hold true for the majority of CDMA cell phones on the Sprint network.

Sprint CDMA devices support CLDC 1.1 and MIDP 2.0 compliant J2ME applications. All cell phone models released since the Sanyo 7050 in early 2007 have shown increasing levels of LBS support that rival iDEN handsets, including capabilities for simultaneous execution of J2ME applications referred to as Multi-Tasking Virtual Machines (MVM) and support of JSR179 Location API 1.0. MVM devices allow J2ME application to continue executing in the background while minimized. Usually, an application will monitor the location of a device then take action when location-related events occur such as, for example, transmitting that location to a server for a tracking application. That is the reason this feature is important to many LBS applications such as TRAC-IT. It appears that all Sprint CDMA devices released after early 2007 will support both JSR179 and MVM. Many devices released prior to the Sanyo 7050 do not have a MVM. In this case, the application can only perform actions such as retrieving position information from a Location API, only when the application is maximized and in the foreground on the device. As soon as the user minimizes the application to run in the background, all activity in that application is suspended. Prior to JSR179 Location API support, several Sprint handsets were supported by the proprietary Qualcomm Java Application Extensions (QJAE) API in order to expose basic location capabilities to J2ME applications (46). Newer handsets such as the Sanyo 7050 support both location APIs. It is preferred, however, to utilize JSR179 Location API since it is standardized.

Sprint exposes multiple types of location fixes to software running on the mobile phone through JSR179 including CellID, Advanced Forward Link Trilateration (AFLT), Assisted GPS, and Autonomous GPS (47). Cell-ID, also referred to as "Cell Sector," on the Sprint-Nextel CDMA network is defined as the center of the area of cellular coverage that is currently serving the device. The geographic location of the cell tower that the phone is currently communicating with is also exposed to applications through proprietary Sprint extensions. All positioning technologies are enabled by the gpsOne chipset in Qualcomm CDMA devices (46). Final location calculations can take place on the network (MS-assisted) or on the device (MS-based), providing the developer with several options to optimize location-aware application performance. MS-based fixes return the location of the handset the fastest for Assisted GPS calculations and therefore is the most commonly used method for high-precision location-aware applications. Sprint CDMA devices must be within Sprint CDMA network coverage in order to obtain a first location fix since the gpsOne system requires communication

with a positioning server to retrieve assist data. Location fixes can be obtained if cellular service is only lost temporarily and assist data stored by the handset is still current.

Data service, including TCP, SSL (secure version of TCP), UDP, HTTP, and HTTPS, is also exposed to J2ME applications on the handset. A compatible SSL digital certificate must be purchased from Verisign for an annual fee of \$499 in to enable secure communications between the cell phone and server via SSL or HTTPS. A digital certificate can be used for many applications in a single company. For data communication initiated by a server (i.e., "push" communication to the phone), server sockets are available on the phone for applications to listen to incoming communications on a particular port. However, while data can be operable when roaming on different CDMA networks, data services are not guaranteed while roaming. The Sprint CDMA network currently has two primary levels of network speed for mobile phones: 1xRTT and EV-DO. 1xRTT data service has been around for several years making it available on all current Sprint CDMA devices. 1xRTT has data speeds around 50Kbps-70Kbps for both download and upload. EV-DO service is available on newer Sprint devices, especially those that feature streaming music and video services, and has speeds averaging 400Kbps – 700 Kbps with peak rates up to 2Mbps for downloads and 40Kbps-70Kbps on average for uploads with peak rates up to 144Kbps. EV-DO Rev. A which is an enhancement to EV-DO is currently available for laptop data cards but is expected to be available in mobile phones in the near future. EV-DO Rev. A boasts speeds on average from 450-800Kbps with peak rates up to 3.1 Mbps for downloads and 300-400 Kbps on average for uploads with peak rates up to 1.8 Mbps (48).

As with most cellular providers today that require the full use of the communication channel for voice when a call is active (i.e., Voice-Over-IP is not yet utilized), a data and voice session cannot exist simultaneously. Therefore, if the user is on a phone call, data will not be sent to the server. Also, if data is being sent very frequently to the server (i.e., a location fix every four seconds), incoming calls to the cell phone may be blocked and may go directly to voicemail. Therefore, the application executing on the cell phone must intelligently manage GPS data transmissions to servers in order to avoid interference with normal cell phone voice communications. As cell phone carriers move towards IP Multimedia Subsystems (IMS) and IP-based voice communication over the next few years, this limitation will disappear and voice and data sessions will be able to coexist.

For position data to be transferred from the cell phone to a server, a data plan is required. An unlimited data plan is preferred so there is an upper limit on the cost incurred by the subscriber. An unlimited data plan can be added to any Sprint CDMA plan for \$15 per month.

Sprint supports network-initiated location requests for CDMA devices through a product referred to as the Business Mobility Framework (BMF). Web applications can query the

BMF using Parlay and Parlay X protocols for the location of a number of CDMA devices, and receive the location of each of these devices in return. GPS, GPS/AFLT hybrid, AFLT triangulation, and cell sector information can be returned, and the BMF will attempt location fixes using the technologies in that order of precision. However, many of the same restrictions apply to the BMF as the Nextel Wireless GPS Platform. BMF requests can only return the position of a handset once every 5 minutes. Additionally, the information returned is restricted to position information only and does not support speed or heading information. The primary target for BMF services is enterprise business customers, so the BMF is not meant to track consumers for commercial services. Therefore, pricing is billed based on transactions at the account level. Also, handsets are opted in and out via the Sprint website, so the mobile phone user has no control over when they are being tracked. The CDMA device must be within CDMA network coverage in order to be tracked. The same advantages apply to the Sprint BMF as the Nextel Wireless GPS Platform. No application needs to be installed or maintained on the handset. The BMF may be sufficient for applications that only require periodic location information. However, for many applications that require frequent position updates and more control over tracking the user, such as TRAC-IT, handset-initiated location requests are preferred over the BMF.

Access to handset-initiated location requests on Sprint CDMA handsets as well as access to the BMF is currently restricted to members of the Sprint Professional Developer Program due to privacy concerns as well as the additional processing load that handset-initiated location requests can have on Sprint network resources. The research team was granted access to this technology through the Sprint Application Developer's Program. If other application developers want to partner with Sprint and gain access to these services, they should contact Sprint through their developer website at <http://developer.sprint.com>.

AT&T

AT&T, formerly Cingular, is a GSM U.S. cellular provider. AT&T announced its plans to support location-based services to third party application developers in early 2007 (48). As of early 2008, AT&T's network is still closed to application developers that want to develop location-based services for commercially available consumer-level mobile phones. Multiple requests were made to AT&T to supply additional information about their LBS platform and to allow access for prototype TRAC-IT application testing, but no response was received by the research team. Information available on the AT&T website indicates that AT&T will first focus on enterprise customers. Commercial LBS applications for the general public may be supported at a later date. Positioning technologies planned to be supported within the AT&T platform include Cell-ID, Enhanced Cell-ID (i.e., Cell-ID with Timing advance), assisted GPS, and autonomous GPS (49). However, other than the Windows Mobile and RIM Blackberry devices, as of January 2008, there are no devices with embedded GPS available on the AT&T network that exposes access to GPS information in J2ME applications. Instead, an external GPS device must be tethered to the cell phone using the Bluetooth™ protocol.

GPS data from this external receiver can be accessed by J2ME applications on devices that support JSR179. A list of devices that support JSR179 and external GPS units via Bluetooth™ is available on AT&T's website at <http://developer.att.com>. Future support is planned for network-initiated requests and handset-initiated requests for devices with embedded GPS, including support of JSR179 and JSR293, but no timeline has been announced. In 2007, AT&T improved their application developer website significantly after their purchase of Cingular. Hopefully, more LBS support on the AT&T network will become available in the near future.

It should be noted that it is possible to use unlocked GSM devices (i.e., international cell phones not locked to a U.S. cellular carrier) on AT&T's network. Access to autonomous GPS information from J2ME applications may be possible on these devices and is dictated by the device manufacturer's specifications. However, these unlocked GSM devices are rare, expensive, and not officially supported by AT&T. No network dependant location features, including assisted GPS, Cell-ID, or cell signal trilateration, would function on these devices while they operate on a U.S. cellular carrier network.

Alltel

Alltel is a U.S. CDMA cellular carrier. It uses devices with Qualcomm chipsets that run the BREW platform for the majority of their commercially-available mobile phones. Alltel provides few developer resources outside of Qualcomm's BREW resources, and therefore the application developer is restricted to the BREW operating system for application development. BREW is not capable of running multithreaded applications, but does support cooperative multitasking (50). However, the absence of multithreading and J2ME severely limits the possibilities of producing an effective, portable LBS tracking application for consumers on the Alltel network. The research team contacted Alltel, through their developer website at <http://developer.alltel.com>, several times for additional information related to LBS on the Alltel network, but did not receive a reply. The fact that Sprint, also a CDMA carrier utilizing Qualcomm technology, has significant support for location-based J2ME applications on its platform indicates that it is possible for Alltel to support similar applications if desired. However, as of January 2008, Alltel is closed to any such applications.

It should be noted that RIM Blackberry, Windows Mobile, and Palm OS devices are not included in this grouping since they are not BREW OS devices, and the specifications of these devices will dictate their capabilities.

Verizon Wireless

Verizon Wireless is also a U.S. CDMA cellular carrier that primarily uses devices with Qualcomm chipsets that run the BREW platform. Verizon Wireless also provides few resources on their website (<http://www.vzwdevelopers.com/>) outside of Qualcomm's BREW resources. As previously stated, the absence of multithreading and J2ME severely limits the possibilities of producing an effective, portable LBS tracking

application for consumers on the Verizon Wireless network. The fact that Sprint, also a CDMA carrier utilizing Qualcomm technology, has significant support for location-based J2ME applications on its platform indicates that it is possible for Verizon Wireless, if desired, to support similar applications. However, as of January 2008, Verizon Wireless is closed to any such applications. The research team requested more information from Verizon Wireless on LBS as well as the ability to test LBS applications on their network but never received this additional information.

It should be noted that RIM Blackberry, Windows Mobile, and Palm OS devices are not included in this grouping since they are not BREW OS devices, and the specifications of these devices will dictate their capabilities. As of January 2008, all devices on these platforms on the Verizon Wireless network require an external GPS unit with a Bluetooth™ interface.

In late 2007 Verizon announced that by late 2008 it will allow open access to any device and application on its network (51). However, the meaning of “open access” is debatable and has not yet been clarified by Verizon. Therefore, at this time it is unknown how this announcement will affect application developers and their ability to produce applications for Verizon Wireless customers (52).

T-Mobile

T-Mobile, a GSM U.S. cellular provider, provides no developer support on their current developer website at <http://developer.t-mobile.com/>. Therefore, developers must consult individual device manufacturers regarding development on their particular platform.

As with AT&T, it is possible to use unlocked GSM devices (i.e. international cell phones not locked to a U.S. cellular carrier) on T-Mobile’s network. Access to autonomous GPS information from J2ME applications is possible on these devices and is dictated by the device manufacturer’s specifications. However, these unlocked GSM devices are rare, expensive, and not officially supported by T-Mobile. Network dependant location features, including assisted GPS, Cell-ID, or cell signal trilateration, will not function on these devices while operating on U.S. cellular carrier network.

Chapter 3. TRAC-IT System Design

This chapter presents the implications of technology assessment on selecting the design criteria for the TRAC-IT system development. The assessment shaped the selection process of the type of location tracking technology, the methods to access that technology, the design, and the features of the TRAC-IT software for the mobile phone. The following is a brief description of the different technologies selected by the research team as the best fit for the TRAC-IT system:

1. Positioning Technology = Assisted GPS as primary, AFLT and Cell-ID as secondary
2. Form-Factor = Commercially-available GPS-enabled mobile phones should be used as the all-in-one data collection device
3. Method to retrieve location data = Handset initiated Location Requests
4. Portability = J2ME as development platform, with JSR179 Location API for access to position information
5. Coexistence of TRAC-IT with mobile phone functionality = J2ME platforms must support a MVM. Intelligent network use techniques must be implemented in mobile TRAC-IT software. The TRAC-IT user should have an unlimited data plan. Smart power management techniques must be implemented in TRAC-IT software to avoid severe impact on battery resources. UDP was selected as the data transfer protocol for location updates due to its lightweight design.
6. Interoperability and Forwards-Compatibility = Java Enterprise Edition 5 (Java EE5) was chosen as the primary development platform for server-side components, with server-side functions being exposed via Web Services for universal access. Within J2ME, CLDC 1.1 (lightweight version of Connected Device Configuration (CDC)), MIDP 2.0 (predecessor to MIDP 3.0), JSR179 Location API (predecessor to JSR293 Location API 2.0) were targeted for development.

The following sections detail the selection process for important TRAC-IT components and the determination of certain TRAC-IT application features.

3.1 Positioning Technologies

With many different types of positioning technologies that can locate the position of a mobile phone, it was important to select a primary positioning technology that will best recreate the transportation patterns of the user. Assisted GPS was selected as the primary positioning technology due to its high accuracy and ability to generate updates

as frequently as once per second on mobile phones. Additionally, GPS has been used in past travel behavior studies with excellent results (11). Secondary positioning technologies, including cell signal triangulation techniques and Cell-ID, can be used when GPS is not available though it is less accurate than GPS. These positioning methods will give a rough position of the user when they are inside a building or outside of GPS coverage.

3.2 Form-factor

Commercially-available GPS-enabled mobile phones provide a compact form factor that includes GPS hardware, wireless communications, and a user interface to receive additional survey input from the user. Compatible devices may be already owned by the general public, thus decreasing potential TRAC-IT deployment costs and requiring only the distribution of a software application.

3.3 Network-Initiated vs. Handset Initiated Location Requests

To target the general population of the United States with a software application that can record a mobile phone users' travel behavior, issues of user privacy and portability across multiple cellular carriers and devices are critical. Also critical are the integrity of the collected data and the control of the method of data collection by the software application. The importance of these issues eliminates the possibility of utilizing network-initiated location requests to record the travel behavior of the general public. The target market for network-initiated location requests is primarily enterprise customers that query the location of devices they own. Therefore, it would be labor intensive for cellular providers to provision large numbers of private consumer phones for access by a server-side software application. Therefore, it appears that cellular providers would not likely consider allowing network-initiated location requests for a large-scale travel survey of the public's mobile phones. Privacy concerns also exist for tracking methods utilizing network-initiated location requests. Once a user opts in to be tracked by network-initiated location requests, the only method available to the user to disable tracking is to turn off the cellular phone or completely opt out of the program via a website. Otherwise, the user's cell phone would be tracked 24 hours a day. This method does not allow the user to control when they are being tracked from their cell phone so they can turn off the tracking feature when nearing sensitive areas such as medical offices or their home. Also, the user may not want certain trips to be recorded by the tracking application and therefore should have the ability to either opt in or out of being tracked at any time from their cell phone without interfering with their normal cell phone operations or their enrollment as a participant in the survey.

From a technical perspective, although standards such as Parlay X and MLP exist, there is currently no consistent implementation among cellular networks for network-initiated location requests. As a result, a different web application would have to be developed for each cellular network in order to request the position of mobile phones on that particular network. The scalability of cellular network resources for such an approach

to a national population sample size would also be questionable. The responsibility of adding system capacity to handle a large number of network-initiated location requests would lie with each cellular provider, which may not have incentive to add such capacity. Last, periodic updates with a maximum frequency of an update once every 5 to 10 minutes is not sufficient to accurately recreate a user's travel behavior even if a GPS fix is returned on every update. Full information for a GPS fix is not returned, only the position (i.e., latitude and longitude) and timestamp. Velocity and heading and other GPS information are not accessible. Also, there is no absolute control over the positioning method used to determine the cell phone's location. Therefore, without guarantee that a GPS fix will be returned for every update, long periods of time (i.e., 10 or more minutes) may pass without a position update. Therefore, it is unlikely that a user's travel path, including starting and ending location could be reconstructed from such infrequent position updates. These facts eliminate network-initiated location requests from consideration as a method for obtaining travel behavior data for TRAC-IT.

Since network-initiated location requests are eliminated from consideration, handset-initiated location requests were selected as the primary means of obtaining the location data for a mobile phone. This method allows the application to specify the frequency of location updates up to a frequency of once per second as well as the positioning method used to calculate the phones location. Additionally, more data is available for each location update including speed and heading information. Another advantage of a software application that utilizes handset-initiated location requests is that it can immediately take action based on the location information provided by the phone. TRAC-IT can therefore manage the mobile phone's resources intelligently in order to maximize battery life and prevent interference with normal mobile phone operations. This also allows TRAC-IT to provide other services such as real-time driving directions or personalized traffic alerts to the user that can be an incentive for them to use the application. These services can be more advanced than typical services since they can make use of the personal travel history of the person using the cell phone. Having the TRAC-IT application reside on the user's phone also allows tight integration of the form used to collect user input such as mode, purpose, etc. with GPS tracking. Tight integration of the two features allows better resource management for phone battery power and memory. For example, when users manually indicate that they are starting or stopping a trip, the phone can immediately turn the GPS chip on or off, which saves power when they are not being tracked. Due to all of these reasons, handset-initiated location requests were selected for TRAC-IT.

3.4 Portability of the TRAC-IT Software

Portability is defined as the ability to use a software application on more than one type of hardware device or software platform with few to no changes of the software code. Since handset-initiated location requests were selected for use with TRAC-IT, a software application must be created that will run on mobile phones. Due to the many different cellular networks in the U.S., and even more cell phone manufacturers,

portability is an important issue to TRAC-IT. The mobile TRAC-IT application should be written in a programming language that has the best chance of being portable to the majority of cellular networks and mobile phones in the U.S. As of January 2008, J2ME remains the most portable programming language across different mobile devices and cellular carriers. J2ME is the only language for mobile devices that has a well-established procedure for platform updates and additions that is open to many different companies via the JCP. Many corporations have an interest in the success of J2ME and are involved in the JCP. J2ME was therefore chosen as the programming language for the TRAC-IT mobile phone application. Additionally, all APIs accessed by the J2ME application to perform various functions should be defined by JSRs when available. JSR179, and JSR293 when available, should serve as the primary software interface to access location data from the handset since this is the only standardized location-related API implemented across multiple carriers and cell phone manufacturers. Proprietary J2ME APIs may be utilized for other features when the functionality has not been standardized under a JSR (i.e., accessing the phone number of the device), but the TRAC-IT application design should abstract that particular API so that proprietary modules can easily be switched out in favor of a new proprietary module when porting the application to a new platform. This feature will ensure easy portability even when proprietary functions must be utilized to access certain features of the cell phone.

Java was also selected as the primary programming language for the server-side systems, since it is also portable across many different operating systems and application servers. It is therefore concluded that Java will enable the least expensive implementation of the TRAC-IT system since information technology departments can utilize the servers that they already have. Having both the server and cell phone software implemented in Java allows sharing of code between the two applications when possible, thus reducing development costs.

3.5 Coexistence of TRAC-IT with Cell Phone Operations

The TRAC-IT mobile application should not interfere with the cell phone's normal operation. The main areas of concern are background application execution, interfering with receiving or making phone calls, incurring additional financial cost to the individual who owns the cell phone, and significant impact on power consumption.

3.5.1 Background Application Execution

For TRAC-IT to successfully coexist with other phone functions, J2ME platforms must support MVMs so that the application can execute in the background while other phone features remain functional. It is not practical to run TRAC-IT on platforms without MVMs since the phone cannot be flipped closed and no other applications can be used or the TRAC-IT application will be paused. If paused, TRAC-IT will no longer monitor the position of the user and cannot send any position data to a server. Therefore, the presence of an MVM is critical to TRAC-IT operation as a tracking application.

3.5.2 Voice Calls

If the device supports MVM, this means that an application can be executing while the phone is idle or another data application is being used. However, current cell phone technology does not allow data and voice sessions to exist simultaneously and therefore interference with voice calls must be examined separately. The user can always make outgoing phone calls using their phone even if TRAC-IT is running in the background. However, the phone will abort any active data sessions if a voice session is initiated by the user. This would result in TRAC-IT temporarily not being able to transfer any data to the server while the user is on a phone call. As soon as the user aborts the phone call, the TRAC-IT application should be able to send data again. However, while TRAC-IT is running and in a data session (i.e., transferring location and trip data to the server), incoming phone calls may be blocked and sent to voicemail instead of ringing at the phone. It is therefore concluded that the TRAC-IT application must intelligently manage the amount of data sent to the server to avoid over-occupying the communication capabilities of the phone. The TRAC-IT application should include software to implement such functionality.

It should be noted that newer Sprint-Nextel CDMA devices released in early 2007 and all Sprint-Nextel iDEN cell phones have the capabilities of simultaneous GPS, or a separate GPS antenna and subsystem that is used for GPS reception. This allows the phone to continue calculating its position and storing it to the phone while the user is on a voice call or in a data session. In next-generation cell phones that support IMS, the limitation of mutually exclusive voice and data sessions will be removed as voice sessions are converted to packet data communication over IP, and thus be capable of simultaneous operation with other data sessions.

3.5.3 Data Plans

It is recommended that any user who will run the TRAC-IT application on his or her cell phone have an unlimited data plan. This is because TRAC-IT transfers user-entered and GPS data to a server for storage, and if the user does not have a data plan they will be charged per kilobyte for data which can be very expensive. Many cell phone providers such as Sprint-Nextel are now offering unlimited data plans for a flat monthly rate, which would be preferred to avoid incurring large costs to the user.

3.5.4 Energy Conservation

Since mobile devices are battery-powered, energy is a critical resource that must be conserved. An application cannot afford to render a device unusable due to extraordinary power consumption. High-precision location-aware applications for cell phones require that additional GPS hardware be active while the device's position is being calculated. Therefore, additional power is consumed during GPS operation beyond the normal power consumption of the phone while it is in an active or standby state. While newer high-sensitivity GPS receivers in modern mobile phones are more

power-friendly than their older counterparts the application executing on the mobile phone should reduce the number of requested GPS fixes to as few as possible to conserve power (53). Since wireless transmissions also consume power, the number of GPS fixes transmitted to the server should be reduced to a minimum as well. Intelligent monitoring software should be developed in order to calculate as few GPS fixes as possible and transmit only critical information needed to reconstruct a user's trip. Lightweight data transmission protocols such as UDP should also be used when possible in order to avoid the additional transmission overhead of "reliable" protocols such as TCP. Location fixes are calculated frequently, so the occasional loss of location data transmitted via "unreliable" protocols is not important.

3.6 Interoperability and Forwards-Compatibility of the TRAC-IT System

Since technology progresses at a rapid rate after a software solution is implemented and deployed, it is often immediately outdated. Mobile technology and cell phones, in particular, progress at an astounding rate. Therefore, forwards-compatibility of the TRAC-IT software solution created as part of this project with future cell phones was also a key component of the TRAC-IT system design. J2ME was selected as the mobile programming platform because it is the most ubiquitous computing platform for mobile devices that is not tied to a single device manufacturer or wireless carrier. Additionally, JSR293 Location API 2.0, scheduled for release in the first half of 2008, is backwards compatible with JSR179 Location API 1.0. This means that any mobile software application utilizing JSR179, including TRAC-IT, will also work with future mobile phone that implement the Location API 2.0. Since TRAC-IT is designed for CLDC 1.1 and MIDP 2.0, it should also be forwards-compatible with MIDP 3.0 when it is released, as well as with CDC devices when they become widely available (54).

To promote interoperability with all types of client devices (both Java and non-Java), all server functionality should be exposed via web services hosted on an application server. There are many open-source Java application servers, including Glassfish and JBoss (55), (56). Production Java application servers include Sun Java System Application Server, BEA WebLogic Server, and IBM WebSphere Application Server, (57), (58), (59). These servers support the TRAC-IT Server application since TRAC-IT is built on the Java EE 5 platform, which was selected as the platform for the TRAC-IT server application (60). This should allow small agencies to use free open-source application servers to host the TRAC-IT application while bigger agencies may choose to use better supported, but more costly, production application servers. The large number of supported application servers ensures that the TRAC-IT server application can be executed on an application server for years to come.

3.7 Final Selection of TRAC-IT System Components

The TRAC-IT mobile application therefore requires the following features to execute on a mobile phone:

- ♦ J2ME-compatible (CLDC 1.1 and MIDP 2.0) (Forwards compatible with MIDP 3.0) – basic Java platform for mobile devices
- ♦ Multitasking Virtual Machine – Allows background execution of the application while other phone features are being used
- ♦ UDP – supports efficient transfer of location data to the server. (Alternative version of TRAC-IT that uses TCP or HTTP could easily be developed for devices that do not support UDP, but performance will decline).
- ♦ JSR179 Location API 1.0 (forwards compatible with JSR293 Location API 2.0) – Allows programmatic access to location data on the mobile phone

As of early 2008, all mobile phones available for the Sprint-Nextel iDEN network should support the TRAC-IT application. On the Sprint CDMA network, all phones released after early 2007 should support the TRAC-IT application. It is expected that Cingular will release phones for its network in the near future that are TRAC-IT compatible, but no timeline has been announced by Cingular. As more wireless providers open their networks and devices to LBS, the ability to run applications such as TRAC-IT on all mobile phones will be possible. It should be noted that TRAC-IT has been designed to conform to standards that are widely accepted. Therefore, it is a limitation of the wireless provider that prevents TRAC-IT from operating on all cellular networks, not the TRAC-IT system. As soon as a wireless provider announces support of JSR179 applications and the other above criteria, the TRAC-IT application will instantly work on the mobile phones that are supported.

Chapter 4. TRAC-IT System Architecture

4.1 Overview of System Architecture

Figure 6 is a high-level conceptual diagram of the TRAC-IT architecture developed for Phase 3 of the research study. It is designed to allow two-way communication between the data-collection device on the user side (i.e., a cell phone) and the server-side modules, including the application server, a database and analysis software that provides feedback on user trips. The four main components of the architecture are the client-side device, the cellular communication network, the Internet, and the server-side elements, shown in Figure 7 in a block architecture diagram.

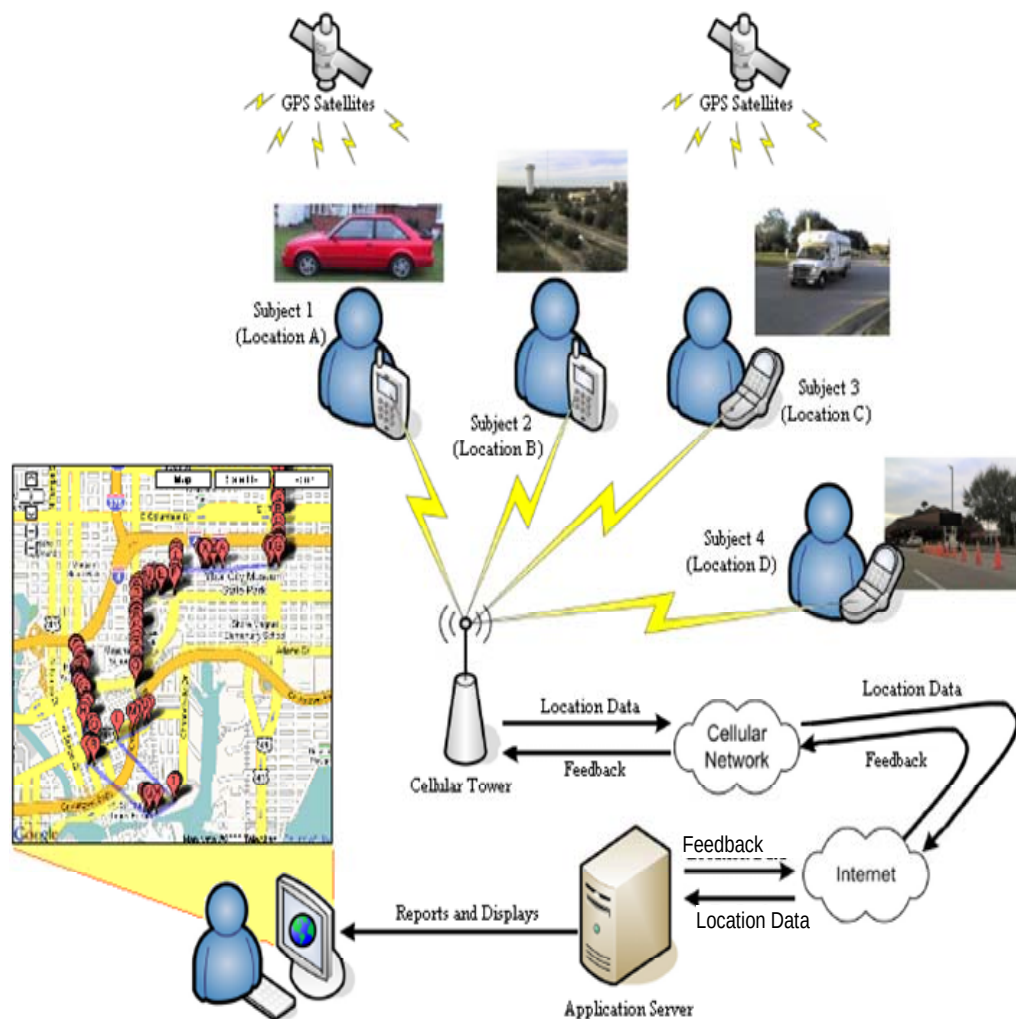


Figure 6 - TRAC-IT System Architecture

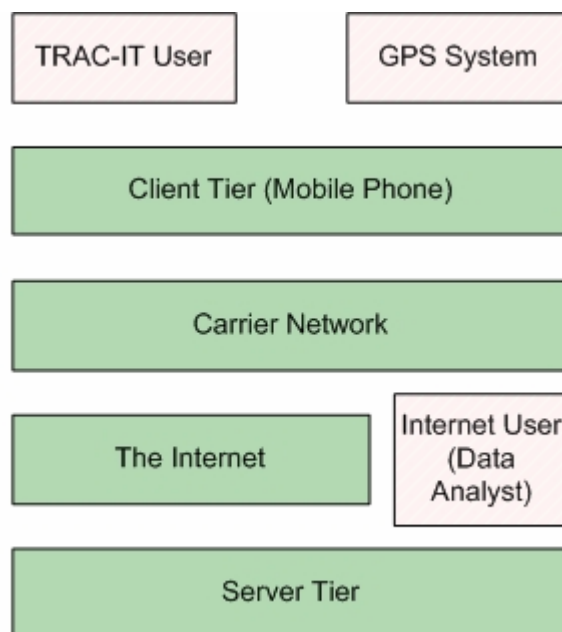


Figure 7 - Main System Components of the TRAC-IT Architecture

The GPS satellite system employed by the mobile phones with embedded GPS chips for the purpose of obtaining location data as well as the TRAC-IT participant that uses the mobile phone and the TRAC-IT analyst that view the collected data are illustrated in Figure 7.

