Metro 2009-2011 Regional Travel Options Grant FINAL REPORT

The OpenTripPlanner Project

OpenTripPlanner (OTP) is an Open Source, Multi-Modal Trip Planning System (OS MMTPS) collaboratively developed by a team of developers from across the world and coordinated by OpenPlans and TriMet.

The OTP Project was initially funded through the Metro 2009-2011 Regional Travel Options (RTO) Grant.

Project Sites:
opentripplanner.org
opentripplanner.com

Prepared by:
Bibiana McHugh, TriMet
Information Technology Manager of GIS and Location-Based Systems

Publication Date:
8/31/11

Drive less.
Save more.
www.DriveLessSaveMore.com

TRIMET
See where it takes you.

Table of Contents

List of Figures 4

Acronyms and Abbreviations 5

OSM Glossary 6

Acknowledgements 7

EXECUTIVE SUMMARY 8

Background 8

OTP Metro RTO Grant 2009-2011 Primary Goals 9

Core Software Functionality 10

Preliminary Trip Testing 11

Next Steps 12

SECTION 1: Background 14

Transit Only and Single-Mode Trip Planners 14

TriMet Open Source, Interactive System Map 15

Multi-Modal Trip Planners 17

Open Source Software Considerations 17

Open Source Approach Selected for OTP 19

SECTION 2: Project Overview 20

Kick-Off Workshop July 2009 20

Project Management Plan September 2009 22

Project Scope and Goals 22

Project Plan and Milestones 23

The Open Elements of the OpenTripPlanner 25

Open Source Development Method 25

Open Source Software 26

Open Architecture 26

Open Data 27

Implementation of Prototypes February 2010 28

Year One Anniversary July 2010 29

Progress in Year Two 30

SECTION 3: Developing a Routable Network in OpenStreetMap 32

Analyzing Routable Street Network Alternatives for OTP 32

OpenStreetMap Improvement Process 34

Geocoding Issues 35

OpenStreetMap Conclusions 36

SECTION 4: OTP Trip Evaluation Studies 38

Preliminary Transit Trip Testing in January 2011 38

Preliminary Bicycle Trip Testing June 2011 38

Preliminary Bicycle-to-Transit Trip Testing June 2011 41

SECTION 5: Next Steps 43

Metro RTO Grant Phase II 43

OTP Metro RTO Grant 2011-2013 Target Goals 43

July 2011 Kick-Off Workshop 44

OTP Fall 2011 Public Beta Launch 44

Continued OSM Improvements 45

Appendices 46

Appendix A – High Level Open Source, Multi-Modal Trip Planning System Requirements 46

Appendix B – Map of Test Area for OpenTripPlanner 48

Appendix C –OTP Bicycle Safety Weights 49

Appendix D – Detailed Procedures for Portland OSM Improvement Project 50

Part 1 – Groundwork and Preparation 50

Part 2 – Evaluations and Results 51

Part 3 – Detailed Editing Procedures 57

Part 4 – Quality Control Process 61

List of Figures

Figure 1: Comparison of multi-modal trip planner (OTP Bike-to-Transit Trips) results with single-mode trip planner (Google Transit Trips and TriMet's Proprietary Transit Trip Planner (PTTP)) results - demonstrates OTP can save significant travel time. 12

Figure 2: Planned trips per month on TriMet Trip Planner. 14

Figure 3: TriMet's Interactive System Map at maps.trimet.org uses open source technology. 16

Figure 4: TriMet's Interactive System Map displays planned transit trips generated from TriMet's Proprietary Transit Trip Planner (PTTP). 16

Figure 5: Not all open source software is the same, and not all proprietary software is the same. It is important to research the alternatives and fully define the requirements. 18

Figure 6: Photos from the July 2009 OTP Kick-Off Workshop 21

Figure 7: Architectural Diagram of OTP 24

Figure 8: Collaborative method of tracking work and progress at opentripplanner.org. 26

Figure 9: Open architecture of OTP offers ease of customization. 27

Figure 10: By default, OTP uses open source datasets (GTFS, NED, and OSM) to build a routable graph. Customization is viable for other datasets and types. 28

Figure 11: opentripplanner.com was released for commercial support options. 29

Figure 12: OSM edit session in JOSM using jurisdictional linework and aerial photography as reference. 35

Figure 13: Comparison of address location methods: geocoding on theoretical address ranges vs. an address search engine such as Solr. 36

Figure 14: Screen capture of a preview of the OTP October 2011 public beta release with new features including a bike elevation chart, Rideshare Locations tab, and multi-modal functionality. 45

Acronyms and Abbreviations

API	Application Programming Interface
CCGIS	Clark County GIS (shapefiles for Clark County)
GTFS	General Transit Feed Specification
IT	Information Technology
NED	USGS National Elevation Dataset
OS MMTPS	Open Source, Multi-Modal Trip Planning System, early reference to OTP
OSM	OpenStreetMap
OSS	Open Source Software
OTP	OpenTripPlanner, the new open source, multi-modal trip planner TriMet will release publicly in October 2011. OTP will replace TriMet's current transit trip planner, which is proprietary software.
PTTP	Proprietary Transit Trip Planner, which refers to TriMet's existing trip planner in this document
RLIS	Regional Land Information Systems (Metro datasets for Washington, Multnomah, and Clackamas Counties)

OSM Glossary

OSM Term	Definition
Node (point)	A spatial point/object with coordinates and ID
Way (line)	An ordered list of nodes representing a line
Area (polyon)	Closed ways that represent an area
Relation	Groups of nodes, ways
Tag	Attribute of a node, way, or relation
Change Set	Edit session
Sandbox Zone	Test instance of OSM data for testing and editing
OSM Editing Tools	JOSM, Potlatch, Merkaartor, Mapzen, Quantum, Osmosis, ESRI ArcMap OSM Editor

Acknowledgements

TriMet would like to thank the developers from a diversity of organizations and companies that have contributed source to the OpenTripPlanner over the last two years. They include:

Andrew Byrd
Brandon Martin-Anderson
Brian Ferris
Colin Zwiebel
David Emory
David Turner
Francisco José Peñarrubia
Frank Purcell
Laurent Gregoire
Nicholas Bergson-Shilcock
Robert Marianski
Vivien Deparday
Zsombor Welker

We would also like to thank the Portland State University Student Interns who worked so arduously on the OpenStreetMap (OSM) Improvement Project. Their contributions to that project and to this document are numerous and we are deeply indebted to their hard work and smarts.

Betsy Breyer
Grant Humphries
Melelani Sax-Barnett
PJ Houser

Metro graciously provided us with the funding for both projects: the OTP Project, and the OSM Improvement Project. Without their support, these projects would not have been possible. In particular we would like to thank Dan Kaempff, Mark Bosworth, and many others at Metro, and at the local jurisdictions who provided their data and collaborated with us on the project.

Special thanks to Jack Newlevant for his help with the local bike network and Paul Cone for his support and many contributions.

And finally, special thanks to the OSM community for their support, feedback, and continued contributions.

EXECUTIVE SUMMARY

This report documents TriMet's and OpenPlans' recent and successful efforts over the last two years to create a new open source, multi-modal trip planner (OpenTripPlanner) for TriMet, as well as for other agencies.   It also describes concurrent efforts to enhance OpenStreetMap (OSM) to be used by the OpenTripPlanner (OTP).

In conclusion, project requirements were met and exceeded. TriMet plans for an official public release of OTP in October 2011. Test results comparing OTP with other trip planners indicate an advanced product in many aspects.

With a secondary grant award, TriMet and OpenPlans have started work on a second phase, which includes additional functionality for call-takers in TriMet's Customer Service Department.

Background

TriMet's older text-based trip planner is a proprietary transit trip planner, providing transit trip itineraries with limited walking instructions; however, customers frequently request the ability to plan multi-modal trips, such as the ability to bike to transit. Similarly, Google Maps is capable of planning several different single-mode trips (transit or driving or walking or biking), however, it cannot plan a trip consisting of multiple modes.

The ability to plan **multi-modal** trips, a combination of transit, walking, and biking, and to bring as much information as possible into one central location would allow commuters to make informed decisions about their transportation choices while encouraging sustainable modes and decreasing the number of drive-alone vehicle trips. Commercial off-the-shelf solutions are cost prohibitive[1] for TriMet and many US transit agencies, and open source, multi-modal trip planners, such GraphServer and A-Train, are relatively new to the market and lack a strong user and developer base and a sustainable support model.

TriMet was interested in bringing interested parties together to collaborate on an open source multi-modal trip planning system that would not only meet TriMet's increasing needs, but could potentially be a viable alternative for other agencies. TriMet partnered with OpenPlans, a non-profit organization that specializes in open source software and civic projects, and several other developers, on applying for a Metro Regional Travel Options (RTO) Grant in 2008 to begin work on an open source, multi-modal trip planner, which came to be known as the OpenTripPlanner. TriMet was subsequently awarded $68,960 and matched this with a 50% in-kind contribution for a two-year project total cost of $137,920.

[1] Chicago's goroo.com, the only multi-modal trip planner fully implemented by a US transit agency, was funded by a $1M federal grant in 2005.

OpenPlans was responsible from the outset to lead the technical development of the project, to facilitate a strong development community around the code, and to provide a sustainable business model around the software to ensure there are maintenance and support options for agencies.

OTP Metro RTO Grant 2009-2011 Primary Goals

1. Demonstrate the feasibility of implementing and maintaining an open source, multi-modal trip planning service (OS MMTPS).
2. Contribute documentation and source code improvements for the benefit of others.
3. Increase and strengthen the development community around the code.
4. Test the usability and accuracy of the MMTPS planned trips.

All primary project goals have been realized, the first three within the first year. Other secondary project goals, are as follows and are being addressed in this document and with the public beta release in Portland:

1. Document information such as lessons learned and feedback from participating partners, which will be useful for future government open source software collaborations.
2. Identify areas of improvement to the core algorithm and regional datasets that are needed to increase the accuracy and usefulness of the planned trips.
3. Encourage use of bike lockers to reduce number of bikes on trains and buses
4. Improve the satisfaction rates and usefulness of on-line trip planners.

TriMet and OpenPlans held a two day workshop to kick-off the project July 14-15, 2009. The results of the workshop were formulated into the Project Management Plan for the following two years. The project was given the name OpenTripPlanner (OTP) at the workshop, and opentripplanner.org was established for communication purposes. The table below shows key project tasks and their completion dates.

Project Tasks	Estimated Completion Date	Actual Completion Date
Project Management Plan	September 2009	September 2009
Software development and implementation of working prototype	January 2011	February 2010
Evaluation studies for data efficacy and OTP planned trip results	May 2011	July 2010 – June 2011

All components of the OpenTripPlanner are open. All of its software is open source; it is built on an open architecture; it was developed using an open source development method; and it utilizes all open data including OpenStreetMap, GTFS[2], and the USGS National Elevation Dataset (NED). The open architecture allows for adding, upgrading and swapping components of the system. Further, the components of OTP were specifically designed to be modular for easy customization, including the use of other datasets.

Core Software Functionality

The OpenTripPlanner software currently:

- Plans multi-modal walking, biking and transit trips
- Takes travel time, road type/safety, and elevation data into account when planning bike trips, and provides a user interface for adjusting the weighting of these three factors
- Shows graphical elevation profiles for bike trips
- Capable of importing data from a variety of sources including GTFS, USGS National Elevation Dataset, OpenStreetMap, Tele Atlas, Navteq, shapefiles, etc.
- Plans trips in about 100ms in a moderate sized city
- Exposes a RESTful API (XML and JSON), which other apps or front-ends can build on
- Provides translations from English into the following seven languages: French, Gaelic, Hungarian, Italian, Marathi, Polish, and Spanish

In February 2010, the OpenTripPlanner had active demonstrations running in Portland, Oregon, New York City, and Poznan, Poland. Additional demos have been released as follows:

- Bilbao, Spain June 2010
- Granada, Spain July 2010
- Pune, India Bus Guide October 2010
- Smartrip Transport for Dublin, Ireland Journey Planner January 2011
- Ottawa, Canada Green Trip Planner January 2011
- Tampa, Florida May 2011
- Gipuzkoa, Spain Transit Android App May 2011
- Tel Aviv, Israel June 2011
- Budapest, Hungary July 2011

[2] GTFS is an open data protocol. The decision to make the actual data itself open and publicly available in this format is made on an agency-by-agency basis.

This experimentation with software by such a diverse group shows the value of the OSS development model and the degree of anticipation of future improvements.

In July 2010, the one-year anniversary of OTP, opentripplanner.com was released to promote the trip planner as well as the support services and customization options. Shortly following, TriMet concluded its analysis on routable street network alternatives for the underlying basemap in OTP and selected OpenStreetMap (OSM). College students were hired to improve the OSM data in the Portland Metro area using local jurisdictional data.

Preliminary Trip Testing

In May 2011, OSM data improvements were completed in a designated test area and preliminary OTP trip testing began in June 2011. The test results demonstrated a very competitive trip planned when compared against Google Maps, Google Transit, TriMet's Proprietary Transit Trip Planner (PTTP), and byCycle.org. The tests also identified improvements for the software, which are currently being addressed. Most revealing were the results of the bike-to-transit trip analysis. When OTP bike-to-transit trip results were compared against single-mode transit trips in Google Transit and PTTP, OTP demonstrated a capacity to save significant travel time when using multi-mode in contrast to single-mode trip planners.

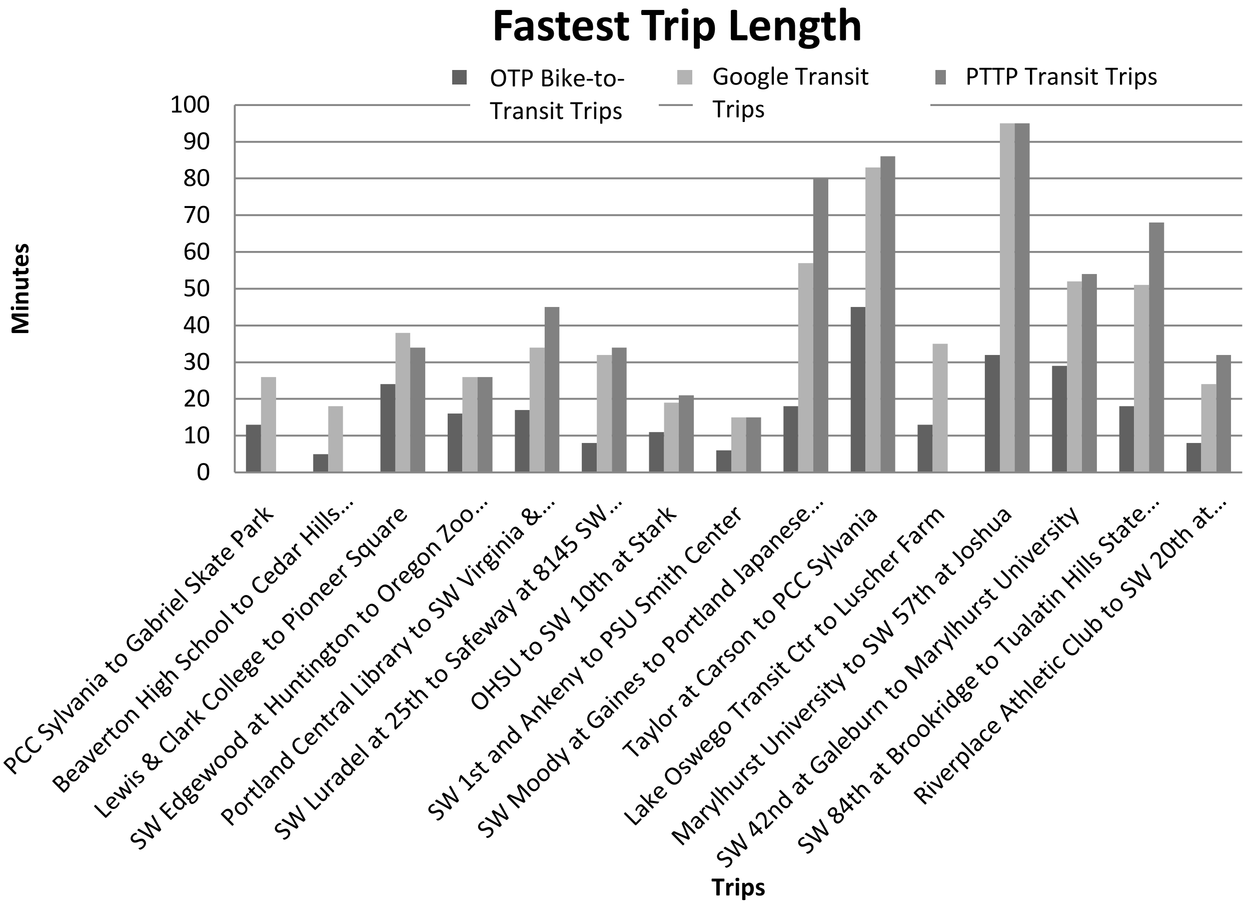


Figure 1: Comparison of multi-modal trip planner (OTP Bike-to-Transit Trips) results with single-mode trip planner (Google Transit Trips and TriMet's Proprietary Transit Trip Planner (PTTP)) results - demonstrates OTP can save significant travel time.

