DIETER VERMEULEN: Hi. Good afternoon, and welcome on this Construction Dynam(o)ite Class. I see, I'm really happy we have a full class almost. A few seats left, so text your friends to join us because we are really going to do lots of exciting stuff in here. So no time for siesta. I hope you had a light lunch, and not like heavy lunch. So we will really go fast today. I will try to keep my voice working for these next 90 minutes. I didn't say any words today to save it for this afternoon, but if you hear, it didn't work well.

Well, we have some small problem with the projector. Apparently it doesn't accept the resolution of this computer. It's not that special computer. So I'm really sorry that you all lose some small parts of it. They will probably jump in the session and fix it on the fly. Yeah, we have to start so they don't run over time that much.

So what are we going to do in this class today? Well, we don't need to read text. You can read it yourself afterwards as well. We are going to do this. Crane, clash, optimization, fractal, cranes, cranes, another time cranes. So I will [INAUDIBLE] you up with cranes, tower cranes, and at the end of the session-- or you will go home and go straight to that crane, or you will say, I cannot see cranes anymore. So we'll be really doing some awesome stuff with that.

A little word about me. So my name is Dieter Vermeulen Not Dieter. Some people call me Dieter, and then it's like no, I'm not going on a diet today. I'm a technical specialist AEC at Autodesk, and working mainly in the Northern European region. So which means Scandinavia, Benelux, and UK, and Ireland. And in my team, I'm responsible for everything around competition and design and engineer. So I'm a structural engineer by trade. So I try to use that experience as well, in all kinds of workflows with Dynamo I present.

So yesterday some of you were attending my class on Dynamo for Rebar. So that's my biotope let's call this, but construction is very close to structure. So first I want to explain you a little bit-- some words about generative design and computational design, and then at the end-- well not at the end, at the real start of the session we are going to have a deep dive in Dynamo for two practical use cases. More about that in a few seconds.

So first, what is generative design? We heard already about it in the keynotes yesterday. We heard about it in the product innovation keynote from MR this morning as well, and in the old way, the way probably how many of you are working now, we have one human with one computer, and you get a limited design. You create something. Right?

And this could be a chair. You could have multiple chairs that you want to design, but you're always thinking already on the outcome of it. It's like I want to have that outcome, and I'm working towards that outcome without having the possibility to think out of the box, and create more options. So for instance, for this chair, this would mean that we would have like a chair with 250 pounds weight for the supports, and 15 pounds for the chair itself. It gives you a cost of $30 for instance.

OK. We could save some material. We can save cost, we could save weight, by for instance, applying generative design tools. In this case, it's something that's been done with Project Dreamcatcher. And all these several design options have been generated within the product, and gave the user, the designer, the possibility to design thousands, or to create thousands of options in the same time as a designer would use to just create one.

So in fact, we still keep that one human. So there is that one human, and we add some AI, artificial intelligence algorithms on top of it. We use some cloud computing on top of it, and then you get hundreds, you get thousands of design options on that. That's the new way. That's how we are working now. That's the way how we make things now. Or should do now.

And these are already some examples of some of our customers applying this technology. I'm not going to explain all of them, but you have seen already the one in the bottom right with that car. It's in the exhibit hall. It's a practical application of generative design. On the top left, we see what happens at Airbus when these dividing panels in the Airbus airplanes, how they did it. So they used generative design to make a lighter element.

So if you go to the octo trailer, in the hall where you went for the registration, there is that big trailer over there. That's the team that is working, that's doing research on generative design and on Project Dreamcatcher. So jump in on there and talk to those guys. They will explain you much more than just these four things.

Now what is computational design then? Computational design is like a part of generative design. There are multiple definitions on computational design when you look on Wikipedia, and I didn't want to just project like a note from Wikipedia. We all can go on our computer or mobile phone and find out ourselves. So let's do it differently.

We could say that computational design exists already for many years, for ages even. We have parametric design. That's part of computational design. Could be an ingredient, you could say it like that. It's an ingredient of computational design. It's where we make relationships between variables, and get a new result, and that relationship is for instance, a function. So in terms of geometry, you could get this. Subtract two geometries from each other and you get a new geometry.

Or scripting, that's a more advanced way. This is really looking scary. It's using Python inside of Python script, inside of your revenue model to create a very complex shape. Or there's sketching. This one is the oldest method that we know already. In the Renaissance they used lots of these kinds of other methods. They used sketching to make relationships between elements. So that's somehow parametric design. Of course, it's not like very responsive on changes, for instance, but it makes relationships. So it was some kind of intelligent design they used for doing that.

And then there is that visual programming. People hear programming and they want to kick out or get out of the room. So I asked to lock the door on this slide because people are scared of programming. Well, visual programming is not the how we think about programming. We don't have to be a PhD for doing that.

On the top left, we see an example on how kids are using visual programming tool. Well, it's more like a game where they put all these things together, all these logical operators together until the fish goes blup. On the bottom right, we see more professional version of that. And in fact, that shows us that visual programming is more like a flowchart, and you connect all kinds of commands and functions with each other to get that flowchart running.

This is computational design. It's not a parametric design. Now, don't call it-- please, if you bump up on me in the hallway over here, don't say Dieter, that was good about that parametric design. Then I will call this guy to come after you. So it's computational design. So what does this mean for you as an architect, or engineer, or contractor? It's all about making things better. Making it better, making it more efficient, using all kinds of techniques to achieve that goal in a quicker way.

So what does this mean? All these things, all this theory I just talked to you about for this presentation. Well first, we are going to do this case. This is the use case I'm going to explain to you. So part of it will be with videos, part of it will be with live demo. And the first one is-- I didn't find the right words to describe it, so I took the name Live Design Clash Verification.

We are going to do some design edits in a Revit model, a structural model, and in the other office there is one working in a Revit MEP model. And that engineer caused a clash due to that. We are going to use Dynamo to stream these two models with each other with Navisworks running in the background. So Dynamo will ask information at Navisworks, and it will stream the results back into my Revit model, and give me live feedback on the design clashes.

And in the second use case, we are going to explain on how to use Revit Dynamo Studio and Project Fractal to do some site planning. So in this case it will be about crane optimisation. Have a look on what is the effect of the crane position towards the building, and will be have problems, for instance, for lifting all the elements. So we are going to do some by diagnoses on that Revit model for that.

OK, the first thing, Live Design Clash Verification. We all know that clash detection, it's very important to have that. So these are all kinds of things we found in this hotel. No, that's not true. That's a joke. But it's very important that we know about this, right? So in a Live Design Clash Verification, what we can do in here is set up our models in Revit. So the structural model and the MEP model, then create a coordination model from it.

So we are going to combine these two files in a Navisworks model, set up already some clash detection rules. Not doing the clash detection, just make the rules already, and then also get the clash results with a package called DynaWorks, authored by Adam Sheather. I don't know, is he here? He promised me to come. No, he's not here. And afterwards we are going to evaluate and modify that design. So the evaluation will happen in terms of clash indicators, those yellow arrows you see in there, and we will create isolated fuse on top of that.

So the first step. What is the first step in here? Well, adding a project parameter to all of our objects so that we have a single parameter that we can refer to inside of a Revit model to do some diagnostics. So in this case, I use a project parameter called clash, and I make a parameter of the yes no type, so that we have some kind of a check. We can switch it on, switch it off.

