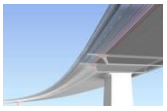
AUTODESK UNIVERSITY

TR20680



CL_2M with BIM in Infrastructure using Vault, Autocad Civil3D and Navisworks.

Joschy Rausch
Royal HaskoningDHV

Learning Objectives

- Understand the principles of CL_2M and learn using it in various projects
- Understand the importance of real-time data in design workflows
- Learn how to use dynamic corridor solids and property data in AutoCAD Civil 3D
- Learn how to use database tools in Navisworks to link real-time data from various sources

Description

The future of Building Information Modeling (BIM) is closed loop lifecycle management (CL2M). With the availability of cost-effective radio frequency identification (RFID) sensors and the LoRa network to get real-time data combined with automatic receivers linked with databases, we can make use of the Internet of Things (IoT) to enhance the built environment. To use CL_2M in projects we have to start in the design phase (beginning of life) and then use this data in the maintenance phase (middle of life). In this course we will model and use data for a tunnel and bridge project to ensure the requirements and the lifetime of prestressed and other lifecycle sensitive parts. With the help of Vault software, AutoCAD Civil 3D software and Navisworks software, you will learn how to make use of real-time sensor data inside of your BIM models. Using these tools, you can implement RFID tags in your design (for example, Dynamic corridor solids with property data) and connect to external databases (in this example, Excel and SQL IoT Hub) communicating with sensors used in the field to measure construction requirements and feeding this info back into the lifecycle. This session features Navisworks Manage, AutoCAD Civil 3D, and Vault Professional.

Your AU Expert

I am Joschy Rausch and I have a bachelor's degree in the built environment and have been in the civil infrastructure / urban planning industry for more than 9 years. I currently work as Building Information Modeling (BIM) civil coordinator in infrastructure and urban development projects and I'm responsible for structured workflows and the integrity of BIM models in project teams for large projects (10-400 km²). I also teach various classes globally within my company, ranging from one-on-one coaching and distributing screencasts, to giving the keynote and running in-depth software classes in my company's annual global designer event (200-plus attendees). My side activities include leading and operating VDC (virtual design and construction) and integrated concurrent engineering (ICE) sessions; providing insight into BIM and product lifecycle management processes through data visualization; developing real-time visualizations based on BIM models; giving real-time presentations of Masterplans (interactive drive through); and developing Oculus Rift models for further enhancement for the perception of BIM models.

Understand the principles of CL_2M and learn using it in various projects

Closed loop lifecycle management has its origin in product data management (PDM) and product lifecycle management (PLM). CL_2M enables information gathering, processing and exchange of a product / object throughout its phases of life:

