

Prepared by

National Center
for Transit Research



Enabling Cost-Effective Multimodal Trip Planners through Open Transit Data

May 2011

Final Report

oTP OPEN TRIP PLANNER

Trip Planner
PLAN A TRIP DEADEND(-82,416)

Walk to Alumni Dr @ Laurel Dr
About 2 minutes - 0.1 mi

1. Walk southeast on way 42705605 (95 ft)
2. Right at way 43069949 (172 ft)
3. Right to continue on way 43069949 (192 ft)
4. Slight left way 43069949 becomes way 43069948 (20 ft)
5. Slight right way 43069948 becomes way 43069939 (16 ft)
6. Slight right way 43069939 becomes way 43069940 (6 ft)
7. Slight left to continue on way 43069940 (14 ft)
8. Slight right way 43069940 becomes way 42702330 (16 ft)

BUS 5
2:16PM Depart Alumni Dr @ Laurel Dr (North to University Area TC)
19 minutes
2:35PM Arrive University Area Transit Center

BUS 18
2:35PM Depart University Area Transit Center (South to UATC/ Downtown)
68 minutes
3:43PM Arrive Marion Transit Center

Walk to E Kennedy Boulevard
About 11 minutes - 0.6 mi

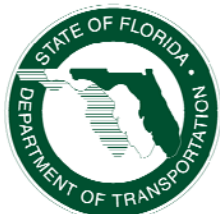
1. Walk south on way 23951255 (254 ft)
2. Left at E Fortune St (232 ft)
3. Right at E Morgan St (0.4 mi)
4. Left at E Kennedy Boulevard (196 ft)

End at E Kennedy Boulevard

Trip Details:
Travel Mon, Jan 10th 2:13PM
Valid January 10th, 2011 @ 2:46pm
Time 100 minutes
Walk 0.7 mi

Funded by

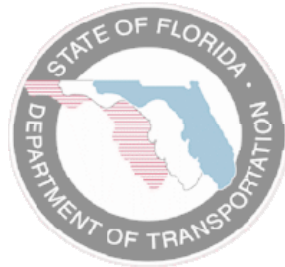
Florida Department
of Transportation



Enabling Cost-Effective Multimodal Trip Planners through Open Transit Data

FDOT BDK85 Task Work Order #977-20

Prepared for:



Florida Department of Transportation
Thomas M. Kelly, Co-Project Manager
Raymond Miller, Co-Project Manager
Amy Datz, Co-Project Manager

Prepared by:



Edward L. Hillsman, Senior Research Associate
Sean J. Barbeau, Research Associate
USF Center for Urban Transportation Research

Final Report

May 2011

DISCLAIMER

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

METRIC CONVERSION TABLE

SYMBOL	WHEN YOU KNOW	MULTIPLY BY	TO FIND	SYMBOL
LENGTH				
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
VOLUME				
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft³	cubic feet	0.028	cubic meters	m ³
yd³	cubic yards	0.765	cubic meters	m ³
NOTE: volumes greater than 1000 L shall be shown in m ³				
MASS				
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
TEMPERATURE (exact degrees)				
°F	Fahrenheit	$\frac{5}{9}(F-32)$ or $(F-32)/1.8$	Celsius	°C

1. Report No. USF 21177926	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Enabling Cost-Effective Multimodal Trip Planners through Open Transit Data		5. Report Date May 2011	
7. Author(s) Edward L. Hillsman and Sean J. Barbeau		6. Performing Organization Code	
9. Performing Organization Name and Address National Center for Transit Research Center for Urban Transportation Research , University of South Florida 4202 East Fowler Avenue, CUT100 Tampa, FL 33620-5375		8. Performing Organization Report No.	
12. Sponsoring Agency Name and Address Research Center Florida Department of Transportation 605 Suwannee Street, MS 30 Tallahassee, FL 32399-0450 Research and Innovative Technology Administration U.S. Department of Transportation Mail Code RDT-30 1200 New Jersey Ave, SE, Room E33 Washington, DC 20590-0001		10. Work Unit No. (TR AIS)	
15. Supplementary Notes FDOT Project Managers - Thomas M. Kelly, District Seven GIS/EDMS Coordinator, Raymond Miller, District 7 GIS Coordinator, Amy Datz, State Transit Environmental Planner		11. Contract or Grant No. BDK85 TWO 977-20 DTRT07-G-0059	
16. Abstract This study examined whether multimodal trip planners can be developed using open-source software and open data sources. OpenStreetMap (OSM), maintained by the non-profit OpenStreetMap Foundation, is an open, freely available international repository of geographic data that individuals contribute about their communities. In the transit industry, Google's offer of a free online transit trip planner based on the General Transit Feed Specification (GTFS) has made GTFS a <i>de facto</i> standard for describing transit systems and a platform for many other Web and mobile applications. Over 125 public transportation agencies in the U.S. have put their data into GTFS format. Bus stop locations can link OSM and GTFS data. OpenTripPlanner is an open-source multimodal trip planning software system with an active developer community. The study team set up an instance of OpenTripPlanner for Tampa, Florida, using biking and walking data from OSM, and GTFS data from local transit agencies, to examine the tool's ability to route using multimodal data. The study team also recorded multimodal data for the Tampa region in OSM to examine the current OSM coding conventions and determine the coding system's ability to support functions required of a multimodal trip planner, such as providing information on access to transit, wheelchair accessibility, or conditions that could affect the safety of a trip (e.g., intersection crossings). This study also investigated the use of open-source software to quickly increase the amount of multimodal data available in OpenStreetMap. The research team created GTFS-OSM-Sync (GO_Sync), a framework and open-source software tool for synchronizing transit data between the transit agency's official GTFS dataset and OSM. GO_Sync connects the wealth of data from GTFS datasets to the ability of the OSM community to augment and improve the data. During a test deployment of GO_Sync in Tampa, OSM users corrected 173 bus stop locations. The project demonstrated that it is feasible to implement a multimodal trip planner using open-source software and open data sources. Based on existing practices regarding GTFS and OpenTripPlanner, transit schedule and route data are best obtained directly from transit agencies' GTFS files. Data on infrastructure for walking and cycling can be obtained from OSM or from other locally available public-domain data. This report suggests a few changes to the OSM coding conventions that would improve OSM's ability to meet the needs of a multimodal trip planner. The principal barrier to developing a multimodal trip planner remains the availability of data and, when using OSM as a source of data, the relatively low participation of U.S. residents in the project, compared to Europe. The OSM community recognizes this as a problem, but additional research is needed on how best to overcome it. Additional research also is needed on how best to communicate results from a trip planner to users who may have varying skill and comfort levels when it comes to bicycling and walking.		13. Type of Report and Period Covered Final – February 2010 to May 2011	
17. Key Word OpenStreetMap, transit, data, OpenTripPlanner, multimodal, public transportation, geographic information systems		14. Sponsoring Agency Code	
19. Security Classif. (of this report) unclassified		18. Distribution Statement 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 website at http://www.nctr.usf.edu/ .	
20. Security Classif. (of this page) unclassified		21. No. of Pages 114	22. Price

ACKNOWLEDGEMENTS

This report was prepared by the National Center for Transit Research (NCTR) at the Center for Urban Transportation Research (CUTR) at the University of South Florida (USF) through the sponsorship of the Florida Department of Transportation (FDOT) and the U.S. Department of Transportation.

The authors would like to acknowledge the significant contributions to this project by the USF Computer Science and Engineering student research team members, Marcy Gordon, Graduate Research Assistant, and Khoa Tran, Graduate Research Assistant and primary author of Chapter 4.

The research team would like to thank the following transit agencies for their cooperation as part of this project:

- Hillsborough Area Regional Transit
- Sarasota County Area Transit
- Pinellas Suncoast Transit Authority
- Space Coast Area Transit
- Miami-Dade Transit

The research team thanks the Sprint-Nextel Application Developer Program for the donation of cellular service which was used as part of this project.

EXECUTIVE SUMMARY

The integration of multimodal data and information systems, including Global Positioning Systems (GPS) technologies and mobile devices (e.g., cell phones), is the next evolution of advanced traveler information systems and has been identified by the Federal Transit Administration Intelligent Transportation System Program as part of their 2009–2029 Strategic Plan. However, most transit information is currently locked into proprietary formats and systems and cannot be easily shared, viewed, updated, or co-mingled without permission from the vendor and expert data analysis. This limits the ability of transit agencies and others to provide information resources, such as trip planning tools, that support the use of transit or other alternatives to driving.

There are three major barriers to developing multimodal trip planning services that can work in more than a single U.S. metropolitan area.

1. The available high-quality proprietary data on street networks is provided by private vendors, such as NAVTEQ or Tele Atlas, that require recurring licensing and use fees for others to access.
2. Although there is an evolving widely accepted standard for recording transit route and schedule information (i.e., General Transit Feed Specification, discussed below) there is not yet a comparable standard for the intersection of multiple modes such as transit, bicycling, or walking, or for some of the infrastructure (e.g., benches, shelters, bicycle parking) that supports use of transit.
3. The total cost of ownership of walking, cycling, and supporting infrastructure data, including collection and maintenance, is very expensive when a single entity (e.g., private company, transit agency) is responsible for the data.

A promising strategy for overcoming these barriers is to take an open approach for providing data and information services. Open-sharing of information, advocated by transit agencies such as TriMet, encourages the improvement, evolution, and innovation of services to the general public since the data is available to everyone without copyright restrictions or licensing fees. For example, OpenStreetMap (OSM), maintained by the non-profit OpenStreetMap Foundation, is an open, freely available international repository of geographic data. The public, or organizations such as transit agencies with public-domain data, can add geographic information to the system. Other people and organizations can easily share and view the data, by requesting a download or viewing a simple webpage-based map. Anyone with an interest in providing and using data about an area is free to do so. Anyone who finds an error in the data is free to correct it. This ability to gather information from multiple different sources is referred to as crowd-sourcing. Thus, the crowd is able to contribute large amounts of information, instead of one organization being responsible for creating and updating the data. Crowd-sourcing can reduce data costs, but does provide other challenges such as quality control of the data. Many open-source software tools exist to work with OSM data. Open-source software allows communities to collaborate on software projects, providing a robust and stable base for improvements. While the open-source software is often free, there is a cost to maintain the software for one's own use. However, typically this cost is less than the cost of comparable proprietary solutions. Proprietary solutions also require dependence on a single

vendor; open-source solutions allow an agency to choose who they rely on for updates, support, and improvements, thereby allowing competition for services at a reduced cost.

In the transit industry, the availability of Google's free Google Transit trip-planning service has encouraged more than 125 public transportation agencies in the U.S. to put their data into the Google Transit Feed Specification (GTFS) format that Google uses to provide its Google Transit service. Google imports this data into its proprietary system to provide the Google Transit service (<http://www.google.com/transit>). However, correcting the data in any existing closed, proprietary systems or updating it (e.g., to reflect damage from natural disasters or newly available routes) requires long lead times and adherence to special procedures established by the vendor (e.g., Google) of the information system. Google Transit also does not offer multimodal data or services that integrate transit, walking, cycling, and park-and-ride lot information; although it offers walking routes, Google does not have complete data on the availability of sidewalks, and warns that the routes it suggests may be missing sidewalks or pedestrian paths. Furthermore, although the GTFS format is open, Google Transit does not serve as a common data repository where the underlying information can be freely shared without licensing fees or copyright restrictions. OSM, on the other hand, is designed to be a common data repository for freely available multimodal information.

As of the beginning of this project, there was no significant effort to support transit in the OSM system in the United States. OSM originated in the United Kingdom, and efforts to add U.S. geographic data to the system began in 2007. This presents a unique opportunity to establish solid guidance and tools for integrating U.S. transit information in OSM before fragmentation in data formatting conventions emerges. Additionally, a standard protocol and software tools are required in order to facilitate the automated update of the OSM data system whenever a transit agency releases a new GTFS dataset. Also, there must be some method of conveying edited information from OSM to the transit agency so the agency could review and vet the edits before integrating the changes into their datasets. These tools will reduce the amount of time and effort required by the transit agency to participate in the OSM community.

This project examined some of the key issues for multimodal trip planners that will use open, publicly available data to link bicycling, walking, and transit by exploring currently available software tools and data in this domain. OpenTripPlanner.org, a collaborative effort among TriMet, OpenPlans, and the developers of Five Points, OneBusAway, and Graphserver, was identified as the most promising open source multimodal trip planning software system with an active developer community. The project team set up an instance of OpenTripPlanner, using biking and walking data from OSM and GTFS data from Florida transit agencies, including an agency not currently participating in Google Transit, to examine the feasibility of the tool to route using multimodal data. The project team also recorded multimodal data for the Tampa, Florida, region in OSM to examine the current OSM coding conventions and determine the ability of the coding system to support functions required of a multimodal trip planner.

This project also investigated the use of open-source software to quickly increase the amount of multimodal data available in OpenStreetMap. The research team created GTFS-OSM-Sync (GO_Sync - <http://code.google.com/p/gtfs-osm-sync/>), a framework and open-source software tool for synchronizing transit data between the transit agency's official GTFS dataset and OSM. GO_Sync

connects the wealth of data from GTFS datasets with the ability of the OSM community to contribute crowd-sourced improvements to large datasets. The GO_Sync tool, therefore, enables public transportation agencies to upload GTFS data to OSM and retrieve crowd-sourced data, while the online community can edit and correct the bus stop locations and amenities based on existing GTFS data. Successfully translating GTFS data into the OSM format will enable over 125 transit agencies across the U.S. to share their public-domain data with the OSM community. Improved bus stop inventories will result in more accurate and complete multimodal trip planning and navigation information across car, bus, biking, and walking. A test deployment of GO_Sync in Tampa, Florida, has yielded improvements to 173 bus stop locations by OSM users. The project team also uploaded GTFS bus stop data into OSM for three other Florida transit agencies, which resulted in a total of 18,203 new bus stops being inserted into the OSM database. This is a significantly quicker and less expensive method of increasing the amount of multimodal data in OSM than manually hand-coding each bus stop by direct observation.

Based on this experience, the research team has drawn the following conclusions:

Implementing a multimodal trip planner using open-source software and open sources of data is very feasible. OTP contains most of the major features desired of a multimodal trip planner and has an active community that is continuing to add new features and improve the software.

Open data specifications and repositories, such as GTFS and OpenStreetMap, are designed to contain much of the data needed to support multimodal trip planning in OpenTripPlanner. OpenTripPlanner can also accept other sources of data as well, including National Elevation Dataset files and ESRI .shp files, to help supplement OpenStreetMap data. However, data coverage remains incomplete. Many public transportation agencies have yet to provide data about their stops, transit routes, and schedules in the GTFS format, which has become the de facto standard for representing this information. Transportation agencies tend to collect and maintain detailed data primarily for those facilities they are responsible for, and this tends to be major transportation infrastructure such as streets, bridges, and traffic signals. Open data repositories such as OpenStreetMap cover a wider range of facilities, including biking and pedestrian facilities, along all streets, not just major ones. However, coverage depends on the willingness and interest of individuals and community organizations to collect and record the data. As a result, geographic coverage in the OSM database is currently spotty, particularly in the U.S.

Additional online documentation and guidance for step-by-step creation of new OpenTripPlanner instances in new areas would be very helpful. Currently, the OpenTripPlanner tutorials require a high degree of technical knowledge in very specific tools used to implement OpenTripPlanner. By creating more robust documentation and online guidance, the implementation costs of new OpenTripPlanner sites would likely be significantly reduced.

Storing data for transit schedules and route alignments in OSM is impractical. We recommend that multimodal trip planners rely on transit data in GTFS format, obtained directly from transit agencies, for scheduling and routing, and use bus stop geographic locations to link transit data to data from other sources for other modal networks.

Transit trip planning requires a route schedule, and OSM was not designed to work with schedule data for any mode. In addition, OSM still does not have clear procedures for recording route information, and the procedures presently in use are complex and require levels of time and attention to technical detail that make them difficult for the average OSM contributor. OTP has been designed to use GTFS as its primary source of data for transit stops, routes, schedules, and other transit data, and to use data from other sources, such as OSM or commercial sources, for other modes. Transit agencies are the primary managers of transit data, including the locations of bus stops. Agencies that provide GTFS data to Google for the Google Transit trip planning service already have an incentive to keep their GTFS data files current. Agencies that partner with others to develop multimodal trip planners for their service areas should have a similar incentive. Agencies can use the GO_Sync software, developed as part of this project, to keep the bus stops current in both GTFS and OSM, so that the stops can be used to link transit data from different sources. However, GTFS data from the transit agency should also be treated as the primary transit data source for multimodal trip planning. This design also implies that all modifications to transit data, such as the improvements of bus stop geographic locations in OSM, must be communicated to the transit agency so the agency can update its dataset. Future research should identify possible barriers for transit agencies integrating data improvements into their own datasets, and identify possible improvements to software such as GO_Sync that could facilitate this process.

Most multimodal trip planners will need to integrate data from a variety of sources and make assumptions when data are unavailable due to data gaps. Although OSM and GTFS can together provide the essential data required by multimodal trip planners, coverage within OSM depends on activity and interest by people and agencies willing to contribute data. As a result, geographic area and modal coverage is often incomplete. Generally, developers of multimodal trip planners should expect to draw on additional sources of data, including data from various transportation agencies, citizen groups such as bicycling clubs, and possibly commercial sources.

Except for transit, when data describing infrastructure are missing, it is possible to use default assumptions when routing. Data collection can then be targeted toward specific geographic areas, or toward specific features, to improve the performance or scope of the trip planner.

The key to relying on crowd-sourced mapping is in motivating people to become involved in collecting and contributing data. Additional research is needed to identify what approaches are effective, and which groups within the general population are most amenable to participating in crowd-sourced data collection activities. OSM and other crowd-sourced information projects depend upon *cognitive surplus*—the free time of an educated, Internet-connected citizenry. The phenomenon of focusing this cognitive surplus toward specific objectives is relatively new, because it depends on technological and market developments that have reduced the barriers to collecting data (e.g., GPS receivers, digital still and video cameras), producing information (editing and reformatting data), and sharing it (via the Internet).

There have been few successful examples of focusing cognitive surplus—and little research on them—to predict which efforts will draw large numbers of participants, or to offer guidance on how to motivate participation in any particular effort. The research team believes it should be possible to engage neighborhood and other community organizations (e.g., Adopt-A-Road programs,

bicycling clubs, etc.) in gathering and reporting data on transportation infrastructure that could be used in multimodal trip planning (e.g., sidewalks, crosswalks, shelters at bus stops). Engaging the community to participate in these activities will probably be most successful if they can show citizens how making such an effort will benefit them directly. And such efforts probably will be more successful if the data collection can be organized in ways that make it easy for participants to observe and record data without requiring each person to do the work of recording the data in OSM or other data repository. For example, many individuals from an organization could record data, and one person from that organization could be responsible for adding the data to OSM. More research is needed to identify, develop, and document successful strategies for mobilizing communities to map their transportation infrastructure. Members of the research team are seeking opportunities to work with communities that are interested in mapping their transportation infrastructure, in order to better understand this process and the motivations for participation. Team members have identified possible leads in the Sarasota and Tampa, Florida, areas.

Risks from using crowd-sourced data in a public project appear to be small and manageable.

During the project team’s discussions about the project with other organizations, several people expressed concern that, if anyone can contribute data, then anyone can introduce errors into the data, either willfully or inadvertently, and thereby render useless any tools that rely on the data. Such occurrences appear to be rare in OSM and, when there is an active community of people contributing data in an area, users tends to find and correct such errors fairly quickly. In addition, trip planners use downloaded copies of the OSM data because the OSM system is not designed to serve data in real-time for trip planning websites. When a new copy is made, it can be compared with the previous copy to identify changes and their sources, and decisions can be made whether to accept the changes or ignore them. Similarly, a transit agency or other public agency that contributes data to OSM still retains its own data, and it can decide whether or not to accept corrections and improvements made by the OSM community. There also are risks from using official, public data sources, or commercial ones, because these data can be compiled without adequate field checking, or they can become outdated if budgets are not adequate for keeping the data current.

Changes to the OSM tagging system would improve its usefulness for multimodal trip planners. For example, additional tagging options are needed to describe how bicycle lanes are placed at intersections, in order to advise bicyclists what to expect as they approach an intersection that is unfamiliar to them. In addition, some cycling facilities such as wide paved shoulders, wide outside lanes, and lanes designated with sharrows (i.e., shared-lane markings) also lack tagging options in OSM. However, the bicycling community needs to first agree on what type of information needs to be communicated to a user of a multimodal trip planner when presenting bicycling options. Terminology for describing conditions to persons who may not be expert cyclists also needs to be considered. Additional research is needed to support defining a more inclusive tagging system for bicycles.

Additional OSM tagging options are needed to indicate whether facilities—especially for walking, cycling, or waiting for transit—have shade during summer months.

Additional OSM tagging options are needed to record information about the space, if any, between a sidewalk and a parallel street.

OSM's option for tagging crosswalks should be extended to apply to ways (i.e., the path of a crosswalk across the street right-of-way) instead of just to nodes (i.e., the intersection of the crosswalk and the street).

If adopted, the recommended changes outlined in this report would be relatively easy for OSM contributors to use. The relative simplicity of the proposed changes should improve their prospects for adoption by the OSM community.

On the other hand, some of the data desired for use in a multimodal trip planner could, in principle, be recorded in OSM, but the difficulty or hazard in collecting the data make it unlikely that even interested members of the public could collect and record them. Among these are data on volumes of vehicle traffic, width of lanes on busy streets, and the operating characteristics of pedestrian signals at intersections. It may be possible to reduce the need for traffic volumes by developing an alternative pedestrian or bicycling level-of-service measure that requires only easily observable data. Agencies responsible for installing and maintaining pedestrian signals should have data on how the signals operate, and these can be merged with OSM data.

Synchronizing data between OSM and other data sources, such as GTFS, is feasible for point (i.e., node) data such as bus stops, but it must be done carefully to avoid damaging data that others have previously contributed to OSM. The GO_Sync software developed as part of this project demonstrated that point data can be synchronized, and that this can be done fairly easily in an urban area for feature types, such as bus stops, that have not previously been entered into OSM for the area. The synchronization allows external sources of data, such as transit agencies, to avail themselves of corrections and supplemental data recorded by OSM contributors, and it enables the OSM community to benefit by receiving updated data when organizations responsible for an activity make changes in it. In areas where individual contributors already have entered substantial amounts of data about the feature, initial steps may require more effort to avoid conflicts between OSM and external data.

Fully automating the synchronization of line (i.e., way) data between OSM and other sources is not practical, although methodologies for efficient manual synchronization supported by GIS tools are emerging. Synchronization of line (i.e., way) data is much more complex than for point data, especially when OSM relations are involved. Characteristics of a street can and do change along its length, and the line representing the street needs to be split to reflect each change. Relations, a type of tool that OSM uses to group line segments into long routes (such as numbered state highways, or individual bus routes), are not well understood by many beginning contributors to OSM and are easily damaged. OSM offers more than one way to represent bus routes in the database. Each of these characteristics makes line synchronization difficult and error-prone, and the research team recommends against automatically synchronizing line data between OSM and external sources, especially if the external data are expected to change frequently.

While the full automation of line data is discouraged, methodologies are emerging that would allow a manual merge and upload of existing line datasets into OSM, with some data analysis supporting the merge process being automated by Geographic Information Systems tools such as ArcGIS. For example, TriMet in Portland, Oregon, is overseeing an OpenStreetMap improvement project for the Portland area in order to improve local data coverage to support multimodal trip planning. TriMet is using a combination of traditional GIS tools such as ArcGIS as well as OSM tools such as JOSM in order to upload existing line datasets into OSM. In addition, Mapquest.com is partnering with OSM to improve OSM's data for the U.S.

Additional research is needed on what information needs to be communicated to travelers for multimodal trips, and on how best to effectively communicate it. People tend to use different skills when navigating through a street or sidewalk network, and multimodal trip planners need to be able to present information effectively for each of these types of audiences. One navigation skill is reading and interpreting a map for spatial navigation. Another is following a detailed set of instructions for procedural navigation. Trip planners for driving provide information to support each of these skills, and multimodal trip planners also need to do so. In general, street names are useful and adequate for providing step-by-step instructions, including for walking on sidewalks that parallel the streets. However, additional research is needed on how best to provide step-by-step instructions for walking or cycling where there are no parallel streets to use as references (e.g., parks, university campuses). GPS-enabled smart phones may be one tool for providing these instructions in real time (e.g., by providing a first-person view with superimposed arrows). Such phones are not in universal use, and the use of GPS services can be a significant drain on mobile device batteries. The ability of modern consumer-grade GPS-enabled cell phones, and new embedded sensors, to provide significantly accurate real-time position updates that could be used for real-time pedestrian navigation instructions, both outdoors and indoors, should also be examined.

People have different perceptions of what kinds of streets, sidewalks, crosswalks, and intersections are safe for bicycling and walking. If one of the purposes of building a multimodal trip planner is to encourage more people to walk, bicycle, or ride transit, then the trip planner needs to use objective data in its calculations and in presenting its results. However, its design also needs to be sensitive to different perceptions of risk and comfort which will be held by different users. Additional research is needed to identify the information that should be presented to the user by default, and what information needs to be available if requested. This is particularly true for bicycling, which occupies an ambiguous middle position between being treated as motor vehicle traffic in some regards but not in others. There seem to be no standards for the type of information to be displayed on maps for bicyclists or for how it should be displayed.

The effect of combining subjective ratings of routes or facilities with objective data to assist users in choosing a route or to encourage them to try a mode they typically do not use is another research need.

Additional research also is needed on the most effective ways to display route information from Internet-based trip planners, both generally and to users who may prefer to see certain types of

information on a map of the route (e.g., potential cyclists may want to see information on speed or volume of motor traffic along a route).

The next step to extend the research in this project is to create a public pilot deployment of OpenTripPlanner for a Florida city or county. The next step to extend the research in this project is to create a pilot deployment of OpenTripPlanner for a Florida city or county that is willing to work with the research team on tasks necessary to create and publicly promote use of an OpenTripPlanner website. The research team recommends the selection of a city with a transit agency that already has its data in the GTFS format or an agency that is willing to create a GTFS dataset for use with OpenTripPlanner. Initial pilot deployments of OpenTripPlanner should also be targeted at areas with existing pedestrian infrastructure and traffic, such as downtowns, parks, or university campuses, to best explore a wide range of trip planner use.

A pilot deployment of OpenTripPlanner would be very valuable as a base for conducting research on the topics mentioned above (including modifications to the pedestrian routing functions) and as a working example that will help generate interest by other communities in such systems. If OpenStreetMap data is sparse in pilot areas, alternate existing datasets (e.g., county pedestrian centerline datasets) could be investigated to help fill in these gaps.

TABLE OF CONTENTS

Executive Summary.....	vii
Chapter One - Introduction.....	1
Project Objectives	4
Report Organization.....	5
Chapter Two - Multimodal Trip Planning Technologies.....	6
General Transit Feed Specification	6
OpenStreetMap	7
OpenTripPlanner.....	10
Chapter Three – Data Requirements for Multimodal Trip Planners.....	14
Overview of Functional Requirements	14
Defining the origin and destination.....	14
Data Describing A Connected Network Between The Origin and Destination	15
Routing Algorithm – A procedure to find paths through networks from the origin to the destination	17
Including User preferences in the routing algorithm	18
Presenting the results to the user.....	21
Overview of Data Requirements.....	24
General Transit Feed Specification Data.....	27
Street network	28
Points of interest and landmarks.....	28
Address range and location file.....	29
Lists of businesses and points of interest, with locations or addresses.....	29
Other modal networks: transit	29
Other modal networks: Bicycling and Walking.....	31
Restrictions on turning, access, direction	31
Location of traffic signals.....	31
Transit schedules.....	31
Posted speed limits.....	31
Tolls, ferries, and road classifications	32
Transit fares.....	32
Type of transit vehicle.....	32
Ability of transit vehicle to carry bicycles	32
Wheelchair access.....	33
Amenities.....	33
Internet Links.....	33
Values to calculate suitability or level of service	33
Informal routes and shortcuts	33
Intersections and Crosswalks.....	34
OpenStreetMap Data.....	34

Street network	36
Points of interest and landmarks.....	36
Address range and location file.....	37
Lists of businesses and points of interest, with locations or addresses.....	37
Other modal networks: transit	37
Other modal networks: bicycling.....	37
Other modal networks: walking	39
Restrictions on turning, access, direction	39
Location of traffic signals.....	39
Transit schedules.....	40
Posted speed limits.....	40
Tolls, ferries, and road classifications	40
Transit fares.....	41
Type of transit vehicle.....	41
Ability of transit vehicle to carry bicycles	41
Wheelchair access.....	41
Amenities.....	41
Values to calculate suitability or level of service	42
Informal routes and shortcuts	47
Intersections and Crosswalks.....	47
Curb cuts and Tactile Surfaces	50
Data outside the scope of GTFS and OpenStreetMap	52
Digital Elevation.....	52
Suitability of Links for Bicycling or Walking.....	52
Crash or safety history	53
OpenTripPlanner – Relationship to Data and Data Requirements	53
Discussion of Open Multimodal Trip Planning.....	56
OpenTripPlanner Software Setup and Maintenance.....	57
Generation and Maintenance of Transit Data.....	57
Data Coverage.....	58
Motivating the Crowd - OpenStreetMap Participation	59
Extending the OSM tagging system.....	61
Crowd-sourced Data Integrity, Quality, and Security.....	62
Licensing issues	64
Chapter Four – GTFS-OSM-Sync – A Tool to Synchronize Transit Data with OpenStreetMap	67
Related Work	67
Problem Definition.....	68
OpenStreetMap Data.....	69
Potential Bus Stop Conflicts	69

Framework Design	71
Bus Stop Comparisons.....	72
Bus Stop Tag Comparisons	74
Bus Route Comparisons	75
Modifications to OSM Data.....	75
Sample Execution.....	77
Evaluation	78
Relevance to other datasets	79
GO_Sync Summary.....	83
Chapter Five - Conclusions.....	85
Summary of Project Findings	85
Feasibility.....	85
Use of GTFS data	85
Data coverage from other sources.....	86
Suggested extensions to OSM tagging system	86
Synchronization of data between OSM and external sources	87
Open Challenges in Multimodal Trip Planning	88
Data.....	88
Communication.....	89
Next Steps	90
References	92

LIST OF FIGURES

Figure 1 – A GTFS dataset from a transit agency consists of several text files within a zip file.....	7
Figure 2 – The OpenStreetMap (OSM) website interface allows any registered user to edit geographic data in the main OSM repository	9
Figure 3 – An OpenTripPlanner deployment created by the research team shows transit directions for the USF campus using HART (Bus 5) and a transfer to the USF Bull Runner Shuttle (Bus D)	11
Figure 4 – The process of creating a new instance of OpenTripPlanner using GTFS and OpenStreetMap data.....	13
Figure 5 – A graph used by routing algorithms consists of nodes (vertices) joined by edges (ways or links).....	18
Figure 6 – Route through University of South Florida campus, and associated walking directions.....	22
Figure 7 – A green-to-red gradation of colors in an area with few green and many red streets gives greater emphasis to red streets (less desirable) and makes it difficult to discern green one (more desirable).....	23
Figure 8 – OpenStreetMap uses key-value pairs to tag objects with attributes, such as the tags shown here for a bus stop.....	35
Figure 9 – Example intersection to illustrate tagging of crosswalks, sidewalks, curb cuts, and signals	48
Figure 10 – OpenTripPlanner shows an elevation graph beneath the map for a bike trip.....	52
Figure 11 – Routing optimization options in OpenTripPlanner	55
Figure 12 – Using a custom thesaurus geocoder, OpenTripPlanner developers can add custom locations, such as the USF Library and nearby McDonald’s restaurant	56
Figure 13 - 50 th Street bus stop conflicts.....	70
Figure 14 – Holly Drive bus stop conflicts	70
Figure 15 – Alumni Drive bus stop conflicts.....	70
Figure 16 – The matching algorithm used by GO_Sync for GTFS and OSM datasets.....	73
Figure 17 – The user interface of the GO_Sync Framework which shows the differences between General Transit Feed Specification datasets and bus stops in OpenStreetMap	74
Figure 18 – The GO_Sync Data Flow Diagram	76
Figure 19 – OSM bus stop data for Hillsborough County before using GO_Sync (a), and after using GO_Sync (b), showing an increase of 3679 bus stops.....	78
Figure 20 – OpenStreetMap users have modified the geographic location of 173 bus stops in the Tampa, Florida, area.....	79
Figure 21 – OpenStreetMap users have modified the geographic location of 6 bus stops in the Pinellas Suncoast Transit Area coverage	81
Figure 22 – OpenStreetMap users have modified the geographic location of 2 bus stops in the Space Coast Area Transit coverage.....	81
Figure 23 – OpenStreetMap users have modified the geographic location of 3 bus stops in the Miami-Dade Transit coverage.....	82

LIST OF TABLES

Table 1 - Data requirements for ideal multimodal trip planners.....	25
Table 2 - Data needed to calculate bicycle and pedestrian level of service measures	43

LIST OF ACRONYMS

ADA	Americans with Disabilities Act of 1990
API	Application Programming Interface
BLOS	Bicycle Level of Service
CC-BY-SA	Creative Commons Attribution Share-Alike
CPU	Central Processing Unit
CSV	comma-separated values
CUTR	Center for Urban Transportation Research
ESRI	Environmental Systems Research Institute
FDOT	Florida Department of Transportation
GIRO	<i>Le Groupe en Informatique et Recherche Operationnelle, Inc.</i> , Montreal, Canada
GIS	Geographic Information System(s)
GPS	Global Positioning System
GTFS	General Transit Feed Specification
GUI	Graphic User Interface
GNU	GNU's Not UNIX
HART	Hillsborough Area Regional Transit
HASTUS	<i>Horaires et Assignations pour Système de Transport Urbain et Semi-urbain</i> (GIRO Transit Trip Scheduling Software)
ID	Identifier
Java EE	Java Enterprise Edition
JOSM	Java OpenStreetMap editor
JRE	Java Runtime Environment
LGPL	Lesser General Public License
LOS	Level of Service
MDT	Miami-Dade Transit
MPO	Metropolitan Planning Organization
NED	National Elevation Database
ODbL	Open Database License
OSM	OpenStreetMap
OTP	OpenTripPlanner
POI	Points-of-Interest
PLOS	Pedestrian Level of Service
PSTA	Pinellas Suncoast Transportation Authority
RAM	Random Access Memory
RCI	Roadway Characteristics Inventory
SCAT	Sarasota County Area Transit <i>or</i> Space Coast Area Transit
SI	International System of Units
TAD	Travel Assistance Device
TIGER	Topologically Integrated Geographic Encoding and Referencing
TRANSTAT	Florida Department of Transportation's Transportation Statistics Office
USF	University of South Florida
WGS	World Geodetic System
XAPI	Extended Application Programming Interface
XML	Extensible Markup Language

CHAPTER ONE - INTRODUCTION

The integration of multimodal data and information systems, including Global Positioning Systems (GPS) technologies and mobile devices (e.g., cell phones), is part of the next evolution of advanced traveler information systems. The Federal Transit Administration Intelligent Transportation System Program has identified multimodal technologies as part of their 2009–2029 Strategic Plan [1]. However, most transit information is currently locked into proprietary formats and systems and cannot be easily shared, viewed, updated, or co-mingled without permission from the vendor and expert data analysis. This limits the ability of transit agencies and others to provide information resources, such as trip planning tools, that support the use of transit or other alternatives to driving. As examples of some of the obstacles:

- A transit agency that wants to integrate walking, bicycling, and park-and-ride access with its Google Transit routing application must either wait for Google to decide to do the work (Google has not indicated that it has any plans to add these data or capabilities to its system); or else it must bear the expense itself of recreating the same type of system to add the functionality it desires.
- A recent multimodal trip planning system, Goroo.com, developed for the Chicago metropolitan area, cost \$1.0 million [2]. Although the Goroo.com system is truly intermodal for walking, driving, bus transit, and rail transit—meaning that it plans trips that allow the user to walk or drive to transit, and then walk from transit to the final destination—it has significant limitations. It does not include bicycling (although this is planned); it is not clear that it actually has data on sidewalks; and its present coverage is limited to the Chicago area for which it was developed. Developing a comparable system for another area would be expensive, whether this is done from scratch or done by acquiring the Goroo.com software and populating it with data for the new area.
- Other multimodal trip planning systems exist but have limited functionality. For example, OpenRouteService.org, which provides coverage in much of Europe, can provide information about how to get between two locations by bicycle, or by walking, or by driving, but it does not work with transit, and it was not designed to plan a trip that begins using one mode (e.g., walking) but needs to switch to another (e.g., transit) to reach the destination. The service uses publicly-available data, but the software is not publicly available and would have to be duplicated elsewhere.

