North/West Passage

Transportation Pooled Fund TPF-5(190)

Project 5.6 Facilitating the Use of Open Source Software

Project Summary

Introduction

The Minnesota Department of Transportation (MnDOT) developed their traffic management software using in-house developers; with no reliance on commercial proprietary systems. Other agencies can receive the Intelligent Roadway Information System (IRIS) software from MnDOT at no cost, and then hire software companies or use their own in-house staff to integrate and even add functionality to the software.

Project 5.6 advanced the concept of open source software and its potential to help the North/West Passage states reduce the costs of developing and maintaining software systems to support transportation solutions. The project has educated the states on how to acquire, modify, maintain and integrate open source software using IRIS as the leading example.

This project summary presents information exchanged among the states during an introductory webinar and also provides key resources associated with finding, assessing, acquiring and using open source software. The summary is formatted to serve as a reference tool for the states when they choose to further explore open source software for their transportation management needs.

Introductory Webinar

An introductory webinar was held on May 9, 2012 to present background on open source software, allow discussion of perceived barriers, present an example of open source software success, and then facilitate discussion of potential next steps for the North/West Passage states. The purpose of this webinar was to introduce the North/West Passage states to and facilitate a discussion about open source software. Participants included North/West Passage steering committee members; agency software and IT staff; and, guest speakers with open source software experience.

Dean Deeter, Athey Creek Consultants, opened the webinar with an introduction to the spectrum of software ownership options and their common characteristics. Within the spectrum of software ownership, options range from the more traditional proprietary tools that are purchased off the shelf to the increasingly popular open source tools that are freely available online. Proprietary software is typically circulated for use but not for modifications. It is generally available either as off-the-shelf or through contractor developed products. Proprietary software is typically developed through an investment of one company that needs to charge a price to regain their investment and make a profit.

On the other end of the ownership spectrum is open source software. This type of ownership typically allows wide latitude for anyone to run, study, improve and redistribute the software. The source code is circulated with the software and it is common for user groups to form around the changes made to the

source code. This illustrates one of the common benefits associated with open source software – nearly real-time dialog with other programmers and users and the freedom to act on that dialog. Once open source software is initiated, the software community contributes to it and that typically results in lower development costs. A summary of these characteristics is presented in Table 1.

Table 1: Common Characteristics of Software Ownership

Software Ownership	Common Characteristics
Proprietary Software	• Circulation for use, but not for modifying / enhancing • Includes 'off the shelf' (e.g. MS Word) • Includes contractor developed products • Prices charged to cover corporate development costs
Open Source Software	• Freedom to run • Freedom to study • Freedom to improve • Freedom to redistribute • Source code circulated with the software • Costs are minimized or eliminated as the broader software community contributes to development

Between proprietary and open source software there are a variety of licensing options that allow, for example, source code modifications to be made but without redistribution. Other licensing arrangements may allow contractors to modify software after signing a non-disclosure agreement. Still other options may accommodate joint ownership of software. A common licensing arrangement that is specific to open source software is a General Public License (GPL). Under this option, software is copyrighted to specify the freedoms granted for use. It may also be "copy-lefted," which requires original freedoms or rights to be preserved in any modified versions of the work. It is also important to note that GPL does not require software be distributed at no cost; it can be sold.

Although the freedoms commonly associated with open source software are often touted as the greatest benefit, there are other advantages and barriers acknowledged in Table 2.

Table 2: Open Source Software Benefits and Barriers

Benefits	Barriers
• Cooperative software development and sharing of ideas	• Deployment and operation requires local configuration and adaptation
• New modules added by one user that other users may not have thought of	• Adequate budget is still needed
• Lower costs for initial deployment and ongoing maintenance (not no-cost)	• Agencies unfamiliar with open source software may be uncomfortable with the uncertainties
• Wider 'pool' of software companies familiar with the software, increased competition	• Perceived lack of corporate technical support

