AUTODESK UNIVERSITY

CS501195

From Drawings to Data

Håvard Vasshaug
Reope

Harsh Kedia
Reope

Learning Objectives

- Understand how to effectively create large and consistent model-based datasets.
- Create and manage model-based project data requirements efficiently.
- Move design and construction data out of file-based drawing production systems and into web-based databases.
- Automate the validation process of delivered data.

Description

The world is moving from dead to live information. In the AEC industry this means several changes, but one of the biggest ones is the disappearance of drawings caused by the availability of model data. The main drivers for this change are two: model-based construction, and model-based operations and maintenance. Contractors and Building Owners are now requiring vast amounts of standardized and consistent data as part of their design team's deliveries.

The problem faced by everyone in this change, is that the tools used for creating this data are file-based systems made for delivering consistent drawing sets. Instead of displaying 3 parameters in a tag on a drawing accompanied with a schedule in Revit, design team's need to deliver hundreds of consistent properties connected with model elements across contracts.

In this talk we will go through how design teams can adjust their processes to adapt to this change, and how contractors and building owners can manage it effectively.

Speaker(s)

Håvard Vasshaug is a true innovator who believes that highly educated architects and engineers should not waste their time using outdated tools and processes to design and construct the built environment. Together with his colleagues at Reope, Håvard transforms how people work by building better workflows together with some of the most renowned companies and people globally. As a trained structural engineer, Håvard discovered how building information modeling could help designers work faster with rapid design changes through project lifecycles and spent a decade teaching people how to master it. Håvard may be best known for his highly rated presentations on digital transformation in the construction industry. From various stages globally, he has challenged the AEC industry to change through fostering coding designers. He is also known as a Bad Monkeys founder, Revit user group founder and

Design Technology blogger. Håvard founded Reope in 2017 and today leads projects and product development. Read more on reope.com.

Harsh Kedia is an architect-turned-coder who works to come up with creative solutions to some of the hard problems in the AEC industry. A jack of all trades, he works across code, design, and business to come up with cross-disciplinary answers to complex questions.

He previously led the design computation group at NBBJ where he helped improve the level of digitization at the firm through strategy, development, and training.

At Reope, Harsh is the product manager for Anker – a product that manages project metadata in the cloud and helps improve data quality and usability on construction projects. Read more on reope.com/anker.

The Story

Something changed. It happened during the last 3-4 years, and we didn't see it coming. It crept right onto our desks without us noticing it at all. And once we realized what it meant, we immediately acted on it. It was important.

New things sometimes enter your life without you noticing it. That's probably why we can't remember exactly when we first saw it.

So, what happened? It's disappointingly undramatic. A lady named Veronika came over to our desk and said something like "Some guy told me that I need to fill in all this data on my doors. I have 2500 doors and 3 kids and would like to spend more time with my kids than my doors. Can you help me?"

10 years ago, Veronika would model a door in Revit, tag it and print a door schedule along with a final drawing set. But, recently, building owners and contractors - Veronika's clients, have been demanding high quality, model-based datasets as the final delivery from project teams. The fire rating on Veronika's door is no longer delivered in a printed schedule, but as a property on an IFC element.

The Norwegian Directorate of Public Construction and Property - Statsbygg - in 2019 relaunched their BIM requirements, Simba. Previously Statsbygg's BIM requirements were limited to the delivery of IFC. With the 2019 relaunch, Statsbygg also required their own structured information - Properties in Property Sets in IFC - in deliveries. They distribute the requirements through a dedicated website where you can find information about the various versions, guidelines for usage and download files that contain the requirements. The data is also maintained and accessible in an online database. The Norwegian Hospital Construction Agency - Sykehusbygg - also has a requirement database; "BIM requirements", that exists in an SQL database on Azure. The state-owned company responsible for the Norwegian national railway infrastructure - Bane NOR - are implementing the same strategy through their KIM - Requirements for Information Modeling. They are all in different stages of development, but the trend is clear: stakeholders require standardized information. We are hearing about the same in many different countries. And it's developing beyond public owner institutions; contractors also are in desperate need for information.

Why is this, you might ask?? What problems do those pesky building owners and contractors have with our beautifully detailed drawing sets? It's been working well for hundreds of years, why change it now?

Well, for a couple of reasons. The first one, is the benefits of model-based construction. When you use a BIM database, or a model, as the central reference point on a construction site, it leads to some real advantages. Contractors can connect model data to other software downstream to plan, calculate, and analyze the project in ways that drawings simply didn't allow. They can check the cost and amount of the gyp in the project directly by doing a takeoff from the model and plan their procurement in a faster and more intelligent manner.

The second one, is for facility maintenance and operations. The Norwegian hospital authorities faced this problem during the Covid-19 pandemic. They didn't know how many rooms they had in their hospitals to treat the expected rise in cases. Seriously. This is because all their records were in 2-dimensional drawings and schedules. To dig out this information from that is extremely time consuming, manual, and expensive. In other words, it would have been too late. On the other hand, if they had a central database with accurate models of their hospitals, to find

this information would just be as simple as writing a query. This was a simple example, but a lot more is possible once you have a model of the building. When a socket stops working, you can find the right one with the exact right specs, by just looking at the identity number printed on it, and cross-checking this with the BIM. When it's time to change your carpets, you know exactly how many square feet to order.

All of this becomes possible, and easier, with a high-quality model.

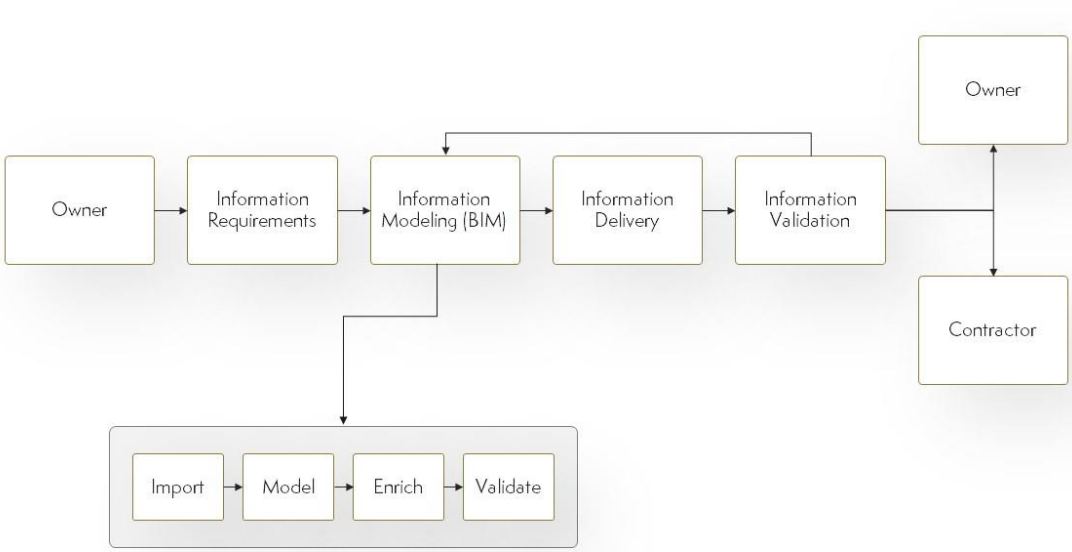


Figure 1: The new design process with Import, Enrich and Validate

The Problem

Figure 2: The Problem: Data Quality

But here's the problem. Theoretically, model-based construction, or "data", does have massive productivity gains. However, it has created challenges for each stakeholder in the value chain. Project Teams (Veronika and her colleagues) are struggling to deliver this data. Building owners are having a tough time checking this data and defining their requirements clearly. And contractors don't have the tools to leverage this information and get insights out of it. But all this stems from one core problem faced by almost everyone in the industry who are trying to move from drawings to data - poor data quality. As the adage goes, and pardon our French here; shit in, shit out.