There are three major barriers to developing multimodal trip planning services that can work in more than a single U.S. metropolitan area.

- The available high-quality data on street networks is provided by private vendors, such as NAVTEQ or Tele Atlas, to whom one must pay licensing and use fees.

- Although there is an evolving standard for recording transit route and schedule information (discussed below) there is not yet a comparable, uniform dataset of infrastructure for multiple modes such as transit, bicycling, or for walking, or for some of the infrastructure (benches, shelters, bicycle parking) that supports use of transit.
- Even if standards existed for the walking, cycling, and supporting infrastructure, it would be expensive to collect, code, and maintain the necessary data using conventional approaches managed by a single entity (e.g., private company, transit agency).

A promising strategy for overcoming these barriers is to take an open approach for providing data and information services. Open-sharing of information, advocated by transit agencies such as TriMet, encourages the improvement, evolution, and innovation of services to the general public since the data is available to everyone without copyright restrictions or licensing fees. For example, OpenStreetMap, maintained by the non-profit OpenStreetMap Foundation, is an open, freely available international repository of geographic data. The public or organizations such as transit agencies with public-domain data can add geographic information to the system that can be easily shared and viewed by others through a simple webpage-based map. Anyone with an interest in providing and using data about an area is free to do so. Anyone who finds an error in the data is free to correct it.

A significant difference between this option and the closed proprietary systems (e.g., Google Maps, Google Transit) is that anyone can export data from OpenStreetMap and use this information to build new information services without paying licensing fees to a company. Such an open information resource thus enables a transit agency or other organization to focus on adding the functionality it desires to an existing open-source trip-planning system, rather than on bearing the cost (including licensing fees) to obtain data and recreate a service such as multimodal trip planning from scratch. Open-source software also allows communities to collaborate on software projects, providing a robust and stable base for improvements. While the open-source software is often free, there is a cost to maintain the software for one's own use. However, typically this cost is less than the cost of comparable proprietary solutions. Proprietary solutions also require dependence on a single vendor; open-source solutions allow an agency to choose who they rely on for updates, support, and improvements, thereby allowing competition for services at a reduced cost.

Many open-source software tools exist to work with OSM data. Open-source software allows communities to collaborate on software projects, providing a robust and stable base for improvements. While the open-source software is often free, there is a cost to maintain the software for one's own use. However, typically this cost is less than the cost of comparable proprietary solutions. Proprietary solutions also require dependence on a single vendor; open-source solutions allow an agency to choose who they rely on for updates, support, and improvements, thereby allowing competition for services at a reduced cost.

Google's decision to provide its Google Transit online trip planner free of charge has encouraged more than 125 transit agencies in the U.S. to put their data into a common data format, known as the General Transit Feed Specification (GTFS). Google imports this data into their proprietary

system to provide the Google Transit service (<http://www.google.com/transit>). However, correcting the data in any existing closed, proprietary systems or updating it (e.g., to reflect damage from natural disasters) requires long lead times and adherence to special procedures established by the vendor of the information system. A transit agency that puts its data into the GTFS format for the first time can wait for months before Google incorporates it into the Google Transit service. In fact, Google has struggled to deal with the demand of new agencies wanting to join, and at times Google has said that it would not be able to accept new agencies into Google Transit [3].

This means that the transit agency can put their data in the GTFS format, but that the Google Transit service may not be immediately available for that agency. Google Transit also does not offer multimodal data or services that integrate transit, walking, cycling, and park-and-ride lot information. Although Google Maps offers walking routes, it does not have complete data on the availability of sidewalks, and warns that the routes it suggests may be missing sidewalks or pedestrian paths. Furthermore, although the GTFS format is open, Google Transit does not serve as a common data repository where the underlying information can be freely shared without licensing fees or copyright restrictions. OpenStreetMap, on the other hand, is designed to be a common data repository for freely available multimodal information.

So far there has been no significant effort to support transit in the OpenStreetMap system in the United States, or to develop a multimodal trip planner that can be implemented easily and inexpensively in multiple urban areas. OpenStreetMap originated in the United Kingdom, and efforts to add U.S. geographic data to the system began in 2007. This presents a unique opportunity to establish solid guidance and tools for integrating U.S. transit information in OpenStreetMap (OSM) before fragmentation in data formatting conventions emerges:

- 1) A standard protocol and software tools are required in order to facilitate the automated update of the OpenStreetMap data system whenever a transit agency releases a new GTFS dataset.
- 2) A method of conveying edited information from OSM to the transit agency should be created so the agency could review and vet the edits before integrating the changes into their datasets. These tools will reduce the amount of time and effort required by the transit agency to participate in the OSM community.

Creating a tool to translate GTFS data into the OpenStreetMap format will enable over 125 transit agencies to import their transit data into OpenStreetMap and share this data with each other, the general public, and open-source software developers. Transit agencies can then use contributions from this community to improve the accuracy of their bus stop and route information, as well as the accuracy of nearby locations of amenities that enable multimodal transportation including bike racks, sidewalks, crosswalks, traffic signals, etc. Some amenities information will be available for upload from public domain data files maintained by local governments. The rest would be contributed by citizens, neighborhoods, or local communities interested in having better trip planning and, ultimately, better transit service available.

This project identified and examined some of the key issues for multimodal trip planners that will use open, publicly available data to link bicycling, walking, and transit by exploring currently available software tools and data in this domain. The research team reviewed several existing multimodal trip planning technologies, including Goroo in Chicago (<http://goroo.com/>) and GraphServer (<http://graphserver.github.com/graphserver/>), for ease of transferability to other cities, use of open data sources, and ease of integration with a website. OpenTripPlanner (<http://opentripplanner.org/>), a collaborative effort among TriMet, OpenPlans, and the developers of Five Points, OneBusAway, and Graphserver, was identified as the most promising open-source multimodal trip planning software system with an active developer community that uses open data sources. Goroo does provide intermodal data for walking, driving, bus transit, rail transit, and biking. However, it is unclear if Goroo actually has data on sidewalks for pedestrian routing or information (e.g., crosswalks, pedestrian signals) for crossing streets. Most importantly, the actual Goroo trip planning system and data are proprietary, and therefore not easily transferred to other cities. Graphserver is an open-source multimodal trip routing engine that uses open data sources such as GTFS and OpenStreetMap. However, Graphserver does not currently offer ready integration with Web-based technologies that are necessary for offering a multimodal trip planning website to the general public. OpenTripPlanner is open-source, uses open data sources such as GTFS and OpenStreetMap, and is readily deployable as a multimodal trip planning website that can be offered via industry-standard Web browsers.

The project team set up an instance of OpenTripPlanner on a USF server (<http://opentripplanner.usf.edu/>), using biking and walking data from OSM and GTFS data from Florida transit agencies, including an agency not now included in Google Transit, in order to examine the feasibility of the tool to route over multimodal data. The project team also recorded multimodal data for the Tampa, Florida, region in OSM to examine the current OSM coding conventions and determine the ability of the coding system to support functions required of a multimodal trip planner. The project also demonstrated the ability to upload data that Florida transit agencies have prepared into OpenStreetMap.

PROJECT OBJECTIVES

This project had the following objectives:

- Develop performance specifications for a fully-operational multimodal trip planning system (a high-level description of how the system should work and what it will do); include in these specifications the ability for a user to emphasize pedestrian and cyclist safety, rather than simply the length of the route or the time it takes to complete the trip.
- Examine coding standards for recording data about bicycling, walking, and transit infrastructure in OpenStreetMap, to support intermodal routing, and compare these conventions to the multimodal trip planning performance specifications
- Create an open-source software tool to transform transit data stored in the Google Transit Feed Specification format into OpenStreetMap and to convert transit data stored in OpenStreetMap back into the GTFS.

- Establish a working version of a multimodal trip planning system, using the OpenTripPlanner project. OTP uses data stored in OpenStreetMap and GTFS to compute routes including walking, bicycling, and transit as needed to accomplish the trip. This working prototype will be used to evaluate OpenTripPlanner and the performance specifications.

REPORT ORGANIZATION

The remainder of the report is organized as follows. Chapter 2 provides an overview of the General Transit Feed Specification, the format used by over 125 U.S. agencies to represent transit data for use by applications such as Google Transit, OpenStreetMap, the repository of open pedestrian and bicycle data used for multimodal data routing, and OpenTripPlanner, the open-source multimodal trip planner examined within this project. Chapter 3 provides a summary of data requirements for a multimodal trip planner and how these requirements relate to the technologies presented in Chapter 2. Chapter 4 presents GTFS-OSM-Sync (GO_Sync – <http://code.google.com/p/gtfs-osm-sync/>), a framework and open-source software tool for synchronizing transit data between the transit agency’s official GTFS dataset and OSM. Chapter 5 concludes this report with a summary of project findings, open challenges and issues in multimodal trip planning, and future research topics in this area.

CHAPTER TWO - MULTIMODAL TRIP PLANNING TECHNOLOGIES

There are three essential components that must be used to create an effective multimodal trip planner. First, up-to-date data, which is uniformly formatted to particular guidelines, must be available for all modes over which a user can be routed. For multimodal trip planners, this typically includes information for walking, biking, public transportation (e.g., bus, rail, subway), and car.

Once data is available, it must be transformed into a graph or network format so that routes can be calculated from one area on the graph to another. These routes must obey the structure of connections within the graph, or topology, and rules defining what type of routes are allowed (e.g., a single direction of travel on one-way streets, cars can be routed on paths labeled as “roads” but not “sidewalks”, presence of transit service at the time when a rider wants to use the bus, etc.).

A routing engine, or trip planner, is software that is able to use a specific routing algorithm, in conjunction with the graph and rules containing information about the transportation infrastructure, in order to calculate a path from a starting location on the graph to an ending location. Typically, this route is then visualized using a map. As implicitly stated above, the data (and corresponding graph) needs to be frequently updated in order to provide directions which are relevant considering the current state of the transportation infrastructure.

Specific examples of the above components as related to multimodal trip planners have quickly emerged over the last five years. The General Transit Feed Specification (GTFS) defines a format in which transit agencies can represent their data for use in routing algorithms to find travel directions via transit. OpenStreetMap (OSM) is a world-wide repository of geographic information, including data for pedestrian infrastructure, public transportation, and roads for general purpose vehicle travel. OpenTripPlanner is an open-source routing engine and website that uses both GTFS and OSM data. Together, these components can form a complete multimodal trip planning system that can be modified and deployed to any server to provide multimodal trip planning services to the public for any area. The following sections provide an introduction to each of these three components in further detail. After these components are introduced, Chapter 3 discusses how each of these components is related to the data requirements for an ideal multimodal trip planner.

GENERAL TRANSIT FEED SPECIFICATION

The General Transit Feed Specification (GTFS, originally called the Google Transit Feed Specification) was created in 2005 by Google and TriMet as a means for transit agencies to represent their schedule, route, and bus stop data for use by the Google Transit Web-based trip planner [4]. Since this time, over 125 transit agencies in the U.S. have placed their data in the GTFS format and have benefited from the free Google Transit online trip planner. GTFS has also been used by many other applications that depend on up-to-date public transportation data, including

Microsoft’s Bing Maps, OneBusAway for real-time transit information in Seattle, BayTripper for real-time transit information in San Francisco, Carnegie Mellon University’s Tiramisu that provides real-time crowd-sourced transit information in Pittsburgh, and USF’s Travel Assistance Device (TAD) mobile phone application that provides real-time navigation instructions to users of public transportation [5][6][7][8][9]. GTFS has also evolved as a format since its creation in 2005. GTFS is now modified through a community process with input from transit agencies and transit app developers.

A GTFS dataset consists of several plain text files which have been formatted as Comma-Separated Values (CSV) and are contained within a single zip file, which is hosted on a transit agency’s website so software applications can automatically retrieve it. Figure 1 shows the contents of a *google_transit.zip* file from Hillsborough Area Regional Transit (HART), and the contents of the *stops.txt* file which contains information about the name, ID, and location of all HART bus stops. The *routes.txt* file contains information about the routes of the transit agency, the *calendar.txt* and *calendar_dates.txt* contain schedule information, and the *trips.txt* and *stop_times.txt* files contain information about the order of visitation of bus stops for a particular route according to a particular schedule. The *shapes.txt* contains information on the spatial representation of a route so it can be accurately drawn on a map. A full explanation of all files and fields is available on the GTFS website [4].

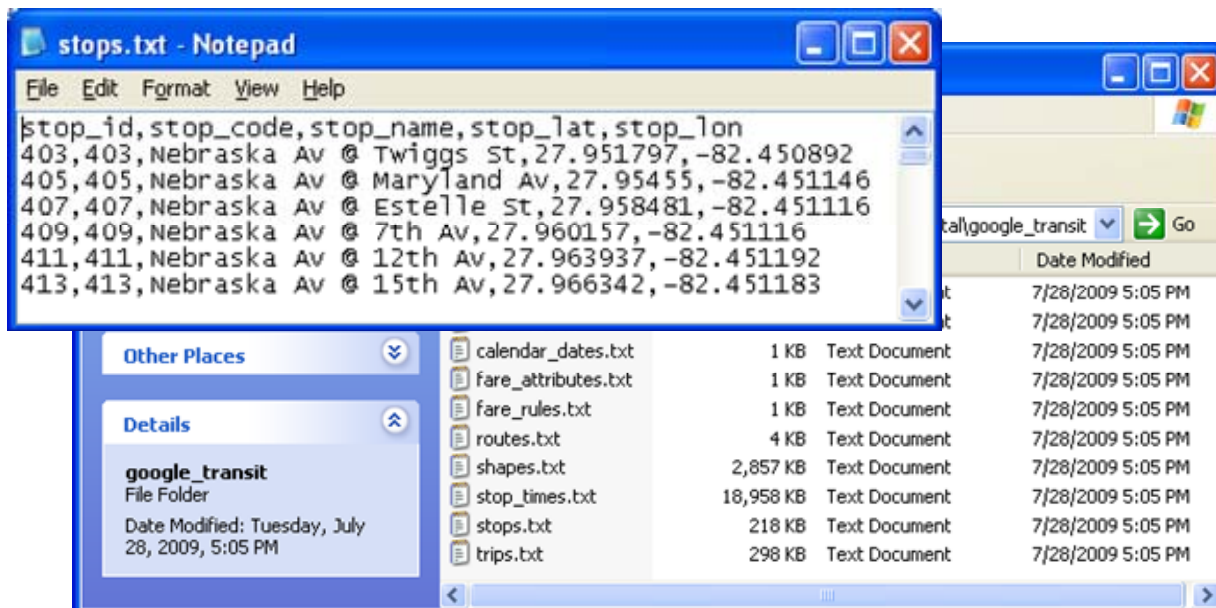


Figure 1 – A GTFS dataset from a transit agency consists of several text files within a zip file

OPENSTREETMAP

OpenStreetMap (OSM) began in England in 2004, and its aim is to create a copyright-free map of the world [10]. OSM is similar to Wikipedia, the free online encyclopedia, but it is a repository for geographic data. Examples of OSM data include the locations of bus stops, traffic signals, and bicycle parking; the paths of roads, sidewalks, and rivers; and the shape and extent of buildings, parks, and

airports. Persons who contribute data to the map agree to add only data that is free of copyright. The three primary sources of such data are:

- Direct personal observation – The observer owns the data and agrees to contribute them under terms of the project license. Observations can be made by using a GPS device to record a path or locations along it. Observations can also be made by using data already entered in OSM to cross-reference something observed in person (“Today I noticed that there is parking for 3 bicycles right next to the bus shelter that has been recorded at the intersection of X and Y streets”).
- Tracing over aerial imagery – To use this source, the imagery must be free of copyright, or it must have been donated to the OSM project. Several series of such imagery are built into one of the main tools for entering data.
- Uploading existing datasets - The third source uses data that must be in the public domain or the owner must give permission to upload the data into OSM. In the U.S., federal government geographic data and maps are in the public domain. Some data and maps held by state and local governments are not, although many agencies do grant permission to enter their data into OSM when asked. In other countries, governments may hold title to data whose U.S. equivalent would be in the public domain. These governments may charge a license or royalty fee to use the data, and they may choose to prevent others from copying the data.

Just as anyone can contribute data to OSM, anyone is free to use data from OSM, provided that they take three actions:

- credit OSM as the source
- do not limit the ability of others to use the data, and
- contribute back to OSM any improvements they make in data they acquire from OSM (e.g., if they correct errors or expand coverage).

These terms allow someone to use OSM data for a business (e.g., making maps and selling them), but the business must credit OSM as the source, and it cannot prevent its customers from copying the maps produced from OSM data.

At present, data in OSM is contributed and released under the project’s Creative Commons Attribution-Share Alike license, although the project is considering a change to a similar license to address some uncertainty over how the existing license applies to data [10]; pp. 235-240].

The OSM project has developed several editing tools that contributors can use to enter and edit data. Figure 2 is a screen shot from the Potlatch 1 editor [10]; pp. 99-108]. As with many graphics programs, the editor allows a user to create points, lines, and areas (sometimes referred to as objects), edit them, and attach information to them.

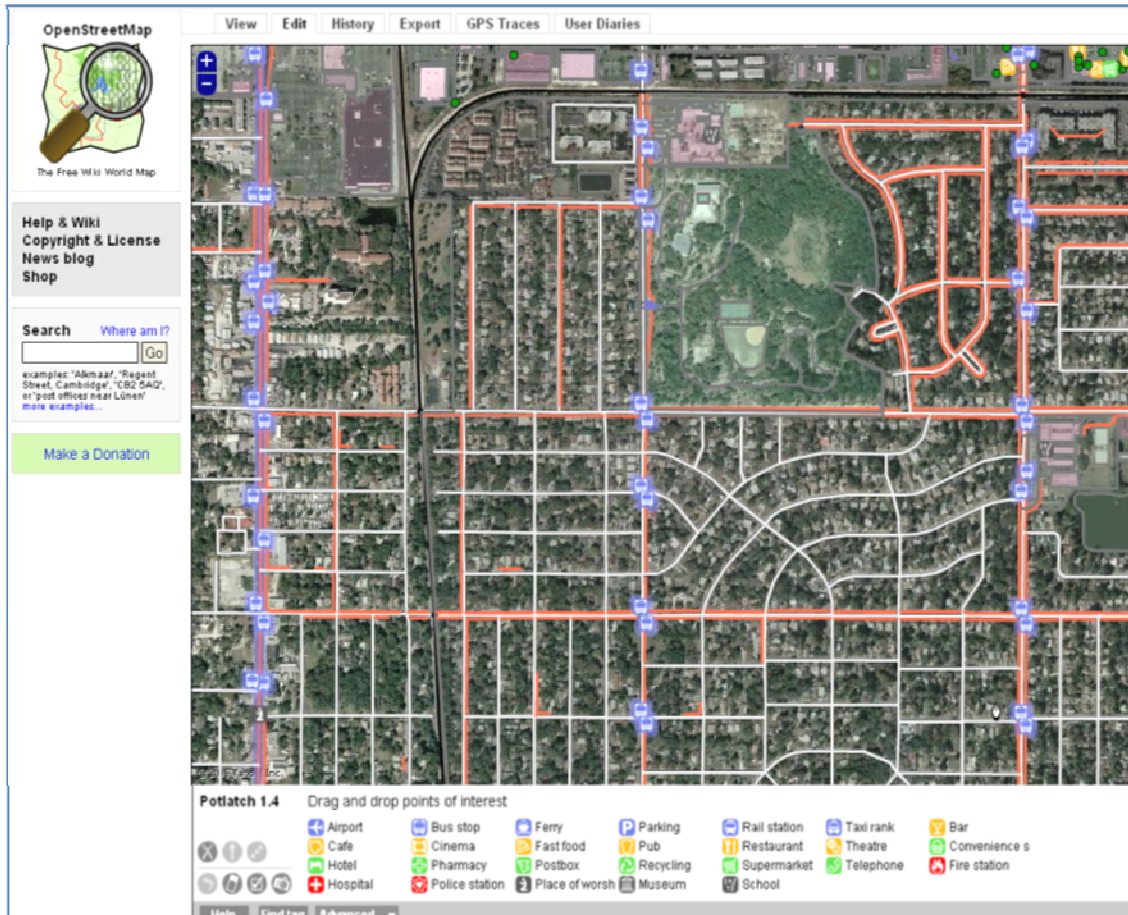


Figure 2 – The OpenStreetMap (OSM) website interface allows any registered user to edit geographic data in the main OSM repository

It is possible for each user to use a custom format to record characteristics of objects, but this would create confusion and greatly limit the usefulness of the data being contributed. Instead, using the project's wiki and its network of listserves and discussion forums, contributors have developed a system of tags for recording information and software to read data and tags and use the information to produce maps. The easiest way for most contributors to see the results of their work is to use the agreed-upon tags so that their contributions show up on the map. The tagging system does not cover every possibility. It evolves as contributors decide to add types of data that have not been included before, or as they encounter situations not covered by the existing tags. There are recommended procedures for proposing new tags or changes to existing ones. However, relatively few people participate in this process, and sensible changes become adopted without going through the process [10], p. 63]. The OSM community has developed a rich set of tools that contributors can use to see what tags are being used, how common they are, and where tags may have been incompletely or incorrectly used (e.g., streets without names, or roads that cross but that have not been recorded as intersecting or having a grade-separated crossing).

OPENTRIPLANNER

OpenTripPlanner is an open-source multimodal trip planner initiated by a collaborative effort among TriMet, OpenPlans, and the developers from Five Points, OneBusAway, and Graphserver [12]. The multimodal trip planner allows users to plan a trip by using multiple modes of transportation such as bike, train, bus, or on foot. In addition to finding the shortest path, OpenTripPlanner is able to route for a safest trip or fewest transfers. Wheelchair accessibility is also taken into consideration. A full discussion of these capabilities and related data requirements is in the following chapter.

Since the primary goal of OpenTripPlanner is to develop a complete open-source multimodal trip planner, the OpenTripPlanner source code is licensed under the GNU Lesser General Public License (LGPL) [13] and publicly available on the OpenTripPlanner.org website. As a result, any software developer can download, compile, and execute the source code to establish an OpenTripPlanner website on his or her own server for their own geographic area. To retain the integrity of the main public version of OpenTripPlanner available for download at OpenTripPlanner.org, independent developers must request permission from the website moderators (e.g., TriMet, OpenPlans, developers from Five Points, OneBusAway, and Graphserver) before they are allowed to commit any changes back to main publically-accessible OpenTripPlanner project. This model allows collaboration, but protects the integrity of the OpenTripPlanner project.

There are no license fees for downloading, modifying, or redistributing the OpenTripPlanner planner software. However, developers must pass the same rights on to users of applications developed from OpenTripPlanner. Unlike most enterprise software, OpenTripPlanner permits developers to check out its source code, modify the software, and apply the changes they make back to the open-source community project. This collaborative development is critical to the evolution of cost-effective trip planning software since one user can benefit from the improvements of OpenTripPlanner made by other users.

To investigate the various capabilities of the OpenTripPlanner software, the research team created a publicly-available deployment of OpenTripPlanner on a USF server (<http://opentripplanner.usf.edu/>) which provides multimodal trip planning capabilities for the Tampa, Florida, area (Figure 3). Similar to other online trip planners such as MapQuest and Google Maps, OpenTripPlanner supports detailed step-by-step directions with an interactive map. In addition to the ability to zoom in and out to view the overall trip, the software shows which routes users should take and where to make transfers. However, unlike other trip planners, OpenTripPlanner is able to provide a trip that uses different modes of transportation within the same trip. Moreover, with only one click, OpenTripPlanner allows individuals with disabilities to plan routes that avoid stairs. Cyclists particularly benefit from the planner with the ability to pick the factor that is most relevant to them: time, safety, or elevation.

The OpenTripPlanner website established by the USF research team also demonstrates a critical benefit of open data formats such as GTFS. As Figure 3 shows, the USF OpenTripPlanner website is able to plan bus trips that transfer from HART's Route 5 to the USF Bull Runner's Route D. While HART already maintains their data in GTFS format for Google Transit, the USF Bull Runner did not

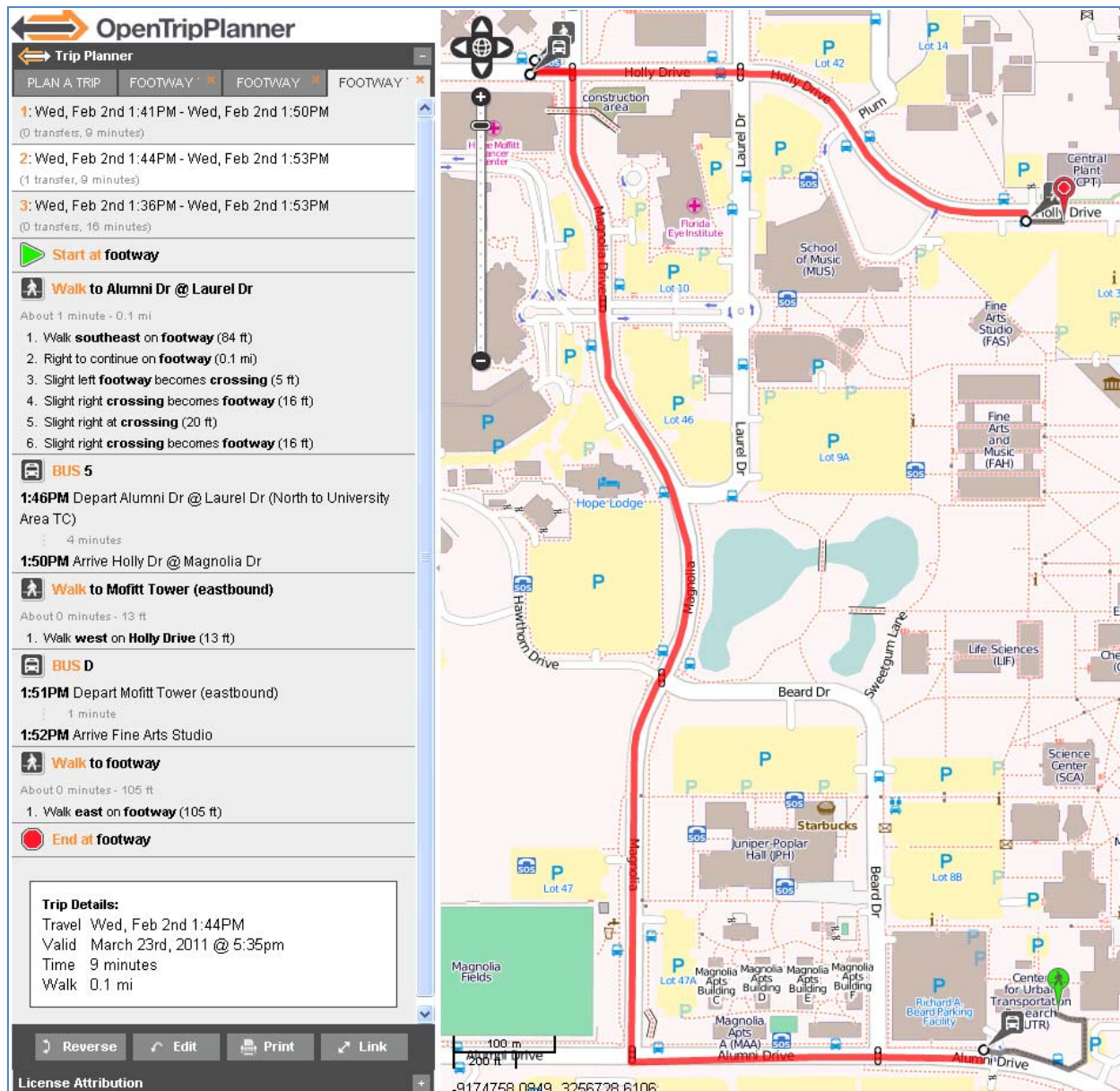


Figure 3 – An OpenTripPlanner deployment created by the research team shows transit directions for the USF campus using HART (Bus 5) and a transfer to the USF Bull Runner Shuttle (Bus D)

have its data in GTFS format at the beginning of this research project. However, because OpenTripPlanner uses GTFS data, and GTFS is an open specification, the research team was able to place the USF Bull Runner data into the GTFS format to offer transfers between the two transit systems on the USF OpenTripPlanner website. Currently, Google Transit does not offer transit trip planning on the USF Bull Runner shuttle service. Google Transit also does not provide trip planning for most Florida public transportation systems, either because agencies have not provided data in GTFS format to Google or Google has not yet processed the data. The research team's deployment of OpenTripPlanner demonstrates that it is possible to add new service to an OpenTripPlanner instance as soon as data is available.

The core of OpenTripPlanner consists of three basic software components: a graph builder, routing engine, and user interface. While the primary user interface (i.e., map and visual representation of route) is written in JavaScript, the other components are written in the Java programming language on the Java Enterprise Edition (Java EE) platform. OpenTripPlanner has an active developer community who contribute their efforts to improve the usability and functionality of OpenTripPlanner. Many resources are available online at <http://opentripplanner.org> that facilitate the process of deploying an OpenTripPlanner instance so that various transit agencies and developers can make use of the software.

The process of creating a new server instance of OpenTripPlanner on a new computer is shown in Figure 4. A graph object (i.e., graph.obj file) must exist before the OpenTripPlanner routing engine can plan a trip. Multiple data sources can be used to build the graph object for OpenTripPlanner: transit data from the agency's GTFS dataset, pedestrian and street infrastructure data from OSM, ESRI's ArcGIS shapefiles (*.shp), and elevation data from the National Elevation Dataset. The Graphbuilder software project is used to automate steps 1 through 4 in the process. Before Graphbuilder is run for the first time, a developer must first edit the graph-config.xml file to include information about the desired GTFS dataset(s) and NED data, as well as an output location for the graph.obj file. When Graphbuilder is then executed, it first retrieves and processes a GTFS dataset to obtain all needed transit information. Then, pedestrian and street data are retrieved from OSM for the area near the GTFS data. Optional NED elevation data can also be included in the graph. Graphbuilder then outputs a graph.obj file which represents the information needed for multimodal trip routing. After successfully building the graph object, the developer can deploy the OpenTripPlanner `opentripplanner-webapp` and `opentripplanner-api-webapp` projects to a Java Web application server (e.g., Apache Tomcat). When OpenTripPlanner is deployed, it loads the graph.obj file and is then ready for users to plan trips. Users interact with the system through the main OTP webpage, which shows forms and maps that aid in planning a trip. The routing algorithm is executed using the network information from loaded graph file every time a user plans a trip. In order for new data to be available for trips being planned in OpenTripPlanner, the graph file must be rebuilt. In other words, the Graphbuilder software must be executed again in order to retrieve the updated data from OSM, GTFS datasets, or NED data and build a new graph which incorporates the updated information. Then, OpenTripPlanner must be redeployed to the Java Web application server with the updated graph.obj file. Typically, this process is performed on a periodic basis in order to make updated transportation infrastructure information from OSM, GTFS, or NED available to users of an OpenTripPlanner deployment.

OpenTripPlanner is designed to be extensible for easy integration with other systems via a Web-based Application Programming Interface (API). In other words, the OpenTripPlanner website that provides the normal map-based user interface for OpenTripPlanner is separated from the routing functionality of the system. Therefore, existing Web or mobile applications can use the routing engine of OpenTripPlanner by interfacing directly with the `opentripplanner-api-webapp` software and retrieving step-by-step navigation directions for all supported modes of transportation.

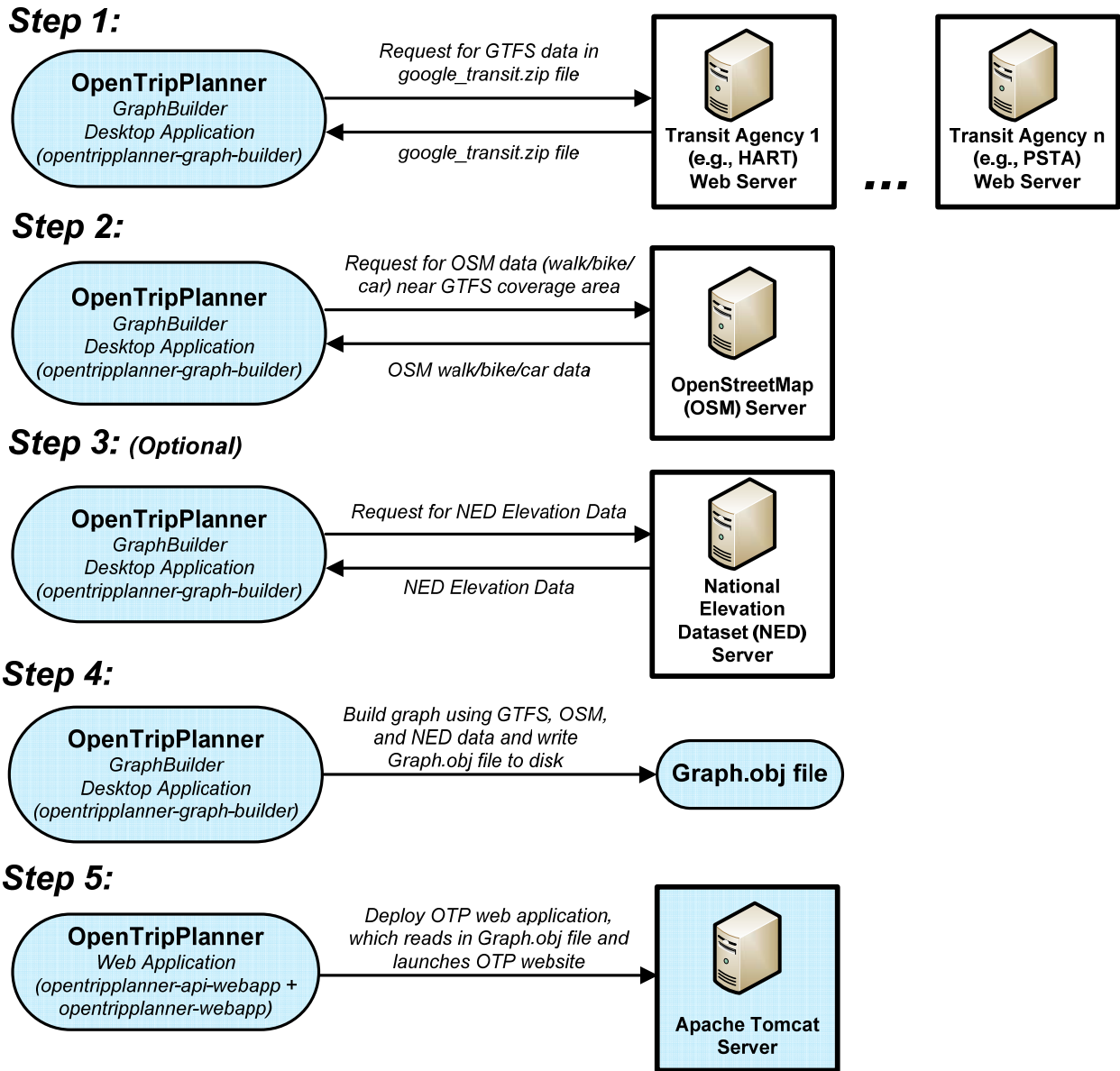
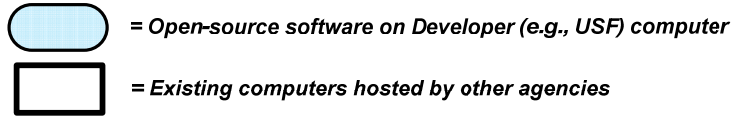


Figure 4 – The process of creating a new instance of OpenTripPlanner using GTFS and OpenStreetMap data

CHAPTER THREE – DATA REQUIREMENTS FOR MULTIMODAL TRIP PLANNERS

Data requirements depend on functional requirements—in this case, what a multimodal trip planner is intended to do, and the features it is expected to provide. The first section in this chapter discusses functional requirements. Subsequent sections present data requirements to support the functional requirements, and discuss meeting the data requirements from GTFS and OpenStreetMap data and how these requirements are related to OpenTripPlanner.

