MATT WASH: Well, good morning, everyone. Congratulations on turning up to your first class. 8 AM's good. Thursday 8 AM? We weren't too keen on that slot, so well done. Thanks very much.

Quick introduction-- my name's Matt Wash. I'm the structural digital design leader for Australasia for Arup. This is Graham Aldwinckle and Xav Nuttall.

Quick class summary, what to expect in the next hour-- so we'll be going through looking at how you can increase productivity, minimize the waste in your workflows, look at interoperability between a number of packages, primarily using Revit as a documentation tool. And then we'll get into the nitty-gritty of the real powerful parametric stuff, with Dynamo and Grasshopper, and look at how Open BIM, so whatever software is the right tool for the job, how that will be able to help you with collaboration with all of the stakeholders on your job. We'll be hopefully looking at it from both complex and simple solutions to many different jobs, so it's not just for complex projects. And yeah, just reiterating, hopefully, you'll be able to take away some efficiencies back to your company and learn some of the future skills that might help you moving forward. That's pretty much a summary of what we've just said.

So moving on to the first part of the talk, this was a project that came into my office a couple of weeks ago. And I was asked, how would you detail the steel connections on that job? It was an interesting question. So I went away and thought about it, and thought, well, I need to know a little bit more about this job.

So I was asked-- this is what we want to achieve. We want to achieve some pretty traditional 2D documentation, along with some interactive views, a nice 3D PDF. So that was the scope of the work. So I wanted to know a little bit more.

So the goal of the study was to obviously minimize the duplication of effort, maximize the flow of the data between the analysis and the documentation. And the scope was to produce LOD300. So there was no scope to do 350. We weren't being asked to do fabrication drawings. It was just information to be provided to the fabricators, so that they could do that detailing.

And the current workflow to get to that point was Rhino and Grasshopper. Then we pushed that into our analysis, which is GSA. And then from GSA, we pushed it into Revit. So that's the point that we were at.

So a quick show of hands, who would detail that steel connection in Revit? Wow, not what I expected, not what I expected. Who would do that in Advance Steel? Good, good. Who would do that in Tekla? Oh, good, good.

And anything else that I haven't mentioned, because I didn't see many hands go up? So 2D detailing in CAD? Right, yep. And anything else? Any other offers? OK, all right.

Well, we decided we'll have a look at everything and work out what we think is the best way to do it. So in order to work out what the best way was, these were the questions that we asked ourselves. How many variations of that type of connection are there? If there's a one-off bespoke connection that only needs to be done once, do you really need to do a fully parametric model?

How are you going to be checking the shop detailer's model and the shop detailer's drawings moving forward? Can we have a workflow that seamlessly integrates into that? So not only the shop detailer benefits from what we're doing, but we benefit from what they're doing, when it comes back to check.

Who's going to be working on the job? What are the training and resources needs of the guys that will be doing the work? And then balancing that act of do we really need to build fully parametric solutions for everything, or can we get away with doing some of the stuff in Revit? And obviously, we need to look at architectural impacts, constructability, and that shot detailer workflow, and how we can collaborate with all of that downstream.

So looking at the first option, this was to do this directly within Revit. So there's a few different ways that we can do that. So the first way that we looked at was building the parametric components directly within the framing and the column families. The second way was to use independent generic connection families. And the third way was just to model in place. And then obviously, the last part is to do the fully parametric stuff in Grasshopper and Dynamo.

So if we take a simple framing member, this is what we needed to do to basically get that to work for a spliced slotted cap connection. So we needed to build in about another 20 parameters, with some visibility parameters to turn it on and off, if you wanted to, obviously, show it not show it, which was fine. Similarly, for the baseplate connection, about another 20 parameters. And we're able to detail that pretty easily.

Where Revit came into difficulties with this is when you have relationships between multiple members. With the baseplate, you're on a slab. The slab's never going to change. It's always in that same position.

When you look at this particular connection, or any connection like this that relies on the angles and size of those intersecting members, yes, you can build that into a Revit family. But it's pretty cumbersome. It's pretty slow. So in this particular solution, rather than building that within the family, we decided just to have a fairly generic framing member and then just bring in some void families to cut those angles off. Not ideal, but if there's a one-off detail, it still might be the best way to go.

And then doing it in place-- if this was a one-off connection, doing that in place in Revit could probably be done in an hour and a half, two hours. Why build a fully parametric solution that's only going to be used once? If you're confident the geometry's not going to change, maybe that's the way to go.

So the other problem that I had with doing it within the Revit environment is it's very difficult to manipulate that data directly in Revit. Only the guy that built that family really understands what baseplate vertical v1 is, baseplate vertical v2. It's quite hard to understand.

So yes, it's very quick to make those changes, because you can just go in, adjust the properties, and away you go. You don't really require any training. Everybody knows how to change parameters within Revit.

But it is hard to distinguish what those functions are. And it's not linked to anything. It's not linked to your engineering calculation. It's a manual input. So there's obviously double handling of information that's from analysis into documentation.

So the second way, and probably a better way of doing it, is why don't you export those parameters into Excel, link Excel to your engineering spreadsheet, push your engineering spreadsheet in, populate the parameters, bring it back in again? So this is a much better workflow, in that you're not double handling that information. But the disadvantage to this one is if you're making changes in your Revit environment, and you don't continuously sync that with your Excel document, you could be making changes. And then you bring your Excel changes back in again, and it overrides everything you've done within Revit, so a bit of a caution on that one.

And then the third option is what I call dumb Dynamo. So why not just make those changes in Dynamo with a nice visual image, so that you're very aware of what are the things you're changing? So you can embed the image within the Dynamo script, have the same parameters, but drive that through Dynamo.

So that's obviously very quick to make the changes. You've got the images in there that help explain what you're trying to do. But the disadvantage is yes, you obviously have some basic Dynamo training requirements. And again, this solution isn't going to work with anything that's requiring relationships between those members.

So let's look at the fun stuff. How would we do that, if we were going to do that in Grasshopper or Dynamo? So the first thing we looked at was doing it within Grasshopper, pushing that through Geometry Gym, which is a plug-in, and then getting that back into Revit. The reason we used that workflow was because up until this point, Grasshopper was the tool of choice for the parametrics. So why not keep it in that environment and see how that works?