Poor data quality seems to exist because of 3 reasons:

1. The workflow from data requirements to design tools and delivery is broken.
2. Consistency is hurt by humans performing manual operations.
3. Data Validation is in almost all cases performed manually and is disconnected from design and delivery processes.

The keywords above are requirements, consistency and validation.

Requirements

Figure 3: Requirements: Broken workflow

Most building owners and contractors don't have data requirements. They rely on existing graphic standards for drawings, schedules and schemas. Some have Excel spreadsheets and PDF's that they distribute with emails to project teams, on shared project drives or at best upload to their websites. A few public building owners have started establishing digital requirements. These are usually based on international standards and are often available through downloads, direct database access or through API's.

The overarching problem we face with managing data requirements has to do with change. If there was one thing that BIM taught us, it was that changes are best managed in one place. A database. Yet, changes to data requirements during (or after) a project lifecycle cause major headaches. Why?

1. First and foremost, a Shared Parameter in Revit cannot be renamed. This means that anyone needing to rename a Shared Parameter in Revit, needs to store all parameter values somewhere else (like a CSV or Excel file), delete the old parameter, create the new parameter and finally write all values back to the new parameter. During this process you will hope and pray that no one added or deleted any elements in the model and that way changed sorting, unless of course you used ID's. You can avoid Shared Parameters and rely on Non-shared Parameters, but then you cannot manage the values in Revit Schedules. Finally, you can of course not change the Shared Parameter in Revit and just change the translation during export processes, but that's not very sustainable long term for an organization that needs to communicate and collaborate. When someone says "Fire Rating" in a meeting, you want everyone to think of the same thing.

2. Second, if the requirements are communicated using distributed documents (PDF's, Excel files, etc.), the process of recreating them as Parameters in Revit is manual, or at best scripted with third party applications. If you have received a PDF with data requirements you have no other way of creating parameters than doing it with your eyes and fingers, like a good old factory worker during the Industrial Revolution. If you have received a spreadsheet, an xml, json or csv, you need a custom addin or a script that creates the parameters programmatically. Neither of these (the downloaded files, addin or script) are usually easy to scale or distribute effectively, which means that the usability threshold is perhaps low for you but high for the person sitting next to you.

Figure 4: The manual process of defining properties based on requirements

The Norwegian Hospital Construction Agency, mentioned above, have developed a Revit integration for their BIM Requirements Database. It makes it much easier to overcome issue number 2 above, the distribution and one time implementation of their information needs. But when changes occur, we are still stuck with Revit's constraints when it comes to Shared Parameters.

Some have just started with this, and some have come a long way. Some have not started at all. But most of the industry now realizes that the process of defining, distributing and implementing information requirements needs to be digitized for effective change management.

Consistency

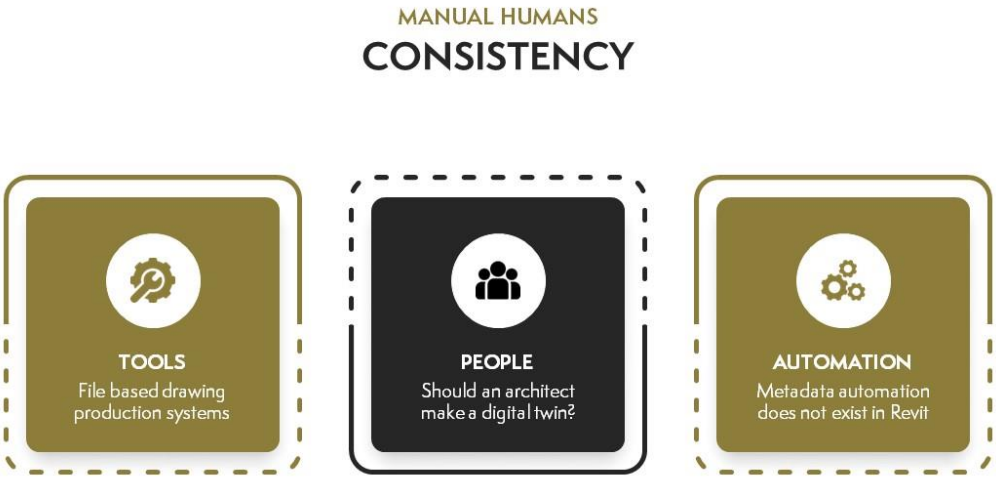


Figure 5: Consistency: Manual humans

We met a guy called Jan once. He was a project director on a big project that was implementing model-based construction. He was also about to retire, after a long and impressive career in the building industry. Jan was asking us about what our thoughts were on the data quality of the deliverables of the design team working with him. He was a bit worried because he had performed random samples in some of the files, and you know how it feels when you have 500000 elements in your project, and you check 4 of them and find a mistake. When we looked at the data, we confirmed his suspicion that there were inconsistencies in metadata between both disciplines, files, categories and even similar elements in single files. And when Jan asked us how this was possible, and we explained to him how parameters work in Revit and that you cannot really make a dropdown of possible values, we saw 45 years of industry experience turn to confusing disbelief in his eyes. “Really?” he said. “In 2022?”.

It’s not easy to manage large and consistent metadata on projects in Revit. Let us guide you through some of the reasons why.

1. Revit is great at many things, but it is not very sophisticated at helping people enter correctly spelled data. Some parameter types help users with not getting it totally wrong, like Integers for example. But most of the metadata that is needed for information-based processes must be entered using Text Parameter type, and in Text Parameters you can basically enter anything (and people do!). We need a better way of making it easier for people to enter correctly formatted data, and an editable dropdown list would be a great place to start.
2. Whenever you ask humans to use their eyes and keyboards to create and manage large text-based datasets, the datasets will differ. It’s like a law of nature. It will

happen, no matter how much you want it not to. People have a bad day. They get disturbed and lose their concentration. They forget what was discussed in that meeting last week. And sometimes they just type wrong.