As shown in Figure 7, the end-user device runs the client-side application software that incorporates the user interface as well as the methods for obtaining user position information. The cellular network (the second major component) is utilized as a transit network. Under the current TRAC-IT implementation users must have cellular service during the operation of the TRAC-IT software to function. The cellular network can also be used to provide user position information through cell signal triangulation or Cell-ID of the base station if the primary means of obtaining this data (i.e., the GPS satellites) is unavailable. While these data are not accurate enough to effectively track a user for analysis purposes, the information is nevertheless retained for study and can be useful to give an estimated position of the user. The Internet, the third component, is the transit network for application data between the cellular network and the server implementing the TRAC-IT service. The final element of the system architecture is the server-side component that runs the service for all users and contains the database holding user and trip data generated by the client application's execution. The various tables of the database schema are accessed during the course of TRAC-IT's operation to perform functions including:

- ♦ establishing a session,
- ♦ verifying valid application users,
- ♦ receiving and recording trip-specific data for each user,

- ♦ receiving and recording the location data elements transmitted from the end-user device during operation,
- ♦ retrieving trip data for the analysis modules that provide feedback for user trips, and
- ♦ transmitting server-side-generated feedback to the end-user device.

Figure 8 shows a high level TRAC-IT system architecture with the communication protocols that connect each module. Two primary communication protocols generated by the cell phone client software are included in the system architecture.

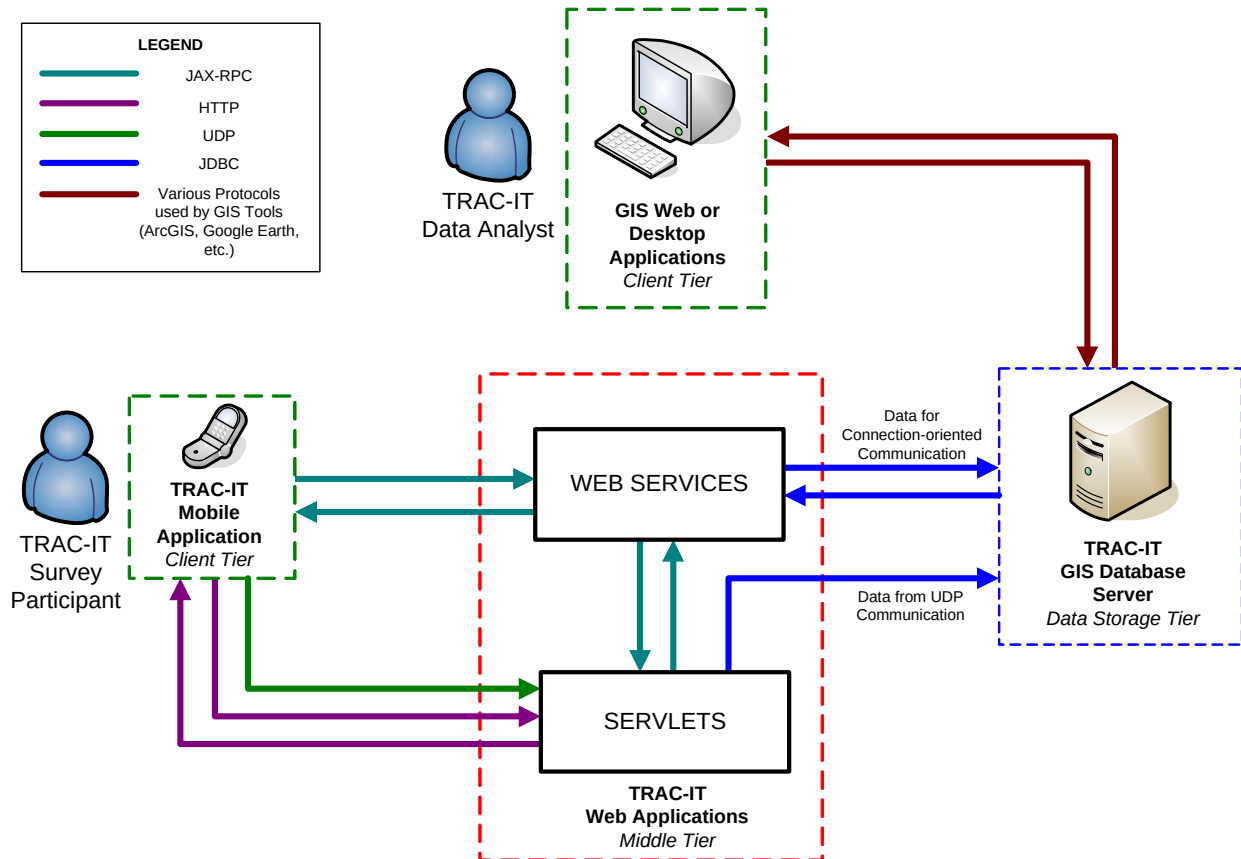


Figure 8 - TRAC-IT Architecture Showing Communication Protocols

As illustrated in Figure 9, information that is critical to the proper operation of the application, such as user log-in data, and the information that defines the start and end points of distinct user trips are transmitted directly by means of the Java API for XML-based Remote Procedure Calls (JAX-RPC) defined by JSR172 or via underlying HTTP depending on the mobile phone's capabilities, (39). HTTPS is also available for secure communications between the mobile phone and server. Since JAX-RPC is based on top of HTTP, it can also be implemented securely using HTTPS. To enable HTTPS, an SSL digital certificate must be purchased through a certificate authority such as Verisign. Therefore, HTTP and JAX-RPC utilize TCP as the underlying network

transport layer providing a reliable service with the server acknowledging receipt of the information and coordinating the retransmission of lost packets as necessary.

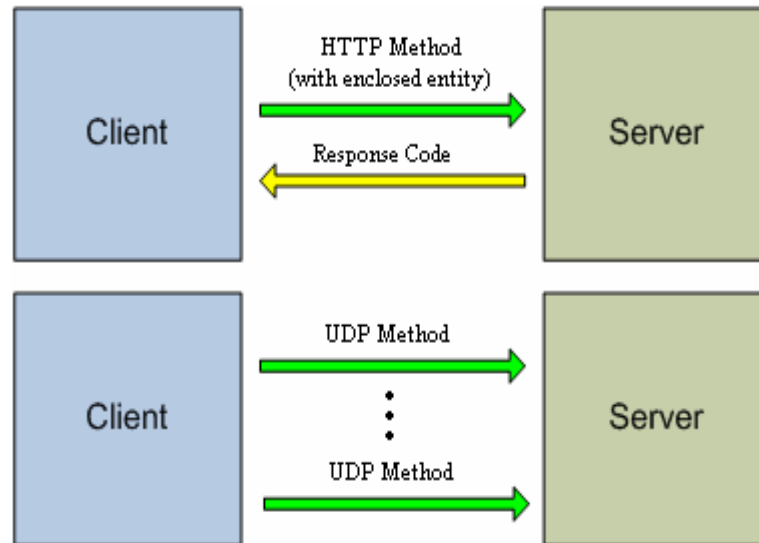


Figure 9 - The HTTP and UDP Wireless Transmission Methods

The updated location of the phone, calculated using GPS or other positioning technologies, is captured by the cell phone software application through the JSR179 Location API. Since on-board storage capabilities of a mobile phone can be very limited, the location information must be transferred to a server. The periodic location updates, for which the loss of a few packets will not necessarily compromise the integrity of the data collected, are transmitted by means of the User Datagram Protocol (UDP), which is an asynchronous (i.e., "send it and forget it") method of data transfer from the phone to the server. Utilizing UDP for this data transfer reduces the network transmission overhead for the mobile phone since no acknowledgements of received packets or retransmissions of lost packets occur. While order of packet delivery is not guaranteed using UDP, the order can be reconstructed using the timestamp of the GPS data in the packet as well as an additional timestamp made by the TRAC-IT application just prior to packet transmission. Utilizing UDP for this type of service can save important resources such as battery power. Since these location transmissions can happen as frequently as once per second, reducing any unnecessary overhead for each transmission can potentially have a significant positive impact on mobile phone performance. Figure 10 shows the interaction between the client and server components utilizing JAX-RPC (or HTTP) and UDP.

The following sections describe each of the four main system components in detail: the mobile phone, cellular network, Internet, and server. The viewpoint reflected is that of a third party application developer's perspective with certain aspects of the end-user's experience noted. A thorough description of the application's use and testing will be presented in subsequent sections of the report.

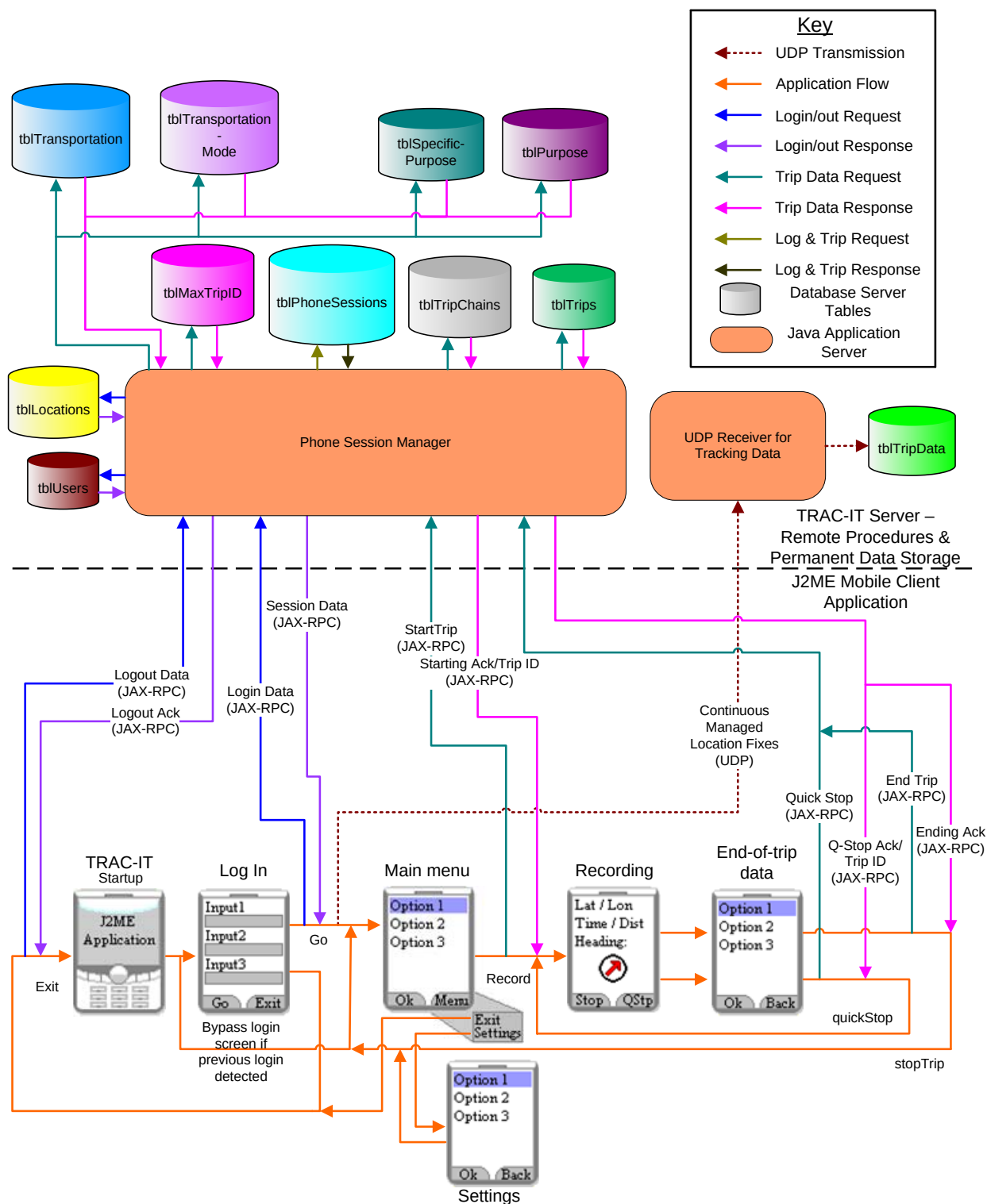


Figure 10 - Detailed System Architecture and Protocol of TRAC-IT Showing Relationship between Client and Server Components

4.2 TRAC-IT Mobile Phone Modules

The client-side software application module, shown in Figure 11, was developed on the Java 2 Micro Edition (J2ME) platform. It provides an environment that specifically addresses the relevant factors of mobile phones utilized for the TRAC-IT system, primarily their limited resources including as power, storage, and CPU cycles. J2ME introduces the concept of configurations and profiles, which adjusts the J2ME environment for the capabilities of the end-user device. Configurations classify and group devices according to their processing power and storage size, so devices with similar resources (e.g., cell phones and embedded systems) are supported by the same configuration. The Connected Limited Device Configuration (CLDC) defines the class of devices with very limited resources, which includes most mobile phones (61). TRAC-IT is based on CLDC 1.1, which supports floating-point operations and other improvements over the previous 1.0 version.

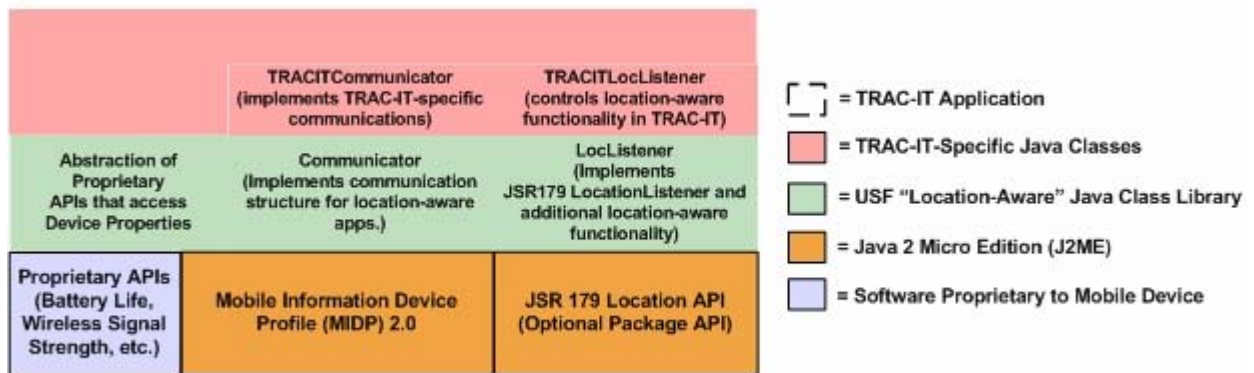


Figure 11 -TRAC-IT Interface with J2ME Architecture

Profiles are specifications that sit atop these configurations and offer features designed to meet the demands of specific applications and end-user devices. For example, most mobile phones use Mobile Information Device Profile (MIDP). The MIDP runs on top of the CLDC and, along with the CLDC, provides Application Programming Interfaces (APIs) for developing software. The profiles can be extended with additional APIs (e.g., the Optional Package APIs and Proprietary APIs shown in Figure 4) to provide more functionality to the environment and automate several important tasks. For example, the J2ME Location API defined by JSR179 is an Optional Package, which is of key importance to the TRAC-IT system. The Location API provides a means to request and obtain GPS coordinates from the GPS hardware or cellular network in a platform-independent manner. It should be noted that the APIs are merely specifications of functionality; the actual system software that makes the functionality available to third party applications referred to as the implementation. Therefore, the exact behavior of different implementations of the Location API can slightly differ (26).

The TRAC-IT mobile phone software application is based on a core library of J2ME software developed by USF. This library consists of many classes that implement basic functionality of a location-aware application, including session management, retrieving and processing location data from the device, and sending this data to a server. The core library also implements advanced functionality such as the State Machine and Critical Point algorithm, features that are discussed in later sections, which are exposed to the TRAC-IT application for use. The core components of the library and how they are inherited by TRAC-IT-specific modules can also be seen in Figure 11.

The TRAC-IT software application divides operation procedures between four entities illustrated in Figure 12: The Main TRAC-IT MIDlet & User Interface, the Timer, the LocationListener, and the Communicator. Threads are launched from each of these entities as necessary. Each component is discussed in detail in the following subsections.

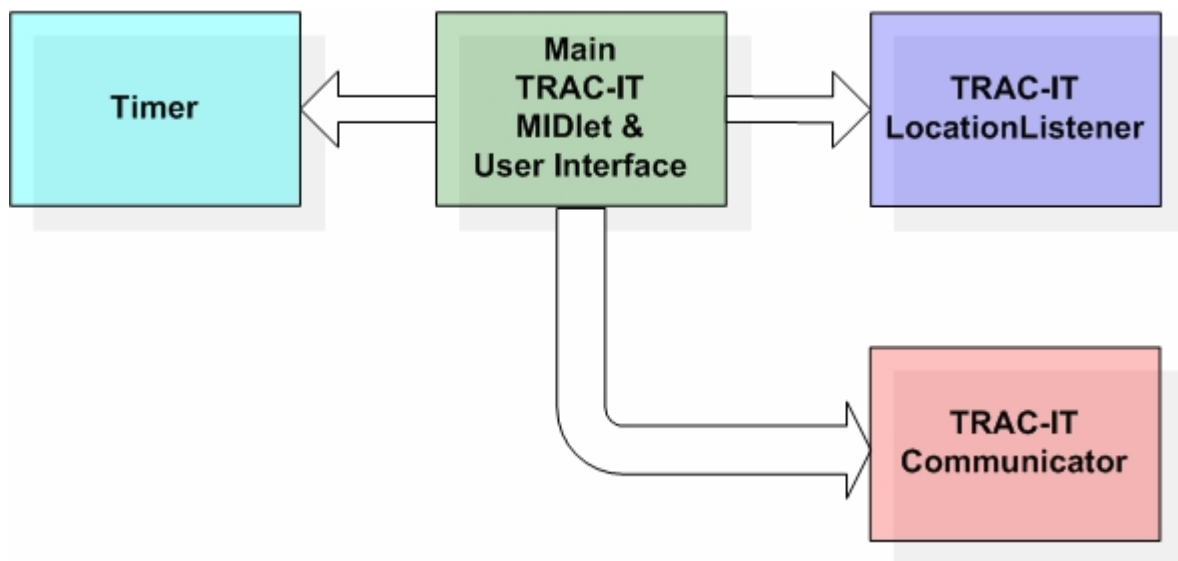


Figure 12 - Major Entities of the TRAC-IT Mobile Application

4.2.1 The User Interface

The main TRAC-IT MIDlet is responsible for the screen switching of the user interface, execution status, and coordinating and launching the other threads. Figures 13 and 14 show the user interface flowchart for the TRAC-IT mobile phone application. TRAC-IT's user interface consists of a series of screens that allow the end users to interact with the system by providing personal and trip information. In addition, the TRAC-IT software provides users with system status information and real-time trip data including the length of the current trip segment and the amount of time spent traveling.

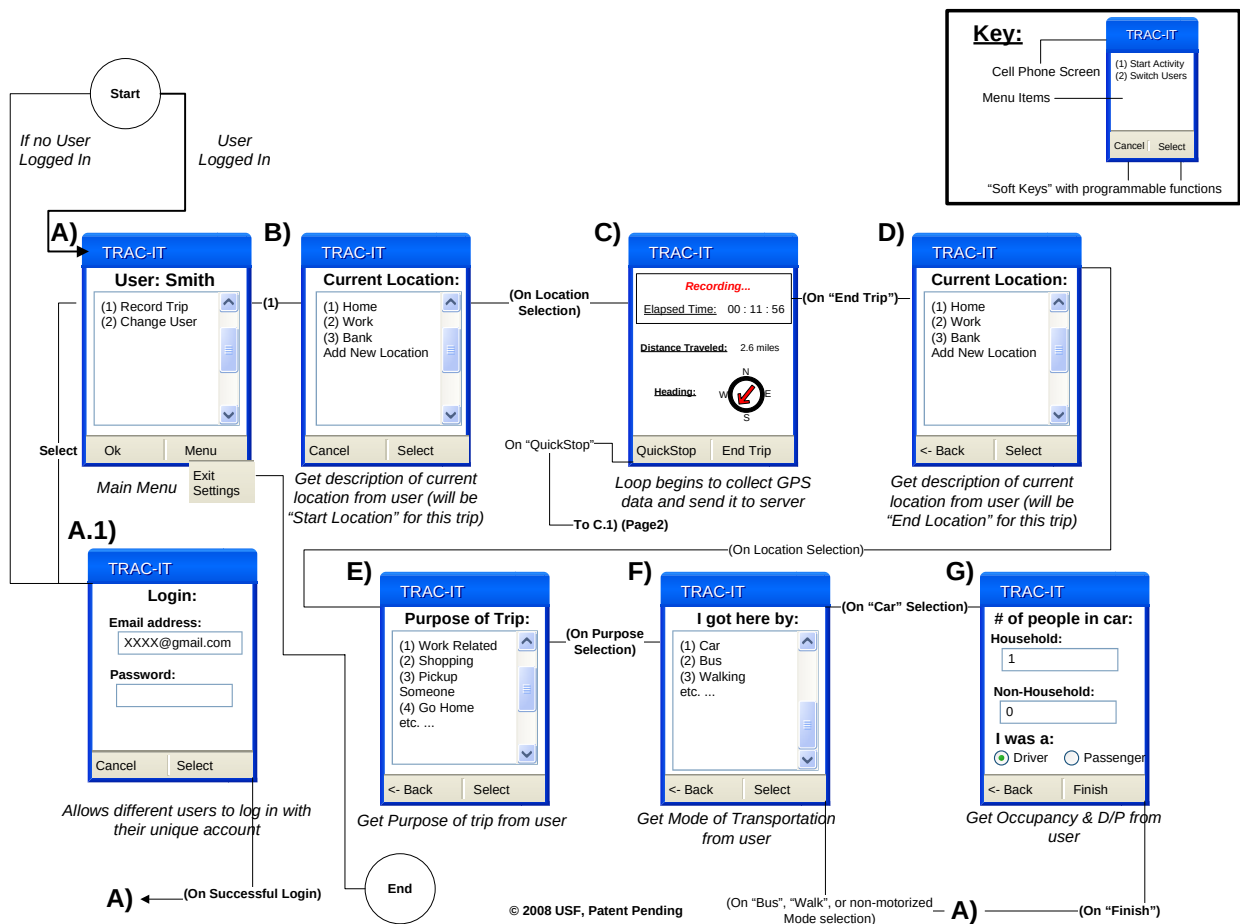


Figure 13 - TRAC-IT User Interface Diagram – Primary Screens

The Record Store, or persistent storage on the phone, keeps a record of the last user that logged in. The first screen displayed to the user may either be the Login Screen (A.1 in Figure 13), if no information from a previous user is stored, or the Main Menu Screen (A in Figure 13) if the user has already logged in. The Main Menu screen appears for subsequent uses of the TRAC-IT system, so that the same user does not have to repeatedly enter his or her login data.

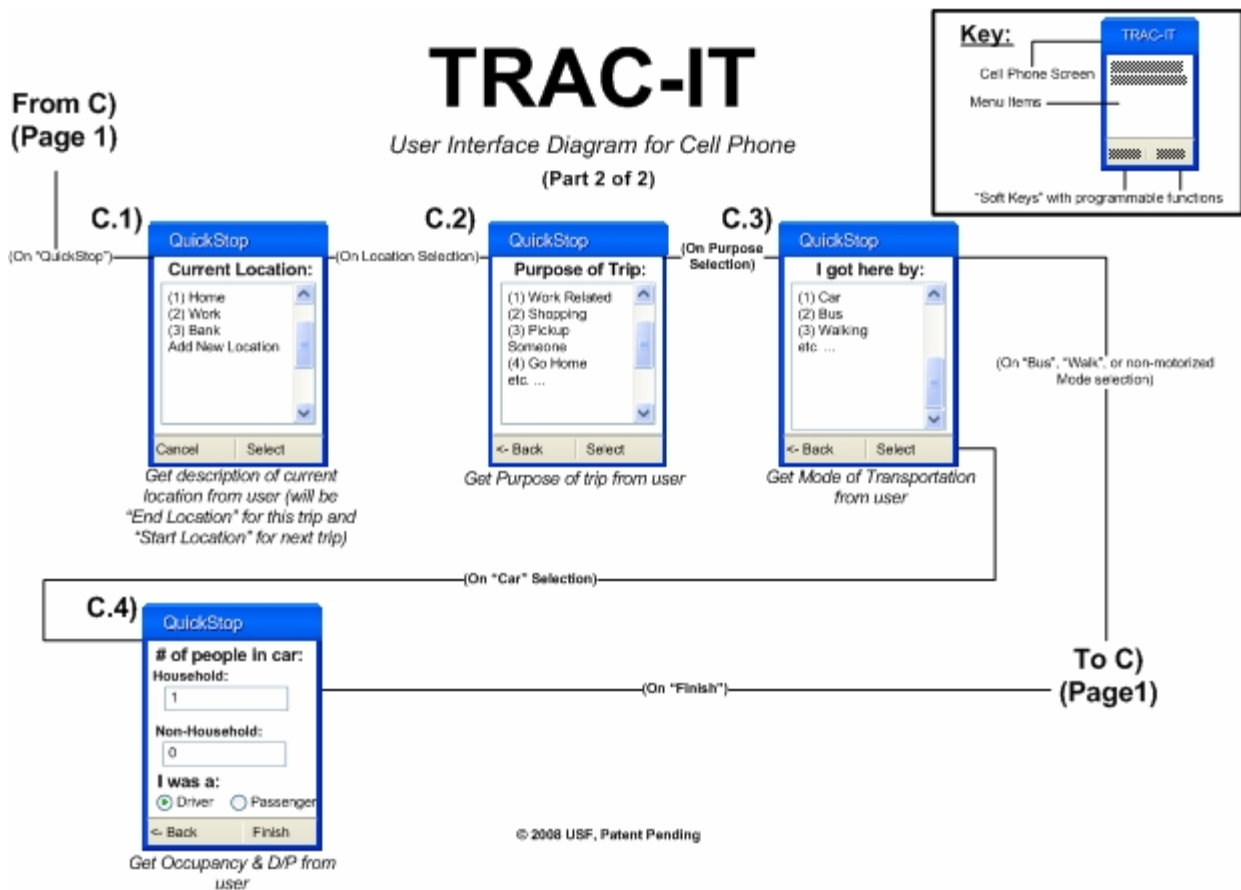


Figure 14 - TRAC-IT User Interface - QuickStop Screens

A Sample Walkthrough of TRAC-IT User Interface

A simple trip begins when the user signs in, either manually by means of the Login Screen, or automatically if TRAC-IT has been previously used. The user may, at that point, choose to record a trip, change the phone settings, or log out of the system.

Recording a trip begins with the selection of a start location from the list of previously visited landmarks, or the addition of a new location from the appropriate form (Figure 13, Screen "B"). The first time a user signs in to TRAC-IT, there may be only the "Home" and "Work" landmarks associated with that account, so these will be the only options available from the Start Location screen. Location can also be entered from a web page or the TRAC-IT database management system in order to prevent the user from having to enter many locations with the cell phone user interface.

Once the location is selected or entered, the device displays the Recording screen (Figure 13, Screen "C"), which displays the time elapsed and distance traveled for the current trip segment. On this screen the user has two options: Stop Trip or Quick Stop.

Stop Trip is used if the travel behavior being recorded consists of a single segment – a one-way trip from home to work, for example. When this option is selected, the user is presented with a series of screens to gather information about the trip. An ending location is requested (Figure 13, Screen “D”) and, as with the Start Location, if the end point is not one that has been previously visited (and saved in the user’s account) a New Location screen can be used to enter this latest landmark.

A Trip Purpose (Figure 13, Screen “E”) is then requested from the user. The options available, including sub-categories, are:

- ♦ Work
- ♦ School or Religious
 - School
 - Religious
 - Library
- ♦ Medical
- ♦ Shopping
 - Goods
 - Services
 - Car Services
 - Personal Business
 - Pick Up/Drop Off Item
- ♦ Social
 - Athletic
 - Rest/Vacation
 - Visit friends or family
 - Entertainment
 - Visit Public Place
- ♦ Transportation
 - Pick up a person
 - Take and wait
 - Drop off a person
 - Change travel mode
- ♦ Meals
 - Eat out
 - Takeout food
- ♦ Go Home
- ♦ Other

Note that several of the purposes listed above have sub-purposes, and the TRAC-IT interface automatically takes the user to a “Specific Purpose Select” screen (not shown in the diagram for simplicity) if one of the relevant main purposes (e.g., Shopping or Transportation) is chosen.

The next question asked of the user is the mode of transportation employed for the trip segment (Figure 13, Screen "F"). The options provided are:

- Car
- Truck
- Van
- SUV
- Motorcycle
- Bicycle
- Bus
- Walking
- Skateboard
- Scooter
- Boat
- Plane
- Other

Unlike the trip purposes from the previous screen, trip modes are specifically coded to user accounts. In other words, selecting "Car" from the account of User 1 does not store the same mode index in the server database as selecting "Car" from the account of User 2. This allows data about specific automobiles to be kept in the server's set of information for the various users of TRAC-IT.

