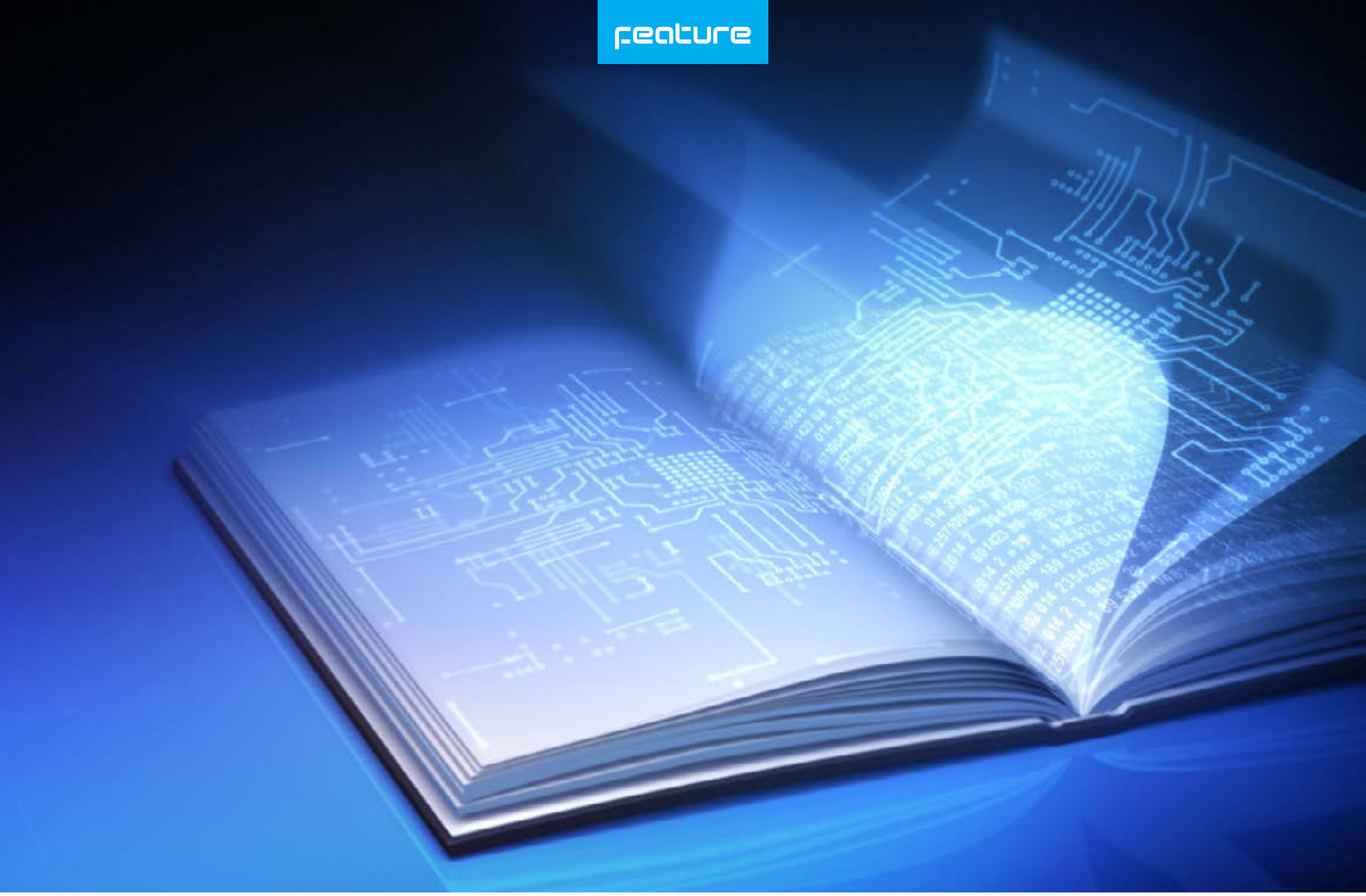

Mitigating the High Cost of PCB Documentation

by Mark Gallant
DOWNSTREAM TECHNOLOGIES

The goal of minimizing the time and cost for engineering a new product and getting it to market before the competition is a constant struggle for developers of leading-edge electronic products. The complexities of the electronic component supply chain, reductions in qualified staff, and managing globally distributed engineering teams are just a few of the challenges. Eliminating bottlenecks can dramatically reduce concept-to-production time in a product life cycle. This is the Holy Grail in the quest for corporate efficiency.