3. There's not much built functionality around metadata automation in Revit. Apart from Rebar Numbering, there are no native processes that enrich elements with data needed after design. Let's not mention the Mark parameter here because that is just annoying and useless, as is Room Number. This means that anyone needing to create large and consistent model-based metadata either must do it manually, or with scripts or addins. Addins can be scalable though they require training. They also tend to add up quite a bit. Scripts are quick to make but very rarely easy to implement on projects with more than 10 people working on them due to a variety of reasons.

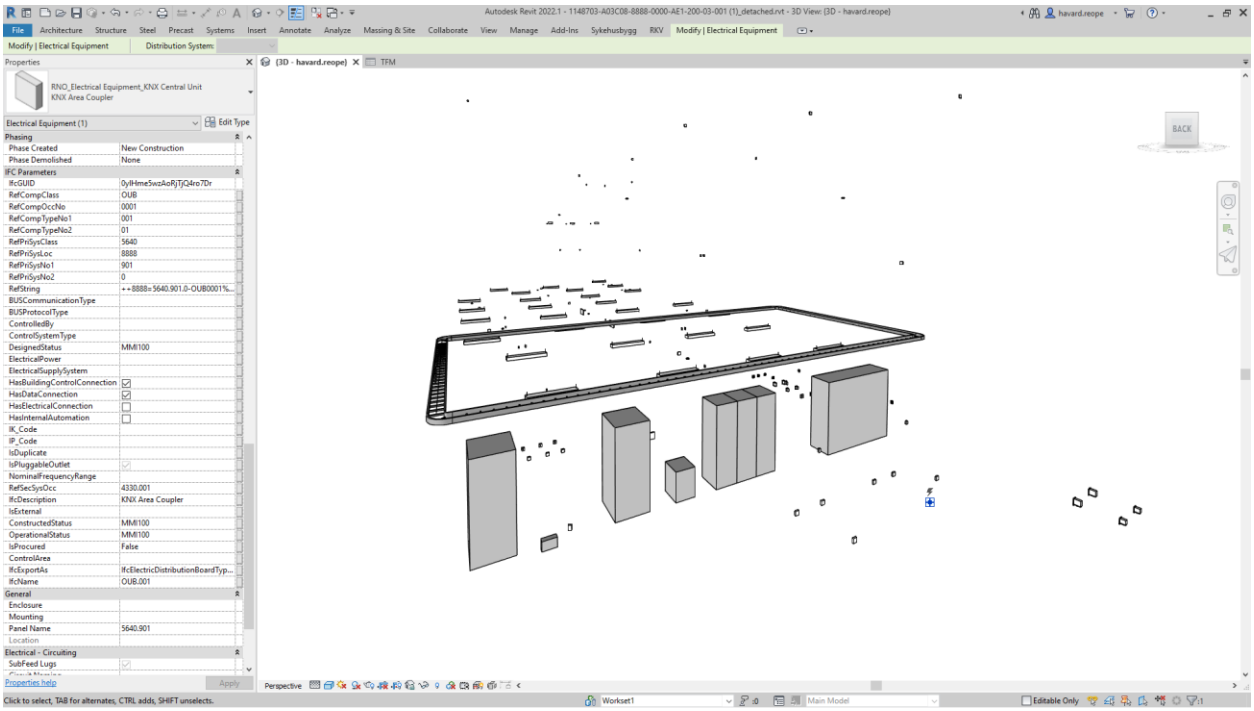


Figure 6: Revit Parameters all set for a wonderful day of manual labor

Validation

Figure 7: Validation: Nobody knows

“You don't need validation if the data is correct”, a friend of ours said. Of course not, but how do you know it’s correct? If you don't, you need an automated validation process or you will lose your mind. It doesn't matter if you are validating your own work before sending it to your client, or you are the client checking the quality of your design team’s delivery. Anyone doing manual validation is set for a short career in AEC. Why? Because it’s the most boring thing ever! Quality Assurance and Quality Control, basically checking your own and other people's work, is one of the hardest positions to fill. It’s not why people go to engineering or architecture schools. And when work is boring, it has a funny tendency to be done poorly, or not done at all. “We have a feeling we’re getting shit data, but we don't know”, said another friend. “It’s too important to not know.”

Data validation is crucial, and it needs to be automated. There are a few tools and processes that can help you today.

Along with their rebranded BIM Requirements, Simba, in 2019 Statsbygg published a set of machine validatable files; mvdXML. They contained structured rulesets that allowed people to programmatically document if a property existed and had a value or not. The next generation of this is the IDS format that, in addition to the above, lets you automatically check if values follow a regex pattern. This is important because it allows us to get data quality feedback much faster than before.

For designers using Revit a few come to mind.

Anyone learning Revit (hopefully) learns about Schedules on day one or two. They are one of the core tools for working effectively with the database. If everything is set up correctly with

Parameters in a Revit file, you should be able to quality assure most of the metadata in your file with your eyes. But there is no built-in processed validation in Revit.

Every time a poor soul on the planet asks, “can you do this in Revit?”, the answer has been “no, but you can do it in Dynamo". Even with minimal programming skills you can extract all parameter values and display them in a list, regardless of their Revit Parameter Settings. With above average skills you can compare them with a set of external requirements from a spreadsheet or database and create a concentrated report highlighting the errors. But a Dynamo script still just works on one file, and they are hard to scale, as mentioned previously.