Next Steps

A public beta launch of the OTP is scheduled for October 2011 after completion of OSM improvements, further testing, and development of the required functionality for the launch.

For the next phase of the OTP project effort, Metro extended the original Metro RTO Grant for an additional two years and awarded TriMet with an additional $75,000. The objective of this next phase is to include additional functionality, such as improved management tools for both the developers and the supporting administrative staff for continued maintenance of the application and data. OTP will eventually replace the TriMet's proprietary transit trip planner functions used by call-takers in the call center.

TriMet and OpenPlans will continue to lead the project and support it with substantial in-kind contributions to ensure the successful implementation of a system that can be supported and maintained over time.

This final report for the Metro 2009-2011 Regional Travel Options grant includes five sections:

Section 1 provides background on the project and the events leading up to the grant award in July 2009.

Section 2 describes the project events over the two-year grant cycle and provides an overview of the open elements of the OpenTripPlanner.

Section 3 focuses on the data improvements to the underlying routable network in OpenStreetMap.

Section 4 provides information about the evaluation studies of the data, the OTP algorithm, and the weighting mechanisms.

Section 5 concludes the report outlining the next steps in the subsequent grant proposal over the following two-years, and provides a vision for the future of OTP.

SECTION 1: BACKGROUND

TriMet provides public transportation for much of the three counties in the Portland, Oregon metro area. It operates a comprehensive transit network including a 52-mile MAX light rail system, 79 bus lines, 14.7-mile WES Commuter Rail, service for seniors and people with disabilities, and enhanced amenities and information. TriMet carries more people than any other U.S. transit system its size. Weekly ridership on buses, MAX and WES has increased for all but one year in the past 22 years, and ridership has outpaced population growth and daily vehicle miles traveled for more than a decade.

Transit is an integral part of livability in the Portland metro area and **TriMet's Trip Planner** is an essential and widely used application in the region. TriMet's web site averages over 950,000 visits per month and about 550,000 of those visits are to plan transit trips. The application consistently ranks among the top Google searches for the region, as indicated by Google Zeitgeist, which annually publishes the top ten searches by city. In 2009, more people searched for the TriMet Trip Planner than any other Google search in Portland, Oregon; in 2010, the TriMet Trip Planner ranked as the second most popular Google search in Portland.

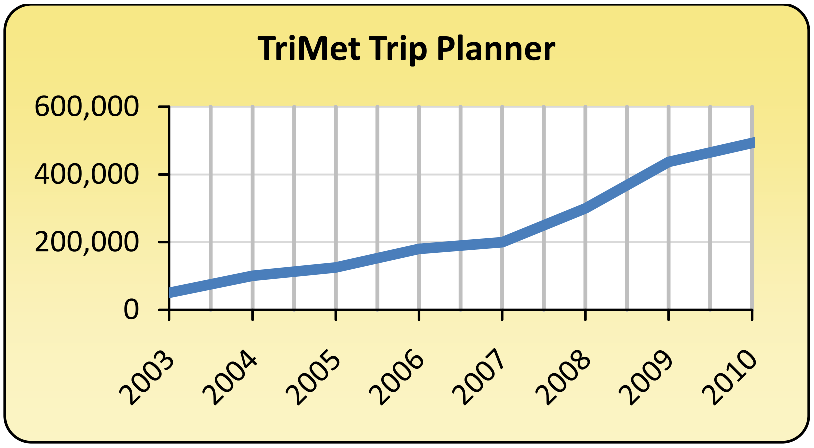


Figure 2: Planned trips per month on TriMet Trip Planner.

Transit Only and Single-Mode Trip Planners

A number of single-mode trip planner tools are currently available to TriMet users, which do not fully meet customer needs. TriMet implemented its existing web-based text trip planner in 2000, which plans transit trips with limited walking directions. It is limited to text-based instructions and lacks a fully integrated interactive map[3]. The first beta release of Google Transit in Portland in 2005 prompted

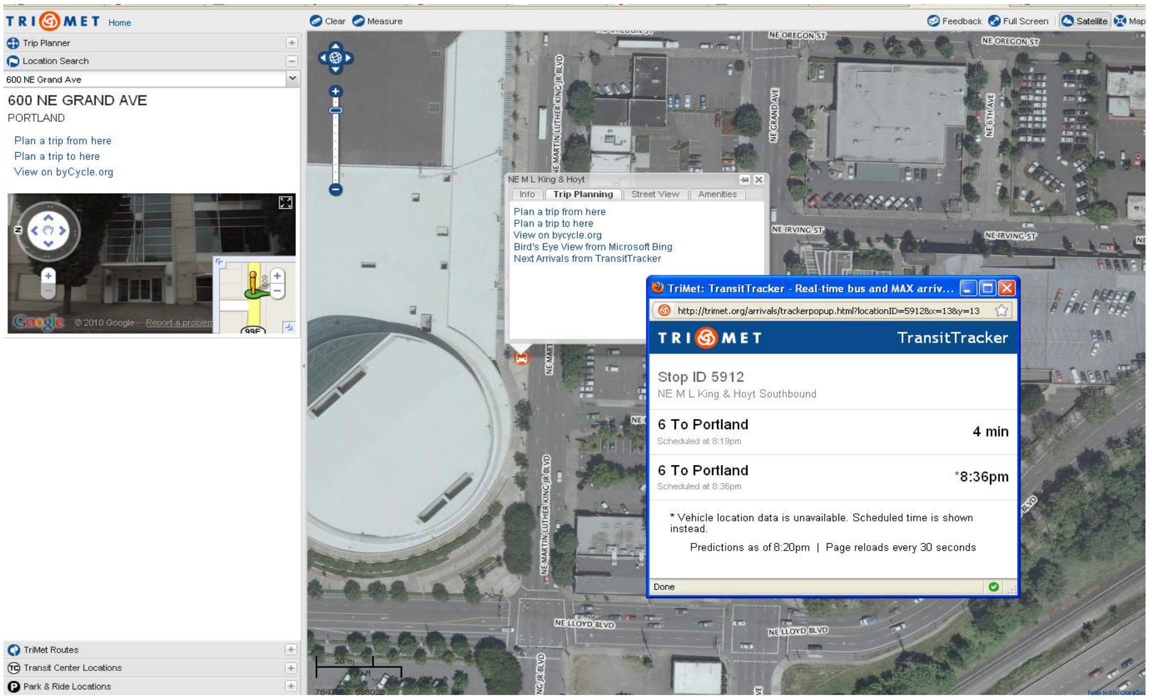
[3] Although transit system maps have historically been distributed in hardcopy, paper maps are usually out of date almost as soon as they are printed. An on-line interactive system map is an improvement over printed maps because they can be kept current automatically.

customers to expect the results of planned transit trips to be displayed on a web-based map, and in 2008, TriMet launched its own web-based interactive system map which incorporates trip planning functions.

TriMet Open Source, Interactive System Map

TriMet has made a number of software infrastructure and application improvements to create more map-based tools for its users. In 2007, TriMet analyzed both proprietary and open source software alternatives for internet mapping platforms. An open source solution was selected because it aligned with TriMet's Information Technology (IT) standards and met the identified functional and non-functional requirements. TriMet's current Trip Planner website, maps.trimet.org, is now powered by Open Geospatial Consortium (OGC) compliant open source technologies, including the OpenGeo Stack: GeoServer, OpenLayers, and PostGIS. Web technologies allow for bringing information from a variety of sources into one application so that customers can make informed decisions about the TriMet system. The application provides Google Street View, TransitTracker information on next arrivals, real-time bus locations, stop locations, amenities directly from the enterprise database, and high resolution aerial photography.

TriMet purchased an annual support contract from OpenPlans to support the OpenGeo Stack prior to public launch. The experience highlights the appeal of open source software: in addition to standard technical support and training, the OpenPlans contract included a third (and most important) aspect - development hours. TriMet used these development hours to improve the map's cartographic features and to create additional functionality, including a "Measure Distance" tool. OpenPlans ensured that the new features went back to the source code for the benefit of all users, not just TriMet. Thus, TriMet's experience shows that adopting open source software not only made rapid code fixes and customized functionality possible, but also increased efficiency as the improvements were made available to the user community as a whole.



600 NE GRAND AVE
PORTLAND

Plan a trip from here
Plan a trip to here
View on byCycle.org

NE M L King & Hoyt

Plan a trip from here
Plan a trip to here
View on bycycle.org
Bird's Eye View from Microsoft Bing
Next Arrivals from TransitTracker

TriMet: TransitTracker - Real-time bus and MAX arriv...

http://trimet.org/arrivals/trackerpopup.html?locationID=5912&x=13&y=13

TRIMET TransitTracker

Stop ID 5912
NE M L King & Hoyt Southbound

6 To Portland — 4 min
Scheduled at 8:19pm

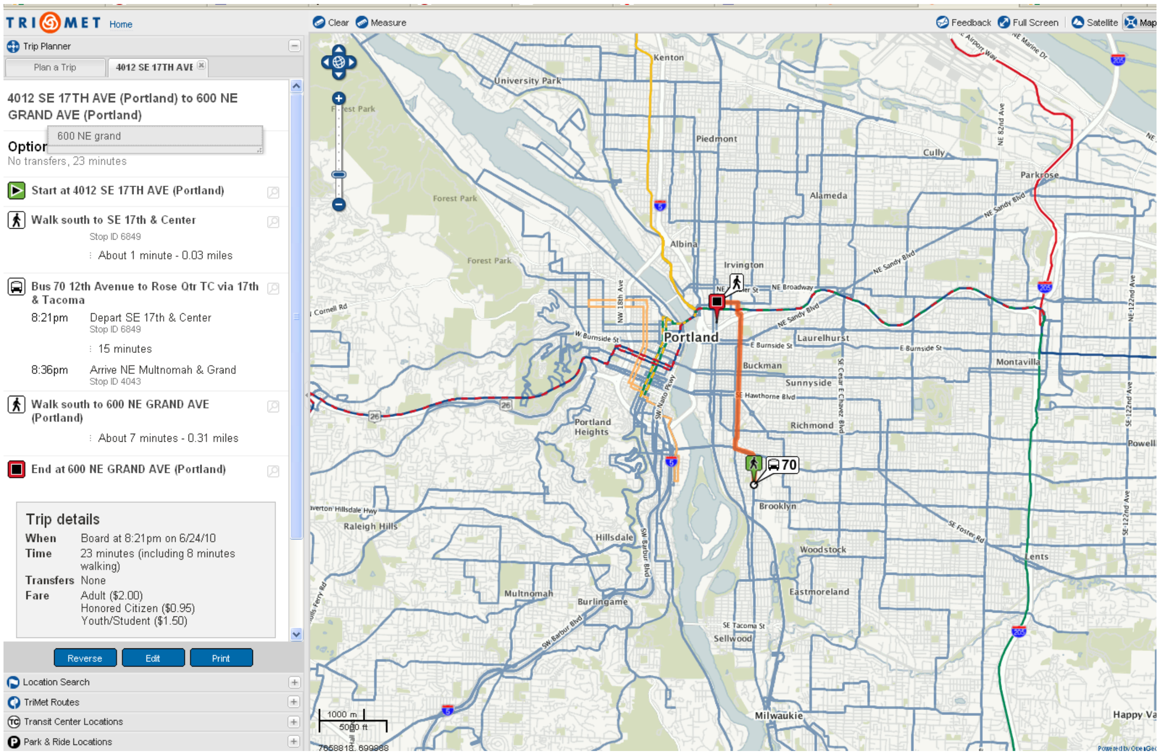
6 To Portland — *8:36pm
Scheduled at 8:36pm

* Vehicle location data is unavailable. Scheduled time is shown instead.

Predictions as of 8:20pm | Page reloads every 30 seconds

Done

Figure 3: TriMet's Interactive System Map at maps.trimet.org uses open source technology.



4012 SE 17TH AVE (Portland) to 600 NE GRAND AVE (Portland)

Optior 600 NE grand
No transfers, 23 minutes

Start at 4012 SE 17TH AVE (Portland)

Walk south to SE 17th & Center
Stop ID 6849
About 1 minute - 0.03 miles

Bus 70 12th Avenue to Rose Qtr TC via 17th & Tacoma
8:21pm Depart SE 17th & Center
Stop ID 6849
15 minutes
8:36pm Arrive NE Multnomah & Grand
Stop ID 4043

Walk south to 600 NE GRAND AVE (Portland)
About 7 minutes - 0.31 miles

End at 600 NE GRAND AVE (Portland)

Trip details

When	Board at 8:21pm on 6/24/10
Time	23 minutes (including 8 minutes walking)
Transfers	None
Fare	Adult ($2.00) Honored Citizen ($0.95) Youth/Student ($1.50)

Figure 4: TriMet's Interactive System Map displays planned transit trips generated from TriMet's Proprietary Transit Trip Planner (PTTP).

Beyond TriMet's interactive web map, several other single-mode trip planning tools are regionally available. **byCycle.org** is a popular regional single-mode bicycle trip planner, developed by Wyatt Baldwin. It does not, however, integrate planned trips with transit routes.

Google Maps is, of course, widely used and capable of planning several different single-mode trips, including transit, driving, walking and biking. However, Google Maps cannot plan a trip consisting of multiple modes, such as bike to transit.

Multi-Modal Trip Planners

Although single-mode trip planning tools dominate the market, several multi-modal trip planners have recently emerged. Although multi-modal trip planners improve on existing single-mode planners, none of these efforts appear to be currently viable for widespread transit agency adoption because the software either lacks a rich developer community or is otherwise prohibitively costly to implement.

The **A-Train Trip Planner** was developed by David Emory of FivePoints software, on behalf of Citizens for Progressive Transit. It is capable of planning a combination of walking, biking and transit trips. It was successfully implemented and publicly launched as the Atlanta, Georgia A-TRAIN Trip Planner (http://trip.atltransit.com/) in November 2007.

GraphServer (http://graphserver.wiki.sourceforge.net/) is an engine that generates multi-modal routes. It was developed by Brandon Martin-Anderson and has been successfully implemented in Toronto, Canada at http://myttc.ca/, and in Seattle, Washington at http://busmonster.com/.

Chicago's goroo.com is the only multi-modal trip planner in the US fully implemented by a transit agency. The Chicago RTA project was initiated by a federal grant in 2005 for $1M. GoRoo utilizes MDV, a proprietary German product subsequently purchased by San Francisco MTA to run their 5-1-1 system. Multi-modal routing is not yet functional in the Bay area.

Open Source Software Considerations

Since 2002, TriMet has saved significant costs by exploring open source alternatives to every new technology system, including internet mapping technologies. TriMet has developed a method of comparing open source software side by side with proprietary solutions against the requirements. The progression of available open source systems, from the backend infrastructure tools, to the front-end customer systems, has enabled a rapid progression of open source solutions throughout the agency.

