Standard Recipe File Format

By Andrew Dugenske

Cookie-cutter process control files may just be your formula for lowered costs, increased quality and reduced cycle time.

Manufacturers of surface-mount equipment have recently recognized a need to standardize process control files for electronics assembly machinery. These files, often referred to as recipes, provide the instruction sets used by assembly equipment to accomplish specific tasks such as component placement. Until recently, no standards existed for developing process control files, and each equipment vendor had to develop a proprietary method for conveying manufacturing information to its equipment. Consequently, process control files could not be shared among equipment produced by dissimilar vendors.

Recognizing the need to improve interoperability on the factory floor, the Surface Mount Equipment Manufacturers Association (SMEMA) of IPC: Association Connecting Electronics Industries (Northbrook, IL) established a task force to produce a standard that outlines a common method of constructing process control files. Over the past few years, representatives from equipment suppliers, electronic product manufacturers and software developers have worked together to produce the SMEMA Standard Recipe File Format (SRFF). SRFF provides a generic method of producing process control files that may be shared among various equipment types produced by any vendor.

Manufacturing Challenges

Currently, the manufacture of products cannot be moved among dissimilar manufacturing lines without a cost penalty. Shifting production requires a manufacturer to develop new process control files, which requires resources. This inability to move production also greatly reduces flexibility because manufacturing cannot be seamlessly moved from one production scenario to another. Time and resources must be invested to produce the necessary process control files, which only add overhead associated with a shift in production. Quality may also suffer. Assuming a manufacturer shifts production among various manufacturing lines, several process control files representing the same product must be generated. If changes are not propagated among all the files, discrepancies may arise that produce errors and lower yield. The goal of SRFF is to decouple the equipment used to produce a product from the generic manufacturing recipe.

General Structure

The SRFF standard describes the file format, outlines the file sections and indicates how data should be represented. The two major sections of an SRFF file are the schema section and the data section (Figure 1). The schema is placed at the beginning of the file and defines the objects that will be used in the data section. The data section contains the actual data in the form of object instances, populated with information. Two different types of objects can be used in an SRFF file: vendor-independent objects and vendor-specific objects. Vendor-independent objects are defined by the SRFF standard and are meant to represent data in a generic manner. Vendor-specific objects have been defined by a particular vendor for a specific application.

By using a schema that describes objects at the beginning of each file, the data structure is extensible and a generic parser may be used to interpret any SRFF file. A parser is a software program that breaks a data file into the individual components that comprise the file. A generic SRFF parser can be written by adhering to the Backnus Naur Form (BNF) that is included in the standard.

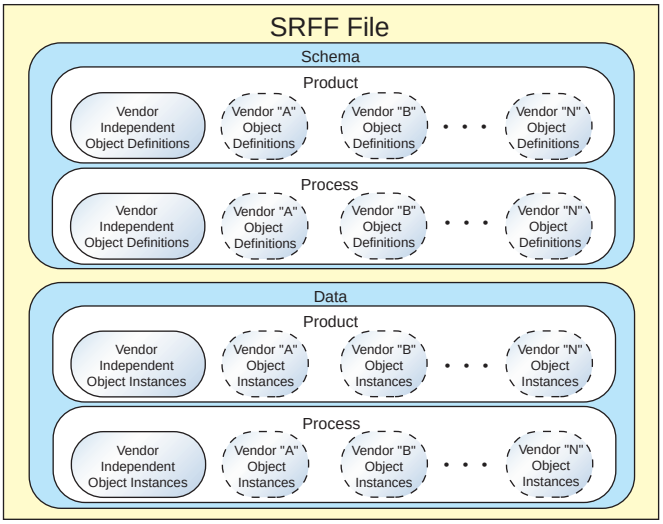

FIGURE 1: General structure of the SRFF.

The BNF outlines the syntactical rules that must be adhered to when producing an SRFF file.

Extensibility is important when standardizing process control files. Therefore, in SRFF objects can be defined and used that are not explicitly defined in the standard. Vendors can define new objects that might be required to describe a new process, yet still adhere to the standard. These new objects can be added to the structure, without breaking tools used to interpret a generic SRFF file.

Extensibility is also important when future versions of the standard are released. Since the method in which the SRFF data structure can be extended has already been determined, tools built to interpret a generic SRFF file will not have to be significantly modified when new versions of the standard are released. Ideally, the tools will be able to interpret any future SRFF file because the path for extending the standard has been established. Another advantage of extensibility is that tools built to interpret future versions of the standard will easily be able to interpret data structures based on the first version of the standard, because extensible data structures are by definition backward-compatible.

Product, Process Sections

The schema and data sections are further segmented into product and process sections. The product section of the schema is used to define objects that pertain to the physical characteristics of the electronic product, such as location, thickness and part numbers. The product section provides the necessary geometric information required by the equipment producing the

Using Data Standards from Product Idea through Production

By using both GenCAM (See "GenCAM: Version 1.0," p. 30) and SRFF, the electronics manufacturing industry has a standards-based path to transfer information from idea to production. In typical product development, a product is conceived, and a computer-aided design (CAD) application is used to develop an electronic version of the product contained in a file. Typically, computer-aided manufacturing (CAM) applications are then used to produce recipe files from CAD files. CAM applications use the data contained in CAD files and the user's requirements to develop equipment control programs. These recipes are downloaded to the equipment and products produced.