To reduce or eliminate these bottlenecks in new product development, many electronic engineering teams are looking inward and reassessing the efficiency of their product engineering process. From concept to design, fabrication, and assembly, there are process inefficiencies that result in delays. The usual excuse for not reassessing, "This is the way we have always done it," may be a sign of complacency in an inefficient engineering process.

Quite often the assessment involves an evaluation of the current EDA tool set proficiency. Frequently, these tool evaluations evolve into an assessment of how their current tools perform against competitive tools. Loyalty to the incumbent EDA provider is a lesser consideration than the efficiency of the tool. As a result, entrenched EDA suppliers are forced to perform benchmarks for current users or risk a decline in their customer base.

The Evolution of EDA Tools

In the CAD revolution of the 1980s, software-based EDA tools facilitated the migration from light tables and drafting boards to rooms full of hardware and software from best in class EDA tool providers. When one component of the EDA tool set was deemed inefficient, it was replaced with a more efficient tool from a litany of niche EDA tool providers. However, there was an inherent inefficiency in the interoperability of tools from different EDA tool providers. Integration was rudimentary at best, and often managed by the end-user. Poor tool integration was a key factor during the EDA provider consolidation in the 1990s as mergers and acquisitions forever changed the EDA landscape. The number of tool providers shrank dramatically, and the multi-provider EDA tool set slowly morphed into a sole-vendor EDA tool set.

With the rapid adoption of a sole-vendor EDA tool methodology, new challenges arose. Being the largest EDA tool provider did not necessarily equate to being the best provider. Technological advancements and engineering process improvement goals often required tools not available from the core EDA vendor. In some cases, vendor offerings were just not suitable for the users' needs. Replacing inefficient or unsuitable components of a single-vendor EDA tool set imposed a significant challenge. Any new component must integrate seamlessly