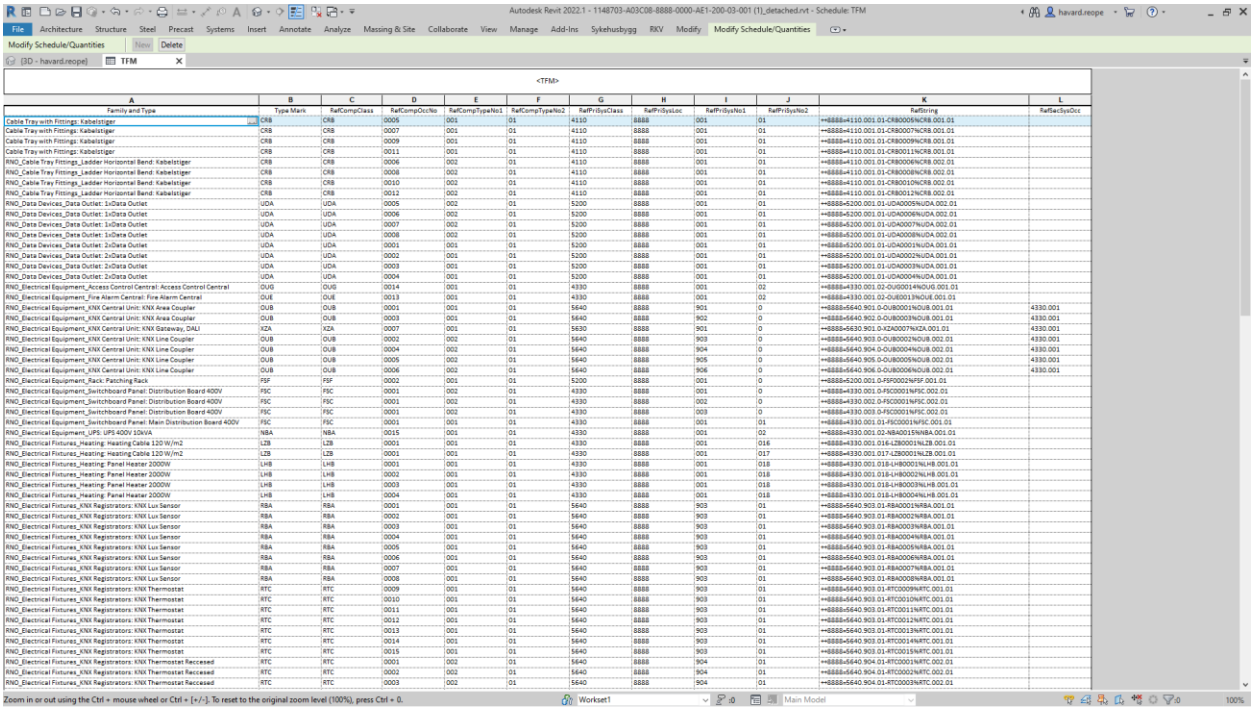


Figure 8: The joys of quality controlling Revit Parameters with your eyes.

For contractors and owners who receive IFC deliveries, or designers who want to check their delivery before sending it, there are quite a few options.

IFC Viewers lets you open IFC files and inspect element geometries and properties. Some of them have property tables that let you aggregate model data across files, so you don't have to click and select everything you want to check in singular models. Neither of these have any programmatic validation functionality built in and like in Revit you are stuck with your eyes.

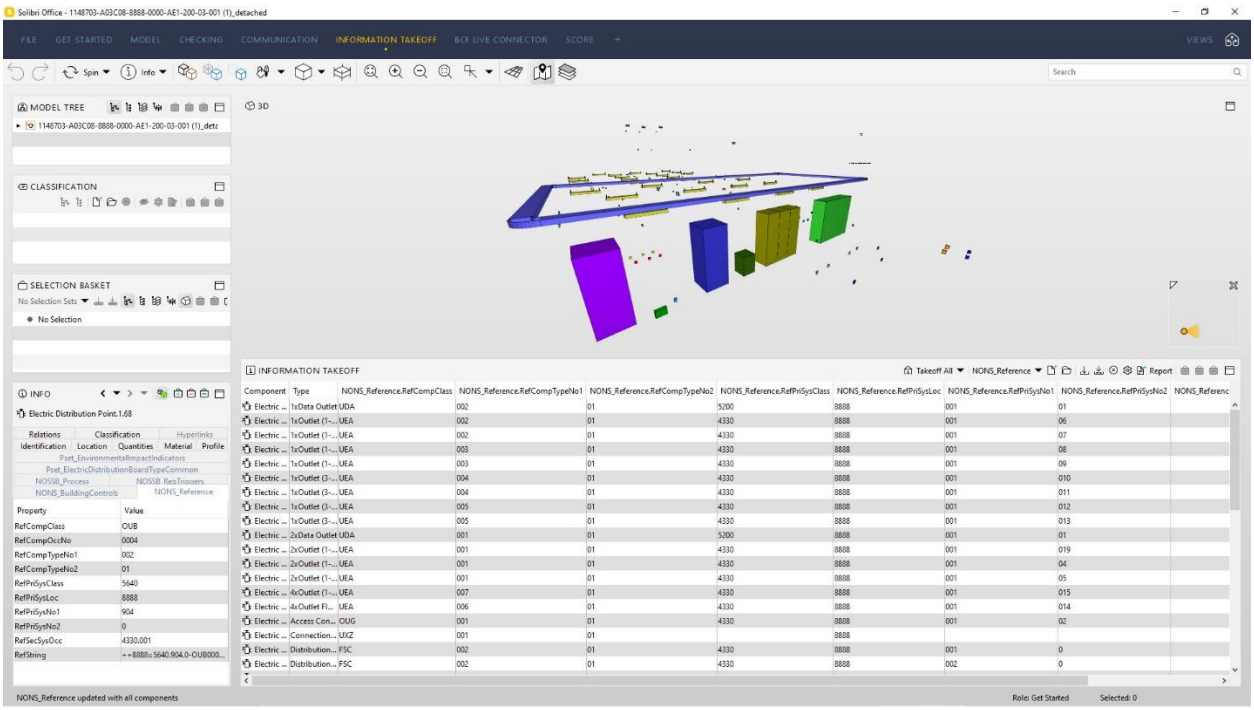


Figure 9: The emotional bliss of quality checking Solibri Information Takeoffs with your eyes

There are some desktop applications that have built-in validation, and even some that are built exclusively for it. For various reasons they are all not widely used. One of the most common reasons is that most of them are desktop applications that need training and skills to be used. For most of the people who receive IFC, having to learn a software just to check if what they receive is good enough for construction or post construction, does not bring a smile. Jan (in the story a few sections above) just wants to see or hear something along the lines of “this delivery has 100 % correct data compared with your information requirement version this and that”. He does not want to learn about “click here, open this, dropdown there, and oh guess what there are a bunch of settings everywhere”. That’s a waste of time for Jan.

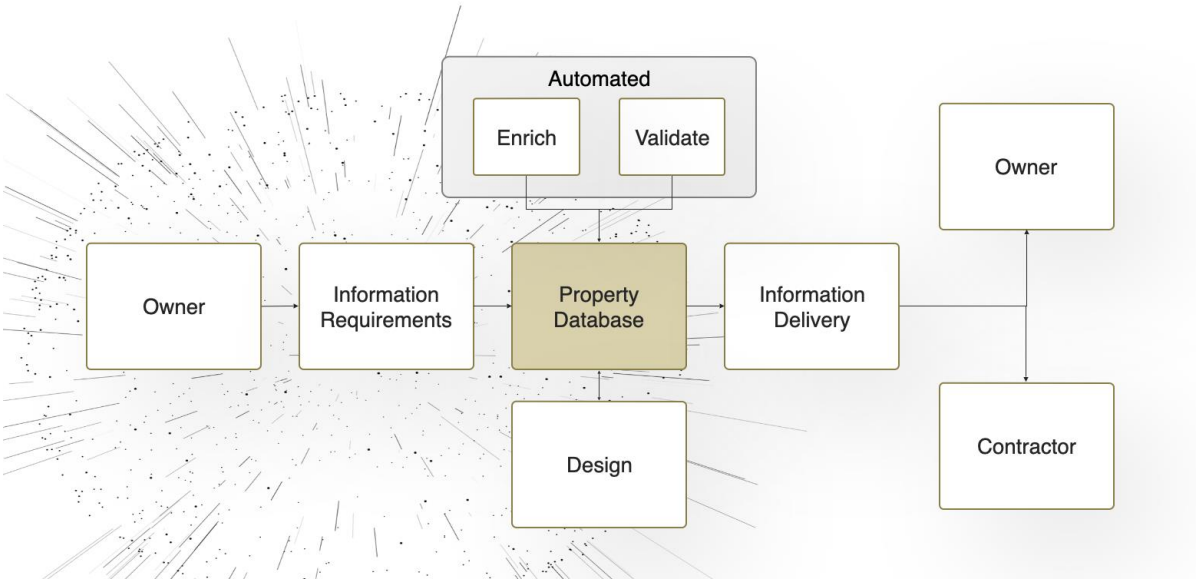
All the problems above we have faced in some form on projects we have worked on during the past years. Håvard has spent a good chunk of his time at one of the biggest in his country: the New Government Quarter in Oslo, Norway, with Statsbygg as client. Despite all the automation and smart systems, he and his colleagues developed and implemented, we cannot escape the thought that a dedicated solution for automating data quality outside design software would have been amazing, to say the least. Is there anything like that on the ever-changing horizon of AEC? Spoiler alert: yes there is.

