Kick-Starting a Revolution: IPC-2581 Meets Gerber

by Karel Tavernier
UCAMCO

SUMMARY: The all-or-nothing approach to improving the CAD-to-CAM workflow benefits nobody. The author proposes keeping what works in Gerber – the image data format – and changing what doesn't work, such as the stackup data format. And stackup is an area in which IPC-2581 excels. Other IPC-2581 sections could be integrated with Gerber in the same way, and "good old Gerber" could eventually be retired.

One of the common misconceptions in the world of PCB design centres around what happens to CAD data when it gets to the PCB manufacturer. It is often believed that the Gerber and Excellon files generated by PCB designers go straight onto the fabricator's NC equipment. This gives rise to all sorts of concerns about how the data must be delivered – whether, for example, a PCB fabricator's drill machine will accept instructions in metric or imperial units, or whether the manufacturing process can handle the resolution, feeds and speeds.

The good news for designers is that their Gerber and Excellon files never, ever go straight into the PCB manufacturing process. One of the several reasons for this is that PCBs are never manufactured as single PCBs as such, but on panels, where they are surrounded by borders and other features necessary for the production process. Incoming files are always read into the PCB fabricator's CAM system, which generates appropriate production data in whatever language and setting necessary for the facility's equipment.

It should be clear from this that designers do not need to concern themselves with how the data will work on the PCB fabricator's equipment. What they really must do, however, and here we come to the purpose of this article, is to make sure that the design data is valid, accurate and complete and can be read into the fabricator's CAM system as easily and reliably as possible.

CAD data reaches the fabricator as an archive, which are normally contained in the following formats:

- Layer images: Gerber
- Drill files: Excellon, generic NC or Gerber
- Netlist file (roughly 50% of cases): IPC-356
- Function of the image and drill files: text file or drawing
- Stack-up, materials, colors (not always included): text file or drawing
- Other manufacturing instructions: text file or drawing

This information must be read into the fabricator's CAM system. It's clear from the types of data formats listed above that it is only partially standardized and machine readable, even if the process can be partially automated thanks to advanced solutions (such as Ucamco's Integr8tor).

In this article, I would like to explore whether it is possible to improve archive structure and its automatic handling by adopting better data formats. In order to analyze this possibility let's look at each archive's elements separately:

Layer Image Files in Gerber Format

I realise this may seem like marketing hyperbole, but this is truly the most reliable part of CAD-to-CAM data transfer – confirmed by the fact that today's most complex PCBs are all manufactured from extended Gerber files, the vast majority of which will read into a CAM system without a hitch. Extended Gerber is tried and tested; it is a simple, compact, yet precise format whose unequivocal, well-documented presentation is easy to interpret. It's complete in that each layer is described by one single file, and it's portable and easy to debug, as it uses printable 7-bit ASCII characters. Furthermore, it can be read by people as well as all CAM systems with viewers such as GraphiCode's free GC-Prevue viewer. In fact, Gerber input and output processors are probably the most reliable software in the PCB industry. The freely available Gerber Format Specification[1] itself is also quite clear and explicit.

That said, I would like to underline the absolute necessity of using proper RS-274X extended Gerber files. Some archives, thankfully fewer each year, are still being transferred in the old RS-274-D Gerber format. This is totally obsolete, severely limited, must be inputted manually, problematic in CAM, and it should be laid to rest as the relic it is.

There is no need for a new format for image transfer.

> **“*Extended Gerber is tried and tested; it is a simple, compact, yet precise format whose unequivocal, well-documented presentation is easy to interpret.*”**

Drill and Route Information

Problems with drill files are almost exclusively caused by the poor or incomplete use of the Excellon format. In too many instances, so-called Excellon files contain just coordinate data and tool numbers, and the CAM engineer has to search the archive for supporting text files in order to discover which tool sizes, scale and measurement units are to be applied[2]. Some designers are even using the EIA codes that were already obsolete back in 1980. This too is a choice, but at this point, why not go the whole hog and do the documents in cuneiform script? I can recommend a good font site[3].

Note that the solution for these shortcomings is not to be found in adopting a more complex new format as this will only aggravate these issues – if files are already being written poorly in the simple Excellon format, imagine the problems in a new and more complex format!