So there's some pros and cons, again, to doing that. The pro is, obviously, it integrates with our workflow up to that point. But the con is that when you generate the Revit components in that translation, you're relying on the IFC translation. And you can see in there, you don't have that many parameters within that connection family that you can manipulate once it's in that environment.

So you're relying on that workflow to update your changes. You can't really make those changes in the Revit environment. And moving that downstream to the detailer, you don't have as many options for the IFC exchange.

So just a quick movie to show you that process. So this information is in Rhino at the moment, [INAUDIBLE] line geometry and the size of the tubes. They're all parametrically driven within the Grasshopper scripts.

And this is when I'm talking about relationships between members. What we're able to do here is create a relationship between those two cap plates to say we need a minimum tolerance between those two caps. So this is something that we wouldn't be able to do within the Revit environment.

And then just pushing it through via Geometry Gym pushes those components into the Revit environment. But you can see on there, it's come up with a random family name. And the parameters in there are just generic.

So after doing it in Grasshopper, we said, well, why don't we just recreate that exact same script in Dynamo, which is exactly what we did, another spaghetti script in the background there. But this time, when you bring in that connection family, it's got all the parameters in there, because it's a Revit family. So you've got all the full functionality of Revit using the Dynamo workflow.

So the advantage there, obviously, is that you're already in the Revit environment. Everything is familiar to Revit users. You're having to learn Dynamo, which obviously is a skill that most organizations are getting into. But it isn't your core skill, so therefore, there's training requirements for that, too.

And again, this is just illustrating that we were trying to do the same thing by building in a parameter to maintain that tolerance between those two cap plates. So this is exactly the same workflow, but using Dynamo, rather than Grasshopper. You can see that the script is pretty similar, and just updating fully parametric solution to bring in those connections, with the benefit that it's obviously [? cutting ?] those members, as well, so they're independent. That would be how you would fabricate it.

So the final option was to do it in Grasshopper through to Tekla. So in exactly the same way as before, the curves and the tube diameters were inputs from the original Grasshopper file. We defined the planes. We defined the plates. And it was able to respond to that geometry.

So within Tekla, this is the fully working solution for both the bisector plates and the splice details. These are all fully integrated Tekla members, so that can be fully used downstream.

But the disadvantage to this is, how many guys in your office can use Tekla? The current team are already in Revit. Do we want to keep it in Revit? Do we want to move it to Tekla? We want to use the best tool for the job. So if this is going to be pushed downstream, and it makes our checking of the fabrication and shop model easier, then this may be the option.

So again, here we're just making all these adjustments on the fly. It's updating automatically. There's no manual input. Change the number of stiffness, thickness of the cap plates, size of the tubes.

So just reiterating the advantages and disadvantages on there, the advantages-- you've obviously got LOD350 out of the box. It might not be in your scope, but further downstream, if it's going to benefit you, then maybe that's a good idea. You obviously get that real-time update. Every time you're making a change, you're seeing the impact of that change.

You can link that to engineering design spreadsheets, so if the spreadsheets update, you can link that back in again. It's fully automated. And the other thing you can do is obviously link that to FE analysis. So you can, if you've got 20 of that similar connection on that job, and they're all slightly different, you can push that to FE analysis and optimize those plate thicknesses.

So the disadvantages-- obviously, currently, we're only required to do LOD300. Do we need to go to that level? And obviously, you've got to train people to use Tekla who are typically working in a Revit environment.

So as a summary of the findings of all of those different methods, there isn't a right and wrong answer. I think you've just got to look at each situation in hand. And if it's a single member which has only got relationships between itself and nothing else, then why not just keep it in Revit?

But as soon as you start looking at relationships between multiple elements coming in at a single point, that's when Revit struggles, so using those workflows for Grasshopper and Dynamo are far more powerful. And if you've got an architect who's continuously changing that geometry, then that workflow is obviously better. So yeah, the effort and reward, you've got to balance what's required to be issued versus how much time it actually takes to do that upfront work to respond to those changes.

This is a little slide that just summarizes all of that in terms of every workflow that we went through and the considerations for each of those. So this is where there's no right or wrong answer. This is all in the handout, guys, so you'll be able to get hold of this.

But the one thing that we didn't do-- and the reason we didn't do it in this study is because we knew that the shop detailer on this project was using Tekla-- we didn't go down the route of looking to do this in Advance Steel. So we've got that as an option for any other project. And I'll be keen to hear from the Advance Steel users as to whether or not you feel there's additional benefits to do it in Advance Steel over Tekla, or whether you see any benefits in doing it in Tekla over Advance Steel. That would be a good comparison.

So the recommendations would be one size doesn't fit all. You have to look at every single project individually, look at how many variations you have of that same connection type. If you've got the time upfront, then obviously, the Dynamo/Revit route is probably the best one to get to LOD300, because the rest of the documentation is already in Revit. If you want to look at going beyond LOD300 to LOD350, then potentially Grasshopper to Tekla or Advance Steel would be one of the better options there.

And moving forward, so I touched on this a little bit, we would like to look at that from an FE analysis point of view. Why not optimize all of your plates on that job? Let's not just rationalize to do one particular connection. If there's 20 variations, push it into FE, optimize that plate thickness. And obviously, at any point in time, because this has all been fully modeled, you can take quantity take-offs and get good feedback from your QS constantly about how much money you're saving by making these optimized changes.

And then I think the last thing I want to touch on is-- and this shouldn't be underestimated-- the stuff that we did there, we were pushing the boundaries of what we were trying to do. When we built the Grasshopper script and the Dynamo script, it didn't work the first time. It took a few goes. It's trial and error.

So if you are going to experiment with this stuff, make your leadership guys aware that we can do it, but it's not going to happen overnight. It takes time. And there's obviously a cost to that. Innovation doesn't come for free. You need to have that investment of time. So with that, I'll hand over to Xav.