OK. Secondly, you need to create those diagnostic fuse with view filters. So what I mean with the diagnostics is that when the Revit model is checked, then I want to see visually what is happening in there. So a red element is an element with a problem. So to have that, we are going to filter the whole revenue model, look on the parameter clash, is it yes? Then the objects should turn red. So that's the view filter. I'm not going to demo that, so I hope you know how that works. Or it is in the handout, also described on how to do that. So we have so many things to talk about today, and I cannot show that.

Additionally, we also need a family, a clash indicator family. This is that yellow arrow we see over there, add some model text maybe next to it so that we get some data information in our view immediately with this. So it will show us for instance, the clash name, and it will also show us the clash result name. So the clash testing that you've defined inside of your Navisworks coordination model will pop up over there after we have run the Dynamo script.

We can optionally also create some clash schedules. You don't have to do that, but that can be very interesting. Have it in your Revit model. Get a list of all the clash indicators, and their unique ID I give to them. Maybe get more information on which objects are really clashing in here, and maybe do some review on it. And imagine if we stream this to BIM 360 Docs-- I didn't even thought about it for this class, but if you stream that information to Docs, you could link that whole thing together with each other.

Another option is link that MEP model inside of Revit, just to have a visual effect on it. Just to see how is that Revit model affecting my structural model, but it's not needed for doing this clash verification because the two elements, the two models actually are binded inside of the coordination model in Navisworks. I should drink a bit.

Then this part is defining the clash rules in the coordination models. So once you've appended the two Revit models inside of your coordination model in Navisworks, You set up some clash rules, some clash tests. For instance, I want to have a test where the walls and the ducts are compared with each other. Have a look on do they clash, what is the clash tolerance maybe. So we can set it up all over there. This is something you cannot do in Revit. You cannot set a clash tolerance. While in Navisworks, you have all kinds of additional rules that can be set on it.

So once this is done, we can start. And I told you Dynamo is a flowchart, or you can compare it with the flowchart. It isn't a flowchart completely, we can compare it. So what we are going to do in here is first we have the Revit building components, which are all the elements that are in the structural model, all the elements that are in the MEP model. So secondly, we have the Navisworks coordination model.

So in Dynamo we are going to read the Revit model, we are going to read the Navisworks model, and then through Dynamo, we are going to run the clash tests, get the results from the clashes, and then actually with that clash, we need to do some information. Or we need to do some additional action. So if the clash is no, then you can take the left part. If the clash is set to yes, then it can take the right part.

There are many more seats up front also in here. So if John wants to re-enter, you tell him that he can sit over you. So clash assets in Revit, do we have them? Yes. Well, then delete a clash few if the clash is solved. So the clash assets are actually isolated few I'm going to make, and also the arrow that needs to be positioned on top of that clash. So these two things are assets.

So if the clash is solved, then we need to delete all these things. If the clash is not solved, or if it's a new clash, then-- no, not if it's a new clash. If the clash is not solved, then just do nothing. Leave those clash elements over there. And at the right side, we see what happens if we get a new clash. So set a clash to one, that clash parameter. Visualize all of this with the view filters, and then create your clash assets inside of the Revit model, which is that isolated few again, and then also that clash indicator.

So this whole workflow, that's something you have to consider. You have to think about it before you start putting all the nodes together in Dynamo. First think about what actually do I want to have at the end, and how do I get to that end. So I need to do four operations. Before I start showing you how this is working, make some advertising for Adam, has great work with DynaWroks.

So that's the Dynamo library for Navisworks actually, that you can do. I'm only using a small part of it even, so you can do much more with that package than just get the clash results and stream it to Revit. And this is how it will look like. At the left, we see the original model. At the right, we see what happens if we run Dynamo on top of it, and we see the isolated fuse, and those clash indicators elements turning up in red. Any other thing you want to do with it, you can add it to that rule set.

So we want these indicators, that's first. That's one thing. Each indicator will also have parameters. So it will contain comments, which are didn't are the Navisworks comments. It will have a mark, which is actually the number. So it will be the unique IDs, the Revit IDs of the structural model element that is clashing with the MEP model element that is clashing with it. So these are the two Revit IDs combined with each other into a new one, and then store it into the [INAUDIBLE] value. So that would be helpful for tracking your elements afterwards, if you want to do some more feedback actions on that.

OK let's go live. This is always the most exciting part. Is it working? Yes, and it's on the screen. It's not fully on the screen. We can easily change that of course, by just minimizing the windows. That should work. Well yeah, you will have most of the information in there. What about Dynamo? Dynamo is over here. OK, good.

So, you see structural model and the MEP model. So they are both set in a coordination view, but I also have a structure view. So this view here doesn't contain that MEP model. It's not visualized in there, so it's not needed. But of course, you want to see what is happening in that MEP model if someone is designing it. OK, now let's open up the--

In the mean time while it's opening , all these data sets that I'm showing you right now, they are all available on the website. So where you registered for this class, you can find the data set, you can find this presentation, and you can also find the handout which explains all of this. So some of you might have received an email from me. Not everybody didn't allow to receive e-mails from us on that. So that's why I'm telling you this again.

OK, so this is how the Dynamo script looks like. It's quite a huge script. There are lots of spaghetti is in there. So I'm going to try to teach you a bit, like in almost-- let's take 20 minutes for that to explain all the several parts of it. OK? But I'm not going to dive in every note and tell you why we use that note, and what is this doing. I'm just going to talk to you about the concept of this whole thing. The more detailed explanation is all in the handout. So I don't know how many pages I wrote for us, maybe like 120 or something. So it's a long read that takes some time to read all of this. OK.

So, you also will see that the scripts has specific groups. So there is a group for inputting. So these are the grey groups. There is a group for data processing. There is a group for the Navisworks data, and there is a group for Revit actions. So this is already identifying what will happen in that part of the Dynamo script. So some elements will just do some data processing inside of Dynamo, and then all the blue ones are all the ones that will do creation things in Revit, or store data inside of a Revit element.

So the input is actually that Navisworks coordination model. Let's have a look on that Navisworks coordination model. How does it look like right now at this moment, after I did append that Revit structure and that Revit MEP model to it, and when I created those clash tests. So that's the one. You see, this path of this file. So data sets demo one, Live Design Clash Verification, and so on. So that's in fact, this file.

Why do I show you this? Well you will see something happening in that folder when I run Dynamo. It will store files over there, and it will do all kinds of crazy saving. So see to date, the last time I used this was on the 8th of November. So that will change to today. I don't know what day we are. I know we are Wednesday, but I don't know the date. So anyway-- So, here we go.

These are the clash tests that are already defined. As you can see, it's all set to zero. So nothing has done yet. So this is just a plain Navisworks file. Nothing has been built up here. So let's close it. This is the same file. If you do this at home, if you use that data set, be sure that you also create the right link to your file, or keep it in the same folder structure as you have received it. But in case you didn't, you have to set it to the right one in here.

And then the clash check input. This is all the elements in Revit that I want to check. So I want to check the floor, as I want to check the walls, the columns, and also the beams. And this is where we put the indicators in. So we have those arrows, and I want the model, or I want a script to check if the clash indicators are already placed yes or no, and if not, which type of clash indicator should I use? Maybe you want one with text. Maybe you have some without text. It depends on what you want to do.