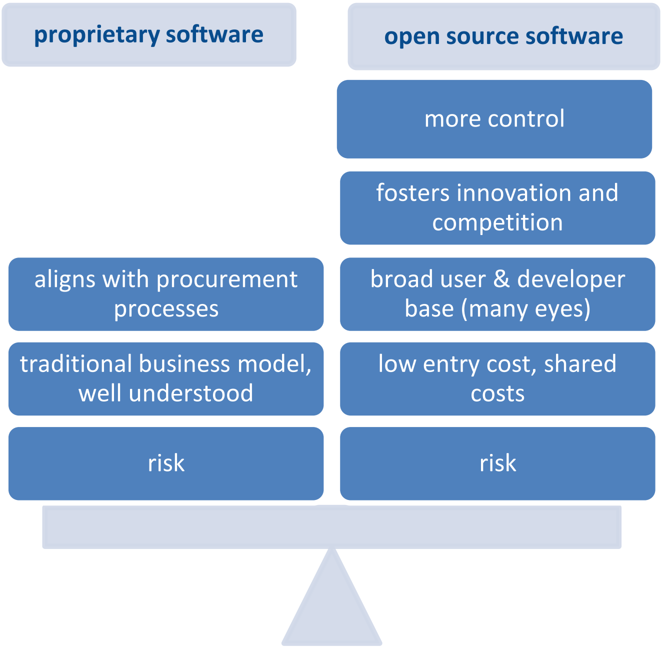

proprietary software
open source software
more control
fosters innovation and competition
aligns with procurement processes
broad user & developer base (many eyes)
traditional business model, well understood
low entry cost, shared costs
risk
risk

Figure 5: Not all open source software is the same, and not all proprietary software is the same. It is important to research the alternatives and fully define the requirements.

In addition to comparing software features against requirements, for both open source and proprietary software, the following criteria are examined by TriMet to determine the risks and the advantages:

- How large is the developer base, how many developers are involved with the project
- How many working implementations exist and how satisfied are the users
- OS Code/Language – ex: can it be supported and maintained internally, does it adhere to internal IT standards
- Terms and conditions of open source license or vendor's contract
- Options for support and maintenance contracts
- Active community support (how quickly are questions answered on support forums or how quickly does the vendor respond to issues)
- Calculations and comparisons of implementation costs, resources, short & long term operating costs
- How customizable is the software for individual needs

With a strong developer community around the software, open source software can have appealing advantages over proprietary software: it provides more control over fixes and new features; it fosters innovation and competition leading to better competitive products; it can have a broader user and developer base which means there are more eyes on the code; and risks and costs are shared within the community, which can be especially appealing for government agencies. In addition, there is no fee for the software so accessibility enables prototyping and testing. However, not all open source software is the same and not all proprietary software is the same and the advantages and risks should be identified and weighed when performing a software alternatives analysis.

Open Source Approach Selected for OTP

When TriMet chose to pursue an open source approach to developing a new trip planner, it chose to apply for a grant with a number of partners, who were also committed to open source solutions. One of the partners, OpenPlans, is a non-profit technology organization based in New York that focuses on open government and they specialize in the development and support of open source software. TriMet worked previously with the OpenGeo group of OpenPlans on TriMet's Interactive System Map and backend open source mapping infrastructure and approached them about partnering with TriMet and several other developers, including Brandon Martin-Anderson, David Emory, Brian Ferris, to apply for the Metro Regional Travel Options (RTO) Grant in 2008. Replacing the only remaining proprietary component of TriMet's Interactive System Map/Trip Planner was a natural progression and tie-in to the OpenGeo software suite.

TriMet was subsequently awarded $68,960 in an RTO grant and TriMet and OpenPlans matched this with a 50% in-kind contribution for a two-year project total cost of $137,920 to implement a prototype of an open source, multi-modal trip planner. OpenPlans was responsible from the outset to lead the technical development of the project, to facilitate a strong development community around the code, and to provide a sustainable business model around the software to ensure there are maintenance and support options for agencies. Please refer to Appendix A for a list of the high-level requirements identified for the project.

SECTION 2: PROJECT OVERVIEW

This section will review the highlights and milestones of the project.

Kick-Off Workshop July 2009

TriMet and OpenPlans held a 2-day workshop to kick-off the project July 14-15, 2009 and invited developers who had experience working with trip planners and open source software to participate in the kick-off workshop. It was critical that all key developers and the stakeholders from participating agencies attend the workshop in order to set the platform and pace for continued collaboration throughout the next two years. The goals of the workshop were to formulate a preliminary scope, goals and project plan for the development of an open source, multi-modal trip planner, which are highlighted in the next section (Project Management Plan September 2009).

Representatives from the following agencies participated in the kick-off event and assisted in identifying issues with current trip planners, gathering requirements, and prioritization: TriMet, Utah Transit Authority, King County Metro, C-Tran, Dallas Area Rapid Transit, RTD Denver, New York State DOT, Volpe for the National Parks of New York Harbor (NPNH) TIS Project, and the New England Transportation Institute.

Those in attendance from OpenPlans included, Nicholas Bergson-Shilcock, Nic Grossman, and Chris Holmes. Also in attendance were the following developers who participated in the technical discussion on day two and created a technical plan and the architecture:

- Brandon Martin-Anderson, GraphServer
- Brian Ferris, One-Bus-Away
- David Emory, Five Points (A-Train)
- Frank Purcell, TriMet (developer of maps.trimet.org)
- Mike Giligan, TriMet
- Wyatt Baldwin, TriMet (previously developed byCycle.org)

At the workshop, opentripplanner.org was established as a tool for on-going communication and for informational purposes. It captured the information gathered at the two-day workshop. The following decisions were made:

- OTP would use an open source development method
- OTP would use open architecture

- Open data would be explored as an option
- An appropriate open source software license would be explored

All the developers in attendance shared a common interest in creating an open source, multi-modal trip planner. This initial workshop highlighted the value of collaboration among developers, emphasizing the possibility of creating a product that was greater than the sum of the individual elements. The OTP workshop thus established an understanding among participants of shared maintenance, shared responsibilities, shared respect, shared goals, and shared rewards.

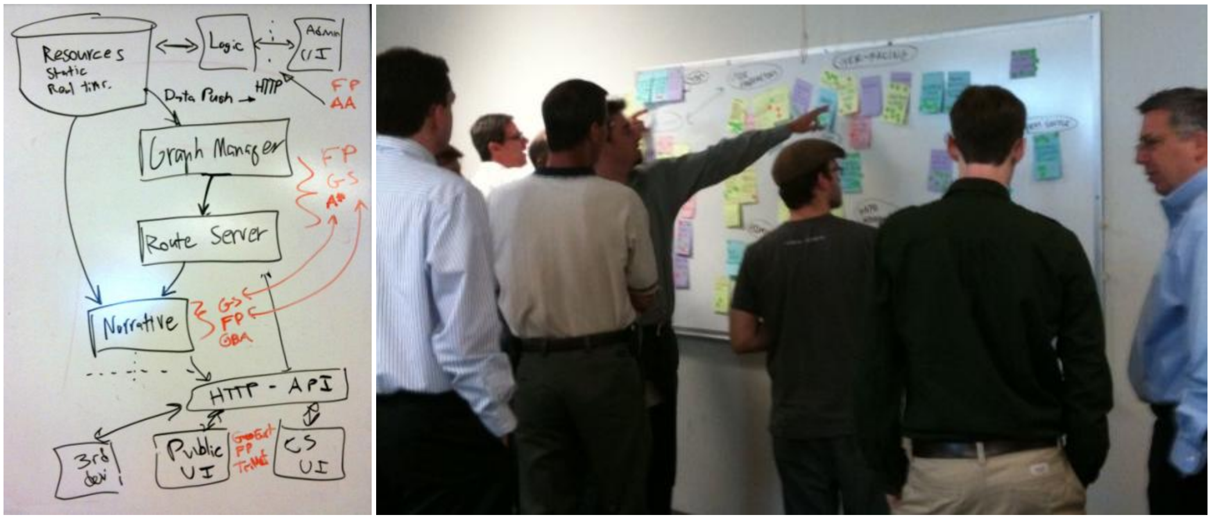


Figure 6: Photos from the July 2009 OTP Kick-Off Workshop

Project Management Plan September 2009

The following scope, goals, and project plan were formulated at the workshop, reviewed, finalized, and then submitted in September 2009. This section can be found on-line at http://opentripplanner.org/wiki/ProjectPlan.

Project Scope and Goals

The primary goals for the OpenTripPlanner are:

- Develop a complete open source, multi-modal trip planner building on existing open source trip planning and routing tools
- Build a healthy development community to ensure long-term growth and support
- Deploy a working trip planning system using TriMet's datasets for use in Portland
- Test usability and accuracy of trips planned using the new system

Further, the project is guided by several broad directives to be taken into account throughout the development process. Chief among these are:

Community
Open source projects flourish when backed by a passionate and dedicated community of developers and users. Such communities, however, arise neither overnight nor spontaneously. Their development requires sustained and careful effort on the part the founding contributors to cultivate an environment that is both inviting and accessible to would-be contributors, helping them become more involved in the project as time progresses.

Usability
One point that became clearly evident during the Portland workshop is that existing trip planners frequently fall short in terms of usability. For instance, existing software provides users with inadequate, misleading and sometimes unnecessary error messages. This has been a point of frustration for many agencies with their current trip planning solutions.

Modularity
Given the complex nature of trip planning software and the diversity of systems and software stacks, a trip planner is likely to be used in conjunction with a driving goal of the project is to maintain a high degree of modularity. Where reasonable, components should remain useful on their own in addition to being part of the larger system.

Deployability
In order to encourage adoption and use at additional transit agencies, developers will make an effort to ensure that the process of deploying an instance of OpenTripPlanner is as painless as possible. The multiplicity of potential installation environments precludes a one-click process. However, through documentation and intelligent design decisions, installation and configuration can be made straightforward and reliable.

Project Plan and Milestones

The major components and milestones for the project are divided into three areas and detailed below. Specific tickets are referenced where appropriate. Eventually, all of the work here will be ticketed and broken into milestones to improve tracking and understanding of progress. Note also that there is obvious overlap even across the work areas and work need not (and likely cannot) proceed in a strictly serial fashion. For example, parts of area three, packaging and documentation, should go forward in parallel with area two, the core development work.

Work Area One: Getting Off the Ground

This area includes the initial setup, infrastructure, decision-making, and planning that will help make the later development work a success.

- Establish project name, domain, and basic infrastructure, including issue tracker, mailing lists, source code management, and initial documentation

- Compare performance and test language differences in core routing engine.

- Design trip planner Application Programming Interface (API)

- Finalize technical decisions and begin implementing core engine

Work Area Two: Implementation of Project Modules

This area includes the development of the core components of the software, building off of the foundational work done in the initial phase. These components are largely taken from the draft architectural diagram that was developed during the technical session at the Portland workshop.

- Core routing engine, comprised of the graph manager and route server in the architectural diagram. This component includes code that compiles the graph from the data store and the algorithm that generates the best path through the network given input constraints.

- Narrative engine. This component prepares the final human-readable trip for the client from the path through the graph.

- Data store and manager, or the "resources" and corresponding logic box in the diagram. This component allows for loading and using existing datasets, including transit graphs and schedules. The data store will also communicate with the narrative engine, allowing for service advisories to be overlayed onto the final trip where appropriate. While initially consisting solely of static data (e.g., GTFS schedule files), this component could eventually grow to support certain real-time data sources that could be taken into account by the routing engine itself, though this is presently beyond the scope of our current work.

- Front-end user interface, or widgets to be used in actually querying the trip planner, e.g., from a website. This work will be largely based on TriMet's existing front-end as well as parts of the Five Points' user interface. Ultimately, we hope to factor out common elements from these projects and use them as the basis for a GeoExt-based "Transit Widgets" library, which could be used as an off-the-shelf toolbox for developers building transit-related web applications.

- Administrative user interface, for performing administrative tasks, e.g., updating the graph data.

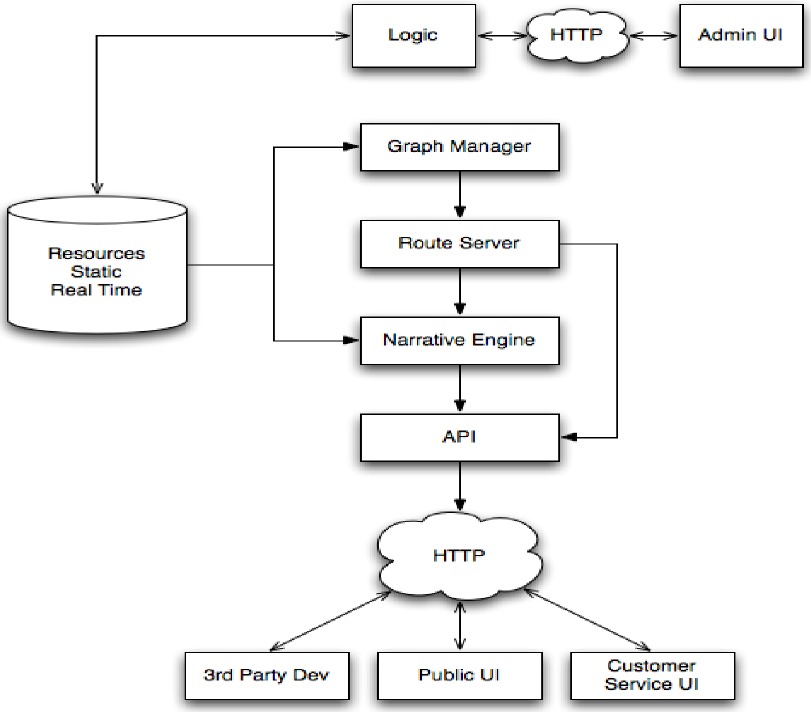


Figure 7: Architectural Diagram of OTP

Work Area Three: Documentation and Packaging

This area includes the work that is outside the core development but which is still essential to a successful project.

- Write and maintain high quality, useful documentation

- Develop and test project packaging and installation scripts to make deployment and configuration straightforward and reliable

- Build test system for testing trip planner on real data

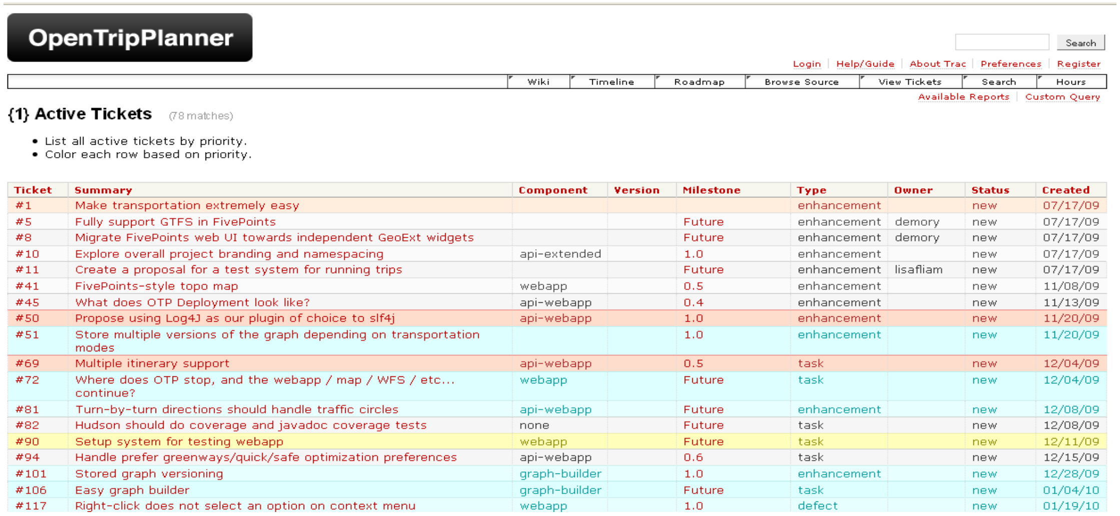
The Open Elements of the OpenTripPlanner

Open Source Development Method

OTP uses an open source development method: a collaborative method of software design, development, and distribution. Developers have full access to source code. All major decisions are voted on for approval including: open source license; changes to the homepage; frequency and time of weekly teleconference calls; code usage and version (for details search for "vote" at http://opentripplanner.org/report/1).