XAVIER NUTTALL: Hello there, great to be here. So I'm going to tell you a quick story about a pedestrian bridge. This is the Lachlan's Line pedestrian bridge.

Now, there are kind of two fields that parametric work can fall into. You can simplify lots of your day-to-day tasks and speed those up, which is one kind of bucket of parametric tools. The other one, which is kind of where this project falls into, is doing stuff that wasn't really possible five years ago, at least not cost-efficient to do five years ago.

So this job is quite an old job for us. It started three and a half years ago. It's taken a long time, for a whole bunch of reasons I won't go into. We're now halfway through our detailed design.

And as a way of introduction, we'll start with the site. So we're about 10 kilometers north of Sydney. So Harbour Bridge, if you were to stretch that screen way down, you'd see the Harbour Bridge down here somewhere. We have, running east to west, which is from right to left, we have Delhi Road, seven-lane motorway. We've got the M2, which runs top to bottom. That's four-lane motorway.

So what we need to do-- in the top left-hand corner of that slide, you will see a little dirt patch where the A is. That's the old stabling yard, where they used to keep the tunnel boring machines for when they were building the train tunnels. And the train station is that three-pronged building down on that side. So when they were building the train station and all the train tunnels that go below it, they used this dirt patch up here for all site sheds and the stabling yards.

Now, our client for this bridge project, company called UrbanGrowth, they inherited that site, this piece up here. That site's about a mile long, about a third of a mile wide. You're obviously only seeing the bottom corner of it. And when they inherited it, they were asked to develop it into a big multi-use space. So there's 20, 30 buildings that are slowly getting built on that site at the moment.

Now, as part of their work, they needed to find a way of getting about 6,000 people from A to B within the next year and a half. So whilst we started three years ago, this has to be done and installed by 2018.

So over the next 10 minutes, I'm going to show you what we've learned in the last 10 years-- sorry, last three years, and introduce a few things which, whilst they're things that we've learned doing this relatively complicated geometry, should be things that are applicable to just about any job. It's a way of thinking. And that's more important.

When you're doing parametrics and that side of design, the clever part is actually having the idea to do it parametrically. Once you've had the idea, and you start thinking about how you could do it, can almost guarantee that you'll find a way to do it. There are hundreds of ways of doing things now. The clever part is actually asking the right question and getting the right people on board to allow you to do it.

So we'll start with our first question. For a pedestrian bridge, the alignment is the kind of change line. It's the center line of the deck, which defines all the levels and the gradients.

Now, architects are great at convincing people that they have the right idea. They use great words. They've got fancy images. Now, I'm an engineer. I don't have such great words and such great images. What I do have, though, is numbers.

So at this point in the job, we didn't have an architect. In Australia, the engineer is the lead consultant for bridge projects. And so it's quite normal to actually start a job without an architect, and even complete a job without an architect.

In this case, we started the job without the architect. And we defined the alignments. We defined it numerically. So what you can see, that orange zone there, that orange zone is the 5.5-meter vertical clearance that we needed to achieve over Delhi Road. So our bridge needed to sit above that zone.

Now, we also know that we want to create a cost-efficient bridge. So we want the shortest bridge we can possibly do. So that means using gradients which are the steepest we're allowed to the access code. And we knew the gradients. And we knew how frequently we would need landings. We knew the gradient of the landings.

So all of that allowed us to put together a little Grasshopper script. So this is work using-- so Grasshopper's a plug-in for Rhino that allows you to create computer code that manipulates the Rhino model.

So this is a visual that comes out of that script. What you can see is it shows the gradients of the bridge deck. Now, we had the actual topology in three dimensions. So what we did is we used a spline curve. So that curve is defined using seven points. And the Grasshopper script calculates where it would land, where the bridge would connect into the topology.

What that means is you can work out exactly how long the bridge needs to be. So in this example, you can see as you go over the road, because the road is relatively flat, the gradients are almost flat. As soon as you get away from the road, the gradients jump to red, which is the steepest part. And what you can see now is we've got seven points that are controlling the spline.

We get to client. We sat him down in front of our laptop, gave him the mouse with our seven points. He was dragging it around, adjusting his bridge, finding an alignment that he wanted.

Now, what you can't see in the right hand, we had a second screen of our client. And then there, there was feedback on total bridge length and feedback on maximum span length. So we knew that we were getting 50, 60, 70-meter spans. And we need a bridge that was somewhere between 180 and 200 meters. And with those two numbers, we as engineers could guide our client to the right kind of solution.

It was a really great way of getting a client involved at a really important stage in the job, because it allowed them to own their bridge, rather than us just giving them the alignment. That was a really quite simple little Grasshopper script. But it was great for the client to actually get involved and see what it could do.

So we have our alignment set. The next question we asked ourselves is, what do we do? How do we make a bridge out of this single line?

So we'd already spoken to UrbanGrowth. We knew they were interested in a tubular type structure. There are some nice examples up here, got the Singapore Helix. We've got the Peace Bridge by Calatrava in the middle. We've got a bridge in France by Dominique Perrault. All of these are basically giant trusses that you walk inside.

We knew that our client was keen. And at this point, we had then appointed as a subcontractor to us an architect called KI Studio, so our Sydney-based architects who are going to help us develop the story. Now, just like we sat the client down, this time, we wrote ourselves a little script that allowed us to manipulate the form-- this was done in Grasshopper-- manipulate the form of the bridge with the architect, so we could both get outcomes that we wanted.

So structurally speaking, we wanted a structure that was deep where we had our largest moment. So if you look on the top right-hand side, you can see there's the bending moment diagram. We have our largest bending moments over the continuous supports. So that's where we wanted the largest bridge form. So you can see the yellow dots on that blue shape are where the support points are. That's where we've broadened out the diameter of the tube.

Architecturally, they're interested in installing benches, [INAUDIBLE] screens, canopies in all of these areas. We made it big enough to accommodate all of those things at this early stage in the project. So that's how we set about defining the form.

Now, on top of that, we started to overlay a structure onto that shape. This was, once again, the workflow Rhino to Grasshopper. Reason we chose that workflow three and a half years ago is because Dynamo didn't exist. Now, I'll get to this later, because it's starting to cause us a few problems now we get to documentation.