If a mode of transportation is selected that is both automated and personal (e.g., a car or a truck, but not a bus) then the user is taken to an Occupancy screen (Figure 13, Screen "G"), where the user may enter the number of companions with whom he or she has traveled, whether or not they are of the same household, and whether or not the TRAC-IT user was the driver or a passenger. When this last element of data is entered, the system transmits the trip information to the server, and the user is returned to the main menu (Figure 13, Screen "A") where another trip may be recorded, or the user may log out.

If Quick Stop is selected instead of Stop Trip, the same sequence of screens are presented to the user (Figure 14), requesting the same end-of-trip information. Instead of returning to the main menu after transmitting the data, however, a new trip ID is returned and the second segment of the compound-trip begins immediately. The ending location of the previous segment becomes the starting location of the new trip, and the Recording screen (Figure 13, Screen "C") is displayed with the elapsed time and distance values reset to reflect the new segment of the journey. This feature saves the user the additional effort of ending a trip, defining a location, and then starting a new trip, and selecting the same location again as the beginning location of the new trip. As many segments as the user requires are recorded in these "trip chains," and related segments are related to each other in the server database by a ChainID.

A "Settings" menu, accessible from the Main Menu screen (not shown in diagram for simplicity), exposes some of settings for advanced internal TRAC-IT functions (discussed later in this report) so they can be manipulated when necessary. Default values of these settings are also supplied inside the application, and could be locked from user access if desired. For testing purposes, these settings can currently be manipulated through the user interface. One feature in the Settings menu specifies whether or not to send all GPS points obtained during the course of the trip or only those considered critical by the phone-side software. This menu also provides the user with the ability to turn Passive Tracking off, in which case GPS coordinates are only sent during user-defined trip segments. If the passive tracking feature is on, UDP transmissions are sent to the server as long as the application is active, regardless of whether or not a start location for the current trip segment has been selected, or the last trip has ended. Passive Tracking set to off also prevent the transmission of location data during the segment-breaks of trip chains as defined by QuickStop. The passive tracking feature is meant to allow TRAC-IT to function simply as a tracking device that requires no manual user input. This feature must be balanced to protect user privacy, since the user's behavior will always be tracked even if they no longer specifically start and stop trips in the TRAC-IT user interface. Additional performance considerations must be taken into account if the passive tracking feature is used, since GPS fixes and UDP transmissions can drain the phone battery quickly if they are not properly managed. Also, rapid UDP transmissions every 4 seconds can block incoming phone calls. When passive tracking is turned on, it is advisable to use the State Machine and Critical Points features of TRAC-IT in order to minimize the application's impact on phone resources.

Another feature available to the user by means of the Settings Menu is the ability of the user to turn the state machine off or on. With the State machine off, the phone will attempt to send GPS coordinates at fixed intervals, regardless of the quality of satellite or cell tower information. With the machine on, the incoming data are evaluated and the timing of GPS requests is adjusted. Finally, the UDP Buffer setting allows users to buffer coordinates and send them in packets of a predetermined size. A buffer size of 0 will require the phone to transmit its coordinates upon each location update. A buffer size of n will transmit $n+1$ stored data points whenever this quantity is obtained.

Trip Feedback

Since the travel advisory feedback system is the main focus of TRAC-IT Phase 2, a complete description of system design, development, and testing is available in the final report of that phase.

Periodic messages can be sent to the user's device with suggestions for refining travel behavior. These suggestions would be transmitted to the phone based upon the activity recorded during the use of the TRAC-IT system. The feedback is sent from the server using a messaging system and is received at the TRAC-IT device as a text or

multimedia message. Therefore, the TRAC-IT application does not need to be active to receive the data.

Route Prediction & Incident Notification

The Path Prediction algorithm, discussed extensively in the TRAC-IT Phase 2 final report, is one tool that generates immediate real-time feedback based on the user's travel behavior history as well as their real-time location. As the user begins a trip, the TRAC-IT Path Prediction algorithm scans their past behavior that has occurred in the past around the same location and looks ahead to see if there are any incidents on these paths. If there is an incident in the user's predicted path, the algorithm gives the user immediate with information on the incident so they can avoid the trouble area completely before they reach areas of congestion. In the current implementation of TRAC-IT, the information is delivered in the form of a text message to the mobile phone.

4.2.2 The Timer

The Timer is responsible for the real-time updating of the screen displayed to the user during trip recording, (Figure 15). The timer thread primarily updates the "Recording" screen, on which time and distance of the current trip for the user are shown. In addition, a "heading" value is calculated from the position data to provide the traveler with a rough estimate of the direction in which he or she is traveling relative to "North." If no satellite fixes are being currently obtained, a question mark (?) is displayed for the heading icon. If the speed at which a user is traveling is too low to accurately calculate the heading, a dot (●) is shown instead of a directional arrow.



Figure 15 -The Recording Screen

4.2.3 The LocationListener

The third component of the TRAC-IT mobile software application is the TRAC-IT LocationListener class. The LocationListener is responsible for the requesting and obtaining of GPS information that can be used to derive coordinates, speed, heading and other position and travel data. The LocationListener operates independently from the user interface and timer. As soon as the application is launched, the LocationListener is registered by the main TRAC-IT MIDlet and it begins the process of determining its location based upon the information it receives from satellites and the cellular network. One of the advantages of this method is to have coordinates ready at the very beginning of a recorded trip; in other words, by the time a user labels a particular point as the “start” of a trip, the cell phone has already established the necessary connection with the server and received the necessary information from satellites to avoid any delays in marking the current position.

The TRAC-IT MIDlet defines an initial interval, in seconds, of how often a new location fix is attempted upon LocationListener registration. Also defined are criteria that the application developer uses to determine what location fixes are usable for that particular application as well as additional information required by the mobile phone. For example, the application can specify that it wants location fixes with an estimated accuracy within 30 meters every 4 seconds. The mobile phone will then use the hardware and network resources available to try and meet this accuracy criterion with a new location provided every 4 seconds. For a 30 meter accuracy requirement and 4-second interval, assisted GPS will likely be used. If a GPS position can be calculated by the phone, then the location data is marked as “valid” by the mobile phone and

provided to the TRAC-IT application. If a GPS position cannot be calculated, then the location data is marked as "invalid" by the mobile phone and provided to the TRAC-IT application. As previously noted, there are times at which certain information, like the heading and speed, cannot be obtained due to a lack of sufficient satellite signals. If a GPS fix cannot be calculated in these circumstances, then the phone will return the location of the nearest cell phone tower or center of its current cellular coverage area (Cell-ID). If the Cell-ID information available to the phone has an estimated accuracy worse than 30 meters (when using cell towers, this will always be the case), then the Cell-ID location will be marked as "invalid" based on the application's criteria. TRAC-IT may also obtain a more precise location data by requesting a location fix that uses triangulation information from cellular signals from multiple nearby tower using technologies such as AFLT. This position information is more precise than Cell-ID, but less accurate than GPS. However, this type of position fix cannot be used very frequently since it places an additional load on network resources for every fix request.

After an initial Cell-ID location is obtained when GPS is not available, additional Cell-ID locations do not need to be stored at the server since they provide little additional transportation information. Additionally, if the application continues to request GPS fixes from the mobile phone when GPS signal levels are not sufficient to calculate a fix, the phone will waste significant resources such as battery power and CPU cycles while trying to calculate a position and send this data to the TRAC-IT server. Therefore, it is desirable to limit repeated transmissions of Cell-ID information to the server, as well as increase the amount of time between new location requests when it is apparent that GPS is not currently available.

Figure 16 shows a State machine, which is implemented in the USF Java Class library and can be utilized by TRAC-IT to regulate the rate at which position information is requested from the GPS hardware and transmitted by the phone. This State machine can be activated or deactivated from the Settings menu in the TRAC-IT user interface in order to facilitate testing. If invalid location information is repeatedly obtained by the application, it will gradually increase the amount of time between new position requests until a maximum value is reached at the right-most state "S3" (i.e., a position request every five minutes). When valid location information is obtained, the amount of time between new position requests will slowly increase until a minimum value is reached at the left-most state "S1" (i.e., a position request every four seconds). The State machine also supports the functionality of "snapping" back to the most frequent polling rate when the first valid location data is obtained. TRAC-IT utilizes this functionality, since applications such as TRAC-IT should begin tracking at a rapid rate as soon as a valid location is found.

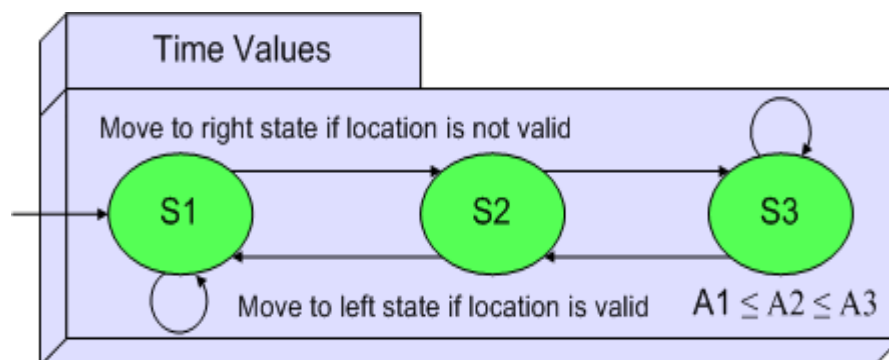


Figure 16 -The State Machine Controlling Frequency of Location Updates and UDP Transmissions. Patent Pending, USF 2008.

It should be noted that the state machine is not limited to three states; the TRAC-IT application can define a larger number of states when it initializes the LocationListener class in the USF Java class library. In its current implementation, the TRAC-IT system only manipulates the time values related to the frequency of location requests. The values related to the timing of the LocationListener are the Interval (the time between fix requests) and Timeout (time allotted for the retrieval of a “valid” fix) and MaxAge (the length of time a fix may remain “current” on the phone before a new location fix must be calculated). The State machine may also be configured to change positioning technologies or manipulate other variables.

When the GPS thread is launched, the initial state sets an interval between location requests of four seconds and a timeout value (i.e., length of time before an “invalid” location is returned) of two seconds. If fixes are obtained within these constraints, the state does not change, and the current settings remain constant. If a valid fix cannot be obtained within the settings determined by these values, however, the machine switches the LocationListener’s settings to reflect the S2 values. If a valid fix cannot be obtained with the new settings, then the S3 values are used, and so on. Likewise, the machine shifts to a lower state if valid fixes are obtained within the current constraints. The machine can also jump directly to the S1 state if it is desired to immediately begin requesting fixes at a rapid rate again once a single valid fix is calculated. Further testing is required in order to determine an ideal number of states and an ideal pattern of changing states. It should also be noted that properties other than a valid/invalid fix can be used as a state transition trigger in the state machine.

4.2.4 The Communicator

The Communicator module of the TRAC-IT system is responsible for wireless communication between the mobile device and the server. Figure 17 highlights the interaction between the LocationListener, Communicator, and TRACIT MIDlet. The TRAC-IT system is a distributed system in which partial data processing takes place on the mobile phone and other functionality is executed on the server. All function calls used to access the server database and return results to the mobile phone are handled

via web services. The mobile phone makes a function call to the server, and the server returns a result to the phone.

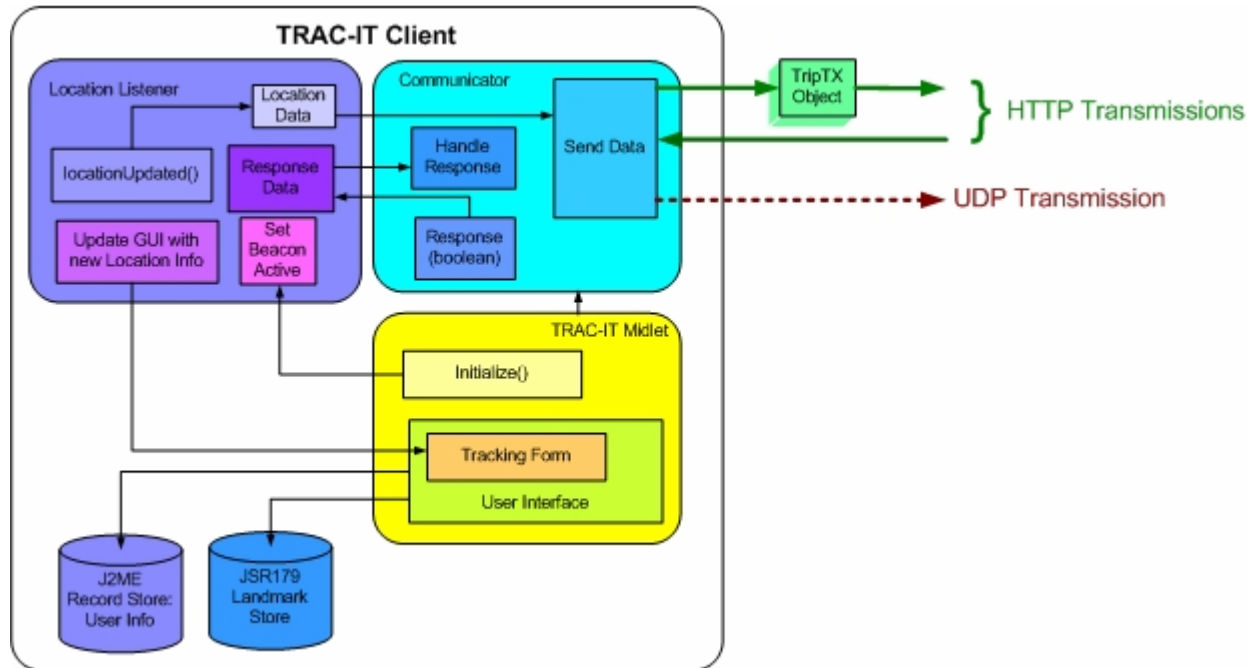


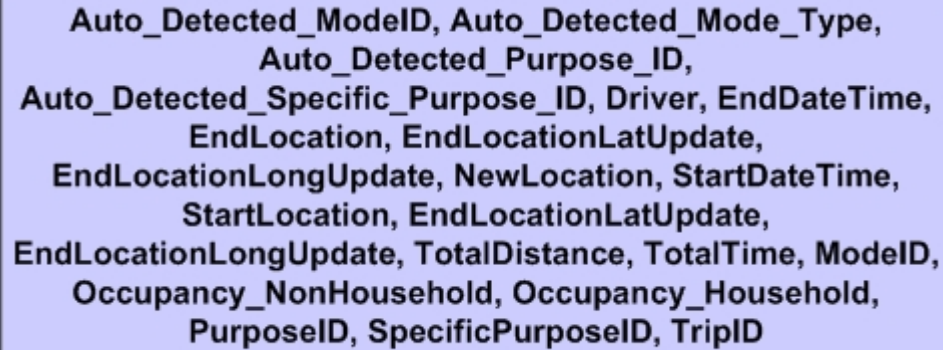
Figure 17 - TRAC-IT Mobile Application Modules

The information for creating and destroying user sessions is transmitted by the Communicator module whenever the application is started or shut down. Sessions are used to tie incoming information for a current “logged in” mobile phone to its information in the server database without requiring that user information such as username and password be continuously passed back and forth. All HTTP and JAX-RPC transmissions, used to login and initiate other server functions, can be secured using HTTPS.

Several trip-related functions are also defined in the Communicator class:

- ♦ startTrip
- ♦ stopTrip
- ♦ quickStop

The startTrip function is called at the beginning of a trip segment; it sends location data relevant to the first point of the segment via the TripTX object (Figure 18), and a unique tripID is identified within the server for the current trip. The stopTrip function is the corresponding function called at the end of a particular segment.



Auto_Detected_ModelID, Auto_Detected_Mode_Type,
Auto_Detected_Purpose_ID,
Auto_Detected_Specific_Purpose_ID, Driver, EndDateTime,
EndLocation, EndLocationLatUpdate,
EndLocationLongUpdate, NewLocation, StartDateTime,
StartLocation, EndLocationLatUpdate,
EndLocationLongUpdate, TotalDistance, TotalTime, ModelID,
Occupancy_NonHousehold, Occupancy_Household,
PurposeID, SpecificPurposeID, TripID

Figure 18 - The TripTX Object

The quickStop function used as a shortcut to enter segments of a trip chain, was developed in Phase 1 and was further enhanced for Phase 3 application. Trip chains consist of connected segments to different locations in a single extended travel period. For example, a trip chain may consist of the following segments: home to gas station, gas station to store, store to home. The quickStop function is used to combine the stopTrip and startTrip function into a single function call when a new trip has a starting location identical to the stopping location of the previous trip. This feature is also exposed to the user so they can simply select "QuickStop" when ending a trip and select a location once. The location is automatically marked as the starting location of the next trip without the user having to select it again.

The startTrip, stopTrip, and quickStop functions are also used to enter new user-specific locations into the appropriate database. The TripTX object contains a NewLocation object that holds both coordinate and descriptive text "tag" information that user may enter via the cell phone keypad upon departure from or arrival to that location.

The trip-specific functions and the session creation/destruction methods depend on a send-and-response transmission method, implemented either by JAX-RPC or via an HTTP mechanism generated by the corresponding web service. On the other hand, the in-trip transmission of position data utilizes the less reliable User Datagram Protocol (UDP) that contains GPS coordinates and other fix data (Figure 19).

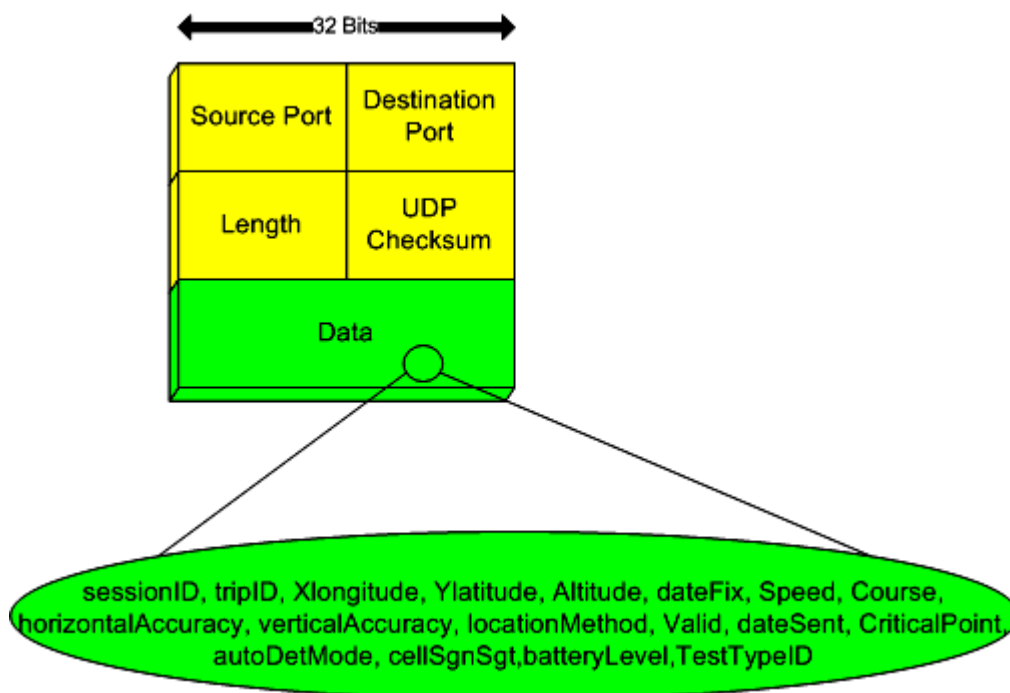


Figure 19 - The UDP Datagram Packet for TRAC-IT

UDP data are transmitted from the phone to the server as long as the GPS thread is running. Tests performed by the research team have shown that not only do constant wireless transmissions reduce the battery life of the cell phone dramatically, but also interfere with the device's ability to perform one of its primary functions; to receive phone calls.

To alleviate these issues, a buffer of variable size is implemented in the phone's memory to transmit sets of fixes all at once. If the buffer size is set, for example, at 7, then eight fixes are stored in memory (fix 0 – fix 7), and these are "flushed" to the server when the buffer is full. This reduces constant communication with the server and leaves openings for incoming phone calls.

UDP is unsecured by nature, and therefore some privacy considerations must be taken when transferring location data. In its current implementation, TRAC-IT transfers all user account information over HTTP when a session is created, which allows this method to be secured by using HTTPS so the information is encrypted. Once the device receives the session ID back from the server, these data are stored in the TRAC-IT application as an identifier. When location data are sent via UDP, this session ID is included in the packet in order to identify which session, and therefore user, it should be associated with when it reaches the TRAC-IT server application. Since the session ID changes upon each login, it would be difficult to associate multiple sessions with a single user. Therefore, if a packet is intercepted, it will not be apparent which user is associated with that location information.

4.2.5 On-phone Storage

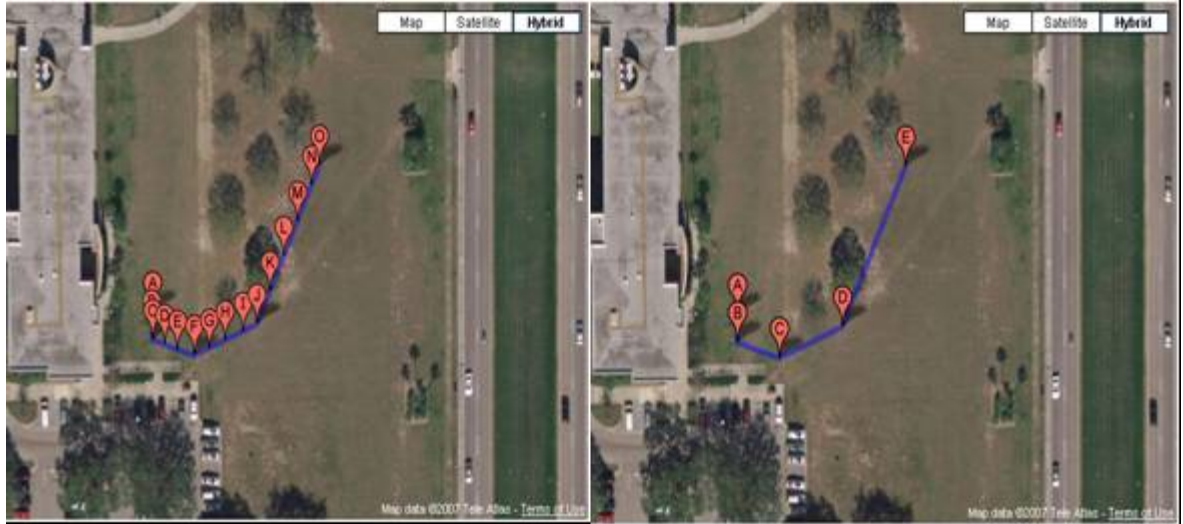
Two types of information are stored in the cell phone's persistent memory between application executions – the "Record Store" and the "Landmark Store." The Record Store capability of the phone is used to store the username and password of the last user, as well as user account settings. This saves time if the same user is using the phone repeatedly; after the first log-in, the user software automatically logs in using this information until the current user logs out or another signs in.

The Landmark Store keeps an on-phone record of the locations associated with individual accounts. This is done to keep the data shared between the end-user device and the server consistent.

4.2.6 Critical Point Detection

When GPS data are recorded frequently by the mobile phone, a large amount of data that are not necessary to reconstruct a participant's trip is generated. Examples of unnecessary data include GPS points collected when the phone is not moving as well as multiple GPS points that create a straight line. The data that are needed to reconstruct a user's travel path are referred to as "critical points." A "critical point" for the TRAC-IT system is defined as a GPS fix that occurs when a traveler changes direction by more a certain threshold. To ensure that directional changes are not calculated due to readings that are inaccurate within the accuracy uncertainty limit, fixes are assessed only if the corresponding speed reported is greater than a speed threshold. The first and last points of every segment are also considered critical. These techniques comprise a "critical point detection algorithm" (Patent Pending USF, 2008) that has been integrated into the TRAC-IT system software.

One of the main benefits of critical point detection is a reduction in the number of wireless transmissions from the phone to the server, complementing the implementation of the buffer. Figure 20 shows a simulation of a brief walking trip on Google Maps with all the UDP data points plotted in the left picture of the figure. As can be easily seen by examination of the route, several of the points hold information that, for the purposes of overall trip analysis, can be extracted to form a much smaller data set. The picture on the right side of Figure 20 shows the same trip with only the critical points displayed. The general route of the trip, the time of travel, and the key events (such as stopping and changing direction) all remain intact, although only a third of the transmitted data was necessary to obtain them. The efficiency of the critical point algorithm is shown in Figures 21 and 22 for driving trips where a considerable number of unnecessary points, recorded when the user is driving in a straight line, are eliminated.



**Figure 20 - Brief Walking Trip – All Points Recorded (left)
vs. Critical Points (right)**



**Figure 21 - Car Trip with all GPS Points (left)
vs. Critical Points (right)**



**Figure 22 - Car Trip with all GPS points (left)
vs. Critical Points (right)**

UDP transmissions consume battery power, CPU cycles, and network bandwidth. Reducing the number of UDP transmissions can benefit these mobile phone resources significantly, especially considering that GPS fixes (and therefore UDP transmissions) can happen as often as once per second. Significant financial savings can also be realized for mobile phone data plans that charge per byte transferred, (Table 5).

**Table 5 - Financial Savings of Data Charges
using Critical Point Algorithm**

Trip	Number of Trip Points	Number of Critical Points	Bytes saved	Financial savings*
1	73	26	5593	\$0.17
2	363	56	36533	\$1.10
3	489	65	50456	\$1.50
4	208	73	16065	\$0.48
5	357	62	35105	\$1.05
6	2330	159	257159	\$7.71
7	1022	139	105077	\$3.15
8	811	137	80206	\$2.40

*Based on 119 bytes per UDP package and a charge of \$0.03 per kilobyte.

The TRAC-IT mobile phone application settings provide users with the option of transmitting all obtained fixes (both those from satellites and those from cell towers) or only the critical points for testing purposes.

The “bridge” between the end-user device and the server-side elements of the TRAC-IT system consists of the cellular carrier network and the Internet.

4.3 The Cellular Carrier Network

Although standards define the network-layer protocols used for the transmission of data in the TRAC-IT system, the mobile phone first establishes contact with its cellular network. The cellular network is illustrated in Figure 23. The network carrier underlies these protocols. The Sprint-Nextel iDEN and CDMA cellular networks were selected for the TRAC-IT application because they are the only two networks to expose access to mobile phone location data to third party applications. The location of the cell tower with which the end-user device is in contact is returned in the event that the TRAC-IT application’s query of the GPS fails to return a fix.

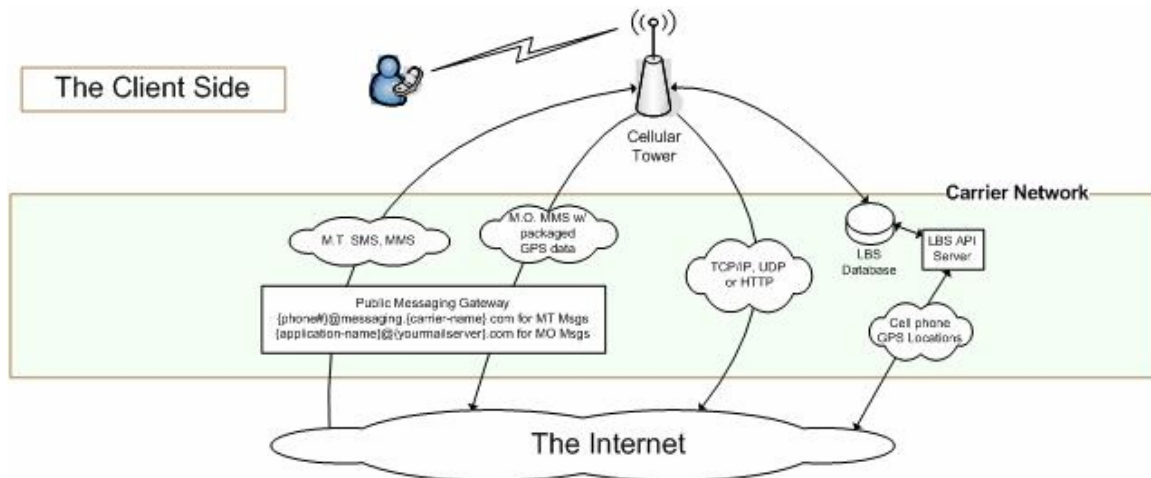


Figure 23 - Mobile Phone Network Carrier

Feedback to the phone from the server-side trip analysis is also provided by means of the network carrier. A public gateway exposed by the cellular carrier converts email messages for reception from the server to the phone in SMS or MMS format. The server then addresses this data to an email address such as user-telephone-number@messaging.mycarrier.com.

Carriers are capable of manipulating transmitted messages like SMS and MMS to configure them for efficient use of bandwidth and storage space (62). For example, video files that are sent to Nextel phones as attachments are converted from video format to animated .gif files. Other limitations such as maximum text length may also be enforced.

4.4 The Internet

The Internet, the second stage of the wireless bridge between end-user and server, handles both the delivery of messages to the phone via either email or SMS, and the transmission of session, trip, and location information by its wireless protocols (Figure 24).