CAM engineers far prefer to receive Gerber drill files as with proper Gerber data there are no problems in transferring drill sizes and locations. Then, when the job is completely cammed, their CAM systems will generate Excellon files dedicated to their drilling equipment.

There is no need for a new format for drill information transfer, but for better usage of what exists.

Netlist Information

Here too, a standard exists. The good old IPC-356-A standard falls short when it comes to driving today's electrical testers, but it is perfectly adequate for transferring netlists from CAD and CAM. A netlist is, after all, a simple structure. That said, the IPC file must be properly prepared – poor implementation will inevitably result in poor netlist files, a typical problem being incorrect handling of NPTH locations. This is not the fault of the format; problems are generally down to poor understanding of the application by the implementors. Using a different format will not resolve the issue and will introduce further problems. The best solution is to promote better use of the standard through education, tutorials and application notes.

There is no need for a new format for drill information transfer, but for better usage of IPC-356.

> ***"The misery starts when we get to data describing parameters such as stackup, for which there are no standalone standards at all. There is an urgent need for a standard format to transfer this information."***

Layer Structure, Stackup, Materials, Colors and Tolerances

Stackup design requires a deep knowledge. For many PCB fabricators this is an integral part of their unique selling proposition, perhaps more so than their ability to design and manufacture complex images. However, the description of a stackup is a pretty straightforward list of materials and their properties.

The problem is that there are no standards for transferring stackup information within the framework of a Gerber archive, so informal text files or drawings are the norm. As they do not have a standard structure, such files can often contain incomplete and/or unclear data, forcing CAM engineers to search through accompanying documents, contact the designer, and manually input data – practices that are frankly unworthy of a high-tech industry like ours.

To recap, we have clear standards for images, drill and route, and netlist information. The misery starts when we get to data describing parameters such as stackup, for which there are no standalone standards at all. There is an urgent need for a standard format to transfer this information.

Umbrella CAD-CAM Formats

Numerous attempts have been made to rectify this by creating total CAD-to-CAM data formats such as EDIF, ODB++, Barco DPF and GenCAM. These have all failed, or at best, have achieved limited acceptance. The reason is that they had to be adopted wholesale and nothing was foreseen to combine them with established workflows. Worse, their use imposes the use of new imaging models.

This is a real minefield because all new geometric applications used to create imaging models are initially plagued by tricky bugs – not because geometric programmers are particularly incompetent or sloppy, quite to the contrary, but because this type of programming is very difficult. *The Algorithm Design Manual*[4], for example, says that: "Implementing basic geometric primitives is a task fraught with peril…There are two different issues at work here: geometric degeneracy and numerical instability..."

And *Computational Geometry in C*[5] states in a rather resigned tone that "There is no easy solution to the fundamental problems faced here [...] There are several coping strategies…"

The TopCoders blog affirms that "Many TopCoders seem to be mortally afraid of geometry problems." The fact is that it can take years to sort out the bugs in new image formats, as *The Algorithm Design Manual* intones: "Expect to expend a lot of effort if you are determined to do it right."

Yet for the CAD-CAM transition, we absolutely have to "do it right." Errors in images, fiendishly difficult to detect, are highly likely to lead to scrap. Knowing this, CAD and CAM professionals are reluctant to rely on new image formats – take for example the readme.txt files that frequently accompany CAD datasets containing both Gerber and ODB++ data format, that give the following instructions:

BARE BOARDS MUST BE FABRICATED WITH GERBER, DRILL AND IPC-356 NETLIST PROVIDED. BOARDS ARE NOT TO BE FABRICATED FROM ODB++ FILE.

This does not indicate that there is anything intrinsically wrong with the ODB++ format; on the contrary, it is included because it may contain useful information. There is, however, a concern about the reliability of the images in the newer ODB++ format which is totally understandable given the abovementioned issues.

So the question is: Do we as an industry really want to change our reliable, known image format for one that may take us years to debug? Let's look at the facts: Images constitute by far the largest and most complex part of any CAD-CAM archive. We have already seen that this part of the data transfer process is pretty solid. The real issue is with the remaining data which, although it is no less important, is far less complex to characterize.