OK, and then some settings for the clash indicator. So I want to set it on top of the clash. You can give it to an offset so that it's appearing a little bit above the clash, instead of right on the clash, and all kinds of things. And also a unit's multiplier, that's a very important one because my Revit project is in millimeters, and the DynaWorks package reads the information in meter. That's just because of the API. It also works for imperial units, but I'm of European so I don't know anything about these imperial units, and I don't understand the logic of it.

So the second part-- and this is the whole core. Only four nodes, and these are the most important nodes of this whole process. And the most important one is this one, the list dot create. And why is it important? Well, I want to follow a specific order, and the operation of all the older nodes. I want that Dynamo first saves the Revit model. No, not first saving the Revit model. Excuse me. It first has to collect all the elements, and secondly, it has to collect the clash indicators.

Three, it has to save the Revit model, and then four, then at that moment it can open up the

Navisworks file. I don't want it to open up Navisworks, then save the Revit file because then Navisworks won't have the latest model. And if you do it in this way, with that list dot create, you can track that order. It's something we call getting dirty nodes, and when does a node gets dirty? That's at the moment that there is a design change in Revit.

So for instance, I change my wall. I make a new opening in that wall, and then do nothing. Just go to Dynamo and run, then the wall nodes over here, this one, is getting dirty. So everything that is behind that element is also dirty, and it will force Dynamo to execute all the things all over again. So that's very good. That's what I want.

If you don't do that, it will not be re-execute. So it will this one is dirty, then this one is dirty, and it goes on, and goes on, and goes on, and then this one is also getting dirty, which means that the list dot create will effect all the other ones that are connected with it, and it will rerun the whole thing in a nicely and proper way.

So after it has collected the elements, as that transaction is done, then I'm telling to Dynamo to save the Revit documents. That's a custom node with some [INAUDIBLE] scripting behind it, taking care of that Revit action to say save them all. All right? If it's saved, you get some warning in here. OK, the document has saved. So you know when the document has done its- - or if the node has done its work.

And then after that operation has been done, then we can open up item three, which is opening up that Navisworks model. Or not opening it, just referring to it, pointing to that Navisworks model on the Windows Explorer. And when all of this is done, go to the open Navis file. So it will then open up the coordination model, which we can use to do all these clash detections afterwards. OK? So that's a very important process that we have to do in here before we can start with all the rest.

So let's do that. Let's run this first part. So the first run takes a bit of time because it has to opened up now Navisworks, it has to read all the Revit elements, and bring all these Revit elements inside of Dynamo. And as you can see, loading structural model. So that's Navisworks running in the background now.

So it's updating its coordination model. It's taking the last version of that Revit file. Which one is the last version? I told you that something would happen in the Windows Explorer. As you can see, structural mobile has been changed to 16th of November, And it has saved that backup file, structural model dot 0001. So that way you know what has happened with that Dynamo script.

If I run it again, nothing will happen because no nodes got dirty. I didn't do any design change in my Revit model, so it won't save it, and it won't even open up Navisworks. So nothing will happen. So we have to force it and to make it dirty, which is making design edits. All right? So not needed for this moment yet because we first need to find out if there are clashes in that Revit Model. So as you can see, nothing has happened in the Revit model. Nothing yet, so we are still in the orange zone, which means that we have subtracted information from Navisworks.

So let's have a deeper look on that. So after the model has been opened, we want to run the clash detection. So run all the tests. Go into here, give me feedback on these tests. So you remember I had a test called wall versus duct, wall versus pipe and so on. When this is done, when all the Navisworks clash tests has been performed, then save the Navisworks file. As I told you, this one-- look at this. The Navisworks model has been saved.

If you open up the Navisworks model now, then you will see that it also has done the clash detection. So in here we have we will find already more information than we will see in Dynamo. So as you can see at the right, the clash detection has run. When I just opened it five minutes ago, everything was set to zero. Now Dynamo has forced Navisworks, and it told to Navisworks, OK, come on. Run all these clashes, and we have that information in there.

So we know how many clashes we have. We can see the clash results in here, but what I want to do now is I don't want to find out in the Navisworks model which Revit element is clashing, and finding out its ID, and then go back to Revit and try to find that element in the model. I could use the Navisworks switchback, but actually, I want to have an overview of all clashes. Not with the switchback, I want to have a real visual overview in Revit.

So the next part that we are going to do in Dynamo is extract all that information that is stored in the clash detection. So all these clash results, we are going to extract that info inside of Dynamo, and do new actions based on that. So we come in here, get the clash results. These are the results that we just saw. So maybe you remember, the first clash test had one clash, then the second one had three clashes, and the third one has one clash too. This is what you are seeing over there. I should take my point for that so that you can follow me a little bit easier.

So you see over there, clash and so on. This is the internal Navisworks name we have in there for that. So what do we want to do with these clashes? First, I want to create a filter. This filter here, this whole structure that has been set up, is going to have a look on the status of a clash because it could happen that a clash is resolved. It could happen that I said in the Navisworks model, or the Build Coordinator in the Navisworks model, has said OK, this is a clash. Theoretically it's a clash, but we are going to solve it on site. So just say, approved. So you can change that status in your Navisworks model.

So in that case, we have resolved and approved, then I want to filter them out. I don't want to do any actions on these clashes. The clash indicators should disappear, and the clash view also has to disappear. So that's why we are going to create a filter for that, and at the end of this loop we get this.

These five clashes that we have found. Which one is equal to resolve or to approved? Well, it's set to fall, so none. They are all real new clashes, or active clashes. OK? And that's also what this guy is telling us, if you go over here. New, new, new. new, new. It could have been approved or resolved. So if we run the script the second time, we will see that this one will say active, new, and resolved in all of these things.

All right? So once we have that filter, or at least this is a Boolean filter, how we call it. If you add all the clash results to this filter [INAUDIBLE], it will create a list with all the resolved ones, and it will create a list with the new and active ones. So there we go. What is the second part? It's about creating clash names. We have the clash names inside of the Navisworks model, and we want to use that information inside of Revit to track it.

So we need the clash tests name, and we need to have the clash result name, and we want to store it in the parameters of that class indicator, and it will also be used to create the views. So this way I will have unique view names inside of my Revit model for those isolated views. So that is what is happening in here. It's just combining these strings with each other, and at the end you get-- come on, zoom in-- you get this. These are the names.

Again, you see on top, it's an empty list. This one is the filter list because this is old and new clashes. Later on this will be all the resolved clashes. And all these resolved clashes will say to Revit, OK now delete that view, and also delete that clash indicator because it's resolved. All right? That's for our second run. .

What it also has done already is it's created a clash indicator ID. So it said IDF, it should be

IDs, so with an s instead. So these are the IDs of the first element clashing. This is the ID of the second element clashing. That has been detected already, also with DynaWorks nodes in this whole loop. So this loop here show us that-- take the clash detection, take the results, take its nodes, take the name, and then at the end take an attribute called elements ID. So we have to dig a little bit into that structure of Navisworks.

