Gateway, the Process and the Program

Contents

Introduction to the Gateway, the Process and the Program

Background

In the fall of 2006, Vermont and Rhode Island became the first additional states to join the other assembly of states that made up the Streamlined Sales Tax (SST) Project. The SST project had been processing sales tax registrations and sales tax returns for just about a year, when the Vermont Department of Taxes joined. The Department had already taken on several projects to get its enterprise tax system ready to process taxes for our state's local jurisdictions (what we call 'Local Option Taxes'). By the time we were voted into SST membership by the member states, we were very close to having our enterprise system ready to process SST Registrations and SST Returns. The only issue was a way to get those registrations and returns into our system. Up to that point we had one 'Black Box' program that took input from all our different sources and created the input for our enterprise tax system. This application was written in COBOL.

The Problem

One of the technical requirements of the SST project was that states be able to accept electronic transmissions of XML data for registrations and returns through Web Services using SOAP messaging protocols. XML, SOAP, and Web Services were not readily supported by our COBOL applications. Our department's 'other' development language (PowerBuilder) had some rudimentary support for these technologies, but was not very reliable. The Department had the vision to recognize this early in itsr SST application process, and started to train and recruit some developers who would be comfortable with these technologies and develop this required SST interface. By the fall of 2006, the Department had three developers, some tools and training. What we did not have was a single line of code. But we did have a Vision, and it turns out it was a pretty concrete vision.

The Vision

For this project we would use the following:

- We would use Agile / eXtreme development practices to produce work quickly, reduce risk, and respond to changes quickly.
- Our Gateway designs would be utilizing Service Oriented Architecture (SOA) practices, making a number of flexible application components that would be the building block of business functions.
- We would follow the Open Standard path as laid out in the project's required XML, SOAP, and Web Services technologies.
- We would leverage Open Source Software to build a robust, tested and reliable application.

Our Goals

Our vision was influenced also by the future need to accommodate the same XML, SOAP, Web Services technology for the Modernized e-File initiatives. Our main objective for the period winter 2006 through the summer of 2007 was the same as most projects, meet our deadlines for SST, do it with the allocated resources and end up with a reliable yet robust application. By the end of summer 2007, we had met those goals very successfully, but somewhere along the way, the process led to the development of other goals, goals larger than the project itself. We had received support from different open software groups, fellow developers working in these technologies and fellow state tax / revenue agencies. The development team, absorbed in the collaborative, sharing environment wanted to give their development work back to those communities. The Department supported the Gateway becoming open source, to be shared with other Vermont agencies, the Vermont public, other state tax / revenue agencies, and the public at large.

How We Developed the Gateway

How Agile / eXtreme development practices helped the Vermont Department of Taxes

When Vermont finally joined Streamlined Sales Tax, the Department had a very short time to get required business functions up and running. We knew that with limited resources and a small staff, the development process would have to be very iterative, agile and component oriented. The short development window for each new required business function lent itself to the Agile methodology. The development team worked very closely, reviewing and refining all along the way. It began with a small application that could process XML Streamlined Sales Tax registrations into our backend tax return systems. We had just over a month from the initial line of Gateway code to running our first batch file. For our next iteration, just over a month later, we developed the required Web Services interface for accepting electronic Streamlined Sales Tax returns. That core application was the Gateway 1.0 but it still needed work, and in the next couple of months, we added a web front end, both for the end users (Streamlined Sales Tax sellers and vendors) to transmit and to assist in the administration of the Gateway.

How Service Oriented Architecture practices helped the Vermont Department of Taxes

Another driving force in our development process was the use of Service Oriented Architecture (SOA) practices. We developed fully functioning independent components that were loosely tied together and used open standards for stringing them together to form flexible business processes. As each component was becoming feature complete, a number of 'unit' tests were developed to ensure quality. Having several individual components that were loosely coupled allowed the developers to re-factor specific parts of the code without affecting other components. Many components provided enough utility to be widely re-used, creating new functionality. As the project evolved, the previously written unit tests helped developers prevent regressions. Utilizing SOA as well as an Agile development strategy enabled the Gateway to take shape much more quickly than most iterative methods allow.

