MICHAEL CHASSER: All right. We'll go ahead and get started here. Can everybody hear me in the back fine? Can you hear me back there? All right. Well, this is 3-D Modeling Practices and Standards for Substations. Just in case you're in the wrong class by chance, feel free to walk out. You won't offend me.

So my name is Mike Chasser. I'm with Burns and McDonnell Engineering out of Kansas City, Missouri. I'm the Substation Design Section Manager. As you can see here, I'm a substation guy through and through. 12 plus years of substation design experience, from 12kV all the way through 765kV. And a little bit of experience with Inventor and Vault, so a couple of years. So I'm very much a substation guy that has learned and adapted to the Inventor environment. And co-speaker is Jason.

JASON RUGG: Yes, I'm Jason Rugg. I'm just the opposite of Mike. I've been with the company for about two years now, but I have over 12 years of Inventor involved experience.

MICHAEL CHASSER: All right. So what we're going to talk about today. So we're going to go over some industry best practices on 3D substation design. And to be blatantly honest, these are a lot of things that we messed up on to get where we're at now. So hopefully you guys can take this and avoid some of the missteps that we had to get where we're at.

And this will be covering using Inventor, as well as the SBS Design Suite. We'll talk about some Vault integration, and how we can use Vault to maximize the efficiency of your 3D substation designs, and capitalize on the data. And we'll go over some processes, so we can get consistent, flexible, and usable 3D models.

So key learning objectives here. We want to learn how to produce models that are accurate, consistent, end user friendly, again. Talk about best practices for both part and assembly creation. Again, use Vault to capitalize on your 3D data. And then learn how to use Inventor to transform the design process. So really what that last one is is how do we use the tool kits on top of Inventor to really design intelligent 3D substations.

So what we found out through all these trials and errors that we've done is that the best place to start with your model development is go to the end, and figure out what it is you want these models to do. And it seems kind of counterintuitive.

But you want to look at it and say, all right, at the end of the day, what do I want these models to be used for? Some folks just want it to be used to get data. Some want it as a pretty picture. Some want it to do training, maybe in the future. So there's a wide array of things that you could use these models for. So you want to take a look at it and understand what it is you want in the end, to know where you have to start.

And you also want to consider who's going to be using the models. Is it going to be engineers? Is it going to be experienced Inventor folks? Is going to be entry level users? And those type of decisions will help you decide how do you go about developing the models. Does that makes sense to everybody?

So then you've got to make some decisions here. And notice we're not even in the process of developing models yet. We're still at the roundtable discussing everything. So anybody in here not familiar with the terms visual fidelity and visual identity? OK. Perfect.

So visual fidelity and visual identity. Visual identity is exactly what it sounds like. Can I look at a part and tell what that part is very quickly? Can I look at an arrester and know that it's an arrester without questioning it? Or has it been modified so that it looks more like an insulator maybe? Or a bushing of some sort? You're not quite sure what it is just by looking at it.

Visual fidelity essentially says, how much detail do I have in the model. So if you have an arrester that has a low visual identity, you have a low visual fidelity. So you're not quite sure what it is, but you've reduced the amount of data or the complexity of the model down. A lot of times what this comes into is computer performance.

So folks always ask the million dollar question of, how much detail is too much? What's the minimum amount of detail that I need? And that's really this visual fidelity versus visual identity argument here. So we'll talk a little bit about this later during the best practices. And how you can use the tools in Inventor to control some of that. And what's some best practices that we've found.

So then you look at what models should be developed. You could take many different approaches to this on how you want to do it. We've seen folks that will compile a huge master list of every part in their data system, and say, we want to develop all of these. We've see folks that will say, hey, we want to do just what's necessary for our pilot project. And then another project after that, just what's necessary for that. And build their database that way.

And also, is it everything new? Or do we also go back and build legacy parts that may be used sometime in the future, but you never know? So a lot of utilities have a lot of legacy parts. So you've got to decide whether you want to actually go back and develop those. And the model development process, again, is how do we go about deciding what models we need to develop.

All right. So continuing on making decisions here. After you've determined what the models are that you're going to develop, you have to develop guidelines on how are you going to develop them. This is often some of the most important stuff that you do because it drives either internal or external folks that are doing the modeling for you, so that you get consistent models.

