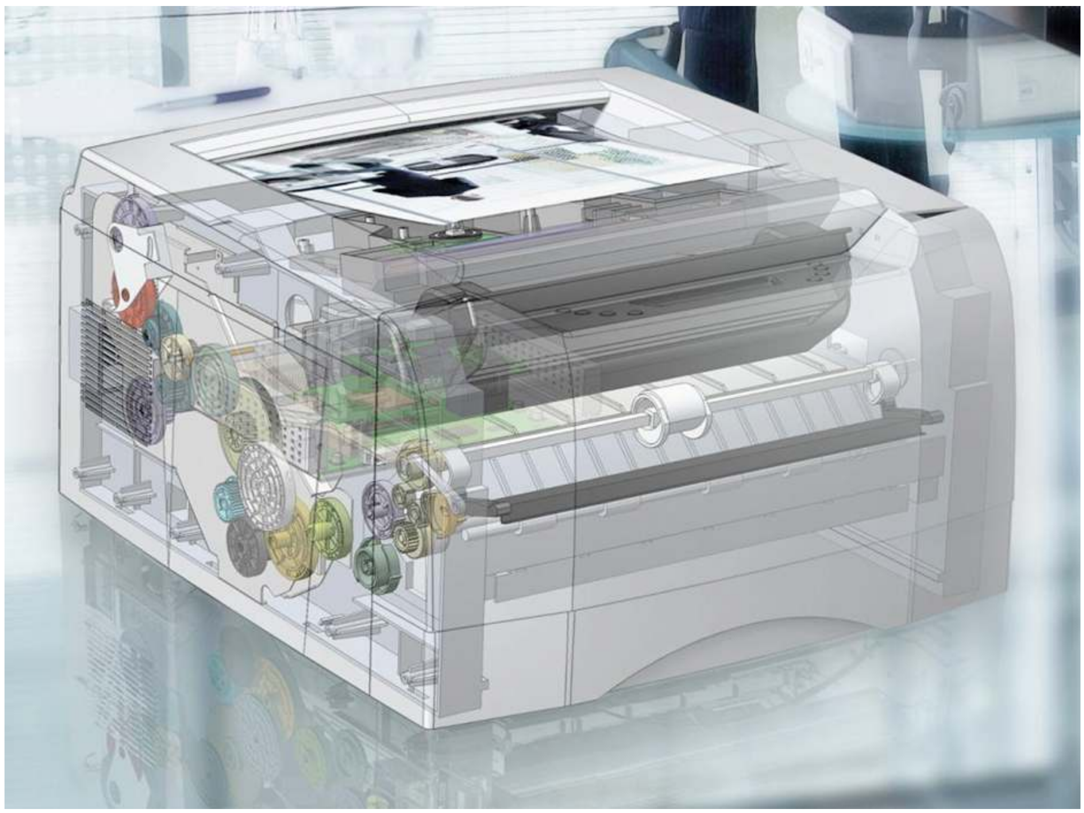
New Math

The Hidden Costs of Swapping CAD Kernels



Schnitger
Corporation

Contents

Introduction 2

CAD's Critical Kernel 2

A Brief History of CAD Kernels 4

Changing Kernels: Vendor Perspective 4

Changing Kernels: User Perspective 7

Users: Deciding Whether to Change 7

Decided to Switch: Now What? 8

Conclusion 10

Introduction

Most companies are like Helena Laboratories, a Beaumont, TX, maker of clinical laboratory equipment and reagents. Many of the products the company introduced over the last two decades are still on the market and now need to be modernized and restyled. This means digging out old CAD models, figuring out how they were created and then modifying them.

In Helena's case, the models were created in wireframe using a legacy CAD system. One of Design Engineer Billy Oliver's tasks was to convert these parts into solid models. But which solid modeler? Mr. Oliver tried exporting Parasolid files from the old CAD system for import into SolidWorks; this didn't work well enough and his team had to revert to their legacy system to complete a project. They were still using SolidWorks for new designs but the problems with legacy data and growing concern over SolidWorks' announced change in kernel, from Parasolid to CGM, led Helena to wonder: Would they have to convert parts twice, once from their current system to SolidWorks/Parasolid, then again to Solidworks/CGM? If they began to use SolidWorks/Parasolid and developed expertise there, how steep would the learning curve be in a transition to SolidWorks/CGM?