The Solution

Something happened last year. We were having one of our internal meetings at Reope and someone was presenting a property manager we had built for a client in Rhino. And there was like a brain fart moment… why are we building a custom property manager for the nth time? Why not just remove all these properties from the design application and manage them in an independent database in the cloud?

And that, ladies and gentlemen is our solution. This might end up sounding like a bit of a sales pitch, and in part it is – we've built this web-based property manager and it's called Anker.

The reasoning is simple. Design applications are good at design. Some are even great. But unfortunately, they suck at managing data. They use files - which leads to scattered input and slow work processes. If you have a data manager, that only deals with properties, and works across files, you have a solution.



This way, the owner easily defines their information requirements, the designer easily creates their designs, and the property database in the middle takes care of the rest.

This property database can be set up with easy imports of the owner's information requirements as well as validation tools to check against this requirement. This way, you know you are delivering data correctly, before it goes out to the client and to the site - where mistakes become a heck of a lot more costly.

Our property database, Anker, relies on 3 main pillars:

- Easy definition of requirements.
- Focus on automation and a simple user experience.
- Powerful validation tools.

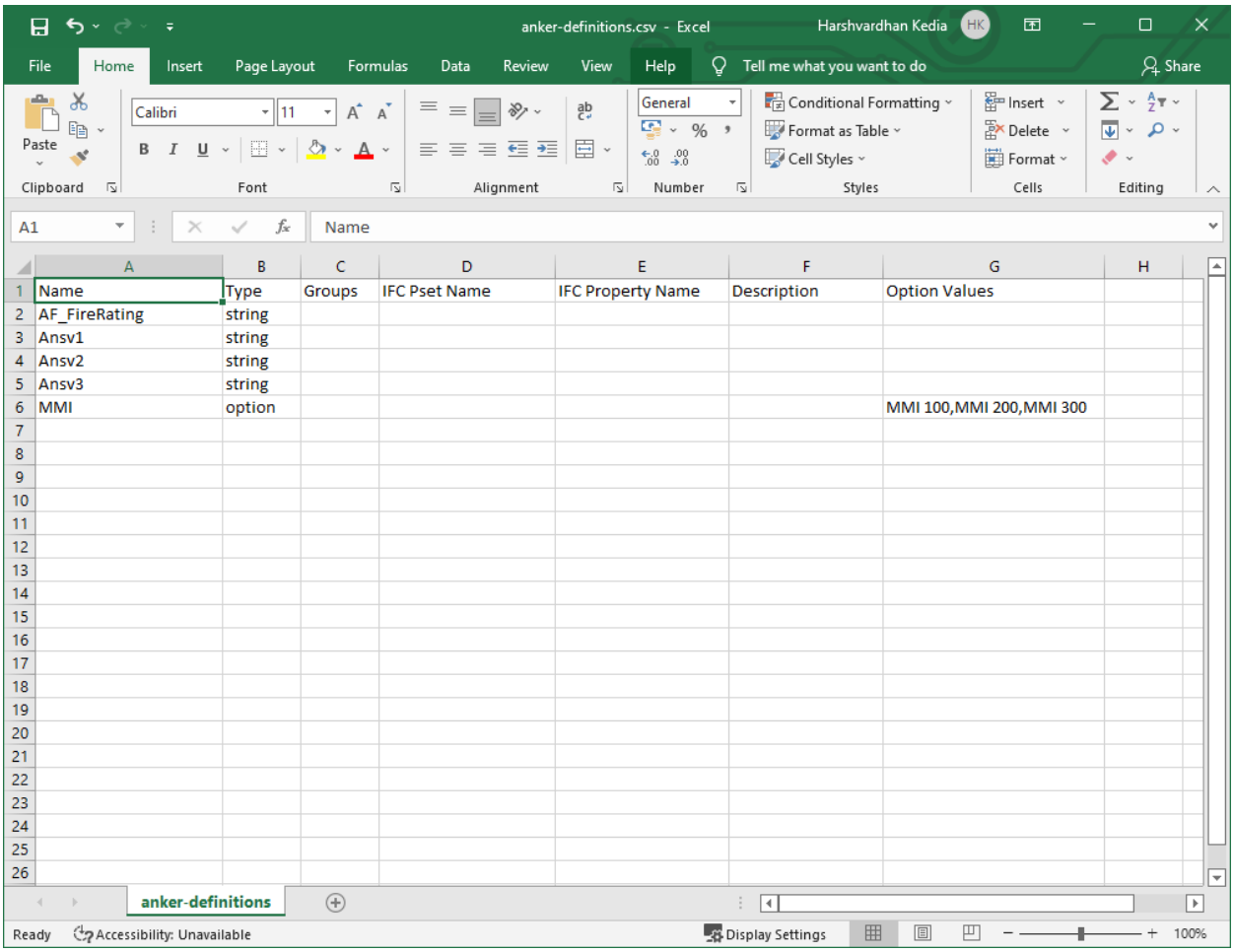
We believe that a focus on each of these, will let you get out of shared parameter hell and focus on what really matters - the design.

Let us dive into each one in more detail.

Defining Requirements

This problem is probably the easiest to solve once you have a property database. As we already know, creating requirements through shared parameters in Revit sucks. But once you have a property database, like Anker, all you need to do is a simple import.