For instance, if you go inside of Navisworks and ask the properties of the elements, you can find these attributes as well. So we can ask many more attributes, different attributes, then just the element ID. So you could also ask something about the volume, or the weight, or whatever you want to have. All right, so this is done for the first model, and this is done for the second model. So structure against MEP. All right? There we go.

So, what do we have at this moment? We just finished this part of the script. So this part, the blue one, is set to freeze. So I didn't execute it yet. So I will immediately unfreeze it, and run it again, and then you will see the result in Revit. So at this point we have the information here about the clashes. So the clashes are filtered. We have information over there to create a unique name, and we have information in here to create a unique ID for the clash inside of Revit.

And now we are going to create-- we are going to visualize the clash elements by setting the clash parameter to yes or no, depending on its status. And then we are also going to create the clash indicators, and create the clash views. So first, the IDs. So take all Revit elements and also the clash indicators over there, and the clash views, it's over here.

I'll explain to you immediately how this works. Let's first run it. Done. That's the time it took for creating all of this inside of my Revit model. So what we have in here is the clash indicator, so that's the arrow. The Navisworks model-- I could get to the clash point, let's say, from the Navisworks model in Dynamo, and that clash point is now used as the placement point for these arrows.

OK, so what has changed in my whole model now is this arrow. We got all the information like the clash test name, and also the clash result name. We got this mark value. So this is the first element. We also have these walls, and the beam up front in here that are colored, and that have different color due to the view filters, and this is because this element-- the clash is set to yes. So if I can manually switch it off, of course. So I can trick around it, but yeah, you have to be conscience with that of course.

And then we also have the separated views. So these are the clash views. So you go to these views, and you can do your design edits from in there and run the script again. So that's what happens, and that's part of the Dynamo script. So let's have a deeper look in Dynamo. How did we achieve that?

So, here we have all our Revit elements. 31 Revit elements and the structural model. Walls, floors, beams, and columns. Take the ones over here and compare their ID with the clash list we had over here. So here we have those Revit IDs that were returned by Navisworks. So compare all the elements with that, and then find the ones that are clashing, and find the ones that are-- no, sorry. The objects that are not clashing, and then find the ones that are clashing. This one is set to zero, which means clash equals no. This is set to one, which means clash equals yes.

So that's the easy one. This is a more difficult one, the clash indicators. You would say, OK, just place them. That's it. You don't have to do that much of an effort for that. Well actually, it is because I don't want to just place it. I also want to delete them if the clash is resolved, and that's difficult because you have to track them. Revit needs to know which-- or Dynamo at least, needs to know which element should be deleted when the clash is solved.

So first-- something happens with my mouse. Sorry. First, you have to go into the clash indicators, flatten the whole list, and so on, and so on, and so on, and delete the element if the clash is gone. So that's also a custom node in here. Well, it's not a custom node, it's a Python node. So if you go into it, I'm not going to explain you Python.

Actually, I'm really lousy at Python. I just go on the forum, on the Dynamo forum, find this, and then someone gives it to me. And it's like, OK, thank you. And put it into your Python script, and like look, I'm a programmer. I can do Python. But don't ask me about it because I will give you a very lousy answer on it, OK?

So there were no elements to deleted, exactly, because there were no clash indicators yet. So we will have them at the end. OK, and add clash indicators. Add the clash point. So the DynaWorks package uses this thing-- I should zoom in a little bit more so that you can see it. The nodes are named like clash detection dot get clash point, which is the imperial one, and you also have m get clash point. That's the metric version. That's just the reference of that DynaWorks package to the API of Navisworks.

So as I told you, you have imperial and metric. So if you are working with metric, which you all should do in my opinion, then take this version of the nodes. OK? So we get the points, we get five bullets because we have five clashes. Take these points with the family instance by point creation node. So this actually will create a family at that point with that coordinates. And as we see here, the family type is connected up front with these clash indicators I just created. So this clash indicator families. In this case, clash indicator with text.

And then besides that, I don't only want to place them in the Revit model, I also want to provide them with additional data. So I want to add some metadata to it, and this is the clash test name, the clash result name, the mark, and the comments if it finds any comments. So this is what's happening in here. Creating all the parameters together, creating their values together, and apply it on all the families that have been placed in the model. That's what this step has been doing.

So as we see arrow with text, these are the five objects that have been placed in the model. And what is cool about this is that if you go in that watch list in here, and let's go to the full 3D model, like this. If I click this ID, it zooms in immediately on that clash indicator. So it helps you to find out your way inside of your model. So click on this, and then we get this node. On this, it zooms out to that proper element.

OK, and then the final part is creating clash views in Revit. So clash views based on their clash names, and based on the clash test result-- sorry, on the clash test name, and on the clash result name. So we combine these two values into, for instance, wall versus duct clash one, wall versus duct clash two. This is always creating a unique name. That's very important when you create views.

As you know in Revit, if you create a view, if you duplicate the view, it cannot have the same name as an existing view. So you have to make sure the Dynamo creates unique names because it's automated. You don't have any control over it, and this is how you can create those unique names. Just combine strings with each other that are unique by definition in Navisworks already.

And then once we have these views, or these view names, we filter them out with that Boolean filter we just created to say is the clash active, new, resolved, or accepted. So we use that filter, that mask, that Boolean mask on any of these filters each time again and again. So it will take the ones that are actually clashing in this output, and then it will create a 3D cropped view.

So this is another package I'm using for it. It's called Steam Nodes. So this package makes it possible to create 3D views, but it also creates them with a section box. So you get a cropped 3D view. And the section box in here will use the bounding of the points. So it's not taking the Revit element, it's just taking that clash point, and it will create the bounding box of 1,500 by 1,500 around that the clash point.

So not around the wall, because if you have a very long wall-- like for instance, an architectural wall, that one single wall of 50 meters long and 20 levels high, which we normally should split up. But if you have a clash, then you don't see it because it's so big, and if you take it to that point then it doesn't take into account the geometry of your Revit object. So you'll always have exactly the same views, which is really nice if you want to document it. You can drag all these views on sheets then, and they're all the same size. All right.

So this is how it creates, and it also deletes it when it finds clashes that were resolved already. So then we have the lead fuse again, of [INAUDIBLE] and nodes deleting this whole thing. All right. So, let's go do some changes in here. OK, this one is an easy one to change. Go to a wall opening. Should be good.

This is a bit straightforward. Of course you will do it more in detail, but we just want to make sure it's in. And don't save it. It's not needed. You can save it, but it's not needed. So what has happened now in the Dynamo script, whole up front in my script. This must be really hard for you if you had a really tough night yesterday evening, and then I'm moving the screen all the time. It's like, oh, stop it.

Now this node got dirty. The walls got dirty, so it will rerun the whole script node. OK, let's do that. Run, forest run. This is the most exciting moment now because I want to be sure it works. Sometimes of course, like any software, if you demo it, it crashes. So let's hope we don't crash in here. So it's loading structure. It loaded the MEP model. It was looking for clashes. This window is always staying up on this, so we kind of ignore it. So let's have a look. There we go. Clash indicator is gone. The wall is not read anymore, and the view for it is also gone.

What if we go now, to the Navisworks model? What has happened over there? Let's first have a look on our structure in here. We see that there is a new save, so the coordination model has been saved, and the structural model has been saved as well. So if you go now to Navisworks, go in there. Then we will see that one clash got the status of resolved.