OVERVIEW OF FUNCTIONAL REQUIREMENTS

Several Internet-based trip planners offer national (U.S.) or global coverage for providing routes for car-based transportation (Google™, Yahoo!®, MapQuest®). They provide similar ways for selecting an origin and destination. Being in common use, these have come to define the experience and shape the expectations of users, especially for car-based travel. We discuss these first before broadening the discussion to multimodal trip planners.

At the simplest level, any trip planner finds a route between a starting point (origin) and a destination. A trip planner requires the following features, which are each discussed in detail in the following sections:

- a method to define the origin and destination
- data describing a connected network between the origin and destination
- a procedure (i.e., algorithm) to find paths through the network between the origin and destination
- a method to include user preferences in the routing algorithm
- a method to present the results to the user

DEFINING THE ORIGIN AND DESTINATION

Internet-based trip planners commonly allow users to define origins and destinations by

- Clicking on a map at the origin and/or destination
- Typing the address of the origin and/or destination
- Typing the name (e.g., business, point of interest, town name) of an origin and/or destination

- For mobile phones, using the phone's GPS hardware and software to find the phone's latitude and longitude, and then link it to the network data as if the user had clicked the location on a map

The first option requires a map and software that can convert the identified points into information (typically latitude and longitude) that can be linked to the network data. This map typically includes the street network and enough information on points of interest and landmarks that help the user figure out where he/she is on the map. The map needs to allow zooming in and out and panning to change the area being viewed.

The second and third options require data linking addresses and names to latitude and longitude (i.e., geocoding) that, again, can be linked to the network data. Several firms sell street network data that contain address ranges for each block and latitudes and longitudes of locations along the streets, and OSM can record similar information, although in most of the U.S. address information is currently too sparse to support trip planning. Software then matches an address to a section of street, which then becomes the origin or destination. Business and similar names may be linked to addresses and then linked to sections of streets, or they may be recorded with latitude and longitude and linked to the street network that way.

The fourth approach has been emerging over the last several years as GPS technology and navigation and mapping apps have become available to cell phone users. This approach is useful if the user does not know his or her location or address. It complements the use of address or name for the destination in cases where the user does not know the location of the destination to be able to click on it.

DATA DESCRIBING A CONNECTED NETWORK BETWEEN THE ORIGIN AND DESTINATION

For planning car trips, trip planners require data describing the street network, including speed limits, tolls, and any restrictions on driving activity. These restrictions may include the following:

- Turning - Can cars approaching an intersection from the south turn left, while those approaching from the west cannot? Are turns allowed only during selected hours or days of the week?
- Access - Is a street divided by a median with few passages through it, or is there a continuous left-turn lane? Where there is a passage through the median, is it one-way or two-way? Are U-turns permitted at a passage? At an intersection? Is a lane available for any car to use, or is its use restricted to carpools, or to emergency vehicles, or to those drivers willing to pay a toll?
- Directions - Is travel permitted in both directions, or just in one (and which way)?

As better real-time information becomes available on traffic congestion and actual vehicle speed, more advanced trip planning may become possible. For example, the route could be planned based on real-time conditions for a departure during the next few minutes of requesting the route.

Any trip planner is limited by the accuracy and completeness of its street data. Most major Internet-based trip planners rely on data collected and licensed by commercial firms. This can lead to long delays in updating network data, depending on the firms' schedules for making the updates. If a road or bridge is closed for an extended period because of a natural disaster, the data may not be updated for some time afterward, and trip planners may continue to find routes that use the damaged sections. Similarly, new interchanges or roads may not appear in the data for substantial periods following their completion. For example, a rebuilt interchange in the Tampa area (I-75 at Bruce B. Downs Boulevard) appeared in MapQuest's data within 8 months of completion, but it did not appear in Google's or Yahoo's for more than a year. Additionally, while mapping the area near the USF campus for OSM, the research team found one road that, based on its condition, appears to have been abandoned for years. Google still uses this abandoned road when planning routes, but does not yet use a road extension completed between 6 and 12 months before being mapped. While a few major Internet-based trip planners (e.g., Google, Microsoft Bing) are moving towards maintaining their own data, either in place of or to supplement licensed data, the process of reporting an error, and seeing that error resolved in the trip planner, can still be a lengthy process outside the control of the user.

Because anyone can edit data in OpenStreetMap, a contributor who lives in an area near a bridge closure or a new route can correct the road data as soon as the change is noticed. The speed with which OSM data are updated depends on the number of contributors in the area and on their engagement with the OSM project. The changes described in the preceding paragraph were corrected as soon as discovered by the one mapper working in the area, but the road abandonment and new construction had occurred an unknown amount of time before the mapper discovered them. In much of Europe, many people contribute to OSM in each city, so there are many observers to detect change. In most U.S. cities, there are only a few persons contributing in each city, and changes may take longer to detect and record (although damage from natural disasters is likely to be publicized and recorded fairly quickly). During the course of this project, MapQuest announced that it has decided to partner with OSM and has begun using OSM data for routing in parts of Europe. In the U.S., MapQuest has started using OSM to display background information but plans to rely on other sources until OSM is complete and current enough to meet its needs. The company plans to support efforts to increase the number of OSM contributors in the U.S. In addition, the founder of OSM recently began working for Microsoft, and soon after, Microsoft made its Bing imagery available for tracing by OSM contributors. Hence, there is increasing interest in OSM among the major providers of Internet trip planning services.

Trip planning for other modes (transit, walking, and bicycling) also requires data on the street network. Bus transit, streetcars, and bicycles operate on the same streets as cars, but sometimes in dedicated lanes, sometimes in general-purpose lanes, and sometimes, in the case of bicycles, on wide shoulders of roads. Most sidewalks are built along streets, and trip planners use street names for reference (e.g., "turn left at 22nd Street; walk three blocks and turn left at Howard Street"). Each of these modes may include some networks that are separate from the street network, but access to these separate networks is commonly made from the street (or parallel sidewalk) network. However, although most of the sidewalk network follows streets closely, sidewalks do not extend everywhere the street network does, and they may be available only on one side of some streets. In

addition, multi-use bicycle/pedestrian trails, rail transit, and sidewalks through parks or university campuses illustrate how sidewalks can extend beyond the street network. Although restrictions on turning, access, and direction may be less common for these modes than for cars, they do occur and need to be recorded in the network data. Design standards for bicycle lanes have changed during the past few decades, so a new lane and an old lane may have different features. In the best case, data would describe differences that could reasonably be expected to affect safety or perceptions of the lanes.

The data for different networks need to record where a traveler can switch from one mode of transportation to another. For transit, this means the locations of stops and stations, and their location on the street and sidewalk networks (and, where applicable, multi-use trails). For sidewalks, this also means where sidewalks cross streets.

In addition, a trip planner for transit also needs data on when transit vehicles depart any given station or stop, and the path which the vehicles take. GTFS files were developed to record this data for the express purpose of supporting Google's Internet-based trip planner, and have become a de facto standard for providing this data to trip planning software.

ROUTING ALGORITHM – A PROCEDURE TO FIND PATHS THROUGH NETWORKS FROM THE ORIGIN TO THE DESTINATION

The core functionality of a trip planner is the routing engine which includes a procedure, or algorithm, to find a path between the origin and destination. In order to find the optimal path between the origin and destination, the routing software needs to be able to prioritize the vertices and edges in a transportation network based on the user's desired criteria, as well as reasonable defaults for criteria that the user does not provide.

In a typical network, an edge (i.e., link) represents a connection between two nodes. A node, or vertex, typically represents a location where multiple edges are joined, such as an intersection (i.e., the node) joining multiple roads (i.e., the edges). A routing algorithm must efficiently search for the shortest path between the start and end vertices. Figure 5 illustrates the nodes and edges within a graph, and a shortest path between a source and destination node.

The routing algorithm becomes more complicated for planning transit routes, because routes must be dependent on when the trip starts or ends (i.e., the transit schedule). Typically, routes on pedestrian and car networks are not constrained by time as most roads and sidewalks tend to be usable at all hours of the day.

Various routing algorithms have evolved from the area of graph and network theory in Computer Science. Perhaps the most famous routing algorithm is Dijkstra's Shortest Path algorithm, which takes a network with nonnegative weights assigned to each edge (e.g., distance between traffic intersections on a map), a source and destination point, and identifies a path that is guaranteed to be the shortest journey from the source to the destination [14]. Dijkstra's algorithm starts at the source, and then keeps track of the total distance from the source to each node in the graph and only keeps the shortest path to each node. By only expanding its search from a node N that is the

shortest path from the source node to node N, Dijkstra's algorithm guarantees that it will find the shortest path from the source to the destination.

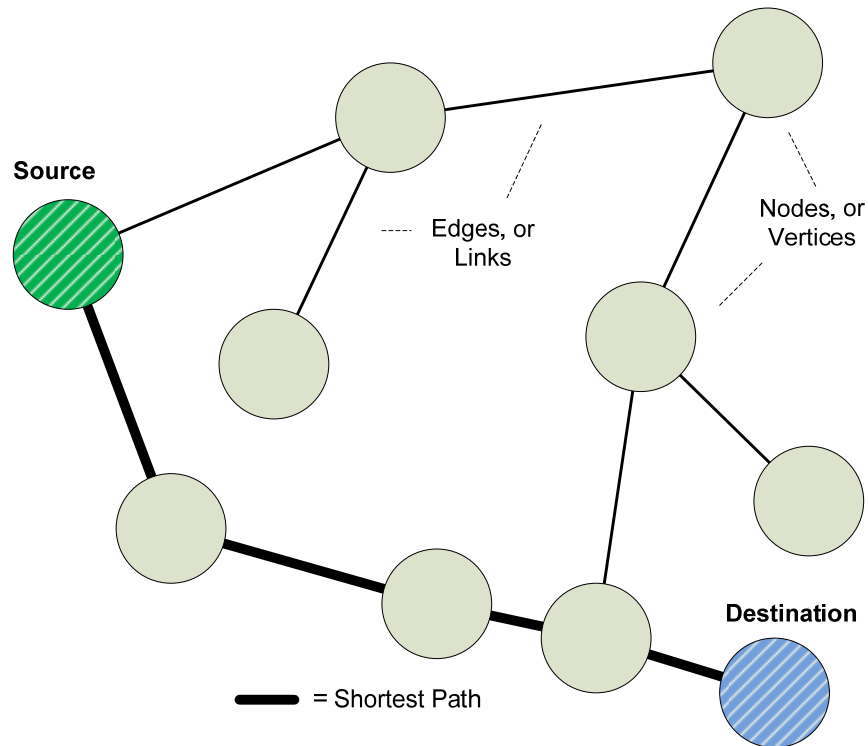


Figure 5 – A graph used by routing algorithms consists of nodes (vertices) joined by edges (ways or links)

The A* routing algorithm, pronounced “A star”, is derived from Dijkstra’s and is popular because it can improve search time. A* uses an estimated cost (i.e., heuristic) from the current node to the destination in order to bias its search towards the destination node [15]. A* is also guaranteed to find the shortest path to the destination, given that the heuristic is an underestimate of the actual cost from the current node to the destination. An example heuristic is the straight-line distance from the current location to the destination when traveling on roads within a city. By choosing nodes that have successively smaller straight-line distances to the destination, the A* algorithm typically finds the shortest path more quickly than algorithms that search all possible paths

INCLUDING USER PREFERENCES IN THE ROUTING ALGORITHM

A routing algorithm will find a path using the network data that are provided. Most trip planners provide some options to let a user restrict the data to be used when planning a route (e.g., do not use toll roads). In addition, most trip planners allow a user to choose some aspects of a trip that the routing algorithm will emphasize (e.g., plan the trip with the shortest distance).

CAR TRIP PLANNERS

Internet-based planners for car trips generally allow the user to choose whether or not to allow a trip to use toll roads, ferries, or freeways. In the past few years, online trip planners have also begun to allow users to request a route and then drag portions of the route to use or avoid different streets. This allows the user to take advantage of local knowledge (such as road closures), or to detour around areas expected to have heavy traffic congestion. Some also allow a choice between the shortest distance and the shortest time, or an option that automatically finds a trip to visit several destinations in sequence.

PUBLIC TRANSPORTATION TRIP PLANNERS

Trip planners for modes of transportation other than driving tend to allow a broader array of options for the user to guide the planner toward the trip that best meets his or her preferences and needs. Varied options are useful for a couple of reasons. First, compared to the street network for car travel, the networks for bus, bicycle, or foot travel tend to be less dense, less well connected, and less uniform. Where the question for a car-trip planner is often “Show me the quickest route”, the question for other modes may first be “Can I get there by this mode?” followed by “How bad (e.g., slow, hazardous) is the trip likely to be?” Second, users of modes other than car travel may perceive more differences in the quality of possible trips by these modes.

Trip planners for public transportation allow users to choose when they would like to depart the origin, or when they need to arrive at their destination. This is a necessary option because transit schedules often vary depending on the time of the day or day of the week. Transit trip planners also can automatically provide several possible trip plans to choose from. In addition, they can allow selection of the shortest travel time on transit, or the shortest walking distance to transit, or the trip that uses the fewest transfers. On transit systems where the fare may vary by route or type of service (local or express), the planner could also plan the trip to minimize the fare. When a transit system has rail, bus, and streetcar service available, the trip planner could let the user specify a preferred mode of travel (as Google Transit does). When a trip uses services from more than one transit system, a trip planner should allow transfers between agencies and can allow the user to choose which agency to begin the trip with. Transit trip planning also could be guided by whether the transit stops are accessible to persons who use wheelchairs, or whether the stops have features such as benches, shelter, or shade. Each of these options allows the trip planner to create a trip that best meets the user’s needs.

BICYCLE TRIP PLANNERS

Bicycle trip planners remain relatively uncommon, but the few that have been developed recognize that different users have different tolerances for safety and preferences for their rides. It is unclear exactly how Google’s bicycle trip planner chooses routes, although descriptions on blogs indicate that it makes some use of data on terrain, bicycle facilities, and feedback that users of the trip planner provide about the quality of the route. Google does not provide options for the user to guide the bike routing algorithm. Atlanta’s A-Train planner [16] allows users to indicate how much emphasis to give to the length of the bicycle trip, to how much of the trip uses facilities designated for bicycles, and to the amount of hill-climbing the route requires. Broward County’s BiKE Broward

planner [17] allows the user to request the shortest route, the fastest route, the simplest route (fewest turns), the route that minimizes interaction with car traffic, or the most scenic route. Chicago's Goroo planner [2] can plan trips by bicycle, but it does not give the user options to guide how it develops the route. For trips that combine bicycling and transit, Goroo does allow the user to choose whether to bicycle the whole trip, take the bicycle on transit, or leave the bicycle at the transit stop. Goroo guides its trip planning using data on which roads cyclists have reported using. These may not be representative of people who do not ride but would like to if better infrastructure were available.

Bicycle trip planners could also use bicycle level of service (BLOS) [18] or similar measures of network acceptability for cycling, or they could use data on frequency of bicycle crashes (adjusted for bicycle activity) to find routes that avoid streets known to be particularly dangerous for cycling. To the research team's knowledge, no bicycle trip planner does so. Another possibly helpful distinction, which does not seem to be in use yet by any bicycle trip planner, is whether the route is paved (and usable by all bicycles) or unpaved (and usable only by off-road bicycles). In addition, cyclists who are uncomfortable riding along or crossing major streets may want to know which intersections have traffic signals to make crossing easier.

PEDESTRIAN TRIP PLANNERS

Pedestrian trip planners also are relatively uncommon. Google's walking trip planner does not provide options to allow the user to guide the algorithm (e.g., to avoid a busy street). For trips involving walking to transit, Goroo allows the option of minimizing the walking distance. TerpNav, a pedestrian trip planner for the University of Maryland's campus, allows the user to request a walking trip and to choose whether to allow the use of stairs or steep slopes, or to require the use of curb cuts. These features allow persons who use wheelchairs to find accessible routes. Walk trip planners also could use pedestrian level of service (PLOS) [19], similar measures of network acceptability, or the presence of shade or street lighting for walking, to favor routes considered more pleasant. This may require data on street crime or the frequency of crashes involving pedestrians (adjusted for activity) to avoid streets likely to be dangerous for walking. Data on unpaved shortcuts, which have been shown to encourage walking and cycling between suburban multifamily housing and adjacent commercial centers when fences and other barriers are absent, may also be desired [20]. As with bicycle trip planners, the research team does not know of any walk trip planner that does so. Data indicating which intersections have traffic signals can be used to guide pedestrians toward such intersections.

MULTIMODAL TRIP PLANNERS

Multimodal trip planning can use any of the above methods to guide route choice, although at this time each uses a different subset of these methods. What distinguishes multimodal trip planners from single-mode trip planners are options to allow the user to combine modes in a single trip (such as bicycling to transit, riding transit, and walking or cycling from transit to the final destination), and options that allow the user to emphasize one mode or another in a trip that uses more than one mode. Thus, a planner that combines walking and transit, or bicycling and transit, or driving to a park-and-ride lot to take transit, can allow the user to minimize the amount of walking,

cycling, or driving to reach transit, or the amount for the total trip. As demonstrated by Goroo, a planner that combines bicycling and transit can allow the user to specify whether the bicycle is to be left at the transit stop (which requires bicycle parking) or is to be taken on the transit vehicle (which requires that the vehicle and policies enable this). At present, the availability of bike-on-bus service in transit systems depends on the transit agency. However, bicycle racks are nearly universal on public transportation buses in Florida, and probably will become available on all such buses by 2015 [21].

Options that allow the user to guide the selection of a route can improve the trip planner's usefulness, but they come with costs. First, they require changes to the trip planner interface to let the user choose them. Even when there are default settings (e.g., to use PLOS to choose routes), if there is no way for a user to change the default (e.g., to request the shortest trip), then the user has no way to indicate their preferences to the algorithm. However, the more options that are offered, the more work the user may need to perform to request a route. This burden can be reduced by enabling the trip planner to store data about the user's preferences, and use them whenever the user plans a trip. Second, most of the options require additional data. To find a shady walking route, the trip planner will need data on which sidewalks have shade and which do not. To give preference in bicycle trips to routes that use bicycle facilities, or to routes that minimize the need to climb hills, the planner needs data on facilities or elevation.

PRESENTING THE RESULTS TO THE USER

Human beings tend to follow two different approaches to navigate a route. One has been termed cognitive, or spatial, navigation. This method involves looking at a map and using information on the map to figure out where one is located, and where to go. The other has been termed narrative, or stimulus-response, navigation. This method involves following a set of directions to get from the origin to the destination ("Turn right. Continue to Cypress Street. Turn left on Cypress Street.") Most people are able to use both but have preferences for one or the other. Most Internet-based trip planners support both navigation strategies, by providing both a map showing the route on a background of streets and points of interest, and a step-by-step set of directions for moving from the origin to the destination. Much of the information presented on the map and in the directions is derived from the data used to find the route and guide the routing algorithm.

Navigation directions can be as simple as a listing of the street names along which travel occurs, with instructions on when to turn or change streets, or when the street name changes. These directions may be accompanied by information on compass direction, distance of travel or estimated travel time until the next step in the directions, or by notes that describe conditions of the route or points of interest along it. This supplementary information can be helpful if street names and addresses are difficult to read along the way, if street names or addresses are missing, or if the street names in the directions do not match those on the street. To date, there seem to be few good examples of providing step-by-step directions for walking along sidewalks that do not closely follow streets (as in parks or on college campuses). Sidewalks in these situations may not have names that can be used to refer to them, and someone new to the area might not understand how to proceed when directed to walk toward the Administration Building. A walking trip through part of the USF campus could easily traverse several dozen short sections of sidewalk, none of

which has names (Figure 6). More research is needed on how to support narrative navigation in these situations.

Similarly, design standards for bicycle lanes have changed over time. Some lanes were not built to the standards prevailing at the time they were built. Sometimes lanes are retrofitted onto an older street in stages as funds become available instead of along its complete length. Any of these situations can create breaks or other unexpected conditions in a bicycle lane, and a hazard for a bicyclist who is using the lane for the first time. Where data describe these situations, the trip planner should make this information available in the directions for a bicycle trip.

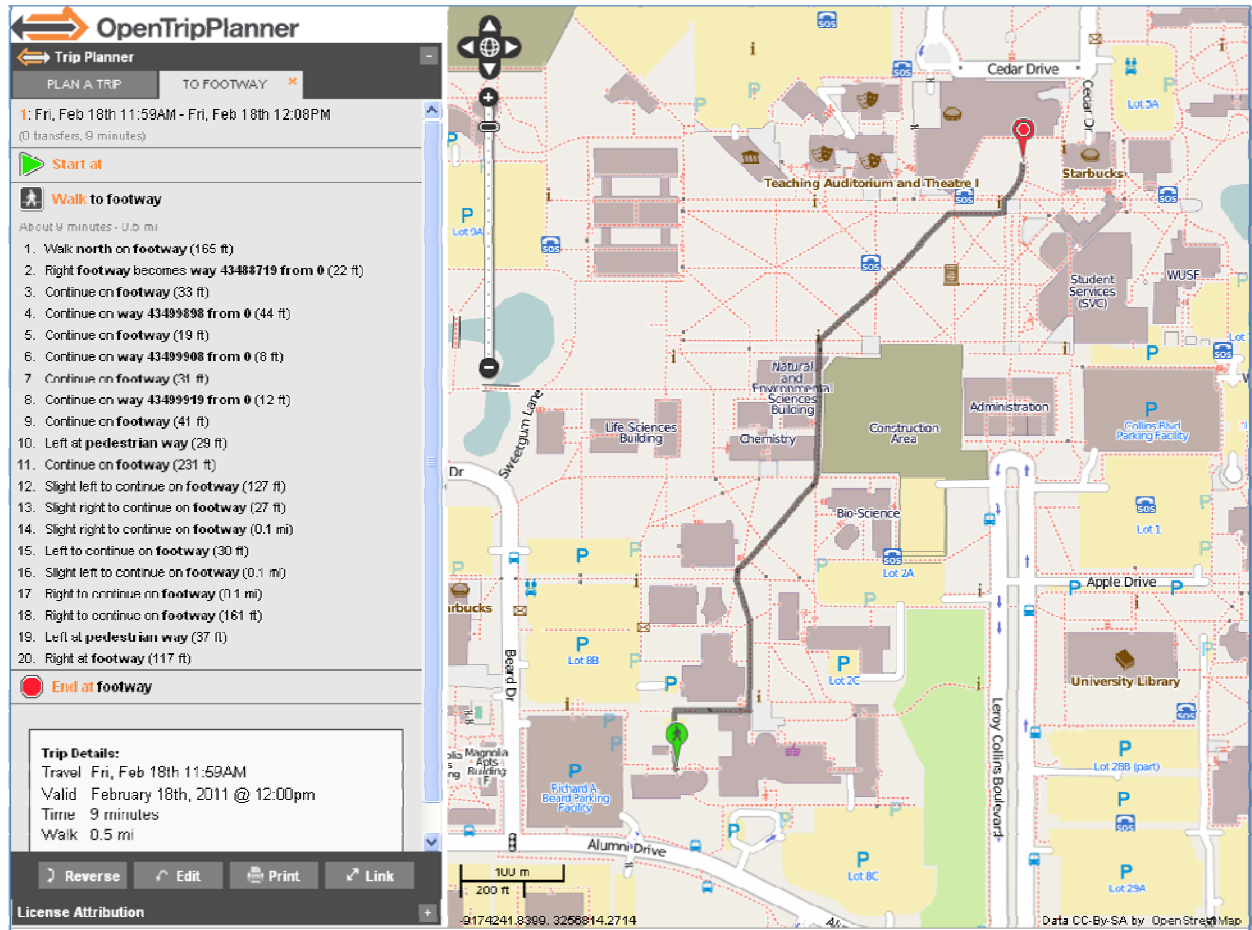


Figure 6 – Route through University of South Florida campus, and associated walking directions.

Producing a map for spatial navigation requires similar data to directions, but it also needs data that can be displayed to show points of interest, major landmarks, and other information that the map reader can use to interpret the map and verify progress in moving toward the destination. When the planner allows users to indicate their preferences to the routing algorithm, the map also can display the information used to find the route (e.g., volumes of car traffic, or presence of bicycle lanes). This is useful for helping users understand how a particular route was developed, and users may adjust their preferences once they are aware of how their preferences affect which route is found. This may also help users develop more confidence in the reliability of the trip planner.

However, display of this sort of supplemental preference information is not required for showing the route, and it should be optional to make the map easier to read.

On the other hand, conditions that create safety hazards for the traveler need to be effectively communicated. For example, a bicycle lane may have a gap where it traverses a bridge. Where the network data contains information that describes the hazard, a navigation map might highlight potential hazard areas and refer the user to the navigation directions for detail.

Any information that is displayed on the map needs to be presented in ways that do not burden the user's ability to interpret the map. Displaying too much information on the map, or using complex symbols instead of simple ones, or using symbols that are inconsistent with common practice, or using colors that conflict with common interpretations or subconscious interpretations, can make the map difficult to read and use for navigation. As an example, a line printed in red draws more attention than an otherwise identical line drawn in another color. Using green to show safe streets for bicycling, yellow to show riskier streets, and red to show unsafe streets is consistent with symbolization used in traffic signals. Figure 7, from the Broward County bicycle trip planner, illustrates such a color scheme for interaction with different levels of motor vehicle traffic; paper or downloadable bicycle maps for many metropolitan areas use a similar green-to-yellow-to-red scheme. However, a person who views a map is likely to place greater emphasis, subconsciously, on the red unsafe streets, so that instead of showing where to go, the map shows where **not** to go. One way to counter this conflict between cultural and psychological effects of color is to draw the red unsafe streets using narrower lines than the less risky streets. This allows the red to retain its

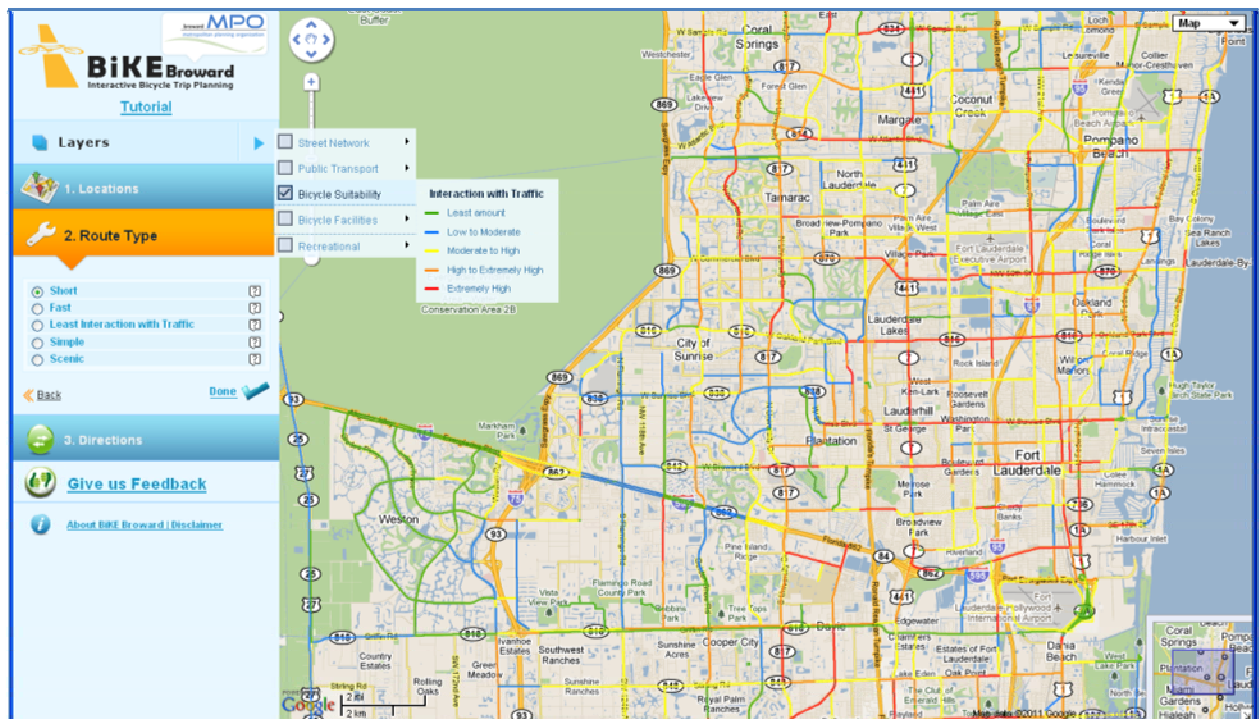


Figure 7 – A green-to-red gradation of colors in an area with few green and many red streets gives greater emphasis to red streets (less desirable) and makes it difficult to discern green ones (more desirable).

meanings without dominating the interpretation of the map. Another might be to use gradations of another color, perhaps from dark blue to lighter shades of blue, to represent different levels of interaction with motor vehicle traffic. The design of Internet-based maps to enhance their readability for navigation is still evolving and needs additional research.

Internet links to photos or other information can provide additional information without burdening the trip planner software or requiring maintenance of detailed data that are not part of the core trip planning function.

Two final requirements of a trip planner are to:

- enable the user to take a copy of the planned trip to use along the way, and
- allow the user to provide feedback about the route, especially if something is wrong.

The first of these is accomplished for Internet-based maps by printing the browser window, although the major car-trip planning websites will reformat the map and directions for printing if desired. Smartphone-based services could store the trip plan and map, and use the phone's GPS to display the traveler's location on the map with the route, or give audible and visual directions as the user travels along the route.

Allowing feedback on a route allows users to report errors that may indicate a need to correct or update data. Even if the trip planner uses OpenStreetMap as a data source, few users of the trip planner may be adding or editing data in OSM, and even OSM contributors may not know how often the trip planner retrieves updated data from OSM. In addition, user feedback may provide information on how to better guide the routing algorithm. The tool for providing feedback should allow the user to easily identify where the problem exists, either by choosing a step in the list of directions, or by clicking on a portion of the route.

OVERVIEW OF DATA REQUIREMENTS

The preceding section has identified different types of data needed by trip planners. Table 1 lists the types of data identified in the preceding discussion of functional requirements. The data requirements can be fairly simple for a single mode. However, as additional transportation modes and options to allow users to guide the trip planner to meet their needs are included, data requirements can increase significantly in complexity. Table 1 shows if required data elements can be represented in the GTFS format for transit data, or in OSM using commonly-accepted coding conventions. However, it should be noted that the ability to represent data in GTFS or OSM does not imply that this data currently exists for all areas. Most of Florida's public transportation agencies have not yet put their data into GTFS format. OSM coverage for many data attributes in most U.S. cities will remain incomplete until additional people or organizations in these cities contribute data to OSM. Data can also be acquired from other existing sources (e.g., departments of transportation datasets, city and county land parcel data, etc.) although the owner of the data and completeness of coverage will vary between cities based on the existing demand for geographic data in public services.

Table 1 - Data requirements for ideal multimodal trip planners.

	ROLE IN TRIP PLANNER				SOURCE	
	Define origin and destination	Describe network	User preferences for routing	Present results	In GTFS	In OSM
Street network	Required	Required		Required	Indirectly, only if bus route is on a street	Yes
Points of interest, landmarks	Required for point-and-click			Required for map output	No, but proposal exists [22]	Yes
Address range location file	Required for address input				No	Yes
Names, and locations or addresses, for businesses and points of interest	Required for named input				For transit stops only	Yes
User's location	Optional for smartphone input			Optional for smartphone output	No	No
Data for other modal networks (e.g., transit, biking), including links between modes		Required for routing transit, bike, walk modes	Attributes of links (e.g., paving steps) allow preference		Yes, GTFS serves as primary transit data source	Yes, some transit, bike and pedestrian data. Needs some improvements
Location of traffic signals			Allows preference in bike, walk modes	Assists in interpreting results for bike/walk	No	Yes
Restrictions on turning, access, direction		Required for all modal networks			Yes, as part of a trip/route	Yes
Transit schedules		Required when transit is included	Allows preference for fastest route using transit	Assists in interpreting results for this mode	Yes	No

	ROLE IN TRIP PLANNER				SOURCE	
	Define origin and destination	Describe network	User preferences for routing	Present results	In GTFS	In OSM
Road speed limits			Allows preference for fastest route with car-based modes, preference for low-speed roads for bike and walk modes	Assists in interpreting results for these modes	No	Yes
Tolls, ferries, road classifications			Allows preference, when modes can use them	Assists in interpreting results for these modes	No, except for ferries	Yes but ferries incomplete
Transit fares			Allows preference for least costly route	Assists in interpreting results for this mode	Yes	No
Type of transit vehicle			Allows preference	Assists in interpreting results for this mode	Yes	Yes
Ability of transit vehicles to carry bicycles			Required for bike-to-transit routing	Assists in interpreting results for these modes	No, but proposal exists [22]	No
Wheelchair access			Required for transit, bike modes	Assists in interpreting results	No, but proposal exists [22]	Yes
Amenities (e.g., shade, bench, shelter, bike parking, street lights)			Allows preference in transit, bike, walk modes	Assists in interpreting results for these modes	No, but proposal/unofficial extension of GTFS may include amenities [22][23]	Yes

	ROLE IN TRIP PLANNER				SOURCE	
	Define origin and destination	Describe network	User preferences for routing	Present results	In GTFS	In OSM
Digital elevation file			Allows preference in bike, walk modes	Assists in interpreting results for these modes	No	No
Suitability of network links for bicycling, walking			Allows preference in bike, walk modes	Assists in interpreting results for these modes	No	No
Variables to calculate suitability or level of service			Allows preference in bike, walk modes	Assists in interpreting results for these modes	Some for transit	Some (e.g., speed, number of lanes, presence of signals)
Crash or other safety history			Allows preference in bike, walk modes	Assists in interpreting results for these modes	No	No
Informal routes and shortcuts			Allows use in walk, bike modes	Assists in interpreting results for these modes	No	Yes
Internet links				Assists in interpreting results or preparing for the trip	Yes, for agency, stop, and route	Yes

Note: "Allows preference" also allows avoidance of routes based on the preference.

The following sections discuss how data formatted according to the GTFS and OSM specifications may be used to meet these data requirements, and options for meeting data requirements that fall outside the scope of these two sources.

GENERAL TRANSIT FEED SPECIFICATION DATA

As mentioned earlier, the GTFS format was originally created to serve as a user-friendly representation of transit information needed to operate an automated transit trip planner. Over 125 U.S. transit agencies have placed their transit data in the GTFS format, which has resulted in the evolution of GTFS to provide the needed fields and attributes to accurately represent transit service

for many different types of transit agencies. There is an open list-serve and website, “General Transit Feed Specification Changes” where any registered user (e.g., developer, transit agency, etc.) can propose changes to the specification [22]. If there is consensus among members that the change would be useful, then the change is included as part of the specification.

Since the transit agencies must also update their GTFS data each time a schedule or route changes, and the operator of the trip planner must also consume this update information and integrate it into the trip planner, GTFS was also designed to be a low-maintenance solution for both the operator of the trip planner as well as the transit agency. In fact, many tools currently exist to aid in the generation, processing, and consumption of GTFS information, including the ability to export GTFS data from two major transit scheduling software vendors, GIRO’s HASTUS and Trapeze [24].

GTFS is used by arguably two of the biggest nation-wide providers of public transportation trip planning services, Google Maps Transit and Microsoft’s Bing Maps. Other transit services, such as the OneBusAway transit mobile app and system for Seattle, WA, also uses GTFS as a data source. A result of these nationwide services and participation in GTFS by over 125 U.S. agencies, the GTFS format has been heavily vetted as a data source for public transportation trip planning. However, since GTFS was designed to serve transit-only trip planners, it does not necessarily meet all the needs of a multimodal trip planner that is capable of providing linked trips using public transportation, bike, and walking. The following sections discuss the GTFS format as it relates to the data requirements defined in Table 1.