But what you can see, going back to this structural system, it's a truss. So our truss has chords, and it has diagonals. I'll just point them out. These curves here, which are kind of spinning relatively gradually, they're the chord members of the structure. And the elements which you can see there, they're the diagonals. They're rotating much faster.

What we began to realize is there's an infinite number of potential geometries for this helical-type structure. So we've made a numbers slide, as you can see here the huge number of potential variations that we could have. And we didn't really know how to choose the right one.

But as a structural engineer, there was a few things on that video which are a little bit concerning. If you look at the columns, beautifully pink columns, they don't often connect to the helix. That was one of the major problems that we had at the beginning.

So what we realized is that we have two variables that we're playing with there. One is how quickly the chords rotate. And the other is how quickly the diagonals rotate.

There are actually more variables that we could play around with. The location of the columns, for example, isn't set in stone. We could move those columns 2 meters up or down the chainage point, if it meant that we could find ourselves a geometry for the helix that would work.

And when I say work, what I mean is if you look at these pink columns, they're like rabbit ears. The idea is that they support the helix at two points, two nodes. And we've got four chords. So if you cut a cross-section through any slice in that form, you're going to have a circle, because it's derived from a tube.

So if you've got four points, and you're supporting your circle using your columns at two sideways points, what it means is that the other two elements are at the top and bottom. Now, that means the chord members of your truss are in exactly the right position to be most effective at the cross-section where you have your highest moment forces. So we wanted a solution where we had chords at the very top and bottom of the cross-section, and nodes on the sides of the cross-section at the support points.

So we now have five variables. We have the rotation of the chords, the rotation of the diagonals. And we could fine-tune the location of the columns.

So using the plug-in called Galapagos-- so far, we've used Rhino and Grasshopper. Now, Grasshopper's a free download. Galapagos sits within Grasshopper. It's also free. It's an evolutionary solver.

So what it does, you feed it variables. In this case, these are the red number sliders down here. It will initially cast 60 random scatter points to get a feel for how the variables work. And then it will slowly, over time, home in on what it feels is the best solution.

Now, you define what that solution is. So you numerically tell it to either minimize or maximize a number. In this case, we measured the distance from the closest node to where the top of the column was. As that distance got smaller, we were obviously homing in on the correct geometry.

So here, we set it to run overnight. 5,300 runs later, we came back in the morning. It had found it. We were within 5 mil of the perfect geometry. That's pretty good for a night's work.

So that was using Galapagos. There are a lot of other evolutionary solvers out there. That this one just is native to Grasshopper works quite well.

So by now, we'd really sold the dream. The architects were busy making their beautiful pictures. The client loved the bridge. And it was already getting state significant development approval. It was really time for us to crunch some numbers.

So we had our center line model, which is what you can see in green on that left-hand side. That's the 3D shape of the helix structure. They're not tubes. They're box sections. They're warped and curved steel box sections, naturally complicate to fabricate, also quite complex to analyze.

So each member of that is a 450-by-250 RHS of varying wall thickness. So we needed a way of converting that geometry on the left-hand side, which is all within the Rhino and Grasshopper environment, into a relatively simple analysis model for us to do some structural work on. So we used a plug-in called Geometry Gym.

Geometry Gym's a great plug-in by Jon Mirtschin. What it does is we subdivided each one of these curve center lines into 300-mil segments. And it then applies structural properties to those straight elements. They have to be straight, because our analysis package can only handle straight lines, which is why we subdivided it into quite small components.

But using Grasshopper-- so we've now got three and a half thousand members in our analysis model. But using Grasshopper, we can actually map the face of the RHS so that it follows the twist of the member. So each one of these members on the right, which is in GSA, is orientated so that the 450 face of the RHS sits parallel with the face of the form. So as it steps around a curve, it follows that.

And what that means is we've now got a link from Grasshopper environment across into structural analysis. Now you can apply all your loads, your load cases. You can even run the analysis from within Grasshopper.

And where that leads to is-- I'm about to show you some videos of the analysis. So this is stage analysis image. So you can see, yes, here, this structure looks like it's breaking. It's not. It's just a result of the stage analysis.

But the important point is that we can now rapidly adjust even the alignment. We can go back to those original seven points. We could move one of those seven points. And we can hit Bake, which means transfer information from Grasshopper across to GSA. And we would have a brand new analysis model ready to go.

We would then hit Run. Honestly, it takes two minutes to make it. And you can easily compare the results from, let's say, alignment one and alignment two.

Now, you wouldn't have optimized all your structural members, so comparing tonnage isn't that useful. But what you can do, because you've got exactly the same structural properties in both models, you can very quickly say this system is stiffer, or this system has more force in compared to the other system. So you can start to home in on what is the best structural solution quite quickly by running 10, 15, 20 different alignment geometries, and even address the scale of the helix to understand where is the most effective place to put the steel.

So by now, we've got ourselves a nice stick model. So that's just using beams within our GSA analysis. As I said earlier, we're in detailed design. So we've moved on from that into a more complex environment. What we wanted to do, because we knew each box section was going to be fabricated, we had the opportunity to decide what plate thickness we wanted for each face of the helix and each face of each box section.

So each box is made of four sides. Because it's a truss, we have significant bending and axial forces within the members. That's because they're so curved. We get a lot of bending forces, which means that the stresses in the plates are definitely not symmetrical. So if you've got a big compression and a big bending, you've got the opportunity to make one side of the box much thicker and one side much thinner to reflect the stresses in the members.

So we've got about 300 elements, so let's say 300 of these curved beams to deal with. And we've got four sides of each of those, plus about 200 nodes, which have two primary faces. Prior to this work, we'd done quite a lot of work in Strand 7. And we were auto-meshing buildings as a quick way of understanding how buildings would behave. We could auto-mesh a building model in a day.

Now, in doing that, we realized that if we would make a Rhino model and bring across two surfaces on different layers into Strand, and we auto-mesh it, it creates two separate properties. Taking that logic to this system, we used a plug-in called EleFront. It's another free plug-in to Grasshopper. What it does is it takes every face of every member of the helix, and it bakes it to a separate layer in Rhino.