How Open Standards helped the Vermont Department of Taxes

The Department recognizes and commends the leadership of the other states and their work through FTA and TIGERS to adopt Open Standards. The support of XML, SOAP, and Web Services allows maximum flexibility for states and vendors to choose technology based on their own business discussions. The Vermont Department of Taxes chose to develop the Gateway based on Open Standards as well, like the Java Enterprise Edition. Java Enterprise Edition (J2EE/JavaEE) is a Java framework standardized by Sun that defines several Advanced Programming Interface (API) specifications. This allows the Department to develop an application that is portable and scalable. The JavaEE application server handles the transactions, security, scalability, concurrency and management of the components that are deployed to it, leaving the developers to concentrate on the business logic of the components. A JavaEE application can run, with slight modifications, on any conformant application server. Several commercial/proprietary and open source options exist, such as JBoss, BEA, IBM, Oracle, Apache, and Sun. JBoss is currently in use by Vermont, but a switch to another JavaEE application server could be completed without much effort. Additionally, most application servers can run on any operating system with a Java Virtual Machine like Windows, Mac OS, Linux, FreeBSD, AIX, etc.

For web presentation layer of the Gateway, we used the Open Standard of JavaServer Faces (JSF). Again, this provides a flexible framework for developing web based user interfaces that a number of software groups have created for us to plug-in to. These open standards allow the Department or any other implementer the flexibility to choose the best technology for a particular solution.

How Open Source Software helped the Vermont Department of Taxes

For the implementation of the Gateway in Vermont, the Department used a number of Open Source Libraries, used Open Source development tools, , and runs the Gateway on Open Source Application Servers, Database Servers and Operating Systems. Though cost was a major factor in choosing to work with Open Source technologies, an added benefit was that we could expand our vision of the Gateway by making this robust application freely available to other states.

There are a number of mature Java Open Source software libraries implementing XML and Web Service technologies. Software libraries like Apache's Axis, automatically created source code for Web Services. With literally the click of a button, a dozen Java source files (Java classes) were built, and provided all the required functions to send and receive Web Service SOAP messages. Kenneth Love from the state of Oklahoma pointed us to another Apache software library called XMLBeans. XMLBeans takes away much of the complexity of dealing with XML data by converting it into simple Java Objects. A function that might take dozens of lines to retrieve a single data item now only takes a line or two. Another nice feature was that XMLBeans automatically updated its references when schemas changed, so no coding changes were needed to maintain the existing functions. The Department also used libraries for a database abstraction layer called Hibernate. It allows the Gateway to interface with a number of different databases such as MySQL, Oracle, DB2, SQLServer, PostgreSQL and others. By changing a few lines in a configuration file, one could use a PostgreSQL database instead of MySQL. A couple of others that deserve mention are Apache's log4j for robust logging functionality and JBoss's JavaServer Faces implementation libraries.

The Department was already using a number of Open Source programs for configuration management and software lifecycle management. To this we added the development process itself, with the Eclipse Integrated Development Environment (IDE). The Eclipse IDE (first developed by IBM) is the most popular Java IDE. Its development is sponsored by a number of large software companies like IBM, Borland, Red Hat, HP, SAP and others as well as individuals. Eclipse's custom plug-ins for many of our implementing technologies assisted the Department with rapid, intuitive configuring of the Gateway. To manage the complex compiling, packaging and documenting tasks we used Apache's Ant and Maven. We should also mention the use of Subversion for version control and Trac for incident tracking and its wiki for living documentation. The Department also used a Firefox browser plug-in called Firebug to assist in the development and debugging of the web based user interfaces.

Open source development tools provided excellent software licensing cost savings. That practice continued in the project's selection of its entire stack of server software. Our production and testing Gateway environments use Apache's Web Server, JBoss's J2EE Application Server, MySQL Database, Java Runtime and Debian (GNU/Linux) operating system. All of these allow unlimited use with no licensing fees.

What the Gateway is, the Program

The Gateway program overview

The Gateway application is in actuality a number of programs that provide functionality to retrieve (pulled) and accept (pushed) electronic XML files through web services. The Gateway provides infrastructure to securely manage electronic transactions, performs business rule validation, formats XML data for backend systems, supports the viewing of the electronic transmission and documents, and provides the tools to configure and administer the whole process. The Streamlined Sales Tax components are complete, stable and deployed on production servers where they are accepting SST transmissions every day. Development of Modernized eFile components is well underway. The developers feel that the goal of creating an extendable framework upon which future tax services could be built, was achieved based on the speed and ease of implementation of the MeF components. The advantage of designing the Gateway the way Vermont did, for extension from the ground up, is that many of the components can be reused with little or no modification. Much of the client application for Streamlined Sales Tax already existed, only the XML schema files and a portion of the program logic had to be changed for Modernized eFile. Similarly, only minor changes were needed for the mock MeF server and submission viewer.

Besides unit testing, the developers did a great deal of functional testing during the development process. To facilitate this, several additional programs were created such as a mock server, a graphical client and web browser extensions. Such applications were another good example of the benefits of open source. Anything can be built on top of existing code to provide new functionality, without incurring extra costs.