So that you've developed the guidelines that tell them exactly how the model should be developed. And again, we'll go through some more best practices stuff that gives you some ideas on that. But documenting these guidelines is vital to making sure that at the end of the day, you have consistent models.

How are we going to track, and organize, control your models? This goes into some of the Vault stuff that we'll talk about a little bit later. But you want to consider things like folder structure, file naming structure, which is the next one. File naming structure is a huge one. This can vary utility to utility, customer to customer.

Folks, is anybody in here not familiar with the SDS consortium? So there's a consortium of member utilities and a couple of AE firms out there that-- what we're trying to do, as a whole, is come together and develop some best practices to try to do things as a standard way throughout the industry.

Some of the things that are difficult is things like file naming structures. Where one utility may name their files 161kV breaker. And then some may name it their catalog ID number, or whatever it may be. So when you look at this, you've got to decide from your file naming structure, what's important? What do you recognize for your company as a good file naming structure for your models?

JASON RUGG: And I'd like to add to that. You need to be consistent with your file naming scheme. And you need to get that developed up front through trial and error. In working with some of our clients, we've run into some big problems when they haven't nailed down what they want their files to be named. And they've come back and either changed it or they've done something to modify that process. And that's caused a lot of headaches for us.

MICHAEL CHASSER: And to piggyback on that, if you have a file naming structure for parts and assemblies that's consistent and/or predictable, does anybody know what some of the things you can do? If I can predict what a file name is or I know based on a file naming system what a file name is, what can I do in the future with those files? Does anybody have any ideas?

AUDIENCE: [INAUDIBLE] ID number. And so we go to build [INAUDIBLE] everything but the [INAUDIBLE]. So you just go in through your Vault, because we have everything vaulted. [INAUDIBLE] and that's how we know [INAUDIBLE].

MICHAEL CHASSER: Yes sir.

AUDIENCE: [INAUDIBLE]

MICHAEL CHASSER: That's a very good point. And one of the things we're really trying to strive for is, can I look at the file name and know exactly what's in there? Where I may look at one client's file name and it may be named 1234, that doesn't mean anything to me. If I look at another one that says 161kV Seamans Breaker, I know what that is, but that may not apply to everybody.

One of the other things you can do is-- kind of endgame, you could started using iLogic or the API to start automating processes. If I know, and I have a consistent, or predictable file naming structure, we could start automating processes down the road, once you get libraries built, and all these assemblies built. Does that make sense? End game over where we're going? Any other questions on that before we move on? All right.

So best practices for part and assembly creation. First thing you need do is develop your project file. Is everybody in here familiar with what a project file is and what it does? All right. I'll take that as a yes. So this will, obviously, map to your templates. You want to develop templates that are complete and well-thought out. This is just a list here of the templates that you may have. There could be many, many templates that you have. There's no predefined number.

One of the things that you want to be careful of, and why I said complete and well-thought out, is that if you develop a template now and then six months from now you mess with the template a little bit, you start getting inconsistent files. You may have hundreds files you started from the old template, and then 1,000 files you get from a new template, and they could mismatch. So you started developing inconsistencies from that point. And that's really what we're trying to get away from with best practices. So haven't well-thought out templates is a big key to understanding what you're doing.

JASON RUGG: And if you do start changing stuff, you'll have to decide how you're going to correct that. And then if you do make changes are these templates stored someplace that you can easily get all the new changes pushed back out to all the users. So that they're up to date.

MICHAEL CHASSER: All right. So we already talked about developing a standard library parts and assemblies. That can be new. It probably should say, and/or legacy, depending on what you decide for your organization is best.

As a general industry practice, which I've specified that the z-axis is always up for parts and assemblies. And this goes back to your templates. Set your templates up so that it's always like that. So that anything that you-- when you do extrusions and things they're in the z-axis. And all your sketches are on say, the xy plane, unless for some reason it makes sense that it goes anywhere. But we try to standardize on that type of stuff. And we've got our templates set that way.

That makes constraints a lot easier and a lot easier to figure out later. If something's going wrong you can figure out-- you have logical constraints that are made when you consistently sketch on the same plane and you have consistent access rotations.