So now we've got over 2,000 layers in our Rhino structure, each one with a different color, which is what you can see here. What that means is when we bring that across into Strand, when we auto-mesh it in Strand, every side of every face of every helix has a different structural property. So here, that green will be slightly different structural property to that green, to that pink. So when we run the analysis, we can now look at, let's say, the peak stresses in plate property number 300.

And we can now use-- in Excel, We can loop through. And we can optimize. And it's a little bit iterative, take half a day. But you will run through and manually iterate through and make sure the results make sense.

But you can strip out all of the steel tonnage that you don't need out of that system. Now, this is-- the steel tonnage, the live load is relatively high. We've got five KPA live load. But the steel weight is relatively significant. So actually, stripping out the weight does then affect your results, which is why you have to iterate a few times.

What that allows us to do is to optimize the steel plate tonnage for every side of every helix. Now, because the fabrication is quite complex, actually reducing the plate thickness from 12 to 10 or 8 mil does make a big difference in the ability for them to roll it and warp the plate. So there's quite a lot of savings to be made, even in a cost per ton, in going from a 12-mil to an 8-mil plate.

Here are the results of the FE analysis. So one important point is this last piece here. Definitely go back and check what this system is giving you. It probably makes sense to simplify some of the results.

So where you've got four sides of an RHS, it might make sense for the two short faces, the two webs, to be the same thickness, just to simplify the node connection. There's a few manual processes that we have to do, just to make sure that the results, when it comes into drawings, are still logical.

And the final question, now we're talking about drawings, is how do we get the geometry from our geometry hub, which is Rhino, how do we get that across into Revit, because we wanted to document the bridge within Revit? Now, I wasn't convinced that Revit could even handle this geometry. There's not a single straight member in it, for a start. But I was very happily surprised to be proved wrong.

What you can see on the left-hand side here, this is our Rhino model. These are solid objects within Rhino. Once again, we've-- Jon Mirtschin from Geometry Gym came in. And he showed us more tools within his plug-in. So we actually had these tools available. We just didn't quite know how they worked.

What it does is we subdivide the helix into its component parts, so the nodes and the members that span between the nodes. It creates an IFC file of each component. Revit can then read the IFC file. And it creates the geometry you can see on the right-hand side. So now we've got an automated link that not only goes through the alignment, but it goes through concept analysis, through the detailed analysis, all the way through to documentation.

Now, at the beginning, I said that we were using Rhino and Grasshopper, and we were starting to struggle. I'll show you on the next slide why. So on the left-hand side, we've got our geometry hub. That was built-- well, started three years ago. We're now at the end of our structural documentation.

What we're finding is that because it's all driven on this left-hand side, getting all the member scheduling to work comfortably and quickly within Revit is actually quite difficult. So we are considering right now redoing this left-hand side entirely within Revit and Dynamo. And we think it's possible. We don't see why it couldn't be done.

What it would mean is that this last piece here, getting all of our scheduling and our plate thicknesses and our member sizes, and also the setout across into Revit, it's going to be a lot easier, because we are contractually responsible for the setout in Australia, as well as the structural properties. So we're looking at that at the moment. That's as of a few weeks ago, seeing what-- really pushing what Dynamo can do, to see if it can get to this kind of geometry.

So one of the key things, other than the workflows, that I've learned in the last three years is-- we never set out to do a parametric project or a BIM project. We never had that intention. We just used tools to find a way to do what we wanted. And Grasshopper happened to be a good one.

We never set out to use GSA or Geometry Gym or Strand. They were just the right tools at the time for the job that we wanted. So we would have never figured out that workflow unless we knew how each one of those tools worked.

So I always get asked, what's the one program you should learn? The problem with that question is really, you need to learn lots of programs to be able to join the dots. There's a nice quote here from Steve Jobs, which kind of explains that. "You can't connect the dots looking forward; you can only connect them looking backwards."

So we would have never been able to create that workflow if I only knew one program. So really, when people ask that question, you need a broad experience and a good understanding of what all of the programs can do before you can start making the efficiencies of really developing a good workflow. So there it is. And I will hand back to Graham.

GRAHAM ALDWINCKLE: Thank you. I'm Graham Aldwinckle. I'm from the UK office of Arup in London. And similar to Matt, I'm a structural BIM leader for the UK, Middle East, and Africa for us.

So I'm going to talk to you about some of the other aspects that we do in Arup, and also in the UK. And that's really picking up on the things that we've presented today. So we want to get across to people in our organization, and you guys, that it's about the mindset, the skill set, and the tool set.

And by that, we're wanting everyone to get into the mindset of being creative and trying to use the right tools for the right job. And the skill sets and tool sets are all very-- there's a lot of overlap in that. They go hand in hand, because you have to provide training to get your users, your teams, to be trained in the right tools. And you have to know which are those right tools in the first place.

So it's very much following on from what my colleagues have said. Because there are so many tools available, which one is the right tool? And we've presented that in one of Matt's slides as an example of the workflow.

So what it boils down to is us as structural engineers, or as engineering teams or technicians, have a multitude of tools at our disposal. And they do what are presented on the screen here, a multitude of things. We have tools that deal with calculations or do model reviews or whatever it might be.

And so we have a number of tools at our disposal, Excel being a key traditional calculation tool. But more and more these days, those calculations are not done in Excel or hand calcs. They're done in the tools that we use, whether it be Revit doing some analysis in its results manager. You've got some ability to do the rebar design in Revit, for example, now.

We want to link as much of this as possible. So it's all about, in our eyes, it's all about the interoperability. And there are many facets of interoperability, as I'm sure you're aware. But it's got to be the key to successful collaboration on projects, both within your engineering teams, and also externally. So you'll hear a lot about that this week, I'm sure.

So interoperability using Dynamo or Flux, even-- we're pushing Flux now quite heavily, which is a tool that links a number of packages, Rhino, Revit, analysis programs, in real time. And Flux is really powerful, although change control is another matter with Flux.

