Introduction

The United States Department of Transportation (USDOT) Research and Innovation Technology Administration (RITA) Intelligent Transportation System (ITS) Joint Program Office (JPO) has set forth an ITS program that focuses on intelligent vehicles, intelligent infrastructure and the creation of intelligent transportation systems. A critical component of the overall program is the development and deployment of quality ITS standards which support the program’s goals. For over a decade, ITS JPO has provided funding for the development of the Advanced Transportation Controller (ATC) family of standards. The ATC standards program is under the oversight of the ATC Joint Committee (JC) and the standards are developed by various technical subcommittees called Working Groups (WGs). The equipment and software specified in these standards provide for powerful general purpose transportation field equipment that is

- Open Architecture – The equipment is buildable by any manufacturer or developer;
- Modular – The equipment has an internal structure such that there is separation in the functions of its subsystems and assemblies and flexibility in the way they are combined;
- Multi-Process/Multi-Application – The equipment can run multiple and diverse application programs simultaneously;
- Able to grow with technology – The equipment provides for evolution within the same framework of the standard to mitigate early obsolescence; and
- Suitable for operation in legacy transportation field cabinet systems (TFCSs) – The equipment provides contemporary performance and capabilities even when used in existing TFCS infrastructures.

At the heart of ATC program are the ATC 5201 Advanced Transportation Controller Standard and the ATC 5401 Application Programming Interface (API) Standard. ATC 5201 specifies a controller architecture where the computational components reside on a single (5” x 4”) printed circuit board (PCB), called the “Engine Board,” with standardized connectors and pinout. It is made up of a central processing unit (CPU), a Linux operating system (O/S) and device drivers, memory, external and internal interfaces, and other associated hardware necessary to create an embedded transportation computing platform. ATC 5401 defines both user interface facilities and C programming language interfaces for ATC controller units that are not provided through ATC 5201 or the standard Linux O/S. The user interface facilities of ATC 5401 include a windowing system that allows operational users to interact with concurrently operating application programs (which in turn have their own user interfaces) and system-wide configuration management utilities. The C programming language interfaces of ATC 5401 provide C language function definitions that allow software developers to create application programs that share resources of the ATC controller unit including the front panel, the TFCS field I/O equipment and the real-time clock. When used with the Linux O/S and device drivers of the Engine Board, ATC 5401 provides for a software environment that allows application programs to be portable (runs on any ATC manufacturer’s equipment), compatible (will run concurrently with other application programs), and interchangeable (assuming they perform the same function) on a single ATC controller unit.

This concept paper has been developed as part of the “Reference Implementation of ATC 5401 Application Programming Interface (API) Standard Version 2” project funded by the USDOT Contract Number DTFH61-11-D-00052, Work Order T-13003 (referred to as the APIRI project). The goal of this project is to develop reliable software that is representative of the ATC 5401 standard and to provide it for free (at no cost to the user) in a manner that is useable and maintainable by the manufacturers, software vendors, consultants and agencies of the transportation industry. There are two bodies of software to be developed within the project. The APIRI software which is the software that runs on ATC controller units
as described previously and the API Validation Suite (APIVS) which is used to test the APIRI software on ATC Engine Boards.

**Purpose**

It has been proposed that the software developed under the APIRI project be hosted and maintained in an Open Source Software (OSS) environment. The purpose of this paper is to create a shared understanding of the OSS concept for the USDOT, the API WG and the contracting team. In the Project Description, goals for the OSS environment are identified. Measurable objectives are proposed that will move the project towards these goals. This concept paper is not intended to capture the result of the objectives but to describe them adequately enough for the effort to move forward. Some of the objectives can be accomplished relatively quickly. Others will require a development process that includes iterative reviews by the working group.

**Project Description**

The OSS environment is intended to create a sustainable software capability to serve the transportation industry for many years. Some of the key factors for success of an OSS project are:

- **Keeping People Interested** – The single most important attribute of a successful OSS project is that people are continuing to work on the software. In APIRI project's case, the ATC manufacturers and software developers have a direct and mutual interest in the project being successful. This initial developer base will need to grow for the project to carry on in the future.

- **Comprehensive Documentation** – The hosted software must be comprehensible not only to the initial developers but also to those new to the project. Documentation will need to be more comprehensive than typically used in a closed environment of a single organization. This includes software documentation, process documentation, user manuals, rules regarding communication, and more.