Sketches said we fully constrained when it's applicable. I would say that's a 99% rule. A lot of times if you have some really complex geometry, say you're doing a section cut of an insulator, and you have all the bells, and you want to revolve that, you're doing a sketch of that, that could be a complicated sketch, and doing the revolution.

One thing you could do, which I don't recommend you could do, but you could use the auto constrain feature. And you'll go in there and throw every constraint that you need on there to auto constrain it. And you can say, oh, well, it's fully constrained. But what happens then is you have no idea, if I need to change something, what dimension do I go in and change. So you're really hurting yourself there in the future. Do you want to talk about what we did with the patterning of the shed's?

JASON RUGG: So yeah. That could be done a couple of different ways. Going back to that auto constrained feature, a good place to use that would be like if you're pulling in a bunch of AutoCAD geometry, you're just copying and pasting. That's not going to be constrained any way. It would be very difficult to go back and constrain all of that geometry, especially if it's really complicated. So you can through that auto constrain on there and it will just constrain it for you.

Another way of doing it is you could just do part of a bell, revolve that, and you could array or pattern that bell down.

MICHAEL CHASSER: How about the block geometry?

JASON RUGG: Yeah. Another way you could do it. If you're pulling in a bunch of AutoCAD geometry, you don't want to do the auto constrain, you could actually make a block out of that geometry you've imported in. Make that a block and then you can just constrain just the block.

MICHAEL CHASSER: But in general, you should make sure that all of your sketches are fully constrained. If you don't, you could start to have funny things happen with your model. Especially if people go into the sketch and start moving things on accident, or changing dimensions. It may make the sketch do weird things which will make your model do weird things.

So we've found out filling out iProperties while you're developing parts seems like-- OK. Everybody should know that. But a lot of times you get in the mode where you're just making parts and you're not really worried about the iProperties.

You can go back retroactively, especially if you're using Vault and things like that, and fill out the iProperties. We've found it to be a better practice to fill them out while you're doing it. So as you do the part, when you think you're done, fill out the iProperties. Check them and get it into Vault. And then we can also, at that point, set-up property compliance and things of that nature involved to make sure that the proper iProperties are filled out. So you can do that later, but it hurts. It hurts a lot. We've spent a lot of time doing it.

JASON RUGG: The thought was like you said, we're focused on modeling. Filling out properties can be a real pain if you're just copying and pasting from an Excel sheet or looking stuff up online, wherever that information's got to come from, and then just filling all that out, that's painful. And we thought at one point, get the models done. Go back and through all the iProperties in later, but we've found that that was more of a headache than just filling them out as we went.

MICHAEL CHASSER: And the other one here is use base origin points, or planes, or axis for constraints. So these are your xy, your yz, your zy axis, or your base inventor, your native planes, points, and axis to constrain. One thing you don't want to do is constrain to a feature, or constrain to other things that may move.

Because if you change a feature, but you've constrained it, all of a sudden you have constraint errors. So you want to make sure that, definitely, you never constrain to a feature. And as often as you can, use your base origin planes and points for your constraints to anything you're doing in an assembly.

All right. So End of Part Marker to control feature visibility. So you can see here-- what we did was we pulled a bolt straight out of content center. And what you can see here on the left side is the End of Part Marker is essentially, like you would do it when you build any part.

So they did the base of the body and then they started putting features on this part. And what we did, was we literally just took the End of Part Marker and moved it above some of the features. And you can see here, we went from a 205KB file, to 158KB file. It's the exact same file. And the features still exist, they're just suppressed. And Inventor doesn't recognize them. So this is a very small difference in file size right now, but think about it on a much greater scale. When you may have 1,000 of these in a large assembly.

So by utilizing this End of Part Marker and moving it above your features-- and also, when you develop parts, doing all your feature modeling at the end, so that you can control what's shown and what's not. This really, really helps control computer performance.

And when you get into very, very large substation assemblies, we've had some that are on the order of 100,000, 100,000 parts, it can make a huge difference in what you're doing. It seems something very simple, but this is a very easy way to control that.

This goes back to visual fidelity and visual identity. So this has a high visual fidelity high visual identity. This one you can still kind of tell it's a part, but you can't see all the features. You've got to ask yourself, is it worth it? Do I have to see the threads, and the champers, and all those other things?