STREET NETWORK

Street network data is not explicitly included in a GTFS dataset. However, there is a file *shapes.txt* which contains a series of points, defined by latitude and longitude, which illustrates the actual path of a transit vehicle as it travels along a route. The shape of the route is important for the visual representation of the transit trip on a map as it is shown to a user. Since a line which travels from one bus stop to another (i.e., a connect-the-dots of bus stops) does not necessarily represent the exact path of travel, especially if the road between two bus stops is curved, the shape information is needed to visualize the path of the route. Assuming that a bus travels on the road, the *shapes.txt* data may be used to provide an estimated representation of a partial road network. However, there are several caveats that must be acknowledged. First, the actual travel path of the bus does not necessarily represent the center line of the road. For example, when a bus pulls into a bus bay on the side of the road, this does not necessarily represent a shift in the road direction. Secondly, there are errors in the creation of any geographic dataset. Therefore, a representation of a road network from existing GIS files will never exactly match another dataset, such as a GTFS file. Therefore, the *shapes.txt* file should be considered an estimation of the location of a road, and not an exact path. For these reasons, it is preferable to use a data source that is designed to handle street data, such as OSM, for the representation of a street network in a multimodal trip planner.

POINTS OF INTEREST AND LANDMARKS

GTFS is not currently designed to contain any information about points of interest or landmarks that are not transit stops. There was a proposal to add landmark information to GTFS in May 2007

[22], but it was never adopted. In the research team’s opinion, landmark and points of interest information for multimodal trip planners, especially OTP, are best represented in OpenStreetMap. Please see the corresponding section for data requirements and OSM on page 36 for more information.

ADDRESS RANGE AND LOCATION FILE

GTFS does not contain any address range and location file information. Please see the corresponding section for data requirements and OSM on page 37.

LISTS OF BUSINESSES AND POINTS OF INTEREST, WITH LOCATIONS OR ADDRESSES

GTFS may inadvertently contain some information about nearby businesses or other points of interest with geographic location information, if the transit agency has defined a transit stop and named it after the business or point of interest. This information would be contained in the *stops.txt* file *stop_name* and *stop_description* fields, which contain the name and description of each transit stop along with the latitude and longitude of each stop. For example, Pinellas Suncoast Transit Authority (PSTA) has created a bus stop where the name is “TYRONE SQUARE MALL SHELTER – E.” If there is no recognizable landmark near the stop, the stop may be named after a nearby intersection, such as “22ND AVE N & OXFORD ST N.” There is an optional *location_type* field in *stops.txt* that defines whether the stop is a “transit station,” (0 or blank = stop, 1 = station) which is defined as a physical structure or area that contains one or more stops. However, even though this information appears in GTFS data, there are no indicators that these transit stops near points of interest are any different from transit stops that don’t have nearby points of interest. Therefore, OSM should be considered the primary repository of any information about businesses or points of interest.

OTHER MODAL NETWORKS: TRANSIT

GTFS serves as a definitive representation of transit data for creating a transit network to use in planning public transportation trips. As mentioned earlier, GTFS has been heavily used by many Web and mobile applications and transit agencies for the purpose of transit trip planning. Various text files formatted with commas are contained within a zip file for a GTFS dataset. It should be noted that some fields in GTFS are required to be populated by a transit agency, and others are optional. The required fields are necessary for the basic operation of a transit trip planner, while the optional fields can be used to enhance the trip planning process as well as the presentation of the information to the user of the trip planner.

The core required files in a GTFS dataset are *agency.txt* for basic contact information about the agency, *stops.txt* for transit stop information, *routes.txt* for route information, *calendar.txt* for a basic transit schedule, and *trips.txt* and *stop_times.txt* to define an ordered visitation of stops on a particular route at a particular time according to a schedule. Optional files include *calendar_dates.txt* to define exceptions to regular transit service schedules, *fare_attributes.txt* and *fare_rules.txt* to define the type and amount of transit fares, *shapes.txt* to represent the actual travel path of the transit vehicle for visualization on a map, *frequencies.txt* for transit systems that have a

fixed headway time between departures from the same stop instead of a schedule, and *transfers.txt* to define precise rules for making connections between transit routes. A description of some of the required GTFS files is below.

The *stops.txt* file contains information about all transit stops in the system, including the name, description, and latitude and longitude.

Example transit stop data:

```
stop_id, stop_name, stop_desc, stop_lat, stop_lon, zone_id, stop_url
1, SW 114 AV@QUAIL ROOST DR, SW 107TH AVE & SW 112TH ST, 25.592446, -80.377954, ,
```

The *routes.txt* file contains records for all routes in the transit system, including route numeric IDs, a description of the route and a route type. GTFS is also capable of representing many different types of public transportation, which is discussed in detail in a following section.

Example route data:

```
route_id, route_short_name, route_long_name, route_type, route_url, route_color
1, 1, Florida Avenue, 3, http://www.gohart.org/routes/hartline/1.html, FF9900
```

Trips in *trips.txt*, and the corresponding file *stop_times.txt*, represent the order of visitation of a series of transit stops on a particular route at a particular time (defined by a schedule, and arrival and departure time from the stop. Direction information is defined by a 0 for travel in one direction (e.g., outbound), and 1 for travel in the opposite direction (e.g., inbound) in the *trips.txt* file.

Example trip data:

```
route_id, service_id, trip_id, trip_headsign, direction_id, block_id, shape_id
9496, 1, 2094524, 1 - PERRINE, 0, 1056922, 90091
```

Example stop times data:

```
trip_id, arrival_time, departure_time, stop_id, stop_sequence, stop_headsign, pickup_type, drop_off_type, shape_dist_traveled
2094513, 6:35:00, 6:35:00, 1, 1, , 0, 0,
2094513, 6:35:28, 6:35:28, 3614, 2, , 0, 0, 0.1549
2094513, 6:36:04, 6:36:04, 3213, 3, , 0, 0, 0.3540
2094513, 6:36:38, 6:36:38, 3214, 4, , 0, 0, 0.5445
2094513, 6:38:05, 6:38:05, 3521, 5, , 0, 0, 1.0269
```

The representation of transit schedules in GTFS, as well as the use of URLs and fares are discussed in further detail in a following section.

It should be noted that the accuracy of bus stop locations in a GTFS dataset may affect the performance of GTFS-based applications, depending on the required accuracy of each application. For example, the Travel Assistance Device (TAD) project team has indicated that errors in GTFS bus stop locations can cause accuracy problems when triggering the TAD GPS-enabled cell phone

application to prompt a rider in real-time to exit the transit vehicle [9]. However, for applications that don't require a precise bus stop location, such as the Google Maps website that shows a broad overhead view of the planned transit trip, the accuracy of the service may only be affected by significant errors.

OTHER MODAL NETWORKS: BICYCLING AND WALKING

GTFS does not include any information about bicycling or walking. Please see the corresponding sections for data requirements and OSM on page 37 and page 39 .

RESTRICTIONS ON TURNING, ACCESS, DIRECTION

GTFS provides formatting for the information needed to create a transit network for routing. As discussed in the *Other Modal Networks: Transit* section above, GTFS provides a format for recording direction information for routes as inbound (1) and outbound (0) as part of trips. *Trip.txt* and *stop_times.txt* define an ordered visitation of bus stops at a specific time according to a particular schedule, thereby encoding rules from restricting turning and direction of a public transit vehicle to the data provided in these tables.

LOCATION OF TRAFFIC SIGNALS

GTFS does not include any information about traffic signals. Please see the corresponding section for data requirements and OSM on page 39.

TRANSIT SCHEDULES

GTFS provides an excellent format for recording transit schedules which can be used to plan transit trips. The GTFS file *calendar.txt* defines when transit service is available (i.e., the transit schedule) by date and day of the week. Another file, *calendar_dates.txt*, can be used to hold exceptions to the typical schedule, for example to hold a special schedule for holiday hours of operation.

Example calendar data for weekdays (i.e., Monday-Thursday), Friday, Saturday, and Sunday from March 29, 2009 to November 7, 2009:

service_id	monday	tuesday	wednesday	thursday	friday	saturday	sunday	start_date	end_date
We,	1,	1,	1,	1,	1,	0,	0,	20090329	20091107
Fr,	0,	0,	0,	0,	1,	0,	0,	20090329	20091107
Sa,	0,	0,	0,	0,	0,	1,	0,	20090329	20091107
Su,	0,	0,	0,	0,	0,	0,	1,	20090329	20091107

POSTED SPEED LIMITS

GTFS does not include any information about posted speed limits. Please see the corresponding section for data requirements and OSM on page 40.

TOLLS, FERRIES, AND ROAD CLASSIFICATIONS

GTFS is able to represent basic ferry information as a form of public transportation, which is discussed in the *Type of Transit Vehicle* section below. GTFS does not provide any toll or road classifications, so please see the corresponding section for data requirements and OSM on page 40.

TRANSIT FARES

Transit fare information is vital to allow the user of a transit trip planner to calculate the cost of a trip. Fare information can also be used to help the user choose mode of transit that is the least expensive. GTFS provides two optional files which can be used to define fare information that is sufficient for transit trip planning. *Fare_attributes.txt* specifies a type and amount of currency which must be used, as well as whether the fare needs to be paid on board or prior to boarding. The number of transfers permitted on a fare is also included. *Fare_rules.txt* specifies how fares in *fare_attributes.txt* apply to a specific route or zones of travel.

TYPE OF TRANSIT VEHICLE

GTFS provides formatting which can define the type of public transportation used for each route, in the field *route_type* for records in the *routes.txt* file, including:

- 0 - Tram, Streetcar, Light rail. Any light rail or street level system within a metropolitan area.*
- 1 - Subway, Metro. Any underground rail system within a metropolitan area.*
- 2 - Rail. Used for intercity or long-distance travel.*
- 3 - Bus. Used for short- and long-distance bus routes.*
- 4 - Ferry. Used for short- and long-distance boat service.*
- 5 - Cable car. Used for street-level cable cars where the cable runs beneath the car.*
- 6 - Gondola, Suspended cable car. Typically used for aerial cable cars where the car is suspended from the cable.*
- 7 - Funicular. Any rail system designed for steep inclines.*

ABILITY OF TRANSIT VEHICLE TO CARRY BICYCLES

A proposal to consider addition of bicycle accessibility data, or the ability to carry bicycles on buses, to GTFS was proposed in early 2008 [22]. This proposal would add additional fields in the *routes.txt* and *trips.txt* files to define whether or not transit vehicles on those routes and trips were capable of carrying bikes. However, this proposal has not yet been adopted into the GTFS specification. In the research team's opinion, this data is best coded using GTFS data, not OSM data, and therefore the team recommends that the bicycle proposal for GTFS be revisited.

WHEELCHAIR ACCESS

A proposal to consider the addition of wheelchair accessibility information to GTFS was discussed on the GTFS Changes site in early 2010 [22]. However, as of the date of this report the proposal has not been adopted into the GTFS specification. OSM provides extensive capabilities to document accessibility of pedestrian infrastructure, so please see the corresponding section for data requirements and OSM on page 41.

AMENITIES

GTFS currently does not provide any format for coding amenities that are near a transit stop, although there are currently efforts underway to examine how GTFS may be extended to include such data as both an official GTFS addition [22] and an unofficial extension [23]. Since GTFS does not currently provide coding for amenities, OSM data should be used to represent this type of information. Please see the corresponding section for data requirements and OSM on page 41.

INTERNET LINKS

Internet links can also optionally be included in the GTFS data that may link back to the transit agency's website so that users can see more detailed information about a transit route or stop, possibly including multimedia information such as pictures. GTFS defines a URL field for the following transit attributes:

- Transit agency, in *agency_url* field in *agency.txt* – Can provide a user with more general information about a transit agency, typically pointing to the main website for the agency
- Transit stops, in *stop_url* field in *stops.txt* – Can be used to show a user a webpage related to a particular transit stop or station (e.g., with pictures, a longer description, or possibly even real-time estimated arrival information for incoming vehicles)
- Transit routes, in *route_url* field in *routes.txt* – Can show a user more information about a particular route.

VALUES TO CALCULATE SUITABILITY OR LEVEL OF SERVICE

The research team reviewed the bicycle LOS (BLOS) [18] and pedestrian LOS (PLOS) [19] reports to determine what variables both LOS measures require. In addition, the research team examined a study that calculated a BLOS for intersections rather than for road segments [25]. Since GTFS data does not currently provide any bicycle or pedestrian data, it cannot be used to calculate BLOS or PLOS. Therefore, OSM and other datasets will need to be used to calculate BLOS and PLOS. Please see the corresponding section for data requirements and OSM on page 42.

INFORMAL ROUTES AND SHORTCUTS

GTFS does not include any information about informal routes or shortcuts. Please see the corresponding section for data requirements and OSM on page 47.

INTERSECTIONS AND CROSSWALKS

GTFS does not include any information about intersections or crosswalks, including tactile surfaces or pedestrian traffic signals. Please see the corresponding sections for data requirements and OSM on page 47.

OPENSTREETMAP DATA

Before describing data conventions in OSM, it is helpful to provide some background. As noted earlier, OSM was created in England in 2004, and much of its design and terminology reflects conventions that differ somewhat between Europe and the U.S. Participation in the U.S. did not become significant until late in 2007, after the U.S. Census Bureau's 2007 TIGER street network files were imported into OSM. OSM tags use English but, by convention, it uses British English [10]. From the perspective of a U.S. contributor to OSM, there appears to be a tendency among European contributors to emphasize official designations when deciding how to tag a feature, and a tendency among non-European users to emphasize functional (instead of official) designations when deciding tag values. For example, in some European countries features tagged with the *cycleway* tag have been designated as part of an official cycling network. In the U.S. there are few officially designated cycling networks. Instead, multi-use paths and bicycle lanes exist, and U.S. contributors use *cycleway* to tag these features even though the features are not officially designated for that sole purpose.

The OSM data coding system defines objects (e.g., points, lines – also called “ways”, and polygons) which are assigned a set of tags, or key-value pairs (e.g., *name=I-75*). OSM is intended to record information that is objective, meaning that if two different contributors observe the same feature on in the field, the way that each would describe it in the OSM database should be the same. For example, both should describe a bus stop as a point, tagged *highway=bus_stop*, adding *bench=yes* or *bench=no*, *shelter=yes* or *shelter=no*, and so forth, depending what features are at the stop, and which features each contributor decides to provide information about. Examples of tags for a bus stop in OSM are shown in Figure 8.

From the experience of the research team, when mapping a new feature or in a new area, it is common, and it is recommended [10], to record the most basic information on the first visit to the area. This provides structure for collecting and adding detail on subsequent visits, so some attributes of the bus stop may not be recorded by the first contributor.

OSM contributors sometimes create proposals for recording information that is more subjective than objective. For example, is a segment of road on a hill steep? Is a road safe for bicycling, or is the pavement in good condition? The OSM community has been fairly consistent in rejecting these proposals, because different contributors could legitimately assign different values to the segment of road. Instead, if a subjective value is important, the community has tended to suggest recording objective values that someone might use to assign more subjective categories outside the OSM database. Thus, recording the degree or percentage slope of a road on a hill allows someone else to decide later whether or not to call it steep. Not all subjective concepts lend themselves to this

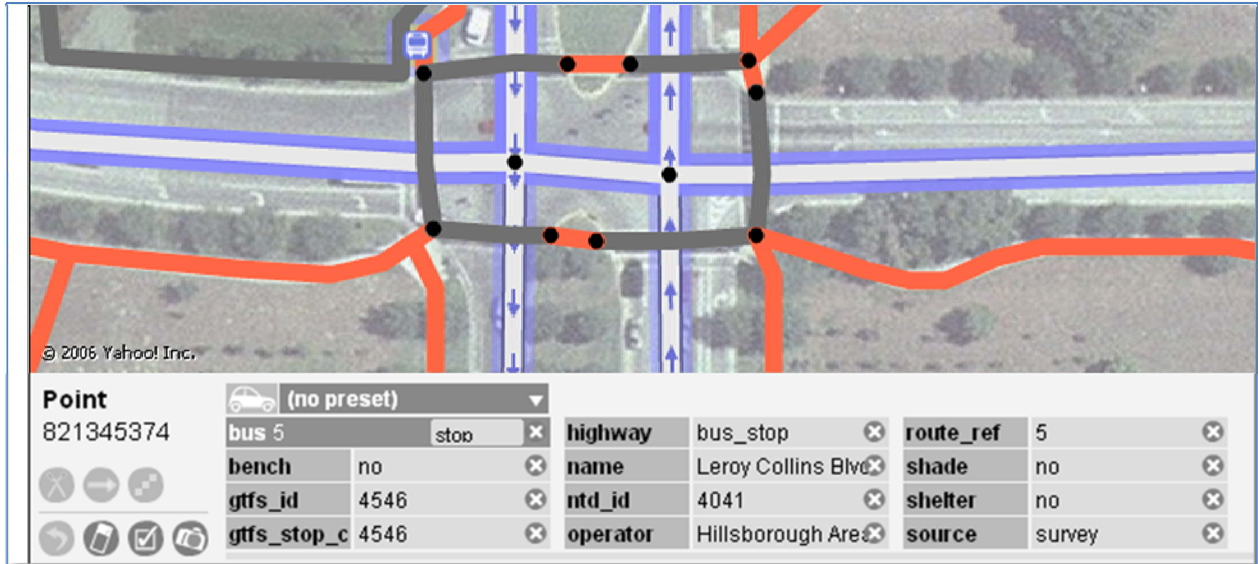


Figure 8 – OpenStreetMap uses key-value pairs to tag objects with attributes, such as the tags shown here for a bus stop

approach of classification by objective measures, but using subjective tags is still universally discouraged.

OSM is also intended to record data about physical features, rather than levels of activity. For example, a bike shop can be recorded in OSM as a bike shop, with a location, owner, address, and the services it makes available. But there is no provision for recording the number of employees, the volume of sales, or the assessed value of the property. Similarly, for roads, there is no provision in OSM for recording the location or frequency of vehicle crashes. If this kind of information needs to be used with OSM data, then the OSM data needs to be exported and the crash data combined and used outside the OSM database. This practice is typical of geographic databases, which provide a spatial framework which can be used to link or structure other kinds of data (such as Census data).

Except when recording the common name of an object (*"Joe's Diner"*, *"East 17th Street"*, *"Everglades National Park"*), all tag names and values in OSM are lower case. Tag names do not contain spaces, and tag values generally do not, except:

- within the name of an object (*"Joe's Diner"*);
- within the content of notes recorded to clarify the description of an object in the database (*"note=sidewalk confirmed to end at this point"*);
- within the content of *"FIXME"* placeholder tags, which are recorded to indicate object properties that need to be verified or corrected; and
- between numeric values and non-International System of Units (SI) measurement units (*"maxspeed=25 mph"*)

Tag names and values may use an underscore or a colon to link two words within a tag name, as in *highway=primary_link*, or *cycleway:left=lane*. As an international project, all measurements in OSM (e.g., speed, width) are assumed to be in SI units unless some other unit (e.g., mph, feet) is included as part of the record.

OSM uses what it calls “relations” to group simple objects into more complex features. For example, a five-block long portion of a street, which is a single line object or way in OSM, may also be part of relations that also help define state highways, city limits, different bus routes, and park boundaries, among other possible roles. Relations help avoid the duplication of simple objects, since the same simple object can be used to define parts of many different complex features. For example, to map the city limit, an OSM user could draw a completely new polygon as a boundary around the city, on top of the street and other features that form portions of the city limit. However, this method would duplicate many of the existing simple features if the city limit overlaps existing roads and rivers. In an alternate method of defining a city limit using relations, a person could create a relation and add existing data (a section of street A, a section of street B, portions of rivers, etc.) to the relation to define the complete boundary. This method reduces the number of simple objects in OSM, since smaller objects are used as building blocks for many different complex objects. In another example of using relations, a bus route can be defined using a relation to group the different street segments traversed by the route, plus the bus stops along the street. While relations, when properly used, can be a more efficient use of OSM data structures, their complexity can lead to misuse and accidental data corruption by new users. This challenge is discussed in additional detail in Chapter 4.

Most of the data coding conventions within OSM are adequate to meet the needs of a multimodal trip planner, so a brief overview of these properties are described in the following section (using the convention “=*” to indicate that there are multiple possible values for a tag). References [10] and [11] contain greater detail, and are the sources for the descriptions and use of tags below. To support several desired features in multimodal trip planners, additional detail beyond the data that OSM currently records is required. The current data coding conventions for these features are described in more detail in the following section. We then present proposed changes to the coding conventions that will support these desired multimodal trip planning features.

STREET NETWORK

Streets and highways in OpenStreetMap are tagged as *highway=** [10] p.64]. The tag values *motorway*, *trunk*, *primary*, *secondary*, *tertiary*, and *residential* encompass most roads on which motor vehicle traffic is permitted; these values may be supplemented with *_link*, (e.g., *motorway_link*) to indicate an on-ramp, off-ramp, or continuous turn lane. *Highway=service* is used to denote commercial driveways, aisles in parking lots, and similar paths generally not for through travel. Unpaved roads (e.g., farm roads, U.S. Forest Service roads), often with restricted access, are tagged *highway=track*.

POINTS OF INTEREST AND LANDMARKS

These are so diverse that it is best to consult the OSM documentation directly [10] or [11]. Tag groups that may be of interest include *amenity=** for fountains, banks, public artworks, and

museums; *leisure=** for parks, nature reserves, golf courses, and sports fields; *power=plant* for power plants; *tourism=** for tourist attractions and zoos; *natural=** for wooded areas or bodies of water; and *shop=** for retail businesses. These data are used in the trip planner to help orient the user on the map but do not affect operation or results of the trip planner.

ADDRESS RANGE AND LOCATION FILE

OSM users document address ranges for roads [10] pp. 86-87] by recording an address at each end of a road segment and then indicating that address values between the endpoints are to be interpolated along the road. Tags used for addressing begin with the prefix *addr:*, as in *addr:housenumber=**.

LISTS OF BUSINESSES AND POINTS OF INTEREST, WITH LOCATIONS OR ADDRESSES

Addresses can be added as tags to specific buildings or businesses, again using the *addr:* tags. Tags for businesses of interest include *shop=** and *office=**.

OTHER MODAL NETWORKS: TRANSIT

OSM records bus stops as *highway=bus_stop*, and transit centers and bus stations as *amenity=bus_station*. There remains disagreement within the OSM community about how to record the alignment of routes and location of transfer points. Therefore, the research team recommends using GTFS data for this information instead of OSM. GTFS and OSM datasets can then be linked if necessary to leverage crowd-sourced contributions to transit datasets. An example software application that synchronizes GTFS and OSM data is discussed in detail in Chapter 4. Please see the corresponding section in the GTFS data requirements section on page 29 for a full description of transit data representation.

OTHER MODAL NETWORKS: BICYCLING

OSM's conventions for tagging bicycle lanes, shoulders, and separated bicycle paths are sometimes confusing, in part because of the original use of some tags to record official designations rather than functions. The clearest description of these attributes is on the OSM wiki [26]. Generally, a bicycle path separate from a road is tagged as *highway=cycleway*. A road that includes a bicycle lane is tagged as *highway=**, *cycleway=lane*. If a two-way road has a bicycle lane on just one side, in just one direction, this is tagged *cycleway:left=lane* or *cycleway:right=lane*; left and right are determined with respect to a direction indicator that is automatically recorded as part of the road when the road is created. The wiki [26] illustrates numerous variations of alignment and direction, with associated tags.

There are three major shortcomings in OSM's present tagging system for bicycle lanes. These shortcomings may reflect differences in conditions between European and North American cities, but it may also reflect the evolution of how bicycle lanes are placed on U.S. roads, or lack of agreement among cyclists on what is important and how to describe it.

First, there is currently no method within OSM to record what happens to a bicycle lane at an intersection. It is useful for a bicyclist to know whether a bicycle lane:

- continues through the intersection, to the left of a right-turn-only auxiliary lane, in what is sometimes referred to as a “slot lane” or “keyhole slot lane” [10] Ch. 3, p.127];
- enters the intersection to the right of right-turn-only auxiliary lanes; or
- stops altogether in the presence of auxiliary lanes on the approach to an intersection but resumes after exiting the intersection.

The present OSM tagging system does not provide a method for recording these differences.

Second, while there are ways within the existing tagging system to record wide shoulders used by bicyclists (e.g., two possibilities are *highway=**, *shoulder=yes*, *shoulder:bicycle=yes*, or *highway=**, *cycleway=shoulder*), there again is no consensus about how to code wide shoulder, or even whether wide shoulders should be coded. At least one contributor, who is one of the most prolific contributors of data within Florida, has expressed opposition to tagging anything with *cycleway* that does not meet current standards for new bicycle lanes, even if an older facility has signs, markings, and is in common use by bicyclists. This methodology would result in recording very few facilities in the state of Florida.

Third, there is no easy and clear way in OSM to indicate a street which has lanes that are wide enough to allow bicycles and car traffic to comfortably share the lane. There was a proposal for tagging lanes that actually have sharrows markings (i.e., markings consisting of a bike with arrows indicating the lane is shared), but the OSM user who proposed this tagging convention did not complete the specification. The research team has not found any consideration of special tagging for lanes that would be appropriate for sharrows designation but that do not have the actual sharrows marking. It is possible to record lane width in OSM, but it is difficult and often dangerous for the average public to measure this value, so few users are likely to record lane width data.

Before attempting to officially correct any of these shortcomings in the OSM tagging system, a consensus needs to be developed within the bicycling community about how best to describe a bicycle route to someone who may not be an expert cyclist. Developing such a consensus may be challenging, because

- the bicycling community is diverse and decentralized, with many local and national organizations; and
- the idea of providing such a description is novel. Such descriptions haven't been seen as necessary until now, when the possibility of a trip planner raises the question of how to describe recommended routes to novice cyclists. It may take time and effort just to persuade people who have been doing fine without it that new technology (trip planners) can create new needs.

Such consensus then can be used to determine what data needs to be recorded, and how. This requires additional research on how best to communicate variations in cycling infrastructure to

bicyclists, particularly to someone who is not familiar with the area, to minimize surprises and the hazards of coping with the unexpected. It is likely that some conventions may be established within the OSM tagging system before the cycling community determines what needs to be recorded. Additional changes can be suggested once consensus has developed.

OTHER MODAL NETWORKS: WALKING

Sidewalks are tagged as *highway=footway*. If there are steps from one elevation level to another, a separate way is created and tagged as *highway=steps*. If a portion of the sidewalk has a steep slope, the section can be created and tagged with *incline=** to record the steepness of the slope. *Handrail=** can be used to indicate whether there are handrails. A multi-use path may be tagged either as *highway=cycleway*, *foot=yes* or *foot=designated*; or as *highway=footway*, *bicycle=yes* or *bicycle=designated*. There appears to be no consensus on this coding convention, so all of these possibilities should be checked when downloading and using OSM data for automated trip planning purposes.

Features tagged as *highway=** include a tag *surface=** to indicate whether they are unpaved or paved, and how they are paved; *surface=paved* is assumed if the *surface=** tag has been omitted. Casual observation of OSM data in Florida indicates that some contributors are not aware of this, because the *surface=** tag has not been recorded in some off-road cycling parks or on some road trails paved with crushed shells.

Intersections between network links, or between links of different modal networks, are recorded as a node common to both links at the intersection. The node need not have any special tags, although some special tags, discussed later in this section, can be useful for multimodal trip planning.

RESTRICTIONS ON TURNING, ACCESS, DIRECTION

OSM can record restrictions on turns. The OSM wiki website includes additional detail and examples to explain how this is done [10] p. 92-93].

A one-way link of any kind is designated by adding a *oneway=yes* tag; *oneway=no* is assumed if the tag is omitted, but it can be added to clarify situations in which *oneway=yes* might be expected. The direction of a link is recorded by confirming or changing the directional indicator that is automatically recorded as part of every link.

Access restrictions are recorded *access=** tag for different types of vehicles, or for roads that are private rather than public. Roads with medians are recorded as two parallel roads in opposite directions, and linked together as a *dual carriageway* [10] p.96].

LOCATION OF TRAFFIC SIGNALS

Traffic signals are recorded by tagging an intersection node with *highway=traffic_signals* [10] p.69]. For an intersection involving one or more roads that is divided by medians, each intersection node receives this tag. Proposals have been made to allow grouping of nodes at complex intersections

and tagging the group rather than the individual intersecting nodes, but these have stalled and not gained much acceptance.

The OSM tagging system does not yet provide a way to record whether or not traffic signals at an intersection include pedestrian (e.g., walk/don't walk) signals, or whether they consist only of vehicle signals. A proposal to resolve this issue is discussed in a later section on intersections and crosswalks on page 47.

TRANSIT SCHEDULES

OSM makes no provision for recording this information, as OSM is designed to record three-dimensional spatial data, but not movement or a time dimension. Therefore, the research team recommends using GTFS data for this information instead of OSM. Please see the section on GTFS data requirements on page 31, which discusses the representation of transit schedules for multimodal trip planners within GTFS datasets.

POSTED SPEED LIMITS

Speed limits are tagged as *maxspeed=**. U.S. speed limits should be recorded in miles per hour and accompanied by mph (e.g., *maxspeed=45 mph*), rather than being converted to kilometers per hour. At locations where the speed limit on a road changes, the road should be split into multiple ways and each way tagged with the appropriate speed limit.

TOLLS, FERRIES, AND ROAD CLASSIFICATIONS

Toll roads and bridges are tagged as *highway=**, *toll=yes*. If there is a uniform fee for the full length of the facility, this can be recorded as *fee=**. If there are multiple tolling points, entrances, and exits along the toll road, a short section on the way near each tolling point should receive the toll and fee tags [27].

Ferry terminals are tagged as *amenity=ferry_terminal*. The OSM documentation on how to record ferries is incomplete, but ferry systems such as the Washington State Ferries have been recorded in GTFS format. Because of the similarity to other forms of public transportation, GTFS is a better format for representing ferry schedule and route information, because the information is similar to that of other forms of public transportation.

The most common road category used to guide routing appears to be freeways, which may be preferred or avoided. Most U.S. freeways are tagged as *highway=motorway*. Some have been tagged as *highway=trunk*, although there seems to be an effort to retag these as motorways. For planning bicycling and walking trips, preference might be given to low-volume, low speed roads, which most likely would be tagged as *highway=residential* or, if accompanied by significant efforts at traffic calming, as *highway=living street*. Unfortunately, the original import of U.S. Census TIGER data tagged some major roads as *highway=residential*, so these data need to be checked if OSM data are to be used for guiding the routing algorithm (instead of just providing the links over which routes are to be calculated).

TRANSIT FARES

Transit fares are not represented in OSM, and instead are represented in GTFS for multimodal trip planning. Please see the corresponding discussion of GTFS data requirements on page 32 for a full description of transit fare data representation.

TYPE OF TRANSIT VEHICLE

The type of transit is generally tagged within OSM as an attribute of the station or transit stop or, for rail modes, of the tracks (*railway=**). *Highway=bus_stop* indicates a bus stop, *railway=tram_stop* for a streetcar or light-rail stop, etc. Transit centers and terminals are tagged as *amenity=bus_station*, *amenity=ferry_terminal*, etc. However, for multimodal trip planning GTFS is a more suitable source for public transportation stop, route, and schedule information. Please see the discussion of GTFS data requirements on page 32 for a list of different types of transit vehicles that can be represented in GTFS.

ABILITY OF TRANSIT VEHICLE TO CARRY BICYCLES

The OSM tagging system does not record whether transit vehicles serving a route or stop can carry bicycles. Although the tagging system could be augmented to do so, this sort of information is more a characteristic of vehicles rather than of the physical features that OSM was designed to record. It would be more appropriate to add this data as an extension to GTFS, which is discussed on page 32.

WHEELCHAIR ACCESS

Any feature in OSM can be tagged with a *wheelchair=yes/no* tag. Thus, *highway=bus_stop*, *wheelchair=yes* would indicate a bus stop that can be used by someone in a wheelchair. There is additional discussion of this attribute in the later section on intersections and crosswalks on page 47.

AMENITIES

For transit stops, attaching tags of *bench=yes/no*, *shelter=yes/no*, and similar attributes to *bus_stops* is fairly standard in OSM. In addition, *amenity=bench* can be coded as a point separate from the bus stop but located next to it (*amenity=shelter* is for tagging hiking shelters in mountainous areas). *Amenity=toilets*, *amenity=drinking_water*, and *amenity=vending_machine* also may be of use when providing information about a planned trip.

As yet, OSM has no provision for recording whether a feature is shaded. The presence of shade can be important for trip planning, especially when a large distance must be traveled on foot in extremely hot weather, such as that encountered in Florida during the summer months. The research team proposes adding the following tags to any feature that is shaded during midday hours during the hot months of the year.

*shade=**

- *shade=trees* if the sidewalk, street, bus stop, or other feature is shaded by trees sufficiently during the middle of a summer day; do not record *shade=trees* for individual trees; record it only for a length of sidewalk with reasonably complete coverage, dividing the way into sections as appropriate.
- *shade=pergola* if the sidewalk is shaded by a pergola supporting vegetation.
- *shade=portico* if the sidewalk is shaded by a portico running the length of a building.
- *shade=building* if an east-west sidewalk runs close enough to the north side of a building that it is shaded at midday during summer
- *shade=roof* if the sidewalk is shaded by some other impervious structure (e.g., awning, roof, or similar canopy other than a portico) that also is capable of providing shelter from rain. If a sidewalk runs beneath an archway or tunnel through a building, code the walkway with *breezeway=yes* instead of *shade=roof*. If a bus stop has a shelter and also is shaded by trees, record “trees” if the trees provide more extensive shade than the shelter
- *shade=none* is assumed if *shade=** is omitted, but if a feature has been checked in the field and found to have no midday summer shade, *shade=none* will indicate that this has indeed been checked.

Amenity=parking is sufficient to record parking lots. The tag can be added to *building=yes* to denote a parking garage. To support routing, there should be a road intersecting the lot or building and the adjacent street network (usually *highway=residential* or *highway=service*). Handicapped-accessible parking spaces can be drawn as a separate area within a parking lot, with the *amenity=parking* tag supplemented with *access=disabled* and, if desired, *capacity:disabled=** for the number of spaces. Proposals have been made for tagging on-street parking, but these have not yet been adopted. *Park_ride=yes* indicates a park-and-ride lot, although the type of public transportation served by the lot can be recorded instead of the *yes*.

Lit=yes/no indicates whether a feature is lit at night. This attribute could be presented to users when planning a trip, as some users may prefer traveling in lit areas during the night for safety reasons.

VALUES TO CALCULATE SUITABILITY OR LEVEL OF SERVICE

BICYCLE AND PEDESTRIAN LEVELS OF SERVICE

As mentioned earlier, the research team reviewed the BLOS [18] and PLOS [19] reports to determine what variables both LOS measures require. In addition, the research team examined a study that calculated a BLOS for intersections rather than for road segments [25]. The following sections discuss the availability of data from various sources in more detail.

The following sections discuss the availability of data from various sources in more detail.

Table 2 lists the variables needed in these LOS measures. For each variable, the table also lists whether it is available in OpenStreetMap and, if not, an assessment of whether it realistically could be added to OSM. The table also indicates the likeliness of availability from state, county, city, or MPO GIS data files based on an examination of a small set of city, county, MPO, and state data files, such as the FDOT TRANSTAT and the RCI Field Handbook. Note that in some jurisdictions, some variables may have been coded in engineering files that could, in principle, be processed to obtain the variables for geographic processing. However, extracting the information from these engineering files is likely to require substantial effort, and direct field measurement might be more cost-effective than attempting to transform existing datasets. The following sections discuss the availability of data from various sources in more detail.