Mr. Oliver ultimately decided that Helena Labs couldn't afford the risk, disruption and uncertainty of continuing with SolidWorks and selected Solid Edge. Using Solid Edge with Synchronous Technology, Helena Labs was able to quickly create a usable and functional model of the imported dumb solids. How he came to this decision and his team's experiences since beginning the journey to Solid Edge, offer a number of valuable insights.

What is a modeling kernel, and how could a transition from one to another be good or bad for a user?

CAD's Critical Kernel

The objective: model a new component to replace an obsolete element of an assembly. In Helena's case, one of its most popular electrophoresis sample handlers had to be redesigned with new components for more effective maintenance. The designer may start with a similar part, modifying only those areas that are unsuited to the new component. He searches his existing part library, finally settling on the part below because it requires so few changes. The dimensions of the base are correct but the flange is too small. He identifies these flanges in the part's history tree to highlight them on the part's displayed image. Next, he clicks on the dimension for the flanges, backspaces over the 0.5 in and inserts 1.0 in into the blank space and accepts the change. The flanges immediately regenerate to show the new dimension. Changes are displayed as the history tree regenerates with each change. He makes a few other edits and saves the part under a new name. (See Figure 1.)

Figure 1: A typical CAD part

Each of the actions taken to display and change the model were executed through the CAD system's geometric kernel -- the heart, brains and engine of the CAD application. Kernels act as the bridge between keyboard, mouse and display and the computer's processor. The kernel turns complex commands such as "Change the dimension of the flange" into machine-intelligible instructions and collects the result for display, through the application, back to the user.

How does this work? Imagine a car. The driver presses on the gas pedal, setting into motion a process that feeds gas into the engine. The engine turns this fuel into power; this power drives the gears, pulleys and electrical system that cause the car to move in response to the driver's commands. If the car is a sports car, the driver will get instant acceleration; if not, the acceleration may be more gradual and the top-end speed is likely to be lower. Drivers have set expectations from each type of engine and see the engine as a fundamental differentiator between cars.

The same is true of the geometry kernel in a CAD system. The user creates the instruction to "Extend this flange" (pushes on the gas pedal) and expects a specific result. Using one kernel, "Extend this flange" may be executed through a specific set of geometry and display instructions (gas), while in another it might use a completely different method for accomplishing the same thing (racing fuel). Both get the job done, but in different ways. Just as you cannot easily put a sports car engine into a minivan, it is difficult to switch geometry kernels. Switching geometry kernels has significant impact on the CAD software vendor and on customers whose designs are based on the legacy kernel.

Each geometry kernel is designed to satisfy the requirements of the type of CAD system for which it is designed and then expanded as those needs evolve. Early kernels were simple geometry tools. Today's kernels are much more sophisticated and use unique algorithms to determine model quality and the continuity of surfaces, which will affect how the flange, in the example above, is processed. While all kernels will enable a CAD designer to change a flange, each kernel does it differently.

The selection of a geometry kernel is one of the first decisions CAD developers must make. They must decide whether to develop their own or use one of the commercially available alternatives. Many factors are involved in this decision, including cost and functionality, but it is most often based on the anticipated use of the CAD product. For example, in what market will this product be used? What unique performance characteristics are required? What interfaces to other products should this CAD tool have? Commercially available kernels offer the advantage of having a rich feature set and high degree of reliability; the downside is that the CAD vendor must pay a

Figure 2 - The engine is the heart of a car, as the kernel is the engine of a CAD system

royalty to the kernel developer for each seat sold and is not in control of the kernel's development. Creating an in-house kernel can take hundreds of man-years of implementation and debugging but offers the advantage of total control.

Professor Vadim Shapiro of the University of Wisconsin/Madison is an expert on CAD and geometric modeling and advises his graduate students and industrial partners to select a kernel based on reliability, the ability to handle geometric errors and inaccuracies, and the kernel's support and documentation. He also sees a commercial kernel as the better alternative, because of the breadth of capability and the likelihood that the kernel will continue to be enhanced.

The choice of kernel will directly influence the CAD users' perceptions of the stability, reliability and performance of their CAD tool. Users of CAD products based on the same kernel can more easily share geometry, so Solid Edge and SolidWorks users today can more readily collaborate than, say, users of CATIA V6 and Pro/Engineer. For some CAD customers, this is a critical consideration when planning their IT investments.