If standards are not used to develop the CAD or recipe files, the data translation activity may be more complex than necessary. As shown in Figure 1, the number of data translators required to transform information from project inception to manufacture increases geometrically as the number of different CAD or recipe file types increases. The transfer of data through the enterprise may be greatly streamlined by using standards (Figure 2). For example, GenCAM may be used as the single product definition file format, and SRFF may be used as the single machine program file type.

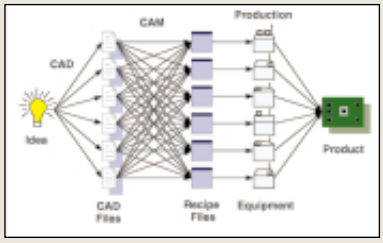

FIGURE 1: The number of data translators increases geometrically as the number of different CAD or recipe file types increases.

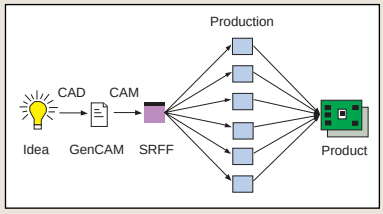

FIGURE 2: Streamlining data transfer through standards.

If products are conceived, designed and produced within a single enterprise, transferring information using standards is not essential. An organization can simply dictate which file formats are to be used within its four walls. However, as the trend in outsourcing continues in electronics manufacturing, data transfer standards will increase in importance.

product, not a comprehensive description of the product. A standard such as GenCAM may be used to fully define a product (Sidebar).

The process section of the schema is used to define objects that relate to the manufacturing of the product, such as placement order, squeegee pressure and oven temperature. These objects are key to SRFF because they provide the generic manufacturing instructions for producing the product defined in the product section. Generally, process objects use data contained in product objects combined with process-specific information. Process objects indicate how the items defined in the product section should be manufactured.

Vendor-Independent Objects

The SRFF standard has defined many vendor-independent objects, which are generic in nature and may be used to represent information common to electronics manufacturing. These objects are grouped in 12 categories of similar object types: common; dispense; inspection; line configuration; material movement; placement; print; reflow; shape; test; unit and wave solder.

The common category contains objects used to define the basic building blocks, such as geometry, of an electrical product. Common objects include images, image definitions, features, fiducials, locations and patterns; groups of these objects are defined in the common section. Images are coordinate frames used to define the special representation of items in the file. They contain a mapping from a parent image to their origin. A primary parent image is defined in the file that defines the most basic location. The position of all other images may be traced to this primary image through parent-child relationships defined in the file.

Location is another object used to define coordinate systems. Locations are different from an image in that they define the position of an electrical component contained in a product defined by an SRFF file. Since each location references an image, the absolute position of a location can be determined by tracing the parent-child image relationship. Although locations can reference images, they cannot reference other locations.

The objects contained in the shape category are used to define basic shapes such as rectangles, diamonds, donuts and discs. The units category contains objects used to define the units used throughout the file. The units of length and torque are defined in the objects, lengthunits and torqueunits.

The other nine categories contain objects specific to a particular equipment type. The placement category contains objects that may be used to define placement operations, and the screen printer category contains objects pertaining to screen

printing, such as screen size, print stroke parameters and squeegee properties.

Vendor-Specific Objects

In addition to vendor-independent objects, an SRFF file may also include objects defined by vendors. Because they are exclusive to a particular vendor, these objects are known as vendor-specific objects. If a vendor-specific object is to be used in a file, its syntax must be included in the appropriate vendor schema section in the file. Vendor-specific object definitions must adhere to the SRFF BNF so a generic tool can be developed to interpret any object definition. In addition to strictly adhering to the BNF, guidelines for developing well-formed vendor-specific objects are also included in the standard. A vendor may use vendor-specific objects to extend vendor-independent objects or introduce new concepts.

If vendor-specific objects are used in an SRFF file, they should be defined in the appropriate vendor section of the schema. To eliminate name/space collisions, each vendor must register with SMEMA to obtain a unique tag. If a vendor wants to define a vendor-specific object, a section in the schema identified by this unique tag is included in the schema. In a similar fashion, the populated vendor-specific objects are included in the appropriate vendor-specific data section. By placing vendor-specific object definitions and data in sections of the file labeled with a vendor-specific tag, information associated with a particular vendor may easily be separated from the rest of the file.

Conclusion

The SRFF is a common, generic, interchangeable and extensible method of producing and interpreting files that provide instructions to electronics manufacturing equipment. Software developers, equipment suppliers and electronic manufacturers may be able to increase quality, reduce costs and improve flexibility through use of the standard. Software developers can concentrate on high return on investment applications versus support-intensive translators. Equipment suppliers can provide a standard method to transfer information to their equipment, which allows this equipment to compete on the basis of performance as compared to a proprietary software architecture. Finally, SRFF will allow manufacturers to use a single standard to program all the electronic production machines within their enterprises. This ability should decrease the time needed to introduce new products and lower costs—both attractive concepts in a highly competitive marketplace.

Andrew Dugenske *is the manager of research services at the Manufacturing Research Center (MARC)—Georgia Institute of Technology (Georgia Tech), Atlanta, GA; (404) 894-9161.*