Table 2 - Data needed to calculate bicycle and pedestrian level of service measures

LOS Data Attributes	Requirement for LOS Data Attributes by Mode			Availability of Data	
	Bicycle (segment)	Bicycle (intersection)	Pedestrian	OpenStreetMap - Current Coding Conventions	Transportation Agency Data Files
Posted speed of motor vehicle traffic	Yes	-	Yes	Yes	Likely
Number of through lanes in direction of traffic	Yes	Yes	Yes	Possible, but some lane counts may include auxiliary turn lanes	Likely
Presence of bicycle lane	Yes	-	-	Yes	Varies
Presence of sidewalk	-	-	Implicitly yes	Yes	Varies
Average daily motor vehicle traffic on segment	Yes	-	Yes	No, and unlikely	Major roads only
% of daily motor vehicle traffic which is heavy vehicles	Yes	-	-	No, and unlikely	Subset of major roads only
Percentage of segment having on-street parking	Yes	-	Yes	No, but possible	Unlikely

LOS Data Attributes	Requirement for LOS Data Attributes by Mode			Availability of Data	
	Bicycle (segment)	Bicycle (intersection)	Pedestrian	OpenStreetMap - Current Coding Conventions	Transportation Agency Data Files
Width of pavement used for on-street parking	Yes, only if bike lane is present	-	Yes	No	When available, only for major roads
Total width of street pavement in direction of travel	Yes	-	-	No, possible but unlikely	Major roads only
Width of road shoulder	Yes	-	Yes (include bike lane if present)	No, but possible	When available, only for major roads
Road pavement condition	Yes	-	-	No, possible but unlikely	When available, only for major roads
Total width of outside through lane	-	Yes (include bike lane if present)	Yes	No, possible but unlikely	Major roads only
Volume of directional traffic during a 15-min time period	-	Yes	-	No, and unlikely	Subset of major roads only
Total width of the side street (including auxiliary lanes and median)	-	Yes	-	No, possible but unlikely	Major roads only
Width of buffer or planting strip between sidewalk and road, and how planted (grass/gravel only, trees, shrubs)	-	-	Yes	No, but possible	When available, only for major roads
Number of driveways intersecting road segment	-	-	Yes	Not directly, but possible to compute from existing coded data	When available, only for major roads

AVAILABILITY OF DATA ATTRIBUTES IN OPENSTREETMAP:

Posted speed can be recorded using the current OSM coding conventions (*maxspeed=**). The LOS models use an estimate of actual speed calculated from the posted speed. Actual speed would be much more difficult for OSM's amateur contributors to measure and record.

The number of through lanes in direction of traffic can be recorded as *lanes=** but generally is not recorded except for major streets. Total width of street pavement in direction of travel can be coded in OSM, but it is difficult for the average person to measure, and on busy streets it would also be dangerous to collect.

Presence of a bicycle lane can be recorded as *cycleway=lane* as an attribute of the road. As noted earlier, the present tagging system does not clearly distinguish between a bicycle lane (with signage and markings) and a bicycle shoulder (which may be of equivalent width to a lane but lack markings or signs). The present tagging system also does not discuss recording the width of bicycle lanes. There is a proposal to add *shoulder* as a separate feature of roads and code attributes for the shoulder, including width and access by bicycles. Bike paths separated from the motor vehicle traffic surface can be coded separately, with width and other variables, but these were not part of the BLOS calculations.

Presence of a sidewalk can be recorded now as a separate way (*highway=footway*), but not as a feature of the road (as discussed with bike lanes above). There have been two proposals to code sidewalks as attributes of the road, but both have been abandoned, apparently because they became too complicated. The research team recommends that sidewalks be coded separately in OSM, as presently suggested, and linked to the adjacent street using GIS or even as a relation in OSM.

There are three LOS variables involving volumes of motor vehicle traffic:

- 1) average daily volume
- 2) percentage which is heavy vehicle traffic (e.g., buses, large trucks)
- 3) average 15-minute volume of motor vehicle traffic in the direction of bicycle traffic).

None of these variables are currently coded in OSM, and it is unlikely that they will be, because they are difficult for an average person to observe. And, except for along major arterial roads, these variables are not routinely recorded by transportation agencies, either.

There are two LOS variables involving on-street parking:

- 1) percentage of street having on-street parking, and
- 2) width of surface used for on-street parking.

At present, OSM does not provide coding conventions for on-street parking. There is a proposal under discussion to do so, but it has become complicated, and therefore likely to be abandoned. The

BLOS variables would need to be calculated using OSM data, because the current proposal would not record these variables in the forms required for the BLOS estimates.

Coding conventions for road pavement condition are currently not defined in OSM, and because road pavement condition is subjective, it is unlikely to be included in the future. In the recent past, proposals to add tags for variables that are inherently subjective have not been embraced by the community.

Total width of the side street being crossed can be recorded in OSM as *width=**. However, street width is not often recorded in OSM because of the difficulty and risk involved to an individual when measuring it.

OSM does not currently provide coding conventions for buffer strips (e.g., grassy areas) between sidewalks and adjacent streets, but it would be simple to propose such a feature as an attribute of the sidewalk in OSM. The research team has tested the concept for sidewalks on the USF campus. Recording sidewalk width would take more work on the part of the mapper, and we suspect it would not be commonly used even if approved. It would be easy to observe the presence and type of any plants or other vegetation in a buffer strip.

The number of driveways intersecting a sidewalk is not explicitly recorded as an attribute of a sidewalk in OSM, but if the driveways are recorded properly then the number of intersections with sidewalks could be calculated from the OSM data using GIS techniques.

AVAILABILITY OF DATA ATTRIBUTES IN PUBLIC DATA FILES

The research team examined documentation for the FDOT TRANSTAT data files, which covers the state highway system. For urban state highways within one mile of the designated urban boundary, it appears that TRANSTAT and the RCI Field Handbook [29] record most of the variables needed for the LOS calculations. However, the percentage of segment length with on-street parking and the number of driveways intersecting the road do not seem to be recorded. The Hillsborough County files examined by the research team include data only for county roads and did not include all of the LOS variables. Arterial roads that are not maintained by the state or county are not included in these datasets.

Thus, data may be available to allow measurement of LOS on roads with high volumes of vehicles moving at fairly high speeds. However, staff at the City of Portland, Oregon's bicycle program [28] estimate that perhaps 7% of the population is willing to ride on such roads when bicycle lanes and similar facilities are available. Sixty percent of the population is unwilling to ride on such roads, regardless of the facilities on them (or the estimated LOS). Data are generally unavailable to estimate LOS for roads with lower traffic volumes and lower speeds, where some might be willing to ride. This suggests that research is needed to develop a measure of LOS for such roads, using only variables that the general public can measure safely and reliably. This research would enable neighborhood groups, bicycle clubs, or the general public to contribute data needed for LOS to a system such as OSM, which would also reduce the burden and cost to a city, county, or state to maintain such records.

INFORMAL ROUTES AND SHORTCUTS

These features can be tagged as *highway=footway* or *highway=path*, with *surface=**. There is a risk that making more people aware of such routes could lead adjacent property owners to block access to them.

INTERSECTIONS AND CROSSWALKS

Given OSM's origins in Europe, and the importance of walking as a transportation mode in Europe, OSM's tagging and terminology reflect conditions and practice in European cities. While the tagging system works reasonably well in older parts of U.S. cities, it can be confusing and inadequate in suburban areas. The following subsections describe the current OSM coding conventions as related to intersections and crosswalks, as well as suggested improvements.

TAGGING CROSSWALKS – ROUTING AND VISUALIZATION

If a sidewalk or trail crosses a street via an overpass or tunnel, the way is tagged with *bridge=yes* or *tunnel=yes*, and the ways are **not** connected with an intersection node. These features are clear and do not create any confusion or problems. If access to the bridge or tunnel is via stairs, elevator, or steep slope, tags are available to record this design clearly on the approaches or, in the case of a sloping bridge, on the bridge itself.

Figure 9 depicts a simple at-grade street intersection with sidewalks and crosswalks. Looking at the sidewalks and crosswalks shown to the right of Street A, and going from top to bottom, there is

- a sidewalk (*highway=footway*)
- a short section with steps (*highway=steps*)
- another short stretch of sidewalk (*highway=footway*)
- an intersection with a crosswalk (labeled “crossing 2” in the Figure) that crosses Street A
- another short stretch of sidewalk
- an intersection with a crosswalk (labeled “crossing 1” in the Figure) that crosses Street B, and also with a sidewalk parallel to Street B
- the crosswalk (“crossing 1”), with its intersection point (labeled “Node 1” in the Figure) with Street B
- an intersection (labeled “Node 2” in the Figure) with a sidewalk on the other side of Street B
- a short stretch of sidewalk (*highway=footway*)
- an intersection with a crosswalk that crosses Street A, and with another sidewalk parallel to street A, and, finally
- the sidewalk continuing along street A (*highway=footway*).

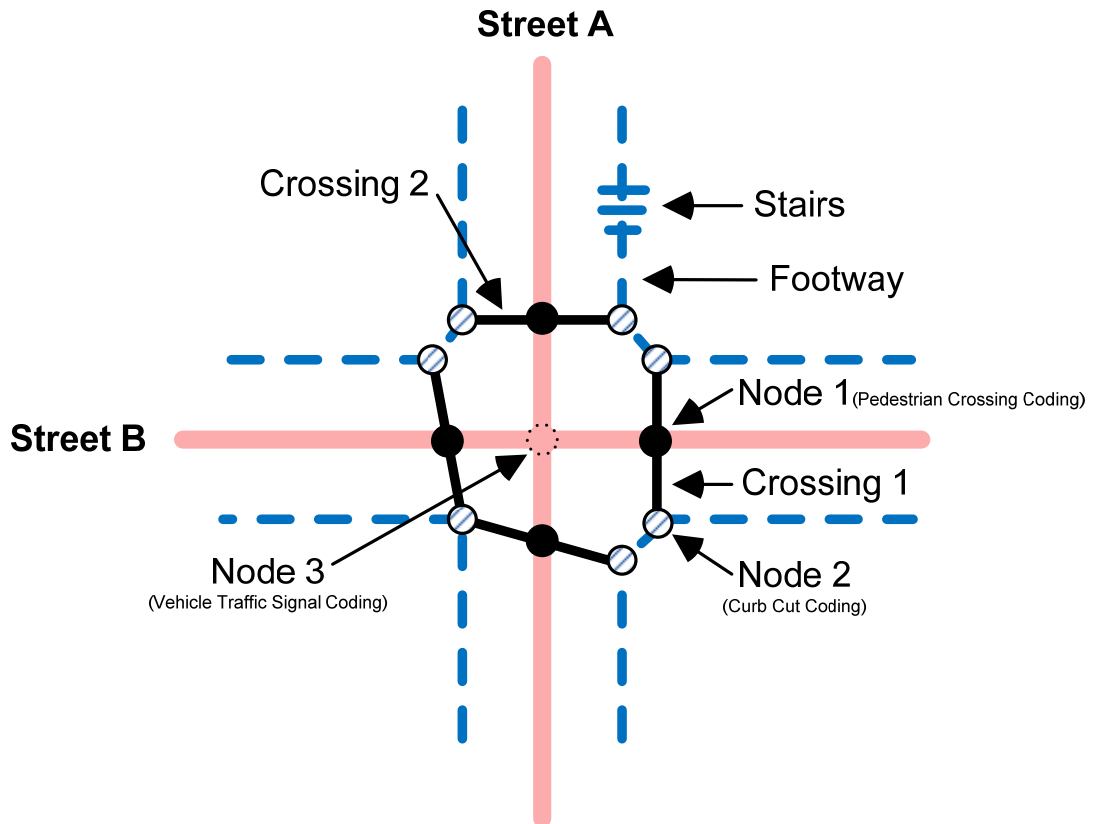
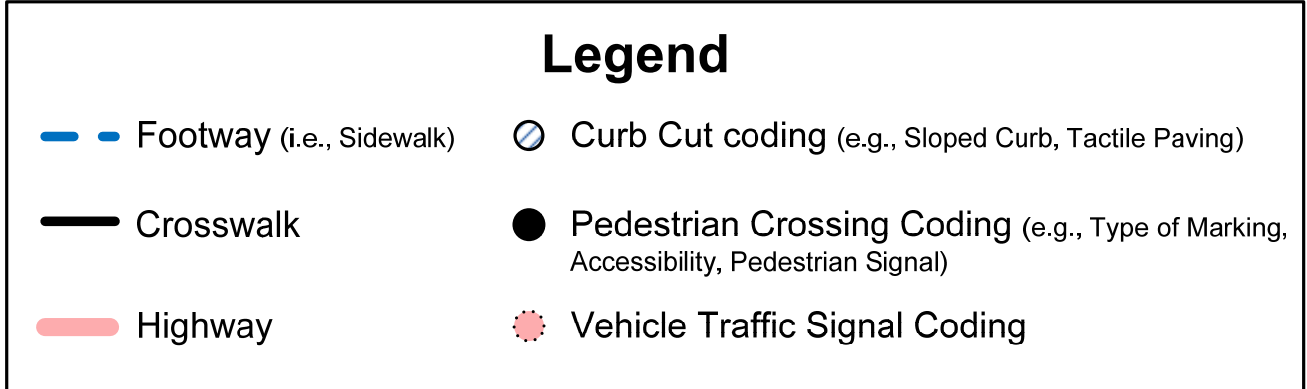


Figure 9 – Example intersection to illustrate tagging of crosswalks, sidewalks, curb cuts, and signals

In OSM, crosswalks are tagged not as crosswalks but as continuations of the sidewalk network on each side of the street. Thus, “Crossing 1” in the above sequence would be tagged as *highway=footway*. If this were a place where a bicycle trail crossed a street, instead of a sidewalk network, then “Crossing 1” would be tagged as *highway=cycleway*. Node 1 in the above sequence links the crosswalk (or trail crossing) to the street being crossed. Characteristics of the crossing itself then are coded to this intersecting node, which is tagged as *highway=crossing* with

appropriate additional tags. These additional tags indicate allowable use of the crossing (e.g., *foot=**, *bicycle=**, *wheelchair=**); marking of the crossing (e.g., *crossing=unmarked*, *crossing_ref=** for marking and structure); controls (e.g., *crossing=uncontrolled* for a simple crosswalk, *crossing=traffic_signals* for signals); and other features. *Crossing=no* can be used to tag a location where crossing is prohibited but might otherwise be expected. Except for *crossing=no*, the *crossing=** values are not mutually exclusive, and can be combined separated by semicolons (e.g., *crossing=traffic_signals;crossing_ref=zebra* for a crossing at an intersection with signals, marked on the road with a series of closely-spaced parallel lines).

If there are four crosswalks at a four-way intersection, as in Figure 9, there would be four ways, each tagged as *highway=footway*, and each containing an intersection node tagged *highway=crossing*. This allows recording of variations that may exist among the crosswalks. If, as often occurs in suburban Hillsborough County, the intersection has only two or three crosswalks, then only these need to be coded, although the missing ones could be coded as *crossing=no*. As noted earlier, the presence of traffic signals for the street is also tagged for the intersection of the two streets, so Node 3 would be tagged as *highway=traffic_signals*.

Confusion arises when using the data to draw a map. A crosswalk that is tagged with *highway=footway* will be drawn to look like the rest of the sidewalk network, which is also tagged with *highway=footway*. If the paving surface of the crosswalk differs from that of the sidewalk (e.g., *surface=paving_blocks* vs. *surface=concrete*), and if the paving surfaces for both have been recorded, then it would be possible to use the surface type to distinguish between the two. However, if the pavement types have not been tagged, or if they have both been tagged simply as *surface=paved*, then the only information about the crosswalk itself is what has been coded to the node with *highway=crossing* (Node 1 and its equivalents on the other crosswalks in Figure 9). The only way around this difficulty is to do additional preprocessing of the data, manually or with automated by software, to retrieve information from the node tagged as *highway=crossing* and apply it to the portion of the link that contains the crossing node. Most commercial GIS software would not do this kind of preprocessing automatically, but customized scripts may be able to do this. In a suburban environment with wide multilane streets, it would be misleading and dangerous to depict a crosswalk as just another section of sidewalk.

While the research team was preparing a proposal for the OpenStreetMap community to address this problem, another member of the OSM community initiated a proposal that defines a slightly different approach [30]. This proposal adds the tag *footway=crossing* to the way that depicts the path of the crosswalk (Crossing 1 in Figure 9) and contains the node tagged *highway=crossing* (Node 1 in Figure 9). Because this proposal still addresses the issues of concern, the research team supported this existing proposal rather than creating a separate conflicting proposal. This proposal was approved in late April, 2011. Where bicyclists are allowed to cross the street by riding in the crosswalk, *bicycle=yes* should also be added. By extension, where a bicycle trail crosses a street, the way that depicts the crossing should be coded as *highway=cycleway*, *cycleway=crossing*, adding *foot=yes* as appropriate. The research team suggests this also be done where bicycle or multi-use trails cross streets at grade, using *bicycle=yes* and, as appropriate, *foot=yes*.

If the information is to be used only for routing, rather than for drawing a trip on a map, recording information on a node is equivalent to recording the same information on a way, as long as the trip planner software has been designed to recognize data tagged to nodes instead of ways. The research team's review of the OTP software indicates that while OTP reads tags on nodes, OTP only uses way tags when routing. For OTP to use any of the crossing data tagged on OSM nodes, the software would need to be modified to identify each node tagged as *highway=crossing*, read the tagged data from the node, and transfer it to the way that contains the node. This process could be automated if such ways are recorded using the new additional tag as *highway=footway*, *footway=crossing* within OSM. Given that more than 99% of the features tagged as *highway=crossing* in OSM are nodes rather than ways [31], it seems unlikely that the OSM community would approve changing the tagging system to shift all of this information from nodes to ways. However, using the new additional tag is a relatively minor change that can easily be adopted by OSM users and simplify the automation of data processing by OTP.

CURB CUTS AND TACTILE SURFACES

It is not unusual for one end of a crosswalk to connect to a sidewalk at a curb cut (such as Node 2 in Figure 9), with or without tactile paving that can be detected by someone who is blind, and for the other end to connect to a sidewalk that lacks a curb cut, tactile paving, or both. If trip planning is to support persons who are blind, then the node at each end of the crosswalk should record how the sidewalk and crosswalk connect. Tagging for this purpose is an evolving practice, but curb cuts can be tagged as *sloped_curb=yes/no*, *tactile_paving=yes/no*. An additional tag could be added to describe the type of tactile surface.

Based on observations of local sidewalks and intersections, the research team suggests allowing additional values for *sloped_curb=**.

- *sloped_curb=at_grade* would indicate that the sidewalk is level with the road grade and requires no change in slope where they intersect. This could occur where a sidewalk meets a raised crossing at curb level, or when a street has no curb and the sidewalk runs at street level. This would let a person who is blind know not to anticipate a change in slope before crossing a street.
- *sloped_curb=ramp* would indicate that instead of the slope occurring within the space occupied by sidewalk (and outside the space occupied by the street), the slope exists as a short ramp built as an extension of the sidewalk into the area of the street or parking lot.

The present tagging system does enable recording information about different kinds of audible and tactile signals for use by persons who are blind.

PEDESTRIAN TRAFFIC SIGNALS

The research team recommends some additional tag values to improve the usefulness of information provided to users of a trip planner.

Crossing=traffic_signals does not distinguish between crosswalks with pedestrian walk/don't-walk signals, and those that have just red/yellow/green traffic signals for car and bicycle traffic. The research team suggests *crossing=pedestrian_signals* for the former, and the current *crossing=traffic_signals* for the latter. We have encountered signalized four-way intersections at which some crosswalks have walk/don't walk signals, and others do not.

Although there are standards for signals at intersections, these standards change over time. Traffic and pedestrian signals that meet older standards might not be updated to meet new ones until an intersection is rebuilt. Thus, signals are not standard across an entire city. A person who is familiar with a particular intersection will know what to expect from the signals at that intersection. A person who is using an intersection for the first time as a result of a trip planner will not know this information and, through ignorance, may engage in risky behavior. In the interest of pedestrian safety, the research team also suggests coding information to better communicate what a pedestrian should expect to encounter at a given intersection, including:

- Are there pedestrian signals, or just signals for car traffic?
- If there are pedestrian signals, is a walk signal automatically displayed as part of the traffic signal cycles, or must the pedestrian actively request it (e.g., by pushing a button)?
- If a pedestrian presses a button at the signal, does this advance the change of signal, or merely request the walk signal to be displayed whenever the signal changes at a fixed time?
- How long is the signal cycle time (how long should a pedestrian who just missed the walk signal expect to wait before the walk signal is next displayed)?
- If the pedestrian does not press the signal button, or if there are no pedestrian signals and no cars arrive to trigger sensors in the roadway, will the signal eventually change on its own?

This information could be posted at the intersection, but if it is widely available it should be part of the trip planner as well to reduce the uncertainty involved when planning a trip. Similarly, when it is available, information should be recorded which indicates whether in-road equipment to detect cars for advancing the traffic signal can also detect bicycles.

The research team considered developing a set of tags to record information about signal operations within OSM. However, this information is time-consuming and difficult for an average OSM contributor to record. Part of the difficulty arises because signal operating modes may be set to vary on daily or weekly cycles. Part also arises because some of this information can vary within an intersection, depending on which street is being crossed. It therefore seems unlikely that the average public contributor to OSM will record this information. If this information is needed for a trip planner, it may need to be obtained from state or local transportation department files and merged with the data from OSM, GTFS, and other sources.

DATA OUTSIDE THE SCOPE OF GTFS AND OPENSTREETMAP

Many datasets outside of GTFS and OSM can provide information that may be useful in multimodal trip planning. Several of these datasets are discussed in the following sections.

DIGITAL ELEVATION

Incorporating elevation data into a multimodal trip planner will bring many benefits. If the slope of the ground is known, a trip planner can use these data to provide better estimates of travel times, and allow bicyclists and walkers to request either flatter or steeper routes. Elevation data can also be used to determine whether the slope of a path is below thresholds for wheelchair accessibility (currently a maximum running slope of 1:12 for ramps as set by the ADA Accessibility Guidelines [32]). Besides using the data for routing calculations, the trip planner can display the data as an elevation graph to allow the user to visualize the elevation changes for a given trip. Figure 10 shows an elevation graph for a bicycle trip using OpenTripPlanner. As shown earlier in Figure 4 in Chapter 2, OpenTripPlanner downloads elevation data over the Web from the National Elevation Dataset (NED) which is supplied by the U.S. Geological Survey [33].

SUITABILITY OF LINKS FOR BICYCLING OR WALKING

As noted earlier, suitability is inherently subjective, reflecting the judgment of an individual or group about which streets and facilities are or are not suitable for walking or cycling. For this reason, it does not fall within the scope of what OSM records. However, based on an informal survey of maps that cities and counties have published showing streets and cycling facilities, many

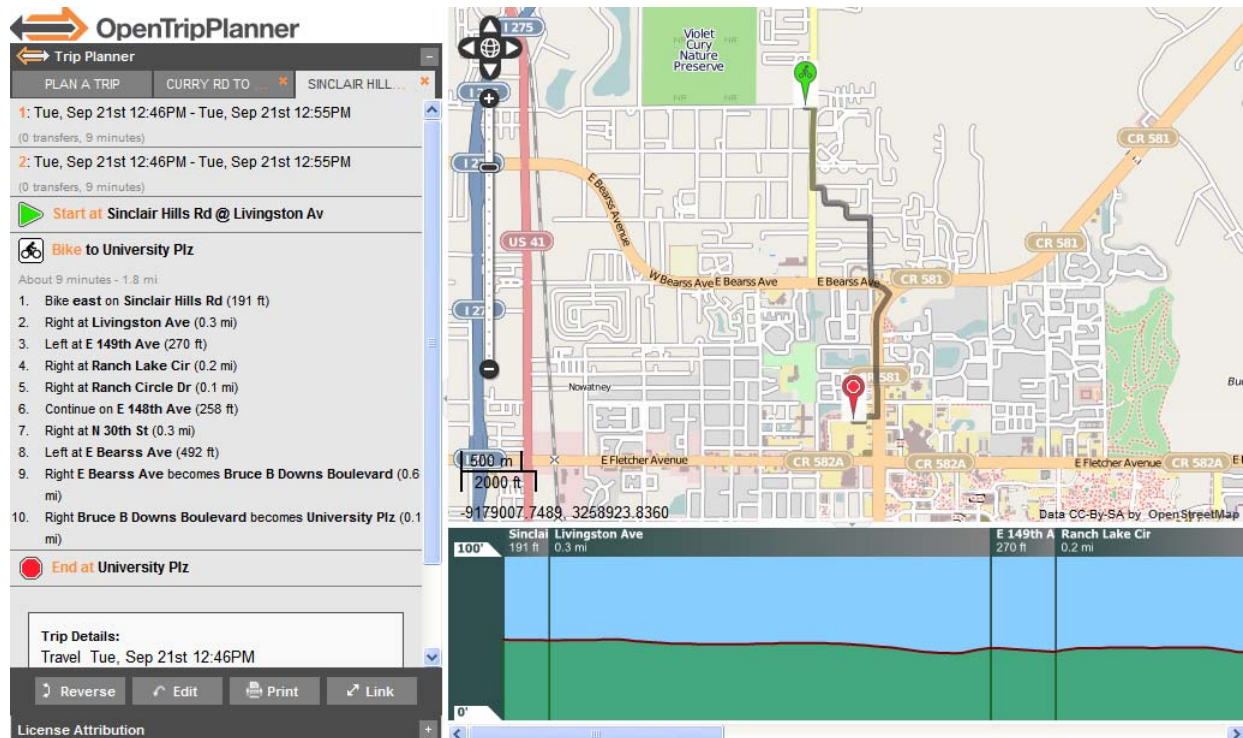


Figure 10 – OpenTripPlanner shows an elevation graph beneath the map for a bike trip

communities have developed local measures of suitability. Where such measures have been developed and have local support, a trip planner could use them to guide the routing algorithm.

CRASH OR SAFETY HISTORY

It is intuitively attractive to use data on crashes, injuries, or fatalities to guide a routing algorithm away from streets and facilities that have higher risk for walking or bicycling. The principal difficulty in doing this is developing an adequate measure of risk. For example, a major road may have a larger number of bicycling injuries than any other in a region, but knowing that fact alone does not mean that the major road is the most dangerous road for cycling. The number of injuries might be high because more bicycling trips occur along that major road than other roads. Indeed, the risk of an injury during any cycling trip along that road could be lower than the risk on a road that has fewer injuries but many fewer cycling trips. Crashes and injuries are relatively easy and inexpensive to monitor. Estimating the number of trips along a given road is more difficult and can be more expensive, in part because in many areas cycling and walking trips are relatively infrequent. Simply using the number of crashes or injuries from a database could be misleading and incorrectly influence a user to walk or bike on other roads which are less safe. However, if adequate measures of risk that include exposure as well as incidents could be estimated for different roads, they could be used to guide the routing algorithm.

OPENTRIPPLANNER – RELATIONSHIP TO DATA AND DATA REQUIREMENTS

OpenTripPlanner uses an edge-based graph to store information about various transportation networks. Therefore, any data that is consumed by OpenTripPlanner to build a graph will be transformed into information for each edge connecting nodes in the graph. The Graphbuilder project, which is the sub-project in OTP that builds the graph, is capable of reading numerous formats of transportation data and storing this data in the graph that will be used later for routing. Data formats currently supported by OTP include GTFS, OpenStreetMap, ESRI ArcGIS shapefiles, and the National Elevation Dataset (NED). Since OTP is open-source software, modifications can be made freely to enhance the use of currently supported data formats as well as to add functionality for the use of other data formats.

In order to affect the routing behavior of OTP, a developer can increase or decrease the weight of an edge in the graph. Traditionally, when routing algorithms seek a path solution from a source to destination, the shortest path is defined by the path with the lowest cumulative weight. This means that a developer can use data attributes for different modes of transportation stored in the graph to affect the edge weights, thereby affecting the routing. For example, a developer can choose to support a “fastest” route request for a car trip. Assuming that the speed limits for the roads in the network are known, by setting the weights of edges with low speed limits to large values and by setting the weights edges with high speed limits to small values the developer can influence the routing engine to return a trip that consists primarily of high speed limit roads. In other words, the roads with lower speed limits are penalized with larger weights to influence the routing engine to prefer roads with higher speed limits. Similar techniques can be used to penalize each transit

transfer from one route to another (i.e., to ensure that the user only has to change routes a reasonable number of times), or to penalize large walking distances so that other faster modes of transportation are preferred.

A developer can also use data attributes to exclude edges from consideration when finding a route. For example, when reading in data from OSM, if a way has the tag *wheelchair=no* or has the tag *highway=steps* with the tag *wheelchair=**, where * is anything other than 'yes', a developer can force the software to mark that OSM way (which will become an edge in the graph) as non-wheelchair accessible. Otherwise, the software will assume that the way is wheelchair-accessible. When the OTP user requests a wheelchair accessible trip from the OTP website, OTP will exclude stairs and any non-wheelchair accessible ways from routing consideration.

OTP's use of edges rather than nodes to affect routing behavior is typical of routing algorithms. However, there is a conflict between this practice and one of OSM's tagging conventions. As noted earlier (Figure 9 and subsequent discussion), OSM records information about whether a crosswalk is accessible to wheelchairs, or whether it has other restrictions, in tags for the node where the crosswalk intersects the street, rather than for the way that represents the crosswalk. For OTP to use the crossing information, it will be necessary to add a module that looks for each node tagged *highway=crossing* and then moves the information from the node tags to the edge containing the node in Graphbuilder.

As of the date of this report, OTP employs two different methods for routing: the A* algorithm and Contraction Hierarchies. Originally, OTP used Dijkstra's algorithm and the A* algorithm to route trips for all modes of transportation. Those algorithms performed very slowly on large graphs and recent research has shown a new technique, Contraction Hierarchies [34], to perform better on large graphs. The idea behind Contraction Hierarchies is that a large graph can be contracted by removing vertices one at a time and replacing any paths through the removed vertex with a shortcut representing that path. As the OTP community implemented this functionality, it became apparent that transit routing would be difficult with this new approach as the nodes for transit were time-dependent. Their solution was to use core-based routing which means that only non-transit segments of routing would receive the benefit of faster performance with Contraction Hierarchies. Additional routing performance improvements in OpenTripPlanner have included implementing improved A* heuristics for transit routing [35]. There are also efforts underway to integrate real-time transit data for estimated arrival times into OTP transit routing, based on research by a UC Berkeley team which has also created the BayTripper (<http://www.baytripper.org/>) mobile transit application [7]. The most recent progress on the OpenTripPlanner project can be found at the OpenTripPlanner Developers Group [36], OpenTripPlanner Users Group [37], and Transit Developers Group [38] listserves/websites, as well as the main OpenTripPlanner website [12].

OTP uses an object called *Request* to store all relevant information on how the trip is to be routed which includes the desired mode(s) of travel, the start and end locations, the arrival or departure time, desired number of itineraries, the maximum walking distance, an optimization type, whether to route for wheelchair accessibility, and an intermediate places to visit in the route. Note that not all of these options are presented to the user via the website interface. The options that are not

shown in the OTP website are given ideal default values, which can be overridden by directly interfacing with the OTP API.

The route optimization types include routing for the *Fewest Transfers*, the *Quickest Trip*, and the *Safest Trip*. The user can select which mode to optimize for via a dropdown box labeled “Show me the:” as seen in Figure 11. If the *Fewest Transfers* optimization mode is selected, a transfer penalty will be included in the routing calculations to give priority to routes with the fewest transfers. The *Quickest Trip* optimization mode calculates the edge weight based on the effective length of the street after its slope is considered. The *Safest Trip* mode calculates edge weight based on bicycle safety values.

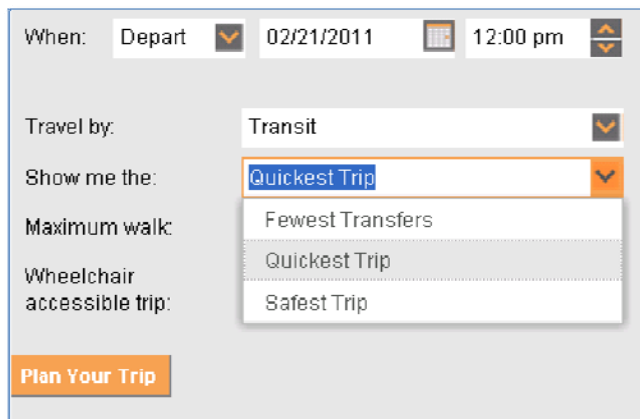


Figure 11 – Routing optimization options in OpenTripPlanner

An important milestone in OTP’s creation is the addition of geocoding functionality. With the Geocoder subproject enabled in OTP, a user can now type in actual location names in the *From* and *To* fields, as opposed to clicking on the map to place the start and end destination markers. Currently supported geocoding methods include using the Yahoo!, Google Maps, and Geocoder.us APIs as well as static lists, or a custom thesaurus, created by the software developer. An example of the formatting of location data for the custom thesaurus is below.

```
<bean class="org.opentripplanner.geocoder.GeocoderResult">
  <constructor-arg value="28.05954982491" />
  <constructor-arg value="-82.412258177082" />
  <constructor-arg value="Library" />
</bean>
<bean class="org.opentripplanner.geocoder.GeocoderResult">
  <constructor-arg value="28.058479962709" />
  <constructor-arg value="-82.416340499205" />
  <constructor-arg value="CUTR" />
</bean>
```

The benefit of the geocoder functionality is that a user may enter the address or the name of a location without needing to know how to locate that place on the map. The research team modified OTP to be able to recognize custom locations on the USF campus as well as addresses around the Tampa Bay area. The research team was able to contribute a new class file to the Geocoder subproject that can first search for a location in a custom thesaurus and if no location is found, it

will then search for a result from a third-party Web API such as the Google Maps geocoder. The results of searching for CUTR with the custom thesaurus geocoder implemented for USF campus is shown in Figure 12.

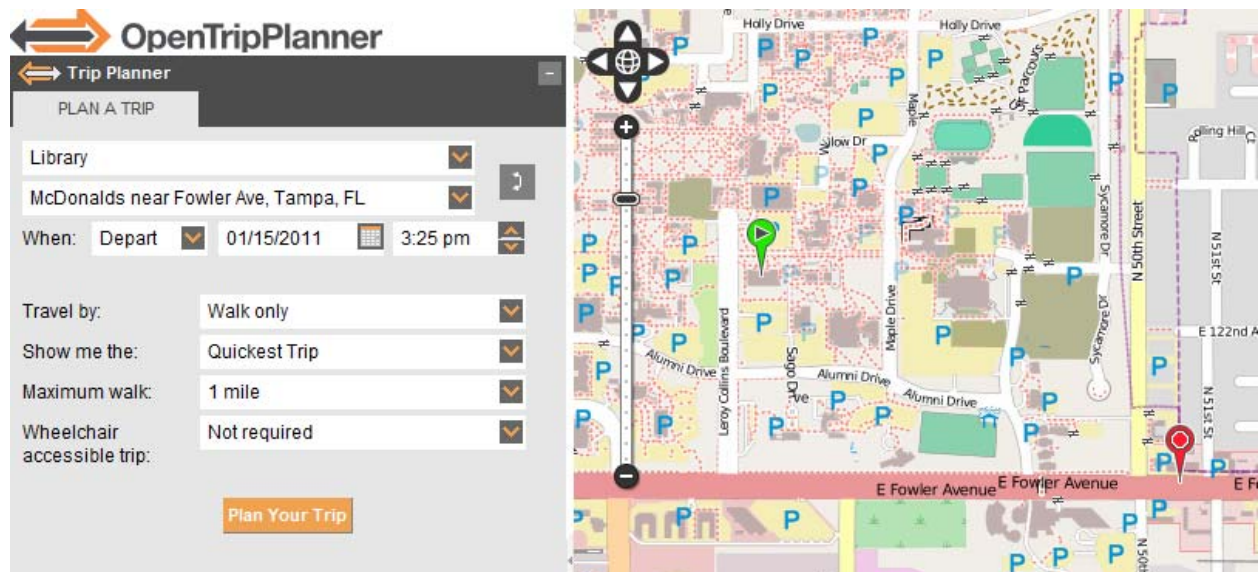


Figure 12 – Using a custom thesaurus geocoder, OpenTripPlanner developers can add custom locations, such as the USF Library and nearby McDonald’s restaurant

Another essential milestone for OTP will be the addition of car routing, so OTP users can compare car travel to bus, bike, and walking trips, or request routes that access transit by driving to a park-and-ride lot. Currently, two main pieces are missing that prevent OTP from routing for the car mode: turn restrictions and speed limits. The research team was able to contribute the integration of speed limits into OTP by using the *maxspeed* tag in the OSM data. As the OSM data is imported, the software reads in the speed limits from the *maxspeed* tag and converts from kph or mph into m/s, which is OTP’s internal measure of speed. When a route is requested for the car mode, the weight of the edge can be adjusted by factoring in the speed limit if known. If it is not known, the software assumes a typical driving speed of 15 m/s, which is around 35 mph. One can also improve the estimate of the driving trip segment time by factoring in the speed limit to the time calculation. The research team also added other enhancements in order to allow the user to select the car mode as well as to properly display itineraries on the OTP website with the car mode.