> ***So the question is: Do we as an industry really want to change our reliable, known image format for one that may take us years to debug?***

For example, we have a well established format for the flawless transfer of soldermask images, but we do not have a proper way to transfer information about soldermask color. Color is therefore communicated using supporting documentation, and must be entered manually into the CAM system. This is not a good practice and needs to be changed, but it makes no sense whatsoever to ditch a reliable imaging language to add a standard to describe a simple thing such as the soldermask color.

A Simpler Proposal

I would propose a simpler and safer alternative. We keep what works – the image data format – and we change what doesn't work, such as the stackup data format.

In essence, the stackup of a single-sequence PCB is nothing more than a list of material layers and their properties. Some of these layers, such as copper layers, have images associated with them. Others, such as FR-4, do not. The drill file can be viewed as an image file that goes from top to bottom. This is simple to describe accurately and completely.

A sequential-build PCB is a little more complex, but not much. Here, the PCB is a list of subassemblies and single material layers. Each subassembly is in turn a list of subassemblies and layers. At the lowest level, the subassemblies are a simple list of single layers, just like a simple PCB. Essentially, a sequential-build PCB is described as a list of layers and assemblies, and the assemblies themselves are again a list. It is an embedded structure. Not terribly complex.

For optimum CAD-to-CAM communication, the stackup must be described clearly in a formal language that leaves no room for doubt. The stackup may be simple to describe, but it takes a lot of application knowledge to define it clearly and completely to ensure that all the necessary fields are included and easily understood.

This is where IPC-2581 excels. It contains outstanding stackup definitions as it is[6], has been reviewed minutely by a very active team of stackup specialists from a wide range of PCB design and supply chain companies, and is currently being fine-tuned for the next revision, a process that illustrates the advantages of an open organization such as the the IPC-2581 consortium.

As a result, IPC-2581 is an open standard with industry consensus. It also offers the most capable stackup specification published to date, its structure reflecting the essence of a stackup, while layers with an associated image are linked to the description of that image in IPC-2581 format. Furthermore, unlike a typical CAM format which is essentially an image processing format and is therefore image-centric, IPC-2581 is PCB-centric, with its developers' specialist industry know-how built in. IPC-2581 can therefore handle the complexities of specialities like rigid-flex boards as well as a wide range of specialist materials, making it more sophisticated than alternatives such as our DPF format and Valor's ODB++ offering.

I know this from years of experience with IPC-2581. Ucamco was a very early adopter of the standard, and our software may very well be unique in that it uses IPC-2581 routinely. Integr8tor, for example, has been using IPC-2581 since 2006 to describe stackups when it outputs engineering data. Given that this data is input daily by our clients' engineering and ERP systems, integrated IPC-2581 solutions have in fact been in use all over the world for some years now. We and our clients therefore have first-hand experience of the immense advantages offered by IPC-2581 stackup as an integrated part of the CAD-to-CAM communication cycle, and are ever more convinced that this is a real enabler for our industry.

Therefore, I propose the adoption of the IPC-2581 stackup description not only by the broader PCB industry, but as an integral part of conventional Gerber archives. This would mean that for layers with an associated image, the 2581 image description is simply replaced by the Gerber file name describing that image. In other words, I propose that we continue to describe image and drill files in Gerber format, but add an xml file describing the stackup according to 2581, as illustrated in Figures 1 and 2.

This can also represent the layer structure rather than the full stackup, reflecting real life CAD to CAM workflows, which often start by transferring the layer structure first and adding materials later – and it is a route that is made entirely possible by the flexibility of the xml structure.

If such an xml structure were included in the Gerber archive, the CAM system could then read the xml file, create the proper job structure, and load the associated images with its existing Gerber input processor, without any operator intervention.