But it's not just about the interoperability. It's about the data. Content is king. So structured data is what it means to us, so the I of BIM, of course. And I'm going to present a few of the examples of some of the content that we use in our Revit templates, because we find that by standardizing data as much as possible, we can push so many useful functionality into Revit that we get huge benefit in time savings.

So I'm going to present a couple of examples here. So we have some internal tools that help develop 2D structural details. We have different tools across UK or Australia or America, depending on the market. And we have the ability to load up these standard content that we just put on our drawings.

So there's an example there. We can create a drawing pretty simply, very quickly. It's a standard detail. It applies at the beginning of a project to make sure your QS is understanding what is needed. They're often replaced, of course, later in the project stage by real detailing out of the live Revit model. But we need to provide these early on.

Also, we're able to generate a typical notes drawing very readily, very easily. And we have standard ways of doing that in our Australasia region or UK region or American region, et cetera.

So it's a spreadsheet to start with, which has a number of fields in it, which are pre-populated by our engineering teams. So it's a case of excluding the ones you don't want to add in. And when you've said, for example, here, there might be two number twos. And you each just choose which of the two is relevant to your project. And you write exclude.

It then creates that text file. And then that gets pumped into Revit. So that's just one example of how we standardize our things for our teams to make sure that the during production and deliverables are as straightforward and simple as possible.

I'm now going to go into some of the rebar content in the checking and temp-- Revit template that we have. Now, last year, at the class that Matt and I gave, in fact, here, it was about our Revit templates and some of the content and standard views that you see on the left. [INAUDIBLE] some of the structural views that we have in our template, all relying on structured data.

So for example, where you've populated the steel grade or the concrete reinforcement content or whatever, we have automated-- we have checking views that are standardized and that instantly show that data on the screen. So you've got an instant feedback loop for the engineers. But I'm going to concentrate here, in the next couple of minutes, on the rebar content that we have.

So we have a number of parameters that we've embedded in our template. And this is some of them on the left here. So those dimensions, for example, are the reinforcement dimensions that go along with the shape codes that we use. So those shape codes are embedded into the Revit families that we use with conditional formatting.

So for example, taking a particular type of bar-- where've they gone? There we go. Taking a particular loop of a bar, we have conditional formatting that tells us whether the bar size that's been modeled complies with the shape code. How that works, then, is that when we've modeled the reinforcement, we have an instant feedback loop for the engineering teams.

So yes, the reinforcement can come from the various tools that we have, the [INAUDIBLE], the [INAUDIBLE] plug-ins that help generate the reinforcement [INAUDIBLE] as well, of course. But where it's complex, we need to add in our own rebar. This allows us instantly to marry up the schedule that we produce with the rebar that's actually modeled. So here, for example, this schedule that's live, the color red is because it doesn't comply with the shape code. So by having that structured data and putting the data in the right place, we're able to instantly show feedback for all those views.

And here are some more examples of how that works. We've got views that automatically show you the steel grade. So it's a way of making sure that that data is consistent. And when we're transferring data, we're doing that more and more. More of our projects are able to get that data handed across via IFC or whatever. This is a way of checking it.

This is just a couple of examples of the reinforcement content that we're getting into. We don't do it on every job. And in fact, many jobs don't require 3D reinforcement and documentation. Design intent is more often than not all that's needed. So this is overkill on a great number of projects, in fact. But where we need to provide it, we can.

So when we have data, we can harvest that data. And so the next part of my talk is about the data harvesting that we do to try and leverage all that amazing amount of knowledge that we have. And as part of that, we have a harvesting team. And they gave us this snapshot the other day. This was back in May.

So a day in the life of Revit-- we had 823 users in almost 640 Revit files. That ebbs and flows. It goes up and down. It just gives you a flavor of how much, across our business, we're using the tool.

And also, because we're capturing this knowledge, if there's a problem on a project, or a problem such as the Revit file has ballooned in size, we can interrogate the data that's coming out of our snapshots and figure out what's, perhaps, caused that, because we might see that the number of families has suddenly exploded, or whatever it might be. We're capturing that data.

The data harvesting can be done in our firm manually or automatically. So we have a tool that allows us to manually harvest it. So you set it up once, and it-- or do it once, and it's set up.

But we also look to do it overnight. So we manually set it to run overnight. And it is also able to publish a few other things in our overnight batch publishing.

So here, for example, we're able to publish Navisworks, DWFs, IFCs, and PDFs automatically overnight. So that's a hugely valuable tool for us, so that we come in the morning, and we've already got a set of the latest work-in-progress drawings. And we can mark that up. We can provide comments to architects, or whatever it might be. So that automation saves a huge amount of time every day.

And that data harvesting captures anything and everything from the Revit model. So here's a little video of just one particular structural model. And it also allows us to check that our content is consistent.

So you can provide-- there are a number of ways of inputting a concrete beam. It might be width times depth, or it might be the other way around, depth times width. And we just need to make sure that our teams are doing this consistently.

Now, the sort of the things that we want to be using this for-- here's an example. So we might want to give our teams knowledge of what's happening on the other thousand projects that are similar. So by harvesting that data and then presenting it back to our teams, we're able to give an idea of-- get an idea of consistency across the other project types. And that's really powerful.

Another thing we do as a global organization is look at how we can help each other and put a bit of investment of our money into our teams and get some good results out of it, which gets shared. So Arup is a privately owned organization. We put all of our profit back into our people and training and investment.

So maybe 10% of our profits every year goes back into investment and new things to do, innovative things to do. So we don't have shareholders to pay. So that means that we have the ability to channel money where we want it to and develop things.

So Matt had an initiative which was looking at Dynamo scripting for everyday structures, so trying to do-- bring the knowledge that we've gained on these fancy bridges, et cetera, and put that into what we can do on every project. And as an example of how we share this knowledge around the world, we have comments on that initiative, including from me, but from people around the Arup organization, just to showcase how the power of getting that knowledge from the teams around the world draws out all that work that someone might have been doing in Los Angeles, or wherever it might be, and bringing that into one place.