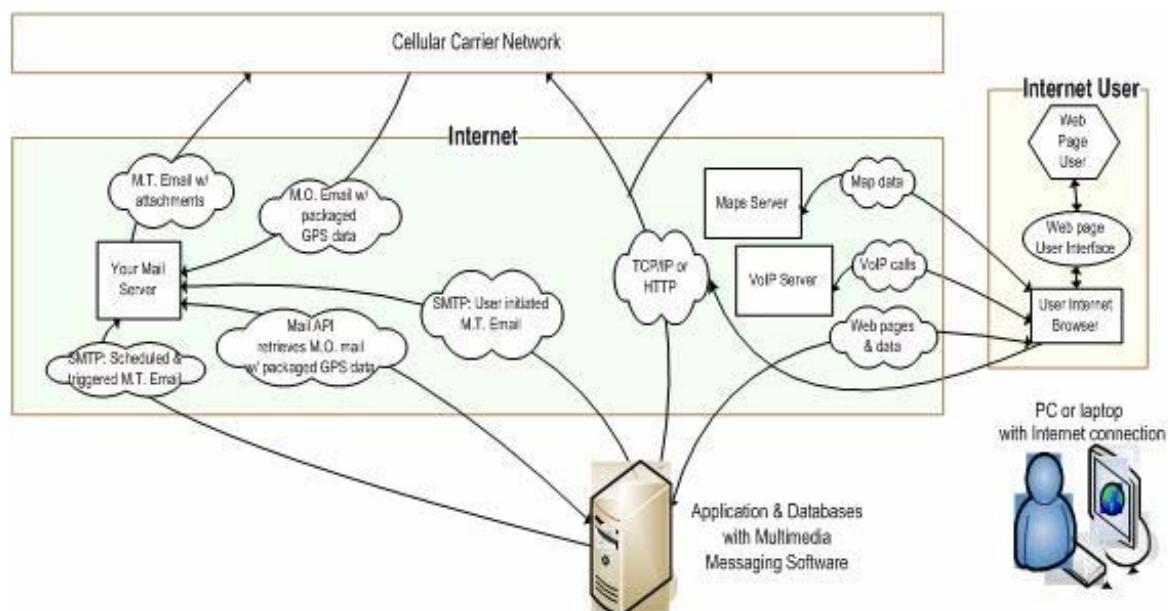


Figure 24 - Internet Network Module

For the messaging system, transmissions from the application server to the cell phone travel over the Internet to the Email Server, and then to the Public Messaging Gateway for formatting and reception by the end-user device (Figure 25). For the transmission of application data, the Internet handles the implementation of the protocols for session and trip data and UDP for GPS information.

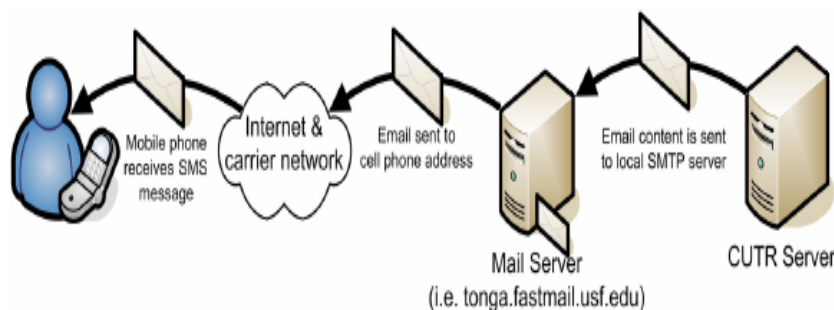


Figure 25 - Messages over the Email Server

The Internet also provides access to the trip information stored in the database of the server which can be retrieved and displayed in many types of GIS analysis software. For example, Figure 26 is that of a wide angle view of travel path information (shown as series of markers) stored in TRAC-IT database, accessible via the Internet. The view is displayed using Google Earth. Another example, displayed in Figure 27, is a close up view of a travel path with information stored in TRAC-IT database and is accessible via the Internet. The data collected are displayed using Google Earth with GPS points shown as markers.

Automated analysis of travel data that is performed on-server by expert systems and artificial intelligence techniques do not require network communication, although remote automated processing from another server is possible.

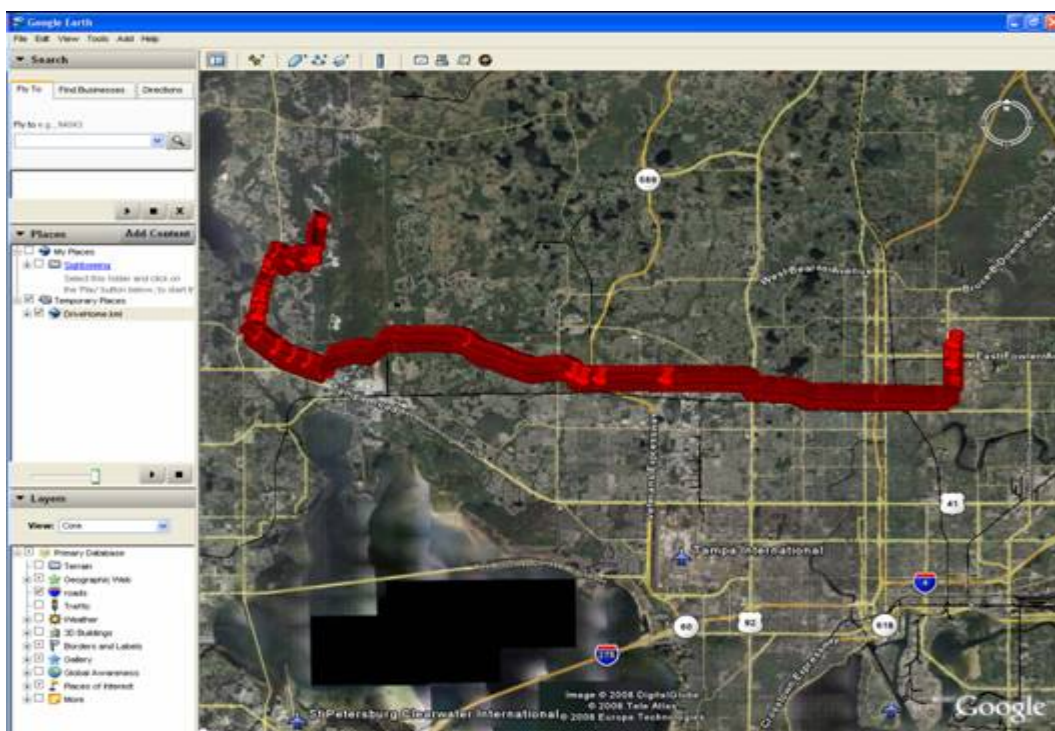


Figure 26 - Wide angle View of Travel Information Displayed Using Google Earth

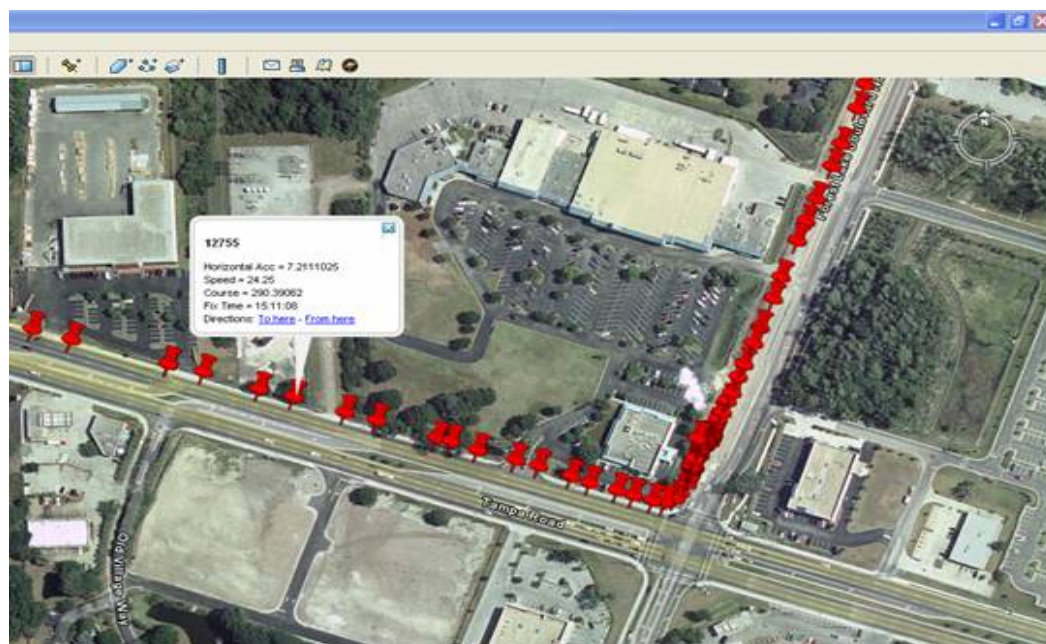


Figure 27 - Close-up View of Travel Data Displayed Using Google Earth

4.5 Server-side Modules of TRAC-IT

The servers-side modules of TRAC-IT are responsible for receiving, analyzing, and providing feedback for travel data transmitted by the end-user device. It consists of two main software elements: An application server that deals with communication and analysis and a database server that deals with storage and retrieval of trip information. For the implementation of the TRAC-IT system, the Sun Java System Application Server (a.k.a. Glassfish) was selected along with the Microsoft SQL Server 2005 Database Server because there are versions of both software available for public use at no cost (55), (56), (66). Since the basic server application conforms to Java API standards, any Java application server or database server that supports Java Database Connectivity (JDBC) can be utilized. Since TRAC-IT is a distributed application between the mobile phone and server, an application server must manage all interactions between the database and the mobile phone. In the following sections, the interactions between the phone, server, and database via remote procedure calls are discussed as well as the individual components that perform TRAC-IT data analysis.

4.5.1 Remote Procedure Calls - Managing the Distributed TRAC-IT Application

The application server as shown in Figure 28 is responsible for a complex set of tasks involving coordinating data from the end-user device, the various database tables containing current and previously-stored information, several internal modules that manipulate the incoming information and the expert system that provides feedback to users.

The user's first interaction with the server is at the login procedure. If the application is being run for the first time, its Record Store is empty and no current username and password exist for automatically establishing a connection. The application server listens for incoming wireless communications then, upon the reception of a new username and password, it validates the user by comparison with information in the database table *tblUsers*. A valid user is then connected to the server by a "session" which is active until the user signs out or terminates the TRAC-IT application. Within the session, when the user interacts with the application by means of the TRAC-IT mobile phone interface, then, a series of trip segments can be recorded.

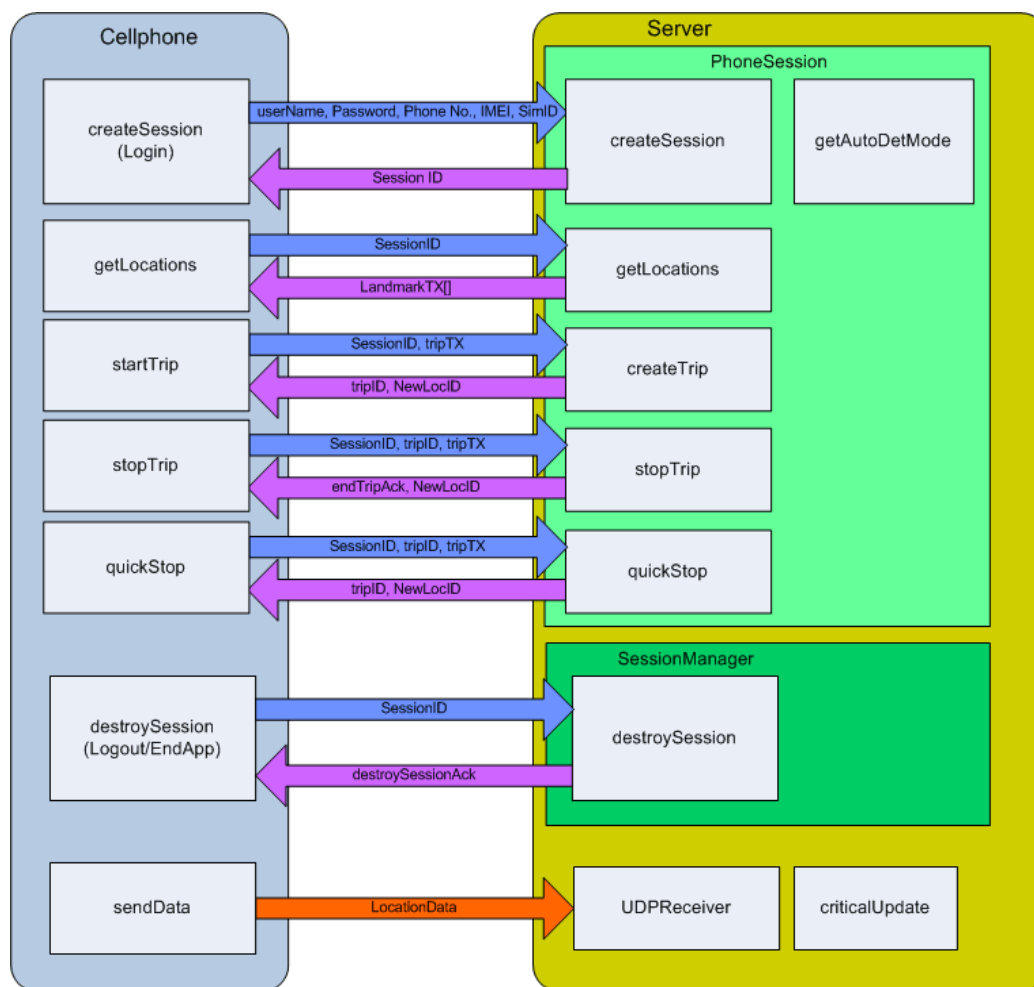


Figure 28 - Server-side Procedures called by Mobile Device

The server receives both the trip boundary information (i.e., start points and end points) and the UDP transmissions of GPS data, which are labeled according to the current trip. GPS information that comes in outside of a specified trip – either before a starting location is selected or after a segment has ended – is labeled as “external” to a trip and stored with a dummy value (-1) for the associated trip ID number. This mode allows passive data collection where the mobile phone will simply collect and send GPS information to the database repeatedly without user interaction required. Position information not associated with trips may nevertheless be recalled from the trip data table by timestamp and user session recorded independently of trip segments. The sets of data associated with user activity are stored in the appropriate tables thus available to the expert system for subsequent analysis. Other information entered by the user such as mode of transportation, trip purpose, and vehicle occupancy are also sent to the server and recorded in the database for storage and analysis.

The server-side procedures are implemented via web services conforming to the JSR224: Java API for XML-Based Web Services (JAX-WS) 2.0 (67).

Interaction with Database Server

TRAC-IT employs a number of database tables to retain information related to both users of the system and their travel behavior, as well as spatial representation of trips and other GIS data (see Figure 29). Some of the main tables are:

- tblUsers – This table stores usernames, passwords, and a unique ID for each record that is used by the system to refer to individual users.
- tblTrips – The set of trips taken on the TRAC-IT system, including start and end times, and the locations at the beginning and end of each segment.
- tblPhoneSessions – A record of the login sessions for each user.
- tblTripData – This table stores the actual GPS fixes for each session.
- tblTransportation – A set of unique transportation modes for each user.
- tblLocations – The set of locations unique to each user; it includes a title as well as geographic coordinates.
- tblHouse – The household ID of each user.
- tblBusRoute – A set of tables. There is a table for each BusRoute that includes the GPS coordinates of each stop.
- tblFeedback – The set of trips analyzed by the expert system and the feedback provided

The specific manner in which these database tables interact with the application software is described in the sections following, grouped by the procedures that reference them.

A number of tables used to store GIS spatial data are also represented using ESRI's ArcSDE system.

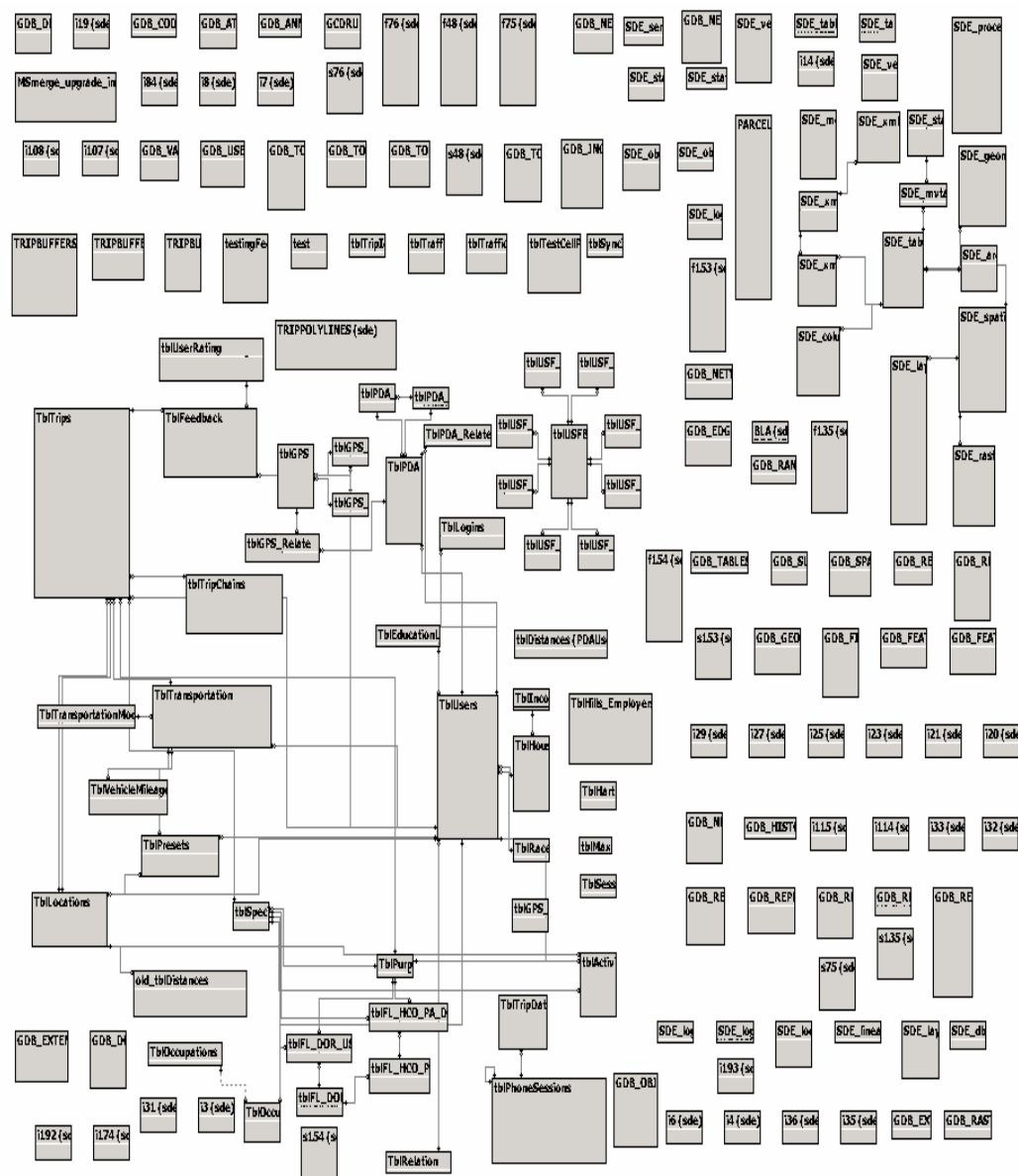


Figure 29 - TRAC-IT Phase 2 Data Schema

Session and Logging Procedures

The login procedure begins with the input of a username and password by the user through the TRAC-IT mobile phone application user interface. This information is transmitted to the server along with data about the device (*i.e.*, the phone number, IMEI and SIM ID), and the function *createSession* compares the user data with information from *tblUsers* to validate the identity of the traveler. If this validation is successful, a unique *SessionID* is returned to the phone; all subsequent communications between the end-user device and the server software are tagged with (and identified by) this code. Following a successful login, the function *getLocations* is called. This retrieves any geographic locations associated with the user from *tblLocations*, and stores them on the phone so that the user can select from a list of the landmarks that have already been visited when starting or ending a trip segment. The first time the user visits a particular place he or she can identify the location through the mobile phone user interface for later reference.

The phone and user information are stored in the database table *tblPhoneSessions*. This table is dynamically updated according to the changes that take place within the session. When a trip is active within the session, for example, this is indicated by the *tripID* being stored in the appropriate field of this table (as opposed to "-1" if no trip has been initiated). If the trip is part of a larger chain the identification number for that set of trips is also contained within the *tblPhoneSessions* table.

One of the reasons current data is stored in the session table is for recovery in the event that the application server temporarily goes down. When the session manager starts upon re-activation of the server, the phone session table is accessed and an attempt is made to restore sessions marked as "active" to the state that they were in at the time of disruption, including actively receiving GPS data associated with a trip that may have been ongoing.

At the end of a period of use, if the user exits the TRAC-IT application or logs out, the *sessionID* is sent to the server, and the session indicated is destroyed.

Processing Location Data

As has previously been stated, the actual transmission of GPS data to the server takes place independently of user-defined trip segments. As soon as the application is started, the mobile phone attempts to obtain GPS fixes, and as soon as a session is established, an attempt is made to communicate this information to the server by means of the *sendData* method (Figure 28).

The protocol for transmitting GPS data (UDP) does not receive an acknowledgement from the receiving server, as the loss of individual packets do not compromise the travel data as would the omission of a starting location, transportation mode, or trip purpose, (Figure 30). The information received by the server is stored in *tblTripData*, and includes the latitude, longitude, accuracy uncertainty estimates, timestamp, the ID number of the current trip (if any), and other variables.

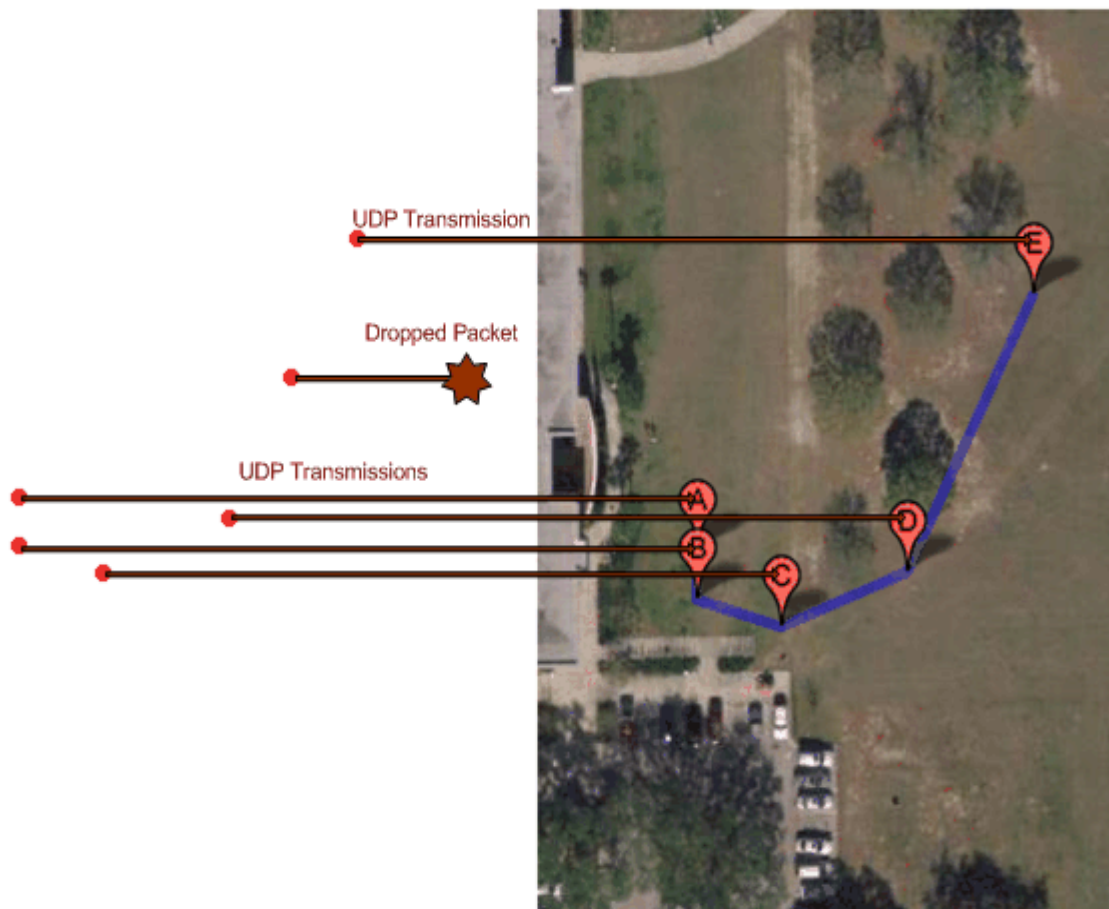


Figure 30 - UDP Transmission of Data Points

Defining Trip Segments

From the server's standpoint, Figure 28, the methods that accomplish this are *startTrip*, *stopTrip*, and *quickStop*. These methods, which define the start point, end point, and intermediate points of a chain of trips respectively, communicate with the server via the responsive method supplied by either the JAX-RPC or HTTP mechanisms.

When starting a trip, the server retrieves a unique tripID and records this as a new entry in *tblTrips*. The initial location of the segment (selected from the *StartLocation* screen, or added as a new landmark by the user) is also stored in this record, and the time of the trip's beginning. When *stopTrip* is called at the end of a trip, the user is prompted for other data about the journey that is used to update the *tblTrips* entry.

The UDP data from the end-user device that is transmitted during the trip is added to *tblTripData* with the tripID that was returned by the *startTrip* function.

The *quickStop* function, used to end one segment and begin another during the course of a trip with several sections, works as a combination of *startTrip* and *stopTrip*. The end-of-trip information is obtained from the user and entered into the *tblTrips* entry for that tripID, but then a new tripID is obtained and returned to the phone as if *startTrip* had been called immediately afterward. A new *tblTrips* entry is added for the new tripID, and the GPS fixes that continue to be sent to the server are labeled with the new identification number. In addition, a unique ChainID, obtained from the table *tblTripChains*, is associated with the *tblTrips* entry for all of the individual segments of the chain.

4.5.2 Critical Point Detection

If the TRAC-IT mobile phone software application is set to send all GPS points to the server (i.e., not only critical points), then further critical point processing must be done on the server. Critical points are useful for server-side analysis as they define the points necessary to reconstruct the user's path. Therefore, critical points are used as the input to the other server-side TRAC-IT function such as Mode Detection and Path Prediction, both discussed later in this report. Utilizing only critical points in algorithms aids in the scalability of the system as well since the number of points processed per phone are reduced. Additionally, when archiving travel data, if they are no longer needed, non-critical points can be discarded saving disk storage space.

4.5.3 Automatic Mode Detection

One of the trip elements that a user is asked to specify by the end-of-trip interface screens is the method of transportation. Choices include walking, bus, car, airplane, and others, and are designed to provide information to the expert system necessary for an accurate analysis of travel behavior.

To move away from user interaction, since the goal of TRAC-IT is to record travel behavior in as unobtrusive a manner as possible, an attempt is made to determine the mode of transportation without the user having to specify it. The algorithm, shown in Figure 31, performs this function relying on two factors: the average speed of travel and the route-segment coincidence.

The algorithm receives the set of speeds for each fix of the relevant trip, and from this derives three values. WP (walk percentage) is the percentage of the trip for which the speed value is within the range ascribed to being on foot (speed ≤ 10 km/h or 3.6 m/s), BP (bike percentage) is a value based upon the pre-determined average speed of a bike trip (10 km/h or 3.6 m/s $<$ speed ≤ 22 km/h or 7.92 m/s), AP (automobile speed) is the percentage that falls within the range of automobile speeds, which in the TRAC-IT system is anything greater than the biking speed of 22 km/h or 7.92 m/s.

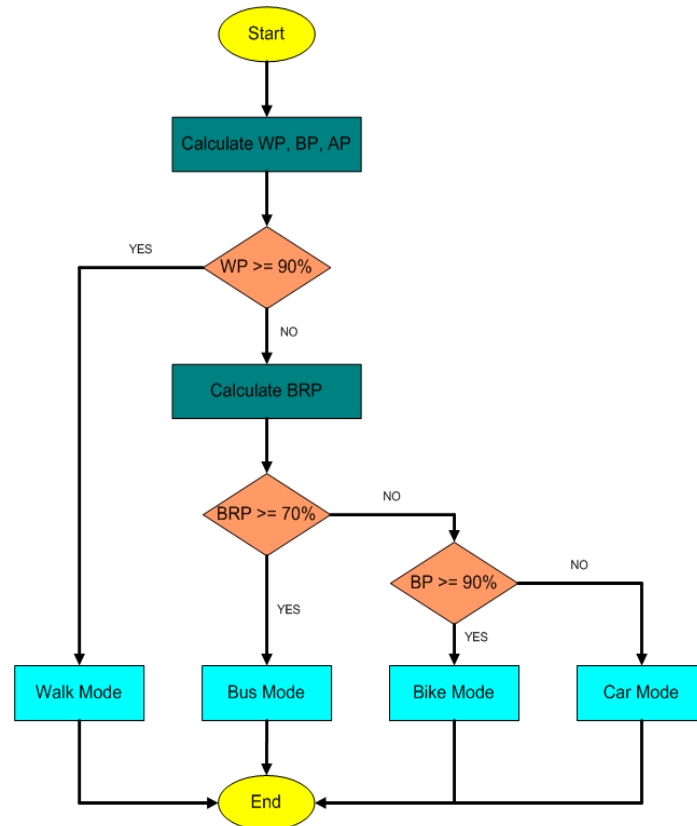


Figure 31 - The Automatic Mode Detection Algorithm.

Source: "Automatically Determining Route and Mode of Transport Using GPS Enabled Phone," Himanshu Gilani, Master's Thesis, University of South Florida, p.41

Trips with a high percentage of walking speed (i.e., very low speed) fixes are immediately considered to have taken place on foot. Even a few elements of location data that reflect a high speed rule out walking as a potential mode, so it is the easiest mode to eliminate from consideration.

Buses are the most complex of the modes to be determined, since they may travel at car speeds or slowly enough on average to be confused with bicycles due to many stops in certain geographic locations. Because of this ambiguity, the first factor considered after the elimination of "walking" as a possible mode, is the route that the traveler takes. The degree to which the recorded trip coincides with known bus routes has been shown to have a high predictive value for mode determination, with a successful detection of walking trips at 100 percent, biking at 96 percent, bus at 92 percent, and car at 100 percent in field tests (68). Total accuracy over all modes was 97 percent. If over seventy percent of the trip is along a bus route, the trip is so labeled, speed considerations aside.