The highlighted .gbr files shown in Figure 2 point to the Gerber files in the same archive. The archive would then fully describe the PCB and contain the following files:

- Stackup.ipc2581.xml
- mm620601.gbr
- mm620632.gbr
- mm620660.gbr
- mm620641.gbr
- ImageOutline.gbr
- netlist.356

Benefits of Combining IPC-2581 Stackup with Gerber Images

Compatibility

Such archives would be compatible with existing systems, enabling the Gerber and

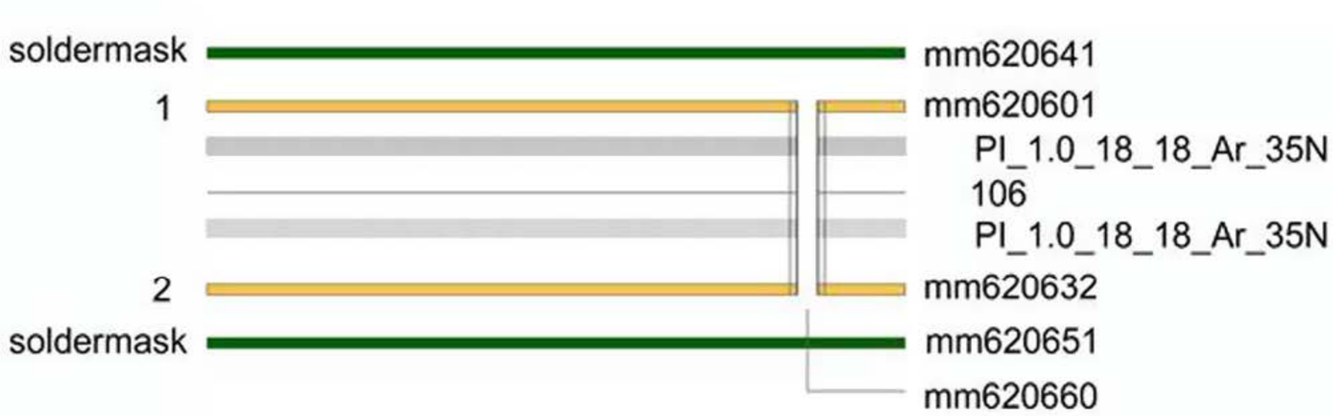


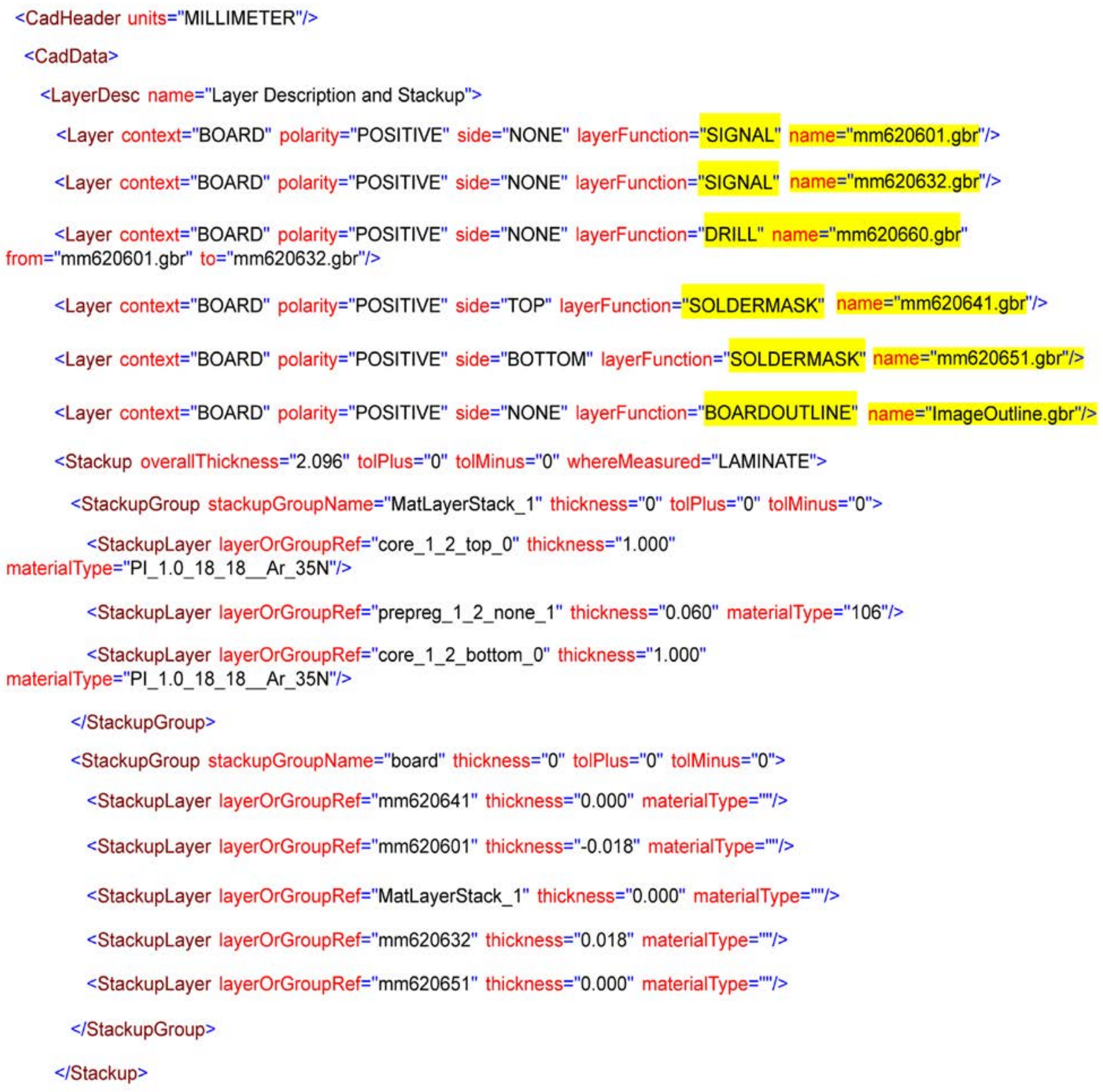
Figure 1: A simple PCB stackup.