The OpenTripPlanner community makes further decisions through a proposal and informal voting process on the project mailing list:

> "While we do vote on proposals, we don't vote in a strict democratic sense, but rather as a way to easily register opinions, foster discussion, and move toward consensus. When responding to a proposal, we use the following system:
>
> | | |
> |---|---|
> | +1 | I support this |
> | +0 | I don't have a strong opinion, but I'm not opposed |
> | - 0 | I'm against this, but I don't have a good alternative / I'm not willing to do the work on the alternative / I won't block |
> | -1 | Blocking no (note: in general and when appropriate, this requires the blocker to propose something else that he/she would help put the time into doing) |
>
> A proposal does *not* need to be a formal or lengthy document; it can and should be a straightforward recommendation of what you want to do, ideally with a brief explanation for why it's a good idea.
>
> Proposals are just messages sent to the list and can be as simple as *"I think we should do X because of Y and Z. Deadline for response is 10/29. Assuming I've heard no blocking votes by then, I'll go ahead."* Note that you should make sure to include a deadline by which you will go ahead and do what you're proposing if you don't hear any blocking responses. In general, you should leave at least 72 hours for people to respond. This is not a hard-and-fast rule and you should use your best judgment in determining how far in the future the deadline should be depending on the magnitude of the proposal and how much it will affect the overall project and the rest of the community. "


OpenTripPlanner
Search
Login | Help/Guide | About Trac | Preferences | Register
Wiki
Timeline
Roadmap
Browse Source
View Tickets
Search
Hours
Available Reports | Custom Query

{1} Active Tickets (78 matches)

- List all active tickets by priority.
- Color each row based on priority.

Ticket	Summary	Component	Version	Milestone	Type	Owner	Status	Created
#1	Make transportation extremely easy				enhancement		new	07/17/09
#5	Fully support GTFS in FivePoints			Future	enhancement	demory	new	07/17/09
#8	Migrate FivePoints web UI towards independent GeoExt widgets			Future	enhancement	demory	new	07/17/09
#10	Explore overall project branding and namespacing	api-extended		1.0	enhancement		new	07/17/09
#11	Create a proposal for a test system for running trips			Future	enhancement	lisafliam	new	07/17/09
#41	FivePoints-style topo map	webapp		0.5	enhancement		new	11/08/09
#45	What does OTP Deployment look like?	api-webapp		0.4	enhancement		new	11/13/09
#50	Propose using Log4J as our plugin of choice to slf4j	api-webapp		1.0	enhancement		new	11/20/09
#51	Store multiple versions of the graph depending on transportation modes			1.0	enhancement		new	11/20/09
#69	Multiple itinerary support	api-webapp		0.5	task		new	12/04/09
#72	Where does OTP stop, and the webapp / map / WFS / etc... continue?	webapp		Future	task		new	12/04/09
#81	Turn-by-turn directions should handle traffic circles	api-webapp		Future	enhancement		new	12/08/09
#82	Hudson should do coverage and javadoc coverage tests	none		Future	task		new	12/08/09
#90	Setup system for testing webapp	webapp		Future	task		new	12/11/09
#94	Handle prefer greenways/quick/safe optimization preferences	api-webapp		0.6	task		new	12/15/09
#101	Stored graph versioning	graph-builder		1.0	enhancement		new	12/28/09
#106	Easy graph builder	graph-builder		Future	task		new	01/04/10
#117	Right-click does not select an option on context menu	webapp		1.0	defect		new	01/19/10

Figure 8: Collaborative method of tracking work and progress at opentripplanner.org.

Open Source Software

Open source software has a non-restrictive license that allows use, modification, and redistribution of the source code. The OpenTripPlanner is currently released under the LGPL: GNU Lesser General Public License Version 3, 29 June 2007 (http://www.gnu.org/licenses/lgpl.html). The type of open source license for the software was debated among the developers and voted on through the established voting process.

Open Architecture

Open Architecture allows for adding, upgrading, and swapping components of a system. The components of OTP were specifically designed to be modular for easy customization. For instance, OTP is designed to use OpenStreetMap out-of-the-box, however, it can easily be customized to use a variety of data.

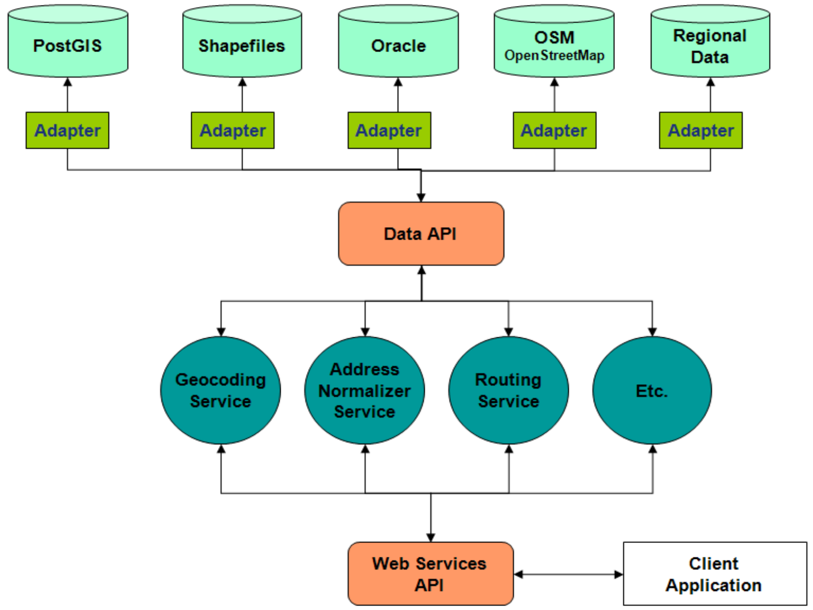


Figure 9: Open architecture of OTP offers ease of customization.

Open Data

Open data implies the data is freely available to everyone. OTP used three open datasets to generate an intelligent routable graph:

1. General Transit Feed Spec (GTFS) is an open and common data format for the temporal schedule data and spatial data that is required for transit trip itinerary planning. GTFS was initially developed collaboratively for participation in Google Transit, which now has 455 participating cities worldwide. GTFS is now used by hundreds of applications in addition to Google. Although the protocol itself is licensed under a Creative Commons Attribution License, the decision to make the actual data itself open and publicly available to the public is made on an agency-by-agency basis. Over 200 agencies, including TriMet, provide GTFS data openly to the public.
2. USGS National Elevation Dataset (NED), available at http://ned.usgs.gov/. NED is a seamless dataset for the United States.
3. OpenStreetMap, a seamless worldwide basemap designed for routing purposes available for viewing at http://www.openstreetmap.org.

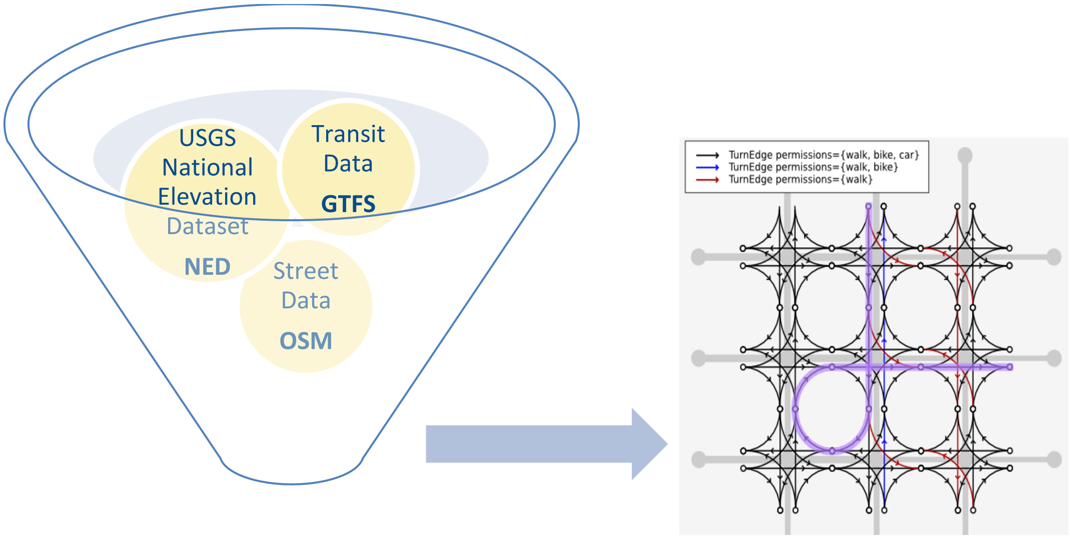

USGS
National
Elevation
Dataset
NED
Transit
Data
GTFS
Street
Data
OSM
TurnEdge permissions={walk, bike, car}
TurnEdge permissions={walk, bike}
TurnEdge permissions={walk}

Figure 10: By default, OTP uses open source datasets (GTFS, NED, and OSM) to build a routable graph. Customization is viable for other datasets and types.

OTP identifies the extent of the area needed by the GTFS dataset, and it then automatically grabs the necessary data coverage from both the NED and the OSM.

Implementation of Prototypes February 2010

In February 2010 the OpenTripPlanner had active demonstrations[4] running in:

1. Portland, Oregon
2. New York City
3. Poznan, Poland

Additional demos have been released as follows:

4. Bilbao, Spain June 2010
5. Granada, Spain July 2010
6. Pune, India Bus Guide October 2010
7. Smartrip Transport for Dublin, Ireland Journey Planner January 2011
8. Ottawa, Canada Green Trip Planner January 2011
9. Tampa, Florida May 2011
10. Gipuzkoa, Spain Transit Android App May 2011
11. Tel Aviv, Israel June 2011
12. Budapest, Hungary July 2011

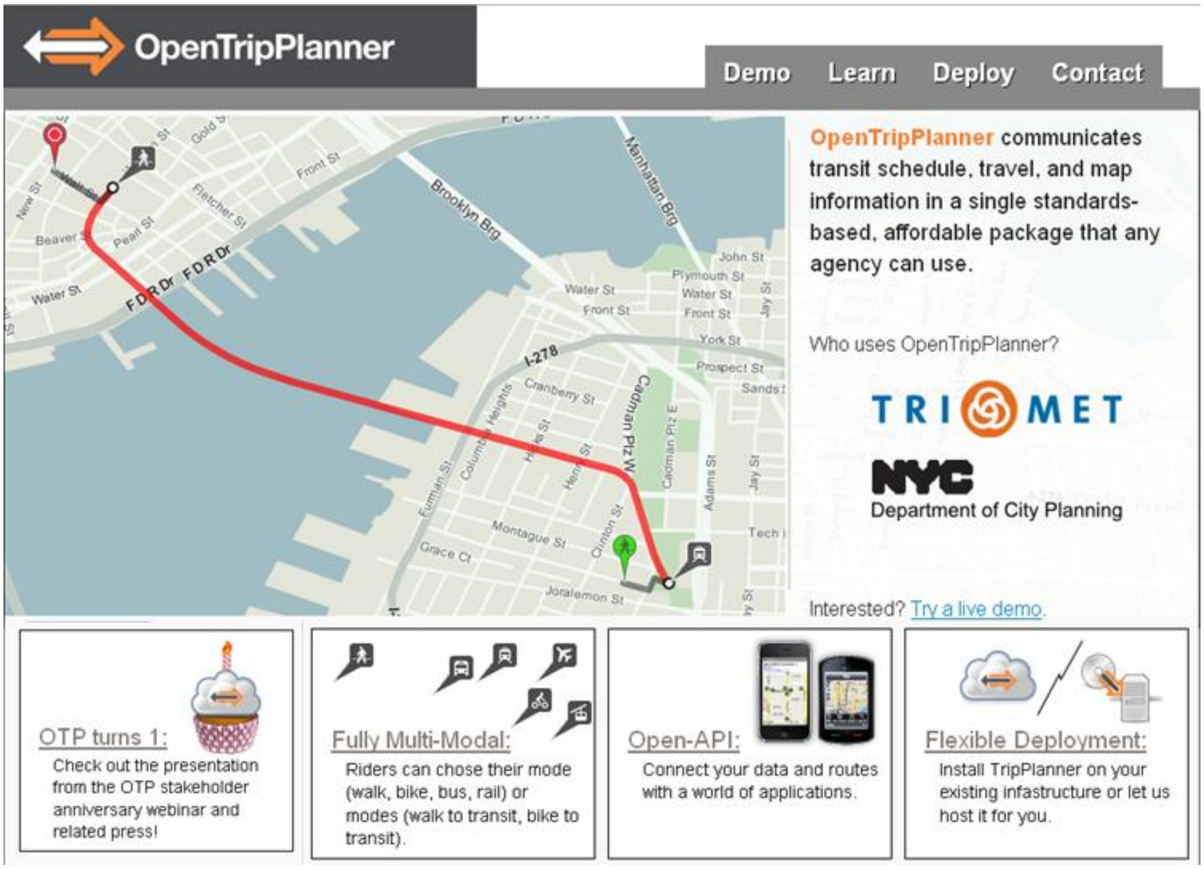
[4] Designed for demonstration purposes only, these demos are intended to be used as a tool for the developers to test the OTP algorithm and the underlying data. Several of these demos have moved to production systems.

TriMet released its first demo using RLIS data. In July 2010, a second instance of OTP was established using OSM data to compare it to RLIS.   Please refer to Section 3 for further details related to street network analysis and the implications of data choice for routing.

Year One Anniversary July 2010

TriMet and OpenPlans held a webinar in July of 2010 to mark the one-year anniversary of OTP development and to provide the stakeholders with an update and demonstration of the prototypes.

opentripplanner.com was also released promoting the trip planner as well as the support services and customization options:



OpenTripPlanner communicates transit schedule, travel, and map information in a single standards-based, affordable package that any agency can use.

Who uses OpenTripPlanner?

Interested? Try a live demo.

OTP turns 1: Check out the presentation from the OTP stakeholder anniversary webinar and related press!

Fully Multi-Modal: Riders can chose their mode (walk, bike, bus, rail) or modes (walk to transit, bike to transit).

Open-API: Connect your data and routes with a world of applications.

Flexible Deployment: Install TripPlanner on your existing infastructure or let us host it for you.

Figure 11: opentripplanner.com was released for commercial support options.

TriMet and OpenPlans followed this with a public meeting at Metro in which OTP was unveiled. The presentations and news releases are available at http://opentripplanner.com/Press.

Progress in Year Two

The majority of the grant funds were depleted in the first year grant cycle for the bulk of the development. All preliminary requirements were met. Despite lack of funding, the development community, that had been successfully established by OpenPlans, and OpenPlans, continued to contribute significantly to the project. Contributions and version releases were lead in the second year by David Turner, Software Engineer at OpenPlans. Development contributions include:

- Code quality improvements
- Improved wheelchair accessibility
- Graph reloading
- Speed improvements
- Memory usage improvements
- Code documentation
- Tutorials and user guides
- Intermediate stops in API
- Better extensibility
- Visualization improvements
- Making bikes on transit optional
- Kept up-to-date with changing technology
- Edge notes
- Nominatim geocoder support
- Cleaner max walk distance support
- Support for certain common GTFS failure modes
- Workarounds for OSM data issues (pedestrian islands)
- Support loading OSM from Protocol Buffers format & bz2 xml
- OSM future proofing
- OSM permission bug fixes
- Traffic circles / roundabouts
- Much improved transfer support (min transfer times; transfers now routed on walking network)
- Support for multiple route names on the same stop pattern (typically night buses)

- Dozens of miscellaneous bug fixes
- Gvsig visualization
- Documentation for graph builder
- Better names for some OSM ways
- Better slope visualization
- Slope override for bridges
- Some integration with OneBusAway
- Preferred/non-preferred/banned routes
- Translations from English into the following seven languages: French, Gaelic, Hungarian, Italian, Marathi, Polish, and Spanish

SECTION 3: DEVELOPING A ROUTABLE NETWORK IN OPENSTREETMAP

As discussed in Section 2, OpenTripPlanner (OTP) uses a street network to generate a graph that determines optimal routes based on selected variables, such as fastest trip, safest trip, etc. Thus, the street network must have attributes to support routing-like turn restrictions and directionality.

There are three alternatives for street networks for OTP in the Portland, Oregon area:

1. Commercial datasets, such as Tele Atlas and Navteq.
2. Regional Land Information System (RLIS) street centerline file, maintained by local jurisdictions in the three-county area in coordination with the Metro Data Resource Center (DRC), and distributed by the DRC.
3. OpenStreetMap (OSM), a crowd-source open dataset designed for routing. OSM has seamless worldwide coverage.

Analyzing Routable Street Network Alternatives for OTP

TriMet analyzed the three alternatives listed above while considering the following:

- Does it have the necessary attribution to support multi-modal routing for biking, walking, and transit in the Portland Metro area? Does it have attribution to support driving directions for a future phase of OTP?
- Is the linework seamless between jurisdictions and is it accurate? The GTFS transit data format facilitates multi-agency trip planning and TriMet has short and long-term goals for planning trips in conjunction with C-Tran in Vancouver, WA, and with SMART in Wilsonville, OR, which makes trips to Salem.
- Is it affordable?
- What are the maintenance and update issues?

