Presentation by David Madeley, Siemens PLM Software

So basically we wanted to talk about what we've got with our turbomachinery module, and take you through some details about it and really give you a status of where we are.

So the turbomachinery module was something that we delivered when we looked into the challenge that people were facing. We started to look at this in NX 7/NX 7.5. Up until then, we didn't have a best-in-class product for that at all. The generic solution in NX 6 was really inferior to the market needs. Customers really wanted industry terminology inside the product. The general-purpose operations, we felt, weren't the way to go.

[~0:1:00]
We got a lot of pushback about things getting more and more complex, and when you try and use general-purpose operations for things like blisk machining, we have more complexity, so that was a clear steer that we shouldn't go that way. And we could see that a one- or two-version project couldn't put out an end-to-end solution. We also recognized that long-term planning ahead over five or six versions isn't optimal because we can't predict what happens in five to six years with technology. And the feedback we got from customers is that relying on major releases meant there was too big a gap. So turbomachinery was one of the first modules where we started to use more of an agile approach, which I'll talk about.
But what we did was we tried to look at part families, so we looked at multi-bladed components, and we kind of categorized them as a part family and thought about the things that we needed that were complementary technology, like turning and that kind of thing, to blisks, impellers, this kind of multi-bladed component.

We considered turbomachinery multi-blade components as a family, and then we started to think about what we needed to do to deliver best-in-class functionality for machining these parts, this family of parts. So instead of enhancing the existing processes, we created dedicated processes for these parts. We created dedicated geometry groups. We built in turbo language terminology. So a hub's a hub; a shroud's a shroud. You're not using things like variable contour or the existing processes and trying to understand the map between generic terms and the real names of the products that you're trying to machine.

With the experience that we've gained with the turbomachinery module, we're also looking at applying this same kind of rationale to rotational parts, to ports, things like tire molds. So we want to provide these dedicated kinds of processes moving forward that will use the terminology of the industry and give you exactly what you need.

So what we came up with was a series of NX CAM processes that are absolutely dedicated to rotational multi-blade products, like blisks and impellers. We came up with a roughing processor, rest milling processor, hub finishing, blade finishing, fillet finishing, and then we're still working on the edge finishing processor. But these are new processors that are completely optimized for turbomachinery components.

We came up with new geometry groups that use the terminology of the industry. So you can define a hub; you can define a shroud, the blade, the blend; you can define a splitter. You use the MCS and the workpiece from the existing solution. But the multi-blade geometry is new, and also the part axis and the number of blades is a new parameter for defining the geometry groups.

And the goals, really, of doing this is to deliver within NX 9 and beyond a superior product to what's available in the market, so that it becomes a best-in-class implementation. We want superior quality. We want the best possible efficiency and the minimum machining time. We really want ease of use. These are complex components, but they don't have to be complex to program if we do things properly. We want robustness, because we can't always guarantee the quality of geometry and things like that, so the processors have got to be able to handle it.
And we want something that's scalable, that can grow with your PLM implementation if need be, so it needs to be implemented in the rest of the framework of our products. All the pictures that you see during this presentation are parts that we've cut with customers using turbomachinery. We're really very proud of the progress that we've made and the components that we're machining with this module, and everything that you see is customer data.

So, again, associativity to the solid model, to the design, all the good stuff that we get in all the other parts of CAM we wanted inside the turbomachinery module. We wanted really intuitive geometry selection, visual feedback. We really wanted the out-of-the-box templates for turbomachinery to reflect the best practice, what we were learning as we went along. So as we've machined parts and we captured good parameters and good conditions, we constantly modified the templates so that they're the default when you create an operation.

We want the system to support all the downstream activities using the master model, creation of drawings, creation of manufacturing work instructions. We need to cope with really, really poor geometry. We need to be able to smooth bad modeling. We need to be able to adapt to poor input from your customers if you're a subcontractor machinist. And the ease of use and the out-of-the-box experience for a novice really has to be top-notch.

So we were kind of delivering an evolving product, and we still are. Out of the high-level requirements we identified back in NX 7/NX 7.5 timeframes, we picked those that would deliver a minimal viable solution. The early releases, we had some issues with the quality of tool paths, but we were actually able to cope with the complex geometry of impellers and blisks, with no CAD manipulation. We'll talk a little bit about that later. We've always been able to machine these parts. But when you look at customers that have been doing it with products before NX 7/NX 7.5, the amount of geometry that they were actually creating, the construction geometry, to allow them to create the tool paths was unbelievable. And I've got some comparisons to show you the difference.

We started to work really closely with a number of customers to identify the next steps that we need to implement. And we've got a list of enhancements and continual development that we maintain, so we're working closely with customers.