A Brief History of CAD Kernels

Kernels started out as relatively simple subroutines, written in FORTRAN, to model the boundary representation (B-rep) of a solid. A B-rep is made up of faces, edges and vertices. B-reps enable the CAD package to calculate attributes like weight, center of gravity, moment of inertia and other mass properties to accurately and completely describe the solid. Modelers can combine solids with intersections and unions to represent complex objects. Figure 3 shows how solids are combined to form a CAD model.

The first commercial kernel was Romulus, a package released by Evans and Sutherland in 1975. Romulus was a distant ancestor to today's Parasolid kernel, owned and supported by Siemens PLM Software. At about the same time that Romulus became Parasolid in 1985, a team at what is now Dassault Systèmes Spatial began to develop the ACIS kernel that was first released in 1989. Over the years other kernels have come onto the market, including Parametric Technology's commercial offering of the Pro/Engineer Granite kernel and Dassault Systèmes' CGM kernel.

Longevity plays a role in assessing the viability of each commercial kernel. Spatial says that ACIS is in use in over 350 applications with more than 2 million seats worldwide[1]. Siemens PLM says that Parasolid has over 3.5 million end-users of more than 350 applications[2]. Parametric Technology's Granite, Autodesk's ShapeManager and Dassault Systèmes' CGM are newer to the market and have more modest installed base numbers. Granite is not a traditional kernel used for geometry creation; its primary use has been to extract data from the Pro/ENGINEER database for translation to other systems, for viewing or for analysis. ShapeManager was derived from the commercial ACIS kernel and is now proprietary to Autodesk, used as the primary modeling kernel for products such as AutoCAD and Inventor, and for interoperability across much of the Autodesk product suite. CGM was first released with CATIA V5 and became commercially available to other developers in 2011. Dassault Systèmes estimates that there are several hundred thousand users of CGM-based CATIA V5 and V6.

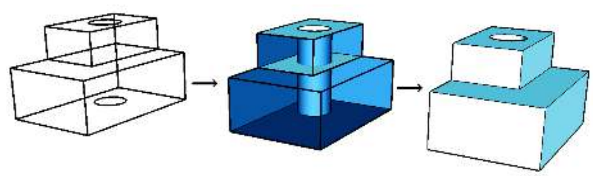


Figure 3: Combining solids to create a CAD model

There are also open source (such as OpenCasCade) and academic products on the market, although none has achieved the level of distribution of ACIS and Parasolid. Table 1 shows the most common CAD products and their kernels, as of the date of publication of this paper.

Changing Kernels: Vendor Perspective

Over the 40-year history of commercial CAD, quite a few solution providers have swapped out the kernels driving their products. Some did this because they saw limitations in the kernel they were using and wanted to offer capabilities that the kernel would not support, others for financial or reasons of control. But each transition has come at a cost.

---

[1] http://www.spatial.com/products/3d-acis-modeling

[2] http://www.plm.automation.siemens.com/en_us/about_us/newsroom/press/press_release.cfm?Component=121507&ComponentTemplate=822

	Kernel		
CAD Product (Supplier)	***ACIS***	***Parasolid***	***Other***
CATIA V4 and earlier (DS)			Proprietary
CATIA V5/V6 (DS)			CGM
Creo Elements/Direct (PTC)			Proprietary
Creo Elements/Pro (PTC)			Granite
Inventor (Autodesk)			ShapeManager
NX (Siemens PLM)		✔	
Pro/ENGINEER (PTC)			Granite
Solid Edge (Siemens PLM)		✔	
SolidWorks 2012 and earlier (DS)		✔	
SolidWorks V6 (DS)			CGM
Spaceclaim (Spaceclaim)	✔		

Table 1: Kernels in use in major CAD products

For the software supplier, changing a kernel is a complex endeavor that can be likened to swapping out a car engine -- relatively easy if one is taking out a V6 and replacing it with the exact same engine but very, very difficult if going from a four-cylinder diesel to a V8 gasoline engine or from gasoline to a hybrid gas/electric. In each case, either engine will move the car, but they will do it differently.