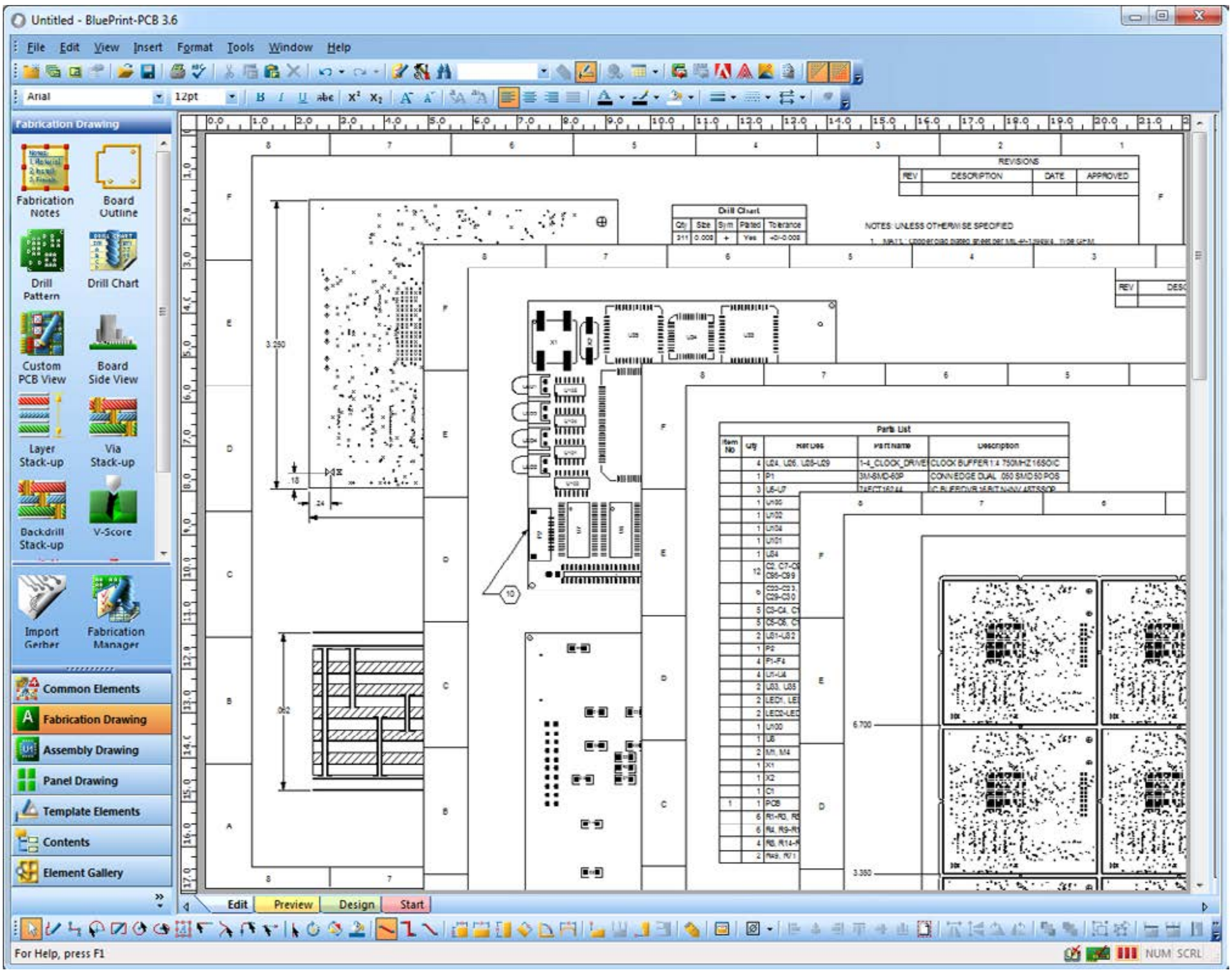

Figure 1.

into the current process and not disrupt the engineering process.

To facilitate this plug and play, engineering teams relied on user-developed custom applications. These applications served as the glue or patch cords between core EDA tools and niche third-party tools to create a manageable work flow. The pain of this methodology became obvious whenever an EDA tool provider required users to migrate away from their tried and true legacy tools to newer, more efficient tools. Typically, userware applications are written in a version specific scripting language. Legacy userware was incompatible and required herculean effort by specialized development resources from a small talent pool to achieve compatibility with the new tool.

These resources were vital to maintaining a smooth engineering process, but they are not without their impact on new product introduction costs. The cost of implementing and maintaining integration of disparate tools was the impetus for industry standard file format specifications for data transfer, such as IPC-2581 and ODB++. Having a neutral file format endorsed and supported by all members of the EDA ecosystem facilitated data transfer among disparate tools.

EDA tools like schematic capture, digital and analog simulation, PCB design and layout,