And here are some of those examples. And it's in the handout, obviously, this list. So these are some of the things that we're trying to push to make using Dynamo-- to make our project lives easier.

And here's an example script. This is simply to replace-- it's a very basic script. But it's amazing how many times you might actually need to do this across various projects. And it's looking at replacing a drawing number with another number in a sequence.

So say your system has-- you architects imposed something on you, and you've got to run through a thousand drawings, or whatever it might be. It's a very quick way with Dynamo, for those that haven't used it yet, to really push the power of automation across your teams.

Now, once we've collected scripts, we now need to share them and make sure that we've got the best breed, if you like. And so we have a tools register that we're creating, which allows us to harness those scripts, whether they're Dynamo, Grasshopper, even Excel VBA scripts. And so the tools register collates that. And we have, at the moment, a snapshot of life into it, where we can search by any field and look for a particular script that someone might have written in another office, and see if it's been done before internally, and then share that, or download that and use it.

So that draws to a conclusion our talk today. So this is the standard slide that we have to show for you. I'm just going to leave you on this last slide here, some of the Arup work that we do, to give you more of a flavor. It's buildings. It's bridges. It's infrastructures, so a whole gamut of work.

So with that, we'll open up to questions. But one quick thing is that I have a freebie for you, which is one of those Google cardboards with Arup branding. So the best question gets it. So it's to encourage questions. We also have our business cards on the table, if that helps.

So just grab a microphone, and we'll start with some questions. Let me just grab a microphone. Thanks, just test it works.

RAY PURVIS: Hello.

GRAHAM ALDWINCKLE: Yeah, thanks.

RAY PURVIS: Hi, Graham. Ray Purvis, Atkins.

GRAHAM Hi.

ALDWINCKLE:

RAY PURVIS: So we know each other. Quick question-- really good presentation. I really enjoyed it. Lots of different technologies going on here. What was your approach to developing staff in the skills in order for them to use these products?

GRAHAM ALDWINCKLE: Good question. We probably have a flavor across our regions. So in the UK, we try and make everyone buy into the belief that doing things digitally is the right thing to do. And so we give them a huge amount of time, if they want it, to go and learn these new products. So we have training days. They can go and learn Dynamo. We get the people in, the experts, if we need to.

We do find that time is a problem. So getting people to get off their day job and go and learn these things is problematic. So by giving them the inspiration of what can be done, that's probably the most powerful aspect to get them to want to learn.

And we do grad surveys every year. And it's surprising how many want to push the boundaries and get on with this stuff. So we need to harness that. And instead of giving them load rundowns for the first year, which is what might have happened traditionally, we need to get them into these tools and pushing the boundaries. Do you have any [INAUDIBLE]?

MATT WASH: Yeah, I'll just add to that. We don't have as much money in the Australian region as these guys do in the UK. So what we do is we identify a champion in each office.

So we've got Melbourne, Sydney, Brisbane, Perth, Adelaide. And we'll identify a champion for each product. And that will be the guy or girl that will go to the next part of-- Revit 2017's out. What's the new features? And then they'll run a lunchtime session for all the users.

So we do it by identifying one person who is the expert in that particular product. And then they relay that knowledge back. We can't send everybody on every course. So that's the way we do it. We identify a champion in each office.

RAY PURVIS: I mean, that's interesting. And it's the same for us at Atkins. We're a global organization. And it's how we educate our people.

And I think we learn differently now. We go and get information, don't we? We go and absorb it. We don't sit and wait to receive it. And there's some really good products out there that can provide training through online tutorials, short YouTube-style tutorials. And it's a really good way of learning.

GRAHAM ALDWINCKLE: Yes. We're looking to roll out Pinnacle from Eagle Point. And they're here this week, just as an aside.

MATT WASH: Just on a personal note, from a Dynamo point of view, what I tend to do is identify a problem I want to solve. I'll then Google it. There will generally be a script that's similar to what I want to do, download the script, understand the way that that's written, understand what I need to do to change it, and then just tweak it. I don't typically go through every single tutorial from Lynda or Dynamo Primer. But I might relate back to them if I find a part of the script that I can't understand.

GRAHAM ALDWINCKLE: So there's another question at the front.

AUDIENCE: Yeah, it's a basic one. So in your office, is it mostly structural engineers that are doing this type of development internally? Or is it project designers who are more technical, but not engineers, but technical degreed? What does that make up? Who leads the way and actually is doing the development?

XAVIER NUTTALL: Yeah, I'll jump in there. So we're all engineers. [INAUDIBLE] work on [INAUDIBLE] jobs. Now, we do have a broad range of [INAUDIBLE]. So we have [INAUDIBLE] engineers, structural engineers, [INAUDIBLE] engineers. They're all [INAUDIBLE] different fields doing their own thing.

Obviously today, we've done a very structural talk. But these talks, I think they're happening across the field. And it's always driven by the engineering team. [INAUDIBLE]. We use them as we deal with things. And [INAUDIBLE].

GRAHAM ALDWINCKLE: Not sure if we're hearing that so much. Make sure your speaker's on, sorry. Did everyone hear that at the back? Not sure. Yes, you did. Good. Did that answer your question?

AUDIENCE: Yeah.

GRAHAM ALDWINCKLE: Another question?

AUDIENCE: Yes. I was curious if you could tell us anything more about the evolutionary solver you used, if you've seen the use growing in the company, if you've had successes with the bridge project specifically, I believe.

XAVIER NUTTALL: Yep. Hopefully you can hear me now. Happy to tell you a bit more about that one.

GRAHAM ALDWINCKLE: No, it's not working either.

XAVIER NUTTALL: [INAUDIBLE]. So in our-- [INAUDIBLE] it works off a whole host of variables. I wouldn't recommend putting in more than eight in there, gets a bit confused. It does need Grasshopper. So once you've defined the variables within Grasshopper, the important part is defining a quite simple numerical solution.

So in the case of the bridge, we were saying when the distance between where the [INAUDIBLE] is on the helix and where the point of the column connection is, where that's the smallest, that's when you're going to have the best results. What it will do is it will initially, for our whole load of random combinations of your variables, it will then look up which one was most effective. It will take the top 10% and move on to the next evolution. So that' why it's called an evolutionary solver.