In a software environment, the connections between the engine or kernel and the other systems are called the Application Programming Interface, or API. Changing the kernel requires programmers to change possibly millions of lines of code in the API to ensure that the CAD package makes the correct subroutine calls. For example, one kernel's APIs looks at arc in a clockwise manner while another uses counterclockwise. Each of these differences must be identified and resolved before a CAD package is functional with a new kernel.

If the CAD package is performing as expected after the kernel switch, the development team must create a conversion tool than helps users migrate their part libraries from the old kernel to the new one. This is usually a batch process that opens each part in an assembly, recreates it from the history tree, validates that the "after" part is the same as the "before" part and highlights any areas that need human intervention. By all accounts, writing this conversion tool is very difficult because of differences in how the kernels do rounding and process edges, surfaces and so on.

History shows that even the best converters are likely to succeed with only 90% to 95% of parts, which means that the other 5% to 10% of parts must be manually reconstructed. In other words, if 100,000 parts need to be converted, 5,000 to 10,000 would need to be rebuilt, assuming that the new kernel supports this geometry.

From the CAD vendor's perspective, it is simplest to change kernels before there are commercial customers for the CAD product. SolidWorks, for example, was initially developed on the ACIS kernel but switched to Parasolid well before first customer shipment to resolve performance and functionality issues.

Once a product is in commercial use, the impact on the customer can be significant. One of the first commercial products to change kernels was Solid Edge, which replaced ACIS with Parasolid in 1998 to improve performance and increase the capabilities the company could offer its customers. Dan Staples, Director of Solid Edge Development, said that

We used ACIS for Solid Edge V1 through V4 and Parasolid thereafter -- we decided Parasolid was a better kernel and knew we had to change before our customer base got too much bigger. We worked on the kernel change for a year with a pretty big team and were helped by the fact that Solid Edge's architecture had not presumed a particular kernel.

We knew we wouldn't be able to translate every part completely. For that, you need the history tree -- at every step of the recipe -- to give the same answer in the new kernel as the old -- and if any one of a thousand steps gives a different answer, the entire history tree has to be scrapped. With today's number of customers, and the number of files they have, a kernel change would mean tens of thousands of files with unusable history trees. Wow.

The move to Parasolid was the right thing to do *but in the short term, Solid Edge and its customers paid a price for the change.*

A colleague of Mr. Staples' at Solid Edge in 1998, Raj Radhakrishnan, had a lead role in converting customers' parts from ACIS to Parasolid. He remembers that

The real problem with conversions was that parts looked right but didn't regenerate all the way back. Some of the problems were related to specific customers and configurations, and others to the kernel itself. There was no real way to predict which parts would fail. We ultimately figured out, based on experience, areas where problems were likely to come up. The biggest risk to our customers clearly was the unpredictability of it all.

Jack Beeckman, of Emerson Network Power and a longtime Solid Edge customer, remembers the confusion created by Solid Edge's decision to change kernels but said that the company handled the process well:

Solid Edge was using the ACIS kernel at the same time that SolidWorks came onto the market with their Parasolid kernel. SolidWorks said that Parasolid was one reason their product was better; the Solid Edge team touted the superiority of their kernel -- and then decided to switch to Parasolid. We all wondered why they suddenly decided to switch. But the Solid Edge team said that some of the new features of the next release were only possible if they changed the kernel. Their openness about the reasons, that there were "good sound reasons why", helped a lot. They also created tools to make the changeover as painless as they could; that, and the enhancements made possible by the Parasolid kernel, definitely improved the experience.

The examples so far have been of kernel changes by relatively new products whose user bases were small. The most mature product to change kernels is CATIA in 1999, when Dassault Systèmes introduced the CGM kernel with the release of CATIA V5. Dassault Systèmes had many reasons for changing kernels, including moving from UNIX operating systems with V4 to UNIX and Windows in V5. The V5 kernel was completely different from V4 (just as Parasolid is different from ACIS) and many users said the conversion was as difficult as moving parts between software packages from two different vendors. Customers were slow to come aboard and CATIA V4 is still being used today in several large, complex aerospace and automotive projects that involved hundreds of thousands of parts, often because of the difficulty of translating between kernels.

Over time, Dassault Systèmes improved the conversion utilities and service providers stepped in to help with remodeling when needed, but the process was still difficult. In some cases, problems were caused by differences in the way the kernels processed information but others were caused by modeling methods that may have relied on tolerances or other laxities in one kernel that were tightened up in another. Users reported significant data loss in some types of parts, and a conversion success rate that ultimately seemed to average about 95%. The remaining 5% of parts required manual rebuilding.