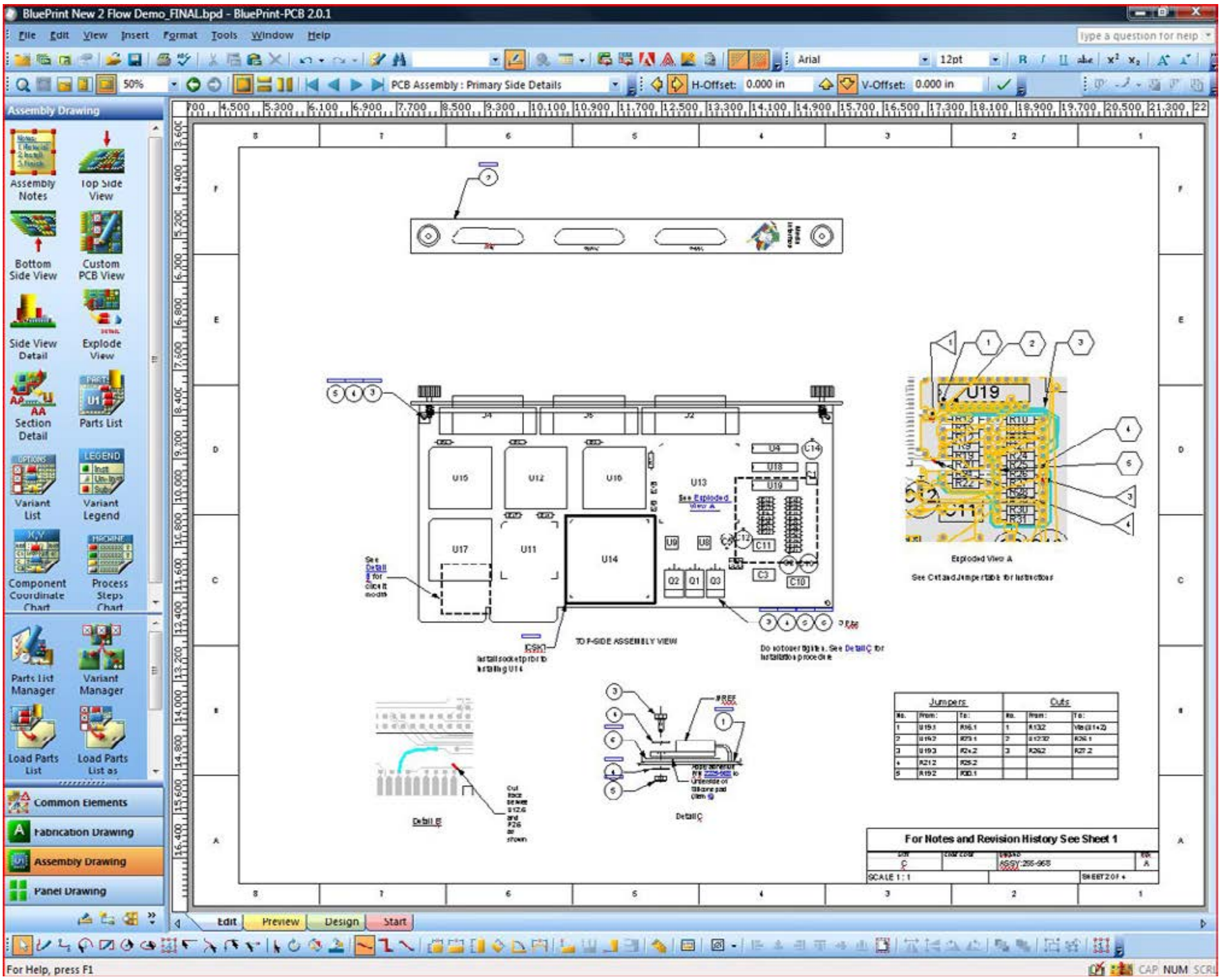

Figure 2.

autorouters, and others have matured, dramatically reducing time to market. But, even though most EDA tool providers have enhanced their tools through integration of new functionality or acquisition, some areas of inherent inefficiency in PCB CAD tools remain.

Post-Design Documentation: Late to the Party?

One glaring example of this: The documentation required for fabrication and assembly of the PCB. Comprehensive PCB documentation is a requirement today, and if the end-product is destined for a military or aerospace user, the documentation requirements ratchet up considerably.

Yet the majority of PCB documentation is created using PCB CAD drafting features that have not improved since the CAD revolution of the 1980s. CAD tools are superlative at reducing the PCB design cycle, but they fall short on some of the most basic documentation tasks. For example, PCB design tools that sell for tens of thousands of dollars per seat lack the spell-checking found in $99 word processor software. Most CAD tools lack support for a paragraph requiring the user to create multiline text strings as basic fabrication notes. Countless hours are spent using rudimentary drafting tools to create PCB detail views and drill charts, one segment or one line at a time.

These antiquated methods show their real colors during a design re-spin. Drill quantities out of sync in a manually drawn drill chart? Select the text string and edit the value. PCB detail view need updating because a component moved on the PCB? Either recapture the entire view or start editing the graphics one vertex or one segment at a time. Some engineering organizations have resolved to use popular MCAD tools to complete the PCB documentation. MCAD tools are superior to PCB CAD tools with respect to creating documentation.

However, converting the design to MCAD file formats results in disassembling the intelligent PCB CAD data. Parts are converted to shapes, traces to lines, copper shapes to polygons, and so on. Creating tables, notes, complex dimensions is certainly quicker, but cultivating intelligent data for a parts list is no longer possible with MCAD tools. Should a design re-spin be required, the MCAD drawings must often be recreated or manually updated with new design content. It also introduces an MCAD design database to an already crowded PCB documentation file collection. Maintaining synchronicity between PCB and MCAD design databases becomes the responsibility of the user.