If a set of data indicates that a user is moving too quickly to be considered on a walking trip, and the route taken does not coincide with a bus route, then the speed is

one again used as a selection criterion. If 90 percent of the trip is within the “bicycle” range, then the trip is so labeled; if not, it is considered a car trip.

Naturally, the applicability of this algorithm depends heavily upon the amount of information available for bus routes. Unlike algorithms strictly based upon speed, or the use of Google maps to obtain a visual representation of the journey from almost any geographic location on earth, the automatic mode detector can only be used in accord with public transportation records specific to the area being examined.

More advanced methods of mode detection are currently being explored to determine whether techniques that do not depend on known bus routes can be utilized. One approach is a machine learning software suite named WEKA, which analyzes sets of data to extract general features for identification. The data from known bus trips are provided as input for the neural network implementation in the software, and then the common characteristics of these records are used for comparison against data sets of unknown type. Trips with a sufficiently high degree of similarity with the training set of bus trips are considered to be of the same transportation mode. Field tests to date have yielded a total accuracy of this method approaching 92 percent for all modes, with 92.11 percent of car trips, 81.58 percent of bus trips, and 100 percent of walking trips successfully detected (69).

4.5.4 Travel Advisory Feedback System

Phase 2 of the TRAC-IT project focused on the ability to provide informative feedback tailored to the participant based on their recorded travel behavior. An expert system in conjunction with the known data of a trip is used to provide feedback to users. The goal of travel feedback advice in general and that of TRAC-IT Phase 2 in particular is to change travel behavior by suggesting more efficient options to driving alone.

The expert system is written in C# programming language in order to more directly relate to the Microsoft SQL Server database server. The following factors are considering during trip analysis:

- ♦ Start time
- ♦ End time
- ♦ Start location
- ♦ End location
- ♦ Trip purpose
- ♦ Transportation mode
- ♦ Household

The start and end times are used to determine the duration of time spent traveling, while the start and end locations provide the distance of the trip. Purpose, mode and household (the latter of which allows the system to know if two users live at the same

place) are used in conjunction with the other variables to provide advice to users that may consist of one or more of the following suggestions:

- ♦ Alternate mode of transportation. If the distance traveled is relatively small, and the purpose of the trip does not require a car, (e.g., to get gas or shop for groceries) then the user may receive feedback such as: *"Your trip on 3/27/2007 from home to the store was less than 1 mile. Please consider walking or biking to this location next time. Purpose: Shopping. SpecPurpose: Services."*
- ♦ Certain kinds of shopping might be done from home using electronic means. In such a case, the following suggestion might be made: *"Your Trip on 3/27/2007 from home to the store was 10 miles or more. Please consider shopping online if possible. Purpose: Shopping. SpecPurpose: Goods."*
- ♦ If several disconnected trips are taken by the same user on the same day, then the system will provide the feedback: *"You took 5 extra trips on 3/27/2007. You could have eliminated these trips by stopping at one destination on your way to another."*
- ♦ If a user visits the same location on multiple days in the same week, the suggestion will be: *"You took 5 trips within one week of 3/27/2007 to the store. Please consider combining these trips into a single trip."*
- ♦ If two users from the same household make trips to locations within one and a half miles of each other on the same day, the system will suggest that these trips be combined: *"You took a trip to the store. You could have taken the trip with John on his way to the library. Date1: 3/27/2007 Purpose1: Shopping. SpecPurpose1: Goods Date2: 3/27/2007 Purpose2: SchoolandReligious. SpecPurpose2: Go to library"*
- ♦ If users of different households live within a mile and a half of each other and travel to nearby locations for work, the system suggests carpooling: *"You could start carpooling with John on the way to work and back home."*
- ♦ If a bus stop is near to a user's work location and home, this is pointed out in the system's feedback: *"You could have taken the bus to work. There is (are) 1 stop(s) near your house and 2 stop(s) near your work."*
- ♦ If different users form the same household make several outings for the same purpose in the same week, the system will suggest consolidating these trips: *"You took a trip to the store. You could have taken the trip with John on his way to the store. Date1: 3/27/2007 Purpose1: Shopping. SpecPurpose1: Goods. Date2: 3/28/2007 Purpose2: Shopping. SpecPurpose2: Goods."*
- ♦ Unless the purpose of the trip is for work, the expert system will suggest that the user avoid traveling during heavily trafficked periods of time: *"You made a trip on 3/27/2007 to the store for the purpose of Shopping->Services. It may be possible to start this trip before 5pm or after 6pm in order to avoid rush hour."*

As may be concluded from the feedback options offered, the goal of the expert system's trip analysis is to save the user time and gas costs. On a larger scale, the suggestions for carpooling and combining trips are designed to reduce unnecessary

traffic by limiting multiple trips, reducing the number of vehicles carrying travelers to their destinations, or timing trips to avoid high levels of traffic.

4.5.5 Route Prediction

A prototype extension to the TRAC-IT Advisory Feedback system was developed that seeks to provide users with information about incidents that may encounter along the path they are traveling in real-time. This service is accomplished by predicting the user's immediate route in real-time based on past travel behavior and then scanning ahead on each potential route to determine if any incidents lie along their predicted path. Please see the TRAC-IT Phase 2 Final Report for more information about this feature of TRAC-IT.

4.5.6 Purpose Detection

Trip purpose is one piece of information that currently must be manually entered by users when recording their trips. It is desirable to transition to a completely passive, automated version of TRAC-IT that requires no manual entry in order to reduce the burden on the user. This section discusses the development of an automated Purpose Detection Algorithm that uses GPS Data collected by GPS enabled cellular phones. The GPS data is manipulated within a GIS map to obtain various spatial and location characteristics. This information is then used by the Purpose Detection Algorithm to derive a traveler's trip purpose based on the ending location of the trip.

GIS and GPS are two related technologies that afford scientists the opportunity to implement a highly sophisticated level of accuracy in travel research. Both technologies have exceptionally powerful means to generate accurate global and local spatial data. GIS technology deals with digitally representing the geospatial and geographic characteristics of a region of Earth. By using a GIS map, a region such as a city or a trip path can be accurately represented digitally using a set of Feature Classes (78). Feature Classes are a group of polygons, points, or polylines (a series of points connected with lines). This study dealt only with polygons, which represent the boundaries of a business' premise, and points, which represent the final position of the located individual. A GIS map attribute table was used to access parcel information for each polygon. Attribute tables contain records related to the polygon's shape, address, ownership, business type, acreage and more. However their fields are customizable and vary depending on the needs of their users. Figure 32 shows a GIS map in ArcMap, a GIS mapping program used for this algorithm, and its attribute table for a polygon.

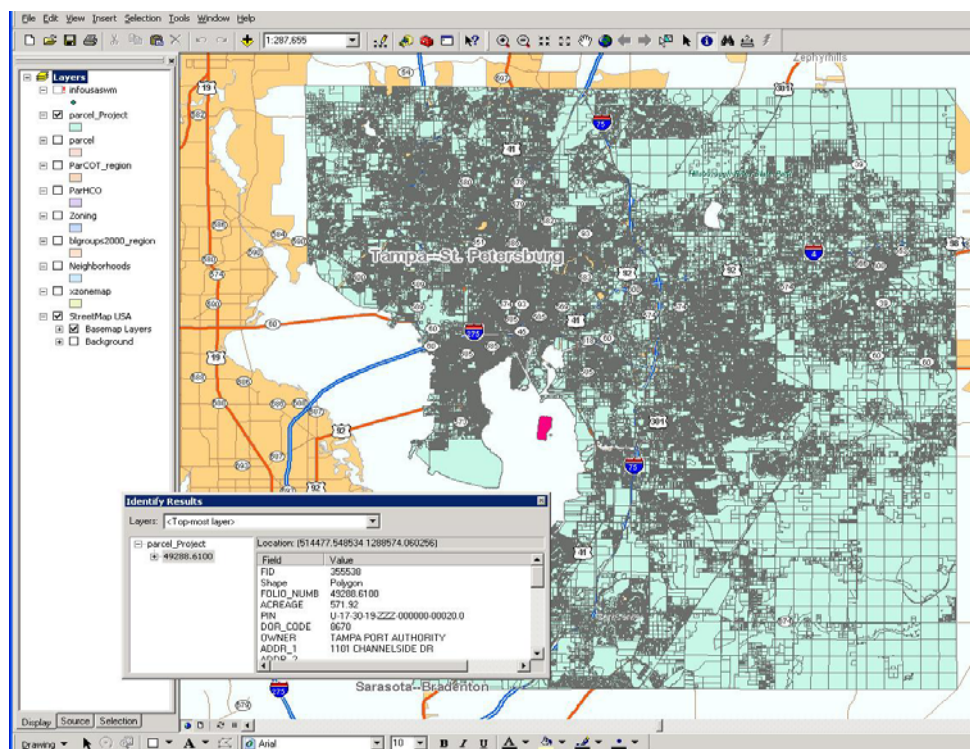


Figure 32 - Hillsborough County Parcels/Properties GIS with Algorithm

The feasibility of using GPS data from a GPS-enabled cell-phone to derive trip purpose through a Purpose Detection Algorithm was evaluated. GPS data, namely latitude and longitude, were compared to GIS parcel information for Hillsborough County to ascertain the necessary information required by the Purpose Detection Algorithm. Verification to determine if the Purpose Detection Algorithm was performing correctly was done in two ways. The first method involved comparing manually-entered trip purposes entered by users in a travel diary to those derived by the algorithm. The second method involved creating synthetic trips to determine if the algorithm would derive an appropriate purpose based on the simulated trip destination.

The TRAC-IT cell phone application was used to record trips for users. A set of pre-defined purposes were associated with individual numbers. When a user ends their trip they would select a series of numbers to specify their trip purpose. The selection process was not unlike a typical software wizard. Based on the National Household Travel Survey, glossary of travel purposes, purposes were categorized into General Purposes (shown with numbers surrounded by '(' and ')') and Specific Purposes (shown with numbers surrounded by '[' and ']') and are defined as follows (71), (72), (73), and (77):

- (1) Work
- (2) School or Religious
 - [1] Go to school
 - [2] Go to a religious activity

- [3] Go to the library: school related
- (3) Medical or Dental
- (4) Shopping and Errands
 - [4] Buy goods: groceries, clothing, house needs, etc.
 - [5] Buy services: post office, bank
 - [6] Car services: pump gas, car maintenance
 - [7] Personal or family business
 - [8] Pick up or drop off an item: dry cleaners, video rental, etc.
- (5) Social and Recreational
 - [9] Go to the gym, exercise, play sports, etc.
 - [10] Rest, relaxation, or vacation
 - [11] Visit friends or family
 - [12] Go out: entertainment, theater, sports event, bar, etc.
 - [13] Visit public place: historical site, museum, park, etc.
- (6) Transportation of Someone or Myself
 - [14] Pickup someone
 - [15] Take and wait for someone
 - [16] Drop someone off
 - [17] Change mode of transportation (at train station, bus stop)
- (7) Meals
 - [18] Go out to eat (restaurant, fast food)
 - [19] To go (fast food, coffee, restaurant takeout)
- (8) Return home
- (9) Other

As a brief example, a user would select (7) [19] if his or her trip concluded at McDonalds.

Participant selection for this study consisted primarily of USF employees. Generally, there were no exclusive criteria to qualify but selecting a diverse array of candidates with different transportation and household scenarios was paramount. This ensured that the TRAC-IT prototype would be tested under various household travel dynamics and different modes of transportation. Once a participant was selected, an account was created for them that defined a "userid." Information such as household size, home and work address, and several other factors that can only be determined manually (currently) were asked of the participants to define their *userid* characteristics. These records would later be used by the Purpose Detection Algorithm and other automated analytical TRAC-IT software.

Data collection for this study began with a GPS-enabled Motorola i870 cellular phone on the Sprint-Nextel iDEN network. Cell phones were distributed to each participant for a few hours to several days, depending on the testing period and travel circumstances. Some participants were particularly asked and directed to take a specific mode of transportation, such as a bus. Prior to their departure, participants were given a brief explanation of the TRAC-IT application operation procedures. These

consisted primarily of explaining where to find and start the application on the mobile phone, and choosing “end trip” from the application interface. The TRAC-IT application installed on the cell phone was then initialized to start a new trip. For every new trip, a tripID was created that was associated with the individual’s userid. While an individual went about his or her errands, the TRAC-IT application collected GPS coordinates for the current trip every 4 seconds. All recording terminated when the user ended his/her trip and defined his or her trip purpose.

Before the Purpose Detection Algorithm could be used, several other SQL tables that the algorithm depends on had to be created. These tables defined specific relationships between land use, and trip purpose. The most critical table called “Tblfl_hco_pa_dor_land_use_codes,” classified what are known as Department of Revenue Resource Codes (DOR codes). DOR codes are used to define what type of business a property is, for example residential or commercial. DOR codes used for this study were specifically obtained from the Florida Department of Revenue (FDOR). Table “Tblfl_hco_pa_dor_land_use_codes” contains an auto-generated primary key field (code_id), DOR code (DOR_use_code) field, a field for the property type description of the associated DOR CODE (Property_Type), and a column that was intended to be used (Use_code_id). These data were inserted into the database table exactly as they had been posted on the State of Florida Department of Revenue website, (74). Following are a few random rows of the table that show typical DOR codes (Table 6):

Table 6 - Sample DOR Codes and Their Description

CODE_ID	DOR_USE_CODE	PROPERTY_TYPE	USE_CODE_ID
1	0	VACANT	1
14	396	STUDENT HOUSING	4
241	8500	HOSPITAL	92
130	3500	TOURIST ATTRAC	42
57	1300	DEPT STORE	14
229	8200	FOREST/PK/REC	89

A table called “tblfl_hco_pa_dor_use_code_relate_purposes” was then created to associate the DOR codes to general and specific purposes. This was accomplished by copying the same code id’s from “tblfl_hco_pa_dor_land_use_codes.” The General and Specific Purposes were related to the code_ID’s using a similar number-purpose association scheme as the Smart Diary. Since each code ID was associated with a DOR CODE, the code IDs were classified by purpose based on the DOR CODE

descriptions from the FDOR website. Table 7 is an example using code_ID 14 from the previous sample table.

Table 7 - DOR Codes and Their Relationship to TRAC-IT Purposes

DOR CODE_ID	PURPOSE_ID	SPECIFIC_PURPOSE_ID
130	5	9
57	4	4
229	5	9
241	3	NULL
14	8	NULL
1	9	NULL

Once all of the tables were populated, the Purpose Detection Algorithm was ready to derive the trip purpose. The algorithm required two components to operate: *tripid*, and its associated latitude and longitude pairs. First, the algorithm was provided with a *tripid*. It then determined if the trip had ended by executing a SQL query against *TBLTRIPDATA*. If the trip had been properly ended and a GPS fix existed for the ending point, the coordinate was stored as an ArcObjects Point Feature. The algorithm then utilized a GIS map of Hillsborough County (HBC) from ArcMap. A proximity test was performed within the GIS map to determine if the distance between the trip-end coordinate and any work or home location (previously registered by the user using an address, or "geo-tagged" by the TRAC-IT application if they have visited their Home or Work while previously recording a trip) was greater than 50 meters. The proximity itself is a parameter that can be changed; 50 meters was chosen for this study through estimation. Multiple home and work locations were factored into the algorithm's creation because if a user had more than one of either the algorithm would check all proximity possibilities. If the condition of the proximity test was true the algorithm updated the trip purpose based on which work or home location was nearest to the trip-end point; "8, null" and "1, null" are the general and specific purpose for home and work respectively. Otherwise, if the proximity test was false a procedure called a spatial query was executed. Spatial queries determine any spatial relationships between geometries in an ArcMap GIS map. The spatial query used for this algorithm was a simple point-in-polygon calculation that determined what polygon the trip-end coordinate lied within on an HBC GIS map. Once the valid polygon was located, the DOR code field of the attribute table was accessed and the DOR code was retrieved by the algorithm.

The next set of steps consisted of a series of SQL queries:

- By using the retrieved DOR code, the algorithm executed an outer join query that linked together the "code_id" fields from "tblfl_hco_pa_dor_land_use_codes" and "tblfl_hco_pa_dor_use_code_relate_purposes."

- ♦ A relationship to the General and Specific purposes from "tblfl_hco_pa_dor_use_code_relate_purposes" was then established through the correlating "code_id" fields from both tables.
- ♦ Finally, once the General and Specific Purpose ID numbers had been retrieved for the TripID, a final SQL query was performed that updated TBLTRIPS.
- ♦ Based on all of the stored and processed data, the update query updated the fields "Auto_Detected_Purpose_ID."
- ♦ "Auto_Detected_Specific_Purpose_ID," "FL_HCO_PA_DOR_CODE_ID (DOR code)," and "Purpose_Detection_Completed" for TBLTRIPS.
- ♦ The field "Purpose_Detection_Completed" was updated to a value of 1 from null once the trip purpose for a particular tripid was defined. This prevents any purposes from being rewritten by the algorithm later.

The steps to retrieve the coordinate and run the algorithm are summarized briefly in the following bullets:

- ♦ Latitude and longitude pairs of the cell-phone's coordinates are determined using GPS.
- ♦ The cell phone coordinate is sent to designated lat. and long. fields in "TBLTRIPDATA" along with an associated tripid for the user.
- ♦ Provided with a tripid whose trip has ended, the algorithm retrieves the trip-end coordinates if both the latitude and longitude are greater than zero.
- ♦ A proximity test is performed within a GIS map that determines if the distance between the trip-end point and work or home location coordinate is within 50 meters.
 - o If they are, the algorithm updates the trip purpose to reflect the closest location to the trip-end coordinate.
 - o If they are not, the algorithm proceeds to the next step.
- ♦ A point-in-polygon calculation is performed within the Hillsborough County ArcMap GIS map to determine which polygon (business) the trip-end point lies within.
- ♦ The attribute table of the located polygon is accessed in the GIS map and the DOR CODE field is located.
- ♦ The DOR CODE is retrieved.
- ♦ The DOR CODE is used to query an SQL table "tblfl_hco_pa_dor_land_use_codes" to obtain a value from it's "code_id" field.
- ♦ An outer join SQL query is performed that establishes a relationship between "tblfl_hco_pa_dor_land_use_codes" and "tblfl_hco_pa_dor_use_code_relate_purposes" using the "code_id" columns from both tables.
- ♦ The general and specific purposes are then extracted for the "code_id."
- ♦ SQL table "tbltrips" is updated with the trip purpose, and DOR CODE.
- ♦ A field in "TBLTRIPS" called "Purpose_Detection_Completed" is also updated to signify that the trip's purpose is complete.

- This update is executed so overhead in future algorithm executions will be reduced.

The algorithm itself could perform in one of two ways. First, immediately after a trip has ended and its data catalogued, the trip purpose was derived by the Purpose Detection Algorithm (as shown above). The second method involved batch updating. After many trips had ended, the algorithm would check the "Purpose_Detection_Completed" field of "TBL *TRIPS*" for any trips whose purpose was not yet defined. Table "TBL *TRIPDATA*" was then queried for all the trips with a "True" in the "trip_end" field. Each trip would then be processed consecutively by tripid in the same fashion as a single trip. This processing included all of the checks and tests described in the single-update method.

"TBL *TRIPDATA*" was then queried for all the trips with a "True" in their "trip_end" field. Each trip would then be processed consecutively by tripid in the same fashion as a single trip. This processing included all of the checks and tests described in the single-update method.

A variety of applications and programming languages were used to develop and successfully run this algorithm. The algorithm was written in the Net Beans IDE 5.0 using Java and ESRI ArcObjects API. All SQL queries used by the algorithm were executed against SQL tables created in Microsoft SQL Server 2005. Point-in-polygon spatial queries were executed within ESRI's ArcMap using its Polygon and Point feature classes.

Successful matches between manually defined trip purposes and automated trip purposes are shown for synthetic trips in Table 8. Approximately 64 percent of trips coded incorrectly because of non-intuitive categorization of these businesses in the FDOR's records. For example, trip 1209 was coded as work because its DOR code defines a federally owned business. Intuitively, it is a restaurant Red Lobster, and the researchers' expectation was that it would have been coded as such (purpose: 7, 18). Similar trip situations occurred for 1212, and 1215. Trip 1212 ended at a federally owned museum that intuitively matched purpose 5, 12. Trip 1215 presents a more complex dilemma because it ended at a gas station that is also a large convenience store. Its purpose should likely have been coded as 4, 6. Last, trip 1211 should have been coded as 5, 11 because it ended at a baseball field.

Table 8 - Algorithm vs. Human for Synthetic Trips

Tripid	Automated General Purpose	Automated Specific Purpose	Correct Purpose?	Detected DOR code
1208	5	11	Y	3200
1209	1	NULL	N	8800
1210	4	4	Y	1310
1211	2	1	N	8300
1212	1	NULL	N	8630
1213	5	9	Y	HH
1214	8	NULL	Y	1630
1215	4	4	N	8400
1216	2	1	Y	8400
1217	2	1	Y	9020
1218	9	NULL	Y	2102

Despite some inaccurately-determined trip purposes, the Purpose Detection Algorithm did successfully use cell-phone originated GPS data with a GIS map to derive trip purpose. The algorithm performed well on properties classified by DOR codes that intuitively match their general use. The incorrectly-derived trip purposes were a result of DOR codes from the HBC's GIS map that did not intuitively match the use of the property. Future enhancements of the system should focus on understanding why certain properties are classified differently than their intuitive business DOR code categorization and then focus on improving the algorithm accordingly. Alternate sources of GIS data that may define land-use codes could also be investigated, including those from private sources such as Google, Navteq, or TeleAtlas. Trip purpose, as well, is largely interpretive. For instance, although most individuals would have agreed with the simulated travel diary's trip purpose, logically the Purpose Detector Algorithm's purpose made sense. Considering the baseball field is owned by a school, the derived trip purpose made sense based on the location.

Several important factors may help improve the purpose detection process in the future. Potential issues that could be targeted for improvement include discrepancies that would be a result of individuals who shop at the same place they work, drop-off's and pick-up's at locations that were not defined by an appropriate DOR code, and buildings such as strip-malls that contain numerous DOR codes. The first two scenarios could simply be resolved by calculating the duration of elapsed time for each event. These time periods would then be compared to the user's work hours or to an estimation of elapsed drop-off/pick-up time (e.g., 20 seconds). Although these scenarios had easy enough solutions the latter scenario presented a bigger challenge. Buildings such as malls and strip-malls have umbrella DOR codes that encompass all stores. In the future a more precise method might be developed to attain the DOR code of the individual store. Presently, researchers decided after some discussion that for practicality these umbrella DOR codes will be used for any trips that end in a mall

or shopping center. For instance, DOR codes 1500 and 1600 define "Regional Shopping Centers" and "Community Shopping Centers," respectively.

This study is an important first step in utilizing assisted GPS data from GPS-enabled mobile phones in advanced travel behavior surveys. Further research is required in order to understand the implications of utilizing a new type of GPS data (assisted GPS, which can include indoor fixes) that has different characteristics from traditional GPS sources.

Chapter 5. Performance Testing of GPS-enabled Mobile Phones

This section discusses performance testing of GPS-enabled mobile phones to evaluate their effectiveness for use as a transportation behavior data collection device. The testing was performed by developing customized J2ME and server-side software applications for the purpose of evaluating a specific aspect of the phone. All applications are very similar in structure to the TRAC-IT system and utilize the J2ME platform, CLDC 1.1, MIDP 2.0, and JSR179 Location API to retrieve location data from the mobile phone.

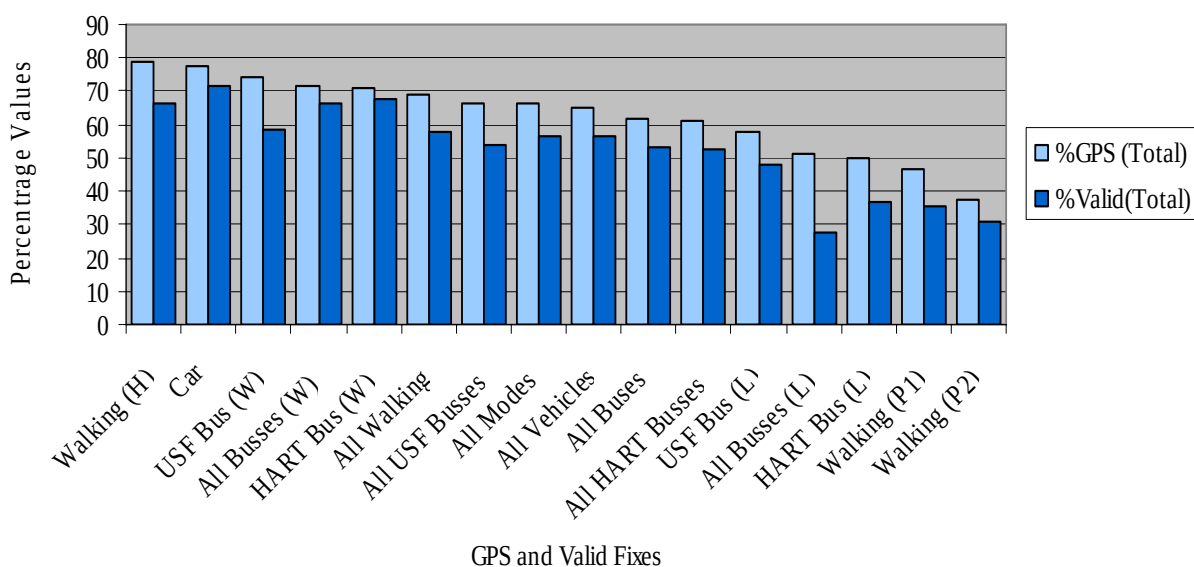
5.1 Evaluation of Transportation Environment Impact\Assisted GPS Estimated Accuracy

A fundamental requirement of location-aware information systems such as TRAC-IT is the GPS-enabled mobile phone's ability to provide highly-accurate, real-time location data and transfer this information to a remote server in a timely fashion. In these tests, a quantitative analysis of a mobile phone's ability to obtain accurate GPS data while walking, driving a vehicle, and riding public transportation is provided. The expectation was that the device would consistently obtain accurate GPS data outside in open areas, but that the accuracy may be reduced when the GPS signal is obstructed by a vehicle or the user's clothing. A total of 86 trip segments were undertaken, evenly split between walking and vehicular travel, providing a data set of 9,547 location fixes transmitted in real time at 4-second intervals by the test application. Accuracy was expressed as the percentage of data points not obtained from the cell tower compared to all data points, taking into account the GPS-enabled mobile phone's estimate of accuracy uncertainty for each individual fix using a Motorola iDEN i870 mobile phone on the Sprint-Nextel iDEN network.

Table 9 shows the data obtained. demonstrating the results of location fix attempts over a range of transportation modes while in an urban environment. Figure 33 shows a graphical representation of the results. The highest percentage of GPS fixes (79.03%) was obtained by users walking with the mobile phone open and unobstructed; walking also produced "valid" GPS data (location data estimated to be accurate within 30 meters of the true position) 66.22 percent of the time. Bus trips returned slightly less accurate results: when the phone was held near the window, GPS and valid fix percentages were 71.72 and 66.06 percent, respectively. When the phone was placed in the traveler's lap, these numbers fell to 51.28 and 27.84 percent, respectively. Car trips provided higher numbers of 77.71 and 71.61 percent, respectively.

Table 9 - Summary of Multi-Mode GPS Data

Transportation Mode	Total Trips	Total Fixes	GPS Fixes	%GPS of Total	Valid Fixes	%Valid of GPS	%Valid of Total
Walking (H)	28	2389	1888	79.029	1582	83.792	66.22
Walking (P1)	5	115	54	46.957	41	75.926	35.652
Walking (P2)	10	676	255	37.722	210	82.353	31.065
All Walking	43	3180	2197	69.088	1833	83.432	57.642
Car	18	1180	917	77.712	845	92.148	71.61
USF Bus (W)	6	483	358	74.12	283	79.05	58.592
USF Bus (L)	7	432	250	57.87	207	82.8	47.917
All USF Buses	13	915	608	66.448	490	80.592	53.552
HART Bus (W)	6	2204	1569	71.189	1492	95.092	67.695
HART Bus (L)	6	2068	1032	49.903	762	73.837	36.847
All HART Buses	12	4272	2601	60.885	2254	86.659	52.762
All Buses (W)	12	2687	1927	71.716	1775	92.112	66.059
All Buses (L)	13	2500	1282	51.28	696	54.29	27.84
All Buses	25	5187	3209	61.866	2744	85.51	52.901
All Vehicles	43	6367	4126	64.803	3589	86.985	56.369
All Modes	86	9547	6323	66.23	5422	85.75	56.793

**Figure 33 - Percentages of GPS and Valid Fixes Broken Down By Mode**

These tests conclude that it is possible to provide real-time location-aware services utilizing software applications on GPS-enabled mobile phones. In areas of heavy GPS signal obstruction such as urban canyons it may be difficult to provide an extremely reliable service unless predictive algorithms are used. For applications such as TRAC-IT that record a user's path, GPS-enabled cell phones can sufficiently provide information that can be used to reconstruct a user's travel path.

Since these tests were conducted, newer mobile phones have been released that have high-sensitivity embedded GPS chips such as the Sanyo 7050 on the Sprint-Nextel CDMA network. Additional tests performed by the research team utilizing these new phones show a significant improvement in their ability to determine their location even when in highly obstructed areas. Improvements in GPS fix accuracy while on board a moving bus showed significant improvement from the initial tests results shown in the figure and table above.