Nemetschek Vectorworks may be the most recent CAD supplier to change kernels. Biplab Sarkar, Chief Technology Officer, says that Vectorworks changed from its prior kernel to Parasolid in 2009 because

We needed a kernel with greater reliability and additional functionality. We wanted a kernel with a proven track record that was well-tested and well-documented. Geometric and topological differences make each kernel unique, so our conversion tool was able to convert only about 90% of the parts. Regenerating solids in the new kernel was problematic because there wasn't 100% compatibility between the feature trees -- the geometric kernels make different assumptions as they build features that can affect the way the part comes through. Tolerancing, how the kernel decides whether two surfaces are adjacent, is very different: in Parasolid, it's an absolute value but in SOLIDS++ it is based on the size of the solid and is calculated every time the solid is regenerated. This difference means that the intersection of two surfaces could be modeled differently in each kernel -- neither is right, they are just different. There's no way to automate a fix; the user must rebuild this by hand.

Professor Shapiro also sees a class of problems during conversion that can be quite subtle:

Even in cases when the conversion tool regenerates a valid boundary representation, it may be substantially different from what the user expects. Holes can move, blends will change, tangency conditions will be violated, and constraints could produce alternate solutions. This kind of failure is really difficult to catch, without performing a comprehensive check on all models -- a tall order.

In 2010, Dassault Systèmes announced that it would swap out the SolidWorks kernel at some point in the future, replacing Parasolid with its own CGM. Mr. Oliver, of Helena Labs, first heard about the change after SolidWorks' annual user conference; the company hasn't released a definitive time table or customer migration plan as of the publication of this paper.

Changing a kernel is a very difficult process for both the software developer and the user community and is not something that either should undertake without careful consideration. Luckily, the lessons that have been learned from past kernel changes, summarized in the sections below, can help users make decisions and plan transitions.

Changing Kernels: User Perspective

A vendor's decision to change the kernel in a CAD system will affect all users who decide to stay with that CAD tool. As in Helena's case, each user team must decide whether staying with a vendor justifies the disruption caused by the change in the "engine" of the CAD tool. The impact on enterprises with many users, high numbers of legacy parts and more complex processes around their CAD installations will be greater than those with more modest installations. All users, however, will face similar issues:

- Potentially significant data loss in existing parts, which could number in the thousands.
- Possible downtime during parts conversion and lost productivity during retraining.
- The expense of hiring an external data cleanup service if internal resources are insufficient to carry out ongoing work and conversion-related tasks at the same time.
- Unknown performance and reliability. Arguably the biggest risk in a kernel switch is the transition from a known, familiar tool to one that is largely unproven.

Dealing with these issues and mitigating their impact requires planning by the user community. Too, as Mr. Oliver at Helena Labs notes, it requires information:

We started reading in 2010 about SolidWorks moving to another modeling kernel, but there was no real clarity about what the change was or when it would happen. There was a lot of speculation; my local reseller couldn't tell me anything more and suggested I called SolidWorks. They confirmed that the kernel was changing but assured me that SolidWorks would support a Parasolid-based product for years to come. The problem with that is that we can't afford to become experts on a version of SolidWorks that is going to go away and really don't want to convert everything twice, once now and then again sometime in the future when SolidWorks moves to CGM. I've been on the wrong side of kernel changes before and don't want to risk going from Parasolid to the V6 kernel.

Users: Deciding Whether to Change

In the end, a CAD user can only react to the vendor's decision to switch kernels. There are really just four alternatives:

1. Stay with the CAD product and plan a transition from the current kernel to the new one.
2. Change to another CAD product that uses the same kernel.
3. Look for the most compelling alternative CAD solution, regardless of kernel and other switching costs, or
4. Do nothing.