1) **Commercial routable street networks**, such as Tele Atlas and Navteq, meet the attribute requirements for OTP. They are seamless worldwide datasets. However, these datasets have considerable disadvantages, in terms of both cost and control over the data. Annual data licensing is prohibitively costly, with estimates starting at $25,000 per year. Additional updates throughout the year add even more cost. Further, if TriMet used commercial data for routing, the agency would lose the control it now has over linework. Currently, TriMet collaborates with the local jurisdictions to maintain segments that are specific only to transit. Ability to update the data to reflect changes

critical to routing, such as street closures and new subdivisions, would be lost. For example, if TriMet began to maintain the Tele Atlas base to meet its needs, it would create problems when the source basemap is updated.

2) **The RLIS street centerline file** supports geocoding and has accurate linework based on regional ortho-rectified aerial photography. However, the only routing attribute it contains is street directionality. In addition, it is not seamless beyond the three-county boundaries. A tremendous amount of work would have to be performed to make the roads between Washington and Clackamas Counties match up to Marion County to support trips to Salem and beyond our region.

   Evaluation of this dataset for routing purposes is detailed below:

   Attributes

   - It lacks required attributes for routing.

   Linework

   - The street network is not seamless beyond the region. A great deal of effort would be required to align the linework (both from a technical and political perspective) to adjacent jurisdictions.
   - It is not lightweight. It supports many different requirements in the region and has grown over time, so there are many attributes associated with each segment. The RLIS centerline file is a very large dataset.
   - It is based on the old ArcInfo model where all linework is broken at every intersection. An enormous effort would be required to fix this; doing so may break other systems.

   Maintenance

   - The centerline file is maintained by local jurisdictions that may not have a vested interest or responsibility for creating or maintaining routing attributes.

3) **OpenStreetMap (OSM)** is designed for routing purposes. It is a free dataset with seamless worldwide coverage.

   Cons

   - It is not consistent with our regional basemap, which can cause issues with alignment downstream.
   - It is risky in the sense that it relies on user-generated data. No US government agency currently relies on crowd-sourced data, although it is successful in Europe.
   - Linework accuracy may need improvement, but some government agencies are not willing to share data for this purpose, particularly with staff viewing OSM as a competing rather than complementary data source.
   - Modifications between basemap updates will need to be rectified. While a rectification process is already in place for our regional centerline file, establishing a corresponding process for OSM may involve additional work.

Pros

- Resources required to edit and verify the attributes would be similar or less than regional centerline file.
- It is seamless across the US and based on the 2007 US Census Bureau's TIGER file, augmented by user-supplied local knowledge.
- It is lightweight.
- It is designed and used for routing purposes. For example, street segments are not broken to the same degree as the RLIS centerline file, implying less effort to improve linework.
- It is affordable (free!).
- It aligns with TriMet's open data policy.
- TriMet will be using tax dollars to make improvements to a product the public can use and benefit from.
- A large geo-community exists as a potential partner with whom to create a maintainable network.
- OTP works very well with the OpenStreetMap data structure. TriMet has automated many of the import processes around this.

After preliminary testing on the RLIS data, a second instance of OTP was released for Portland, Oregon with OSM data for comparative purposes. Further analysis concluded that the OSM street network should be used for routing rather than the local RLIS dataset.

OpenStreetMap Improvement Process

TriMet requested additional grant funds to hire a total of four student interns to begin improvements to OSM attribution and linework. The first intern was hired in November 2010 to begin research and define procedures. Three additional interns were hired in February 2011 for a six-month time-frame to perform the following improvements in OSM:

Phase 1 – February through August 2011

- Improve the geometry of linework using regional six-inch aerial photography, ensuring that the linework falls within the right-of-way.
- Add additional and missing linework: missing streets, trails, bicycle lanes, pedestrian paths, and major waterways. Ensure connectivity between segments for routing purposes.
- Add and correct attributes, including street names, street type, type of bicycle infrastructure, surface type, and transportation mode permissions.
- Verify directionality and turn restrictions for bicycle routing.

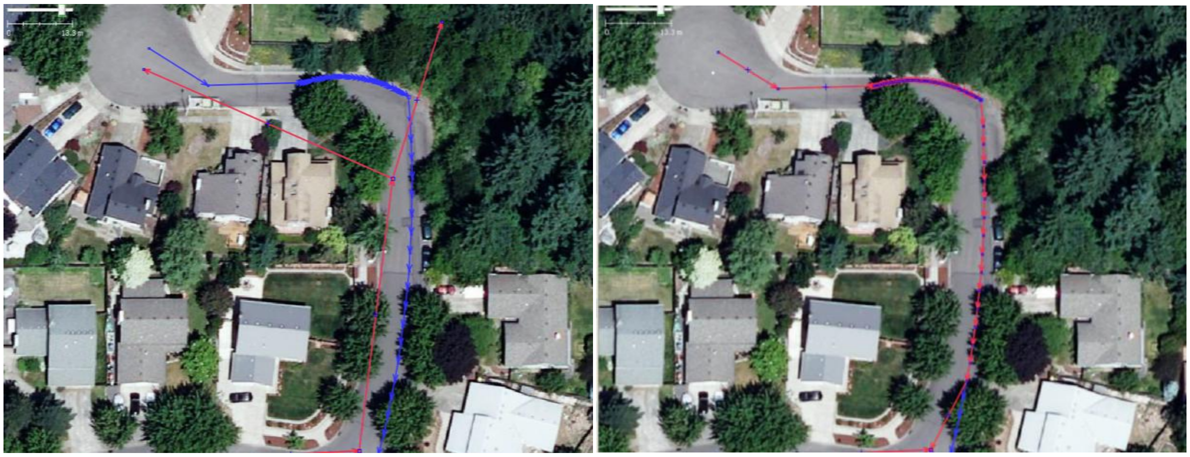


Figure 12: OSM edit session in JOSM using jurisdictional linework and aerial photography as reference.

Please refer to Appendix D for details regarding procedures and the editing processes that were used.

Geocoding Issues

Initially, geocoding was a requirement of the basemap, however, upon further research into alternatives, it was decided to use an external web service for geocoding, as opposed to using the underlying street network, which would require a tremendous amount of work and de-segmentation of linework in OSM. The pros and cons of each method are outlined as follows:

Address geocoding (via GIS) on theoretical address ranges in centerline file

This method returns via GIS/geocoder approximate XY coordinates based on interpolation of theoretical addresses. It requires a great deal of maintenance for address approximations and for accuracy, it requires the break of line segments to represent changes in address ranges.

Address search engine (Solr) on address points in database

This method returns via web service accurate XY coordinates based on database records compiled from tax lots, landmarks, intersections, etc. The accuracy of the database can improve with new records and access to the data is in real-time and does not require a new basemap import to see improvements to geocoding. An address search engine also forces the user to enter/determine accurate spelling, address, etc. providing matching selections in the dropdown menu.

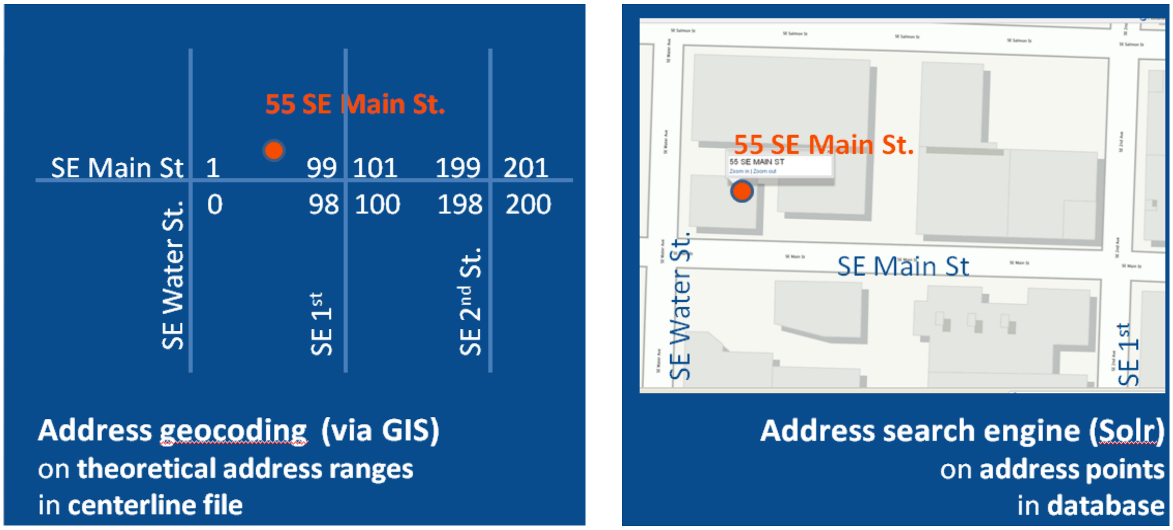


Figure 13: Comparison of address location methods: geocoding on theoretical address ranges vs. an address search engine such as Solr.

OpenStreetMap Conclusions

Based on our analysis, it was determined that OSM as base data for OTP is preferable to using standalone jurisdictional data.

Using local jurisdictional data as a reference to manually improve OSM is a very viable alternative that can be done rather quickly at a minimum cost in comparison to commercial routable networks. There is also great benefit to working with the community on updating attribution because they may have a greater vested interest in maintaining data than jurisdictions. For instance, local users who use OSM can now help improve the biking and trail networks that can benefit them directly through use of OTP and The Intertwine.

Bulk importing shapefiles or other data into OSM is not the best alternative in the US. Unlike Europe, and many other areas worldwide, Tiger was imported for the US in 2007, so rather than a focus on creating new linework that doesn't exist, the emphasis in the US is on improving the existing linework and attribution. In addition, the OSM community has made contributions that are extremely important to maintain. Imports run the risk of losing valuable community information.

The evaluations and analysis that were performed on jurisdictional data, editing processes, map editors, and scripts were important for this project, however, agencies should perform their own evaluations for new projects, as software changes frequently, and data varies between jurisdictions.

The most important skill of the student interns appeared to be knowledge of the local physical street network and the reality of biking in the region. GIS and software skills helped significantly; however, they were secondary to on-street knowledge.

Appendix D is designed to be a guideline for other agencies who are interested in improving OSM for their region.

SECTION 4: OTP TRIP EVALUATION STUDIES

Three main OTP trip evaluation studies were performed in the last year. The first study looked at preliminary transit trip testing in the region, as it did not require extensive work in OSM. The next two studies were performed in June 2011 after OSM improvements were completed in the designated test area (see Appendix B). These studies analyzed bike and bike-to-transit trips.

Upon completion of OSM data improvements in the region, more detailed testing will be conducted. TriMet plans to continue to reach out to the OSM and local bike community to facilitate collaboration around improving weight mechanisms for routing.

Preliminary Transit Trip Testing in January 2011

Public transportation in mid- to large-sized cities is extremely complex. Trip planning software may offer a number of similar itineraries from the same input. Trip plans frequently vary significantly based on time of day. A matter of five to ten minutes in either direction can dramatically change results, turning a long, transfer-heavy trip into a quick and simple trip. Trip planning software offers a tool, but a more experienced user will know to change query parameters to optimize the final result.

The TriMet Call Center assists customers who call for person-to-person assistance with planning trips. Two highly experienced customer service representatives made printouts of every successful trip they planned with the Proprietary Transit Trip Planner (PTTP) planner over two days using– approximately 250 trips. These printout trips were then planned through OTP and compared. The only significant difference between the PTTP printouts and OTP results was that OTP lacked a minimum time for allowable transfers, which has since been addressed in the TriMet OTP implementation.

OTP itineraries were found to be consistently accurate and optimal. A series of odd or poor trip plans turned out to be the result of a missing parameter: a minimum transfer interval. Any transit property not committed to timed transfers will require the ability to set this time, globally and by specific locations.

Preliminary Bicycle Trip Testing June 2011

To measure the efficacy of OTP bicycle trip planning, OTP was tested against two other bicycle trip planners, the Google bicycle router and the byCycle.org router. These two trip planners were first tested against the default OTP weighting configuration. The OTP weights were then customized, a new graph was built, and the same trips were tested again. Weights were customized so that the router would interpret a variety of OSM tags related to bicycle infrastructure and street type in generating routes. These tests are outlined below and in the weights are identified in Appendix C.

Trip selection: Preliminary testing began with the selection of 15 trips in the TriMet service area. Trips were limited to a test area (see Appendix B) where OSM tagging of bicycle-relevant data (including bicycle infrastructure, road type, and surface type) was complete. This area spans Multnomah, Clackamas, and Washington counties, and comprises all of Southwest Portland, Lake Oswego, and West Linn, extending west to include parts of Beaverton. Within this test area, 15 plausible trips were selected that would provide insight into the following questions:

- Does the OTP router choose safe, efficient bicycle routes?
- Do user specifications (quickest trip vs. safest trip) generate meaningful results?
- Are the bicycle routes simple to follow? Are the itineraries user-friendly?
- Does the OTP router break up elevation gain/loss efficiently?

Weighting: OTP users are provided with two options in planning single-mode bicycle trips, quickest trip and safest trip. Multi-modal bike-to-transit trips include a third option, fewest transfers. For the bicycle component of each type of trip, weights are used to identify optimal bicycle routes. Weights have the effect of changing the length of an OSM way/line, as perceived by the OTP router, to reflect the way's suitability for cycling. Bike-friendly streets are given weights between 0 and 1 to effectively shorten the way. Conversely, streets that are unsuitable for biking are given weights greater than 1 to effectively lengthen the way. An optimal weighting configuration is achieved when the shortest path, as perceived by the router, is equivalent to the most suitable route for the bicyclist. By default, OTP provides no weighting for the quickest trip and limited weighting (only considering the presence of bike lanes) for the safest trip. The routing implications of this default weight configuration are discussed below.

Testing default configuration: First-round testing generated single-mode bicycle trips using OTP's default weighting configuration. For each trip, two OTP routes, quickest and safest, were generated, along with a Google bicycle route and the byCycle.org router, selecting only 'normal' trips for the latter.

The default weighting configuration does not have any weights for quickest trips. This means that the router lacks criteria for determining bicycle suitability and selects routes based solely on shortest geometric distance. Unsurprisingly, the quickest trip OTP bicycle router did not generate viable bicycle trips. Blind to the presence of bicycle facilities and street type, the router missed opportunities to guide cyclists towards low-traffic streets with bicycle infrastructure. Instead, it favored high-traffic arterials since these are often the most direct routes. The other two bicycle routers consistently yielded safer, more sensible bicycle routes for the same trips.

The default weighting configuration for safest trip, by contrast, contains weights for the presence bike lanes. As a result, the router was aware of bike lanes and routed cyclists accordingly. The OTP safest trip router generated more acceptable bicycle routes. However, the other trip routers tended to generate better routes because they likely involve more nuanced routing criteria, including awareness of other types of bicycle facilities, such as multi-use paths, as well as a preference for low-traffic streets.

Moreover, the 'safe' routes generated by OTP were not necessarily the safest, which is to say that even safer routes exist that further reduce cyclist interaction with motorized vehicles.

Generally, first-round testing found that OTP quickest trip results were unacceptable and that safest trip results resembled the routes that ought to be generated for quickest trip. *These first-round results highlight the importance of weighting OSM ways to reflect suitability for bicycles. They also show that the default weight configuration should not be used by agencies, particularly for quickest trip.*

Testing customized weight configuration: Second-round testing began by customizing a set of weights related to two types of bicycle trip planning variables: bicycle infrastructure type and street type. These customized weights give preference to OSM ways tagged with bicycle facilities, including bike lanes, bicycle boulevards, and separated multi-use paths or cycle ways. The weights also prefer OSM ways tagged as low-traffic residential streets and penalize (lengthen) streets tagged as high-traffic arterials. TriMet interns determined values for these custom weights by consulting an empirical study that used GPS to document cyclist road preferences ("Calibrated Labeling Method for Generating Bicyclist Route Choice Sets Incorporating Unbiased Attribute Variation," by Joseph Broach, John Gliebe, Jennifer Dill. Pages 89-97 in Transportation Research Record 2197, published in 2010.). The complete list of weights is provided in Appendix C.