Following the overview of open source software, the introductory webinar transitioned into featuring an example of open source software for transportation management. Jim Kranig, MnDOT, described the IRIS software used by the department's Regional Transportation Management Center (RTMC). The software was developed in the late 1990's to accommodate a new RTMC facility, new National Transportation Communication for ITS Protocol (NTCIP) standards, an increasing number of dynamic message signs and the rapid growth of the freeway management system in the Twin Cities Metro Area. MnDOT assessed the proprietary software available at the time, along with their staff capabilities to develop and then maintain software in-house. IRIS was developed by MnDOT staff and several modules, such as ramp metering and managed lane controls, have been added in the years since. In 2008, MnDOT made IRIS available as open source software under a GPL. The California Department of Transportation (Caltrans) was the first agency to use IRIS under the new licensing. The department added an automated warning system feature to the software and has expanded the community of IRIS knowledge. Both of these accomplishments will greatly benefit all the agencies as new programmers are introduced to the software.

The City of Chandler, Arizona recently became another IRIS user and a direct beneficiary of the automated warning system feature added by Caltrans. Mike Mah, City of Chandler, and Tomas Guerra, Oz Engineering, shared their experience with selecting, modifying and using IRIS. Chandler sought control software that would allow them to display travel times on their local dynamic message signs. Oz Engineering was hired by Chandler to explore available options and recommend software for integration. IRIS was discovered and assessed in relation to the needs expressed by Chandler and the existing software and hardware that would need to be integrated with it. The software was selected, integrated and is being successfully used by Chandler today to post automated messages to the city's dynamic message signs.

In addition to the successful uses of IRIS by Caltrans and the City of Chandler, other agencies have assessed the software and determined it does not currently meet their needs. The Wyoming Department of Transportation (WYDOT), for example, recently reviewed IRIS. One of their key needs was the control of several highway advisory radio stations around the state. Because none of the IRIS users to-date have had a similar need, the capability does not yet exist in the software. WYDOT's need was more immediate than the time it would have taken to develop the function so they chose another option. However, as WYDOT's transportation management needs expand, they expect to revisit IRIS as a broader control option for meeting those needs.

During the final webinar discussion about the applicability of open source software for the North/West Passage states, information was requested about other open source transportation software and resources for assessing open source software. The remainder of this project summary addresses those requests. Additional information about IRIS and a copy of the webinar recording and presentation are available online through the North/West Passage Project 5.6 Facilitating the Use of Open Source Software web page.

Additional Open Source Software for Transportation

There are a number of software packages available to meet many transportation needs – modeling and simulation, maintenance management, infrastructure design, trip management, etc. National transportation organizations such as the American Association of State Highway and Transportation Officials (AASHTO) and the Transportation Research Board (TRB) are good reference points for finding transportation related software. Below are several examples resulting from a cursory search of "open source software" on the TRB Transportation Research International Documents[1] (TRID) and AASHTO web sites.