It is undeniable that changing kernels creates risk. The question for each user team is how well that risk is balanced by the benefits of that new kernel. Each alternative carries with it pluses and minuses and their relative weight will be different for each group making the decision -- there is no one right answer for all. Some questions to consider:

- **Does the new version, with the new kernel, offer capabilities that make the switch worthwhile?** In many of the cases discussed above, the kernel change led to core usability enhancements.
- **Does using the new kernel fix problems that are relevant to your work process?** If the enhancements are not relevant to your processes, then it may not be beneficial to change.
- **How completely will your specific parts convert to the new kernel?** If your modeling process or most-used features cannot be easily regenerated using the conversion tools, consider the options of rebuilding these parts yourself or hiring an outside firm to do so.
- **How much will it cost to rebuild or remodel all parts that do not convert in a batch process?** If the cost is prohibitive, consider looking at another CAD product that uses the same kernel.

- **How much retraining is required to use the new version of your current product?** How does that compare to the training required to make your team competent on another CAD tool?
- **What other processes are affected by changes in the CAD kernel?** For example, what third party tools may no longer work because of a kernel incompatibility? What interfaces will not work? How long will it take internal or partner resources to create compatible solutions? How much additional cost will be incurred? Many third party products are connected directly through the kernel, but others use their own or neutral file formats.

One important additional consideration, according to Jack Beeckman of Emerson, is justifying the decision to management:

Everybody is getting more and more lean and needs to be more productive. Even if the conversion tool is 95% effective, that 5% fail means we need to spend time to fix things. Can the vendor show that the benefit of the change, the justification, creates a payback that makes it worthwhile? If a part takes 5 minutes to regenerate today, but requires a minute or two to fix after the conversion and then 3.5 minutes to regenerate every time after that conversion, it's worth it to make the change. We, as administrators, need to know this so that we have the ability to sell that to our management. But without that information, we can't make the decision.

Deciding to do nothing, the last of the four choices listed above, is simply delaying the point at which a crisis will occur. Eventually, the CAD vendor will have to stop supporting its legacy product and what is today a process that can be planned and managed can turn into a fire drill. Delaying the transition to another CAD product or kernel is perfectly reasonable until the CAD team is ready and able to switch -- but all the experts advise: don't put it off too long.

Mr. Oliver's Helena Labs team chose the second alternative, opting to replace SolidWorks with Solid Edge, which also uses the Parasolid kernel. His team has a good working process to migrate parts from SolidWorks and their legacy wireframe modeler to Solid Edge and feels that this will help Helena Labs remain competitive. He believes that, while SolidWorks continues to migrate to the CGM kernel, Solid Edge developers are continuing to push the envelope to create better, more usable features within Solid Edge. For Mr. Oliver, perhaps the biggest benefit was getting the transition over with and eliminating the uncertainty caused by SolidWorks' impending kernel change.

Table 2 lays out a conceptual framework for evaluating the costs of a kernel change.

Decided to Switch: Now What?

If you opt to stay with your existing CAD vendor through a kernel change, the first step after the decision is planning. Convert all parts at once? When? Or do it in stages? Train users now? Later? Hope they pick it up informally?

Planning a kernel switch is a bit like staging the rollout of any other major product update, with the added complexity of ensuring that parts are converted before designers need them.

Many companies have found that it is easier to do this type of migration all at once, rather than piecemeal. Batch converting the parts ensures that everything is on the same kernel, and has been translated using exactly the same settings and methodology for consistency. If all goes well, nothing will need to be converted "on-the-fly."

History shows that, if a conversion tool can be developed, it might successfully migrate only 90% to 95% of parts, and that the remaining 5% to 10% of parts will require anything from minor hand editing to a major redesign. The odds of success can be increased by doing a pre-conversion cleanup, choosing the appropriate options (if offered) in the conversion utility and checking parts at key points during the conversion itself.

Experts Billy Oliver, Jack Beeckman, Raj Radhakrishnan, Biplab Sarkar, Vadim Shapiro and Dan Staples suggest the following for coming through a kernel transition with the least amount of disruption:

- Get as much information as you can, as early as you can, and verify the vendor's claims by speaking with early adopters and reference accounts. Consider, too, what your current supplier and all others you're looking at offer in addition to basic CAD: is there some feature offered by a CAD supplier that is so attractive that it trumps all other concerns?