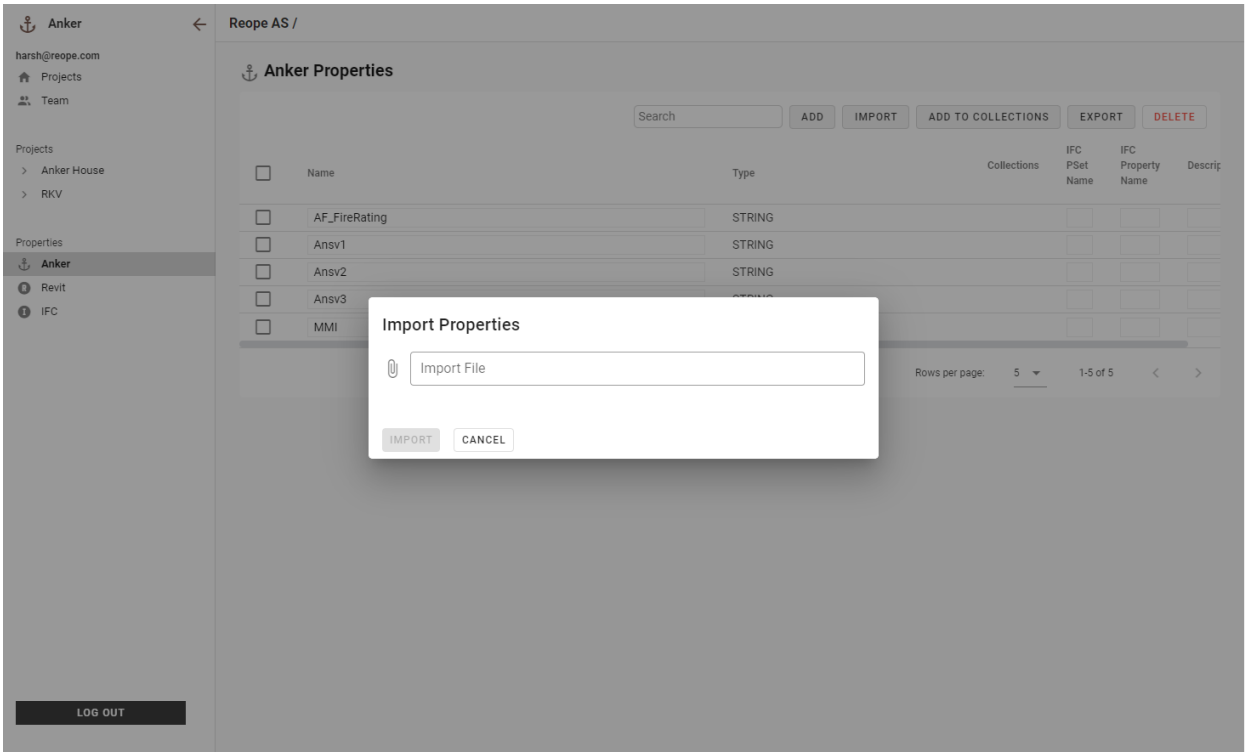
Say you have a property requirement document, like the one below.



	A	B	C	D	E	F	G	H
1	Name	Type	Groups	IFC Pset Name	IFC Property Name	Description	Option Values	
2	AF_FireRating	string						
3	Ansv1	string						
4	Ansv2	string						
5	Ansv3	string						
6	MMI	option					MMI 100,MMI 200,MMI 300	

As you can see the requirements specify the type, aka “string”, “integer”, or “option”. And the owner also defines some accepted values – “MMI 100”, “MMI 200” etc.

All you then need to do, is import this directly into your property database, like so:



The best part about this process is that then the UI only allows you to enter valid data. If it is not part of the options, then you cannot enter it.