> ***"Every PCB design has its share of tasks related to documentation. Stringent documentation standards for military, aerospace, automotive and other segments require highly detailed, time-consuming documentation sets. Documentation requirements are not limited to one department across an entire organization. Many downstream processes in product manufacturing have unique documentation requirements."***

The bottom line: Creating PCB documentation is not free or well automated in PCB CAD tools. Every PCB design has its share of tasks related to documentation. Some documentation sets can be dozens of drawing sheets. Stringent documentation standards for military, aerospace, automotive and other segments require highly detailed, time-consuming documentation sets. Documentation requirements are not limited to one department across an entire organization. Many downstream processes in product manufacturing have unique documentation requirements. This includes PCB rework instructions, assembly process steps, PCB panel fabrication drawing, assembly inspection, and more. There is more to a PCB documentation set than a simple one-sheet assembly and fabrication drawing. All of these documentation requirements should be part of any documentation efficiency calculation.

When considering the efficiency of your PCB documentation process, you must measure both creation and maintenance of a complete documentation set for the full life cycle of the product. A new product's documentation cycle typically begins at the prototype stage. From there, the number of documentation iterations can easily reach double digits. Consider that each iteration of a documentation set has both tangible and intangible costs, the greatest tangible cost likely being the salary of the individual creating the documents. Assuming an average PCB designer salary of $75,000, and 20% of his time devoted to originating documentation, annual documentation costs will average $15,000 per designer. That amount does not include time spent revising original documentation or documenting rework instructions to update in-house product inventory.

If we assume that revising a document takes 50% of the time it took to originate, add another 10% or $7,500, and your annual documentation costs rise to $22,500, a hefty sum if you are creating a basic two-sheet documentation set. Developing military standard PCB documentation can usurp more than 40% of a PCB designer's time. For those products, the cost of creating and maintaining documentation can easily escalate. Some engineering organizations operate without dedicated PCB designers and require electrical engineers to complete the PCB design process. Because the average salary of an electrical engineer is higher than that of a PCB designer, this could factor into the high cost of creating documentation.

"*Developing military standard PCB documentation can usurp more than 40% of a PCB designer's time. For those products, the cost of creating and maintaining documentation can easily escalate. Some engineering organizations operate without dedicated PCB designers and require electrical engineers to complete the PCB design process. Because the average salary of an electrical engineer is higher than that of a PCB designer, this could factor into the high cost of creating documentation.*"

There are also intangible costs that are difficult to quantify. An inefficient documentation process that results in delays or a lengthy new product introduction process bears lost opportunity costs. Errors in the documentation set can result in multiple unplanned documentation revisions, re-spins of the PCBs, and errors in the assembly or part procurement process. Having a PCB designer focus 20% or more of his time on documentation rather than PCB design will delay the start of the next PCB design project. Sharing a networked PCB design product license to create documentation and design PCBs may require users to postpone critical tasks until product licenses are made available.

To mitigate the cost of documentation, begin by evaluating your PCB documentation tools and considering how well they meet your documentation requirements. Using a PCB CAD tool with minimal support for basic documentation tasks may require additional effort for working around its shortcomings. For example, what does it take to draw a complex layer stackup detail or multi-row drill chart? How easily can these details be updated when the layer count is increased, drill counts are changed or new drill sizes added? Secondly, you should examine the documentation process and documentation requirements across all members of your organization. Follow the documentation trail to learn how each member of the entire enterprise uses the documentation set.

Are they creating documentation to augment their process that is currently not part of your standard documentation set?

Over the course of the CAD tool evolution, the suite of tools required for PCB design has somewhat stabilized. Schematic capture tools are used for drawing schematics, simulation tools for analog and digital simulation, PCB design tools for circuit design, and so on. All are automated tools designed and maximized for a specific part of the electronic product design process. Automated PCB documentation tools are not new, but their use is not widespread. Many are still reliant on the outdated method of creating PCB documentation within the PCB CAD tool to meet their requirements.

You can't design a PCB with a schematic capture tool, so why are so many designers using a PCB CAD tool to create documentation? This is akin to manually routing each trace of a 16-layer 8,000 net PCB design rather than using automated routing. The results will be similar, but the effort far greater.

The old adage "Use the right tool for the job" rings true for PCB design and documentation. If you are creating PCB documentation today, you should consider dedicated PCB documentation tools, such as Mentor Graphics' Fablink or DownStream Technologies' BluePrint-PCB.

The benefits of using a dedicated, automated tool specifically designed for PCB documentation are many. As automation has reduced the tedious task of routing traces individually, so too have documentation tools reduced the tedium of creating and maintaining PCB documentation. An automated documentation tool can reduce documentation tasks, reduce errors, and offer greater efficiency. **PCBDESIGN**

Mark Gallant is senior product marketing manager at DownStream Technologies.