We're gathering their feedback, and we've got this list of the direction that the customers want to take this product. The process is very much still ongoing. It will continue, and it's the main way that we define the product evolution, working with customers.
For this particular product, this is one of the first products that we started to develop in a very agile way. So everything about this product has been a customer collaboration. We picked customers that want to do this with us, that were committed to do it with us, and everything that we've developed has been customer tested. So we develop software. We test it with a customer. We cut blisks. We cut impellers. We make changes to the code, and we give them another DLL for them to test, and that's the way this product's evolved. And this is going to be a lot more of what you see moving forwards in terms of our development strategy.

But we work with the customers. We set the expectations for the scope and when we can deliver what they need. When the code's ready, we can provide it to the customers for them to verify it and try it. In most cases, it was tested within a couple of days, and we got feedback. So they actually stood up machines, machine parts, and gave us feedback. And then the enhancements are provided to everybody. So it goes into the product, and anybody using the product gets the enhancement.

And then the process really has helped the customers, because the customers that are collaborating with this module and the ongoing development of this are really ensuring that it supports the use cases and that the issues are addressed. But also, other customers get a more mature product with a better set of defaults. And from our point of view, it helps us to make sure we develop the right things and make sure that we go in the direction that you guys are telling us we need to.

The solution releases. So solutions are developed on top of the recent available version, the current release version, where we'll provide for testing DLLs or maintenance patches to customers that are collaborating with us. Or we provide the next version to be released as a beta with the code already inside of it, for customers that are working to develop and test with us. Solutions are always available on the current release with a patch or on the next available release. We made a conscious decision that we don't backport. This product's moving forwards. We're not wasting development time to backport to previous versions. We're actually spending the time and the money to get the changes into the versions that are moving forwards in the existing version.

And as long as an enhancement was not proven production-ready or being used for production parts, then we hide it using an environment variable, which means we can expose it to the customers that are working with us. The other customers can get at the standard functionality of this module. And if they want to engage with us and begin to do testing, we can share the environment variables and all of the currently-in-development aspects of this module with them for the test. And that's the only way, really, that we can control the audience to the preview functionality and the test.
functionality and get feedback. So, again, conceptually, we deliver DLLs. You can test those DLLs against your current version, and then we roll it into the next release. And in an agile way, this is the process that we go through as we go forwards and step through the development lifecycle.

[~0:12:00]
So, many things have changed. I'm going to start up the software and show you a little bit about it, but we don't really have major releases for this product, because we have stuff, we have functionality, we have new capability as we go along, as we get point releases.

Blend finishing. New type of operation to finish the blends of multi-bladed parts. Controls very similar to the blade finish. Full control over the cut area. And it can be based on a reference tool diameter. You've got individual overlap control on the hub and the blade. You've got all the normal sequencing options that you've got as per the rest of the operations in turbomachinery. And it can work with or without the blends actually being modeled on the geometry. It gives incredibly smooth and gouge-free tool paths. One of the things that we've done within the whole of the turbomachinery module is really focused on tool path smoothing.

[~0:13:00]
Other things since the last release. The new tool axis options which we've added to the multi-blade operations for better tool axis control. You can control the rails at the top and the bottom of the blade, and you can actually slide the tool along those rails using the manual drag handles to create interpolated tool axes. While you're moving the tool, you've got dynamic preview of collisions and that kind of thing.

Safe locations are displayed in green; gouging is displayed in red. You can configure the tolerance, and you can actually see, as you're dynamically moving the tool path, whether you're going to cause a collision, whether you're going to cause a problem, whether you're clear of a blade, or whether you need to take different tool path measures. The processor will also automatically tilt the tool away as needed so that we don't get gouges. So we will try and optimize the path with the processor to make sure it's gouge-free.

[~0:14:00]
The blade finishing. So we added the blend and blend-finished tangential extensions so that you can go out and around splitters and blades when you're machining. Again, traditionally, all of this kind of stuff would've been a modeling activity. So prior to the turbomachinery module, you would've created quite a lot of surface geometry to give you geometry that you could use variable contour or sequential milling operations on. We don't have to create any additional geometry inside this module now. We can actually define all of these parameters inside the operations.
The alignment of the output. So I don't know if you can see the bottom right-hand picture, but we had all sorts of issues with the distribution of points on the splines being very misaligned. We've actually done a lot of work in the algorithm to try and align the points from cut level to cut level so that you don't get the banding effect that you sometimes get when using a high-speed machine in a multi-axis, especially where you've got lots and lots of fast tool axis change. So this is just something that I wanted to go -- I wanted to pull up NX and give you an example, because on the left is this component programmed in NX 5, and on the right is the same component programmed in the multi-blade module. So I'm going to switch to NX and just really give you an overview of the difference.