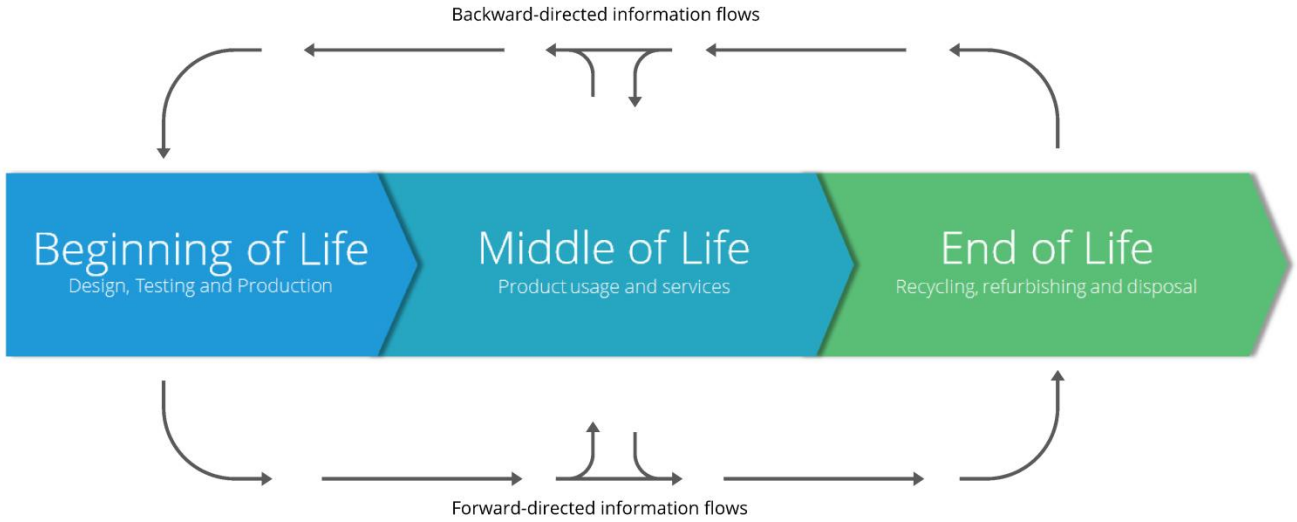


IMAGE 1: CL_2M TIMELINE

The information gathering, processing and exchange and all technologies involved can be used in a wide variety of lifecycles; supply chains, product manufacturers, retail, healthcare etc.

But why do we want to use this lifecycle solution? The answer in short is to use our resources with the best possible outcome. And with this fine-tuned outcome we can change our processes in a way so that we can produce and manufacture less, more specific or use different materials, make products that are needed and tailored to their functions and make a smaller carbon footprint. For this class we'll be focusing on the Infrastructure design and building chain lifecycle.

If we focus on the CL_2M timeline as seen in image 1 for Infrastructure we have the following phases: design, maintenance and demolition/recycling. To incorporate CL_2M we have to establish new information flows. In traditional infrastructure engineering (with or without the use of BIM) the forward directed information flow is already established. We have to add the backward-directed information flow into the process.

Middle of Life (maintenance phase) and winning of Realtime data

The most important backward information flow is the winning of real-time (sensor) data in the construction and maintenance phase. Here we get the insights of the built objects. We can see if the object is still in range of the specified requirements. There is a wide variety of sensors available; concrete embeddable sensors, load and strain sensors, acceleration sensors, pressure and load transducers, displacement sensors. The list is long and not only focused on constructions. We can think of noise calculation sensors, particulate matter sensors. The possibilities are almost endless. For our purposes we have to link the gathered data from the various sensors to a useful medium to use it in our BIM models. In this class we will look into two different methods, but there are more options and also variations possible on these methods:

RFID

Radio frequency identification (RFID) tags are simple, cheap and easy to use. These tags give out a signal with data to a RFID reader. This reader will then communicate that data with the provided solution (local database / network database / cloud database). Due to the technique of the radio frequency the range is not far. This means that the data has to be gathered manually. While this is a good method to change and maintain objects in easy to reach places with not so regular data updates, it can be impossible to use in for example cables in a cabled stayed bridge 50 meters high. For these instances we can make use of LoRa:

LoRa

Lora is Long Range Low Power LPWAN; this technology can exchange small amounts of data between objects and systems with ultra-low power consumption. This network can be used in a long period of time (LoRa units on batteries can function for 15 years) in hard (or near impossible) to reach places. With this method the data will be automatically send to the desired database logistics, and can be used in various ways.

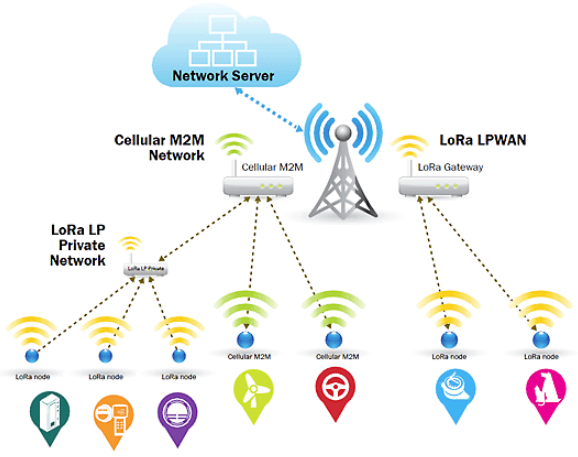


IMAGE 2: LORA PRINCIPLE

Why do we need this information in our BIM model?
To have our information always gathered in one place we need a BIM model that can handle the incoming data and link it to its representative counterpart in the model. The visual understanding of the incoming data is of big importance. The BIM model can help with these visuals.