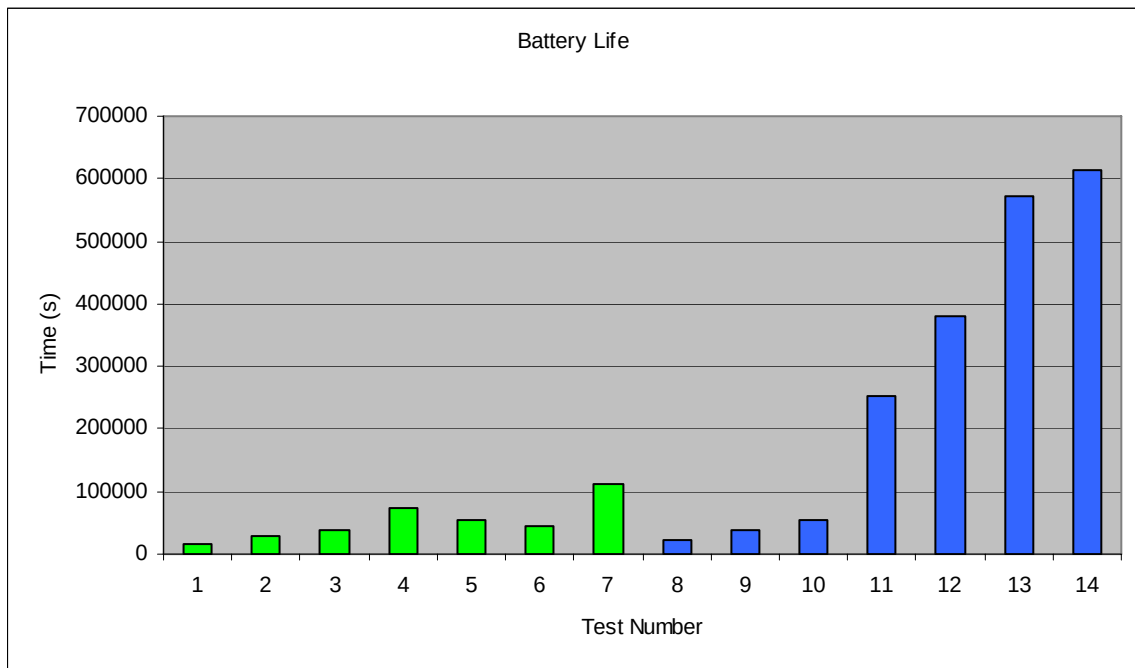
A thorough discussion of these tests is provided in a research article, "Quantifying the Position Accuracy of Real-time Multi-Modal Transportation Behavior Data Collected using GPS-Enabled Mobile Phones," available from the Transportation Research Board (82).

5.2 Impact of GPS and UDP transmissions on Battery Life of GPS-enabled Mobile Phone

Power consumption of GPS-enabled mobile phones was measured under a variety of circumstances while the phone was simultaneously receiving GPS data and transmitting the information via UDP to the remote server. A Sanyo 7050 on the Sprint-Nextel CDMA network was used for these tests with a Sanyo SCP-22LBPS 3.7V battery that was packaged with the phone. The testing conditions for each test are shown in Table 10. For each test, the phone battery charged for the duration recommended by the manual for a full battery charge and until the phone indicated that it was fully charged. Then, the settings were input for the specific test, and the phone was set to continuously run the application until it powered itself off due to lack of battery power. A timestamp was periodically recorded in the persistent storage of the phone as it transitioned through four levels of battery charge (1-4) indicated by an API that exposes battery level to the application. Once the lowest level of battery charge was reached, a timestamp was written to persistent storage on every interval completion in order to ensure that the battery life for that test was accurately recorded. It was expected that the battery life would increase as the interval between GPS fix requests also increased. It was also expected that the indoor tests would result in lesser battery life than the outdoor tests, since the high timeout values would cause the GPS chip to remain active longer while it tried unsuccessfully to obtain a GPS fix. The results of these tests are shown in Table 10 and Figure 34. As a comparison, a baseline test of the phone's battery life running in standby mode with no active J2ME applications lasted 331980 seconds or approximately 92.22 hours.

Table 10 - Battery Test Results using Sanyo 7050 Mobile Phone

	Test #	Interval / Timeout / MaxAge (seconds)	Battery Life (seconds)	Battery Life (minutes)	Battery Life (hours)
Indoors	1	4/2/2	17,550	292.5	4.875
	2	30/15/15	27,996	466.6	7.77
	3	60/30/30	39,933	665.55	11.09
	4	300/150/150	73,947	1232.45	20.54
	5	600/300/300	55,692	928.2	15.47
	6	1800/900/900	45,678	761.3	12.69
	7	3600/1800/1800	112,573	1876.22	31.27
Outdoors	8	4/2/2	22,289	371.48	6.19
	9	30/15/15	37,654	627.57	10.46
	10	60/30/30	55,193	919.88	15.33
	11	300/150/150	252,480	4208	70.13
	12	600/300/300	381,664	6361.07	106.02
	13	1800/900/900	571,395	9523.25	158.72
	14	3600/1800/1800	614,014	10233.57	170.56

**Figure 34 - Battery Tests Results using Sanyo 7050 Mobile Phone**

The tests that were run outdoors, where access to the GPS satellite signals was less obstructed, revealed a clear pattern of increasing battery life as the interval of time between location requests also increased. As the interval, timeout, and maximum age values increased the length of activity time for the mobile phones increased correspondingly. At higher polling frequencies, the battery life results are similar to those of the indoor tests, although the higher intervals do produce slightly higher numbers. At intervals greater than five minutes, however, the difference between the two conditions becomes very apparent.

The results indicate that, at the intervals tested, the indoor trials provided a relatively small difference in battery life. Except for the 3600 second duration, all fell below 28 hours (roughly 100,000 seconds), and no clear trend was indicated. As expected, this likely difference in results between indoor and outdoor tests is the percentage of successful position signals received from the GPS satellites. In the indoor tests, with greater environmental obstructions, the mobile phone hardware must expend greater effort over a longer period of time, in an attempt to obtain these signals, and this may account for the increased use of battery power, leading to the shorter duration observed. Further indoor tests will be run in the future with a timeout value no greater than 30 seconds for all tests. It is expected that with smaller timeout values, the trend in indoor tests will become more significant.

These results show that, when possible, the interval between location requests placed by the J2ME application using the JSR179 Location API should be increased to avoid negatively impacting the phone's battery levels. If a GPS fix is requested every 4 seconds and that information is transmitted instantly to the server using UDP, the battery of the Sanyo 7050 would last from around 4.9 to 6.2 hours depending on how often the individual was indoors. When the phone is detected to be indoors and a GPS fix cannot be obtained, the software should reduce the frequency of GPS requests until it is determined that the phone is back in GPS coverage. Future tests will focus on differentiating the energy cost of using GPS from energy cost of sending the fix via UDP. Since each action costs energy, it is expected that by transmitting only important or "critical" GPS fixes and discarding "non-critical" fixes additional energy can be saved. These results are extremely important if a passive version of the TRAC-IT application is going to be deployed that is always recording user behavior without manual input from the user. For the active version of TRAC-IT that requires the user to manually start and stop recording his or her behavior, this information can still lead to additional energy savings but is not as critical since the application will only be actively consuming large amounts of power when the user has indicated that they are traveling. From current tests, the active version of TRAC-IT that is manually started and stopped by the user did not noticeably impact battery life if the phone was recharged on a nightly basis.

5.3 Comparison of GPS Fix Times and Estimated Accuracies Returned By JSR179 Location API and Motorola OEM Position API on iDEN Phones

Several methods of obtaining GPS coordinates by mobile communication devices have been developed in recent years. Some companies, such as Motorola, have developed proprietary Application Programming Interfaces (APIs) for the purpose of receiving data from GPS hardware. These methods provide a way to receive coordinates and other related information from positioning satellites, which may then be used in a variety of applications. Software implementations of each API are different, and therefore may exhibit different behaviors even when utilizing the same underlying hardware. It is important to understand these tradeoffs when selecting an API for software development on a mobile phone, but to date there has been little quantitative data that illustrates performance differences.

These tests illustrate a comparison of two such APIs, the Motorola OEM Position API and JSR179 Location API on iDEN phones. The tests focused on the length of time for obtaining fixes and the estimated accuracy of obtained data. A Motorola iDEN i580 phone on the Sprint-Nextel iDEN network was utilized for these tests. Input settings to each API were selected to be as similar as possible on each platform.

The results shown in Figures 35 and 36 indicate that the JSR179 Location API, after initial activation, retrieves GPS data in a timelier manner (with a 34% difference in fix times) than the Motorola Position implementation that was also studied; however, the coordinates obtained thereby are estimated to be less accurate with a 69 percent difference.

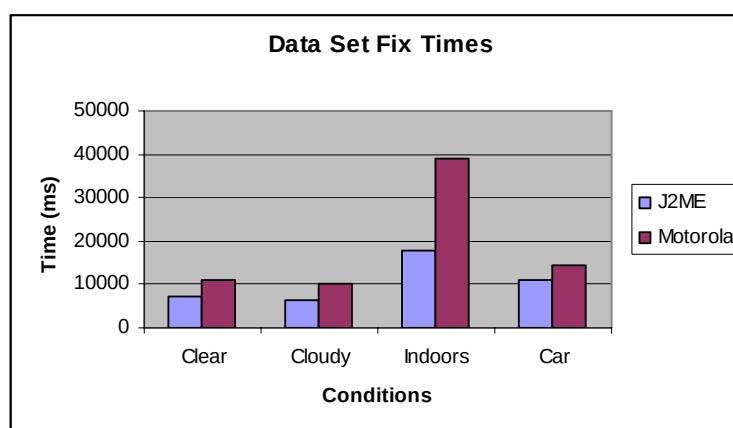


Figure 35 - Fix Time J2ME JSR 179 Location API Versus Motorola OEM Position API

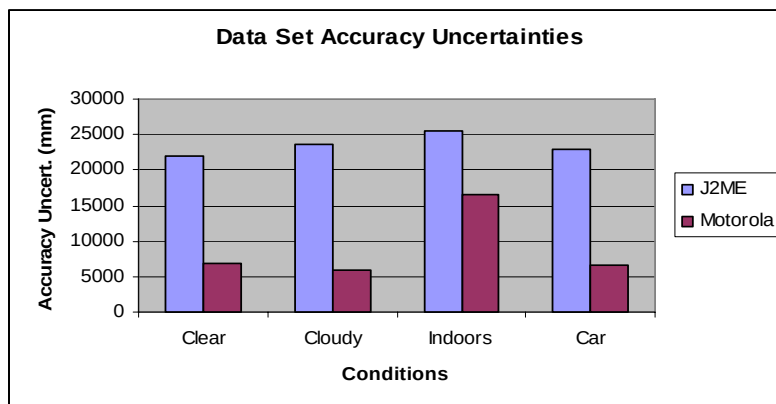


Figure 36 - Estimated Accuracy Uncertainty J2ME JSR179 Location vs. Motorola OEM Position

As mentioned in Chapter 2, "Technology Assessment," the use of the JSR179 Location API has a significant advantage of being portable to mobile phones by many different manufacturers. Therefore, unless there are significant advantages to utilizing a proprietary API such as the Motorola OEM Position API the JSR179 Location API is preferred. It should be noted that these test results cannot be generalized to other JSR179 implementations on other manufacturer's mobile phones (i.e., non-iDEN phones), since the software that implements the JSR179 Location API on that particular phone may be different. Therefore, further tests are required to quantify the difference between JSR179 Location API implementations on other manufacturers' phones and proprietary APIs, one example being Qualcomm's Java Application Extensions (QJAE) API for devices utilizing Qualcomm chipsets on CDMA networks (21). The results in these tests can be generalized to other iDEN handsets as long as it can be confirmed that the handset has the same software implementation of the Motorola OEM Position API and JSR179 Location API and that the GPS hardware of the phone is the same as that i580 used in these tests.

A detailed research paper on these specific tests, results, and analyses, "A Comparison of Fix Times and Estimated Accuracies in Application Programming Interfaces (APIs) for GPS Enabled Mobile Phones," was published in the *Proceedings of the 11th World Conference on Transport Research*, Berkeley, CA, USA, June 2007 (22).

Chapter 6. TRAC-IT Field Testing

The following sections discuss the field testing of the TRAC-IT system for Phase 3 of the project including the survey protocol, data collection process, and analysis of results.

6.1 Survey Method

A total of 14 volunteers were individually solicited to participate in the TRAC-IT Phase 3 field tests. An attempt was made to solicit the same individuals that participated in the TRAC-IT Phase 1 field tests. These individuals were provided the letter of solicitation shown in Appendix A, and then provided the Informed Consent document shown in Appendix B for review. Once the participant signed the Informed Consent form and agreed to participate in the testing, a form (Appendix C) was given to the individual that asked him or her to specify a short description of several locations that they visited regularly. These locations were entered into the database by the research team to save the user the time of inputting these locations into the phone user interface as “new” locations. As a result, when the participant first started the TRAC-IT application and selected to begin a trip, the descriptions of location provided by the user were already available for selection as a trip starting or ending location.

A number of mobile phones, including models Sanyo SCP-7050 on the Sprint-Nextel CDMA network and Motorola iDEN i580 on the Sprint-Nextel iDEN network, were distributed to the 14 volunteers. Only one volunteer had a personal mobile phone plan with Sprint-Nextel and therefore qualified to have TRAC-IT installed on the same phone that they used for personal calls. However, this volunteer’s mobile phone was not compatible with TRAC-IT. The research team replaced their personal mobile phone with a Sanyo 7050 purchased by the research team and added an unlimited data plan for \$15 per month to their account so TRAC-IT could be tested on their personal phone.

The TRAC-IT application was set in active mode (i.e., not passive tracking mode) so it would record only GPS data when the user selected that they were starting a trip via the user interface. The application was set to query the GPS position of the phone and send this information to the server every four seconds once the user indicated that a trip had begun. All location data (i.e., not just critical points) calculated by the phone was sent to the server to ensure an adequate amount of data was collected during the tests. A belt clip and charger were also provided to the volunteers. Car chargers were also offered to volunteers, although only two volunteers opted to use them.

The volunteers were asked to carry the cell phones around for a period of time that averaged between 2 and 3 weeks and to charge the phone battery nightly. Participants were given instructions regarding the user interface and were encouraged to log their travel behavior during the course of daily activities. No specific instructions

were given to the user for where to position the phone during trip recording. In other surveys involving GPS, the users are often told to position the device on the dashboard of the vehicle in order to maximize the potential for the device to receive adequate GPS signals in order to calculate a position. However, since the cell phone is a personal device frequently used for other activities such as voice calls, it is not practical to request the volunteer to position the phone in this manner. The research team wanted an accurate representation of what type of GPS data could be collected when the mobile phone is carried as a normal phone, and therefore no specific instructions for locating or carrying the phone during trips was provided.

The location of the subjects' activity was primarily in the University of South Florida area of Tampa, although some trips were recorded in Seattle, Fort Lauderdale, and Orlando, to test TRAC-IT in other geographic locations. A variety of modes were utilized, including walking and travel by bus, with car trips being the most common.

For these field tests, the State Machine and Critical Point algorithm on the phone were both turned off to avoid interfering with data collection, since extensive testing of those software modules was not possible before the field tests. Passive tracking was not used because the State Machine and Critical Points algorithms are required to offset the negatively impact of passive tracking on the mobile phone's battery life.

6.2 Trip Information Feedback to Participants

For each trip recorded using TRAC-IT, participants were emailed trip data that included a description based upon their use of the TRAC-IT interface, as well as a graphical representation of each trip. Participants who used Microsoft Outlook were emailed their trip information in the form of a meeting invitation so it could easily be saved to their calendar for future reference. Additionally, participants could easily accept or reject the information as an appropriate representation of their actual travel behavior and provide comments back to the research team by replying to the message. A meeting invitation reflecting a participant's trip is shown in Figure 37.

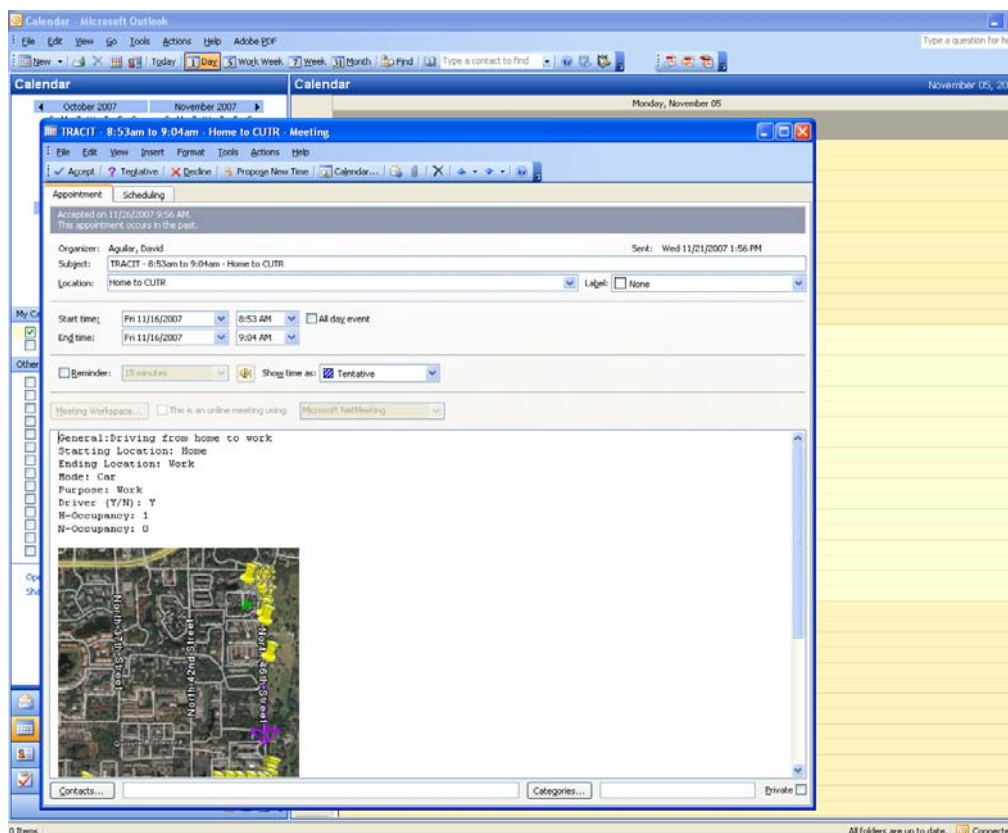


Figure 37 - TRAC-IT Trip in MSOutlook Meeting Invitation Format

The trip description included date and time of the trip segment, as well as the user-defined start and end locations. Mode purpose and occupancy information was provided as well. The graphical representations were generated by creating a .kml file from the geographic and user-supplied data present in the server database using a custom application developed by the research team. These files, which are opened in Google™ Earth, display visual markers of the GPS fixes superimposed over satellite images of the location. The feedback provided to users is a screenshot of this picture, with the start of the trip (a green triangle) and its end point (a red square) marked according to the initial and final data points. A close-up view of this data is shown in Figure 38.

To reduce clutter on the graphical representation of the trip, only “critical point” GPS fixes were shown to the user, and then a path was drawn on the graphic to indicate the direction of travel as recorded in the GPS data. Therefore, it should be noted that the figures showing trips do not reflect all GPS data collected for that particular trip and that additional GPS were recorded by TRAC-IT that lie along the straight line portions of the trips. If the “Send Critical Points Only” option was turned on in the TRAC-IT mobile application settings, then only the points shown on in the email would have been recorded. For testing with participants, all GPS data was collected to ensure an adequate amount of GPS information was collected. The portions of trip segments that lacked consistent GPS fixes due to loss of signal or transmission errors were also

indicated by a directed path (a purple arrow) linking the two GPS points on either side of the gap. In a few cases, GPS fixes were not obtained near intersections where the vehicle turned and therefore a straight line between the GPS points on either side of a gap would not have followed the road network. In these cases, a line was drawn showing the most likely route taken in the interim between the two points. It is expected that this task could be automated in future TRAC-IT field tests by shortest path algorithms that make use of road network information.

General: Driving from Home to Work
 Starting Location: Home
 Ending Location: work
 Mode: Car
 Purpose: Work
 Driver (Y/N): y
 H-Occupancy: 1
 N-Occupancy: 0

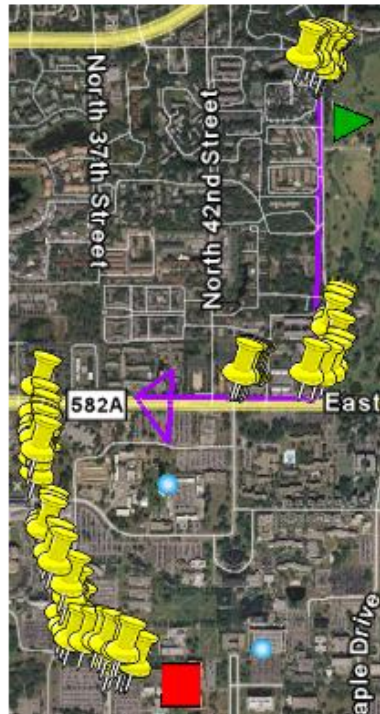


Figure 38 - TRAC-IT Close-up View of Trip Data (Only Critical Points shown)

Participants were encouraged to provide feedback to the developers regarding the accuracy of the data provided as well as the advice generated by the Expert System. Development and testing of the system is documented in the TRAC-IT Phase 2 Final Report.

6.3 Results

During the initial phase of field testing for TRAC-IT, 317 trips were recorded in the remote database by 14 volunteers spread out over a period of three months resulting in the collection of 66,523 location data points. Table 11 shows a partial record of information collected in the database for each trip, and Table 12 shows a subset of location data points that were recorded for a single trip.

Table 11- Sample Trip Information collected during the TRAC-IT Field Tests

TripID	2763	2764	2766	2767	2768
UserID	1	5	24	24	5
ModeID	1	6	67	67	6
ModeTypeID	1	3	8	8	3
ModeType	Car	Minivan	walk	walk	Minivan
PurposeID	1	7	1	1	8
Purpose	Work or Work-Related	Meals - Go Out /Get Take Out	Work or Work-Related	Work or Work-Related	Return Home
SpecificPurposeID	NULL	19	NULL	NULL	NULL
SpecificPurpose	To go (fast food, coffee, restaurant takeout)				
StartDateTime	12/17/2007 8:52:58 AM	12/17/2007 12:18:09 PM	12/17/2007 2:30:08 PM	12/17/2007 2:34:32 PM	12/17/2007 5:48:38 PM
EndDateTime	12/17/2007 9:04:13 AM	12/17/2007 12:27:17 PM	12/17/2007 2:34:26 PM	12/17/2007 2:41:40 PM	12/17/2007 6:07:28 PM
StartLocationID	67	474	255	266	476
StartLocationDescription	Home	Work	Work	USF Library	Work
StartLatitude	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx
StartLongitude	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx
EndLocationID	306	475	266	255	477
EndLocationDescription	CUTR	Restaurant	USF Library	Work	Home
EndLatitude	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx
EndLongitude	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxxx
OccupancyHousehold	1	1	NULL	NULL	1
Occupancy Non-Household	0	0	NULL	NULL	0
Driver	TRUE	TRUE	FALSE	FALSE	FALSE
TotalTime (ms)	7E+05	547792	257732	428231	1E+06
TotalDistance (miles)	2.499	1.7521	0.212497	0.38434	5.2103
FeedbackFlag	FALSE	FALSE	FALSE	FALSE	FALSE
ChainID	NULL	2764	2766	2766	NULL
Chain_link	FALSE	TRUE	TRUE	TRUE	FALSE
Chain_End	FALSE	FALSE	FALSE	TRUE	FALSE
Valid	TRUE	TRUE	TRUE	TRUE	TRUE

Table 12 - Sample Location Data Points Recorded During Trip

Altitude	horizontalAccuracy (m)	verticalAccuracy (m)	LocationMethod	LocationMethodDesc.	Fix_Valid	DateTime GPS Fix	DateTime PhoneSent	DateTime ServerReceived	cellSignalStrength	batteryLevel	CriticalPointPhone
486	20	24	327681	AssistedGPS	1	12/17/2007 12:24:23 PM	12/17/2007 12:24:26 PM	12/17/2007 12:24:25 PM	-1	3	0
490	26.83282	24	327681	AssistedGPS	1	12/17/2007 12:24:28 PM	12/17/2007 12:24:30 PM	12/17/2007 12:24:29 PM	-1	3	0
489	20	24	327681	AssistedGPS	1	12/17/2007 12:24:31 PM	12/17/2007 12:24:34 PM	12/17/2007 12:24:33 PM	-1	3	0
487	20	24	327681	AssistedGPS	1	12/17/2007 12:24:35 PM	12/17/2007 12:24:38 PM	12/17/2007 12:24:37 PM	-1	3	0
481	20	24	327681	AssistedGPS	1	12/17/2007 12:24:39 PM	12/17/2007 12:24:42 PM	12/17/2007 12:24:41 PM	-1	3	0
483	26.83282	24	327681	AssistedGPS	1	12/17/2007 12:24:43 PM	12/17/2007 12:24:46 PM	12/17/2007 12:24:45 PM	-1	3	0
488	26.83282	24	327681	AssistedGPS	1	12/17/2007 12:24:48 PM	12/17/2007 12:24:50 PM	12/17/2007 12:24:49 PM	-1	3	0
488	20	24	327681	AssistedGPS	1	12/17/2007 12:24:51 PM	12/17/2007 12:24:54 PM	12/17/2007 12:24:53 PM	-1	3	0
488	20	24	327681	AssistedGPS	1	12/17/2007 12:24:55 PM	12/17/2007 12:24:58 PM	12/17/2007 12:24:57 PM	-1	3	0

Of these location data, 53,290 data points represented GPS data points with an associated latitude and longitude and estimated accuracy. The remaining 13,233 location points represent the location of the cell tower the phone was communicating with when a GPS fix was not possible.

Note that additional data are also recorded for system operations and executions of algorithms such as Purpose Detection and Path Prediction but are not shown. Also, certain identifying information, such as the latitude and longitude of locations marked "Home" and "Work" have been omitted to protect the privacy of the participant.

6.4 Discussion and Analysis of Results

Of the 317 trip segments recorded by the participants, only 7 of the resulting records were reported by participants as an inaccurate assessment of travel behavior. Five of these appeared by inspection of the recorded data and subject interaction to be errors in user input via the phone user-interface when labeling locations. In the other two cases, the GPS reception faded near the ending location, and therefore it appeared on the map that the ending location was earlier in the trip. In 34 cases, there were

insufficient GPS data to provide meaningful feedback to the volunteers for verification of the trip path. In 32 cases, most of them overlapping the 34 instances of no GPS being recorded, the end trip information recorded by the participant using the user interface of the phone was lost and not recorded in the database due to technical issues with the TRAC-IT system discussed in a following section. These intermittent technical issues also prevented users from logging in and starting the TRAC-IT application at times, so an unknown number of trips were not recorded due to these technical problems. While recruitment of additional participants beyond the initial phase of recruits was planned when field tests began, once these technical issues were observed the TRAC-IT research team focused on troubleshooting tasks and maximizing the amount of data collected from this single phase instead of deploying additional phases that could potentially exacerbate the problem. Despite these technical difficulties, there was still a significant amount of trip and GPS data collected that demonstrated the proof-of-concept of the TRAC-IT system.

Trips made by certain modes seemed to have reoccurring GPS characteristics for different users that reflect the initial observations made during the performance testing of the GPS-enabled mobile phones for estimated accuracy uncertainty. For example, walking trips (Figure 39) sometimes have frequent GPS gaps in the path since the phone is usually obstructed by a bag or the user's clothing. However, some walking trips were observed that were a perfect recreation of the users path. Car trips usually had the most accurate portrayal of the user's path (Figure 40). This is likely because the user tends to physically place the phone in the cup holder or other location in a car where it has a clear view of the sky. However, occasional GPS gaps do occur. Bus trips showed characteristics somewhere between car and walking trips depending on the specific bus route and area. Figure 41 shows a bus trip that had very good GPS reception and shows the route in extreme detail. A few trips were also recorded using a golf cart by one TRAC-IT participant (Figure 42). The GPS data recorded during these trips seem to have properties somewhere between walking and car trips.