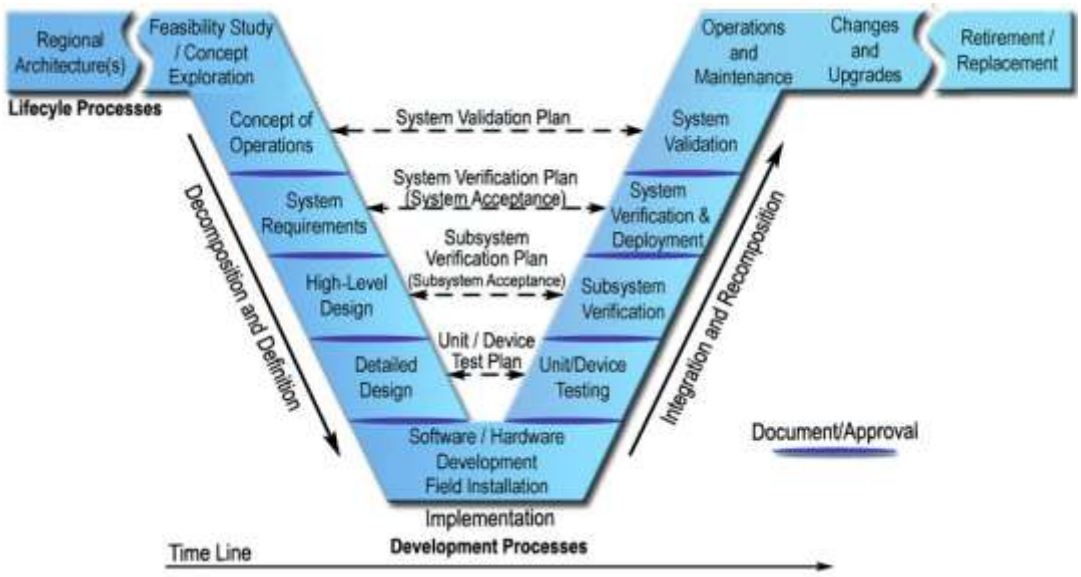
- Alternate Route Project. This project serves as a focal point for the creation of Open Source Bridge Engineering Software. The web site promotes open source software, provides information to end users, developers, and managers, and provides links to open source bridge engineering software and development projects. Even though the Alternate Route Project is created and managed by the Washington Department of Transportation it is viewed as an activity of the entire bridge engineering community.
- OpenTraffic. This is an open source platform for traffic simulation that enables the collaboration of academics from different geographic areas and disciplines within the traffic domain to work together, and to contribute to a specific topic of interest ranging from travel choice behavior to car following and activity planning. An introduction to OpenTraffic occurred at the TRB 91st Annual Meeting.
- OpenDRIVE®. This is an open file format for the logical description of road networks. It contains all key features of real road networks. It was developed and is being maintained by a team of simulation professionals with support from the simulation industry. Its first public appearance was on January 31, 2006.
- *OneBusAway*. This is a transit oriented software. It provides real-time arrival information for a number of transit agencies; arrival info for every bus stop; and easy access to information across a variety of devices. OneBusAway was started by students at the University of Washington, and it supports research on improving the usability of public transportation. It was developed as open source so that others may reuse and build upon their efforts.
- *Service Interface for Real Time Information (SIRI)*. SIRI is an XML protocol that allows distributed computers to exchange real time information about public transportation services and vehicles. As part of their real-time BusTime API pilot project in early 2011, the Metropolitan Transportation Authority (MTA) in New York implemented a modified version of SIRI to share their data with mobile app developers.

[1] TRID is a newly integrated database that combines the records from TRB's Transportation Research Information Services (TRIS) Database and the Organization for Economic Cooperation and Development Joint Transport Research Centre's International Transport Research Documentation (ITRD) Database. TRID provides access to over 940,000 records of transportation research worldwide.

Open Source Software Resources

When assessing open source software one of the most important factors in the assessment is having a clear understanding of the needs the software must address. It is also important to thoroughly understand the environment (e.g. hardware, other software interfaces) that the software will operate within. Systems engineering is one of the most readily available resources for clarifying needs, describing an operational environment and defining requirements. Once primarily used in software development, systems engineering is now commonly used in transportation projects. It is a natural progression to then use a systems engineering approach to evaluate open source software needs for a transportation function. Systems Engineering for Intelligent Transportation Systems is an excellent reference published by the Federal Highway Administration.

Figure 1 Systems Engineering Process



There are a number of other references available on the broader subject of open source software. Government Technology and its sister publications are an award-winning family of magazines covering information technology's role in state and local governments. Through in-depth coverage of IT case studies, emerging technologies and the implications of digital technology on the policies and management of public sector organizations, Government Technology chronicles the dynamics of governing in the information age. A general search of “open source software” resulted in a number of articles that have been written on the subject, particularly the increasing number of policies requiring agencies to consider open source on equal footing with proprietary software.

Open Source for America is another useful reference. It is an effort to raise awareness in the U.S. Federal Government about the benefits of open source software. The resources section of Open Source for America provides a number of useful references to case studies, policies and helpful guides. One reference may be particularly useful to agencies evaluating open source software – How to Evaluate

Open Source Software / Free Software (OSS/FS) Programs. The paper presents a step-by-step process for determining if open source software/free software (OSS/FS) is right for your organization. It outlines four basic steps: identify candidates, read existing reviews, compare the leading programs' basic attributes to your needs, and then analyze the top candidates in more depth. As these steps are described in more detail, the paper also covers issues related to functionality, cost, market share, support, maintenance, reliability, performance, scalability, usability, security, flexibility/customizability, interoperability, and legal/license issues.

Just as open source software is developed in a community fashion, resources to support finding and evaluating software are also widely available and offered in a collaborative spirit that encourages growth of the open source community.