Functions for Streamlined Sales Tax processing

- Processing SST Registrations; the Gateway performs Web Services 'pulls' from the SST central registration server on new and changed SST registrations. It performs basic validations and transforms the data to formats for our back end system. Electronic acknowledgments are created and sent back to the SST central registration server. This activity is logged and email notifications are triggered.

- Processing SST Returns; the Gateway accepts 'pushed' electronic returns from tax software vendors and end users/sellers. Again, basic validations and transformations for our back end system are done. Electronic acknowledgments are created and are retrievable by the vendors and sellers. This activity is also logged and email notifications are triggered.
- SST testing client; the SST central registration server supported testing for SST Registrations, but there was nothing in place to do the development testing of the pushed SST Returns. The Gateway SST testing client supports the testing of these processes by acting like the vendor's or seller's software. It has all the functions as defined by the SST project as well as the ability to validate the transactions passing back and forth though those functions. The testing client runs on any operating system supporting Java.
- Mock SST Registration Server; the Gateway's mock SST Registration Server supports testing of components of the Gateway's SST registration functions.
- SST web based UI; the Gateway's SST web based user interface supports three group of users. Administrators can manage users and search logs, tax staff can monitor transactions and pull reports, and vendors and sellers can use a web interface to send their SST Returns as an alternative to the Web Services. It runs on any operating system supporting Java

Functions for Modernized eFile processing (work in progress)

- Processing MeF Returns; the Gateway performs Web Services pulls from the IRS MeF server. Performs basic validations and transforms the data to formats for our back end system. We also check binary attachments for viruses. Electronic acknowledgments are created and sent back to the IRS. This activity is logged and email notifications are triggered.
- MeF testing client; the Gateway's MeF testing supports testing of components of the Gateway's MeF functions. It performs validation on the transactions passing back and forth through those functions. The testing client runs on any operating system supporting Java.
- Mock MeF server; the Gateway's mock MeF server acts like the IRS server. It sits available to receive MeF Returns requests from any of the functions as described by the IRS in the MeF specifications. It runs on any operating system supporting Java.

Common functions of the Gateway

- MeF / XML document viewer; the Gateway supports the viewing of the XML and binary attachments of registrations or returns being processed by the Gateway. The XML documents are transformed into viewer friendly html.
- Gateway command Line interface; the Gateway command line interface allows the Gateway business functions to be performed from the command line or scheduled with common scheduler applications. Any service the Gateway offers, or a combination of them, can be configured to run through this interface. The Department's production Gateway for SST has been completely automated by scheduling command line calls (we just monitor the emails).
- Configuration client; the new process we are working on currently is a configuration client. Much of the configuration of the Gateway is through editing of XML files. We hope to simplify many of the configurations activities through this client. The configuration client runs on any operating system supporting Java.

Other byproducts of the Gateway project

In the course of the project we also created several independent plug-in parts that are now available for others.

- Maven-har-plugin and maven-sar-plugin were created to assist the packing JBoss applications.
- TMX Editor and locale4j support localization of application.
- Gateway Anti-Virus and inoclam are two virus protection plug-ins for SOAP messages and directories.
- Cshampoo is an open source library for web servers and web services application written in C.
- A number of other simple utilities have been developed within the Gateway as solutions to common problems. These tools are freely available.

How the Gateway Project has Given Back to the Community

Creating the Gateway Open Source product

The Gateway project itself has become an open source project. By open sourcing the Gateway code, Vermont Department of Taxes and its developers could legally copy and paste existing, tested, debugged and working open source code into the Gateway, and distribute it freely. This means that anyone obtaining a copy of the source code, available as a free download, has the freedom to run the software for any purpose, study the source code, redistribute copies of the program and change the program to suit their needs.

How other state tax / revenue agencies can benefit

Of course, the audience that could benefit the most would be other state tax /revenue agencies. Many states were and are still implementing Streamlined Sales Tax, Modernized eFile and other XML-based electronic return systems. Collaborating and sharing the code with fellow states can help them accomplish their goals while reducing their costs. Nevada’s Department of Taxation has already begun modifying the Gateway for the state of Nevada. This saves developers in Nevada hundreds of hours of development time because the basic structure of the system is there. Nevada just needs to change the Vermont specific settings and develop an interface between the Gateway and their existing tax processing systems. The conversion is a simple and relatively painless process because the developers in Vermont were careful to separate Vermont business rules and settings from the baseline program logic. Changes require only elementary knowledge of Java and XML, and because of the open standards, .Net shops like Nevada can bolt on the Gateway to their backend systems just as Vermont has done for PowerBuilder. In addition to the program code, the documentation for the Gateway is freely available, allowing any state to look it over and get an idea of how to adapt the Gateway to suit their needs.