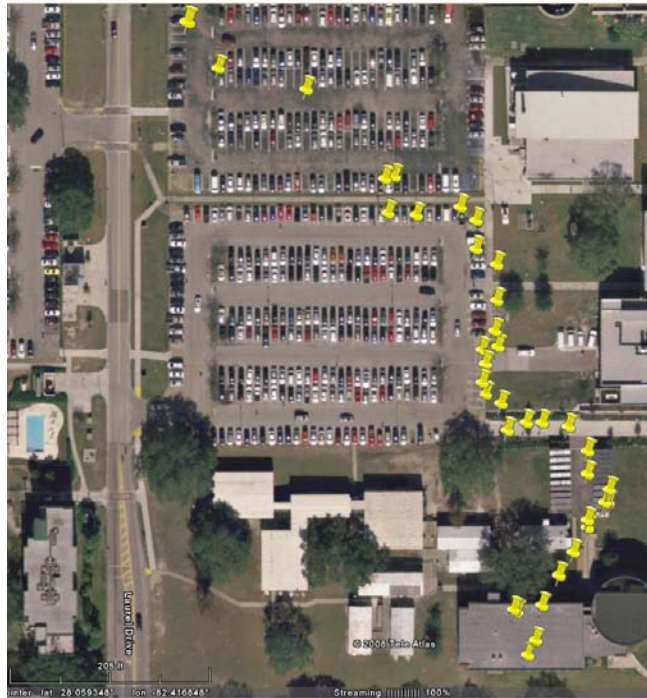


Figure 39 - Walking Trip Recorded by TRAC-IT Participant Shows all GPS Points Recorded

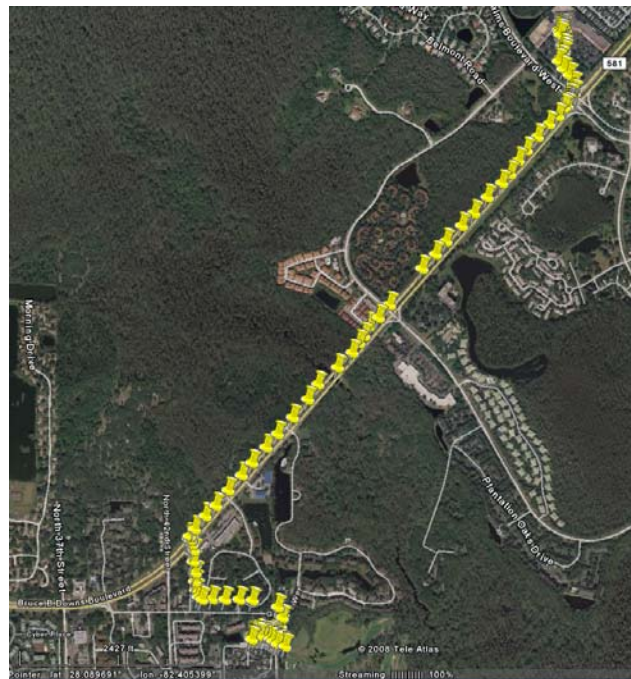


Figure 40 - Car Trip Recorded by TRAC-IT Participant Shows all GPS Points Recorded

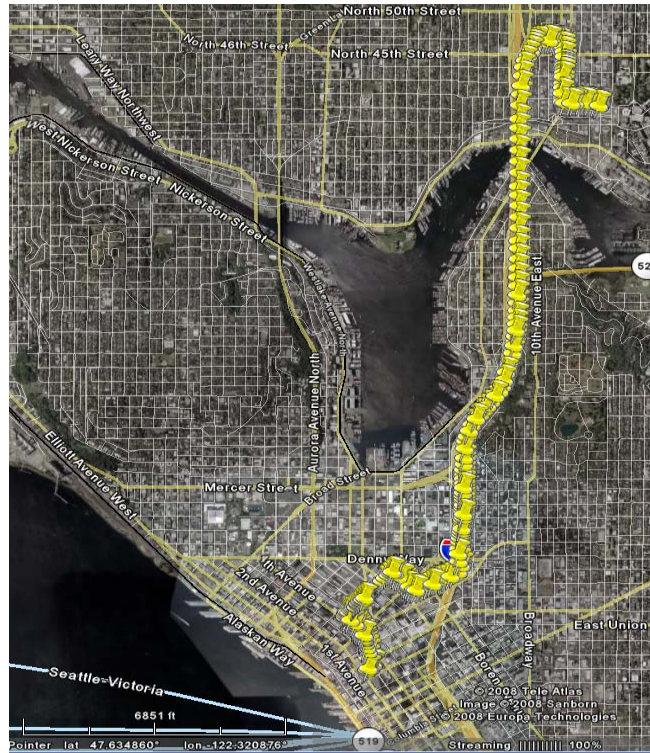


Figure 41 - Bus Trip Recorded by TRAC-IT Participant - Shows all GPS Points Recorded

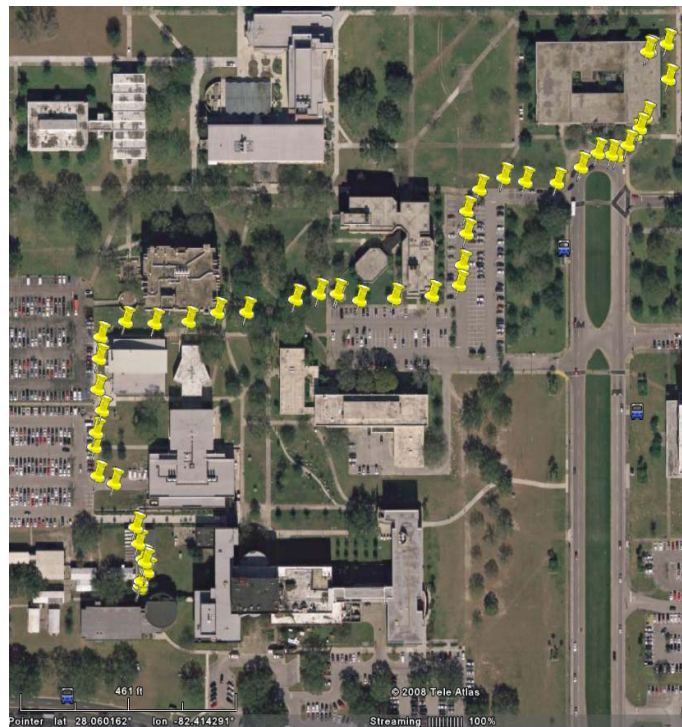
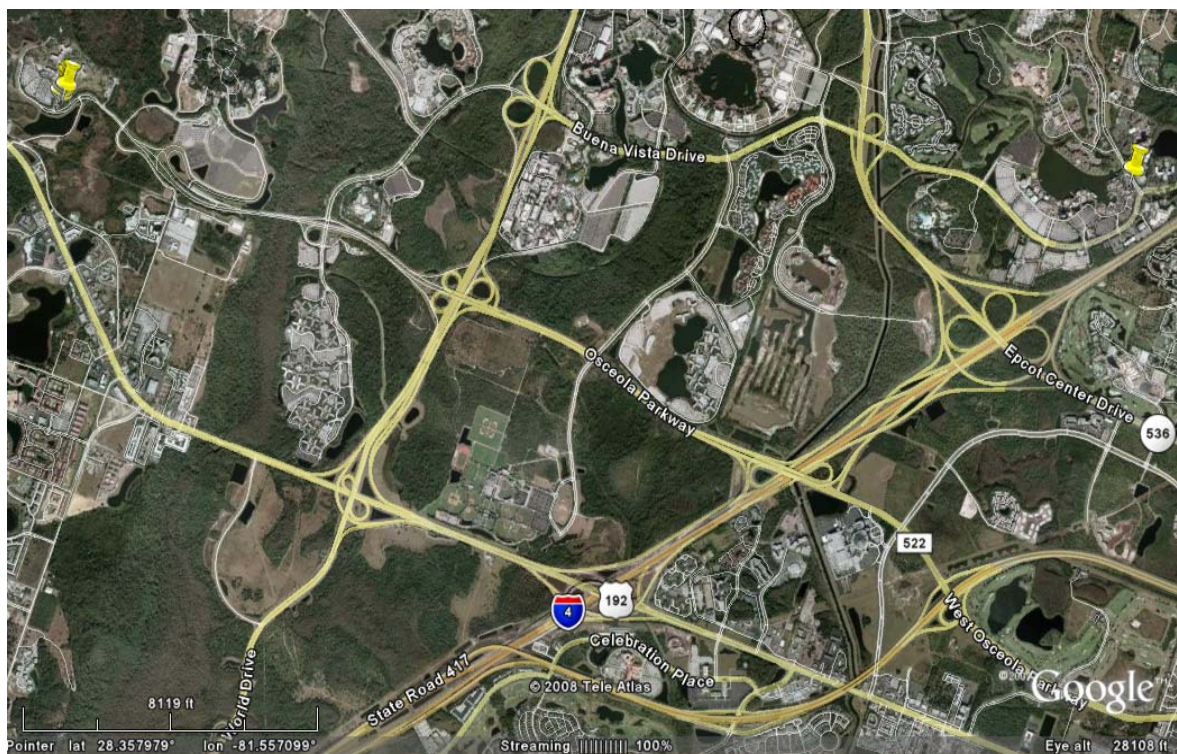


Figure 42 - Golf Cart Trip Recorded by TRAC-IT Participant Shows all GPS Points Recorded

For trips where GPS was not available (i.e., a travel path was not recorded), at least one GPS fix was almost always recorded at the starting and at the ending locations of the user's trip (Figure 43). This may be because the user must hold the cell phone to input information about the start and stop locations of the trip via the user interface and the user will likely keep the phone out in the open (i.e., not in a bag or other concealed location) for at least several seconds during this time. While the phone is out in the open, it is more likely to receive GPS signals and obtain a position fix. This characteristic of TRAC-IT seems to indicate that origin and destination information will be recorded for the vast majority of trips. Origin and destination information could also be extracted from user's manual entry of starting and ending location information. Each time the user visits a location, it is "stamped" with latitude and longitude information if GPS is available at that time. Therefore, for future trips to the same location when GPS may not be available, latitude and longitude information can still be extracted from the database for the location the user indicates via the user interface. This process was possible for several few trips recorded during field testing that lacked any GPS data.



**Figure 43 - Bus Trip where GPS Path was not Recorded
but O/D GPS Points were Recorded**

6.4.1 Qualitative Volunteer Feedback

When the participant completed the survey period and turned the cell phone back into the research team, they were given the opportunity to give feedback to the research

team regarding their experience using the TRAC-IT application. The number one issue reported by the participants was the issue of remembering to start recording a trip by using the TRAC-IT user interface. Interestingly, most participants found that they remembered to carry the phone with them, but simply forgot to start recording their trip. Comments to the research team included the question of why the phone would not always record their travel behavior so they would not have to manually start and stop the trips. This feedback may indicate that users are willing to be constantly monitored by a passive version of TRAC-IT as long as they can turn the tracking feature off when they do not want to be tracked. The “active” version of TRAC-IT used in these tests assume that users would rather “opt in” for each trip to be recorded by starting the recording manually via the user interface for privacy reasons, since the individual could always claim that they forgot to start the application if they did not want the research team to know why they chose not to record a trip. Further research in this area is required to determine if passive tracking can occur without negatively impacting the phone’s resources, which would require further analysis of the state machine and critical point algorithms in use.

Other comments include the recommendation to allow users to enter trips they forgot to record via a web page or as part of the feedback process. A related idea was to allow the editing of trips for situations where the user forgets to begin or end a trip at the appropriate time. An example provided to the research team was the user leaving their work location and then remembering that they should record their trip after around one minute had passed. Another example was provided by another participant of forgetting to end a trip when they arrived home, and therefore they received an email the next day showing their trip last a period of eight hours. Therefore, TRAC-IT seems to serve as a good initial record of travel behavior that could be edited by the participant after they receive the feedback showing their recorded trip.

Several users expressed confusion with the purpose categories and that they had trouble classifying some of their trips with the given purposes. Sample scenarios include going to a football game, or taking a trip that could have been classified in several categories. An example scenario of this is someone traveling from work to their home to eat lunch, since neither purpose of “Return Home” or “Eat Out” seems to fit the activity.

Some users also said that they did not like using the cell phone key pad to input text descriptions of locations when they visited them. One user also asked if future versions of TRAC-IT could include voice recognition or audio recordings of location descriptions that could be used in place of text entry.

6.4.2 Analysis of Technical Issues Discovered During Field Tests

There were many technical issues discovered during the field testing of the TRAC-IT system since there are many components to the system that must work perfectly in order for the system to function correctly. For example, a end trip information not

being successfully recorded at the database can be the result of many issues – lack of wireless coverage at ending location, the cell phone latching onto roaming networks that do not guarantee data service near the end location, user error in forgetting to end the trip, software errors in the TRAC-IT mobile application, software (firmware) issues in the mobile phone, software errors in the TRAC-IT server application, software issues in the Java application server, connection pool issues in the Java application server, JDBC driver issues in the connection between the Java application server and SQL Server database, and software issues in the SQL Server database.

While the most frequent cause of lack of GPS information is a GPS signal not being available, other issues can affect the ability to obtain a GPS fix and transfer this information to the server: lack of wireless coverage at a location (since the phone must communicate with a location server before obtaining a GPS fix), the phone roaming on networks that do not guarantee support of UDP (which would result in a GPS fix being obtained, but not being transferred to the server), wireless or cellular network issues causing excessive loss or corruption of UDP packets, an error in the TRAC-IT mobile phone software, an error in the firmware of the mobile phone that would prevent network communication from being successful, an error in the TRAC-IT server application that prevents the data from being inserted successfully into the database, issues with the connection pools in the Java application server, software issues in the Java application server causing temporary crashes in the server, and SQL Server issues preventing the insertion of data. It is estimated that these errors contribute to some of the cell tower location fixes in place of GPS. Additionally, in several cases users reported that they occasionally forgot to end a trip and proceeded to carry the mobile phone indoors. This situation resulted in location data for the cell tower being continuously sent to the server since a GPS fix was not possible. In future tests, the State machine and Critical Point algorithms running the mobile phone should help prevent excessive amounts of this type of data from being sent to the server.

Most technical issues were resolved before the TRAC-IT field tests, but certain issues such as SQL table locking during multiple simultaneous accesses as well as connection pool issues resulting from multiple simultaneous connected devices did not appear until the field tests since lab testing could only simulate a limited number of concurrent phone sessions. These two issues resulted in the TRAC-IT server occasionally locking up where two phone sessions actions happened to occur simultaneously, resulting in the lost end trip data of 32 trips. Occasional server crashes also resulted in the loss of incoming GPS data in the form of UDP packets until the server was brought back online. Due to the design of the TRAC-IT system these crashes were transparent to the mobile phone application and the user could continue through the survey process as normal, but as mentioned GPS packet loss did occur during the crash and reboot.

Certain locations also seemed to create problems not observed in earlier tests. For example, certain users had more trouble with server connectivity than others. One user reported lack of server connectivity frequently, but also stated that they also had problems with their own phone near their home location, which was on a wireless

carrier other than Sprint. Some users also reported intermittent GPS gaps in portions of their trips on their travel feedback maps in specific areas. Repeated GPS gaps in specific areas could indicate issues with UDP transmissions in that area. For one user that reported this problem, the research team instructed the user to set the phone's options to prevent roaming on other wireless carrier networks even if other carrier's networks had better coverage in the area. The resulting data showed that GPS data was successfully collected in the problem areas. While not conclusive, this may indicate that UDP transmissions have a higher probability of failure when the phone is roaming on other wireless carriers. Future tests should provide more information on this issue.

Most of the above issues are resolvable with additional testing, troubleshooting and debugging, and TRAC-IT software errors found during testing are believed to be resolved as of the current date. An upgraded version of the JDBC driver and Java Application Server used also seemed to resolve some issues of server instability. Future enhancements to TRAC-IT should include additional features such as the logging of counts of UDP packets sent from the phone to know how many UDP packets are lost for each trip in specific areas. Other simple additions such as status condition printouts in critical areas of server code will allow for better debugging of seemingly random errors that occur due to complex multithreaded server applications.

Perhaps the most important feature that should be included in a future version of TRAC-IT is the ability to temporarily cache trip and GPS data in persistent storage on the phone in locations where wireless coverage is not available or server connectivity is not currently possible. This feature would help avoid many key issues that resulted in the loss of trip and GPS data including the lack of current wireless coverage and server down time due to all of the above mentioned possible problems. A temporary caching feature must be implemented carefully, since many J2ME mobile phones do not have very much memory for internal storage of data. Therefore, caching algorithms must be highly efficient and should only cache data when absolutely necessary. This feature should be investigated further in future versions of TRAC-IT.

6.4.3 TRAC-IT Deployment

There are many issues related to actually deploying the TRAC-IT mobile software application to GPS-enabled mobile phones in the public, including access restrictions by wireless carriers as well as the process of transferring the application to be installed on the phone. This section of the report focuses on two categories of deployment: 1) development and 2) production.

Development Deployments

Software development deployments are usually small scale for research and testing purposes. In the case of developmental deployments of TRAC-IT for research purposes (up to around 200 active phones), USF can actively manage the application

and the related phones on the Sprint-Nextel CDMA network. Sprint CDMA devices support Over-The-Air (OTA) installation, meaning that a J2ME application can be downloaded from a website and automatically installed on the device. On the Sprint-Nextel iDEN network, the application must be digitally signed by Sprint-Nextel; otherwise, the application is limited to 48 hours of actively before it must be re-installed on the phone. Also, access to an online web management tool managed by Sprint-Nextel is required for remote deployment to iDEN phones; otherwise, the device must be connected to a laptop with a USB cable for installation of the mobile application. Verizon Wireless and AT&T are currently closed to all third-party development for location-based services and did not respond to the research team's requests for further information. Alltel allows BREW development on its network for location-based services for certain companies but did not respond to the research team's requests for further information. T-Mobile could not be reached for comment since its developer website does not appear to be operational.

Production Deployments

For wide-scale production deployment of TRAC-IT to the general public, permission of the wireless carrier will be required. Most U.S. wireless carriers require that production LBS applications be digitally signed by the carrier to execute on the phone. According to the wireless carriers, this mandate assures that the application has passed certain testing and certification requirements before an application is widely available to the general public and helps in preventing the spread of viruses as well as quality assurance of the mobile applications.

It should be noted that each wireless carrier has its own restrictions when working with Location Based Services, and therefore the agencies conducting wide-scale deployments of TRAC-IT will have to work closely with the wireless carrier to ensure a smooth deployment. Many privacy implications exist with a tracking application such as TRAC-IT for the general public, and therefore all involved parties must ensure that the user is correctly informed about how his or her behavior is going to be recorded and used to ensure that his or her privacy is not violated. CTIA®, a non-profit advocate group for the wireless industry, has created a "Location Based Services Action Team" that is looking to develop a "Best Practices and Recommendations" guide for location-aware applications that should be followed by application developers (84).

Some of these guidelines include making the customer opt-in before they can be tracked, giving the user multiple periodic notifications that they are being tracked, and notifying the customer how location is being used and who has access to the information.

As mentioned earlier, there does seem to be promise that the carriers that are currently closed to LBS will open in the near future. Specifically, AT&T and Verizon have each announced intentions to allow access to LBS to devices with embedded GPS in the near future, although no specific timeline has been announced.

6.5 Status of Current Research

The TRAC-IT system continues to be tested internally by the research team, including testing of the application on personal cell phones to evaluate the impact of the application on items like blocking incoming calls, battery life, and user privacy.

New phones also are being tested by the research team as they become available to assess their compatibility with TRAC-IT.

As of early 2008, deployment of TRAC-IT could take place only on Sprint-Nextel's iDEN network as well as Sprint-Nextel's CDMA network. However, as new cell phones continue to emerge that support TRAC-IT and more companies advocate open access of cellular networks and devices, it is extremely likely that TRAC-IT could be deployable across many different wireless networks by late 2008 or early 2009.

Chapter 7. Recommendations for Deployment

The use of GPS-enabled mobile phones and an application such as TRAC-IT presents a unique opportunity to departments of transportation (DOTs) to collect high-resolution travel behavior data for individuals. These data represent an improvement over traditional OD data, since GPS data accurately represent the path of travel and additional user input can be collected through the survey in the user interface. These data are instantly transferred to a server for analysis by transportation professionals. Additionally, field equipment costs are eliminated since TRAC-IT is a software application that is deployable to any privately owned GPS-enabled mobile phone that supports the popular J2ME platform and the standardized method of accessing location data, JSR179 Location API. As more “indoor” or (high-sensitivity) GPS chips become available in the mobile phone market in late 2008 and early 2009, DOTs can expect to collect GPS data from mobile phones that are more accurate and should be able to obtain location fixes where present phones can not. For all of these reasons, the TRAC-IT research team believes GPS-enabled mobile phones will be an important resource for travel behavior data collection in the future.

As discussed in detail in the Technology Assessment section of this report, it is important to differentiate between the two main methods of utilizing mobile phones to obtain travel information. The first method uses use of anonymous cellular signaling data gathered at cell towers to estimate travel time on nearby highways. Travel time and estimated travel speed on a specific road segment are the only items of information available from these types of systems, which may be sufficient for traffic operations. Companies operating these systems in the U.S. include AirSage, Cellint, Delcan, and IntelliOne. They have been evaluated by various DOTs with varying results of system accuracies (7), (8), and (9). Conclusions differ in each report, but the general consensus was that the speed and travel time were fairly accurate when traffic is at a free-flow state, but that the accuracy of estimated speeds and travel times decay as congestion increases and the vehicle speeds fall below free-flow levels.

The second method is the use of an actual positioning technology to gather position data for many mobile phones, which each have their position calculated individually by the phone or network using GPS, network triangulation, or a similar positioning technology. TRAC-IT focuses on this second method, which can yield extremely precise position data within 3-5 meters of the true position for each time point that is captured with a supported frequency of up to one fix per second. Other information such as speed and heading is also captured with each GPS data point, and therefore travel speeds and estimated travel times can also be extracted from this data (although values would have to be aggregated from multiple individual GPS data points to accurately reflect average speed instead of a single vehicle's speed). Speed data derived from GPS has been shown to be accurate within .2 meters/sec in past studies (10). This method yields OD-type data, as well as travel path information for each individual traveler and is a modern version of the traditional trip diary. Research, such

as TRAC-IT, is still active using this method of data collection, and no commercial company has yet been established to collect travel behavior data using this method to the research team's knowledge. It is important that DOTs understand the difference between the two methods of data collection, as they yield very different types of data to be used for different purposes.

As of early 2008, TRAC-IT can be deployed only to customers on Sprint-Nextel's CDMA and iDEN networks. Currently, this does provide some limitations to the statistical validity of using TRAC-IT to sample the general population's travel behavior. However, it appears as though other wireless providers will be opening up their networks in the near future (end of 2008 to early 2009), which would allow TRAC-IT to be deployed to nearly everyone who owns a mobile device. Since there are an estimated 254, 154, 584 U.S. wireless subscribers, this is a significant population to survey (85). Additionally, the recent FCC auction of the 700MHz spectrum to be used for future commercial services will be held to the condition of "open devices and open applications," thus further opening up wireless carriers to application such as TRAC-IT.

There is concern that sampling cell phone users would miss certain segments of the population that do not own mobile phones. However, an equal concern is the growing number of individuals that own only mobile phones and do not own landlines, which can potentially affect all surveys that are conducted using random calls to households with wired telephones (86). Surprising to some, cell phone-only households tend to have lower incomes. More research would have to be performed in order to analyze the implications of using data collected via a cell-phone based application such as TRAC-IT.

Because of the many issues involved in the deployment of an location-aware application such as TRAC-IT, USDOT, along with FDOT and others DOTs if possible, should engage wireless providers in a single dialog to discuss how a large-scale test of TRAC-IT (i.e., over 200 users) could be realized. As part of this dialog, it should become apparent whether there are any policy-based or logistical issues that would prevent a large-scale deployment of TRAC-IT, including any access restrictions by cellular carriers. For wireless providers that do not currently support the TRAC-IT application, USDOT should request an estimated timeline for technology deployment that would enable TRAC-IT to be used on those particular wireless providers. If there are no access restrictions or logistical barriers to deployment of TRAC-IT on a wireless provider, testing should be slowly escalated in the number of simultaneous mobile phones to determine whether there are any unforeseen scalability issues with the TRAC-IT system. While the design of the system has scalability in mind, there can be unknowns in the real-world implementation of technology.

Due to the many different mobile phone devices on the market, it is also recommended that DOTs create a process to test and certify specific devices with the TRAC-IT application before TRAC-IT is deployed to that particular device. This is especially important when devices are tested across multiple wireless providers, since significant

differences can exist in J2ME applications across providers. Since GPS hardware in mobile phones have different levels of sensitivity, accuracy testing should also be performed on specific GPS chipsets to determine whether that particular platform would yield data accurate enough for TRAC-IT purposes. Also, since the end-user device is privately owned, testing should ensure that no personal mobile phone functions will be compromised by TRAC-IT. As J2ME for mobile devices matures during the next year, including the release of MIDP 3.0 and OSGi which will standardize interactions between mobile applications and background application execution, future development and testing for new devices should become easier and TRAC-IT should be less likely to interfere with normal cell phone operations.

Since participants indicated that the biggest challenge they faced was remembering to start a trip via the TRAC-IT user interface, the passive recording feature of TRAC-IT should be investigated further. In this mode the phone would always be sending location data to the TRAC-IT server unless the user selected to turn this feature off via the TRAC-IT user interface. This feature would not require the user to always remember to start a trip, since it would be recording even if no action was taken by the user. However, passive tracking could negatively impact cell phone performance, including battery life. Therefore, further analysis of the State Machine and Critical Point functionality should take place to determine if these “smart” resource management techniques can offset any negative impact on cell phone functionality when passive tracking is turned on.

The unique methodology of TRAC-IT enables departments of transportation to explore new types of incentives in coordination with wireless service providers. For example, DOTs could potentially arrange with the wireless carrier to provide free unlimited data plans during the survey period to any participants willing to install and use TRAC-IT on their phone. Participants then have access to new services on their mobile phone in addition to TRAC-IT, including email, internet browsing, online search, etc. Once the participants are exposed to these services, they would be much more likely to continue to pay for the service after the TRAC-IT survey period ends, thus providing additional revenue to the wireless service provider. DOTs would benefit from reduced deployment costs through cost reductions in unlimited data plans, and the wireless carrier would benefit both by the DOT paying some cost for unlimited data plans during the survey period and by the participant potentially paying full cost for the plan after the survey period ends. Similar arrangements may be reachable by providing free participant access to real-time GPS navigation applications or other software for mobile phones.

The concept of path prediction and traffic incident alerts described in this report and the TRAC-IT Phase 2 report could also be used as an incentive by itself. For example, if participants use the TRAC-IT application to record their behavior, then they would automatically receive free, personalized traffic alerts based on their own travel behavior as well as their real-time location. This service could be valuable enough to mobile phone users that they overcome any reservations regarding privacy and would

be willing to allow their travel behavior to be tracked, at least using the passive version of TRAC-IT, and used for DOT analysis.

7.1 Future Research

While TRAC-IT Phases 2 and 3 accomplished some groundbreaking research in using GPS-enabled mobile phones for recording travel behavior at an individual level, there is still much future research to be completed.

7.1.1 Enhancements to the TRAC-IT system

Enhancements to the TRAC-IT system can be made to improve system performance and the amount of data collected.

A critical TRAC-IT feature that should be further evaluated is the passive collection of GPS data that continues to minimize or replace manual user input to begin recording travel behavior. Many participants reported that the biggest challenge they faced was remembering to indicate that they were starting a trip via the cell phone user interface. For participants who are comfortable always being tracked (with the ability to turn off tracking if desired), the passive version of TRAC-IT will not require user input to record travel behavior data since it is constantly in a "recording" mode. Most participants said they usually remembered the device but simply forgot to start recording their trip; having the device always collecting GPS data should significantly enhance the number of trips recorded. However, the impact of continuous tracking on device resources such as battery life should be closely studied. Intelligent mobile software features such as the State machine and Critical Point algorithm should be tested and evaluated, as they will likely be necessary to avoid negatively impacting mobile phone resources. Future research should quantify and compare the exact amount of power saved by the state machine and critical points techniques to demonstrate the usefulness of the algorithms.

Future enhancements to TRAC-IT should also include additional features such as the logging of counts of UDP packets sent from the phone to know how many UDP packets are lost for each trip in specific areas. Simple additions such as status condition printouts in critical areas of server code will allow for better debugging of seemingly random errors that occur due to complex multithreaded server applications.

One important feature that should be included in a future version of TRAC-IT is the ability to cache temporarily trip and GPS data on the phone in locations where wireless coverage is not available or server connectivity is not currently possible. This feature would help avoid key issues that resulted in the loss of trip and GPS data including the lack of current wireless coverage and server down time due to all of the aforementioned possible problems. A temporary caching feature must be implemented carefully, since many J2ME mobile phones do not have large amounts of memory for internal storage of data. Therefore, caching algorithms must be highly efficient and

should only cache data when absolutely necessary. Researchers should investigate this issue further in future versions of TRAC-IT.

User comments from field tests include the recommendation to allow users to enter trips they forgot to record via a web page or as part of the feedback process. A related idea was to allow the editing of trips for situations where the user forgets to end or begin a trip at the appropriate time. Therefore, TRAC-IT may serve as a good initial record of travel behavior that can be enhanced by user input after participants receive the feedback showing their trip. Some users also said that they did not like using the cell phone key pad to input text descriptions of locations when they visited them. Future versions of TRAC-IT should investigate whether voice recognition or audio recordings of location descriptions could be used in place of text entry.

To help protect user privacy, a future version of TRAC-IT could also include the ability to limit tracking when the user is in certain "sensitive" areas. Therefore, when the user enters this area the TRAC-IT application would prevent any data from being sent to the server. If the area was large enough, this may be adequate to help protect user privacy.

David H. Holtzman, author of *Privacy Lost*, identifies three basic meanings of privacy: (1) seclusion: the right to be hidden from the perceptions of others; (2) solitude: the right to be left alone and (3) self-determination: the right to control information about oneself (87). He identified seven "sins against privacy" that should be addressed. TRAC-IT addresses each of these sins.

TRAC-IT avoids the sin of intrusion by requiring each person to opt-in to use TRAC-IT; it is a program that must be downloaded and installed on the handset. This avoids the uninvited encroachment of a person's virtual space. The uses of TRAC-IT are clearly identified through the opt-in process so there is no sin of deception. There is no use of personal information in a way that was not authorized by the person involved.

The use of TRAC-IT to suggest travel options rather than directing them to use a particular mode is a reflection that it avoids the sin of profiling and does not judge the participant's decision to use or not use a particular mode simply by the data. TRAC-IT looks for opportunities for more efficient travel but it is clearly recognized that circumstances may not allow the traveler to choose that option. This approach also avoids the sin of lost dignity that could humiliate the participant with their own private information (e.g., why did you drive one-quarter mile to the store when you could have walked?)

There remain some privacy issues to be resolved. Holtzman's sins of latency and identity theft refer to the protection and excessive hoarding of personal information. Policies need to be developed and enforced to erase the data from the servers. Future TRAC-IT implementations should examine adding additional security to UDP data transfer. Encryption must be lightweight if a large amount of data will be repeatedly

encrypted and decrypted, and therefore Elliptical Curve Cryptography (ECC) appears to be a good candidate for UDP security (64). ECC has been used in sensors networks in the past as a lightweight encryption system (65). Since all encryption methods require CPU cycles and therefore battery power and will also increase the data packet size, the impact of adding encryption to UDP data transferred by the TRAC-IT mobile application should be investigated.