- **Continued Oversight** – As the project matures and more developers are included, there will still need to be someone watching over the project as a whole, enforcing standards for communications, resolving conflicts and administering the software development process.

With the success factors in mind, the following goals for the APIRI project's OSS environment have been identified:

1. **Provide free access to the APIRI and APIVS software** – A secure server providing a common place to store and access the API software. This builds confidence in the software product and allows manufacturers to incorporate the software in their own ATC controller units.

2. **Facilitate the collaborative maintenance of the software** – Provide tools and product life cycle processes to ensure a high quality, robust, complete and correct software product. This includes the standards to be used and the written procedures for proposing changes, incorporating changes, creating an official distribution, testing, and decision making.

3. **Reduce barriers to entry for anyone desiring to understand concepts or use the software** – This includes documentation not only for developers but also for non-technical users who may specify ATC equipment. Appearances, presentation and packaging are important.

4. **Build both the developer base and the end user base** – Without end users requiring the API software in their specifications and utilizing it operationally there is no market for it. Alternatively, if there are few developers using the API software and creating applications that leverage it, there is little benefit perceived by the end user.

To accomplish these goals the following objectives are proposed:
1. Choose a Name – This name will become the website and the focal point for the entire project including the OSS environment. The APIRI Project uses the name “APIRI” to refer to the software that is developed conforming to the ATC 5401 Standard and “APIVS” to refer to the software that is developed to test the APIRI. The name should be discussed within the working group and confirmed with the ATC Joint Committee and the USDOT. It should be verified that the name does not infringe on any trademarks and there are not any other OSS projects with that name.

2. Create a Mission Statement – The mission statement should be short, concrete and limiting.

3. Choose the Technical Infrastructure – This includes choosing the tools that are used to version control, bug tracking, mailing lists, and real-time chat.

4. Establish the Product Life Cycle Processes – This includes the development of the life cycle procedures that state how things will be done as they relate to both the technical infrastructure and the supporting stakeholders.

5. Establish the formal licensing to be used for the software. The GNU General Public License (GPL) and Lesser General Public License (LGPL) have been proposed for the project.
   a. GPL requires that the covered software be distributed at no cost, that it is modifiable and requires all modified and extended versions of the software that are distributed to also be available at no cost. This allows users of the software to enhance it for their own purposes but requires them to make any distributed enhancements available at no cost. This type of license would apply to the entire APIVS software and portions of the APIRI software.
   b. LGPL allows developers to use and distribute the covered software (typically libraries) linked with their own applications but not have their applications covered by the license. If any modifications to the covered libraries are made, those changes fall under the rules for the GPL license as described in (a). This allows the intellectual property of an application program to remain with the developer. This type of license would apply to the library portions of the APIRI software.
   c. The exact licensing and versions must be set by ITE in consultation with the APIRI project team and with the other ATC standards development organizations, NEMA and AASHTO.

6. Develop a Website that:
   a. Includes the mission statement;
   b. States that the software products are free;
   c. Provides a listing of the features and requirements;
   d. Provides the developmental status based on the features and requirements lists;
   e. Provides downloads in a standardized packaging;
   f. Provides access to the technical infrastructure;
   g. Provides the communications channels for the project;
   h. States the licensing for the products;
   i. Provides documentation that is of interest primarily to developers including product life cycle documents, design documents, and manuals;
   j. Provides documentation that is of interest to non-developers (e.g. general information about the ATC program, how to register for distribution lists, links to standards, training materials); and
   k. States that the software has been developed and tested for distribution by the USDOT.

Benefits

Some of the benefits of this APIRI project performed in an OSS environment are as follows:
- Facilitates application software portability and compatibility by providing a common API software;
- Diminishes inconsistent operational characteristics due to varying software architectures in an industry where operational reliability is critical;
• Promotes fast deployment by providing API software for third party software developers, agencies and manufacturers in the transportation community;
• Promotes collaboration of industry software developers;
• Provides a forum for users to express ideas and concerns;
• Promotes quick bug fixes and alternative solutions to issues;
• Facilitates the introduction of new application developers to the industry as they do not have to purchase software to experiment with writing new applications for ATC equipment;
• Lowers the point of sale and maintenance costs to the industry; and
• Consistent in concept to the open source Linux Operating System specified in the ATC 5201 Standard.