And this is also displayed in here. While it's opening, I can show you some more stuff in here, in the Dynamo script. You see the first one? It's set to resolved now. So it triggers a whole new, let's call it-- you know what, I don't know how they say it in English. It's also Domino? Those domino blocks you put beside it. So it has launched a new track in the domino design we had in here, because there are now, in my filter, one of the objects got to the status true. So that Boolean mask that we just have created is now making all kinds of other elements active. So we have in this one-- no, not in this one. In here we get the wall versus duct clash one, which is the view name. This is [INAUDIBLE] in that top list, so this means that this view should be deleted. Yeah, and that has been done at the end.

In here we got to that specific view with that view name, and we say to Revit, OK, delete views, and it gives us feedback at the end. Yes, you have deleted one view from the Revit model. OK, so in the meantime, the Navisworks model got opened up, and as we see in here, we still have our five clashes, but one of them is not active anymore. One of them is resolved, you see.

One is resolved. Now, there is one of the clashes in here, beam versus pipe, or maybe wall versus pipe. That is the one over there, you see. It's clashing over there. Let's go to its results, and say for instance, it's clash number three. Let's set it to-- yeah, we can keep it to active, this one. No, sorry, we will say approved.

And this one, the clash number two, we will keep it to active, but we will add a command to it. OK? Add command. There we go. Now we have to save it. So save this one. Close it. Run the script. Why does the script run now? Because the Revit model got dirty because I have added these clash indicators in the previous run. So then it's always dirty. Every time you run it, one of the notes will get dirty. So this triggers your script to run all the time.

So imagine that we integrate this with the Dynamo player, for instance. Then the user doesn't have to open this up because this might scare people in your office, because they are not used to working with Dynamo, and maybe you are the one building this whole script. But if you add this to Dynamo player, it's just like click, click, click, and so on, and you have your live design clash verification.

All right. What happened now in the model? As you can see, the arrow over here disappeared because we have set the status to approved. The wall is still red because that other pipe is clashing with it, but if we go in here, what should we do? OK, here create a round opening. The comment has been stored inside that clash indicator now.

So if I go to my clash indicators in here, I get feedback from the Navisworks guys who told me that the first clash indicator-- or the one in the list over there, create a round opening here. So what you can do then is open up a view, or click your clash indicator. Highlight and model. OK, there we have to make a round the opening. So that's another view that you get on that to create or to edit your clashes in here. All right?

OK, Breit. Not [INAUDIBLE] Breit.

[CHUCKLES]

Let's go back here. Similar things can be done with architecture. So it's not just about structure, but as I told you, I'm a structural engineer. So I tend to do it more in that way, but it could be done with a window as well. Right? There is also some other workflows that can be done on it. Instead of creating that opening manually, you could also use Dynamo to create the opening automatically. So use the clash point, and then create an opening around that clash. So with voice, for instance. So that's also a possible workflow that could be done on top of that.

Now, second case, the Crane Positioning Optimisation. When we do site planning, it's very important that we do it in the right way. You don't want to have cranes that are in enclosed in your building of course. You don't want to position them, and then suddenly notice that some elements cannot not be reached or cannot not be lifted.

Of course, every contractor is doing the right ways on that to make the right analysis for the position of that crane, but it's a very long process to find the ideal position of that crane. You have to do some calculations, manual calculations maybe, and that can take some time to achieve that.

Why should we optimize it? Well first, define the optimal number of cranes, and find the ideal location of that crane. Also something, avoids conflicts between those cranes. We don't want them to clash with each other. Detect possible hoisting problems. More important, save time and costs for that. And the project I'm going to use in here is actually one of the Revit sample files you have. If you open up files, you always have these sample files, and they're like structural model, architectural model, and so on.

So I use the structural model for it, and I have to say special thanks also to Jesper from

MTHojgaard. I don't know, is anyone from MTHojgaard in the room? OK so, Jesper gave me the idea in fact. He came up one day, oh Dieter, I want to do some crane positioning. What is your idea on that, and could we introduce maybe some genetic optimization to achieve that? I told him, OK yeah, give me your ID. I will make a genetic optimization routine on top of it, and then showcase it at AU. So thanks for that again, Jesper, for bringing up the ID.

So first, the workflow that we are going to do is one, you need to create a model. Of course, that's. A very obvious step. Secondly, we need to initialize the supply and delivery point. What is that? Well the supply point is the truck. So there is that yellow element over there, that yellow truck is where we are going to place the truck delivering all the pre-cost elements, and then delivery point is the position of the crane.

So we just set it somewhere up, a position where we think that it could be the best position for this whole project. Then we are going to read all the results, or all the elements at least, back into Dynamo, and creates an analysis of it and do some evaluation of the results inside of Revit, again, by using view filters.

OK, half an hour. I have to go fast forward in here. So again, a flowchart for that. A flowchart which is a little bit more complex than the one we had for the clashes. So we have the building components. There is a capacity table in Excel, telling us that for that specific range, the crane can lift that amount of weight. And then we also have our truck element. So we have the building components, the truck, the crane, and it's capacity table.

So we are going to define first, the distance between the element and the crane. On an xy-plane, we are going to take that distance, and we are going to compare it with the distances that are set up in the table. So that are set up in the lift capacity table. So the final element weight, that's also something that is very obvious. We are going to take the solid geometry of that object in my Revit model and calculate the weight of it. So if you have distance, and if you have weight, then you can compare it completely with that capacity table.

At the end, it will give us four possible status or states. So what if the distance of the element to the crane is bigger than the maximum crane range? Well, in that case we have a problem. We have a unreachable element. If the weight is bigger, so as you see here, is the distance smaller? OK, then you can continue. Then calculate the weight. Is the weight less than the crate load depending on its specific range? Yeah, then OK, you can continue. If not, then you can say it's reachable, but it's not liftable.

If you continue in here, then we are going to compare the weight with the maximum loads for the truck position, because it's not just about delivering data points in your model, it's also about supplying it. So you have the crane at the left-- sorry, the truck at the left, and the delivery point at the right. So this distance between the crane and the truck is also important because you have to take that element and then bring it and put it into the building.

So if that distance is not OK, then we also get a status for that, and this status is called liftable, but there is a truck issue. So just drive the truck a little bit closer to the crane, and problem is solved. If none of these problems occur, then we have a green element, which is called liftable. So the goal in this whole thing is to have all the elements in green. OK? You're with me on that?

How do we evaluate that? Well first, we visualize them, these elements, inside of our Revit model, and secondly, we give it a score because if you do optimization, you need to have a tangible value, a measurable value. So it has to give a score, and the score calculated like this. So we have a bunch of elements which are liftable, which are in the yellow status, the orange status, and the red status, and we give them a score. So there are 600 elements which are green.

The status score is zero because that's perfect. We want to have the lowest score. So if the lift score, which is a total of all of these values, equals zero, then we have the perfect solution. In this case, the score is 5475. There is no units for that. That's just a value. So you can choose in here which value you want to add a score for a specific element.