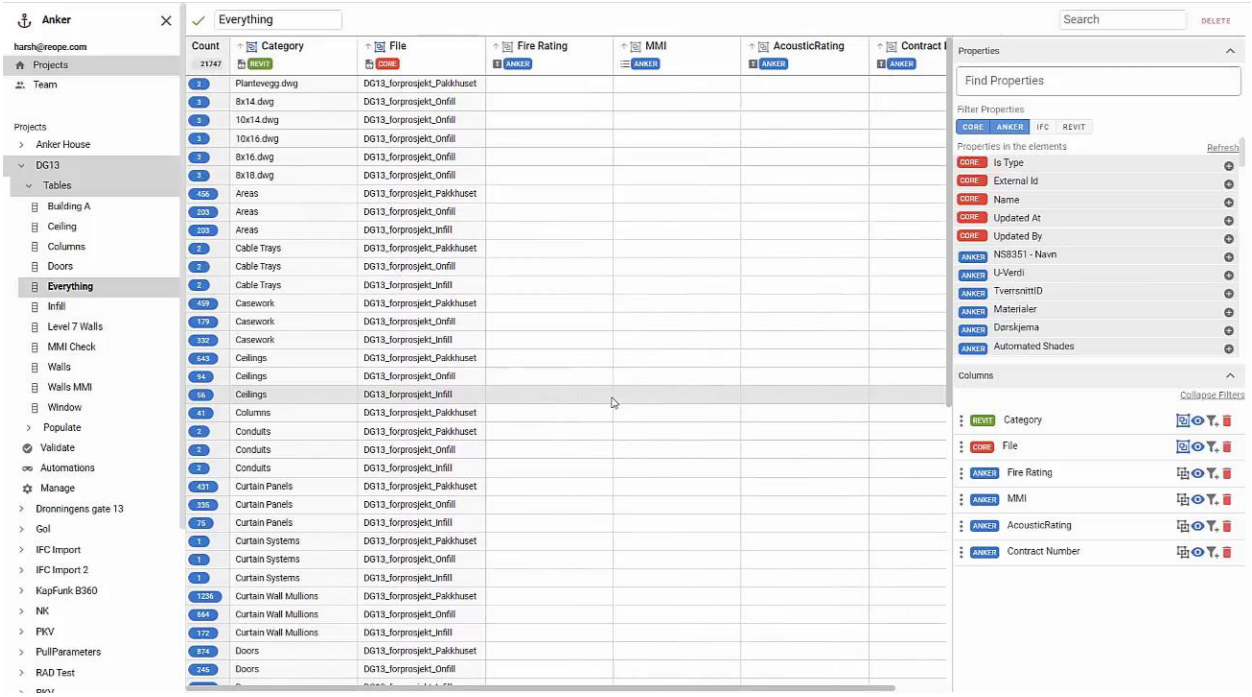
Walls

Count 854	AcousticRating ANKER	File Filter CORE	Type REVIT	MMI ANKER	Phase ANKER	M ANKE
11	10db	DG13_forprosjekt_Infill	YVI09 - Isolert sjaktvegg	200	Phase 1	
24	10db	DG13_forprosjekt_Infill	IVI09 - Påforing 71MM	200		
7	10db	DG13_forprosjekt_Infill	MTV 03 - Massivtre 220mm	100		
15	10db	DG13_forprosjekt_Infill	MTV 04 - Massivtre 180mm	200		
21	10db	DG13_forprosjekt_Infill	IVI06 - Sjaktvegg 150MM	300		
12	10db	DG13_forprosjekt_Infill	YVI10 - Kledning alu	400		
4	10db	DG13_forprosjekt_Infill	YVI11 - iso	200		
39	10db	DG13_forprosjekt_Infill	IVI07 - Vegg badekabin 100MM	200		
2	10db	DG13_forprosjekt_Infill	YVI12 - Iso med kledning alu	200		
7	10db	DG13_forprosjekt_Infill	YVI14 - Parapet 02	200		
13	10db	DG13_forprosjekt_Infill	IVI02 - Lettvegg 83mm ensidig	200		
6	10db	DG13_forprosjekt_Infill	YVI06 - Iso med trekledning	200		
10	10db	DG13_forprosjekt_Infill	YVI13 - Parapet 01	200		
5	10db	DG13_forprosjekt_Infill	YVI05 - Betong 200	200		
147	10db	DG13_forprosjekt_Infill	IVI03 - Lettvegg 96MM	200		
6	10db	DG13_forprosjekt_Infill	YVI07 - Yttervegg kledning av tr	200		
31	10db	DG13_forprosjekt_Infill	IVI08 - Vegg badekabin 113MM	200		
4	10db	DG13_forprosjekt_Infill	YVI04 - Stender med tegl 396M	200		
8	10db	DG13_forprosjekt_Infill	YVI03 - bindingsverk med Tegl	200		
46	10db	DG13_forprosjekt_Infill	IVI05 - Påforing 118MM	200		
47	10db	DG13_forprosjekt_Infill	YVI01 - Indre Sjikt Trekledning	200		
7	10db	DG13_forprosjekt_Infill	YVI08 - ISO trepanel	200		

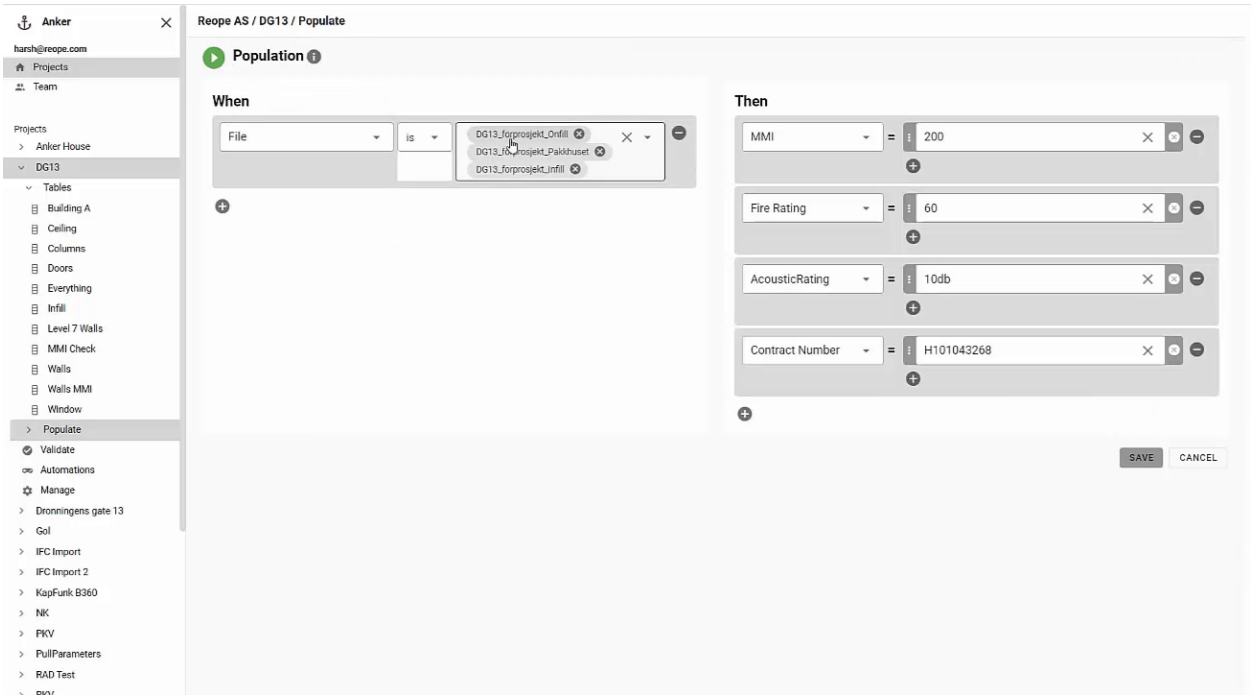
Automation

Let’s face it, there’s a lot of manual work involved when you work in the construction industry. This is doubly true when you build from models.

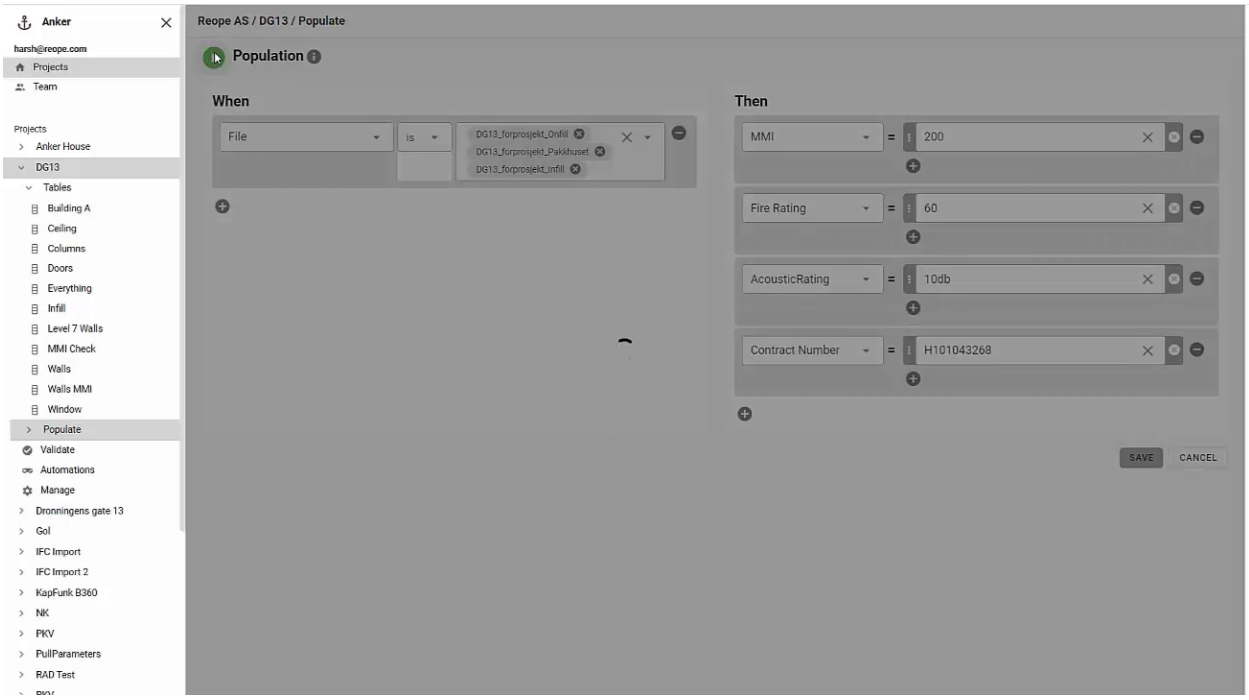
Data entry for a simple project can look something like this:



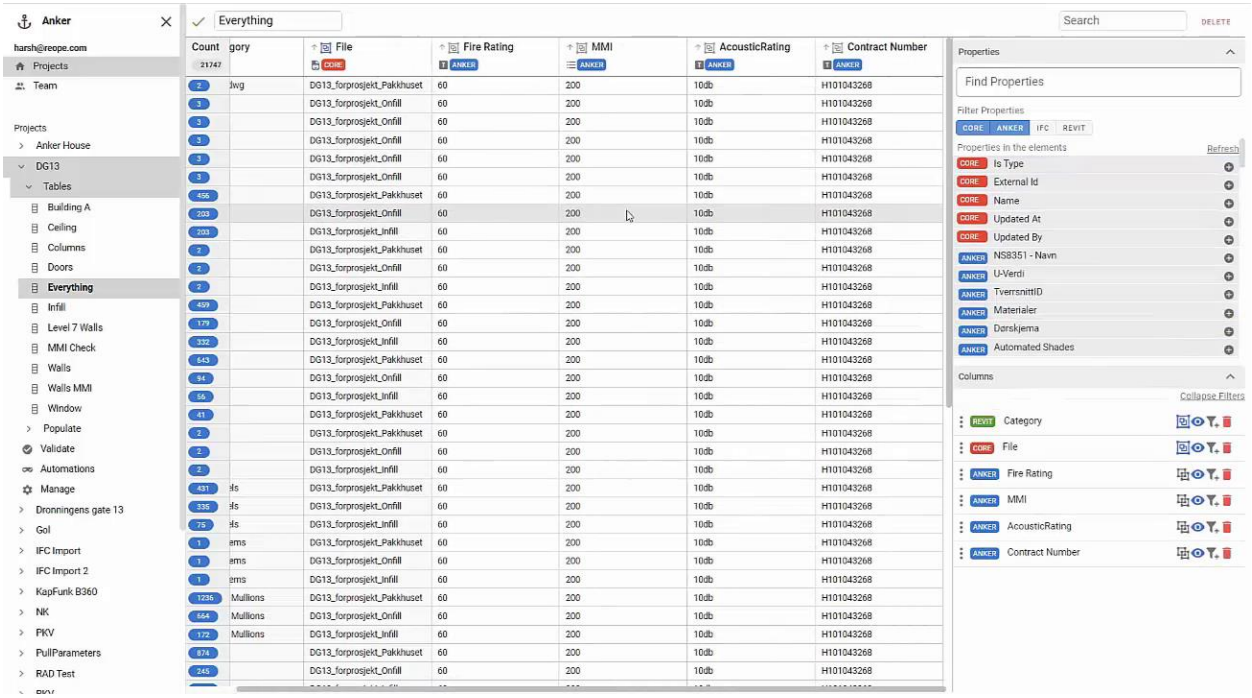
You have 4 basic properties to fill out on around 20,000 building elements. Doing that math adds up to 80,000 total data inputs. That's a lot of work to do manually, especially if it can be easily automated. And that's where Anker's population tools come in.



You define basic rules – "in files x, y, z I want the properties to have these values" and then hit "Play".



It runs for a few seconds and viola; your data entry is done. No opening 5 different Revit files, no errors because someone did not sync to central.



This was a relatively small task, on a small project. But you can imagine how easily it can get out of hand when you increase the number of properties, and number of people on your project. Which is why we have an internal motto at Reope – *automate, automate, automate.*

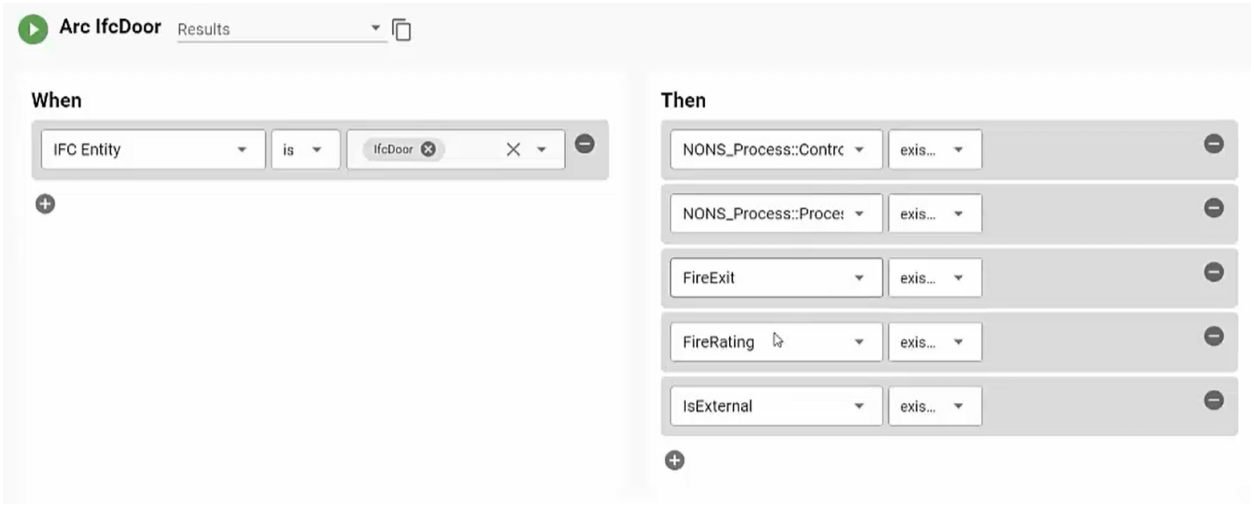
Machine Validation

If you work as an architect, the practice of “red lining” drawings is well known to you. In part, this is a useful technique to learn about your project and the rethink the decisions that have been made. But in part, this is also just a practical requirement since a machine is not able to check this for you first. But the moment you deliver data, not drawings, you can ask machines to validate the basics before you use your human criticality to check your choices.

What does this look like?

Well, first you import property requirements from the building owner.

You can see that the owner says, all IFC door elements, must have these properties on them.



And if you hit the big “Play” button, the machine tells you what is missing in your models. Instantly.

This way, you can be sure of a baseline of your delivery to the site before you send it.

The Conclusion

If model- and information-based construction, operations, facilities, maintenance and development is the future, the corner stone of that transition is information integrity. People need to be able to trust the quality of the data to be able to extract the value of the information that comes from it.

If we are going to trust the data, we need better tools for creating and quality assuring it.

The next step is inviting other people than the main design disciplines to work with the information in databases. 20 years ago, architects and engineers learnt databases. Now it’s time for the contractors and facility operators to do the same.

If everyone is going to work with data, we need better tools for accessing and manipulating it.

We need better tools that support this transition. Let’s create some!