With customized weights, OTP routes showed remarkable improvement. OTP now generates viable bicycle routes that compare favorably with both the Google bicycle router and byCycle.org. Quickest routes tended to utilize bicycle lanes on high-traffic through streets, while safest trips prefer residential streets and separated paths. Evaluation of second-round testing concluded that OTP produced bicycle routes that were as good as or better than existing bicycle trip planners for 7 of 15 single-mode trips. The other 8 trips were viewed as inferior primarily because OTP bicycle routes were overly complex, involving unnecessary turns, or because OTP itineraries presented confusing, overly-detailed information to the user. OTP consistently produced itineraries with more steps that Google or byCycle.org even when the routes themselves were geometrically equivalent. This is due to peculiarities of OSM data. OSM stores a relatively large number of unnamed service roads, paths, and other fine-scale linework that is lacking from Google and the byCycle.org datasets. OTP uses these unnamed ways in routing, as it should, but it adds a step to the itinerary for each time an OSM way changes name, even for negligible distances on service roads (which is unnecessary). Over-reporting of negligible turns has the effect of inflating the number of steps in an itinerary, effectively providing too much detail to the user, resulting in a trip plan that is difficult to remember and follow.

Results: Preliminary testing of OTP single-mode bicycle trip planner yielded several insights:

- OSM is capable of storing valuable, routable information related to bicycle routing
- OTP is capable of combining OSM data and elevation data to produce viable bicycle routes
- Weighting will be critical to generating optimal bicycle routes in OTP

- OTP needs further development to generate user-friendly itineraries from OSM data. This is being addressed for the TriMet OTP public release in October 2011.

Second-round testing yielded OTP bicycle routes that are roughly comparable with existing bicycle trip planners, indicating that more complex weights are critical to bicycle trip planning. While further refinement of weights could yield even better routes, the greatest development need lies in improving how the OTP router interprets OSM data. Routes should minimize the number of turns so that users find them easy to follow. Itineraries should be smoothed so as to avoid over-reporting of unnecessary details. These improvements should make OTP more intuitive and user-friendly.

Regarding the questions posed in the process of trip selection, preliminary testing results are as follows:

- **Does the OTP router choose safe, efficient bicycle routes**? Yes, assuming OSM contains complete and accurate data on bicycle infrastructure and road type, as is the case in the test area.
- **Do user specifications (quickest trip vs. safest trip) generate meaningful results?** Yes, once the specific values selected for each weight are given careful consideration and several rounds of testing. TriMet recommends backing up old versions of the graph so that routes generated using different graphs can be visually compared for fine-scale refinement of weights.
- **Are the routes easy to follow**? **Are the itineraries user-friendly?** Not initially, however, improvements are being made and tested in preparation for the public release in October. Testing showed that the other bicycle routers were superior in terms of generating simple, easy routes with fewer steps than OTP. Empirical studies have shown that cyclists prefer to reduce the number of turns as part of the routing process. OTP could minimize the number of turns in routing. In addition, OTP needs improvement in how it interprets OSM data, particularly for condensing itinerary steps that involve unnamed ways.
- **Does the OTP router break up elevation gain/loss efficiently?** Yes. OTP results are comparable to existing bicycle trip planners, particularly for quickest trip.

Preliminary Bicycle-to-Transit Trip Testing June 2011

Overview: The same 15 trips were used as in the single-mode bike testing. For each trip, three trip planning options were analyzed: quickest, safest, fewest transfers. Each option, in turn, generated 3-4 routes, for a total of at least 12 routes per trip. These trips were then compared with Google Transit and ATIS. Each of these routes was evaluated against the following criteria:

- How many steps are involved in the narrative? Routes should be simple.
- Do the bicycle and transit components of the routes make sense?
- Do they conform to physical/legal turn restrictions?

- Do user specifications generate meaningful results (does quickest trip really generate the quickest trip)?
- How long from departure to arrival at destination? How long biking? How long on transit?

Results: 15 out of 15 trips were faster with OTP vs. single-mode transit trip planners. Bike-to-transit allows the user to access more transit. This can dramatically reduce travel times, particularly in low-density areas with infrequent transit service.

OTP generates more options for the user since each trip planning option (quickest, safest, fewest transfers) usually generates several results.

OTP can generate a multi-modal route for trips that can't be routed by single-mode trip planners. Three of the fifteen trips could not be routed through the PTTP because they began at points that were more than 1 mile from a bus stop. OTP generated efficient bike-only or bike-to-transit routes for these trips.

Improvements to OTP: Many of the OTP improvements suggested by preliminary single-mode bike testing hold for multi-modal:

- Consolidate number of steps in narrative
- Find a more user-friendly approach to reporting unnamed ways in OSM, reduce narrative clutter

New issues for multi-modal:

- Snapping: improving where the trip starts and ends if the user selects a start point that is not on the road
- Quickest and safest trips were frequently the same, which may indicate issues

Additional data issues specific to OSM:

- Access tags to indicate where cyclist must walk, dismount and carry bike down stairs
- access=license needs to be applied to OSM ways in areas where users must pay to enter (e.g. the zoo)
- Turn restrictions in OSM will improve routing

SECTION 5: NEXT STEPS

Metro RTO Grant Phase II

Metro extended the original Metro RTO Grant for an additional two years and awarded TriMet with $75,000 to take the existing OpenTripPlanner (OTP) from a demonstration prototype to a functioning open source, multi-modal trip planner. The objective of this next phase is to create a more complete and enhanced user experience which will include additional functionality, such as a mobile version and management tools for both the developers and the supporting administrative staff for continued maintenance of the application and data.

TriMet and the contractor, OpenPlans, will continue to lead the project and support it with substantial in-kind contributions to ensure the successful implementation of a system that can be supported and maintained.

OTP Metro RTO Grant 2011-2013 Target Goals

- Incorporation of Alerts into the planned trip itinerary
- Ability to print output
- Bike Preference Triangle
- Improvements to the descriptive itinerary
- Resolution of bug issues
- Speed & performance improvements
- Email and SMS output
- Ability to calculate fare
- Analytical tools and batch trip planning
- Multiple language support beyond current translations
- Return and prioritization of multiple itineraries
- Extend the availability of the service to mobile devices
- Trip disability options
- Create administrative tools for staff to continue maintenance of the application and data. These include easy and well-documented installation, deployment, and data loading. Routing support for transit, walking and biking all implemented. Improve stability of the API.
- Continue to document information such as lessons learned and feedback from participating partners, which will be useful for future government open source software collaborations.

July 2011 Kick-Off Workshop

TriMet and OpenPlans held a second workshop July 13-15, 2011 to kick-off the next phase of the project. As with the first workshop, interested and participating developers and agencies were invited to participate. The results of this workshop will be used to develop the project plan for the following two years, and to identify and prioritize requirements and that can be delivered within the two-year timeframe and budget. Following this workshop opentripplanner.org was moved to https://github.com/openplans/OpenTripPlanner/wiki/.

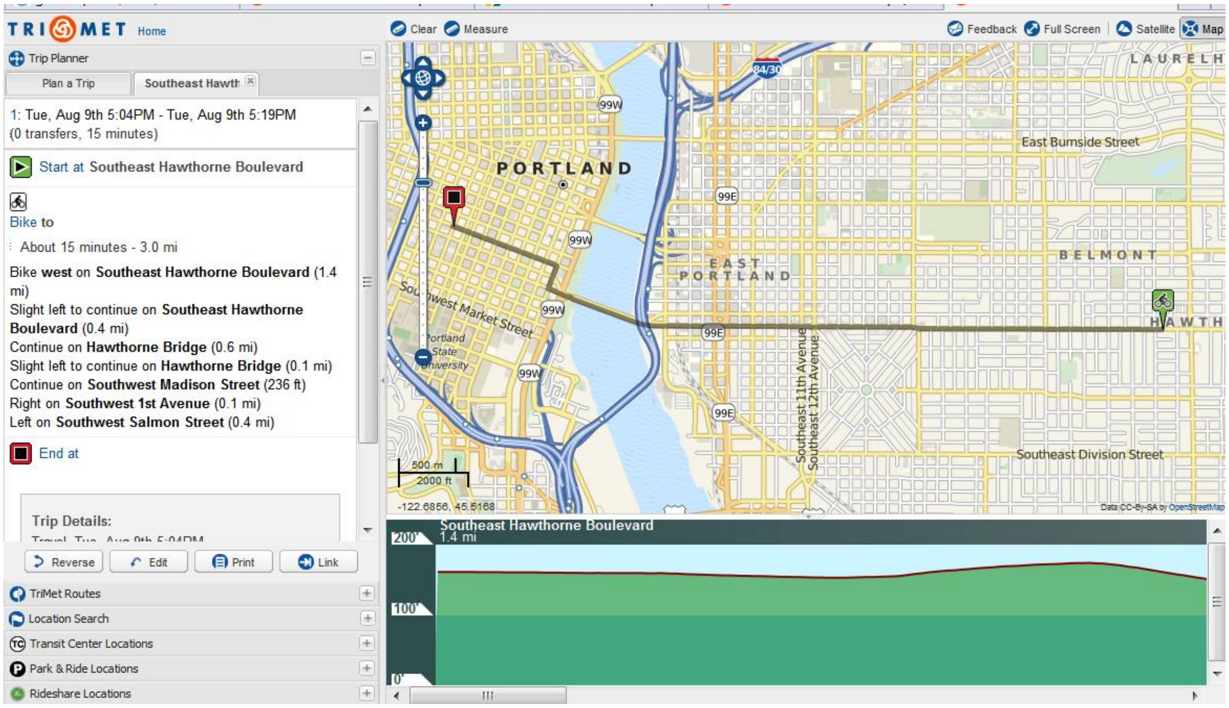
OTP Fall 2011 Public Beta Launch

A public beta launch of OTP is scheduled for October 2011. TriMet will work with Metro and other partners to develop a marketing strategy to unveil the planner to the general public, as well as targeted audiences.

The interface of TriMet's OTP implementation is nearing completion in regards to replacing the PTTP routing engine with the new OTP backend (a second version of maps.trimet.org). There will be several new additional map features in the formal public release that do not exist in the current map: 1) the bike elevation chart, 2) the bike preference triangle, 3) the Rideshare Locations tab, 4) the ability to plan multi-modal trips. New features will be kept down to a minimum for comparative purposes, to reduce the learning curve, and to avoid scope creep.

Assuming the mandatory requirements are met by the public beta, we will take the approach of rolling out new features periodically in bundles. This can have several advantages: it can keep users coming back to the site; we can extend the marketing campaign with new features; we can demonstrate progress; the users can learn progressively along with the development of the app; and we can reduce the amount of negative comments by letting users know what we are working on.

We will establish a mechanism to review, organize, and direct comments to the appropriate staff prior to the public release. A defined set of categories for the comments will enable comment analysis: general comments (likes, dislikes); comparative trip comments (vs. PTTP or Google); new user comments (is it encouraging them to bike/walk?); interface and feature comments; new feature requests; etc.



1: Tue, Aug 9th 5:04PM - Tue, Aug 9th 5:19PM
(0 transfers, 15 minutes)

Start at Southeast Hawthorne Boulevard

Bike to

About 15 minutes - 3.0 mi

Bike west on Southeast Hawthorne Boulevard (1.4 mi)
Slight left to continue on Southeast Hawthorne Boulevard (0.4 mi)
Continue on Hawthorne Bridge (0.6 mi)
Slight left to continue on Hawthorne Bridge (0.1 mi)
Continue on Southwest Madison Street (236 ft)
Right on Southwest 1st Avenue (0.1 mi)
Left on Southwest Salmon Street (0.4 mi)

End at

Trip Details:

Figure 14: Screen capture of a preview of the OTP October 2011 public beta release with new features including a bike elevation chart, Rideshare Locations tab, and multi-modal functionality.

Continued OSM Improvements

TriMet will schedule OSM “Mapping Parties” with the local community to assist with testing and feedback prior to the fall release. OSM data accuracy is a major priority for OTP and improvements and testing will continue.

Working with local jurisdictions and the community on long-term maintenance plans will also continue.

TriMet has identified additional funds to retain the interns for the fall to continue improvement in OSM in support of TriMet operations. TriMet’s new Computer Aided Dispatch/Automated Vehicle Location (CAD/AVL) System and the Mobile Data Terminals (MDTs) in vehicles also require a routable network, however, additional features need to be added to OSM in support of vehicular and bus navigation.

APPENDICES

Appendix A – High Level Open Source, Multi-Modal Trip Planning System Requirements

TriMet OS MMTPS High Level Requirements[5]
December 9, 2008

Grant Requirements:

- Should demonstrate reduction in vehicle miles driven and single-occupant vehicle (SOV) trips
- Increase multi-modal trips – walking, biking, transit
- Evaluation Study required after implementation

Short Term Project Objectives:

- Implement Five Points' A-Train application for the Portland area using TriMet's datasets
- Contribute back source code improvements and documentation for the benefit of others
- Increase and strengthen the development community around the code to ensure continued support
- Demonstrate the feasibility of an OS MMTPS implementation
- Compare the accuracy of the routing algorithms between the following: Five Points, GraphServer, TriMet's Proprietary Transit Trip Planner (transit only)
- Compare bike-only routes with byCycle.org
- Review scalability issues

Long-Term Objectives:

- Must be feasible for transit agencies to implement this application, which means flexibility regarding system deployment upon a modest server infrastructure
- Should be modular in design, with individual components (e.g.: routing, geocoding, etc...) able to be run as standalone processes
- Shouldn't require a particular DB in the long term but should support PostGIS first
- Ideally platform independent, but GNU/Linux/OSS should be supported first
- Develop tools to support data loading, especially for bike data/routing (perhaps develop a "bicycle feed spec")

Functional Requirements:

- Calculate the following routes:
  - Walking only
  - Biking only
  - Bike to Transit
  - Walk to Transit
  - Bike + Transit shortest and fastest routes
  - Drive Only & Drive to Transit – Should have capacity to include this in the future, however, our basemap does not support this (no street direction, turn restrictions, or speed limits data)
- Utilize Portland's regional street base – will evaluate OSM and other alternatives

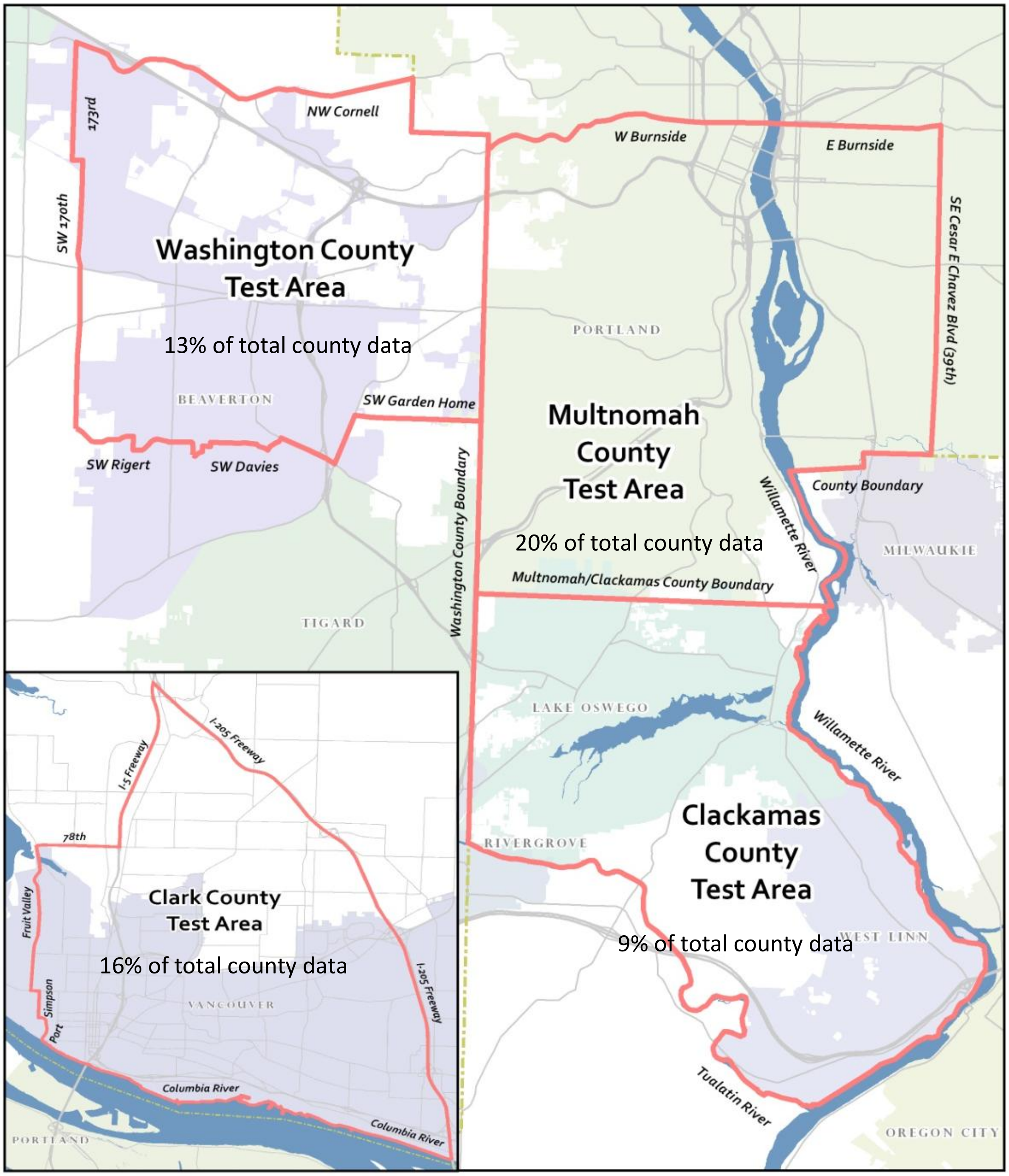
[5] Since the time these initial requirements were gathered, it was determined that the Portland instance of OTP would utilize OpenStreetMap, not the RLIS street centerline file, and that a web search engine would be used instead of a geocoder that uses address ranges in the basemap.