JASON RUGG: You've got a question over here.

MICHAEL Sure.

CHASSER:

AUDIENCE: Is there a way to do a global pull it forward to show a feature? Like a [INAUDIBLE]. Like 100,000 parts? [INAUDIBLE] is there a way to make it render really quick?

MICHAEL CHASSER: iLogic.

JASON RUGG: Yeah. The first thing that comes to mind would be iLogic to control that.

MICHAEL CHASSER: It would be a lot easier in iLogic to move it to the end than it would the other way because you'd have to have something to specify to stop it. But you're saying, OK, if I move it up and I put all these in there, but then for some reason I do want to see all these, can I move it down? It can be done through my iLogic? Anything else?

JASON RUGG: And that would be another reason not to constrain to your features. Because if you're moving the End of Part Marker up, and you suppress a filet, or a chamfer, or an extrusion that you're constraining to, that will take that out. And you will break your assembly.

MICHAEL CHASSER: You sir.

AUDIENCE: Well, and the other part is if you want to show movement in an assembly, you're destroying the teacher's block in the [INAUDIBLE]. [INAUDIBLE] and then show movement later.

MICHAEL CHASSER: We did the same thing. You're not in that boat alone.

All right. So the next thing here, this is just a test we did. We had a couple thoughts of, how do we control computer performance, especially when we get into these huge assemblies? How do we control computer performance?

And so we said, OK. A lot of times when you're modeling, especially in substations, a lot of things are the same, right? So logically, you think when you're modeling, I'm going to use patterns. I'm going to use mirrors. I'm going to be very efficient modeling and everything's going to be great. And that works.

What works better is using multiple instances-- say a switch, so you have a three phase switch. You develop one phase of it. And the logical thing is, as a modeler, is to say, well, I'm just going to pattern this three times over. I'm going to array it. And I've got a three face and it's perfect. And that works and you can do that.

But what we found that works better is doing multiple instances and constraining them. And you can see here-- so what we did was we set up a base assembly. We did some things with patterns and mirrors. And we just did a basic like turning operation. We were watching our computer performance. And just doing it that way we were at about 15% to 30% of the overall computer performance.

So we said, what happens if I take out the mirrors, I take out the patterns, and I just use multiple instances, constrain each of them, and do the exact same operation? What happens to my performance? You can see it went from 15 to 30, to 10 to 20, by doing that exact same thing.

So you can see how we're already starting to improve computer performance, right, just by modeling practices. So it may be a little bit more time on the front end development, but you're helping computer performance on the back end a lot. And this wasn't a very big assembly. This was something simple.

JASON RUGG: This is very small.

MICHAEL CHASSER: But this is representative. So then we said, all right. What happens if I just ground everything? And notice this does not say, removing constraints. This is literally just grounding. Once you ground that, Inventor recognizes that a parts been grounded, or anything's been grounded, and it stops calculating your constraints.

You don't want to remove your constraints. Because what happens if you want to come back in and move it later? Then you got to constrain them. Say your face spacing changes from eight feet to 10 feet. If you remove your constraints, you got to go back in, unground it, reconstrain them, and ground it again. So leave your constraints, but ground everything.

So if you're in a high level assembly model, theoretically, you could have everything grounded. And you can see we've taken it from where we began and cut that in half at the worst case. Does this makes sense? This is a big thing when you're looking at very large assemblies that some of the substations get. And when we're looking at 100,000 part assembly, this becomes a big deal.

All right. So verify degrees of freedom. Is anybody in here not familiar with degrees of freedom in Inventor? So degrees of freedom is a big thing. If you have either your rotational or your translational degrees of freedom, and you go in there, and somebody that's maybe not experienced is messing with things, if you have degrees of freedom, they can click on something and freely move it, rotate it, things of that nature.

So you want to check your degrees of freedom. And it'll tell you what part, and whether it's a rotational or a translation degree of freedom, and how many of them you have. Use your constraint constraints to solve those, so that you don't have user error or human error if they're less experienced users.

JASON RUGG: And that too, will even animate that movement that it thinks it still has there.

MICHAEL CHASSER: Yeah. There's an animate button. And it'll zoom into where it's at. And it'll show you that it can rotate, or move up and down, or turn, or whatever it may be. And then also, utilize your analyze interference tool.