Understand the importance of real-time data in design workflows

Now we know what CL_2M is, and what new things and methods we have to introduce, we need to know what will happen in the design phase. To create a BIM model where we can make use of all the real-time sensor data gathered we need a framework to implement where, what and how many sensors we need in the maintenance phase. We must decide in advance what the object to be built needs in its maintenance phase and what we want to improve in regards to the characteristics of these objects. When we have these prerequisites, we know where we have to shift our focus (for example: steel strength in bridges, or displacement in junctions in tunnels etc.)

The definition of sensors will happen in the design phase. In this phase, we'll dictate where the sensors will be placed, how many and what kind of sensors, but more importantly the unique IDs from the various sensors will be assigned.

How do we separate and sort in data?

IDs are the most important part of the whole lifecycle sensor approach. The IDs are the link between raw and expressionless data and the BIM model with the data linked to the actual designed object.

If we look at image 1 we can see that the information flows go in every direction. If we create the IDs in the design phase we can securely monitor what data goes where and what it is used for.

What do we want to do with our BIM model? What are our BIM goals?

All these steps are necessary to ultimately get these results, a BIM model which facilitates a link between Realtime data and designed objects and visualize this data:

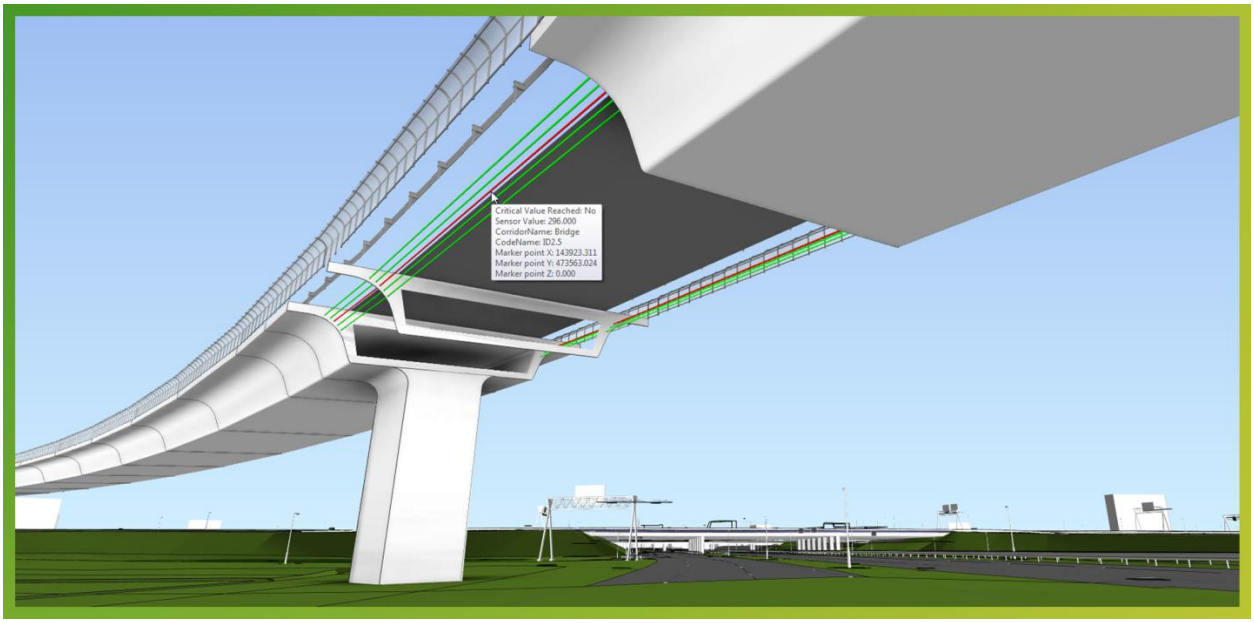


IMAGE 3: BIM MODEL WITH LINKS TO EXTERNAL REALTIME DATABASES

Learn how to use dynamic corridor solids and property data in AutoCAD Civil 3D (Beginning of Life)

For this class we use an Infrastructural project which includes a highway bridge and a highway tunnel:

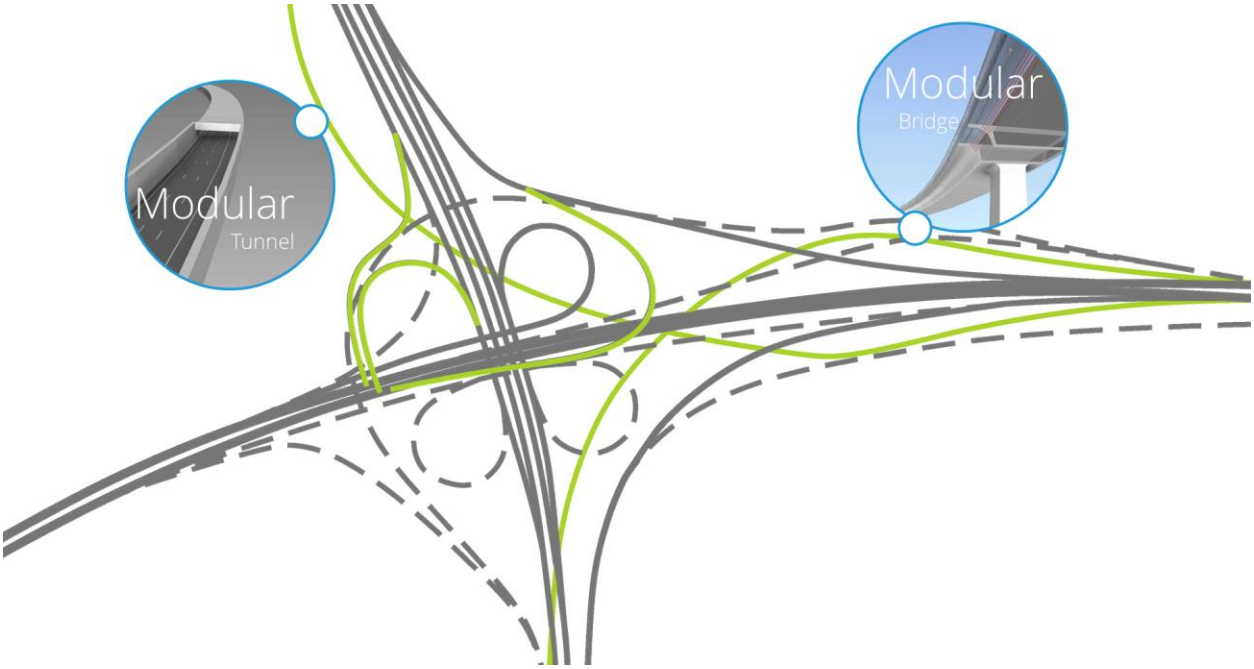


IMAGE 4: OVERVIEW EXAMPLE PROJECT

So how do we incorporate these Sensor IDs and where? We want to store the information in a way so we can use it in our BIM model.

In Civil3D 2017 we can use property data, which is an extra metadata information layer that can be used on different objects inside Autocad / Civil3D. If we use this metadata setting with automatic corridor solids we can control the IDs in an automated way. We can use two options when we are adding this metadata:

Add custom property data object related.

We can add property data for each object in our model. With this way we have all the flexibility to choose what goes where. The disadvantage of this method comes to light if we work with corridor solids where a lot of changes might occur. Here it can become very cumbersome to update and regulate all the changes in ID codes. Here option two comes in place.

Property data combined with automatic corridor solids.

Due to the automatic character of property data with corridor solids we can update codes faster and with better control. If we start at the root of the corridor (assembly codes) we can use the best of both worlds.

How do we put the wanted data in places we want?

At the start of coding for using IDs stands the subassembly. Here we are going to state which solid will get an ID and thus a sensor attached to it. You can do this with standard subassemblies as found in Civil, or you can use custom made subassemblies in the SAC. For this example we have two different subassemblies made: a modular bridge and a modular tunnel. These building blocks have the character of prefabrication and easy and quick assembly on the project site. The Civil model will be built in the same way.