- Utilize transit data in the GTFS format for transit routing
- Utilize the National Digital Elevation Dataset for topography charts
- Develop a standard model for bike data
- Must have options to change the following variables:
    - Time and day of travel
    - Quickest Trip
    - Fewest Transfers
    - Shortest Walk
    - Walking distance radius (0.25 – 2.0 miles)
    - Walking speed (8-20 MPH)
    - Biking distance radius (0.25 – 2.0 miles)
    - Biking speed MPH (2-3 MPH)
    - Bike Factors – topography adjustment, safety/road adjustment – see prototype of bike factor triangle at http://new.atltransit.com/
    - Plan bus only transit trips and train only transit trips (see TriMet trip planner)
- Select Start and End location by selecting on map or by keying in an address
- Route Explorer Feature – see http://routes.atltransit.com/
- The routing engine will include options to refine a given itinerary that includes transit, where the routing engine will further calculate either the first or last possible trip of the day for a given itinerary, as well as the next possible trip (with an offset in time, as an addition parameter to the itinerary being refined).
- High want is the ability to plan a multiple leg trip, with pauses in between each leg, ala Google's driving directions, and its ability to add additional destinations to a given trip plan. The ability to drag line to change route is also desirable.
- Must have interface to maintain bike data
- Average query response time should be reasonable: around 1 second for itinerary generation for the core algorithm.
- Work towards a standard schema for trip planning (both inputs and outputs).

System/Design Requirements:

- Produce a set of RESTful web services that enable trip planning. Atop these services, a reference implementation will consist of both a map and (printable) text itinerary UI (envisioned is a set of complementary RESTful web services, which support at least JSON or XML payload formats, that are able to deliver a trip plan based on human input trip queries)
- If an external database is used, Postgres/PostGIS will be (one of) the standard option(s).
- Platform – Linux and Windows
- NOTE: The deployment requirements should aim for flexibility, and not dictate or require a specific operating system or a specific database. At a minimum, the system will run on both Unix/Linux and windows systems, and if a third party database is required, support given for at least Postgres.
- Language – Java (with deployment to the Tomcat level of J2EE). Secondarily, Python could be an option.
- Application should be designed to be scalable for larger agencies.
- System Architecture should be modular in design for interchangeability - client application must be easily interchangeable
- Must include a reference client application

Appendix B – Map of Test Area for OpenTripPlanner



Appendix C –OTP Bicycle Safety Weights

The OTP bicycle safety weights are controlled by OSM tags by default as follows but can be customized:

```
<entry key="cycleway=lane"                value="0.3,0.3" />
<entry key="cycleway=opposite_lane"       value="1.0,0.3" />
<entry key="cycleway=opposite"            value="1.0,2.0" />
```

Customized weights for bicycle safety as follows showed remarkable improvement is trip results:

```
<entry key="cycleway=lane"                value="0.87,0.87" />
<entry key="cycleway=opposite_lane"       value="1.0,0.87" />
<entry key="cycleway=opposite"            value="1.0,2.0" />
<entry key="cycleway=shared_lane"         value="0.77,0.77" />
<entry key="cycleway=track"               value="0.73,0.73" />
<entry key="bicycle=designated"           value="0.95,0.95" />
<entry key="bicycle=yes"                  value="0.95,0.95" />
<entry key="RLIS:bicycle=caution_area"    value="1.6,1.6" />
<entry key="highway=service"              value="0.87,0.87" />
<entry key="highway=residential"          value="0.87,0.87" />
<entry key="highway=tertiary"             value="1.0,1.0" />
<entry key="highway=secondary"            value="1.18,1.18" />
<entry key="highway=track"                value="1.3,1.3" />
<entry key="highway=primary"              value="2.06,2.06" />
<entry key="highway=path"                 value="0.73,0.73" />
<entry key="highway=cycleway"             value="0.73,0.73" />
<entry key="surface=unpaved"              value="1.3,1.3" />
<entry key="cycleway=opposite_track"      value="1.0,0.87" />
```

Appendix D – Detailed Procedures for Portland OSM Improvement Project

A student intern was hired in November 2010 to research methods and determine technical editing procedures for OSM improvements. Three additional interns were hired in February 2011. All four interns were hired for a six-month project duration. The task was to use local jurisdictional data to improve OSM to support multi-modal biking, transit and walking trips in OTP, and in support of the regional four-county InterTwine Project. Each intern was assigned a county to be responsible for based on their local knowledge of the street network within that region.

These procedures are outlined in the following four parts:

1) Groundwork and Preparation - includes creating OSM accounts, establishing communication tools, and learning the basics

2) Evaluations and Results - includes editing processes, map editors, and scripts

3) Detailed Editing Procedures

4) Quality Control Process

Part 1 – Groundwork and Preparation

Created OSM Map Editing Account and OSM Wiki Account

To stay consistent with other OSM user accounts, intern names were used, not agency names. Work can be tracked by searching for RLIS or CCGIS in the tags.

OSM: http://openstreetmap.org
OSM wiki: http://wiki.openstreetmap.org/

Established Communication Tools http://openstreetmap.org/

Joined local OSM group

- osm-pdx

Joined talk and talk-us: OSM general and American listservs

- Communicated plans and directed questions to these listservs.
- Learned to approach community respectfully using questions and not statements. As a government agency, it's important to ask permission and input from the entire community, even if the local community is small.
- Other lists that may be of interest can be found at http://lists.openstreetmap.org/listinfo

Set up OSM user page

- http://www.openstreetmap.org/user/PJ%20Houser
- This page to communicate agency involvement, plans, and general schedule.
- This page can also be used to find local mappers and "friend" them.
- Finally, this page also has a "diary" that can be used to communicate with the entire community (under the tab "User Diaries"). This diary was used to make an introductory statement with TriMet's plans and also to invite community members to a formum.

Discovered local active mappers

- Some local mappers can be found using the user page's near function or OSM Mapper by ITO at http://www.itoworld.com/static/osm_mapper.html

Created OSM wiki pages:

- Wiki pages were created for the datasets. These pages are updated as data becomes available and tag conversions are decided upon. The pages should include an overview of the data, the dataset license, planned/completed process, and attribute conversions.
  - http://wiki.openstreetmap.org/wiki/RLIS
  - http://wiki.openstreetmap.org/wiki/CCGIS
- Updated relevant wiki pages, as listed below.
  - http://wiki.openstreetmap.org/wiki/Potential_Datasource
  - City and state pages
  - http://wiki.openstreetmap.org/wiki/OpenTripPlanner
- Set up Meet & Greet with local community on Friday May 13, 2011. This meeting was intentionally set this up after the process of editing OSM data based on jurisdictional data had been determined and tested. The Meet & Greet was used to answer questions about the process and the OpenTripPlanner, and to get input on further editing, such as bike routes and waterways.

Studied OSM Basics

Reviewed OSM wiki

- http://wiki.openstreetmap.org/wiki/Beginners'_Guide
- http://wiki.openstreetmap.org/wiki/Import
- http://wiki.openstreetmap.org/wiki/Open_Database_License
- For more in-depth information on a topic, click “Discussions” tab at top of wiki page

Read OpenStreetMap: Using and Enhancing the Free Map of the World by Frederik Ramm, Jochen Topf, and Steve Chilton. UIT Cambridge 2011.

Gained experience. First intern made some small edits in Potlatch and JOSM based on local knowledge. Reviewed changes at http://openstreetmap.org

Part 2 – Evaluations and Results

Jurisdictional Data

Jurisdictional datasets are seen as the authoritative data, however, there are only a few maintainers with thousands of segments to maintain. TriMet’s datasets, RLIS and CCGIS, are primarily used for geocoding rather than routing.

Visual inspection:

- RLIS street data was overlayed on top of OSM data in ArcMap with Metro’s aerials behind them and the differences were visually checked.

Software inspection:

- ArcMap was used to select RLIS segments over 10 m from OSM ways. A significant portion of the street network was selected, enough to cause a reconsideration of our priorities.
- As a result it was decided to only focus on streets and trails that are part of a bike or transit route. (If time allows, later focus will be on non-bus and non-bike routes that are major roads. Then, if time continues to allow, focus work on insignificant segments.)
- Assume the aerials to be the most accurate because they are 6 inch ortho-rectified aerials. Additionally, they are found to be within 3 - 5 m of Bing’s aerial imagery.

Open Database License: data used in OSM needs to be compliant with OSM’s ODbL, found at http://www.osmfoundation.org/wiki/License/Contributor_Terms. RLIS and CCGIS are both open to the public and compliant with OSM’s ODbL.

Editing Process
It was determined that the best editing option for the street base was a manual process, over importing or conflating. See table below for the selection process.

Manual edits refers to the visual inspection of each OSM way's alignment, directionality, topology, and attributes with RLIS or CCGIS data and aerials, to determine if the OSM way is accurate, and changing, adding or removing ways, direction, or attributes where necessary.

Edit Process	Pros	Cons	Our Experience	Resources
Import	- Good for datasets that are not represented in OSM yet or very inaccurately represented (i.e. waterways in many areas) -Quickest	-Difficult and prone to errors -Distrusted by the OSM community -Most of the time is spent prepping the data and testing the import process on test OSM servers. -Not to be used for areas already consisting of user-edited data, i.e. street networks in most cities.	We decided against importing our data because: -RLIS and CCGIS are not always more accurate than OSM; therefore, we needed to inspect each difference and decide which dataset is most accurate based on aerials. -Imports are highly technical, and the intern did not have sufficient technical experience to supervise the process. -Our area of interest, the Metro region, already contains most of the road network. -Importing the data would overwrite many users' changes, infuriating the community.	http://wiki.openstreetmap.org/wiki/Import Imports listserv at http://lists.openstreetmap.org/listinfo/imports
Conflation via Software	-Reduces time spent looking for errors in OSM -May be more accurate than visual analysis -Some experience already exists. According to the wiki, one conflation software package - RoadMatcher - has been used with at least two OSM imports. http://wiki.openstreetmap.org/wiki/G	-Getting conflation software to work with OSM data could be difficult. -As of early 2011, ArcMap OSM Editor causes unstable conditions when using OSM data (converted to a geodatabase). -As of early 2011, RoadMatcher would not work for TriMet. In terms of Java, it gave a null-pointer exception when loading a small sub-set of data.	We tried a couple software packages that included conflation, but were unable to successfully use RoadMatcher or ArcMap's OSM Editor.	http://wiki.openstreetmap.org/wiki/Conflation

Edit Process	Pros	Cons	Our Experience	Resources
	eoBase http://wiki.openstreetmap.org/wiki/Albania_TPGInc_Import/Roadmatcher			
Manual Edits	-This method allows for comparison of OSM and jurisdictional data and allows the editor to decide, based on aerial imagery and other considerations. For example, in the OSM community, local knowledge by OSM users trumps jurisdictional maintainer data. -This method limits the risk of overriding OSM users' data and inflaming the community.	-Requires the most time (and therefore, funding). The work can be tedious, but requires alertness and wisdom in making the right choices for hours at a time. -If multiple editors are hired, methods may not be as uniform as a software solution.	We used RLIS and CCGIS data, datasets that are reasonably well maintained and ODbL compliant, along with high resolution aerials (with permission to derive ODbL data), and OSM data. First, roads or ways were examined and determined to be important based on bus and bike routes, and adjusted their alignment, directionality, and attributes based on the editor's discretion. Missing data in OSM from jurisdictional data were copied and pasted. Corrected data was tracked and examined by maintaining a diff file. This was done by creating a shapefile in ArcMap containing only the ways that were considered important - we selected by attribute, location, etc from the jurisdictional data; extracted to new shapefile. In our case, we modified ways that fell on a bus or bike routes. The shapefile was then converted to .osm format. As ways are examined and corrected, the corresponding ways in the diff file are deleted. To make editing easier in JOSM, we duplicated and converted the .osm diff file to .gpx format, and set the lines thicker and brighter for highlighting. Lessons learned: Make sure to keep versions of the diff file in case of errors during upload.	

Map Editors - http://wiki.openstreetmap.org/wiki/Comparison_of_editors

Editor	Pros	Cons	Our Experience
JOSM http://josm.openstreetmap.de/	-Most common for more advanced users. -Includes a wide variety of plugins for flexibility. -Includes aerials, from Bing to customizable WMS layers -Allows layering of .osm, .gpx, and aerial files. -The documentation and support system is sufficient. -New versions are regularly being released.	-Difficult for a non-technical user to set up because of Java. -Default allocated memory may be insufficient for multiple layers and aerials. -Symbology of .osm layers is difficult to change -Requires time to learn the software.	JOSM has good reviews from many in the OSM community. We also liked its flexibility, and it provided most of our needs. However, we couldn't load shapefiles or compare layers algorithmically, so we had to use conversion software and ArcMap.
ArcMap OSM Editor	-We are already familiar with ArcGIS. -Most agencies already use ArcGIS. -ArcMap has a lot of flexibility and tools.	-The OSM Editor is slow at loading OSM data. There is an intentional throttle on the speed so that slower computers can use the software. Until this is changed, the speed is insufficient for large datasets, like a city's street network. -Additionally, OSM Editor crashed the computer regularly, significantly wasting time restarting and reloading data. The Editor converts .osm format to a geodatabase format, but any time OSM data existed in the map editor, despite being in geodatabase format, the software and even the computer would regularly slow and crash. -Finally, as of early 2011, OSM Editor could not edit relations. It can keep track of relations, but it cannot edit them.	-We worked with ESRI on the above problems, but could not wait for a more stable version to be released. Potentially, ArcMap OSM Editor could be great for conflation routines. -We did use ArcMap to create diff files, the files we used to track the OSM and jurisdictional segments that needed checking and editing. --First, we selected all OSM ways that fell over 10 m from RLIS. This diff file was used to highlight the OSM ways in JOSM that needed editing. However, OSM ways are longer than RLIS segments, so it was difficult to find the error. --Thus, we decided to select RLIS ways that fall over 10 m from OSM (or are missing from OSM) because this pinpointed the errors. --However, after a few weeks of editing, we discovered that RLIS and CCGIS are not always accurate. --Finally, we created a diff file

Editor	Pros	Cons	Our Experience
			consisting of only RLIS or CCGIS data that we thought important. We selected bus and bike routes (or any other attribute we'd like to isolate), and extracted a shapefile of just those. This was converted to .osm and used to highlight all important ways in OSM that needed to be manually checked. Not loading OSM data into ArcMap made ArcMap much more pleasant to work with.
Potlatch	- Simple	-Far too simplistic. -Cannot layer datasets. -Not enough flexibility.	Potlatch is a good way to get beginners' experience with OSM. The interns used it for initial experimentation when they first started.
PostGIS	We didn't explore this route, but potentially, PostGIS could be a great way to deal with jurisdictional and OSM data simultaneously.		
Other Editors	We briefly looked at other editors, like Quantum GIS and Mapzen and Merkaartor, but JOSM worked well from the beginning, so the others were not worth pursuing.		