So you can see here, what we did was we just made this happen. Inventor will let you do things like this, where things will clash. And here we have an example of a cable coming into the cable barrel of a terminal, and it's clashing. So it wasn't quite centered.

So at a high level when you're zoomed out, you may not see those type of things. But if you utilize the analyze interference on the entire thing, it'll tell you, hey, here you have an interference. And it even tells you your total volume of interference. So you can go in there and adjust your cable, so that you don't have those things. Because really, there should be no excuse for things like that, especially when you're in a 3D modeling environment.

JASON RUGG: And I don't know that that wouldn't cause a computer performance. But when you get into doing drawings and stuff, it's going to look really messy.

MICHAEL CHASSER: It's going to look off. All right. So if you take anything from this entire thing, modular design approach is probably the number one thing you should take away from this. Jeff Cowgill has a class tomorrow, 3:00 or 4:00-- is it 3:00 to 4:00? Somewhere around 3:00 to 4:00. But Jeff has a great class on this. It's going to be very similar to this and piggyback off the modular design approach.

And really these three bullets here, list out exactly what this modular design approach is. And this is where you're going to gain a ton of efficiency in your modeling process here. So you're going to develop your parts. And you're going to place those parts into commonly used, I term them, component assemblies.

Basically, you can see here, these two legs of an a-frame for this particular client, that constituted a part. So then you place your component assemblies into subassemblies, which are larger assemblies. So subassemblies made up of component assemblies. Component assemblies made up of parts. And then you take the subassemblies and put them into your master assembly, which would be your entire substation.

And we have some examples here. So this would be an example of a component assembly, where you can see the legs would be constituted as one part. They constituted the cross arm as another. And then, obviously, the legs are another part.

You could also do something like this. You may consider adding the foundations as part of your component assembly, if your foundations are always included with that. So it's very flexible. You can include more than just this into a component assembly.

And this would be what I would term, a subassembly. So you can see, it's made up of component assemblies. We would have a component assembly that is the switch with the foundations, and the steel, and the breaker, and the foundation is there. So it's a bunch of component assemblies making up a subassembly.

And then from there, you can see highlighted there. That is a subassembly placed into a master assembly. And what this does is reduces the amount of constraints you have in this master file because it's likely going to be a very big file.

And if you're in here and you're saying, well, I'm just going to start building an assembly. And I'm going to start placing all these components into this assembly. You're going to get to a point where you're not going be able to open your file. There's going to be so many constraints that this thing's recognizing. It's going to become unmanageable.

So Inventors only going to recognize the constraints that are in this file. It's not going to dig down and recognize all the constraints that you have for your subassemblies and your component assemblies. Does that make sense? So modular design approach is huge.

Level of detail is great. Hopefully everybody is familiar with the level of detail. Big for when you're doing very large stations. A lot of times we'll set-up a level of detail for each voltage class, so that I'm not utilizing a 69kVR and I'm working on the 230kVR and things like that. It helps computer performance, again.

JASON RUGG: And if you're consistent with that level of detail name, you can use the link level of detail. And in a top level master assembly, you can activate that same level of detail down the tree of assemblies there.

MICHAEL CHASSER: Keep your browser tree organized. So that's, obviously, just good modeling practice at that point. You can create folders and subfolder under that to try to keep things organized as best you can. And utilize the enforce unique file names feature within Vault, so that you don't have a whole bunch of instances of the same exact file name. And you have no idea what's being used where. You get into very hairy situations. Andrea is my IT lady. She's yelled at me like, 1,000 times, when I mess this thing up.

JASON RUGG: That is also an on your desk recommended best practice too.

MICHAEL CHASSER: And it's as simple as a toggle within Vault. But that's where it goes back to your file naming structure. Make sure that it's flexible, so that you don't have to use the same file name for something. Questions, comments, concerns on any best practices? All right. Jason will talk to you about Vault.

JASON RUGG: OK. We're going to get onto Vault. This is going to be pretty high level. Vault can be set-up many different ways, just depending on what your organizational needs are. To begin with, you want to think about your user groups and permissions. How that's set up is going to determine how you're able to interact with these models in Vault, doing revisions, being able to check in and out, just all different kinds of functions that will control.

