

SRFF Tool Enhances Data Transfer

A new syntax checker allows PCB assembly equipment manufacturers to check the validity of their SRFF files.

By Andrew Dugenske

In the electronics manufacturing industry, much money is spent transferring data from one format to another. The information contained in the data remains the same, but the format is often different. For example, each equipment supplier often has its own language to program its machines. This approach allows the equipment suppliers to build machines that operate at maximum mechanical effectiveness.

However, as the electronics industry has matured, the need for interoperability has increased. Original equipment manufacturers (OEMs) now desire the flexibility of transferring products to machines of varying types, based upon the current economic situation. Without a standard machine programming language, the OEMs must create new programs whenever production is moved from one vendor's machine to another.

OEMs are not the only companies challenged by dissimilar machine control programs. Computer aided manufacturing (CAM) companies that produce machine programs from computer aided design (CAD) files must constantly review, rewrite and test their software to make sure it meets the criteria of machine vendors. Many CAM vendors have reported that this procedure is not highly profitable because constant maintenance of their software is required.

Equipment vendors also noticed the advantages of a standardized machine control language. By using a standard control language, the vendors do not have to design, develop and document their own programs.

To combat the cost of transferring information among electronic manufacturing equipment, the IPC SMEMA Council started a project to

standardize the language used to program equipment. This Standard Recipe File Format (SRFF) project was established to respond to the needs of large manufacturers that require additional flexibility in their factories. The SRFF project included a varied group of electronic manufacturers, software developers and equipment suppliers. More than three dozen companies participated in the development of the SRFF standard.

The council decided to accelerate the standard's acceptance by funding a reference implementation as soon as the standard was released. A reference implementation puts a standard through its paces by writing an application that actually uses the standard. In developing the reference implementation, errors in the standard can be determined and fixed quickly so the first implementers do not "skin their knuckles" in developing their own applications. Because the standard is debugged through the reference implementation, the first implementers will be very confident that the standard is of high quality. They also will not spend precious resources finding faults in the standard or lose time waiting for the standard to be fixed as their development cycle grinds to a halt.

The Manufacturing Research Center at Georgia Tech was chosen to develop the SRFF syntax checker reference implementation. The purpose of the syntax checker is to provide an unambiguous way of determining if SRFF files meet the SRFF standard. The syntax checker is easy to use and can be accessed through a Web browser. A file that needs to be checked is simply uploaded to the checker, and the checker indicates if any syntax errors were found in the file. The syntax checker can be found at www.fis.marc.gatech.edu/srff.

In addition to the SRFF syntax checker, the Website also provides developers with several free tools to assist them in adopting the new SRFF standard: file examples, documents, additional online tools, an API and a potential XML schema for SRFF. Some tools include an XML translator, which produces an XML version of a SRFF file, semantic checkers, error detectors and various parsers.

The Structure of SRFF

An SRFF file contains two main sections: the schema and data sections. The schema section defines a set of objects and the data section contains populated objects that were defined by the schema

section. The SRFF standard has defined more than 50 objects that can be used in SRFF files. These predefined objects can be used to represent information such as the size of a panel, the position of components, the geometry of component pads, and the order in which components should be placed.

Recognizing that users require the flexibility to define their own data objects to represent new and unique technologies, the standard allows users to define their own data objects. By allowing users to develop their own objects, the standard accommodates innovation, yet provides the rigidity to reduce costs.

Because SRFF files are object oriented in their structure, a simple syntax checker would not suffice. Rather, a checker capable of determining whether objects have both been defined and used correctly was needed. This SRFF checker performs both functions.

At the core of the SRFF syntax checker is a traditional lexer and parser, which are often used to determine a file's validity. These tools were augmented with additional software routines that check a SRFF file more effectively than a general development tool. The tool is written in Java, contained in a package, and can be accessed through a simple Java API. The API can be viewed at the syntax checker Website.

To access the new syntax checker through the Web, a user just needs an Internet connection and a Web browser. No software downloads or plug-in installations are needed. The user simply uploads a SRFF file to the tool through an HTML form and then receives an indication about the validity. Figure 1 depicts the HTML form used to interact with the tool.