How tax software vendors can benefit

Sharing the code with the tax software vendors who send us data allows the transmitters to benefit by having a platform for testing changes to their software and debugging problems more easily. The transmitters can examine exactly what the code does with their input or setup their own private testing environment. We also have had software vendors use the SST client to validate errors in their transmissions when they were struggling to get their application to send transmissions to the states.

How others can benefit

Additionally, other government agencies and departments could adapt and enhance the code to meet their needs. Students or others who are studying Java software development could learn by studying our code. The code base has a lot of interesting and useful pieces to it that aren't directly related to taxes (the event logging system, authentication system, unit testing, etc). Developers of other open source projects could legally copy and use those components in their projects.

How open sourcing the Gateway benefits the Department

The Department was proud of the Gateway product and wanted to be a good neighbor and share the work that could help others. With the use of open source code and an open source license, we were able to make that happen. With more people using our code and more developers looking at our code, bugs can be more easily identified and fixed. Additionally, it gives developers an incentive to write good code because their peers, superiors, past and future employers and old college professors can examine and critique their code. We hope that this will result in higher quality software.

Creating an open source project was very cheap. There are many free services that can host the project; we used a number of these free services including the most popular called Source Forge. Source Forge hosts our web site and provides download capacity, a version control repository, metrics, mailing lists and incident tracing. We also have been given server resources at the Oregon State University's Open Source Lab where a Gateway demo is running for anyone to try, and where our nightly build process is hosted.

We hope more states will join with Vermont and Nevada in using and supporting the Gateway. As states join, we could see some large cost savings. States can share the burden of supporting these complex systems, build better products, and reduce risk by collaborating together. The Department has already received fixes and enhancements from the open source community, and those changes cost the Vermont Department of Taxes nothing. Why do 20 states all agree on rules, formats, and protocols, and then go back to their states and develop 20 different solutions? We hope this serves as an inspiration for other multistate collaborative projects.

The Gateway project has improved the Department's image as technically forward and has improved the candidate pools when recruiting for positions working on the Gateway.

How the Gateway is opening the conversation about government Open Source development

Several presentations about the Gateway project have been given in various venues. In October of 2007, Thomas Cort presented at the third annual Government Open Source Conference (GOSCON) in Portland, Oregon. His presentation was titled "Starting a State Government Open Source Project" and focused on the mechanics of starting an open source project in a state government agency using the Gateway project as an example of a successful project. Thomas gave a similar presentation at a meeting of the

Vermont Area Group of Unix Enthusiasts in September. Another member of the project team, Shawn Kasulka, presented “Open Source Web eFile Gateway for Streamlined Sales Tax Processing” at the Federation of Tax Administrators 23rd Annual Technology Conference and Exhibition in August of 2007.

Articles about the Gateway project have appeared from several news sources. The Gateway project was the ‘Cool Idea of the Week’ in TaxExpress. Tom Cort’s presentation at GOSCON was discussed in ‘Collaboration: Best Reason for Government Open Source?’ on LinuxToday.com. The initial press release about the Gateway project was carried by several internet sites including LinuxPR, Chinese Linux University and Open for Government.

What is in Store for the Future of the Gateway and the Gateway’s Process

Support for current and future eFile tax projects

We look forward to Nevada going live with the Gateway to support their entry into the Streamlined Sales Tax group. We know there are some changes coming in SST with an expanded return and shuffling around of the XML schemas to match MeF layout. Vermont and Nevada will be working together to implement those changes. Vermont is also fully committed to extending the Gateway functions to the Modernized eFile project, and much of that work is complete or is near completion. We hope to show off a fully functional MeF Gateway at the FTA Technology Conference this August in Chicago. The Gateway would be very capable of supporting Fed/State Employment Taxes (FSET) or other taxes based on XML, Web Services and SOAP and out there for any state to implement.

Vermont looks forward to expanding our Gateway family and working with other states supporting electronic filing of taxes.

Conclusion

While the Gateway provides something that many states have already implemented, the Department and its developers believe the project is radically different. From the beginning, the Gateway was designed with collaboration and freedom in mind, for other states to use and improve as a community. It is the first open source tax software project developed by a state government agency. Unlike some other submissions for the award, you (the reader) can take the software out for a test drive and read the documentation if you wish.

Where to find out more

Gateway Demo Site (the portal and web service)

http://gateway-demo.osuosl.org/

Gateway Project Page (documentation, presentations and other project info)

http://gateway.sourceforge.net/