You also want to think about your folder structure.

MICHAEL CHASSER: You want to avoid getting way too many subfolders. One of the big things we ran into was when we're trying to transfer data back and forth, and we're doing pack and goes out of vault, you can run into-- if you get subfolder, after subfolder, after subfolder, plus a long file name, you start running into character limit issues. And it won't even let you do the pack and go.

JASON RUGG: Vault, that character limit is based on Windows, whatever that character limit is. I want to say it's like 160 characters or something like that, but you don't want to exceed that.

Lifecycles. You can use those to enforce adherence to your policies and procedures. And then you can use states to progress through these processes. We have a lifecycle set-up where we go from a work in process to a Q3 review, a self-review. And then we go to a Q4 review, which is a peer review. And then from that it gets [? IFR'd ?] to the customer. And then from there, they can comment back changes or whatever they'd like. You can also use categories to enforce these policies and procedures that your company might have set-up for quality control.

MICHAEL CHASSER: And so we have differing entities within one company. So what we're trying to get at here is if you have different QA/QC procedures, say for your transmission department. But your distribution department has a differing QA/QC process, you can set them up under different categories. And then the lifecycles that are assigned to those categories will control how those files move. So if you have different processes you can do that, depending on how your organization is set-up.

JASON RUGG: Copy design. This is a big one. A lot of people still try to do the File, Save as Copy, on their models. Using the copy design tool, it makes that a lot easier. It's more automated. You have the ability to go in and individually select components in assembly and whether or not to reuse them or copy them. And you can also set-up some prefixes or suffixes that automatically get appended to that file name.

Vault can also be used for managing your metadata. You're able to grab an entire list of parts. And you can mass edit all those iProperties. And that's proven to be very useful, especially whenever-- you can just get in there. And you can copy and paste directly from Excel, just the whole column of information in there.

It's also got some pretty advanced search capabilities. If you've got the metadata iProperties filled out, Vault can search on it. File tracking and linking. Vault knows what's happening with those files at all times. Who has it checked out. When it's been checked out. What changes have been made. Where parts are used.

MICHAEL CHASSER: And the Where Used can be very helpful when you're trying to figure out, if I make a change to this particular model, what else is it going to affect? What other assemblies are using this part that I'm about to change? So that tells you, well, if I don't need it to affect these other things that's connected to, then I need to do a copy a design. And get it to a different file name, so that I'm not making changes to the things that I don't need to be making changes to.

So I find the Where Used tab one of the most useful things in all of Vault.

JASON RUGG: And you can also step back in time too. If you've made some changes and you don't like that, you can go back and restore a previous version of that file.

Vault reporting. Vault comes out of the box with some standard report templates already set-up. You can use these reports to report on sales, engineering, project management, suppliers, purchasing, whatever you'd like to do. But you can also set-up your own custom reports.

We've been working on a report that, basically, will give us a cost breakdown of an entire station. Do you want to talk anymore about that?

MICHAEL CHASSER: Yes. So one of the things that we're doing through Vault reporting is that we're reporting on an entire assembly. Say, a substation assembly, and we have cost data associated with each of the parts. We can run a report and know what the total cost is.

And then to tack onto that, then we can go and assign construction labor units, overhead costs-- basically, any other data that we want to develop an entire full blown cost reporting, almost at the click of a button. And these are things that have taken weeks in the past to develop. And you have to go through different departments. So if we can link our cost database with our Vault database, and make sure that those two databases are talking, we can get real time reporting on how much it's going to cost us to build an entire substation.

JASON RUGG: Any questions on that?

MICHAEL CHASSER: How many folks in here are using vault? A good number. A good number.

AUDIENCE: Meridian.

MICHAEL CHASSER: What's that?

AUDIENCE: We're using Meridian. [INAUDIBLE]

JASON RUGG: I don't know on that. I've never used Meridian.

MICHAEL CHASSER: I don't know about that one. No.

AUDIENCE: [INAUDIBLE]

JASON RUGG: OK. Now we're going to get into more of the actual SBS specific tools. So these are some pretty powerful tools that SBS has created specifically for substations. SBS is just an add-in for Inventor.