So if an element is liftable, but there is an issue with the truck, that's not a big problem on site. You can just drive the truck closer to that crane and problem is solved. So we give a score of five for that element. So it's not going to weigh that much into my total lift score. 20 Is the score for the ones that are really giving a problem because they are too far away, or they are too heavy for that crane. So that's a more important score.

So you see, these 20 elements give an influence of the score 400 on the total, while these 15 elements only give 75 as an issue, and the unreachable ones, that's really dangerous. That's red alarm. Bing, bing, bing, so there's a problem in here. So on that one, we give a lift score of 100, and we get to that 5,000 value.

So what we are going to do now is Dynamo is going to drive that model that way, that it finds this score for all the elements. OK? It uses also a package for it. So you might have seen it. I published a package called BIM4Struc.CraneAnalysis two weeks ago. So you need to install that package before you can run with this whole thing. And the reason for it is it contains the whole calculation in that custom node. Otherwise, your script becomes really, really, really big, and with these custom nodes, you can make it smaller and readable. OK? Plus it's ready for genetic optimization, which we will see immediately.

So we are going to do four approaches. I really have to go a bit fast, so we might run a bit over time due to those issues we had with the projector. So we are going to do a single situation calculation. Going to do a parametric run, which means that we are going to feed the whole thing with a bunch of parameters, and then find out the ideal one.

Then we are going to tell Dynamo to just do it itself. Do some genetic optimization. Go with it. So I'm not going to feed it with parameters anymore, or with values anymore. We are going to use a package called optimal to do that, and then at the end we are also going to do some crane layout exploration by using fractal, project fractal.

First, the single situation calculation. So it will just take the situation as is in the Revit model, and tell me is it OK? Yes or no. So no optimization on that, and if you want to do optimization, that's very important to do that because you need to figure out if your analysis will work in just a single situation. So that's the first step you have to do. So let's have a look.

I'm going to go really quick on this part because the other ones are really exciting, and I definitely want to show you before the class ends. I don't want to hold you up too much if we run over time. So again, it's going to use this color coding, and it's always the same. Input, initialization, analyze, evaluation. So we could make a mantra from it if you want to do some exciting thing, and make some music behind it, whatever.

So the input is, of course, the most important one. It's the part where we are going to detect the elements, the parts, the structural framing in this case, and the structural columns. So it's very important that you have the right elements. If you have floors and walls, split them up with parts So don't take the whole floor of course, because it will be an element which is not liftable. Right? So we used parts in here.

There is some user input telling us the weight of the wall elements, the weight of the floor elements, because I don't want to just take the volume of the elements and-- no, sorry. I could store this inside of my Revit element, but I decided to put it into Dynamo because it can change every time. This weight is the combination of the concrete weight, and also the reinforcement that is inside that element. So it's not just the concrete on it's own. For a floor, if it's a hollow core, then of course you cannot use the weight of a massive concrete floor. So that's just an indicator. You can choose how to do it if you want.

And then secondly, we need to select the crane. Select a crane, that's the one. Select a truck. That's the one. All right, and let's run it. It takes some time. So don't click too much on the window. Oh, it doesn't work. Click, click, click, click, and doing a bit nervous on that because Revit will say not responding, and then it crashes. So just leave it alone like this, and it works. There it goes. Open up Excel, reads the capacity table, stores all the results inside of Dynamo, and let's have a look what Dynamo tells us about this whole thing.

Nothing has happened yet in my Revit model because parts of the script are still frozen. So it just has detected all the elements. It has detected all the solids of it. So this model has 1,181 objects. OK? It has calculated the weight of each element. So just take the volume and multiply it by the weight. We need to know the crane calculation point. So take actually the insertion point of that Revit family, and detect its coordinates. Do the same with the truck, and this way we can measure the distance, because there is a very cool node inside of Dynamo which is called geometry dot distance, and it calculates the shortest distance between two points. OK? So that's the one we are using in there.

OK, and this is the whole core of that crane analysis. So that's the custom node that is delivered with the BIM4Struc.CraneAnalysis package, and in there you can find the whole calculation. I'm just going to double click on it, that you can see a little bit more of insight on that, but I'm not going to explain this. It's explained in the handout, and this is the calculation engine that is driving that flowchart I just showed you, with unreachable, non-liftable, liftable, but truck issue, and so on, and so on. So this is how it's been calculated for that.

So if we unfreeze that node, and run, it will take all the building components, it will take their weights, it will take their calculation points, and the lift status, and the crane arranges, the crane loads, and so on. So the crane range is the distance. The crane load is its corresponding capacity, so its load capacity, and then it's also taking the calculation point of the truck, and there is a specific one called lift status values.

So the lift status value is just a reference for yourself to indicate, OK, once all the elements are grouped into their category of liftable, non-liftable, unreachable, then I want to add a value to that. So I want to tell Revit this is value zero, or this is value one, and this is used then for calculating your lift score. OK? So this is what these values are doing.

So what we have in here is now a sorted list-- no, it's not a sorted list. It's an assigned list. So what we have there is the first family gets lifts status three. The second family gets lift status three. So what has it done now in Revit? Well afterwards, it will-- it's not yet done by Dynamo. We have, again, a project parameter. You cannot see it, sorry.

We've got a project parameter over here, called lift status. So this will be filled up with these values zero, one, two, and three. Each of them is corresponding with unreachable, liftable, and so on, and so on. You could also use text and say OK, let's fill in unreachable, but I like to use numbers for that. That's better because you can calculate more. And then a view filter it applied together with the view template to color the model immediately.

So I just have created a list. This family should have that parameter equal to three. So that's the next step in the Dynamo script in here. There we go. Let's unfreeze it and run them. So what it has done right now, it has calculated this course. So this construction, this setup has a lift score of 42,580, which is a really high score. Which elements has now influenced that score?

Well, let's have a look a little bit up front in here, and what we see in here is that this is the score for liftable but truck issue. The score for non-liftable and the score for unreachable is really high. Actually, we have 712 elements that are liftable, but 54 are liftable and have an issue with the-- are non-liftable, sorry.

Let me rephrase. 712 are liftable, zero are liftable but with a truck issue, so that's good. But 54 are non-liftable because they are too heavy or they are out of range of the crane, and there are 450 elements that not even can be served. They are unreachable. So that's a really bad situation of that crane we have over there. So let's visualize it now in Revit, by assigning these lifts status values to their specific parameter. That parameter name is lift status, as I just showed you in the model, and run it.

And this is what happens now in the Revit model. These are those more than 400 elements that are unreachable by that crane. So if we position now, the crane in a different place, like for instance, let's put it over there. Go back to Dynamo. Run it. So again, the script got dirty because the crane has changed. So it will rerun up front the whole script. So that means it's also going to read all the elements back into Dynamo. So that's the reason why it takes some time. It has read 1,181 objects and create it's solids inside of the Dynamo interface.

So let's wait for a few seconds on that. That's the reason why movies are always better to demo. You can cut that wasting time. And there it goes. No red elements anymore. A few yellow elements, but what I have done now, this is a really obvious situation. You could notice already upfront, that position would be the ideal one for the crane because we have that circle which is stored inside the Revit family of that crane. But still, there are some elements that are on yellow. So what we could do now is instead of doing this manual operation, why not do some optimization. And a first step for that is using methods, and I call it parametric run.