DISCUSSION OF OPEN MULTIMODAL TRIP PLANNING

Based on the evaluation of the research team, it is very feasible to implement a multimodal trip planner using open-source software and open sources of data. Since both the existing data and basic software are available at no cost, this significantly lowers the expense of establishing a multimodal trip planner for a specific area. This section discusses some of the primary strengths and challenges that the research team has identified in terms of using a system such as OpenTripPlanner for multimodal trip planning.

OPENTRIPPLANNER SOFTWARE SETUP AND MAINTENANCE

The development progress of the OpenTripPlanner software has been more rapid and comprehensive than anticipated when this research project began. OTP has most of the the major features desired of a multimodal trip planner and has an active developer community that is continuing to add features and improve the software [12]. Installing the OTP software on a new server requires technical expertise, although the research team expects that this burden will diminish over time for basic installations as more thorough documentation is produced. An initial setup of OpenTripPlanner on a USF server took approximately 10 hours with basic high-level documentation from the website that did not include low-level step-by-step instructions. One challenge to initial setup of OTP is that an active developer community means that new changes and enhancements are added almost daily, which can sometimes cause undocumented issues.

Modifying the OTP software also requires software development expertise, as OTP is a very large piece of software consisting of nine subprojects that are tightly interwoven. OTP is heavily dependent on many other projects and technologies, such as OneBusAway, GeoAPI, GeoTools, OpenLayers, Apache Tomcat, Maven, the Spring Framework, Hibernate, and Ext JS. The OTP API subproject contains 75 Maven dependencies, which are third-party libraries. However, once the organization of the OTP software is understood, modifications to the system are relatively simple to make. The majority of the software specific to OTP is written in Java, with the website written in Javascript, which are both common programming languages known by many software engineers and Web developers. Using common programming languages reduces the overhead for making modifications, since the engineer only needs to understand the structure of the system, instead of having to learn both a new programming language as well as system-specific structure. For example, the customized geocoder was added in around five hours, and a large portion of this time was spent simply understanding the current system. Additional modifications, for example to pull custom points of interest and their locations directly from a database, should be straightforward and only take a relatively small amount of time to implement. The enhancements to car routing were completed in approximately eight hours. These are not large development efforts for such large improvements in functionality.

For transit agencies or other organizations without in-house software development expertise, a non-profit company, OpenPlans.org, offers technical support for installing and customizing the OTP software. OpenPlans has indicated that it may perform these tasks for a basic transit-only implementation for approximately \$30,000 [39], which is fraction of the cost of developing new software from scratch, as evidenced with the cost of the Goroo multimodal trip planner in Chicago being funded by an approximately \$1 million Federal Transit Administration grant [40]. TriMet, the public transportation agency in Portland, Oregon, is using the open-source software underlying OpenTripPlanner as its primary transit trip planner and is using OpenPlans to provide enterprise-level support [41].

GENERATION AND MAINTENANCE OF TRANSIT DATA

From a practical standpoint, based on the existing technologies used for open data and open-source transit routing software, OSM should not be used as a direct primary source of transit data for

multimodal trip planners. The research team recommends that multimodal trip planning rely primarily on transit data in GTFS format, since GTFS was designed to support transit routing and OSM was not. However, OSM can still be leveraged as a source for secondary transit information such as bus stop amenities, and OSM can be used to improve existing GTFS data, such as the accuracy of bus stop geographic location.

Key information required for transit trip planning is the route schedule, and OSM was not designed to work with data that spans an amount of time, for any mode. In addition, OSM still does not have clear procedures for recording route information, and the procedures presently in use are complex and require levels of time and attention to technical detail that make them difficult for the average contributor. The research team coded a number of bus routes into OSM by hand, and concluded that at present it is not practical to develop software to automate this process, although it is possible to develop (or use existing [42]) tools to make manual coding of route information simpler and more reliable. On the other hand, OSM does handle the storage and updating of simple transit stops very well, and other data (e.g., amenities) can be linked easily to the stops. Using software synchronization tools, such as a tool discussed in Chapter 4 developed under this project, developers can still leverage the crowd-sourcing strengths of OSM for coding bus stop amenities and updating geographic locations of existing bus stops. This format is particularly attractive to transit agencies, since they will receive updated information about their bus stops at a fraction of the cost of a full bus stop re-inventory.

In accordance with the issues outlined above, OpenTripPlanner has been designed to use GTFS as a source of data for transit stops, routes, schedules, and other transit data, and to use data from other sources, such as OSM or commercial sources, for other modes. Some of the commercial scheduling software in common use by transit agencies, such as Trapeze and HASTUS, contains support for processing a transit agency's schedule data into a GTFS file [24].

Transit agencies are the primary managers of transit data, including the locations of bus stops. Agencies that provide GTFS data to Google for their Google Transit trip planning service already have an incentive to keep their GTFS data files current. Agencies that partner with others to develop multimodal trip planners for their service areas should have a similar incentive.

DATA COVERAGE

Open data sources such as GTFS and OpenStreetMap are designed to contain much of the data needed to support multimodal trip planning. However, data coverage remains incomplete, as current GTFS datasets and the OSM database do not provide dense data of uniform quality in all U.S. cities. While over 125 U.S. transit agencies have put their data in the GTFS format, many public transportation agencies have yet to provide data about their stops, transit routes, and schedules in GTFS format.

Open data repositories such as OpenStreetMap cover a wider range of facilities, but collection depends on the willingness and interest of individuals and community organizations to collect and record the data. As a result, geographic coverage in the OSM database is spotty, particularly in the U.S. For example, TriMet currently does not offer multimodal trip planning (e.g., bicycle, pedestrian

routing) even though TriMet is using OTP as their primary transit trip planner. This is primarily due to limited availability of OSM pedestrian and bicycle data in Portland. TriMet has hired a staff member and interns to improve and add to OSM data in its service area in anticipation of offering multimodal trip planning via OTP. Tracking TriMet's efforts to improve the OSM data could yield information valuable to other communities considering multimodal trip planners.

TriMet has also recently developed some useful methodologies for leveraging existing datasources, such as ArcGIS shapefiles for road centerlines, as part of this effort to expand OSM data in Portland [43]. These techniques appear to be useful to many different areas and agencies that want to quickly increase the amount of OSM data locally, although the methodology does require ESRI ArcGIS software as part of the data analysis process.

When leveraging public agency sources of GIS data, there may also be data gaps for inter-jurisdictional areas. Public agencies tend to focus data collection on the facilities for which they have responsibility or jurisdiction. For example, a state agency may collect data only for state roads, and a county agency only for county roads, so that some data may be available only for major roads. Some local governments may have data for roads within their jurisdictions, but others may not. For example, the City of Tampa made available to the research team a file that shows which intersections have traffic signals. The neighboring City of Temple Terrace does not have such a file (although the jurisdiction is small enough that members of the city staff were able to quickly compile a list).

Because of gaps in data coverage, it is likely that any multimodal trip planner will need to integrate data from a variety of sources and make assumptions when data are unavailable.

Except for transit, it is possible to use default assumptions in routing when data describing infrastructure is missing, and then target geographic areas or types of data to be completed and added to the planner.

MOTIVATING THE CROWD - OPENSTREETMAP PARTICIPATION

Analyses of the OSM database indicate that a majority of people who have registered with the OSM project website (and therefore can contribute data) have never actually contributed data. In other words, the majority of registered OSM users have not actually created or edited any geographic features in OSM. Also, many OSM users have contributed only once, likely near their initial user account creation stage. Additionally, most OSM that are considered active actually contribute very little data to OSM [44]. In fact, roughly 8% of registered participants have contributed most of the data to the OSM project. A sample survey of participants found that 96% are male, that 64% are 20-40 years old, and that 78% had at least a college degree. The authors note that participation inequality such as this is common for crowd-sourced information projects, and a similar difference in the contribution rates of men and women to Wikipedia [45] has also received recent attention.

A recurring theme on some of the OSM listserves is a desire to increase the number of people actively contributing to the project. In Europe, a common practice has been to hold mapping parties, where people get together for an evening or weekend to map an area, and invite newcomers to participate in the process [10], p. 23]. Although there have been mapping parties in

some U.S. cities (a mapping party held in Tampa in February, 2008 is what introduced members of the research team to OSM), this approach has not caught on in the U.S., and the reasons are unclear.

Another barrier to participation in the U.S. is that while some features can be mapped using GPS devices or aerial imagery, many features in a community cannot be easily mapped without walking around in an area. Based on the research team's experience, except for using a GPS device to record the path of a street, sidewalk, or trail, it is difficult and unsafe to collect data features and attributes from a moving car or bicycle. In addition, the Internet-based tools that the OSM community has created for entering and correcting data in the OSM database do have a learning curve. The OSM community recognizes this is a barrier and in response has developed tools that are simpler to learn and use. However, the learning curve remains a challenge for new OSM users.

OpenStreetMap and other crowd-sourced information projects depend upon what Shirkey [47] refers to as *cognitive surplus*—the free time of an educated, Internet-connected citizenry. The phenomenon of focusing this cognitive surplus toward specific objectives is relatively new, because it depends on recent technological and market developments that have reduced the barriers to collecting data (e.g., GPS receivers, digital still and video cameras), producing information (e.g., low-cost tools for editing and reformatting GIS data), and sharing it (e.g., via the Internet).

There have been too few successful examples of focusing cognitive surplus—and too little research on existing examples—to predict which efforts will draw large numbers of participants, or to offer guidance on how to motivate participation in any particular effort. Shirkey notes that participants' motivations are important. Different projects appeal to altruism, desires for self-expression, ability to create, enjoyment, recognition, belonging to or building a community, career development, problem-solving, or the possibility of monetary or social reward. In addition, the way that projects are designed and implemented affects their ability to appeal to different motivations. Shirkey cites examples of several pairs of projects that had similar objectives, of which only one succeeded because its implementation enabled different motivations to be satisfied.

The research team believes it should be possible to engage neighborhood and other community organizations in gathering and reporting data on transportation infrastructure that could be used in multimodal trip planning (e.g., sidewalks, crosswalks, shelters at bus stops). Such efforts will be more likely to succeed if they can show citizens how making such an effort will benefit them directly. If the data collection can be organized in ways that make it easy for participants to observe and record data without everyone in a group having to do the work of recording the data in OSM or other data repository, this effectively lowers the OSM learning curve for the majority of the group. This design may prove to produce the most successful data collection efforts. More research is needed to identify, develop, and document successful strategies for mobilizing communities to map their transportation infrastructure. The research team has identified several potential opportunities in Sarasota and Hillsborough counties in Florida where neighborhood organizations might undertake such work. If these come to fruition, members of the research team hope to learn from them.

In the research team's discussions with other groups about the research project and the possibility of developing a multimodal trip planner for the Tampa Bay region, several other ideas have

emerged for increasing public involvement in OSM or other data collection efforts. One suggestion is to design targeted collection projects that could be completed by students who attend high schools that have requirements for participation in service projects. Another is to design targeted projects that might appeal to persons who have retired, and who might have both time to participate and interest in alternatives to driving. Simply getting coverage of OSM in one of AARP's publications might increase participation. A third suggestion is to invite participation by various groups that have volunteered to clean up litter along roadways through an Adopt-a-Road/Highway program; such groups have demonstrated willingness to contribute time for a public good, they would be familiar with the roads they help maintain, and some might have an interest in mapping these roads. A fourth suggestion is to work with universities that have classes with curriculum or class project requirements in subject areas that might overlap with OpenStreetMap (e.g., civil engineering, community design, architecture, surveying, geographic information systems, global positioning systems, location-based services). For example, the research team has been successful in collaborating with the Global Positioning Systems (GPS) class offered by the USF Department of Geography. The professor who teaches the GPS class now offers the option to students to collect GPS data for specific OSM attributes as part of one of their class projects. For greatest success, requests for participation and examples of OSM projects should be tailored for each of these groups of potential OSM users. Future work should examine how outreach to different organizations that may be interested in collecting OSM data can be targeted towards priorities and features than have the greatest impact to those groups.

In July, 2010, MapQuest.com announced that it would partner with OSM, with a goal of making OSM a major source of the data used in MapQuest routing [48]. As part of this process, MapQuest committed to provide \$1 million to support the growth of open-source mapping in the U.S. in areas where one of its affiliated organizations is active. The company also has developed an open.mapquest.com website for the U.S., and similar sites for other countries. These sites invite people to contribute data about points of interest and other features, and the company then makes the contributed data available to OSM. As of late February 2011, details of the company's \$1 million program to support open-source mapping in the U.S remain unclear.

EXTENDING THE OSM TAGGING SYSTEM

The OSM tagging system generally supports recording of the data needed in multimodal trip planning, but some changes to the tagging system would improve its usefulness for this purpose. The suggested changes were discussed in detail earlier in this chapter and are summarized below.

- Additional tagging options are needed to describe how bicycle lanes are placed at intersections, in order to advise bicyclists what to expect as they approach an intersection that is unfamiliar to them. In addition, some cycling facilities such as wide paved shoulders, wide outside lanes, and lanes designated with sharrows also lack tagging options in OSM. However, the bicycling community needs to first agree on what needs to be communicated, and on terminology for describing conditions to persons who may not be expert cyclists. Consensus may take time to develop, given the diverse, decentralized nature of the bicycling

community. Some additional research is needed to support defining a more inclusive tagging system for bicycle infrastructure.

- Additional tagging options are needed to indicate whether facilities—especially for walking, cycling, or waiting for transit—have shade during summer months.
- Additional tagging options are needed to record information about the space, if any, between a sidewalk and a parallel street.
- OSM's option for tagging crosswalks should be extended to apply to ways (the path of a crosswalk across the street right-of-way) instead of just to nodes (the intersection of the crosswalk and the street).

Although some data desired for use in a multimodal trip planner could, in principle, be recorded in OSM, the difficulty or hazard in collecting the data make it unlikely that even interested members of the public could collect and record them. Among these are data on volumes of vehicle traffic, width of lanes on busy streets, and the operating characteristics of pedestrian signals at intersections. It may be possible to reduce the need for traffic volumes by developing alternative measures of bicycle and pedestrian LOS that requires only variables that the general public can measure safely and reliably. Research is needed to develop and validate such measures of LOS. Such measures are especially needed for streets that have traffic volumes and speeds between those for residential streets and for major arterial streets. With regard to pedestrian signals, the agencies responsible for installing and maintaining them should have data on how the signals operate, and this can be compiled and merged with data from OSM and other sources.

CROWD-SOURCED DATA INTEGRITY, QUALITY, AND SECURITY

Throughout the project, the research team has discussed the possible use of open data sources such as OSM for multimodal trip planners with many different communities and government and non-government entities. Several parties have expressed concern about the integrity and quality of crowd-sourced OSM data, reasoning that if anyone with an OSM account can edit the data, then anyone could register, sabotage the data, and render the trip planner useless. The project team recognizes this concern as important. However, having participated in the OSM community for slightly more than two years, the project team believes that incidents of deliberate damage to the OSM data are currently rare.

In the experience of the research team, most data corruption problems seem to arise from one of three situations:

1. Unintentional damage by manual edits from new users. A user new to OSM begins editing and, in the process, damages data that someone else has contributed. Sometimes the person realizes the problem he or she has created, and contacts others in the community to request help in correcting it. Frequently, established contributors quickly notice the problem, communicate with the new person, and try to resolve the issue (e.g., explaining what has been done wrong and how to avoid or correct it). If that fails, experienced users raise the problem with the larger community through a forum or listserv, and an uncooperative new

contributor may be blocked from further editing. The established contributors then go back and correct the damage, or the community can remove the new user's contributions and revert to an older correct version of data from the OSM database.

2. Unintentional damage by automated processes. A contributor imports new data from an outside source, or executes an automated error-checking and correction procedure, that inadvertently damages existing data. Typically, as in the first scenario, existing OSM users will quickly notice the problem and work with the offending user to correct the issue. The OSM community is divided on whether to encourage or discourage bulk imports. Most users who favor imports believe small sections should be imported at a time, and checked before proceeding with additional sections. Additionally, the community recommends that established contributors in an area should be alerted to the possibility of unexpected changes prior to the bulk import. The project team followed these practices when using `GO_Sync` to import bus stops, as discussed in Chapter 4.
3. Damage resulting from bulk removal of unauthorized copyrighted data from OSM. A user intentionally or unintentionally contributes copyrighted data to the OSM database without permission of the copyright holder. Upon discovery, the OSM Foundation will remove the data and, if it is unclear which of the user's contributions are suspect, all of the data the person has contributed. This situation also seems to be rare but, unlike the other two cases, copyright violation can go on for a period of time, and other contributors may improve the copyrighted data before the problem is discovered. The removal of the data can result in the loss of some data contributed by others. One such case occurred in Tampa during this project, and a small amount of data contributed by the research team was lost from OSM as a result. The OSM Foundation is aware that this automated removal process can create problems and says it is working on ways to reduce the issues while still maintaining the copyright-free integrity of the OSM database.

It appears that most people who contribute to OSM intend to contribute valid data. It also appears that the best defense against the inadvertent or willful corruption of data in the system is to have enough people actively contributing in an urban area that the work of new contributors gets noticed soon after it begins.

In the absence of an active, diverse community of contributors in an area, a multimodal trip planner has two key defenses against corruption of data from OSM that would affect the behavior of the system.

1. The main OSM system database is implemented to facilitate real-time editing, but not real-time routing. A practical multimodal trip planner needs to work with a copy of the OSM data running on its own server, and not with the OSM data that anyone can edit using the OSM website. Therefore, a major step in setting up OTP is to download data from OSM and to compile the data to enable routing for different modes of transportation. This in effect creates a buffer between OSM and the multimodal trip planner. Each time the multimodal trip planner is updated with data from OSM or another open data source, the new data can be compared to the old data to check what has changed. OSM data contains a tag that

indicates who has contributed the most recent version. An agency can check tags of new data to see who has made the changes and, over time, develop a list of trusted contributors whose changes might be accepted routinely, leaving changes by others to be checked before they are accepted. A transit agency can also check its own data when synchronizing with OSM using a tool such as GO_Sync, discussed in Chapter 4.

2. A transit agency participating in a multimodal trip planner retains control of its own data and, if desired, can elect to use its own version rather than OSM's. If a transit agency is using crowd-sourced improvements to its bus stop inventory, the agency can refrain from accepting any edit that it does not trust. The agency's only real obligation to OSM is to refrain from overwriting correct crowd-sourced improvements to data in the OSM database.

The project team discussed the integrity concern with the University of Maryland which, because of the newness of pedestrian trip planners, is one of the few systems relying primarily on OSM data. Their staff noted that they work with copies of the OSM data and that, over the approximately two years the system has been operational, they have not encountered any instances of willful damage to the OSM data they use.

The research team believes the factors above create a situation in which the benefits of using crowd-sourced geographic data greatly outweigh the risks. However, each agency can examine the tradeoffs between up-to-date crowd-sourced data and verifying individual user edits, and develop a policy with which they are comfortable for protecting the integrity of the data for their trip planning system.

It should be noted that data from official or other established sources is also subject to error, either from simple human mistakes, from willful fabrication, or from failure to keep it up-to-date when conditions on the ground change. These sorts of error can be difficult to detect until the data are field-checked. The advantage of an open data source, with an active community of contributors and users, is that field-checking can be frequent compared to what an agency with limited budget can accomplish.

LICENSING ISSUES

A public transportation agency that wants to contribute data to OSM must either ensure that the data to be shared (e.g., their GTFS bus stop inventory) are in the public domain, or else agree to OSM's data license. If a public transportation agency wants to download data from OSM and use it to improve its own data, it also needs to agree to OSM's license for use of the OSM data. The following paragraphs in this section outline the research team's understanding of the OSM license, based on experience and community documentation. However, all users wishing to participate in the OSM community must review the OSM license on their own terms as required when creating a new OSM account to ensure they agree with the provisions before participating in OSM.

At present, OSM uses a Creative Commons Attribution Share-Alike 2.0 (CC-BY-SA 2.0) license for its data. The intent of the license is to ensure that anyone can use OSM data without significant restrictions. The license allows a person or organization to use OSM data provided that they acknowledge OSM as the source and, if they improve the data (e.g., by correcting errors or

extending data coverage), that they make the improvements publicly available for addition back into OSM. Users are not actually responsible for uploading the improvements into OSM, but must at least make the improved data available so that someone else could do so [49]. The license also prevents a user of OSM from placing restrictions on use of the data as part of the service that user is providing. For example, a company can download data from OSM, and use it in a commercial product, and sell the product, but the company cannot prevent someone else from copying the map data in the product for their own uses. Also, the company would not be allowed to require that the 3rd party notify the company if the 3rd party plans to use the company's OSM-derived data.

In general, if a public transportation agency wishes to work with its local community and with OSM to improve its data on bus stops, in the research team's opinion the terms of the present OSM license should not present significant obstacles to doing so. Many, but not all, public transportation agencies have made their data publicly available for anyone to use. In fact, many public agencies are obligated by law to share information under the Freedom of Information Act. If an agency incorporates improvements from OSM into its database, and then wants to publish a map that uses the improved data in combination with copyrighted data acquired from another source, it would need to obtain permission from the copyright holder to allow users of the map to copy it freely. Otherwise, an agency that is willing to make its data open for anyone to use should encounter few problems in working with OSM.

The OpenStreetMap Foundation, which manages OSM, has proposed to change the present license to an Open Database License 1.0 (ODbL 1.0). The reasons given for the proposal are that the present CC-BY-SA2.0 license was designed for creative works but not for databases, and that the ODbL 1.0 license has the same attribution and share-alike features but was designed for databases. In other words, copyright law which the CC-BY-SA2.0 license depends upon, may not apply to non-creative works such as direct data observations, a definition which seems to apply to most data currently within OSM. The Open Database License, however, is not based upon copyright law, and therefore may be more appropriate for the type of data maintained within OSM. In addition, the ODbL 1.0 license would allow someone to publish a map that combines OSM data with copyrighted data without requiring that the entire map be copyright-free. Under ODbL, improvements to OSM data would still need to be made available for incorporation into OSM.

The proposed change to the OSM license has been controversial within the community and appears to have stalled. However, because it appears that the proposed new license would not place additional obligations on contributors or users of OSM data, uncertainty about if or when the license may change should not be a barrier to agencies that are interested in working with OSM. However, as mentioned earlier, it is the OSM user's responsibility to review the current OSM license that is displayed to all users when joining OpenStreetMap to ensure that they agree with all provisions of the license.

As noted in the previous section, the OSM community occasionally becomes aware that someone has contributed data from copyrighted sources without approval from the copyright holder, which violates the terms of OSM membership. When this happens, the administrators of the OSM data will block that contributor from making further contributions or changes to the OSM data, and they will remove all of that contributor's questionable contributions from the OSM database. OSM maintains

a versioning system that lets it trace back through the history of each node, way, and relation in the OSM database. Any object contributed or edited by a banned contributor is traced and reverted (i.e., rewound) back to its condition prior to when the contributor worked with it. Unfortunately, if the banned contributor deleted previous data to replace it with other data, the previous data are lost. And, if another contributor has edited data contributed by the banned contributor, those edits may also be lost. The OSM community is aware of this problem and some work is being done to limit the loss of data. However, banning and removal occur very infrequently.

License issues are somewhat different for GTFS. The GTFS format, which defines the fields and attributes for how the transit data should be organized, is licensed under the Creative Commons Attribution 3.0 License. This license essentially provides GTFS as an open format to the transit community and industry for any use. However, unlike OSM, GTFS does not define a universal license that must be applied to the data contained within the GTFS format. This situation is a result of a key difference between OSM and GTFS. OSM both defines a format for how data should be represented (i.e., the tagging system), as well as a centralized world-wide database that contains all OSM data contributed by OSM users. However, GTFS is only a format for how data should be represented, and no centralized world-wide repository for GTFS data exists. Therefore, there are no requirements by the GTFS creators for the type of license that should be applied to data represented in the GTFS format. This design is also a byproduct of how GTFS feeds are stored and maintained: each transit agency hosts their own GTFS zip file for download from a website, so the transit agency (or entity acting on behalf of the transit agency) is the ultimate keeper of their own data. Since there are no restrictions on the use of the GTFS format, agencies or companies can use the GTFS format internally without being required to publicly share the data they are representing in GTFS format. However, most agencies do choose to openly share their own GTFS data, since they see value in the service offered by the many mobile and Web applications that use the GTFS format, often at no cost to the transit agency. GTFS Data Exchange is a website dedicated to tracking the publicly accessible website links from which transit data for many different agencies can be downloaded, and also serves to provide updates to application developers when a transit agency has posted a new GTFS dataset to the agency's website [50].

CHAPTER FOUR – GTFS-OSM-SYNC – A TOOL TO SYNCHRONIZE TRANSIT DATA WITH OPENSTREETMAP

Most U.S. cities have relatively small amounts of activity in OSM, and relatively few bus stops have been mapped there. This suggests a transit agency could contribute data about its bus stops to OSM to quickly increase the coverage of U.S. transit stops in OSM. The transit agency also could benefit if it could identify which stops get edited in OSM, and then retrieve these crowd-sourced improvements to either correct or augment its transit stop data. Increasing the accuracy of bus stop geographic location data, as well as better documenting nearby amenities and attractions, can increase the quality and performance of multimodal trip planners and make transit a more attractive mode of transportation. Because the GTFS format is becoming a de facto standard, it would be useful to have a tool that could transfer data between an agency's GTFS file and OSM, in either direction.

This chapter describes the development and use of such a synchronization tool. GTFS-OSM Synchronization (GO_Sync) is a framework to synchronize mapping contributions from online communities and transit agency bus stop inventories. GO_Sync enables a transit agency to easily share its GTFS datasets with online open-data communities while tracking crowd-sourced improvements to the bus stop inventory so these contributions can be integrated with the agency's official GTFS dataset. OpenStreetMap, a Wikipedia-type source for spatial information, is used as the online free-content repository of geographic data. GO_Sync is used to compare agency GTFS data and existing transit data in the online OSM database, and to detect and present differences between the two sources. Using the application programming interface (API) provided by OSM [52], GO_Sync then uploads bus stops into the OSM server, while crowd-sourced improvements to the inventory are presented to the transit agency for processing and integration into the official bus stop inventory. Successfully translating GTFS data into the OSM format will enable over 125 transit agencies across the U.S. to share their public-domain data with the OSM community. The GO_Sync framework is made available to the transit developer community as an open-source software application to assist in the open sharing and improvement of public transportation information.

RELATED WORK

The OSM community provides several editing tools available to mapping contributors. Among those tools are Potlatch 1 and 2, JOSM, Vespucci, Mapzen, and ArcGIS editor for OpenStreetMap. Potlatch 1 is a light weight online Flash-based editor which allows users with an OSM account to add, update, or delete geographical data through an easy-to-use interface [67]. Although Potlatch 1 is currently the main editor for the OSM website, it is only ideal for quick manual editing contributions. In the same manner, Vespucci and Mapzen are also the two OSM editors for Android and iPhone. They do not support a large import of geographic data into OSM. The two most popular

editors that support a bulk upload are JOSM and ArcGIS editor for OpenStreetMap. These two editors are great tools for geographic analysts and advanced OSM users since they support editing of a locally stored dataset before uploading changes as a whole. However, neither JOSM nor ArcGIS editor for OpenStreetMap allows users to automatically analyze and display conflicting information between a large independent dataset and the OSM database when the two datasets are related. Additionally, none of the existing OSM editors focus on bus stop inventories in GTFS format. Therefore, it is currently difficult for transit agencies to either contribute their GTFS dataset to OSM or retrieve crowd-sourced improvements specific to their bus stop inventory. GO_Sync differs from the above existing OSM tools in that GO_Sync focuses on automated information analysis and synchronization between a transit agency and OSM based on the agency bus stop inventories. The targeted end-users of this framework are therefore public transit agencies, or transit information software developers working with or on behalf of public transit agencies. Using GO_Sync, they can visualize crowd-sourced improvements to bus stops, as well as view which OSM contributors are active and trusted. GO_Sync can also export bus stop data from OSM into GTFS format (i.e., comma-separated values (CSV) files) that could be re-used for other routing software or integration back into the transit agency's official stop inventory and GTFS dataset.

PROBLEM DEFINITION

The concern of this software application is primarily with the synchronization of data from GTFS format and the OSM repository. The primary spatial component of the GTFS dataset is the location and attributes of bus stops and bus stations, which are the data that GO_Sync processes. At first appearance, it seems that GO_Sync could simply do a bulk import of new GTFS data into OSM every time the transit agency generates a new GTFS dataset. However, GO_Sync cannot blindly insert the entire GTFS dataset into OSM, as it would destroy the existing crowd-sourced efforts by other OSM users.

Automated synchronization between OSM and GTFS datasets is not a trivial task. Although there is a suggested coding system of tagging bus stop data in OSM, many OSM contributors do not follow these guidelines exactly, because OSM is an open-tagging collaborative system with users with various levels of OSM experience. Furthermore, as noted in Chapter 3 [10], OSM's guidelines for tagging public transportation are not yet completely standardized. Therefore, matching existing crowd-sourced work with new GTFS data programmatically is the most challenging issue in synchronizing GTFS data and OSM data. In addition, for all executions after the very first bulk upload of GTFS data from the transit agency, GO_Sync needs to compare the current contents of OSM against the GTFS data so that GO_Sync does not duplicate or overwrite the existing OSM data.

Solving the above problems requires several steps:

1. GO_Sync must determine what has already been mapped in OSM in the area covered by the GTFS data.
2. GO_Sync must identify bus stops that may be in conflict between OSM and an agency's GTFS data.

3. GO_Sync must make decisions on how to manage the conflict, with the aid of user input.
4. GO_Sync uploads GTFS data into OSM, managing any conflict that the user cannot reconcile.

This process must be repeatable so that when the transit agency changes its bus service and generates a new GTFS dataset, the agency can repeat the above four steps and benefit from any corrections made by individuals mapping in OSM while updating the OSM data with any improvements or modifications by the transit agency to their own data.

OPENSTREETMAP DATA

OSM exposes an application programming interface (API) for fetching and saving raw data to and from the OSM PostgreSQL database. The current version in use is API v0.6 [53]. This API is primarily based on RESTful API under the form of HTTP GET, PUT, POST, and DELETE requests, using HTTP basic authentication with a username and password [52]. Furthermore, the payload is a XML document using MIME type text/xml and UTF-8 character encoding.

Two OSM data primitives widely used in this application are nodes for representing bus stops and relations for representing bus routes. A node also consists of a geospatial point defined by latitude and longitude, along with other attributes. A relation has attributes to specify affiliation among OSM objects (e.g., nodes, ways, and polygons). A relation can have zero, one, or many nodes as its members. Specifically, GO_Sync determines which bus stops relate to which routes; then, it links all bus stops to corresponding OSM relations. Bus routes, therefore, are defined as properties of bus stops.

Since OSM is a collaborative mapping project, the data are created by different users from various backgrounds such as professional mappers, developers, and students. Therefore, OSM employs an open tagging system, which does not enforce a standard set of fixed tags. However, OSM does provide guidance for best-practices on tagging and recommended tags. Applications that use OSM data must take this open tagging system into account and recognize that tags with different names can have the same meaning. For instance, when mapping a bus stop, the bus route could be defined using a tag named *route*, *routes*, *route_id*, while the recommended tag is in fact, *route_ref*. [54]. This problem becomes more challenging as GO_Sync must consider bus stops coded with multiple operators (i.e., transit agencies) at a single stop (e.g., a transfer station between two different transit agencies). The recommended tagging practice for OSM users and multiple operators is to input the key name as operator and use a semicolon as the delimiter to separate different operator names in the corresponding value field, but this practice is not always followed.

POTENTIAL BUS STOP CONFLICTS

Keeping a bus stop inventory accurate and update-to-date is a tremendous effort for transit agencies. In other words, a bus stop inventory not only needs to have correct information about bus stop locations and attributes, but this information must also be maintained so that it always reflects the condition of the real world. This problem is made worse with jurisdictional overlaps, as agencies with other jurisdictions (e.g., departments of transportation, universities, etc.) may move

bus stops without the agency’s knowledge during construction. Hence, agencies are not always able to keep a centralized database with current positions of bus stops.

The authors of this paper found several conflicts between the existing OSM stops and the GTFS file of Hillsborough Area Regional Transit (HART) in the Tampa, Florida, area. In the context of GO_Sync, the term “conflict” refers to an OSM stop with a geographic location that does not exactly match the location of a GTFS bus stop, or a stop that exists in the OSM database but does not have an associated GTFS ID (The GTFS ID is used to track the stop once it is imported from a GTFS file). In other words, a bus stop conflict involves at least two distinct stops nodes that may represent the same logical bus stop. Figure 13 shows a set of conflicting bus stops along 50th Street in Tampa, Florida, USA.

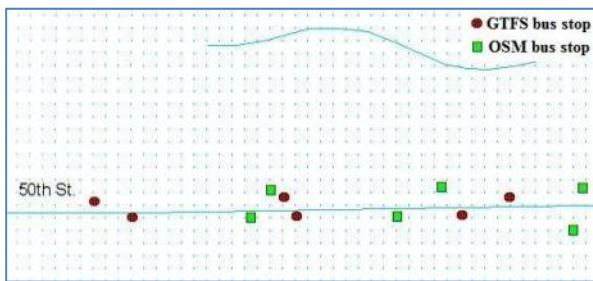


Figure 13 - 50th Street bus stop conflicts



Figure 14 – Holly Drive bus stop conflicts

The red circles represent GTFS stops while the green squares symbolize existing stops in OSM. In this case and the ones that follow, field-checking showed that the OSM data have greater position accuracy than the GTFS data. Another conflict is captured on Holy Drive in Tampa, Florida, USA where GTFS data shows a shift in location compared to OSM stops (Figure 14). Sometimes, stops that have been discontinued by the transit agency are still included in GTFS datasets (Figure 15).

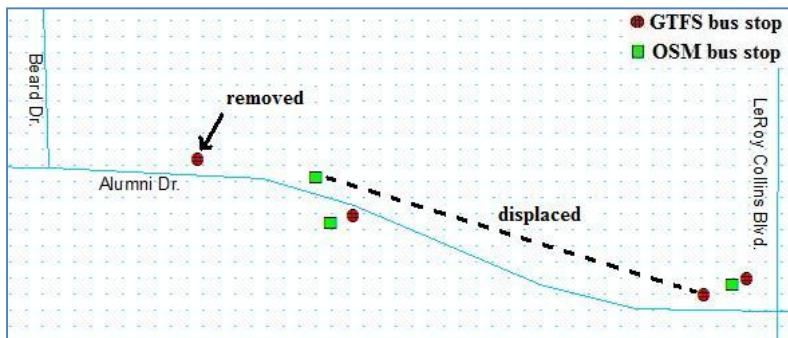


Figure 15 – Alumni Drive bus stop conflicts

The stop with the “removed” label in Figure 15 still exists in the GTFS dataset but no longer exists. Figure 15 also shows a stop that was displaced due to nearby construction.

After the analysis of various conflicts with transit stops data in the University of South Florida area, a 400-meter threshold was chosen to associate potentially conflicting GTFS stops and OSM data. In other words, if there is an OSM stop within 400 meters of a GTFS stop, these stops need to be compared since they may represent the same logical stop.

As GO_Sync is a Java desktop application, a geospatial database is not involved. Therefore, the geodesic distances between nodes, defined as the shortest path along the ellipsoid of the earth at sea level, must be calculated within the application. By definition, the geodesic distance is the shortest path along the ellipsoid of the earth at sea level between two geographic coordinates. Over the years, a significant amount of work has been done to derive formulas that compute geodesic distance of two points [55][56][57][58]. The Vincenty inverse solution produces distance measurements that are accurate within 0.5mm [59].