netlist files to be read as they are now, while stackup information would be read from the documentation and entered manually as per current practice. CAM operators would reap benefits from the use of this format because the 2581 structure provides unequivocal stackup description data that can be read either in ASCII or using a generic xml viewer of which there are many available as freeware.

And nobody would be forced to buy new software, so PCB designers would be happy in the knowledge that all their manufacturers can handle the Gerber/2581 archives.

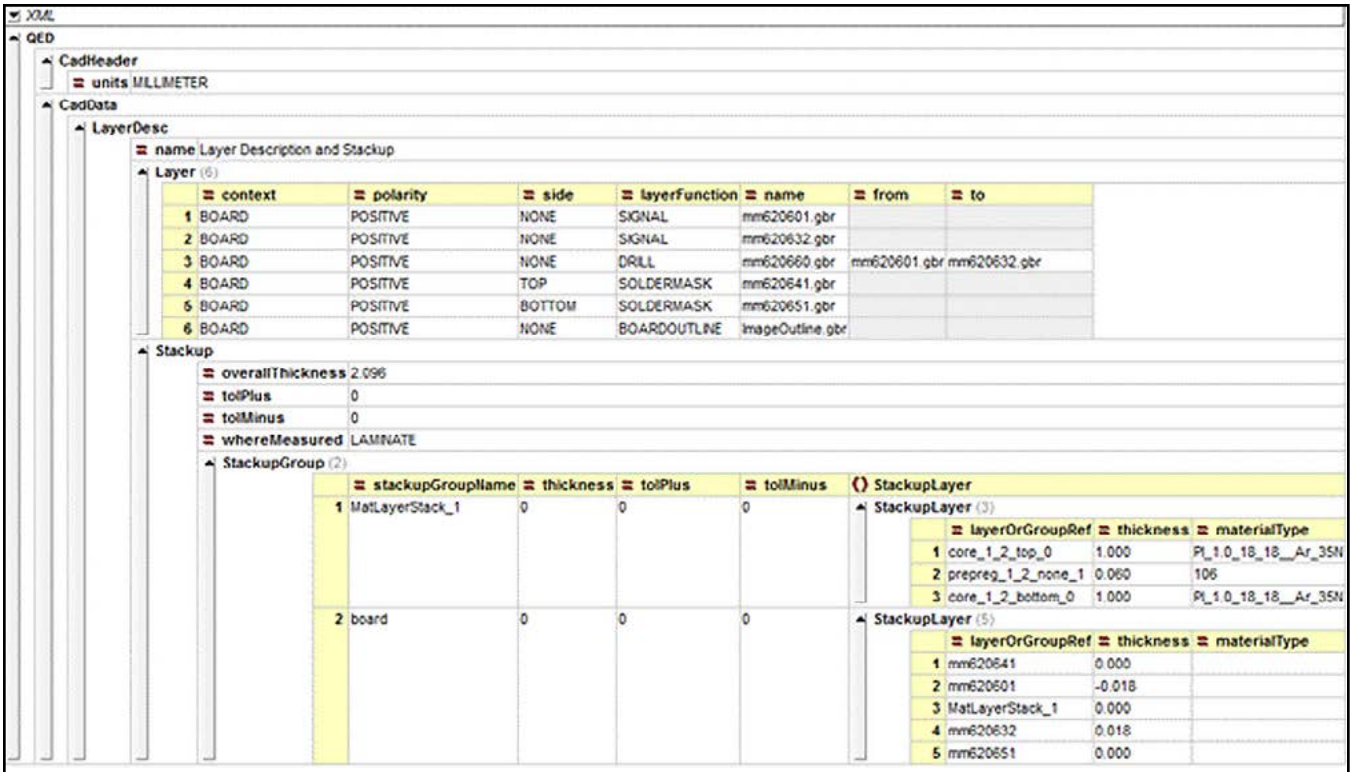
Lower cost

Implementing a new image format is a major undertaking and requires long and painstaking validation, something that would



```
<CadHeader units="MILLIMETER"/>
 <CadData>
   <LayerDesc name="Layer Description and Stackup">
     <Layer context="BOARD" polarity="POSITIVE" side="NONE" layerFunction="SIGNAL" name="mm620601.gbr"/>
     <Layer context="BOARD" polarity="POSITIVE" side="NONE" layerFunction="SIGNAL" name="mm620632.gbr"/>
     <Layer context="BOARD" polarity="POSITIVE" side="NONE" layerFunction="DRILL" name="mm620660.gbr"
from="mm620601.gbr" to="mm620632.gbr"/>
     <Layer context="BOARD" polarity="POSITIVE" side="TOP" layerFunction="SOLDERMASK" name="mm620641.gbr"/>
     <Layer context="BOARD" polarity="POSITIVE" side="BOTTOM" layerFunction="SOLDERMASK" name="mm620651.gbr"/>