Last year I did a class in here also, called Dynam(o)ite design for engineers, where we did structural optimization. So this whole process is also explained in that class. So if you go on the AU online library and type in my name, you will find that the class as well, and you can find more information. What is a situation calculation? What is parametric run, and what is genetic optimization?

So I'm going to talk a little bit faster in this part. So what we are going to do in that part of the script is instead of putting that crane, manually, we are going to tell Dynamo that OK, the crane should be positioned there, there, there, there, there, on all these green dots. Evaluate the design and return all these results back inside Dynamo. So all these green elements are showing the result of that.

So this is how it's done. So again, as I showed you, we have the lift status. In this case, it's set to nine. That's just effective number to trick bit around that view filter. I want the view filter not to show it in red or green, and so on. And if you open up that script, which is in the data sets, the parametric run, you will see it's practically the same, but not completely. So there's a few changes.

One change that we have in there is that we are not calculating the truck position anymore, based on its current position, on its single situation inside of the Revit model. No, we are going to take that truck and the crane, and we are going-- we only take the truck. We are going to take the crane and see in here, we are going to create samples.

So what I want to have is 10 objects in the x-direction, 10 objects and y-direction. So 100 positions, actually. So it will create 100 points on that surface. It will also detect which points are not on that boundary surface. That's the surface where the crane can be positioned on. So we don't want a crane inside the building, so these points should be ignored. So that will ignore them. So that's the part where it happens, what happens in here.

So we still have how many? 88 several positions. So there are 12 positions-- these are the ones that should be here then-- that are not taken into consideration. So actually, it will run the same script I just did 88 times, which is too much. I don't want 88 times because I know that this point will be the baddest one, and I know that these points also will be bad ones. Right? So that's what we are going to do a bit later in the next step, when we are going to introduce the genetic optimization.

So once these 88 situations have been analyzed, they feed back that information, that really huge data flow through that lift score calculator. So that lift score calculator gives us for every situation, a lift score. So the top one is 1,860. This one is 1,000 and so on. The more down we go, the bigger the lift score value, and at a certain point, one will be the smallest one.

So it starts with a quite big one, 1,860, and then suddenly it goes down to 1,000. So probably that 1,000 situation, lift score 1,000, will be the ideal situation for that thing. So it's not the ideal one with a value of 0, but it's the ideal one for the ones that I just have asked to be analyzed. So that's this point over there.

Now what you can do then afterwards, is tell the script that it should take that situation with the smallest lift score value, and feed that back to Revit and reposition the tower crane to that point. So that is happening in here. So the crane was initially somewhere here in the middle, and this script has calculated all situations, asked for feedback-- what is the ideal position? And repositioned that crane to that point. And as you can see, it has X 21, Y minus 2. These are these values that were returned back by Dynamo.

And an additional step that you could take on that is by-- come in guys. Don't be scared. So additionally, what you could do also is for every situation make a capture of it. So make a screen shot. So there are some nodes inside of Dynamo which makes it possible to make PNG files of your Revit fuse, add them in Windows Movie Maker to each other, and then you get these awesome views on that. So it gives you some possibilities to review all the design options.

What's next? Well, we could also use a package called Mandrill from a Konrad Sulbon, and Mandrill is a package that makes it possible to analyze your data, and to visualize it, for instance, in these parametric coordinate, or parallel coordinates systems. So this is helping you to find the ideal solution. You remember, position number one had a lift score equal to 1,000. So that should be 21 meter, and X was 21 meter, and Y was minus 2 meter. So if you look all these data, if we evaluate it in the parallel coordinates system, we are going to see position X 21, position Y 0.

So, still have 10 minutes. Yeah. The videos are at this moment not yet available in the PowerPoint because it's a PDF that we have to upload on the site, but I will make all these videos available on YouTube. If you go on my block, called Revit Beyond BIM, you will find all these videos afterwards. OK? So I'm going to skip this video, but actually it's just taking all the analysis and it makes that parallel coordinate system. OK. Next.

What about if we would solve the problem with two cranes? This is exactly the same situation, but just taking two cranes, and do that analysis on both cranes together, and combine their results into one single result, which is telling us if the cranes are in that configuration, good or not good. So again, it will calculate the minimal lift score, but it will take the lift score for the first crane, it will take the lift score for the second crane, and then combine them into the worst situation.

Now the third step is genetic optimization. Genetic optimization in fact, is an application of one of the laws of Darwin. The natural selection, or the survival of the fittest. Actually, it's about having a population, and the population in this case is crane positions, and the situation, the shape, is called the phenotype.

The parameters that we add to it-- so like the weight and the calculation points, and so on, are called the genotypes, and then we have the analysis function, which is calculating their distances, and which is making the lift score that is called the fitness . function. So these are the three pillars of genetic optimization, in fact. And there is a package called optimal, which is used for doing that, and actually, that optimal is creating this.

It's creating an evolutionary population, so it makes all these several situations of that crane, and it takes two of them. One is called the dad, and one is called mommy. They fall in love, they make babies doing some crossover and mutation, and you get a child solution, which is in fact the baby situation of that crane. It's going to be brought back into the-- you cannot see to the left part in hear of the PowerPoint, due to problems in here.

So it's going to insert it back into the evolution, and it will make a new situation. So actually, we are going to create generations. That's what's happening with us as humans too. In the beginning we were running like this, and now we are doing this, and when we get old, we do this again. So that's also what's happening in here, but it's doing multiple generations. So we will have like eight generations, or 10, or 20 generations, until we find the ideal solution. It's a bit of a scary thought. It's like cloning people until we have the ideal person. Right? But it's actually doing that. It's just cloning and manipulating the genotype of that situation.

So the objective that we need to have here in the optimization process, is find minimal lift score. To achieve that, we have our design variables, which are the UV-parameters. So actually, it needs to create points on that boundary surface. So the U-point is the x-coordinate. The V-point is the y-coordinate of that point. So it will create a scattered, very auxiliary situation on top of the boundary surface, and it will create points which will be the possible placement points for that crane.

And it should be within a domain. So I want to limit these design variables. I don't want to have a calculation until infinity. So limit these parameters between zero and one. That's how U and V parameters on a surface in Dynamo actually work. And the constraints in here, these are defined by that flowchart we just run through in the beginning. Right so, unreachable, non-liftable, liftable but truck issue, and liftable. These are the constraints of that design for the optimization.

And at the end, it will find what we call Pareto optimal. So it will find the ideal solution based on that convergence. So instead of going through a fixed amount of samples, it created a-- in the meantime, I'm going to play that video. It has created very auxiliary points on the surface. So it has to stay within that boundary surface, create a number of points, like for instance, 30 points, take the two best of them, make a crossover mutation, insert it back into the population, and do that same process again. But instead of making these big bunch of nodes, it will converge into a solution where it finds like 30 nodes, which are at exactly the same position, and that's the point where we have our Pareto optimal.

So you will see that the results at the end, instead of having 80 situations with all kinds of different lift scores, we will have like 10 situations, and all the lift scores will be identical. Like in this case, it will be around a thousand, and that will be the ideal solution. So for achieving that, it didn't have to go through 88 iterations. It only took, you will see it after it when it's running. It's only took-- over there, if you follow that node. It only took eight iterations to get there instead of 88. So that's much more faster for achieving that.