Vincenty's formula [57] follows:

a, b = major and minor semi axes of the ellipsoid (meters)

f , flattening = $(a-b)/a$

φ_1, φ_2 = geodesic latitude

L = difference in longitude

U_1 , reduced latitude, defined by $\tan U_1 = (1-f)\tan\varphi_1$

U_2 , reduced latitude, defined by $\tan U_2 = (1-f)\tan\varphi_2$

$$\lambda = L \quad (1)$$

$$\sin^2 \sigma = (\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda)^2 \quad (2)$$

$$\cos \sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos \lambda \quad (3)$$

$$\tan \sigma = \frac{\sin \sigma}{\cos \sigma} \quad (4)$$

$$\sin \alpha = \cos U_1 \cos U_2 \frac{\sin \lambda}{\sin \sigma} \quad (5)$$

$$\cos 2\sigma_m = \cos \sigma - \frac{2 \sin U_1 \sin U_2}{\cos^2 \alpha} \quad (6)$$

$$C = \frac{f}{16} \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)] \quad (7)$$

$$L = \lambda - (1 - C) f \sin \alpha \{ \sigma + C \sin \sigma [\cos 2\sigma_m + C \cos \sigma (-1 + 2 \cos^2 2\sigma_m)] \} \quad (8)$$

The procedure starts from Equation (2) through (6), and then a new λ is obtained from Equations (7) and (8). The iteration repeats until the change in λ is negligible. Afterward, the distance s is obtained. Distance is then computed between two coordinates on the Earth surface:

$$u^2 = \cos^2 \alpha \frac{(a^2 - b^2)}{b^2} \quad (9)$$

$$A = 1 + \frac{u^2}{16384} \{ 4096 + u^2 [-768 + u^2 (320 - 175u^2)] \} \quad (10)$$

$$B = \frac{u^2}{1024} \{ 256 + u^2 [-128 + u^2 (74 - 47u^2)] \} \quad (11)$$

$$\Delta \sigma = B \sin \sigma \{ \cos 2\sigma_m + \frac{B}{4} [\cos \sigma (-1 + 2 \cos^2 2\sigma_m) - \frac{B}{6} \cos 2\sigma_m (-3 + 4 \sin^2 \sigma) (-3 + 4 \cos^2 2\sigma_m)] \} \quad (12)$$

$$s = bA(\sigma - \Delta \sigma) \quad (13)$$

The WGS 84 reference system is used to represent the earth in Global Positioning Systems [60]. Vincenty's formula parameters are therefore configured for WGS 84 accordingly with $a=6378137$; $b=6356752.3142$; $f=1/298.257223563$.

FRAMEWORK DESIGN

Two difficulties in constructing a communication tool between a transit agency and an online community are the unpredictable nature of tagging in OSM and the high potential for conflicts between GTFS dataset bus stops and existing bus stop information within OSM. Since GTFS is a

well-designed data model that identifies bus stops with unique IDs for each agency, the difficulties are thought to be easily overcome by removing all the conflicting OSM nodes and inserting new stops from the GTFS dataset. However, that method would violate the basic principle of open-source software and the social mapping model: respect others' works. Therefore, the framework must not damage crowd-sourced contributions to public transportation data in OSM. Instead, the framework should supplement the existing OSM data with the transit data from the GTFS dataset, while properly resolving potential conflicts between the two groups of data. Additionally, the framework should present crowd-sourced improvements of transit data to the transit agency so the agency can improve their official bus stop inventory.

The GO_Sync framework starts by processing a new GTFS dataset. From the *stops.txt*, *trips.txt*, *routes.txt*, and *stops_times.txt* files, all agency's stops and associated routes are read into the software tool. Next, the bus stops and routes that have been previously coded in OSM must be retrieved by GO_Sync. GO_Sync searches the input GTFS dataset to find the maximum and minimum of both latitude and longitude among all the bus stop coordinates. A bounding box is created using those values. To reduce the payload size downloaded from the server, GO_Sync only retrieves OSM primitive nodes coded with tag *[highway=bus_stop]* and relations coded with tag *[route=bus]*. Since the main API does not support a straightforward method to query data from OSM server, GO_Sync uses the read-only extended API (XAPI). The response from XAPI server consists of all bus stops and routes that satisfied our criteria formatted as a XML document. After processing this response stream, all OSM bus stops and routes in the described bounding box, along with their associated tags, are collected into corresponding lists within the software application for further calculations. The following sections discuss the functionality that is implemented within the GO_Sync software using these lists, including the comparisons of GTFS and OSM bus stops and routes, resolution of tagging differences, and the upload of a dataset with resolved conflicts into OSM.

BUS STOP COMPARISONS

Bus stops in OSM that are in close proximity to stops in GTFS datasets are potentially duplicates of the GTFS stops; conversely, GTFS datasets that are in close proximity to stops in OSM are potentially duplicates of the OSM stops. To determine and resolve conflicts, the GTFS and OSM stops must be compared.

The GO_Sync framework first retrieves OSM bus stops which have the operator name of the transit agency that has been specified by the user (e.g., Hillsborough Area Regional Transit). GO_Sync also retrieves all OSM bus stops where the operator field is null, or the operator field is missing, since these stops may also represent stops belonging to the same agency. Bus stops are identified by their operator names and the *gtfs_id* fields. In some cases where the operator names and *gtfs_id* fields are not present in OSM, GO_Sync uses the geographic latitude and longitude to make distinctions among stops. A bus stop object in GO_Sync is instantiated with an operator name, bus stop ID from GTFS (i.e., *gtfs_id*), latitude, longitude, and additional data. In the case of a stop from OSM that does not have a *gtfs_id*, the field is recorded as 'none.' When inserting into OSM, bus stops may also include other useful information from GTFS data such as stop code, description of a stop, fare zone for a stop, etc. The transit agencies can also receive extra information from OSM users such as the existence of bus shelters, benches, lighting facilities, as well as if maintenance is needed at specific

bus stops. In general, the goal of the GO_Sync framework is to build a communication bridge between formatted GTFS data and the OSM online community by categorizing bus stops into four different groups:

- New GTFS stops having no conflict with stops in OSM
- New GTFS stops having possible conflicts with stops in OSM
- GTFS stops which already exist in OSM but have updated information from the agency, and
- GTFS stops which are identical to OSM stops.

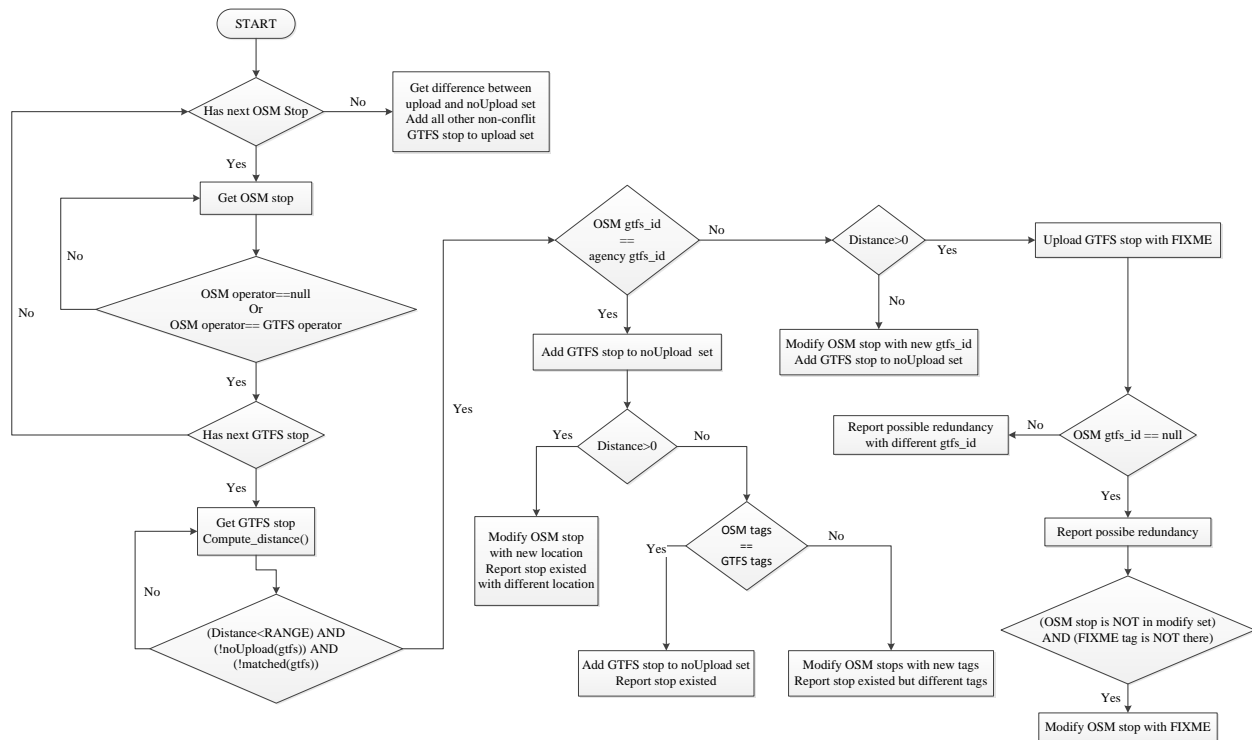


Figure 16 – The matching algorithm used by GO_Sync for GTFS and OSM datasets

Items in the GTFS data and OSM lists are compared in pairs to determine which of the four above groups they belong to (Figure 16). A *FIXME* tag is also added to the stop to notify other OSM users about the update to the stop. One of the main parameters in this algorithm is the distance used to verify redundancy between the new stops and existing ones in the OSM server. Currently, that constant is set to 400 meters, because errors of this size have been observed in existing GTFS data. However, this value can be easily changed if other values are found to be more appropriate. Once the GO_Sync algorithm finishes the classification stage, a graphical user interface report is displayed to allow the end-user to review what data are going to be uploaded, as well as the conflicts (i.e., inconsistencies) between their provided GTFS input and existing OSM data (Figure 17). Using this interface, the user can choose to edit or update any necessary tags, or accept the existing values.

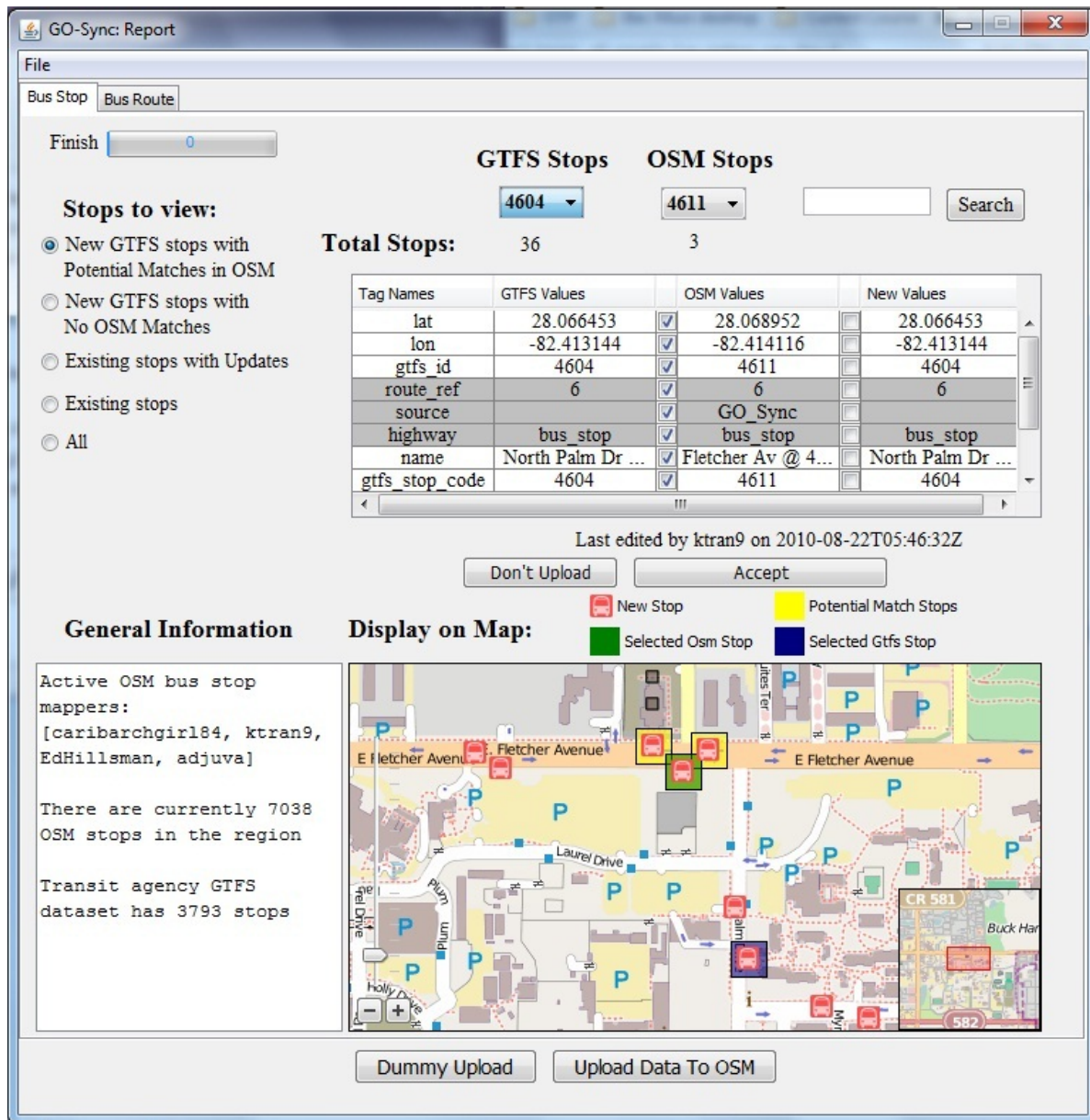


Figure 17 – The user interface of the GO_Sync Framework which shows the differences between General Transit Feed Specification datasets and bus stops in OpenStreetMap

BUS STOP TAG COMPARISONS

In OSM, a tag is a 2-tuple (i.e., key-value, or ordered, pair) in which the first element is often called the key and the second element is regarded as the value. OSM does not place any restrictions on tags but does not allow duplicate keys in the same node, way, or relation. Tags are also widely used by OSM users to label nodes and provide supplementary data for corresponding objects. GO_Sync takes this social mapping model into account and determines which tags are new, which tags already exist in the OSM server, and also detects any changes in tag values. GO_Sync performs these comparisons with existing OSM tags and adds the record attribute information included in GTFS input as OSM tags.

BUS ROUTE COMPARISONS

Transit agency bus route data are generated through the use of OSM relations. In practice, an OSM relation can only be linked with existing objects. Therefore, of the four total bus stop groups, GO_Sync takes into account bus routes from only two bus stop groups: stops which exist in OSM but have updated information from the agency, and stops which are identical between the two datasets. Those bus stops have been previously coded in the server and, therefore, have their own OSM ID.

Bus routes are identified by their operator names and route_ref fields (i.e., route number). Unlike bus stops, when the route_ref field is empty, GO_Sync simply creates a new relation without any further considerations.

The process begins with constructing new route objects for all bus stops that already have an OSM ID. All stops are linked as relation's members accordingly. After getting OSM relation data from the specified bounding box, the algorithm compares all routes to detect if any relations need to be modified. This method is very similar to the tag comparison routine but with one extra evaluation for bus stop members. After this procedure is complete, all the routes are categorized into three different groups: new, modified, and routes that are already in OSM and do not have any new updates from the agency.

MODIFICATIONS TO OSM DATA

BULK UPLOAD

When the GO_Sync framework finishes classifying the four sets and the application receives the upload command from the end-user, it is ready to apply modifications to the OSM server. While GO_Sync retrieves OSM bus stops data from the XAPI server, the application uploads data to OSM using the main API database. Appropriate data are written into an output stream to the OSM API database as follows:

- **Create:** new bus stops for all elements in the two sets '*new GTFS stops having no conflicts*', and '*new GTFS stops with possible conflicts*' are created. In addition, new bus routes for all objects in the new route set are created. Appropriate tags and corresponding values are also added to the OSM server.
- **Modify:** Stops for all elements in the set '*GTFS stops which already exist in OSM but have updated information from the agency*' are modified. Analogously, all routes in the modified set are modified. Proper tags are also inserted for each Stop and Route object.
- **No action:** No action is taken for the stops and routes that are identical between the two datasets.

The data flow diagram for the relationship between GO_Sync and OSM is presented in Figure 18.

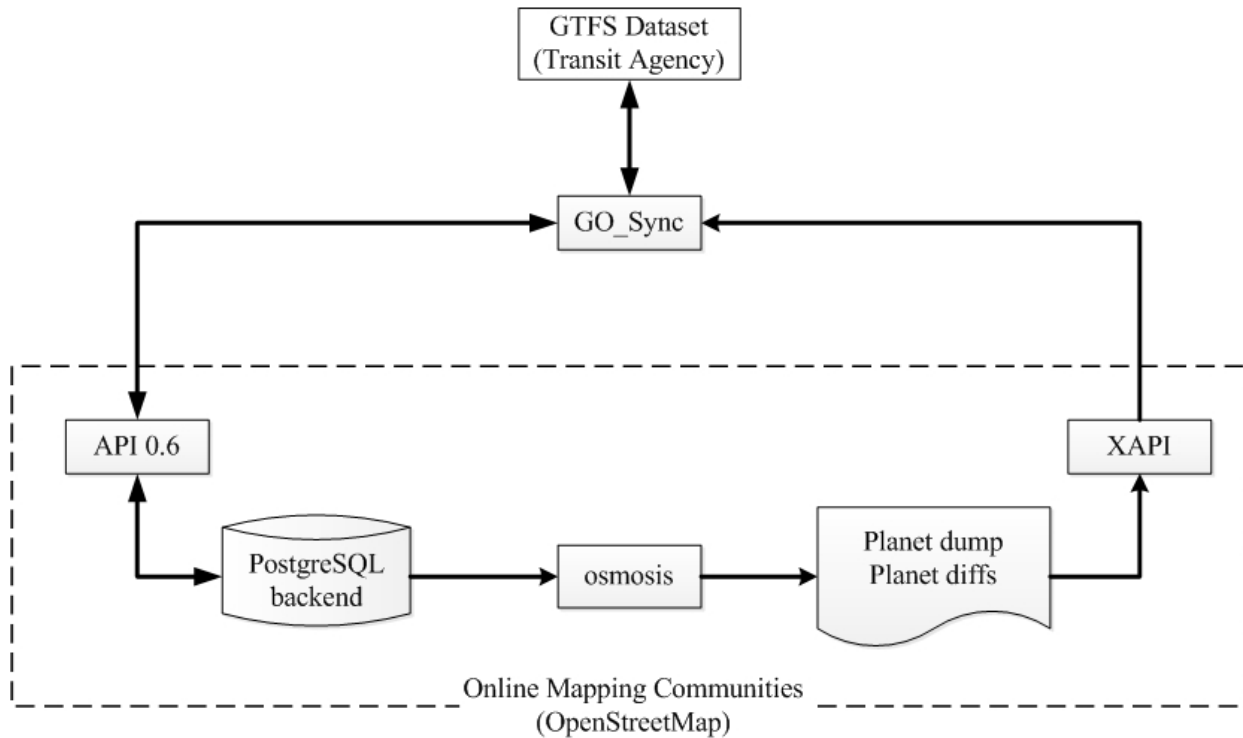


Figure 18 – The GO_Sync Data Flow Diagram

Any edits in OSM are done within a changeset. Therefore, a new changeset must be created before importing any data into the OSM server. Additionally, uploading large datasets into the OSM server requires that the data be in OsmChange file format, which is used by Osmosis, a command line Java application for processing OSM data, to describe differences between two dumps of OSM data [61][62]. In the context of GO_Sync, once the bus stop record is successfully uploaded to the OSM PostgreSQL database, Osmosis would apply the changes into the planet dump, which performs an update every minute from the database. In fact, planet dump is the raw data for many OSM services, including the XAPI. Consequently, there is a time-lag resulting in differences between PostgreSQL database and XAPI database. Thus, after uploading the new data to OSM, a GO_Sync user should wait a short period of time before running GO_Sync again to ensure that enough time is given for the updates to appear in OSM.

REVERT CHANGESET

As mentioned earlier, a changeset is an OSM object which contains all user activities when they interact with the server for uploading new dataset. Occasionally, a GO_Sync user may make mistakes in importing stops into OSM. As a result, the user may want to reset all changes that they have made into earlier states. If no changes have been made by other users to the same data, the changeset is fresh. However, if other users have edited the data, the changeset is considered dirty.

GO_Sync provides users a convenient way for correcting their mistakes. Initially, users are asked to input a changeset id that they want to revert. Based on the changeset ID, all changes that have been made in that specific changeset are obtained. There are three possible actions that could have

occurred within a changeset, which GO_Sync can reverse. These three actions, and the corresponding reversal, are:

- **Create:** GO_Sync simply deletes that bus stop.
- **Modify:** GO_Sync obtains the OSM ID and the current version of that specific node. Then, the application retrieves the previous state of the node. Next, GO_Sync modifies the node with corresponding information. The three different attributes from its previous state after the reversion are the creator name, version number, and timestamp value.
- **Delete:** Similar to the modify process. However, instead of modifying nodes, GO_Sync creates a new node on the server with the retrieved data. Again, the creator name, version number, and timestamp value cannot be restored.

Two limitations of this implementation are that the revert only takes place on bus stops (e.g., changeset created by GO_Sync) and the changeset must be 'fresh.'

SAMPLE EXECUTION

The main end-user of this application is the transit agency, or transit software developers working on behalf of the transit agency. Public transportation schedules and associated geographic information are provided with the GTFS format. The user first needs to sign up for a free OSM account. When GO_Sync is started, the software asks the end-user for their OSM username and password, as well as a description to include with the changeset when it is uploaded to OSM. The user then chooses whether he or she wants to analyze bus stop data or revert a changeset.

SCENARIO 1 – ANALYZE BUS STOP DATA

Comparing bus data requires the user to enter the agency full name, possible alias, GTFS source file, and the operator ID from the National Transit Database. When the agency finishes inputting information, the application defines a bounding box that includes all new GTFS bus stop locations. GO_Sync then downloads OSM bus stops and routes in the described bounding box and stores them into a set of stop objects and a set of route objects accordingly. Next, the transit stop comparison algorithm executes to provide information about which bus stops and associated properties should be uploaded. During this process, the tag comparison algorithm classifies the nodes. When the stop comparison algorithm finishes, the route comparison is then invoked. Then, all stops and routes are categorized in their corresponding groups. A report is generated for the end-user to review before new or modified data are to be inserted into the OSM database. The report is an interactive graphical user interface (GUI). The GUI helps users to manage bus stop inventories using an online map. Users can remove bus stops, edit and add tags, as well as re-classify stops and routes into different groups. When a user finishes editing the dataset, they can continue to upload their data into OSM.

SCENARIO 2 – REVERT A CHANGESET

This task requires only one input field, which is the changeset ID. After the changeset ID is filled in, GO_Sync downloads the entire changeset. It then processes the XML data and categorizes the data

into *create*, *modify*, and *delete* sets. For every created node, GO_Sync deletes the nodes while nodes in other two sets need to be retrieved one by one. The time to process all the nodes depends primarily on the number of elements in the *modify* and *delete* sets.

EVALUATION

The GO_Sync application was installed in a personal computer with an Intel Core 2 Duo CPU T5750, 4GB RAM, and a 64-bit operating system. Since GO_Sync requires a Java Virtual Machine, the latest Java Runtime Environment (JRE) 6 was also installed in the same computer [63]. The authors obtained permission from Hillsborough Area Regional Transit Authority (HART), located in Tampa, Florida, USA, to upload their GTFS data into OSM.

Before the GO_Sync tool was executed using HART data, little public transportation information existed in the Tampa, Florida, area even though HART serves nearly 1.2 million people [66]. Only 133 HART bus stops were available in OSM prior to running GO_Sync (Figure 19a). After the authors received HART's approval to use its GTFS data, in July 2010 GO_Sync uploaded 3812 new bus stops to OSM (Figure 19b). Since OSM originally started in Europe in August 2004, it wasn't until September 2007 that the Topologically Integrated Geographic Encoding and Referencing system (TIGER) data for U.S. streets, produced by the U.S. Census Bureau, was uploaded to OSM [65]. Therefore, OSM is still relatively young as a major geographic data repository in the U.S., and many other U.S. cities also likely lack OSM public transportation information.

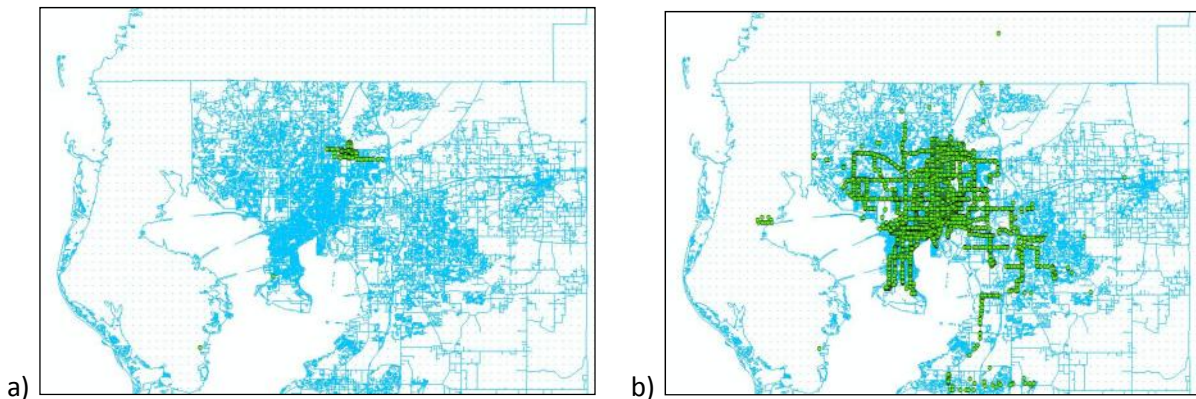


Figure 19 – OSM bus stop data for Hillsborough County before using GO_Sync (a), and after using GO_Sync (b), showing an increase of 3679 bus stops

To determine the activity of OSM users as related to the uploaded public transportation data, the authors analyzed the OSM transit data for the Tampa area again in January 2011. Between July 2010 and January 2011, there were modifications to the geographic location of 173 bus stops by OSM users, including four unique OSM users other than the authors of this report. Figure 20 shows the distribution histogram of the distance that bus stops were moved by OSM users.

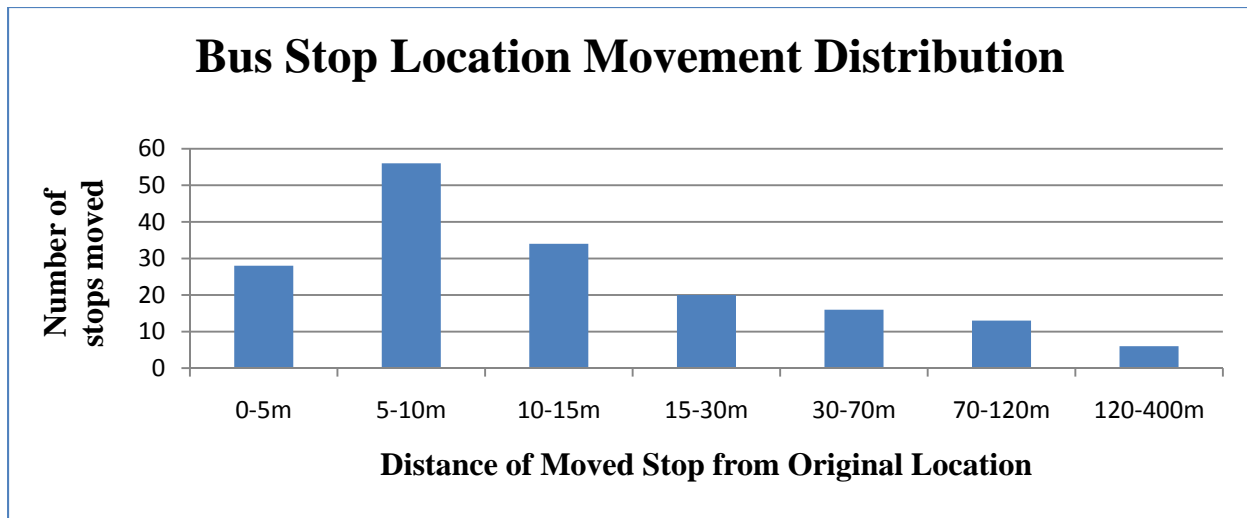


Figure 20 – OpenStreetMap users have modified the geographic location of 173 bus stops in the Tampa, Florida, area

The authors reviewed a subset of the modified bus stop locations and confirmed that many of the moved stops in OSM were indeed corrections to the initial inaccurate geographic location of the bus stop as it was represented in the GTFS dataset. The authors also confirmed with a representative of HART that the GTFS bus stop latitudes and longitudes are approximated and would not necessarily reflect the true geographic position of the stop.

Based on these findings, transit agencies should be able to benefit from these crowd-sourced modifications in order to improve a bus stop inventory without significant cost to the agency. GO_Sync assists transit agencies in this task by generating a report for the transit agencies which identifies the bus stops that have been modified by OSM users. Ideally, active OSM users will continue to update information about local stops so that the transit agency would be able to periodically run GO_Sync to upload new GTFS datasets, and in the process receive additional improvements from OSM users to bus stop locations and amenities. GO_Sync is designed according to a social mapping model to ease the data management and maintenance efforts of a transit agency for their transit data.

GO_Sync can be downloaded by developers and users at <http://code.google.com/p/gtfs-osm-sync/> [68]. The research team has been tracking the use of the site hosting the GO_Sync tool using Google Analytics, and there has been significant interest in GO_Sync. From September 1, 2010 to March 7, 2011, there have been 798 visits to the site from 192 unique visitors from 29 different countries, for a total of 3091 page views. The research team hopes that the OSM, OTP, and transit developer community will work to improve the open-source GO_Sync tool in the future based on input from all.

RELEVANCE TO OTHER DATASETS

Using GO_Sync, other transit agencies can use their GTFS datasets as input to import bus stop inventories into OSM. At the same time, the agencies can also benefit from crowd-sourced

contributions in improving data quality and managing their bus stop inventories. In addition to the HART data as previously mentioned, the research team has gained permissions from four other transit agencies in Florida to use their GTFS datasets and test them with GO_Sync. The following sections discuss the work with several agencies' GTFS datasets, and a final section discusses how the GO_Sync model might be applied to other types of geographic datasets so those datasets can be imported into OSM.

SARASOTA COUNTY AREA TRANSIT (SCAT)

While preparing to upload the Sarasota dataset, the research team noticed that many of the stops appeared twice in the GTFS stops file. Although the duplicate stops did not affect the execution of GO_Sync to import the data into OSM, the duplicate stops were coded on top of each other, using the same latitude and longitude. Therefore, if OSM users wanted to move the stops from one place to another, they must do it twice, or figure out which stop to delete. Moreover, when OSM users edit bus stops, they would not know about the existence of the same stop underneath unless they moved the one on top.

Because the upload would introduce obviously erroneous data into OSM, and because correcting it would create confusion among OSM contributors, the research team decided not to upload SCAT's stops data until the team could work with SCAT to understand and correct the problem in their data. It appears that the duplication issue is the result of the use of some of tools used to post-process GTFS data after it is exported from Trapeze, perhaps when using the merge tool. However, the research team was not able to spend enough time to identify the exact issue. The issue remains unresolved as of February 2011, but the research team expects that the problem could be resolved with some additional effort.

PINELLAS SUNCOAST TRANSIT AUTHORITY (PSTA)

The research team gained permission from PSTA to upload PSTA's data into OSM in August 2010. In September 2010, the team uploaded PSTA's GTFS dataset into OSM. Before the imports, there were only 2 PSTA stops previously coded in OSM. GO_Sync then uploaded 5333 PSTA stops into OSM. To determine the activity of OSM users in the region, the team analyzed the data again in February 2011. Figure 21 shows the results of OSM edits to the locations of PSTA stops.