[INAUDIBLE] through that. And it does have 50, 60 evolutions, each one with about 60 different analysis runs, and slowly comes in on the right solution.

AUDIENCE: What was the name of that product, again?

XAVIER NUTTALL: So it's called Galapagos.

AUDIENCE: Galapagos.

XAVIER NUTTALL: There's another one called goat, which is quite good, as well. And Octopus.

AUDIENCE: [INAUDIBLE] , nothing for [INAUDIBLE]?

XAVIER NUTTALL: So far, I've heard rumors of one coming in for Dynamo, too.

GRAHAM ALDWINCKLE: Does that answer your question? If anyone else knows of one for Dynamo, please shout and share it.

AUDIENCE: So this is sort of an operations question. What we've seen is that there's a real versioning issue with these open-source softwares, like Dynamo, and also not just versioning, but packages. And so your team probably has access to all of these. But then, when we start to push it out to the engineers, they may not have the right package downloaded. They may be working with a different version of it. How do you solve that problem?

GRAHAM ALDWINCKLE: It's an ongoing problem, isn't it? But we do try and push out and roll out the necessary updates to our teams, much like Microsoft would. So we do standardize where possible, but it's not always possible. And Matt, you probably want to add something.

MATT WASH: I was just going to add, this slide highlights-- we had the exact same problem. We were sharing scripts around, but you wouldn't know which packages you needed, which version of Dynamo you needed to run it. So this helps with that.

So now, every time we do send a script that we want to share, we make sure-- this is just a simple text file that says who created the script, which version it works with, what it requires, and the purpose. So it's not a perfect solution, but at least you know the basis of how the script works and what packages you need.

AUDIENCE: So my next question is, the engineers, when they get this, how do they know where to go to find these packages--

MATT WASH: OK, yep.

AUDIENCE: Is it something that your IT department pushes out?

MATT WASH: We've got something called Arup Shopping. So you go arupshopping/shopping on the internet. You get it, and you can search for any of the programs that have been approved. So you can just download it onto your own machine. Certain products you can't do, if there's license agreements. But all the open-source stuff, you just go to the internet and download.

GRAHAM ALDWINCKLE: And what we now are using Dynamo for is very similar to what we have been using and still use Grasshopper for. So we have a huge number of people that are familiar with Grasshopper. And now with Dynamo, we're now needing to make sure that we we're pushing out the right tool for the right project. So we're trying to adapt our Grasshopper scripts into

Dynamo, as well. There's another question at the front.

AUDIENCE: Thank you. Just real quick-- so I think we all know that Arup does very complex and large, famous buildings. But most of us do the regular two, four-story buildings, whatever. When your engineers begin their workflow for a regular building, let's say, is there a design check-- is there an Arup way that your engineers are taught to, like you were saying, like go search for the Dynamo scripts or the-- is that a natural workflow, or is that something you have to encourage?

MATT WASH: Well, what we do, we formally have to do an inception review on every single job. And regionally, I get involved in every single one of those jobs, to make sure that the best tools are being used and that the right team is put on that job. It's not always perfect, because the availability of staff mean that you can't always use the right product or the right people. But at the start of every job, we'll analyze how we're going to deliver that.

And that's through from how we're going to analyze it, how we're going to push the information into documentation, how we go from documentation to fabrication, construction, operation. Some jobs, you'll only need to do a certain part of that. Some jobs are fully BIM through to FM. It just varies from job to job. But the workflow and the process is the same, whether it's a complicated bridge or a car park. We just go through that same process.

GRAHAM ALDWINCKLE: And we have internal forums, where we encourage people to just pose a question, how would you do this? And Matt did exactly that on the connection we showed at the beginning, just to get a flavor of what other Arup offices would use. But that also can be used for the good jobs, regular projects, as well. So it's definitely a tool that-- encourage that to take up.

AUDIENCE: OK, great.

GRAHAM ALDWINCKLE: And this talk was about pushing BIM to the limits, so we didn't talk about those projects. But we do have them. Any other questions?

I'm conscious of time, as well. So if anyone needs to run, because we're now at 9:00, then please feel free. I won't hold it against you. And of course, thank you for not going to Marcello's talk.

AUDIENCE: Hi, guys, great presentation. I had a couple of questions. One's very quick. What was the link used for Grasshopper to Tekla for the parametric?

MATT WASH: So it's the in-built Grasshopper canvas parametric tools.

AUDIENCE: Oh, OK. That was quick.

GRAHAM ALDWINCKLE: Which is relatively recent, as well. I think it's only within the last six months that they've brought that out.

MATT WASH: I think Kevin Lee is here in the audience. So if you do have any questions on that-- or if I'm telling lies, Kevin--

KEVIN LEE: No, that's correct.

MATT WASH: Thanks.

GRAHAM ALDWINCKLE: And of course, we have some Advance Steel experts in the room, as well. If you don't put your hands up, people can talk to them. Thanks, Ralph and Stephanie. Go on.

AUDIENCE: The other one was just on if you had any thoughts on converting the Grasshopper files directly to Dynamo, right through.

MATT WASH: Without rebuilding them, you mean?

AUDIENCE: Yeah.

MATT WASH: Yeah, Flux was the one option that we considered. We haven't taken it any further than the consideration. But essentially, the data's the same. The variables are the same. The constraints are the same.

GRAHAM ALDWINCKLE: And to give you an example of Flux's use, actually, on the Sagrada Familia in Barcelona, huge basilica, we've been having a team of engineers helping finish the towers on that. And they've been linking their Grasshopper scripts via Flux, so that we can have multiple Grasshopper scripts for the various towers, rather than just the one script, which is easier to manage when you have multiple scripts. And so that's an example of what we use Flux for.

AUDIENCE: Thanks.

GRAHAM ALDWINCKLE: Any other questions? Feel free. I think that's it, then. Great. Shall we wrap up?

Thank you very much for coming along today. I hope that has been helpful. Please take a card. We're happy to answer questions during the week we're here. So thank you again.

[APPLAUSE]