Script Comparisons

Scripts	Pros	Cons	Our Experience	Resources
RoadMatcher Conflation plug-in for OpenJUMP	-Allows for manual and automated matching, merging, and adjusting of road segments. -Used during a couple imports (see RoadMatcher resources).	-Did not work in our experience.	As of early 2011, RoadMatcher would not work for TriMet. In terms of Java, it gave a null-pointer exception when loading a small sub-set of data.	http://www.vividsolutions.com/products.asp?catg=spaapp&code=roadmatcher http://wiki.openstreetmap.org/wiki/RoadMatcher
Osmosis	-Osmosis breaks up large	-Requires knowledge of	-We use Osmosis to break a large OSM file into smaller manageable	http://wiki.openstreet

Scripts	Pros	Cons	Our Experience	Resources
	sets of OSM data into smaller, more workable sets. -Osmosis tasks can be scripted, thus repeated on a nightly basis with no involvement once set up properly.	Java and command line. -A second visual tool (JOSM) is needed to view the extracted data. When creating our set-up, it took a lot of back-and-forth between Osmosis and JOSM to find the proper bounding-boxes. If Osmosis and JOSM were integrated better (say an Osmosis plug-in to JOSM), it would streamline the setup for 20 bounding boxes (each containing 10-15 mb of data).	chunks (too large to be usable in JOSM). We download that large file from the OSM XAPI web service on-demand on a nightly basis OSM data needs to be very current for editing. We found that editing from out-of-date .osm files can, upon upload, generate numerous conflicts between edited data and the most current version of OSM. Even nightly downloads may not be recent enough, so in most cases, we download the most up-to-date data directly from the OSM server when editing. The server only allows small areas, so Osmosis is not always needed	map.org/wiki/Osmosis
ogr-to-osm	-Converts any OGR supported vector format, such as ESRI shapefiles, into .osm -Capable of transforming between coordinate systems -Python script	-Requires some knowledge of Python and command line	We use ogr-to-osm to convert shapefiles to .osm. It works most of the time. However, sometimes attributes are lost in the conversion process. We found that adding one or two blank attribute columns at the end of every shapefile's attribute table prevented actual attributes from being lost in the conversion. We also found that if something is corrupted in the shapefile, the conversion will fail, and the error is not always clear. Despite this, we continue to use the script because it works most of the time.	http://wiki.openstreetmap.org/wiki/Ogr2osm
shp-to-osm	-Converts shapefiles to .osm -Can convert attributes to osm key-value pairs.	-Requires knowledge of the script (in our case, Java) and command line -Multiple versions in multiple languages - not all the versions work -Doesn't convert coordinate system	We tried a shp-to-osm script to convert ESRI shapefiles to .osm, but the version didn't work correctly. It lost a lot of data in the conversion, making the result unusable.	http://wiki.openstreetmap.org/wiki/Shp2osm http://wiki.openstreetmap.org/wiki/Shp-to-osm.jar

Part 3 – Detailed Editing Procedures

1. **Convert jurisdictional data to OSM key=value format**

   OSM attributes are called tags, using the form key=value, i.e., highway=primary. Jurisdictional attributes do not always have the same criteria as OSM attributes, so they need to be converted.

   Street types:

   - OSM street types are based on United Kingdom classifications rather than American standards. Converting RLIS & CCGIS street types to OSM street types proved somewhat difficult. Additionally, RLIS & CCGIS street types seem to be based on physical characteristics while OSM street types tend to be based on functionality and physical characteristics. To make our conversion table, we used jurisdictional metadata, the OSM wiki, examples in OSM, and the OSM community. Once we started editing, we continued to change OSM types on a case by case basis. For example:
     - Residential streets with a stripe on the road were upgraded to tertiary in OSM.
     - The OSM street type "trunk" is difficult to define for American roads because trunk is a European designation. We had to aggregate our own standards for trunk. We defined trunk as a highway with occasional grade level intersections and limited access, but not quite as limited as a motorway (freeway). In some states, non-motorized vehicles are not allowed, but in many states, including Oregon, many trunks and motorways allow non-motorized vehicles; therefore, we could not use this as a defining characteristic.
     - Due to rendering, it's important to maintain consistency in a street type and limit the number of times a street type changes for a way.

     http://wiki.openstreetmap.org/wiki/Highway

   Directionality:

   - RLIS maintained directionality via the start/end points of the segment, and an attribute that determined if the directionality was two-way, one-way in the direction of the topology, or one-way in the opposite direction. In ArcMap, we gave all one-way segments an attribute of oneway=yes, while leaving the value blank in two-way streets because JOSM deletes attributes without a value. In other words, two-way streets are considered the default. We also made an attribute indicating if the one-way segment needed to flip start/end points to topologically show the correct directionality. Once converted to .osm format, we selected all segments needing to flip directions, reversed them in JOSM, then deleted the attribute. In this manner, all one-way segments pointed in the direction of travel.

   Name:

   - Initially, we converted the name parts (directional prefix, base, type, and directional suffix) and the full name to OSM form. This is how the TIGER data was originally converted - http://wiki.openstreetmap.org/wiki/TIGER_to_OSM_Attribute_Map
   - We eventually decided that the name parts are unnecessary tags because geocoding will be done using a separate tool drawing upon a separate database of points and addresses in latitude, longitude format. OSM does not have an appropriate data model for geocoding.
   - OSM does not use abbreviations, so all directional prefixes and road types needed to be expanded to their full name, i.e. NW to Northwest.

   Zip codes:

   - We maintained the adjacent zip codes as attributes zip_left and zip_right because the TIGER conversion maintained the zip codes. However, because OSM is not set up for geocoding with its own database, the zip codes are probably unnecessary.

   RLIS-specific:

   - We added RLIS:reviewed and RLIS:localid to each feature to indicate the source and maintain a connection to the original data. RLIS:reviewed=no indicates that the data is based on the RLIS dataset, but no OSM user has ground-truthed that particular feature. This convention is based on the tag TIGER:reviewed, which was used in the TIGER import. We later learned that this convention is rarely used by OSM users, and often stays as TIGER:reviewed=no, even if the feature has been edited by multiple users.

Bike:

- Bike routes can be controversial. We tagged for bike routes because OTP needs to know what streets or paths can be accessed by cyclists and which streets or paths are safest if a user indicates he/she prefers safer routes. Safety can be subjective, so we tried to use tags that did not indicate preferences, but objective observations.
- RLIS has multiple types of bike routes - some have infrastructure to indicate a bike route (traffic calming devices, painted signs on the road, etc), while others are just low, medium, or high traffic through streets that RLIS maintainers thinks might be a good bike route. Because the bike routes indicated by RLIS maintainers are used on a variety of bike maps handed out by bicycle shops, groups, and on a website, we felt that even the ways without infrastructure were important to tag.
- Some new tags were created because OSM tags are insufficient. Two such tags are RLIS:bicycle=designated and RLIS:bicycle=caution_area. We applied RLIS:bicycle=designated to all ways that have any sort of bike route designation in RLIS, whether it is due to bicycle infrastructure or an RLIS maintainer’s opinion.
- RLIS:bicycle=caution_area indicates ways that connect bike routes and are often considered part of a bike route, but may be dangerous due to small shoulders, high traffic, low visibility, or some other reason.
- A tag that is only in the proposed stage is also in use because no other tag seemed sufficient - cycleway=shared_lane. This tag was used for Portland’s bicycle boulevards, which do not have a bike lane, but share a lane with traffic. The traffic is usually low, “sharrows” are painted every couple hundred feet to indicate bicycle traffic, and traffic calming devices are often used.
- Finally, for paths in the RLIS bike route dataset, we add bicycle=designated along with RLIS:bicycle=designated.

Trail types:

- Initially, we converted all trails to highway=path. However, we quickly learned this was inaccurate.
- Our trail and street datasets overlapped, so in some places, ways are considered part of a trail but also a service road for vehicles. We had to go back and review each of these overlaps to determine which designation was most accurate - could motorized vehicles fit on the way? Should it be tagged as a highway=track (service road) or highway=path?
- Finally, OSM users tend to switch trails back and forth between footway and path. These two are similar but footway indicates a heavier use by pedestrians and not necessarily exclusive of other users, while path indicates an equal balance between pedestrians, cyclists, and other users. Eventually, we settled on adding access tags (see below) to ensure that the ongoing OSM user battles would not affect routing capabilities.

Surface:

- Surface data was included in the trail dataset. We had some trouble converting RLIS trail surface attributes to OSM attributes because the available OSM tags are limited. We ended up with some paths alternating between pebble stone, ground, and paved surfaces. Perhaps it would have been better to keep the conversion simple: paved versus ground.

Access:

Access refers to what users are allowed on the way. For example, sometimes only motorized vehicles are allowed on the way, sometimes only emergency vehicles, and sometimes only pedestrians.

- Access tags are particularly relevant for trails, paths, and bike routes. Certain highway types are contentious, such as cycle way vs. path vs. footway. We decided to include foot=yes/no/designated and bicycle=yes/no/designated on all paths and trails.

- Additionally, state law allows cyclists on most motorways except a few. Because this is different than some states, we are trying to add bicycle=yes where bicycles are allowed on the motorways and bicycle=no where they are not. Finally, we are adding bicycle=designated for streets and paths that include bicycle infrastructure.
- Foot=yes/no/designated follows the same line of reasoning as bicycle access tags.

Conversions:

- http://wiki.openstreetmap.org/wiki/RLIS
- http://wiki.openstreetmap.org/wiki/CCGIS

Lessons learned:

- Jurisdictional tags do not always mean the same thing as similar OSM tags. For example:
  - access=private: in RLIS, this indicates the maintainer of the property. In OSM, access refers to legal right-of-way.

If a tag needs to be explicit and remain unchanged, it may be best to create a source-based tag, i.e. RLIS:localid. However, this limits the ability of other routers to use the tags. Attribute names cannot be very long in ArcMap, so we had to change some attribute titles in JOSM after the conversion. Also, attribute names in ArcMap cannot contain a colon, so we had to add source-based data tags (RLIS:localid) in JOSM.

2. Analyzed differences between OSM data and jurisdictional data - "diff files"

First method

Made diff files by extracting OSM ways that were more than 10 m from jurisdictional data. However, OSM ways are much longer than jurisdictional segments, so it was hard to pinpoint where the difference existed.

Second method

Made diff files by extracting RLIS & CCGIS ways that were more than 10 m from OSM ways. This made it easier to pinpoint the erroneous OSM way. However, we quickly realized neither OSM nor jurisdictional data are consistently accurate.

Final method

- Made diff files of all important ways and checked them visually in JOSM.
- Loaded jurisdictional data in ArcMap. Selected and extracted all important ways. Our first priorities were bike routes and transit routes. Overlaid bike routes and transit routes on jurisdictional street and trail data, selected the streets and trails that are a bike route and transit route, extracted those. Then converted this shapefile to .osm format.
- Opened the diff file in JOSM and used it to highlight jurisdictional ways and OSM ways that needed to be looked at for geometry, attributes, and directionality.
- Deleted diff ways as each section was reviewed and corrected.
- Once done with the bike and bus routes, make a diff file of all the major roads that are not transit or bike routes and correct those. Major roads could be extracted based on street type for jurisdictional data, and selected as "not part of bus or bike routes."
- Make a diff file of unimportant roads if we have time using similar 'select by location' or 'select by attributes' methods in ArcMap.
- ogr-to-osm.py is used to convert shapefiles to .osm
- JOSM allows layers to be duplicated and converted to .gpx tracks, which allows symbology changes. We changed the symbology of gpx layers so that they acted more as highlights in the background. Then as we check and correct each segment, we delete the segments from the .osm diff file. Every so often, we delete the .gpx diff and make a new one (duplicate .osm diff, convert copy to .gpx, symbology should be saved).
- Line thickness has to be changed in JOSM's map styles. Line color can be changed via right clicking on .gpx layer and choosing line color.

Lessons learned

It might be best to keep versions of the diff file in case an upload fails.

3. Editing

At first, the interns worked together on converting the data, but eventually, the interns did all the steps, from conversions to diff files to editing, for each county. OSM data was downloaded nightly, but due to the slow process of editing, it could be downloaded manually in small sections directly from the OSM server. The nightly download is handy for occasional reviews of the OSM data and analysis (i.e. searching an entire county for all corrected ways or certain tags). The OSM server has a limit for download sizes.

4. Current issues

Tagging - We continue to struggle with how tags should be applied and will probably continue to change our tagging scheme as we learn more.

Bike routes - how do we tag to indicate safety? This is a problem because:

- Street type is not sufficient to indicate speed or traffic volume
- Max speed and traffic volume should probably be included
- RLIS has multiple types of bike routes, some without bicycle infrastructure. Do we give all these routes bicycle=designated or do we only give them RLIS:bicycle=designated tag?

Street types

- Types are not clearly standardized in OSM
- Street types have different legal accessibility issues based on state
  trunk vs. motorways: in OSM wiki, trunks and motorways typically do not allow bicycles. However, in Oregon, bicycles are allowed on all freeways and highways except a few.
- Users changing tags
  Because highway=cycleway, path, pedestrian, and footway are debated, we decided to use access tags in case users changes highway types.

Turn restrictions

- In OSM, turn restrictions are included using relations. A relation sets up a relationship between multiple components. In the case of turn restrictions, a “from” way, a “via” node, and a “to” way need to be included, along with a type of turn restriction, such as no left turn or only straight on.
  A few problems have been encountered while adding turn restrictions to OSM:
  - Complete and accurate jurisdictional data is difficult to find.
  - Relations for turn restrictions can be difficult to understand.
  - Some routers seem unable to read certain turn restrictions (like straight on).
- Experimentation continues to more fully understand their usage.

Sidewalks

- Sidewalks as separate ways have been recently approved for use, though not many people have added in sidewalks yet. Most streets in our area have sidewalks, so it is not feasible to add in all sidewalks. We have been adding sidewalks where they connect a trail to a road.
- As this feature continues to develop, users will have to decide if sidewalks are still implied on all roads as they have been.
- It is also unclear how a user should connect sidewalks to roads. For example, should highway=crossing be used? http://wiki.openstreetmap.org/wiki/Tag:highway%3Dcrossing
- http://wiki.openstreetmap.org/wiki/Proposed_features/Sidewalk_as_separate_way

Naming

- RLIS names ramps based on what they connect. Ramps do not usually have names, and this name should probably be removed
- Highway names: Often, there are multiple names for highways and freeways
  - OSM user names have been retained, and if the RLIS name is different, we add the RLIS name and make sure the other names are included as name_1, name_2 or alt_name. Wikipedia is also useful in determining highway names.

Segmentation of ways

- We were worried about adding too many attributes that would segment the OSM ways. However, some exploration of European areas showed that segmentation of OSM ways is common as more data is added and doesn't seem to be a problem.
- Every additional attribute should be carefully evaluated against the importance and capacity to maintain the attribute, against splitting segments, as larger amounts of added segmentation will mean a slower graph-building by OTP.

Part 4 – Quality Control Process

The quality control (QC) process will begin in September. Detailed OSM maps will be printed at an appropriate scale by geographic areas defined by Police Precinct Districts. We will work with regional police departments on reviewing the printed maps. Errors will be marked by the police who have a high level of familiarity with each district. Corrections will be made in OSM based on feedback from the police. The following attributes will be verified for accuracy and correctness:

1. Street names
2. Street directionality / One-ways
3. Legal turn restrictions for vehicles, bicycles, and buses
4. Speed limits
5. Impedances

The following attributes will be reviewed and edited by the interns. Each intern will QC a county they have not edited. Through viewing the data on the screen and using a variety of rendering methods, the following attributes can be verified:

1. Physical turn restrictions
2. Bicycle lanes (using aerials)
3. Directionality
4. Alignment

Evaluations of planned trips for biking, walking, and transit will also be used to test data. Driving functionality will need to be enabled in OTP for testing of vehicular driving. This can also be done with the OSM iPhone app.

In addition, two OSM Mapping Parties will be scheduled prior to the fall beta release of OTP. Regional cyclists and local users will be recruited to obtain feedback on planned trips and bike weighting results.