And therefore you need to have that specific node called single crane analysis, and the single crane analysis is a combination of the node we just used in the situation calculation, plus some additional nodes to make it ready to get through genetic optimization. So this is the result you get from it. So again, with the results you get the optimal results. You can drive then, your Revit model, and let Dynamo tell to Revit to position the crane to that ideal situation, which should be the point X is 24 meter, and Y is like one meter.

So that's two meter difference than the one I got from my parametric run. But if I would have wanted to detect that in my parametric run, then probably I had to do like 200 iterations to find that exact point. So this is what has happened in here. So run the script and that will reposition the crane to a point somewhere over there, which will be the ideal one. Still, we have some problems in there, but that's just the limitation of that crane. So we could also consider to take a bigger crane.

Maybe you say OK, these are only four elements. We will find a solution on site for that instead of installing a much more expensive crane for that. Same could be done with the double crane situation, where you also get this as a result for instance, but then the design variables are not only U and V for on that one surface. Then it's U and V for this surface, and then U and V for this surface.

Who heard already about project fractal? Good. Who likes it? Almost the same amount of hands. That's good. With fractal, it's still in an alpha version. You get more information on the exhibit hall. I'll be there at 11:30 tomorrow to do some demos on fractal, if you want to talk about it. So to stay within time in here, I'm going to do the most exciting part in here, which is the video. So it's exactly doing the same operation again, but instead of building this up in Revit, Fractal cannot connect with Revit. It's something that's done with Dynamo studio.

You have to break up the model into SAT files, export it to SAT files. Bring these SAT files into Dynamo studio in here, and send them to Fractal, and then let Fractal create all these possible design options, which is in fact, the position of the crane, but they will also change the position of the truck. So it's not only the crane. I would also ask to drive the truck around the crane, and to see if that is possible. If every position of the truck will be valuable or valid in that analysis.

So, you have to know that Dynamo studio, a lot of people asked me that, what's the difference between Dynamo and Dynamo studio? So the Dynamo one, which is integrated in Revit, is the free version, and Dynamo studio is a standalone environment. So it's based on subscription. But with Dynamo studio, you also have access to the cloud services. So you can access Dynamo Reach, which helps you to create geometries inside a format 360. Or you can also upload your design to Project Fractal.

So, first we have those SAT files. These are brought in here. These are the building elements. So this is the same like we had in Revit, where we have walls, columns, floors, and structural framing. Now we have the SAT files for doing that, OK? So if you run it, then we can explore it inside of Dynamo Studio. I have to be careful, Dynamo Studio. We will do so much more operations in here. So I want to reposition the truck. I want to reposition the crane. I want to consider multiple options in there. So I want to consider multiple surfaces, multiple situations of the truck, and have much more options in that.

So I know we are already at 2:30, so if there are people that need to go, one more thing I want to tell you before you go is that don't forget to evaluate this class. It's really important for us as Autodesk, to know if our classes are good. If you want to have more, stuff like that, next year. Yes or no. So it's doing exactly the same analysis, but now instead of doing this on the Revit elements, it's working on the solids. It will sort all the solids into a specific sub-list, which is telling us if an element is unreachable, if it's liftable, or if it's non-liftable, and so on.

You can also explore it on Dynamo Reach, which is doing exactly the same. This was the video over here. I clicked a little bit too fast. So Dynamo reach, or Dynamo Customizer-- Dynamo Reach is the name of the server, and Dynamo Customizer is that online platform, which is connected with Dynamo Studio. So I have uploaded the SAT files plus the Dynamo file, the analysis file, to that server. You share this link with someone else.

So you will actually also find the link inside of the presentation, and you can play with it yourself then. And then scroll through all these different design options, get immediate feedback inside of your model. So this means that if someone else in the company doesn't want to work with Dynamo, you can give him this and he just needs to drag the sliders. Let's call it like that. Choose a specific situation for the truck and for the crane, and get instant feedback on that in your script.

And with Project Fractal, instead of doing his operation manually, you are going to generate multiple design options just by using that cloud service. So we are going to introduce the basics of optimization in here. So it's not doing genetic optimization yet, it's just creating a wide range of design options, and then we have that parallel coordinate system again, to evaluate all the specific designs.

So it's using exactly the same link as the one we had with Dynamo Customizer. But we open it up in a different service. In this different web service called Fractal. There are multiple sliders again that can be used. So you can drag these slides again, manually, and explore all the possible solutions for the truck and the crane. Get feedback again, inside of your visual canvas, plus you also get feedback here for the elements, the percentage of elements that are liftable, that are non-liftable, and you also get a lift score.

But what we also could do, which is of course, better, if we click on this button in here called generate, which will happen in a few seconds, then we are going to let all these design options being created by that cloud service. So the cloud will tell, OK, set the crane U position on zero. Set it on 0.5. Put it on one, do this for the V-position, do this for the truck position, and so on, and so on, and so on.

So it will create all these possible combinations. Might happen that you have thousands of possible combinations then, but as you can see, we are not just considering the position of that crane. We are also considering the position of the truck, plus we are considering the position of that crane on multiple places around the building, and not just over there, or on the left and the right side. No, it will go everywhere around and see if it finds an ideal one.

So, the thing is that we are going to receive all these thumbnails as a feedback. Each of the thumbnails represents a situation of that crane plus truck. You can filter them in here, or sort them in here. You can filter them in the parallel coordinates by just dragging in here. So for instance, if I select all the ones with the lift score, which is very minimal. These are all the ones with the very minimal lift score. Select the option you want to have. That's the option, that's the best solution.

Again, as you see, it's the one we already found out. But the truck is now in a different position, which is actually easier to get there for that truck. It doesn't have to try to run around that crane. It can easily stay outside of this area. So that was what Project Fractal could tell us. We have more computing resources to achieve that goal. So if you had done that in Revit, it would take maybe like two hours to find all these options and get that result.

But there are multiple variations. You will find also in the handout, there are a few links in there with multiple variations on that. For instance, variable truck position, but also variable boundary choices. So having multiple boundaries over there instead of just that one boundary. OK, I clicked too fast. And that was it about crane optimization. There are more other examples on construction that I wanted to talk to you about, but as you see, we don't have time enough. I'm already going way over time. So these are a few ones that you can check on my block, like clearance height on stairs. So doing model checks or creating views on your elements. It's all explained on the blocks, so I want to talk about it at this point.

So conclusion, we are talking about construction. We are talking about Dynamo. If you want to see more stuff, one of my colleagues is doing a session after this one where he is going to use Dynamo and Dynamo Player to drive views, which will be uploaded to BIM 360 Docs. So it's creating a link between Rivet and BIM 360 Docs using Dynamo for that.

Come to the Project Fractal booth if you want to know more about that fractal stuff. So we are happy to see over there and help you around with your questions, or do some custom demos. I have a model with me also with the parking house configurator of MTHojgaard if you heard about that. So we are going to showcase that tomorrow between 11:30 AM and 1:00 PM.

OK, so this is the end of my voice as well. So as I told, don't forget to fill in a survey, and I'm still walking around in here. I'm still like 15 minutes in this room. So if you have questions, feel free to come over. So, thank you very much.

[APPLAUSE]