- Be sure that you understand the benefits of the new kernel. What processes does your supplier say will be easier? Have reference customers who have made the switch seen these benefits? Test the new version of software using your processes and practices. Set

	Option 1: Same CAD tool, different kernel	Option 2: Same kernel, different CAD tool	Option 3: Different CAD tool and kernel
Tasks	Convert all parts to new kernel, retrain as needed	No parts conversion, learn new CAD tool	Convert all parts to new kernel, learn new CAD tool
# of parts to convert	100,000	0	100,000
Time to convert using automatic tool	2 minutes/part	0	2 minutes/part
Conversion success rate	90%	100%	90%
Manual time to rebuild	20 minutes/part	0	20 minutes/part
Retraining on new product	2 weeks/user	2 weeks/user	2 weeks/user
Fixing connections to third party products	TBD	0	TBD
Total time to productivity	3,000 hrs compute time + 30,000 hrs manual rebuilding + 80 hrs retraining/user + cost of fixing connections to other products	80 hrs retraining/user	3,000 hrs compute time + 30,000 hrs manual rebuilding + 80 hrs retraining/user + cost of fixing connections to other products
Total cost of conversion (depending on labor costs)	**Cost of training plus** **Part Conversion: $750,000 - $2,000,000**	**Cost of training**	**Cost of training plus** **Part Conversion: $750,000 - $2,000,000**

Table 2: A framework for considering the cost of migrating CAD kernels

up a test environment for a few weeks to verify whether everything works as the vendor promises.

- If your testing proves out the vendor's claims, consider how to upgrade to the new version. Will these benefits make it worthwhile to consider converting your parts? If so,

- Try to understand the differences between the kernels. How do they process self-intersecting surfaces, calculate tolerances and handle other geometric and topological issues? What parts of your models and your modeling process are pushing the envelope of the new kernel? Will you be able to model these features, for example, in a different way or will the kernel fail? If the kernel is up to the task,

- Understand how your vendor's part conversion tool works. What settings can you use to maximize success, given your modeling practices? How do they (and you) determine if a part correctly regenerated?

- Is your supplier's support organization geared up to assist? Does the documentation for the kernel and conversion tool help you work through problems? Does the vendor have resources allocated to fix files when things don't go right?

- Recompute your most critical 10, 20 or 30 parts on a test basis and thoroughly check the results. If they don't convert quickly and cleanly, reconsider.

- Proceed with caution since conversion failures can't be predicted. Recompute at the lowest possible feature to make sure parts are regenerating correctly.

- Plan the transition very carefully and allocate enough time to rebuild key parts before production work resumes.

- Check the results of the conversions immediately afterwards. Do not file converted parts away for long-term storage without first checking that they converted correctly at all levels. You don't want to try to figure out why a part doesn't regenerate two years after the changeover with no idea that it is because the part failed to convert.

- Convert all current parts at once. You don't, as Jack Beeckman said, want any confusion between converted versions of parts and archived versions.

- Understand how your PDM system will interpret the conversion. Will you be assigning new version numbers to converted parts? What does that do to your control scheme?

Conclusion

A CAD system's geometry kernel is the heart, brains and engine of the application. Most CAD systems were designed with a specific kernel in mind but can, with time and care, be modified to rely on another kernel for application-to-machine instructions. Each kernel does, however, have its own strengths, weaknesses and peculiarities, which can make a change from one kernel to another difficult on users. Part libraries must be converted, connections to third party programs and other interfaces must be tested, and all changes must be carefully rolled out through design organizations. In the end, each user team must decide which is less disruptive: Converting parts to take advantage of the benefits of the new kernel or moving to a different CAD package, either to maintain the same kernel or because this change represents the opportunity to reevaluate past choices. Each alternative involves risk but also the potential of great benefit.

Mr. Oliver, of Helena Labs, sums it up this way:

We opted for Solid Edge because we were able to keep our Parasolid parts in Parasolid -- it's a known, proven technology and we're comfortable with it. We chose Solid Edge because it offered the capabilities we needed for our modeling process. Synchronous Technology, the sheet metal modeler, the steering wheel ... we've cut our design time tremendously. We achieved 100% conversion success in going from SolidWorks to Solid Edge and have a feeling of calm and certainty about where we're going to be in a year. That's a good feeling.

About this report

This report was sponsored by and created at the request of Siemens PLM Software. Schnitger Corporation has used the best information available to us in preparation of this report. It includes our interpretation of information in the public domain or released by responsible officers of the named organizations. The analysis and opinions in this report reflect our judgment as of the date of publication but are subject to change without notice. Schnitger Corp. is not liable for any loss or injury resulting from use of this information. All trademarks are the property of their respective owners.