SPACE COAST AREA TRANSIT (SCAT)

In December 2010, shortly after Space Coast Area Transit authorized the team to upload their GTFS data into OSM, the team uploaded their bus stop inventories into OSM using GO_Sync. There were no bus stops in OSM for the Space Coast region before the upload, and GO_Sync added 654 new stops to OSM. In February 2011, the team examined the OSM activity and the result is presented in Figure 22:

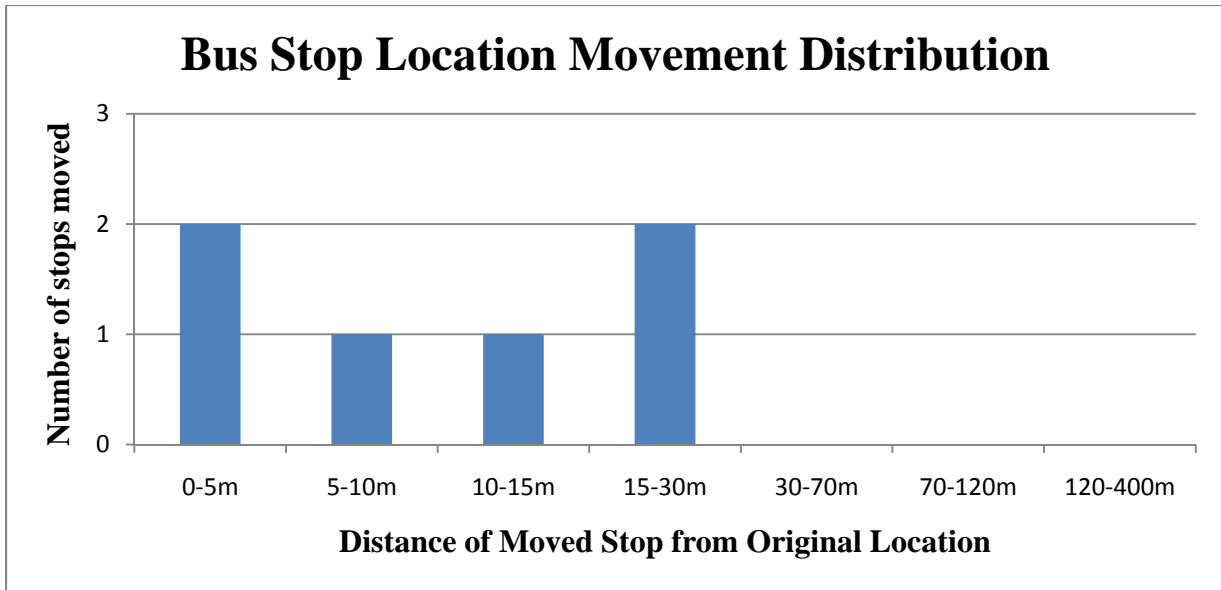


Figure 21 – OpenStreetMap users have modified the geographic location of 6 bus stops in the Pinellas Suncoast Transit Area coverage

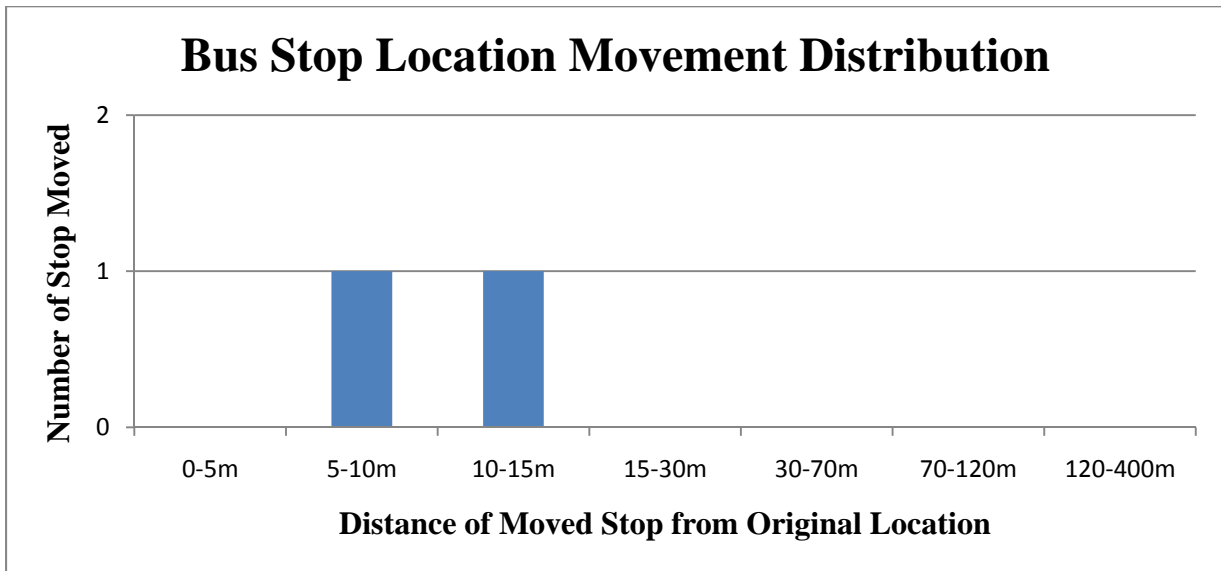


Figure 22 – OpenStreetMap users have modified the geographic location of 2 bus stops in the Space Coast Area Transit coverage

MIAMI-DADE TRANSIT (MDT)

The research team gained permission to upload Miami-Dade Transit (MDT) GTFS data into OSM in December 2011, and proceeded to upload their data using GO_Sync. Prior to the import, there were 42 MDT stops coded in OSM. GO_Sync added 8404 new stops. OSM activity on MDT bus stop inventories after the initial import to February 2011 is shown in Figure 23:

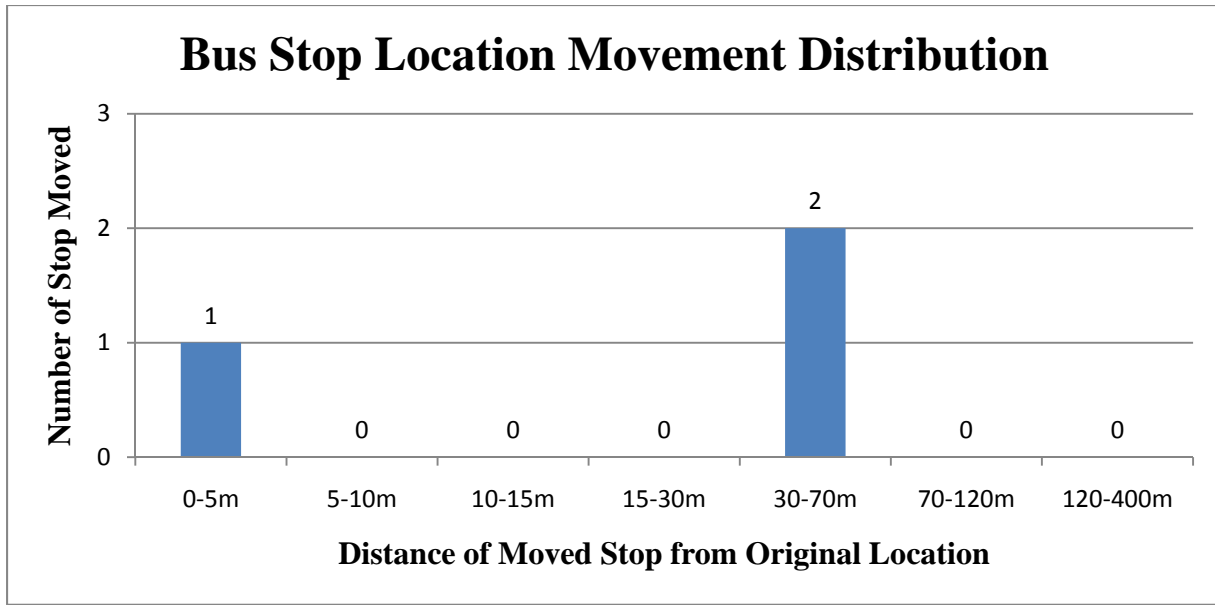


Figure 23 – OpenStreetMap users have modified the geographic location of 3 bus stops in the Miami-Dade Transit coverage

FDOT AND OTHER GIS DATASETS

The general approach that GO_Sync uses to import point or node data such as bus stops into OSM, and to synchronize the external and OSM versions, could be used in principle to import and synchronize line or way data as well (although parts of the code would have to be modified to work with line data). This methodology could be attractive for importing data from FDOT’s RCI files, or from city or county datasets. However, importing and synchronizing line data runs a much greater risk of damaging or interfering with existing data than does working with point data such as bus stops. There are several reasons for this, including:

- Each location in the bus stop point data stands alone. However, in line data, each line can intersect with other line data, and in the case of streets it usually does. Hence, inserting a new line may require changing other lines. For this reason, inserting point data into line data (e.g., uploading traffic signals, which need to be coded at the node where two streets intersect) also would be more complex than uploading bus stop data.
- The length of a line is arbitrary, and depends on where characteristics of the line change along its path. Recording a change in speed limit or in the number of lanes along the path of a street requires breaking the line and tagging each piece with the characteristics that apply on each side of the change. Synchronizing the lines requires synchronizing the pieces and their respective intersections with other lines. Doing this manually along a single street can be a complex task involving decisions about exactly where to break the line and to realign its path, and therefore automating this process would be extremely difficult.
- Lines are much more likely than points to be parts of OSM relations. Bus routes need to be represented in the OSM database using relations, because routes often turn from one street

to another. As discussed in Chapter 3, the use of relations tends to lead to the division of street records into smaller segments. In addition, because of the need to divide existing street data into segments, creating and maintaining the accuracy of relations and the objects they contain is a more complex process than creating and maintaining data for a simple section of street that is not contained in a relation.

Synchronizing linear data requires figuring out where characteristics change, where streets have already been segmented and, when streets have not yet been segmented but need to be, where to split the streets. Performing these tasks requires more effort and decision-making than matching one node (point) with another, even manually for a single way. Therefore, attempting to fully automate this process could be disastrous and may end up damaging OSM data. Because OSM is a community of people who contribute data, it is important to avoid damaging data that others have already contributed. Writing computer code to add linear features to the OSM database is fairly simple. Writing code to perform other steps—intersecting the new linear features with other linear features, in the proper locations, or matching them to existing features, splitting existing features, and retagging the portions—is more difficult, because it is difficult to anticipate all of the ways in which the code could damage the previous contributions.

In an urban area where there are few contributors to OSM and where few changes have been made to the OSM data, a simple import of linear data may be feasible, such as the initial import of the U.S Census Bureau's 2007 TIGER street network files. However, such an import should proceed in small increments, with manual checking at each stage to make sure that the import is doing no damage, and with some field checking to make sure that the data being imported are indeed accurate. Based on the complexity involved, and the risk of damaging data that others have already contributed, the research team recommends against continually synchronizing linear OSM data with linear data from outside OSM, especially in a fully automated method.

However, while the full automation of importing linear GIS data into OSM is discouraged by the research team, there is some progress being made towards efficiently adding linear datasets to OSM manually. As mentioned earlier in this report, TriMet has also recently developed some useful methodologies for leveraging existing data sources, such as ArcGIS shapefiles for road centerlines, as part of their effort to expand OSM data in Portland [43]. These techniques appear to be useful to many different areas and agencies that want to quickly increase the amount of OSM data locally, although the methodology does require ESRI ArcGIS software as part of the data analysis process. The results of these efforts in Portland may provide an efficient model to other areas that would like to perform a similar process. However, it should be noted that it appears that TriMet's model still applies to one-time imports of line feature data, not repeated updates to continuously synchronize line feature datasets with OSM.

GO_SYNC SUMMARY

Open data sets such as the General Transit Feed Specification (GTFS) and OpenStreetMap (OSM) are transforming the landscape of information systems for public transportation and are leading to innovations in mapping and location-based services. While transit agencies struggle to keep large bus stop inventory datasets up-to-date, the OSM community has thousands of users that are willing

to contribute improvements to geographic data. However, in the U.S. there is currently little public transportation information in OSM. The GO_Sync framework presented in this paper is able to synchronize large GTFS datasets from transit agencies with crowd-sourced OSM data. GO_Sync is therefore able to contribute large amounts of public transportation data to OSM while helping transit agencies maintain and update their inventories by leveraging crowd-sourced improvements to the data.

A test deployment in Tampa, Florida, has yielded improvements to 173 bus stop locations by OSM users. Data for several other bus stops have been improved as well for different datasets from Pinellas Suncoast Transit Area, Space Coast Area Transit, and Miami-Dade Transit. There are several possible explanations for the small number of OSM edits in these three regions compared to the edits for HART. First, the HART stop file was uploaded in July 2010 and the others were uploaded later, with the stops for Space Coast and Miami-Dade uploaded in December 2010. In addition, research team members discussed the project with USF students who were enrolled in a class on GPS technology; some of these students checked and corrected the locations of some HART stops as part of a class project. In addition, a member of the research team had entered a number of HART stops into OSM prior to the upload. A spot-check of several of the PSTA bus stops suggests that the PSTA file of bus stops had much greater geographic accuracy than the HART file. Based on our experience during the initial mapping of an area for OSM, large errors are more likely to be noticed and corrected than small ones, which may be corrected in subsequent visits.

The project team has made GO_Sync available to the open-source transit developer community as an open-source application, which is available at <http://code.google.com/p/gtfs-osm-sync/>. Future research will focus on further evaluation of this tool, additional improvements to better fit transit agency's needs, as well as determining the ideal distance used to identify conflicting GTFS and OSM bus stops.

CHAPTER FIVE - CONCLUSIONS

This chapter of the report outlines a summary of the project findings, and presents some open research issues in the area of multimodal trip planning.

SUMMARY OF PROJECT FINDINGS

Several major findings of the project are discussed in the following subsections.

FEASIBILITY

It is very feasible to implement a multimodal trip planner using open-source software and open sources of data. Development of the OpenTripPlanner software has made tremendous strides since the beginning of this research project. OTP has most of the major features desired of a multimodal trip planner and has an active community that is continuing to add features and improve the software [36]. In addition, this research project has demonstrated that the OTP software can be modified to provide additional features, sometimes with surprisingly little effort. The research team also demonstrated the ability to include new transit systems in an OpenTripPlanner instance as soon as an agency's data is available in GTFS format, without having to wait for an outside organization (e.g., Google Transit) to include them in its service offering. Installing the OTP software requires technical expertise, although the research team expects that this burden will diminish over time for basic installations as additional documentation is created. Modifying the software also requires software development expertise. A non-profit company, OpenPlans.org, offers technical support for installing and customizing the OTP software for a transit agency or other local organization without in-house software development expertise that wants to provide a trip planner. OpenPlans has indicated that a basic transit-only implementation may be setup for approximately \$30,000 [39]. TriMet, the public transportation agency in Portland, Oregon, is using a customized version of OTP as a transit trip planner on its website and is using OpenPlans for enterprise-level support.

USE OF GTFS DATA

The research team recommends that multimodal trip planning rely on transit data in GTFS format for scheduling and routing. Key information required for transit trip planning is the route schedule, and OSM was not designed to work with schedule data, for any mode. In addition, OSM still does not have clear procedures for recording route information, and the procedures presently in use are complex and require levels of time and attention to technical detail that make them difficult for the average contributor. The GTFS format is an open specification, and OpenTripPlanner has been designed to use GTFS data as a source for transit stops, routes, schedules, and other transit information. Data from other sources, such as OSM or commercial sources, are used by OpenTripPlanner for other modes. Because GTFS is open, anyone can use the format to create

transit datasets that enable OpenTripPlanner to support multimodal trip planning in an area with transit service.

Transit agencies are the primary managers of transit data, including the locations of bus stops. Agencies that provide GTFS data to Google for their Google Transit service already have an incentive to keep their GTFS data files current. Agencies that partner with others to develop multimodal trip planners for their service areas should have a similar incentive. Agencies can use the GO-Sync software, developed as part of this project, to keep the bus stops current in both GTFS and OSM, so that the stops can be used to link data from different sources based on the geographic location and stop identifiers (e.g., GTFS stop ID).

DATA COVERAGE FROM OTHER SOURCES

Open data sources using formats like GTFS and datasets such as OpenStreetMap are designed to collect much of the data needed to support multimodal trip planning. However, data coverage remains incomplete. Many public transportation agencies have yet to provide data about their stops, transit routes, and schedules into the GTFS format, which is the emerging standard for providing this information. Transportation agencies tend to collect and maintain data primarily for those facilities they are responsible for (i.e., major facilities), and more minor facilities may not be covered in detail. Open data repositories such as OSM cover a wider range of facilities, but collection depends on the willingness and interest of individuals and community organizations to collect and record the data. As a result, geographic coverage in the OSM database is spotty, particularly in the U.S. TriMet in Portland has hired a staff member and interns to improve and add to OSM data in its service area. TriMet's experience in this effort should be valuable to other communities considering multimodal trip planners, and the research team recommends tracking it.

The research team has briefed staff at the Hillsborough County MPO about experience with OTP, and this has led to discussion about how to collect more of the data needed to support a multimodal trip planner in the county.

SUGGESTED EXTENSIONS TO OSM TAGGING SYSTEM

The OSM tagging system generally supports most of the data needed in multimodal trip planning, but some changes to the tagging system would improve its usefulness for this purpose. Additional options are needed to describe how bicycle lanes are placed at intersections, in order to advise bicyclists what to expect as they approach an intersection that is unfamiliar to them. In addition, some cycling facilities such as wide paved shoulders, wide outside lanes, and lanes designated with sharrows also lack tagging options in OSM. However, the bicycling community needs to first agree on what needs to be communicated to bicycle users of trip planners, and on terminology for describing conditions to persons who may not be expert cyclists. Some additional research is needed to support defining a more inclusive tagging system for bicycle infrastructure before making proposals specific for biking to the OSM community.

However, the project team has begun the process of proposing the following changes to the OSM tagging system for other multimodal data attributes:

- Changes to the way that crosswalks are tagged, to make it easier to identify these ways and retrieve data about the crossing from the node where it is stored (The research team backed an existing the OSM community proposal that includes tags to addresses this problem, which has now been approved [30]).
- A tag to indicate information about the space, if any, between a sidewalk and a parallel street (this has been delayed by discussion of the proposal that included the change to how crosswalks are tagged); and
- A tag to indicate whether facilities—especially for walking, bicycling, or waiting for transit—have shade during summer months.

If adopted, the recommended changes would be relatively easy for OSM contributors to use when coding data. OSM proposals tend to fail if they are complicated, so the relative simplicity of the proposed changes should improve their prospects for adoption.

While several other data attributes desired for use in a multimodal trip planner could, in principle, be recorded in OSM, the difficulty or hazard in collecting the data make it unlikely that even interested members of the public could collect and record them. Among these are data on volumes of vehicle traffic, width of lanes on busy streets, and the operating characteristics of pedestrian signals at intersections. It may be possible to reduce the need for traffic volumes by developing alternative measures of bicycle and pedestrian LOS that requires only variables that the general public can measure safely and reliably. Research is needed to develop and validate such measures of LOS. Such measures are especially needed for streets that have traffic volumes and speeds between those for residential streets and for major arterial streets. With regard to pedestrian signals, the agencies responsible for installing and maintaining pedestrian signals should have data on how the signals operate, and this data can be compiled and merged with data from OSM and other sources.

SYNCHRONIZATION OF DATA BETWEEN OSM AND EXTERNAL SOURCES

Synchronizing data between OSM and other data sources, such as GTFS, is feasible for point (i.e., node) data such as bus stops, but it must be done carefully to avoid damaging data that others have previously contributed to OSM. Synchronizing line (i.e., way) data between OSM and other sources is not practical. The GO_Sync software developed as part of this project demonstrated that point data can be synchronized, and that this can be done fairly easily in an urban area for feature types, such as bus stops, that have not previously been entered into OSM for the area. The synchronization allows external sources of data, such as transit agencies, to avail themselves of corrections and supplemental data recorded by OSM contributors, and it enables the OSM community to benefit by receiving updated data when organizations responsible for an activity make changes in it. In areas where individual contributors have already entered substantial amounts of data about the feature, initial steps may require more effort to avoid conflicts between OSM and external data.

Synchronization of line (i.e., way) data is much more difficult than it is for point data. Characteristics of a street change along its length, and the line representing the street needs to be

split in OSM to reflect each change. Relations, a type of tool that OSM uses to group line segments into long routes (such as numbered state highways, or individual bus routes), are not well understood by many beginning contributors to OSM and are easily damaged. In addition, OSM does not have agreement on a single way to represent bus routes in the database. Each of these characteristics makes line synchronization difficult and error-prone, and the research team recommends against the automated synchronization of line data between OSM and external sources, especially if the external data are expected to change frequently.

OPEN CHALLENGES IN MULTIMODAL TRIP PLANNING

This research has identified two major challenges in multimodal trip planning: data, and communication. Additional research is needed on each of these topics, and the facets of these challenges are discussed in greater detail in the following subsections.

DATA

In most urban areas, much of the sidewalk, bicycling, and amenity data needed for multimodal trip planning has not yet been collected, recorded and compiled. Developing the data is labor-intensive, and conventional approaches, such as using agency staff or hiring contractors, are expensive. Crowd-sourcing has the potential of being much less expensive, provided that enough people are interested in volunteering time and effort to observe and record data, and to check each other's work.

OpenStreetMap and other crowd-sourced information projects depend upon *cognitive surplus*—the free time of an educated, Internet-connected citizenry. The phenomenon of focusing this cognitive surplus toward specific objectives is relatively new. There have been too few successful examples of focusing cognitive surplus—and too little research on them—to predict which efforts will draw large numbers of participants, or to offer guidance on how to motivate participation in any particular effort. Participation depends on the interplay among motivation, opportunity, and the way that activity is implemented. The fact that present participants in OSM tend to be male, young, and well-educated might be the result of an inherent attraction of such people to mapping, but it also could be an effect of how the present OSM project is organized, operated, and marketed.

Additional research is needed to identify, develop and document what approaches are effective in motivating volunteers to help collect and record data about transportation infrastructure in their communities. Research is also needed to identify which groups within the general population are most amenable to participating in such projects.

We believe it should be possible to engage neighborhood and other community organizations in gathering and reporting data on transportation infrastructure that could be used in multimodal trip planning (e.g., sidewalks, crosswalks, shelters at bus stops). Success will likely be increased if citizens can be shown that making such an effort will benefit them directly. Also, if the data collection can be organized in ways that make it easy for participants to observe and record data without requiring each person to do the work of recording the data in OSM or other data repository, the OSM learning curve could be largely avoided by the majority of data collectors in the group.

Members of the research team are seeking opportunities to work with communities that are interested in mapping their transportation infrastructure, in order to better understand this process and the motivations for participation. Team members have identified possible leads in the Sarasota and Tampa areas.

TriMet in Portland is attempting a middle course between conventional approaches and crowd-sourcing. They have hired a staff person and interns to improve portions of the OSM data in the Portland area, even though the area appears to have more participation in OSM than most U.S. cities. TriMet's experience should provide useful lessons, and should be monitored in order to learn from them.

Any trip planning project using OSM or other crowd-sourced data will provide an opportunity to assess the actual risks and advantages of using such data, including the ongoing integrity of the information as it is modified by the OSM community. Agencies implementing such trip planners will need to find a balance between up-to-date crowd-source data and the effort required to verify data before it is added to the trip planner.

COMMUNICATION

Internet-based trip planners for car travel are well-established and in common use. They provide information to assist a behavior (i.e., driving) that most of the U.S. population is familiar with and that occurs on infrastructure that is relatively standard. In other words, car trip planners provide information for a relatively uniform activity, to users who have relatively uniform needs. However, trip planners for other modes of travel are less common. They provide information about behaviors (e.g., walking, taking the bus, bicycling) that many people in many cities do routinely perform, and about services and infrastructure that are not uniform within an urban area or between urban areas. To be successful, a multimodal trip planner needs to help users, who may be unfamiliar with an area, cope with this lack of uniformity.

Different people have different perceptions of what kinds of streets, sidewalks, crosswalks, and intersections are safe for bicycling and walking. If one of the purposes of building a multimodal trip planner is to encourage more people to walk, bicycle, or ride transit, then the trip planner needs to use objective data in its calculations and in presenting its results, but its design also needs to be sensitive to different perceptions of risk and comfort. Additional research is needed to know what information needs to be presented to different groups of users by default, and what information needs to be available if requested. Additional research is also needed on the best ways to display route information from Internet-based trip planners, both to the general user population and to those users who may prefer to see certain types of information on a map of the route.

These problems are particularly relevant for bicycling, which occupies an ambiguous middle position between being treated as motor vehicle traffic in some regards but not in others. For example, potential cyclists may want to see information on speed or volume of motor traffic along a route. There seem to be no commonly accepted practices regarding the types of information to be displayed on maps for bicyclists, or for how information should be displayed.

Additional research is also needed on how subjective ratings of routes or facilities might be combined with objective data to help users choose a route, or to encourage them to try a mode they typically do not use.

People tend to use two different skill sets when navigating through a street or sidewalk network. One skill is reading and interpreting a map. Another is following a detailed set of instructions. Trip planners for car travel provide information to support each of these skills, and multimodal trip planners also need to do so. In general, street names are useful and adequate for providing step-by-step instructions, including for walking on sidewalks that parallel the streets. However, additional research is needed on how best to provide step-by-step instructions for walking or cycling where there are no parallel streets to use as references, as in parks and on university campuses. GPS-enabled smartphones may be one tool for providing these instructions in real time using first-person point-of-view with arrow overlays, but such phones are not yet in universal use, and the use of GPS hardware can be a significant drain on phone batteries. The ability of modern consumer-grade GPS-enabled cell phones, and new embedded sensors, to provide significantly accurate real-time position updates that could be used for real-time pedestrian navigation instructions, both outdoors and indoors, should also be examined.

NEXT STEPS

Under this project, the research team created a demonstration instance of OpenTripPlanner (<http://opentripplanner.usf.edu/>) and evaluated its ability to plan routes using open multimodal datasets such as GTFS and OSM. During this process, the research team noted several improvements that could facilitate pedestrian routing at intersections and crosswalks, and have proposed new OSM tags to the OSM community to address these issues.

The next step to extend the research in this project is to create a pilot deployment of OpenTripPlanner for a Florida city or county that is willing to work with the research team on tasks necessary to create and publicly promote use of an OpenTripPlanner website. While any city could use OpenTripPlanner, OpenTripPlanner requires a GTFS dataset for transit routing. Therefore, the research team recommends the selection of a city with a transit agency that already has its data in the GTFS format, or an agency that is willing to create a GTFS dataset for use with OpenTripPlanner. To the research team's knowledge, the Florida transit agencies that have their data in the GTFS format include Hillsborough Area Regional Transit (HART), Sarasota County Area Transit (SCAT), Pinellas Suncoast Transit Authority (PSTA), Space Coast Area Transit (SCAT) in Brevard County, Miami-Dade Transit (MDT), Broward County Transit (BCT), and Regional Transit System (RTS) in Gainesville. Initial pilot deployments of OpenTripPlanner should also be targeted at areas with existing pedestrian infrastructure and traffic, such as downtowns, parks, or university campuses, to best explore a wide range of trip planner use.

A pilot deployment of OpenTripPlanner would be very valuable as a base for conducting research on the topics mentioned in the above sections (including modifications to the pedestrian routing functions) and as a working example that will help generate interest by other communities in such systems. As mentioned previously, staff members of the Hillsborough County MPO have expressed

interest to the project team in implementing a demonstration site in partnership with HART. Other CUTR staff members have heard similar interest from Pinellas County's MPO.

Members of the research team intend to continue their involvement with OSM and OpenTripPlanner, and to continue to seek opportunities to fund the implementation of local pilot instances of OpenTripPlanner.

REFERENCES

- [1] Federal Transit Administration (FTA) Intelligent Transportation Systems Program's 2009-2029 Strategic Plan, 2009 APTA Annual Meeting, Seattle, WA. May 6th, 2009.
- [2] Goroo: RTA, CTA, Metra, Pace, Driving and Biking Directions, <http://goroo.com/goroo/index.htm>, [Accessed March 23, 2011].
- [3] Google, Inc. "Add Your Transit Data to Google Maps," Internet: <http://maps.google.com/help/maps/transit/partners/participate.html>, [Accessed February 16, 2011].
- [4] "General Transit Feed Specification – Google Code." Internet: http://code.google.com/transit/spec/transit_feed_specification.html, January 11, 2010 [Accessed July 27, 2010].
- [5] Hendricks, B. "Bing Maps Gets Transit Directions," September 16, 2010. [Accessed February 14, 2011].
- [6] Ferris, B., K. Watkins, and A. Borning. Location-Aware Tools for Improving Public Transit Usability. IEEE Pervasive Computing: Location-Based Services, Vol. 9, No. 1 Jan-March 2010.
- [7] Jariyasunant, J. et al. "Mobile Transit Trip Planning with Real-time Data," Proceedings of the National Academy of Sciences' Transportation Research Board 90th Annual Meeting, January 2011.
- [8] Steinfeld, A. et al. "Mobile Transit Rider Information Via Universal Design and Crowdsourcing," Proceedings of the National Academy of Sciences' Transportation Research Board 90th Annual Meeting, January 2011.
- [9] Barbeau, S., M. A. Labrador, N. L. Georggi, P. L. Winters, R. A. Perez. "The Travel Assistance Device: Utilizing GPS-enabled Mobile Phones to Aid Transit Riders with Special Needs," Institution of Engineering and Technology (IET) Intelligent Transportation Systems, 2010, Vol. 4, Issue 1, pp. 12–23. doi: 10.1049/iet-its.2009.0028. © The Institution of Engineering and Technology 2010.
- [10] Ramm, F., Topf, J., Chilton, S. (2010). *OpenStreetMap: Using and Enhancing the Free Map of the World*. Cambridge, UIT Cambridge Ltd.
- [11] OpenStreetMap wiki, http://wiki.openstreetmap.org/wiki/Map_Features, November 23, 2010. [March 23, 2011].
- [12] OpenTripPlanner wiki, <http://opentripplanner.org/> [Accessed March 23, 2011].
- [13] Free Software Foundation. GNU Lesser General Public License. <http://www.gnu.org/licenses/lgpl.html>, [Accessed March 23, 2011]
- [14] Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". *Numerische Mathematik* 1: 269–271.
- [15] Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics* SSC4 4 (2): 100–107. doi:10.1109/TSSC.1968.300136.
- [16] A-Train, <http://www.trip.atltransit.com/>, [Accessed March 1, 2011].
- [17] BiKE Broward, <http://bikebroward.fiu.edu/mpobike/>, [Accessed March 1, 2011].
- [18] Sprinkle Consulting, 2007. Bicycle Level of Service: Applied Model, Tampa, Florida, <http://www.dot.state.fl.us/planning/systems/sm/los/pdfs/bikelosmod.pdf>, [Accessed March 1, 2011].
- [19] Landis, B. W. et al., 2001. "Modeling the Roadside Walking Environment: A Pedestrian Level of Service," Paper 01-0511 presented at Transportation Research Board annual meeting, January,

- 2001, <http://www.dot.state.fl.us/planning/systems/sm/los/pdfs/pedlosmod.pdf>, [Accessed March 1, 2011].
- [20] Larco, N., 2010, *Overlooked Density: Re-thinking Transportation Options in Suburbia*, Report OTREC-RR-10-03, Portland, OR, Oregon Transportation Research and Education Consortium, <http://www.otrec.us/project/152>, [Accessed March 1, 2011].
- [21] Datz, A., Florida Department of Transportation Public Transportation Office, April 5, 2010 telephone communication.
- [22] General Transit Feed Specification Changes, <http://sites.google.com/site/gtfschanges/open-proposals>, [Accessed March 20, 2011].
- [23] Catalá, M. "Expanding the Google Transit Data Feed Specification to Support Operations and Planning," Florida Department of Transportation and National Center for Transit Research. Grant Number BDK85 977-15. <http://www.transitgis.org/research/general-transit-feed-specification/>, [Accessed March 1, 2011].
- [24] OtherGTFSTools, <http://code.google.com/p/googletransitdatafeed/wiki/OtherGTFSTools>, [Accessed March 15, 2011].
- [25] Landis, B. W., et al., 2003. "Intersection Level of Service for the Bicycle Through Movement," TRB paper No. 03-3292, *Transportation Research Record* No. 1828, <http://trb.metapress.com/content/r6t63r2g723661u8/fulltext.pdf>, November 1, 2010. [Accessed March 10, 2011].
- [26] OpenStreetMap wiki, <http://wiki.openstreetmap.org/wiki/Bicycle>, [Accessed March 1, 2011].
- [27] OpenStreetMap wiki, <http://wiki.openstreetmap.org/wiki/Toll>, [Accessed March 1, 2011].
- [28] Geller, R., no date. *Four Types of Cyclists*. City of Portland Office of Transportation, <http://www.portlandonline.com/transportation/index.cfm?&a=237507&c=44597>. [Accessed March 23, 2011].
- [29] Florida Department of Transportation, 2009. Transportation Statistics RCI Field Handbook, Tallahassee, FL. <http://www.dot.state.fl.us/planning/statistics/rci/fieldhandbook/ch3.pdf>, [Accessed March 1, 2011].
- [30] OpenStreetMap wiki, http://wiki.openstreetmap.org/wiki/Proposed_features/Sidewalk_as_separate_way, [Accessed May 3, 2011].
- [31] OpenStreetMap wiki, <http://wiki.openstreetmap.org/wiki/Key:crossing>, [Accessed March 1, 2011].
- [32] U.S. Access Board, <http://www.access-board.gov/adaag/html/adaag.htm>, [Accessed March 1, 2011].
- [33] U.S. Geological Survey, National Elevation Database, <http://ned.usgs.gov/>, [Accessed March 1, 2011].
- [34] Geisberger, R. "Contraction Hierarchies: Faster and Simpler Hierarchical Routing in Road Networks," Diploma Thesis. http://algo2.iti.kit.edu/documents/routeplanning/geisberger_dipl.pdf [Accessed July 1, 2008].
- [35] Merrifield, T. "Heuristic Route Search in Public Transportation Networks," Master's Thesis. University of Illinois at Chicago, 2010. https://www.cs.uic.edu/pub/Bits/TransitGenieDocs/tm_thesis.pdf, [Accessed March 23, 2011].
- [36] OpenTripPlanner Developer Group, <http://groups.google.com/group/opentripplanner-dev>, [Accessed March 23, 2011].
- [37] OpenTripPlanner User Group, <http://groups.google.com/group/opentripplanner-users>, [Accessed March 23, 2011].

- [38]Transit Developer Group, <http://groups.google.com/group/transit-developers>, [Accessed March 23, 2011].
- [39]Open Trip Planner 1-year Anniversary Update Webinar, July 21, 2010.
- [40]Hilkevitch, J., “RTA hopes travelers go along with Goroo,” Internet: http://articles.chicagotribune.com/2009-05-15/news/0905140725_1_travel-preferences-commuters-toll-roads/3, May 15, 2009 [Accessed February 16, 2011].
- [41]TriMet Online Trip Planner, <http://ride.trimet.org/>, [Accessed March 23, 2011].
- [42]OpenStreetMap wiki, http://wiki.openstreetmap.org/wiki/JOSM/Plugins/public_transport, [Accessed March 23, 2011].
- [43]McHugh, B., “Open Street Map Improvement Project – RLIS Partner and Regional Street Centerline Meeting,” <http://www.slideshare.net/bibianamchugh/osm-improvements> [Accessed March 2, 2011].
- [44]Haklay, M. and Budhathoki, N. “OpenStreetMap—Overview and Motivational Factors”, presentation at University of Nottingham, March 19, 2010, <https://www.ideals.illinois.edu/bitstream/handle/2142/16461/Horizon%20March%202010%20%28Haklay%20and%20Budhahtoki%29.pdf?sequence=2>; and personal communication from Budhatoki, N. February 23, 2011 to clarify information presented in this document.
- [45]LaVallee, A., 2009. “Only 13% of Wikipedia Contributors Are Women, Study Says” *The Wall Street Journal*, August 31, 2009, <http://blogs.wsj.com/digits/2009/08/31/only-13-of-wikipedia-contributors-are-women-study-says/>, [Accessed March 23, 2011].
- [46] MapQuest blog, <http://blog.mapquest.com/2010/07/09/mapquest-opens-up/>, [Accessed March 23, 2011].
- [47]Shirkey, C., 2010. *Cognitive Surplus: Creativity and Generosity in a Connected Age*. New York: Penguin.
- [48]OpenStreetMap wiki, <http://wiki.openstreetmap.org/wiki/Mapquest> , [Accessed March 23, 2011].
- [49]OpenStreetMap Foundation wiki, <http://www.osmfoundation.org/wiki/License>, [Accessed March 23, 2011].
- [50]GTFS Data Exchange, <http://www.gtfs-data-exchange.com/> , [Accessed March 23, 2011].
- [51]Mahmoud, Q.H., (2004, March). J2ME and Location-based Services [Online]. Available: <http://developers.sun.com/mobility/apis/articles/location>, [Accessed March 23, 2011].
- [52]“API v0.6 – OpenStreetMap Wiki.” Internet: http://wiki.openstreetmap.org/wiki/API_v0.6 , June 1, 2010 [Accessed July 27, 2010].
- [53]“API – OpenStreetMap Wiki.” Internet: <http://wiki.openstreetmap.org/wiki/API>, June 24, 2010 [Accessed July 27, 2010].
- [54]“Tag: highway=bus_stop – OpenStreetMap Wiki.” Internet: http://wiki.openstreetmap.org/wiki/Tag:highway%3Dbus_stop, July 23, 2010 [Accessed July 27, 2010].
- [55]Rainsford, H. F. (1955). “Long geodesics on the ellipsoid.” *Bull. Geod.*, 37, 12–21.
- [56]Kivioja, L. A. (1971). “Computation of geodetic direct and indirect problems by computers accumulating increments from geodetic line elements.” *Bull. Geod.*, 99, 55–63.
- [57]Vincenty, T. (1975). “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations.” *Surv. Rev.*, XXII (176), 88–93, http://www.ngs.noaa.gov/PUBS_LIB/inverse.pdf, [Accessed March 23, 2011].

- [58] Bowring, B. R. (1981). "The direct and inverse problems for short geodesic lines on the ellipsoid." *Surveying and Mapping*, 41(2), 135–141.
- [59] Thomas, C.M. and Featherstone, W.E., "Validation of Vincenty's Formulas for the Geodesic Using a New Fourth-Order Extension of Kivioja's Formula," *Journal of Surveying Engineering ASCE*, February 2005. [Online]. Available: <http://www.cage.curtin.edu.au/~will/thomas-featherstone.pdf>, [Accessed March 23, 2011].
- [60] "World Geodetic System." Internet: <https://www1.nga.mil/productsservices/GeodesyGeophysics/WorldGeodeticSystem/Pages/default.aspx>, August 4, 2008 [Accessed July 27, 2010].
- [61] "OsmChange – OpenStreetMap Wiki." Internet: <http://wiki.openstreetmap.org/wiki/OsmChange>, May 5, 2010 [Accessed July 27, 2010].
- [62] "Osmosis – OpenStreetMap Wiki." Internet: <http://wiki.openstreetmap.org/wiki/Osmosis>, June 26, 2010 [Accessed July 27, 2010].
- [63] "Java SE 6 Documentation." Internet: <http://www.oracle.com/technetwork/java/javase/documentation/index.html>, [Accessed February 27, 2011].
- [64] "HART – Hillsborough Area Regional Transit." Internet: <http://www.gohart.org/> [Accessed July 27, 2010].
- [65] "History – OpenStreetMap Wiki." Internet: <http://wiki.openstreetmap.org/wiki/History>, July 5, 2010 [Accessed July 27, 2010].
- [66] "Hillsborough County QuickFacts from the U.S. Census Bureau." Internet: <http://quickfacts.census.gov/qfd/states/12/12057.html>, April 22, 2010 [Accessed July 27, 2010].
- [67] "Potlatch – OpenStreetMap Wiki." Internet: <http://wiki.openstreetmap.org/wiki/Potlatch>, December 6, 2010, [Accessed January 20, 2010].
- [68] "gtfs-osm-sync – General Transit Feed Specification (GTFS) OpenStreetMap SYNChronization (GO_SYNC)", Internet: <http://code.google.com/p/gtfs-osm-sync/>, January 28, 2011 [Accessed January 28, 2011].