MICHAEL CHASSER: Is everybody aware of what happened with Automationforce Autodesk? And now SBS? Is anybody not aware of that? So it formerly was Automationforce.

JASON RUGG: Automationforce, they were the original developers of this add-in. Autodesk bought them. And here recently, probably a couple of months ago I would say, they turned this product over to SBS, Spatial Business Solutions, I think. Autodesk still owns the code for the software, but SBS is now managing the licensing and development of that product.

AUDIENCE: [INAUDIBLE]

JASON RUGG: So in these tools you've got Above Grade Tools. You can insert different kinds of connectors, cable bus fittings. You can do rigid bus routing, cable routing, and lightning protection. Those are just a few.

MICHAEL CHASSER: And so out of the box, you can see the top ribbon there, it comes with a bunch of different types of fittings. That's straight out of the box what you get with the SBS toolkit. But if you go in and you develop your own models then, you can go through the Content Editor and you can start specializing your models then.

So if you have your own models that are specific to your company, you can modify that to look exactly like you want it for your own company. So it doesn't have to look like that. You can add more, or less, or categorize them, or any number of things. This is very customizable.

JASON RUGG: So Below Grade Tools, it's basically grounding trench in conduit. We're still working in these areas. We've got some clients that have come up with some other ways of doing things. So it's going to be kind of a learning process for your company, how you want to do that.

MICHAEL CHASSER: And we've had some clients that have considered building a substation at a zero elevation. And there is no grade in the yard, which for most folks is not reality. And so we do have some clients that have very complicated site grading. And so we've had to come up with ways to get around, how do we route conduits that actually go with a greater trench, that goes with the grader. Your grounding that actually goes with a multigrader or a stepped site.

So out of the box, the tools are not necessarily going to do that. So you're going to have to develop your own ways of doing that. There's tools that have been invented that will help you do it. We've found some workarounds to do it. But out of the box, these tools won't let you do it.

JASON RUGG: You can finagle your way to make it do it. I know SBS. They are developing these tools to, hopefully, allow for slopes. So hopefully that works better in the future too.

AUDIENCE: [INAUDIBLE] simple 3D drive [INAUDIBLE].

MICHAEL CHASSER: You can import simple 3D data. But when you use the grounding tool, or the conduit tool, or the trench tool, it essentially puts it on the xy plane.

AUDIENCE: OK. So it's more of a 2D analysis that you're going to have ground?

MICHAEL CHASSER: Right. So what happens is you may have you're grading down here below the xy plane, but then you put your tools here, and it's not quite right. So that's where you got to start getting creative. So you can import simple 3D data, but when you start trying to then lay in conduit, grounding, trench, you're going to have to get creative to actually make it below grade at that point.

So a lot of clients at this point are still feeling this out because it's very new to the electric utility industry. So they're really saying, hey, we're just going to start at zero elevation. And that's the beginning point for most clients at this point.

AUDIENCE: But we are pushing for using that simple 3D [INAUDIBLE].

MICHAEL CHASSER: We just had them in our office a few weeks ago going over this ad nauseum. I mean, really hammering on them that substations are not built like this. We need to do these certain things. So we're really pushing them. The SDS consortium is pushing them as hard as we can to make these things more real world.

AUDIENCE: [INAUDIBLE]

JASON RUGG: And as you can see here, you've also got the Design Check Tools. I believe these are all based on IEEE standards, phase to ground clearance, phase to phase clearance, fence safety, live part to roadway clearance. These tools, they run very quickly for the most part. And then it will spit out a report and let you know if you've got any issues with that.

MICHAEL And so the clearances are based on IEEE, but you can customize it through the Content Editor

CHASSER: for your specific clearances if they exceed IEEE or if you have different clearances other than IEEE. You can customize that for your organization.

JASON RUGG: Turn it back over to Mike here for ACAD E.

MICHAEL CHASSER: So there are also, I guess, 2016 and beyond offers in AutoCAD Electrical Integration Tools. The way that they're pushing this is through what's called an Electromechanical Link or an emx file. So it sets up, you have an AutoCAD Electrical project and you have an Inventor project. You link them via emx file. And you can facilitate interdisciplinary checking.