Finally, there remains the need to add protection for revealing information – what Holtzman refers to as the sin of “outing” by providing information that a person would rather remain hidden. For example, a future version of TRAC-IT could also include the ability to limit tracking when the user is in certain “sensitive” areas. Therefore, when the user entered this area the TRAC-IT application would prevent any data from being sent to the server. If the area were large enough, this may be adequate to help protect user privacy.

Researchers should examine the use of indoor positioning technologies such as AFLT to determine if occasional fixes can be used to augment tracking when GPS is not available. Researchers also should analyze the impact of using network-based methods such as AFLT on device resources since these techniques require additional battery power to accomplish position calculation.

Further evaluation and enhancements of the Path Prediction mechanism developed in TRAC-IT Phase 2 should also be conducted. This path prediction feature has a significant value to the end user and it would encourage active use of the TRAC-IT application. Path prediction will help travelers avoid areas of congestion completely by suggesting different routes, and therefore could have a significant positive impact on traffic congestion if deployed to many mobile phones. Future enhancements could include driving directions for alternate routes and verbal feedback to the user.

7.1.2 Evaluation of Mobile Technology

Mobile technology has an extremely short shelf life, and therefore new devices are being continually released. It is important to benchmark the performance of different mobile phones in terms of GPS accuracies and their capabilities of running TRAC-IT before TRAC-IT is installed.

7.1.3 Evaluation of Survey Protocol

Future testing should target more users that have plans with wireless carriers that are compatible with TRAC-IT. It is expected that if the same mobile phone is used as the participant’s personal phone the user will be less likely to forget to carry the phone. Only one user in the field tests described in this report had a compatible wireless carrier. This user reported that he never forgot to carry the phone with him or forgot to charge it since he is used to carrying it as his ordinary personal phone. The user did report forgetting to record trips manually at times. It is believed that the survey

method of TRAC-IT that will yield the highest percentage of reported trips is a passive version of TRAC-IT running on a participant's personal mobile phone. This method should hopefully eliminate both the failure of the participant to remember to carry the phone as well as the failure of the participant to remember to manually activate TRAC-IT to record their trip. However, as previously mentioned there can be negative impacts on device resources using this method so further analysis of passive tracking is required.

7.1.4 Recruitment for Future TRAC-IT Testing

Further testing of TRAC-IT on mobile phones owned by participants is suggested in order to further evaluate TRAC-IT. Since these individuals must be on particular wireless providers and have certain model mobile phones, recruitment can be a difficult obstacle. The research team identified a promising new method of recruitment referred to as "Wireless Application Protocol (WAP) Advertising." WAP Advertising is a version of online advertising for web pages appearing on mobile phones, referred to as WAP pages. A banner appears on the user's phone while the user is browsing the web or reviewing his or her account information, and the user can click on it to be directed to a page with further instructions or advertising. WAP Advertising differs from traditional online advertising in that it is controlled by the wireless carrier, so a banner ad can be inserted by the carrier on any web page that is visited. Additionally, ads can be targeted to individuals in certain geographic areas, certain cell phone models, and plans with certain characteristics (i.e., family plans). Therefore, WAP Advertising has the potential to reach the exact intended population, even down to which cell phone models are supported by the mobile TRAC-IT application. Further research is required into this technology to fully assess its potential.

References

- (1) Cleland, F., and P. Winters (1999), "Reducing vehicle trips and vehicle miles of travel through customized travel options, final report: results of survey and conclusions." Center for Urban Transportation Research, College of Engineering, University of South Florida. Department of Transportation State of Florida.
http://www.cutr.usf.edu/tdm/pdf/reducing_VMT.pdf, accessed February 11, 2008.
- (2) Winters, P., N. Georggi, and S. Barbeau (2005), "Traveling Smart: Increasing Transit Ridership through Automated Collection (TRAC) of Individual Travel Behavior Data and Personalized Feedback." National Center for Transit Research, Center for Urban Transportation Research, University of South Florida. Florida Department of Transportation. <http://www.nctr.usf.edu/pdf/576-16.pdf>, accessed February 11, 2008.
- (3) "FCC E911 – Wireless Services" Official Website at <http://www.fcc.gov/911/enhanced/>, accessed February 11, 2008.
- (4) K. Ridley (2007) "Global Mobile Phone Use to Hit Record 3.25 Billion," Reuters. © Reuters 2007. Available online at <http://www.reuters.com/article/email/idUSL2712199720070627>.
- (5) "GPS-enabled Location-Based Services (LBS) Subscribers Will Total 315 Million in Five Years," ABI Research, New York.
<http://www.abiresearch.com/abiprdisplay.jsp?pressid=731>, accessed February 11, 2008.
- (6) "Personal Locator Services to Reach More than 20 Million North American Consumers by 2011," ABI Research, New York.
<http://www.abiresearch.com/abiprdisplay.jsp?pressid=766>, accessed February 11, 2008.
- (7) Wunnava, Yen, Babij, Zavaleta, Romero, and Archilla (2007), "Travel Time Estimation Using Cell Phones (TTECP) for Highways and Roadways," Final Report for Florida Department of Transportation.
- (8) Guin, Bachman, and Demidovich (2007), "Strategies for Evaluating the Usefulness of Cellular Probe Traffic Data in Representing Aggregate Traffic Conditions on Freeways," Final Report for the Georgia Department of Transportation, 2007.
- (9) Liu, Danczyk, Brewer, and Starr (2008), "Evaluation of Cell Phone Traffic Data in Minnesota," Transportation Research Board 2008 Annual Meeting (CD-ROM), Washington, D.C.
- (10) Witte and Wilson (2004), "Accuracy of non-differential GPS for the determination of speed over ground," *Journal of Biomechanics*, 37(12):1891-8.
- (11) E. Murakami and D. P. Wagner (1999), "Can using global positioning system (GPS) improve trip reporting?" *Transportation Research Part C: Emerging Technologies*, Volume 7, Issues 2-3, p. 149-165. © 1999 Elsevier Science Ltd.

- (12) Federal Communications Commission (FCC), "Enhanced 911 – Wireless Services," <http://www.fcc.gov/911/enhanced/>, accessed February 11, 2008.
- (13) Federal Communications Commission (1999), "FCC Acts to Promote Competition and Public Safety in Enhanced Wireless 911 Services," Press Release, Sept. 15, 1999. http://www.fcc.gov/Bureaus/Wireless/News_Releases/1999/nrw19040.html, accessed February 11, 2008.
- (14) "Dispatch Monthly Magazine" Official Website at <http://www.911dispatch.com/db/index.php>, accessed February 11, 2008.
- (15) van Diggelen, F., "Indoor GPS Theory & Implementation," IEEE Position, Location & Navigation Symposium, 2003. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=998914, accessed February 11, 2008.
- (16) Küpper, A. (2005) "Location-Based Services, Fundamentals and Operation," p. 152, John Wiley & Sons Ltd., 2005.
- (17) Sun Microsystems, Inc. "The Java ME Platform – the Most Ubiquitous Application Platform for Mobile Devices," <http://java.sun.com/javame/>, © 1994-2007 Sun Microsystems, Inc.
- (18) Nokia. "Forum Nokia – Location Based Services," http://www.forum.nokia.com/main/resources/technologies/location_based_services/documentation.html.
- (19) Qualcomm. "Qualcomm BREW – Developer Home," <http://brew.qualcomm.com/brew/en/developer/overview.html>.
- (20) Motorola. iDEN J2METM Developer's Guide 2005, Version 1.98, © Motorola, Inc. 2005, p. 487-498.
- (21) Qualcomm. "Qualcomm Java Application Extensions," http://www.cdmatech.com/products/qvm_qjae.jsp. © Qualcomm 2007.
- (22) David P. Aguilar, Sean J. Barbeau, Rafael A. Perez, Miguel A. Labrador, Philip L. Winters (2007), "A Comparison of Fix Times and Estimated Accuracies in Application Programming Interfaces (APIs) for GPS Enabled Mobile Phones", *Proceedings of the 11th World Conference on Transport Research*, Berkeley, CA.
- (23) Sun Microsystems, Inc. "Java Specification Request (JSR) 179: Location API for J2ME™," <http://jcp.org/en/jsr/detail?id=179>, © Sun Microsystems, Inc. 2007.
- (24) Sun Microsystems, Inc. "Java Specification Request (JSR) 293: Location API 2.0," <http://jcp.org/en/jsr/detail?id=293>, © Sun Microsystems, Inc 2007.
- (25) Google. "Android – An Open Handset Alliance Project," <http://code.google.com/android/documentation.html>, © Google 2007.

- (26) S.J. Barbeau, M.A. Labrador, P.L. Winters, R. Pérez, N.L. Georggi (2008), "Location API 2.0 for J2ME – A New Standard in Location for Java-enabled Mobile Phones," *Computer Communications*. Future online address doi:10.1016/j.comcom.2008.01.045.
- (27) European Telecommunications Standards Institute. <http://www.etsi.org/>, © ETSI 2004.
- (28) Open Mobile Alliance (OMA) Location Working Group. "TS 101 - Mobile Location Protocol Specification – Version 3.0.0," <http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html>, © 2007 Open Mobile Alliance Ltd.
- (29) The Parlay Group. "Parlay/OSA Specifications," <http://www.parlay.org/en/specifications/>, © Parlay 2007.
- (30) Palm. "Palm Developer Network," <https://pdn.palm.com/regac/pdn/index.jsp>, accessed February 11, 2008.
- (31) Research In Motion, Inc. "Blackberry – BlackBerry Developer Program," <http://na.blackberry.com/eng/developers/>.
- (32) Netbeans. "Netbeans Integrated Development Environment," <http://www.netbeans.org>.
- (33) Eclipse. "Eclipse – an open development platform," <http://www.eclipse.org/>.
- (34) Sun Microsystems, Inc. "Sun Java Wireless Toolkit for CLDC (formerly known as J2ME Wireless Toolkit)," <http://java.sun.com/products/sjwtoolkit/>.
- (35) Sprint Nextel Corporation. "Overview of Sprint Location Based Services," © Sprint Nextel Corporation 2006.
- (36) Sun Microsystems, Inc. "JSR118: Mobile Information Device Profile 2.0," <http://jcp.org/en/jsr/detail?id=118>.
- (37) Information Sciences Institute (1981) "Request for Comments (RFC) 793 – Transmission Control Protocol DARPA Internet Program Protocol Specification," Information Sciences Institute.
- (38) Postel, J. (1980) "Request for Comments (RFC) 768 – The User Datagram Protocol."
- (39) Sun Microsystems, Inc. "JSR172: J2ME Web Services Specification," <http://jcp.org/en/jsr/detail?id=172>, accessed February 11, 2008
- (40) W3C. "Simple Object Access Protocol v1.2," <http://www.w3.org/TR/soap/>, accessed February 11, 2008
- (41) "Nextel to Deploy Motorola's WiDEN Data Technology" (2003) *Converge! Network Digest*. <http://www.convergedigest.com/Daily/daily.asp?vn=v10n220&fecha=11%2F17%2F2003>, accessed February 11, 2008.

- (42) Sprint. "Push-to-Talk on New Devices Will Give Sprint Customers More Ways to Get Things Done at SprintSpeed," August 16, 2007.
http://newsreleases.sprint.com/phoenix.zhtml?c=127149&p=irol-newsArticle_newsroom&ID=1041306&highlight=, accessed February 11, 2008.
- (43) Sprint (2008) "Sprint Confirms Commitment to Industry-Leading Push-to-Talk Service," http://newsreleases.sprint.com/phoenix.zhtml?c=127149&p=irol-newsArticle_newsroom&ID=1101815&highlight=, accessed February 11, 2008
- (44) Sprint-Nextel (2006) "Sprint Location-Based Service Platform Summary," Version 1.0.
- (45) Sprint-Nextel Corporation. "Overview of Sprint Nextel's MIDP 2.0 API Security and How to Sign MIDP 2.0 Applications for CDMA devices,"
<http://developer.sprint.com/getDocument.do?docId=84465>, May 1st, 2006. © 2005 Sprint Nextel Corporation
- (46) Qualcomm. "Qualcomm Technology and Solutions – Location Based Services," <http://www.qualcomm.com/technology/location.html>, © Qualcomm 2008.
- (47) Sprint. "Sprint Business – Sprint Mobile Broadband Coverage,"
http://www.sprint.com/business/products/products/evdoCoverage2.jsp?map=Tampa_FL&mrkt=Tampa/St.%20Petersburg,%20FL, accessed February 11, 2008. © 2006 Sprint Nextel.
- (48) AT&T. "devCentral | Location-based Services (LBS)," <http://developer.att.com/developer/index.jsp?page=toolsTechSection&id=7000080>, accessed February 11, 2008. © 2007 AT&T Intellectual Property.
- (49) AT&T. "Cingular devCentral Location Based Services for Enterprises," http://developer.att.com/devcentral/community/webcasts/docs/LBS_Webcast_V15.pdf, accessed February 11 2008. © 2007 AT&T Knowledge Ventures.
- (50) Qualcomm. "FAQ – Does BREW support multi-threading?"
<https://brewx.qualcomm.com/bws/content/qi/common/appseng/en/developerfaqs/docs/creatingapps/coding/coding.html>, accessed February 11, 2008. © Qualcomm 2008.
- (51) Verizon Wireless. "Verizon Wireless to Introduce 'Any Apps, Any Device' Option for Customers in 2008," Press release, Basking Ridge, NJ, 11/27/2007.
<http://news.vzw.com/news/2007/11/pr2007-11-27.html>, accessed February 11, 2008.
 © 2008 Verizon Wireless.
- (52) Reed, Brad (2007) "What does Verizon's open-access network option mean for you?" *Network World*. <http://www.networkworld.com/news/2007/112807-verizon-analysis.html?page=1>, accessed February 11, 2008. © 1994-2008 Network World, Inc.
- (53) Abraham, Charles and van Diggelen, Frank (2001) "Indoor GPS: The No-Chip Challenge," *GPS World*, September 1, 2001.
- (54) Sun Microsystems, Inc. "JSR271: Mobile Information Device Profile 3," <http://jcp.org/en/jsr/detail?id=271>, © 1995-2008 Sun Microsystems, Inc.

- (55) Sun Microsystems, Inc. "GlassFish – Open Source Application Server," <https://glassfish.dev.java.net/>, © 1995-2007 Sun Microsystems, Inc., accessed February 11, 2008.
- (56) JBoss. "JBoss Application Server," <http://labs.jboss.com/jbossas/>, © Red Hat Middleware, LLC., accessed February 11, 2008.
- (57) Sun Microsystems, Inc. "Sun Java System Application Server – Downloads," <http://developers.sun.com/appserver/downloads/index.jsp>, © 1994-2008 Sun Microsystems, Inc., accessed February 11, 2008.
- (58) BEA Systems, Inc. "BEA WebLogic Server 10," <http://www.bea.com/framework.jsp?CNT=index.htm&FP=/content/products/weblogic/server/>, © 2008 BEA Systems, Inc., accessed February 11, 2008.
- (59) IBM. "WebSphere Application Server," http://www-306.ibm.com/software/webervers/appserv/was/index.html?S_TACT=103BGW01&S_C_MP=campaign, © IBM Corporation 1994, 2007, accessed February 11, 2008.
- (60) Sun Microsystems, Inc. "Java Platform, Enterprise Edition 5 (Java EE 5)," © 1994-2008 Sun Microsystems, Inc., accessed February 11, 2008.
- (61) Sun Microsystems, Inc. "JSR 139 – Connected Limited Device Configuration," <http://java.sun.com/javame/reference/apis/jsr139/>, © 2006 Sun Microsystem, Inc.
- (62) Barbeau, Labrador, Winters, Perez and Georggi (2006) "A General Architecture in Support of Interactive, Multimedia, Location-based Mobile Applications," *IEEE Communications Magazine*, Vol. 44, No. 11, pp. 156-163.
- (63) Sun Microsystems, Inc. "JSR 218 – Connected Device Configuration," <http://java.sun.com/javame/reference/apis/jsr218/>, © 2006 Sun Microsystem, Inc.
- (64) N. Koblitz, Elliptic curve cryptosystems, in *Mathematics of Computation* 48, 1987, pp. 203–209
- (65) Sun Microsystems, Inc. "Next Generation Crypto," <http://research.sun.com/projects/crypto/>, accessed February 11, 2008. © 1994-2008 Sun Microsystems, Inc.
- (66) Microsoft. "Microsoft SQL Server 2005 Home," www.microsoft.com/sql/, © 2008 Microsoft Corporation.
- (67) Sun Microsystems, Inc. "JSR224: Java API for XML-Based Web Services," <http://jcp.org/en/jsr/detail?id=224>, © 1995-2008 Sun Microsystems, Inc.
- (68) Gilani, Himanshu (2005) "Automatically Determining Route and Mode of Transport Using GPS Enabled Phone," Master's Thesis, University of South Florida, 2005 Himanshu Gilani.
- (69) Gonzalez, P., Weinstein, J., Barbeau, S., Labrador, M., Winters, P., Georggi, N., Perez, R. (2008), "Automating Mode Detection Using Neural Networks and Assisted

GPS Data Collected Using GPS-enabled Mobile Phones,” submitted for publication to 15th World Congress on ITS, November 2008.

(70) Wolf, J., R. Guensler and W. Bachman (2001), “Elimination of the Travel Diary: An Experiment to Derive Trip Purpose from GPS Travel Data,” *Transportation Research Record* 1768, p. 125-134. Transportation Research Board.

(71) U.S Department of Transportation. National Household Travel Survey (NHTS), <http://www.fhwa.dot.gov/policy/ohpi/nhts/index.htm>, accessed February 11, 2008.

(72) 2001 NHTS User’s Guide: Chapter 3 – Survey Procedures and Methodology http://nhts.ornl.gov/2001/usersguide/chapter_3.pdf , accessed February 11, 2008.

(73) Letter Report- National Household Travel Survey (NHTS) <http://onlinepubs.trb.org/onlinepubs/reports/nhts.pdf> , accessed February 11, 2008.

(74) Hillsborough County Property Appraiser. “Hillsborough County Property Appraiser – ORDER MAPS AND DATA,” http://www.hcpafl.org/www/downloads/order_maps.shtml, accessed February 11, 2008.

(75) Wolf, J. (2004), “Applications of New Technologies in Travel Surveys,” *International Conference on Transport Survey Quality and Innovation*, Costa Rica, August 2004.

(76) Griffin, T., Y. Huang and R. Halverson (2006), “Computerized Trip Classification of GPS Data,” *International Conference on Cybernetics and Information Technologies, Systems and Applications*, Orlando, Florida.

(77) (1993) “1990 Household Travel Study: Final Report.” Atlanta Regional Commission.

(78) Welcome to the EDN Documentation Library for ArcObjects. <http://edndoc.esri.com/arcobjects/9.1/>, accessed February 11, 2008.

(79) Wagner, David P. Battelle. Report: Lexington Area Travel Data Collection Test; GPS for Personal Travel Surveys. Final Report for OHIM, OTA, and FHWA, September 1997.

(80) Draijer, Geert, Nelly Kalfs, and Jan Perdok, “GPS as a Data Collection Method for Travel Research,” Paper No. 00-1176, Presented and accepted for publication at the 79th Meeting of the Transportation Research Board, Washington, D.C., January 2000.

(81) Casas, Jesse and Carlos Arce; NuStats International (1999) “Trip Reporting in Household Travel Diaries: A Comparison to GPS-Collected Data,” 78th Annual Meeting of the Transportation Research Board, Washington, D.C., January 1999.

(82) David P. Aguilar, Sean J. Barbeau, Miguel A. Labrador, Alfredo Perez, Rafael A. Perez, and Philip L. Winters (2007), “Quantifying the Position Accuracy of Real-time Multi-Modal Transportation Behavior Data Collected using GPS-Enabled Mobile Phones”, Transportation Research Record: *Journal of the Transportation Research Board*, No. 1992, pp. 54-60.

- (83) Schreiner, Keri (2007), "*Where We At? Mobile Phones Bring GPS to the Masses*," IEEE Computer Graphics and Applications. Volume 27, Issue 3, May-June 2007.
- (84) CTIA. Location Based Service Action Team.
<http://www.ctia.org/content/index.cfm/AID/10331>, accessed February 11, 2008.
- (85) CTIA – The Wireless Association. <http://www.ctia.org/>, accessed February 11, 2008.
- (86) Fram, Alan (2007), "*Trend sees cell phone only use growing*," USA Today. © 2007 The Associated Press.
- (87) Holtzman, David H. (2006), "Privacy Lost: How Technology Is Endangering Your Privacy," Wiley Publishers.

Appendices

Appendix A - Letter of Solicitation

Appendix B - Informed Consent

Appendix C - Participant Information Form

Appendix D -TRAC-IT Orientation Session



Center for Urban Transportation Research
University of South Florida
4202 East Fowler Avenue, CUT 100
Tampa Florida 33620-5375
(813) 974-3120
SunCom 574-3120
Fax (813) 974-5168
Web: <http://www.cutr.usf.edu>

October 29, 2007

John and Jane Smith
1234 Tracit Lane
Tampa FL 33620

Dear Smith Family:

What do we need most to make the transportation system serve you better in the future? Can traffic congestion be reduced? Will new technology improve traffic conditions? Can you save on gas and other auto costs? The Center for Urban Transportation Research (CUTR) at the University of South Florida is conducting a survey to answer these and other important questions and we need your help! Only by knowing and understanding your travel needs can planners and decision makers effectively work on improving the mobility, safety, and security of our transportation system. We are asking you to help by telling us about your travel needs, the kinds of trips that you make, and the ways that you make these trips.

You have our promise that we will protect your privacy and that none of the information about your individual household will be used for any other purpose or provided to any other agency or business.

You will be provided with a cell phone equipped with global positioning system (GPS) capabilities to chronicle your travel activities with some input from you. Details of the process will be further discussed at the orientation session we prepared for you (please see attached consent form).

If you have any other questions or concerns about this survey, please feel free to contact me.

Sincerely,

Philip L. Winters
TDM Program Director
Center for Urban Transportation Research
University of South Florida
4202 E. Fowler Ave., CUT100
Tampa, FL 33620-5375
(813) 974-9811
winters@cutr.usf.edu

Informed Consent

Social and Behavioral Sciences
University of South Florida

Information for People Who Take Part in Research Studies

The following information is being presented to help you decide whether or not you want to take part in a minimal risk research study. Please read this carefully. If you do not understand anything, ask the person in charge of the study.

Title of Study: Testing the Impact of Personalized Feedback on Household Travel Behavior (TRAC-IT Phase 2)

Principal Investigator: Philip L. Winters, Center for Urban Transportation Research, USF

Study Location(s): USF campus and Tampa Bay Area

You are being asked to participate because personal transportation is part of your daily activities (e.g. transportation to work, shopping, entertainment, medical, etc.) and we want to better understand travel choice decisions and travel patterns people make.

General Information about the Research Study

The purpose of this research study is to

1. Test a travel advisory feedback system on a sample of households providing customized travel advice to encourage alternatives to drive alone or driving in peak hours of traffic.
2. Quantify changes in travel behavior patterns after providing personalized travel advice to encourage individuals to choose a mix of travel choices to satisfy their travel needs rather than only choose the single occupant vehicle.

Plan of Study

You will be required to carry a GPS-enabled mobile phone during the study period. The duration of your participation, including data collection periods and interviews with the research team, will be no more than six months.

The GPS system will record your trip ends, route, travel time, date/time of day and travel distance for each trip. Trips may be made by personal vehicle, bus, bike, walk or other mode. The cell phone also will prompt you to provide additional information not collected by the GPS unit about the trip such as mode of travel (e.g., drive or walk) and the number of occupants in the vehicle with you. We anticipate that it will take you a total of 10 minutes per day to complete the data record for trips made that day. We also anticipate that you will carry the cell phone for at least two one-week periods.

Upon your consent, we will spend about one hour to instruct you on the use of the unit. We will contact you by email or phone to make sure no problems have been encountered and answer any questions you may have.

After data collected are analyzed, we will meet for the second visit (approximately 1 hour) to review your travel habits and gauge the effectiveness of our advice on ways you can reduce vehicle trips.

Following any necessary revisions to the software on the unit, you will carry the unit and record travel behavior for another one week period. A third visit (approximately 60 – 90 minutes) may be held with you so we can assess the following:

- The experience associated with participating in the project,
- The accuracy of the information provided by the unit,
- The quality of the feedback based on the data collected from those units, and
- The changes made, if any, to reduce their vehicle trips based on advice received from researchers.

This project is funded by the Florida Department of Transportation and U.S. Department of Transportation under grants to the National Center for Transit Research at the University of South Florida.

Payment for Participation

You will not be paid for your participation in this study

Benefits of Being a Part of this Research Study

You will not directly benefit from participating in this study. However, by taking part in this research study, you may increase our overall knowledge on the travel behavior and patterns. Our overall knowledge enables us to assist communities and transportation agencies in planning improvements such as bike paths, sidewalks, transit routes and road improvements. Your participation will also allow us to test better ways to provide traveler information, meaningful advice and feedback on alternative methods of travel that realistically meet the mobility needs of Floridians.

Risks of Being a Part of this Research Study

There are no foreseeable risks in participating in this study.

Confidentiality of Your Records

Completed informed consent forms and travel information forms will remain with researchers at all times during data collection. They will be kept on file in secure offices at CUTR during analysis and storage. Data will be stored on password-protected computers. Documents will be stored in lockable research office. Data will be retained for the life of the project and comply with contractual requirements.

The results of this study may be published. The published results will not include your name. However, the inclusion of travel patterns in the published results in a tabular and/or map format and the accuracy of the GPS units could permit others to identify a particular household or even individual and link them with dates, times, and locations visited.

Your privacy and research records will be kept confidential to the extent of the law. Authorized research personnel, employees of the Department of Health and Human Services, and the USF Institutional Review Board and its staff, and any other individuals acting on behalf of USF, may inspect the records from this research project.

Volunteering to Be Part of this Research Study

Your decision to participate in this research study is completely voluntary. You are free to participate in this research study or to withdraw at any time. There will be no penalty if you stop taking part in the study.

Questions and Contacts

- If you have any questions about this research study, contact Phil Winters at 813-974-9811.
- If you have questions about your rights as a person who is taking part in a research study, you may contact the Division of Research Compliance of the University of South Florida at (813) 974-5638.

Consent to Take Part in This Research Study

By signing this form I agree that:

- I have fully read or have had read and explained to me this informed consent form describing this research project.
- I have had the opportunity to question one of the persons in charge of this research and have received satisfactory answers.
- I understand that I am being asked to participate in research. I understand the risks and benefits, and I freely give my consent to participate in the research project outlined in this form, under the conditions indicated in it.
- I have been given a signed copy of this informed consent form, which is mine to keep.

Signature of Participant

Printed Name of Participant

Date

Investigator Statement:

I certify that participants have been provided with an informed consent form that has been approved by the University of South Florida's Institutional Review Board and that explains the nature, demands, risks, and benefits involved in participating in this study. I further certify that a phone number has been provided in the event of additional questions.

Signature of Investigator

Printed Name of Investigator

Date

Participant Information Form

Thank you for taking part in this research study that will increase our overall knowledge on the travel behavior and patterns.

The information you provide here will help us set-up the cell phone you will be carrying while participating in this survey. We will pre-set as many options as we can so you can make your choice with a click of a button.

Last Name: _____ First Name: _____

Email Address: _____

Gender: _____ Male _____ Female Married: _____ yes _____ no

Home Address: _____

Work Address: _____

Number of children living at home: _____ How do your children get to school or daycare?

	Child age	Takes Bus to school	Parents take child to school	Child Drives to school
1				
2				
3				
4				

On regular basis, where do you make trips?

Places you drive/walk/bike to frequently	Example: Publix on 56 th and Fowler
Grocery store	
Bank	
Wal-Mart/Target	
Church	
Gym	
Mall	
Gas station	
Other	

TRAC-IT Orientation Session

conducted by



The "TRAC-IT" Team

Phil Winters
Nevine Labib Georggi
Sean Barbeau

We need you
*to participate in a
daily travel survey,*
using a GPS-enabled
cell phone.



What, Why, Who ... of Travel Surveys

- What is a travel survey?
 - What is it used for?
- Why collect this data?
- Who is conducting the survey?
 - Who is participating?
- Where is the data collected?
- When is the data collected?
- How is it collected?
 - How is my privacy protected?

What is a Travel Survey?

- A tool to collect data about daily journeys of individuals or households including information about:
 - an individual (socio-economic, demographic, etc.),
 - their household (size, structure, relationships),
 - their vehicles if any (age, make, model), and
 - a diary of their journeys on a given day (their start and end time and location, mode of travel, purpose of trip, number of household members making the trip).

Why conduct a Travel Survey?

- Transportation infrastructure improvements are very expensive and require extensive planning in advance.
- Data from survey is used to predict future demand on area roads, trains, and buses (at least 20 years into the future).
- If we know where and what we need to build, then we have a better chance of making timely, cost-effective improvements.

Who is conducting the survey?

- The National Center for Transit Research at the Center for Urban Transportation Research at the University of South Florida
- Person in Charge
 - Philip L. Winters
TDM Program Director
Center for Urban Transportation Research
University of South Florida
4202 E. Fowler Ave., CUT100
Tampa, FL 33620-5375
(813) 974-9811
winters@cutr.usf.edu

The Where? . . . and the When?

- Where is the data collected?
 - The cell phone collects (with your help) and transmits to a computer server here at USF.
- When is the data collected?
 - Cell phone is sending its location when the TRACIT application has been started and you've pressed 'Record'.

How is My Privacy Protected?

- Completed informed consent forms and travel information forms will remain with researchers at all times during data collection.
- The forms will be kept on file in secure offices at CUTR during analysis and storage.
- Data will be stored on password-protected computers.
- Documents will be stored in lockable research office.
- Please refer to the informed consent form for more details

Trip – how is a trip defined

- Any activity involving travel is a trip
 - walk to snack bar
 - drive thru Mc-burger-subway-taco
 - Stop on way home to pick up milk
 - fill up car
 - drop off at soccer
 - pick-up from church
 - go to park
 - get to work
 - go home