So this is basically -- I've just opened this. This has been saved from NX 5. This is an impeller. Already you can see that in order to rough and semi-finish, we've got a pretty extensive list of operations, cavity mill, and a lot of variable contour, where we're kind of -- lots and lots of straight-line moves, lots and lots of sharp corners with very little smoothing. You'll also notice on the geometry that there's a huge amount of additional modeling. I've actually got a lot of it switched off, but there's a lot of faces and planes created. There's a lot of associative modeling that's been done in the manufacturing part to extract the design geometry and give you the additional surfaces, give you the horizontal surfaces, to give you closed areas so that you could drive sequential mill around the pocket to effectively rough out and this kind of thing. And this was exactly what we wanted to alleviate when we went to the turbomachinery module.

If I switch to the same component programmed with NX 7.5 or NX 8, the first thing you'll see is that these are the operations. So instead of that huge complex list, we've got just a few operations. We've got no construction geometry. The geometry is the design. We've not had to manipulate the geometry. We've not had to do any additional modeling in order to create the tool paths. We didn't have to extract surfaces. We didn't have to create closed areas.

If we look at the geometry view, this is the multi-blade geometry, this one object. And if I edit it, you can see we've defined the hub here; we've defined the shroud; we've specified the individual blades, the blade blend, and the splitter. We've given it a number of blades, and we've given it an axis of rotation. So this is all the geometry that we need.

Within the operations themselves, like for example the multi-blade rough, if we look at this, you'll immediately begin to see that the tool paths and the control that we've got, there are no sharp corners. There's no long engage-and-retract where we're pulling
right the way away from the part. They're all very smooth tool paths. The connections are all optimized for high-speed machining, for the type of machines that you do this on.

The finishing operations. You can see that we've got very, very optimized tool paths for finishing. Even in the central areas where you would tend to get overlap, we've actually got good constant tool path.

So there's a huge difference between what we had in NX 6 and what we have now within NX 8.5 and coming in NX 9, in that we think we're now at a point where we've got really an industry-leading multi-blade module. We've done machining trials with the aerospace companies. We can use a general-purpose geometry group like this. We can create the tool paths, and we can get better output than companies that are specializing, so we're really, really pleased with the outcome of this. And we're not done. There's still a lot to do. If we go back to the presentation, we're working with these lead customers, and we're getting a lot more input into the things that we need to do and the gaps, and it sometimes surprises us.

Creative solutions – we've been given blisks, for example, where the blades are not evenly spaced around the disk. So we've managed to use what we've already got and define the blades on a blisk as blades and splitters, and actually vary the distance between them, and this kind of thing. So even with what we've got today, we're able to do creative solutions with them. So we can deliver something now, but we can capture it as a requirement and deliver something moving forwards.

With large blades, we've got some really quite large blades, which are kind of roughed in segments, starting at the tip. You'll do a first segment. You can't get right down the blade because of tool wear. So you end up with a scenario like this, for example, where you see a banding as you change tool and actually come back in to machine the part for the finishing.

We've created – well, the solution is to actually create smooth overlaps. So at the end of each path, you kind of create almost an offset, kind of trumpet or cone kind of shape, where you can gradually disengage from the material, and when you change tool and come back, you engage onto the next section with a similar kind of profile.

So this colored segment shows how we're able to take multiple operations in the solution today and keep adding blade offset stock to create a perfect blend between the different patches, which gives a perfect finish. This has been taken as a requirement, and in the DLL that we've got today for the customer that's testing this, we've actually created a new kind of operation that can do this automatically. But, again, we were able to very quickly and in a very agile way turn this around and make the changes and put
something out there that makes it easier for the guy. It doesn't have to create five or six operations to do this. It can now do it with one operation.

[~0:22:00]
There are some big enhancements for smoothing. So there's some environment variables that we've put in place that really optimize smoothing. Smoothing can mean a lot of different things. It can be the non-cut portions or it can be the cut portions, and it can be tool path optimization or it can be optimization based on the control system of the machine. And there's multiple environment variables and multiple things that we've got in place today that we've got customers trying for us and testing, all of them coming back with very favorable results that will then be included into the product.

[~0:23:00]
We've got projects where we're looking at this kind of thing as well, specifically edged around this kind of functionality. And as we go, as we constantly evolve this product, we're constantly capturing and defining best practices. Every time we work with a customer, every time the customer gives us feedback, every time we learn something new, we change the templates that we provide with the software.

[~0:24:00]
We add better settings, because the goal is that a novice will be able to get a good tool path. You shouldn't have to be an expert to make these components. The default settings that we provide you with should be the right settings to give you a good-quality component. So we capture those best practices, we document those best practices, and we build them back into the technology that we deliver to you in the next version.