     <Layer context="BOARD" polarity="POSITIVE" side="NONE" layerFunction="BOARDOUTLINE" name="ImageOutline.gbr"/>
     <Stackup overallThickness="2.096" tolPlus="0" tolMinus="0" whereMeasured="LAMINATE">
       <StackupGroup stackupGroupName="MatLayerStack_1" thickness="0" tolPlus="0" tolMinus="0">
         <StackupLayer layerOrGroupRef="core_1_2_top_0" thickness="1.000"
materialType="PI_1.0_18_18__Ar_35N"/>
         <StackupLayer layerOrGroupRef="prepreg_1_2_none_1" thickness="0.060" materialType="106"/>
         <StackupLayer layerOrGroupRef="core_1_2_bottom_0" thickness="1.000"
materialType="PI_1.0_18_18__Ar_35N"/>
       </StackupGroup>
       <StackupGroup stackupGroupName="board" thickness="0" tolPlus="0" tolMinus="0">
         <StackupLayer layerOrGroupRef="mm620641" thickness="0.000" materialType=""/>
         <StackupLayer layerOrGroupRef="mm620601" thickness="-0.018" materialType=""/>
         <StackupLayer layerOrGroupRef="MatLayerStack_1" thickness="0.000" materialType=""/>
         <StackupLayer layerOrGroupRef="mm620632" thickness="0.018" materialType=""/>
         <StackupLayer layerOrGroupRef="mm620651" thickness="0.000" materialType=""/>
       </StackupGroup>
     </Stackup>
```

Figure 2: How the stackup in Figure 1 would be described in a 2581-style xml structure.