They're also working on Enterprise Tools. So they're also coming out with Enterprise Tools, which is basically all back end SQL database stuff, that will allow the same thing to happen. So there's two different ways that Autodesk is looking at this. So it's either through the emx file, which will allow you to check breaker rotation.

If somebody is in an Investor model and they rotate a breaker not thinking about what's going to happen on the protection and control side, it'll throw you a flag in either one of them. And say, hey, these don't match. That way you don't have CT's out of orientation or leaving your breaker out of your zones of protection, things like that that happen now in the 2D world.

And then you can easily create smart schemes, smart wiring, things that will automatically update. So this is definitely a new push on their front, but they're tackling it from two different ways, through the Enterprise Tools, and the SQL database, and this emx file.

And that's pretty much all we have, so I'm going to open the floor for questions, comments, concerns. Anybody have experiences with this that we could talk about? Is anybody in here just kicking the tires on this trying to see if it'll actually work?

AUDIENCE: When you're looking at component versus [INAUDIBLE] was at what point are you [INAUDIBLE]? So if I'm on a breaker and I identify that as the equipment [INAUDIBLE] and the substation model, looking at exactly what you have. Are you looking at the breaker minus the bushings [INAUDIBLE]?

Those types of decisions become pretty critical as you move forward into [INAUDIBLE], insulators. Do you define all that within the Grounding Tech Tool. [INAUDIBLE] things. And it was to me very, very easy to get-- really a lot of the duplicate stuff. [INAUDIBLE] breakers [INAUDIBLE] and I select the assembly, it's not defined as a breaker. So the moment I define it as a breaker [INAUDIBLE]. And finding some of that stuff was rather interesting. So I'm just curious of your experience in [INAUDIBLE]?

MICHAEL CHASSER: It's normally at this-- it's got to be at the component assembly level when you get there. So you got to be back here. Obviously, this could easily just be a breaker, and a foundation, and other stuff. So you have to define it where you can define that piece as a part.

So you wouldn't be able to do it then back here, where those parts are part of an assembly. It would have to be back at a component assembly level where you just have, say, your breaker as a single part. Anybody else have anything? There's got to be more questions than this.

AUDIENCE: My only question is [INAUDIBLE]?

MICHAEL CHASSER: No. So there's certainly workflows for brownfield. The vast majority of what we see clients doing is greenfield, just because that's the low hanging fruit. That's the easy stuff to knock out. It can certainly be used for brownfield applications.

A lot of times what we see folks doing is say, you have an existing AutoCAD drawing with some raster. And you're just adding a bay. Then you do the bay in 3D and we'll just draw a match line and say, see Inventor drawing, whatever. And then on the Inventor drawing, we'll draw another match line and say, see AutoCAD drawing, whatever.

And then as you go back and you start replacing other things, start adding onto the Inventor drawings and taking off of the AutoCAD drawings. You're moving that match line back and forth. That's been the best way that we've found. Does that make sense?

AUDIENCE: Similar to [INAUDIBLE].

MICHAEL CHASSER: I wouldn't say it's similar to the workflow for that because you could define smart points on a raster image in AutoCAD Electrical, and it could still be a smart electrical drawing. So I wouldn't say it is necessarily like that. Electricals, that's a whole different whole different animal.

AUDIENCE: Electrical works a lot more fluidly than [INAUDIBLE].

MICHAEL CHASSER: It does.

AUDIENCE: And for us, the ideal approach [INAUDIBLE]

Where do the wires go kind of stuff. But yet, minimize the amount of work I have to do. [INAUDIBLE]

MICHAEL CHASSER: Exactly.

AUDIENCE: And that's the way we're [INAUDIBLE]

MICHAEL CHASSER: So you, essentially, use it as a greenfield approach. You use it as a greenfield approach, and over time, you're going to build a greenfield site, basically, as you rebuild things. But good question. Anything else?

AUDIENCE: [INAUDIBLE] AutoCAD everything's going to talk to everything. [INAUDIBLE]

MICHAEL CHASSER: I think that's the end game.

AUDIENCE: [INAUDIBLE]

MICHAEL CHASSER: That's the end game. All right. Well, if you guys have nothing else, we're up here for questions, comments, concerns, if you have anything. I have business cards up here. Thank you very time.