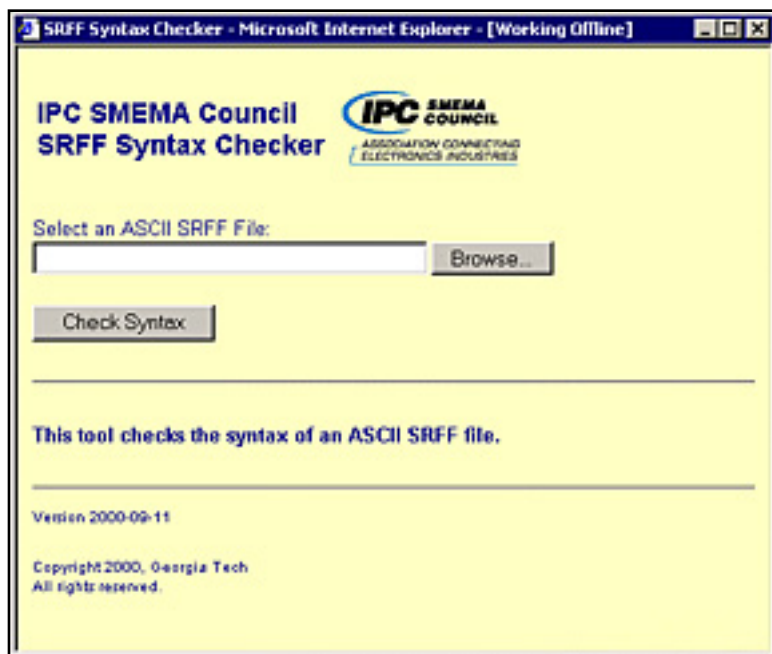
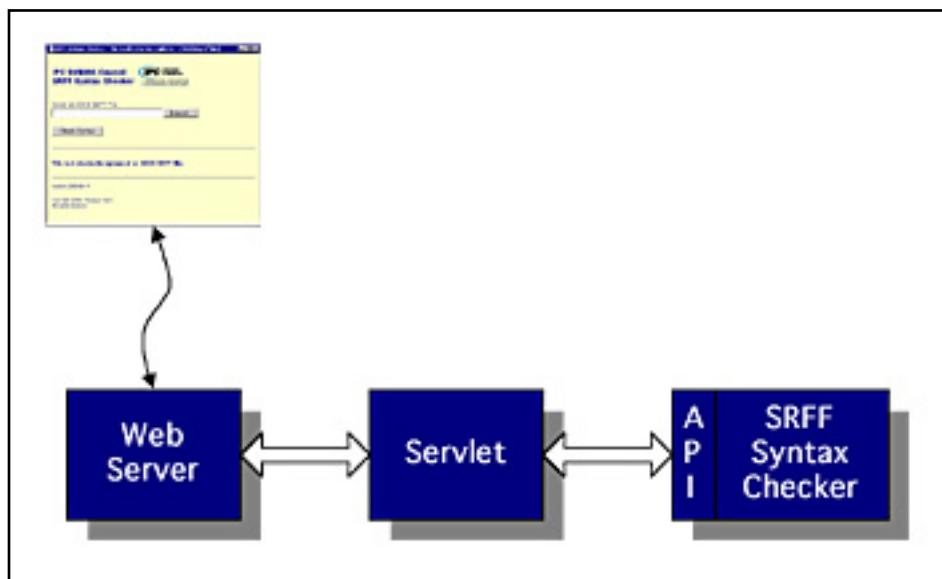


Figure 2 depicts how the SRFF syntax checker is accessed through the Internet. The Web page shown allows a user to select a local file for testing. When the “Check Syntax” button is clicked, the local file is uploaded to a Web server and a servlet is invoked. The servlet reads the uploaded file from the Web server hard drive and passes the data to the syntax checker.



The syntax checker determines if any errors are present in the file and passes the appropriate data about the check to the servlet. The servlet receives the data from the checker, erases the uploaded file and produces a simple Web page, which is passed to the user through the Web server. If errors are found in the file, the Web page contains information about the type and location of the error.

Conclusion

The goal of the IPC SMEMA Council was to provide a simple tool for the electronics industry to check the validity of SRFF files. By providing this tool, the council believes that the SRFF standard will be more rapidly accepted and the cost savings of using the standard can be quickly realized.

Andrew Dugenske is manager of research services at the Manufacturing Research Center, Georgia Institute of Technology, Atlanta, GA; e-mail: andrew.dugenske@marc.gatech.edu. Dugenske led the SRFF standardization effort and served on the IPC SMEMA Council during 1999 and 2001.