XML
QED
CadHeader
units MILLIMETER
CadData
LayerDesc
name Layer Description and Stackup
Layer (6)

	context	polarity	side	layerFunction	name	from	to
1	BOARD	POSITIVE	NONE	SIGNAL	mm620601.gbr		
2	BOARD	POSITIVE	NONE	SIGNAL	mm620632.gbr		
3	BOARD	POSITIVE	NONE	DRILL	mm620660.gbr	mm620601.gbr	mm620632.gbr
4	BOARD	POSITIVE	TOP	SOLDERMASK	mm620641.gbr		
5	BOARD	POSITIVE	BOTTOM	SOLDERMASK	mm620651.gbr		
6	BOARD	POSITIVE	NONE	BOARDOUTLINE	imageOutline.gbr		

Stackup
overallThickness 2.096
tolPlus 0
tolMinus 0
whereMeasured LAMINATE
StackupGroup (2)

	stackupGroupName	thickness	tolPlus	tolMinus	StackupLayer
1	MatLayerStack_1	0	0	0	StackupLayer (3)
2	board	0	0	0	StackupLayer (5)

StackupLayer (3)

	layerOrGroupRef	thickness	materialType
1	core_1_2_top_0	1.000	PI_1.0_18_18__Ar_35N
2	prepreg_1_2_none_1	0.060	106
3	core_1_2_bottom_0	1.000	PI_1.0_18_18__Ar_35N

StackupLayer (5)

	layerOrGroupRef	thickness	materialType
1	mm620641	0.000	
2	mm620601	-0.018	
3	MatLayerStack_1	0.000	
4	mm620632	0.018	
5	mm620651	0.000	

Figure 3: A view stackup example in Figs. 1 and 2 using a generic viewer.

likely be beyond the reach of smaller software vendors. Make no mistake, it is a costly affair for all parties involved. To quote *The Algorithm Design Manual* again, "Expect to expend a lot of effort if you are determined to do it right." By contrast, implementing the 2581 stackup model and combining it with an existing Gerber processor costs far less. This benefits the industry in general, and it translates into lower user costs.

Low Risk

The risks involved in adopting the 2581 stackup format are negligible. Not least because this would be a massive improvement on the chaos that reigns now, but more importantly because a stackup transferred via 2581 can be verified visually for plausibility, or compared to conventional drawings. This is impossible with the highly complex layer images, where errors are likely to escape notice, enter production, and create scrap – a risk that, as we have seen, is greatly amplified when implementing a new image format.

The Route to Full IPC-2581 Implementation

In this article we have thus far addressed stackup and materials, the area which most urgently needs a standard, by proposing a solution that combines the appropriate section of the 2581 standard with the incumbent Gerber image format. The same principles could be applied to other new elements in the 2581 standard, such as its component description.

The same cannot be said for its image section. At this moment in time, there is no significant benefit in adopting it, but should the 2581 standard evolve to a point where the benefits of integrating the image description are commensurate with the costs of doing so, there would be good reason to adopt it instead of the Gerber format.

Parallels with the Printing Industry

In looking for the route forward, our industry would do well to take a leaf from the graphic arts industry, which faces challenges similar to our own. The way in which data flows within the PCB industry can be compared to how data moves within the graphic arts industry, where the printer receives a digital description, mostly image data, of a magazine or consumer package, and then produces the required number of copies.

In the 1980s, data transfer from customer to printer was even more dismal than it was in our industry. Then the "PostScript Revolution"[7] kicked in as Adobe's PostScript page description language was used to transfer data digitally. PostScript was developed through three major iterations as the industry placed ever greater demands on it, and then in the 1990s, the PDF format was created[8] . Using exactly the same imaging model as PostScript, PDF has been developed over the years to the point that today it offers powerful wide-ranging functionality, interactive options such as annotation and dialogue, and the security of certification.

Its development took time, and there were discussions over the years over whether or not to "kick out" the format's "stupid" forerunners, but as PDF expanded, even the graphic arts industry's last hold-outs were finally won over.

Today, virtually all graphics production uses PDF, a great format that enables PR agencies' magazine ads to go straight to offset print without operator intervention or even visual checks – a feat that our own industry can only dream about. And it's not because the graphics industry is any less demanding than ours: Listen to an ad manager insisting on the precise color contrast of his full page advertisement, or a product manager worrying about the shape and color of a new consumer package, and you'll understand what I mean.

The graphic arts industry got there by gradually improving its existing, functioning workflow. This in turn was made possible by progressively developing its existing imaging model rather than attempting to overthrow it.

Our industry too went through something of a revolution in the 1980s as manufacturers started to take digital data rather than film – by analogy we could call it the Gerber Revolution. But we have made little progress since then. I believe that this is because the only alternatives that have been proposed have focused on completely replacing the image format instead of addressing the shortcomings in the workflow as the graphics industry did.

I believe that we can learn some valuable lessons from the tremendous success achieved in graphic arts, and that we too should follow the route to progressive improvement by making our workflows increasingly compatible. I am not suggesting that we should aim for total hands-off operations, but I think that, with intelligent and step-by-step improvements we could foreseeably arrive at the point at which simple, repetitive boards could be manufactured without operator intervention.

> ***"Our industry too went through something of a revolution in the 1980s as manufacturers started to take digital data rather than film – by analogy we could call it the Gerber Revolution."***

Conclusion

PCB designs are typically transferred from CAD to CAM in Gerber-based archives. These leave much to be desired, but the issues have little to do with the RS-274X extended Gerber format: Proper extended Gerber files can be read in without a problem. What is lacking is a standard, machine-readable way to transfer non-image information, such as the stackup and components. In other words, the so-called problems of Gerber are not about what Gerber does, which it does superbly, but about what Gerber does not do, and was never designed to do.

This issue could be resolved simply and cost-effectively by using the IPC-2581 standard, which has a well-designed stackup description format. The problem is that as the standard is defined now, in order to use this gem, PCB professionals are also obliged also to use the image section of the IPC-2581 format. This new image format offers no material benefits, if any, over Gerber. Developing, debugging and validating a new image format is a daunting task, and carries the risk of creating a lot of expensive scrap. The industry dislikes this prospect, and has accordingly shunned new image-based formats altogether, or has only adopted them to a limited degree, as in the case of ODB++.

In my opinion, the all-or-nothing approach repeatedly attempted in improving the CAD-to-CAM workflow benefits nobody. We are currently in a deadlock because adopting IPC-2581 demands the new software simply to do what can already be done now, so it can only take off once enough users have adopted it. Yet the new software will only be acquired once the new format is used widely. IPC-2581 would in fact be adopted faster and more broadly, and its benefits enjoyed by the industry sooner and more generally, if what is new in it could be accessed without having to adopt a new image format, buy new software and upset existing workflows.

This is eminently possible, and surprisingly simple. If slightly tweaked, the IPC-2581 stackup description would allow linking to Gerber images rather than to the new image formats. Both could be combined within the same archive – an approach whose development, test and validation would cost just a fraction of the investment needed to introduce a new image format. Everyone would benefit from this: the combined format would kick-start the adoption of IPC-2581, and users, no longer forced to buy new software, would work with the new archives semi-manually, and buy the software later on. Other IPC-2581 sections such as components could be integrated in the same way. And eventually, when there are enough benefits in adopting the IPC-2581 image format, good old Gerber could finally be retired after its many long years of faithful service to the PCB industry.

For now, though, in discussions about CAM to CAM data transfer, large numbers of PCB professionals express their preference to stay with Gerber. It's not broken, after all, so why fix it? That's not to say that they like the way in which other information is currently transferred; on the contrary, they sorely need a standard for information like stackup and component data. Let us give them what they want, and need: Gerber images and a proper standard for stackup and other information.

A CAM manager to whom I explained these ideas exclaimed, "Good old Gerber files with an IPC-2581 stackup – this is the best of both worlds!" So let's follow the example of the graphic arts industry by keeping what works well and integrate it intelligently with new structures that complement and enhance it, working with care and determination towards a better way of communicating, collaborating, and building quality into our industry.

If you would like to join me in enabling all to move forward with this, I look forward to hearing from you at the PCB Forum at LinkedIn. **PCBDESIGN**

References

1. The Gerber Format Specification. Revision I1, Ucamco 2012

2. Excellon Wikipedia entry

3. Cuneiform fonts

4. The Algorithm Design Manual by Steven S. Skiena, Springer-Verlag, New York, Inc. 1998.

5. Computational Geometry in C by Joseph O'Rourke, Cambridge University Press 2001.

6. IPC 2581

7. The History of PostScript

8. The History of PDF

Karel Tavernier is managing director of Ucamco. He has 30 years of experience with software and imaging equipment for the PCB and electronic packaging industry, including sales, service and R&D. He has been in his present role since July 1995. Tavernier received a master's degree in electronic engineering and a postgraduate in computer science from the University of Gent. He holds a management degree from Vlerick Management School.