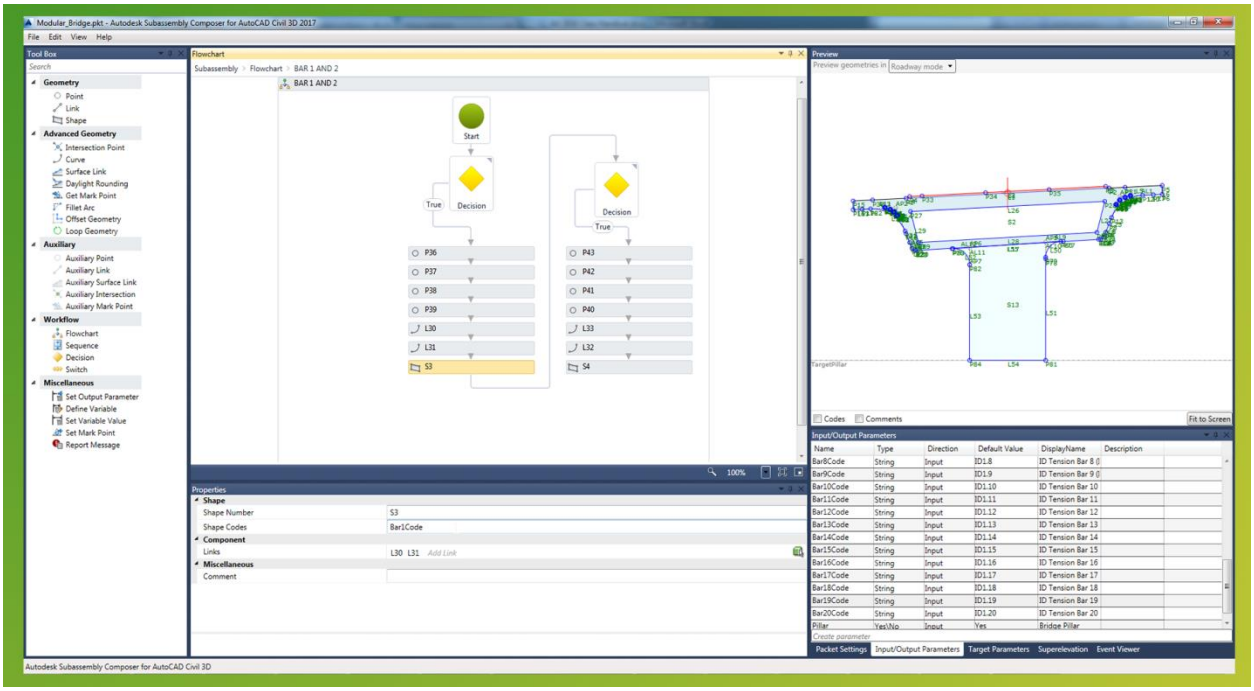
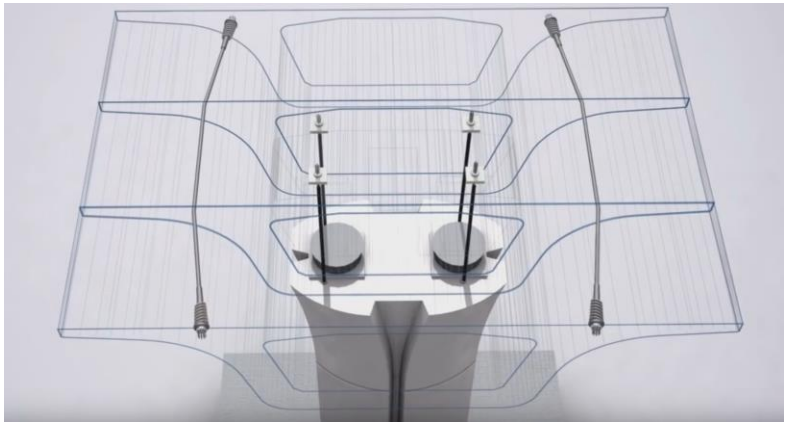


IMAGE 5: ID CODES OF REALTIME SENSORS ARE SET IN THE ASSEMBLIES OF THE DESIGN

The most important part is where we are actively putting the ID, and in this example we are putting these as shape codes in the subassembly.

In our project we are making use of tension meters to measure the tension in stressed parts that will hold our modular bridge parts together:



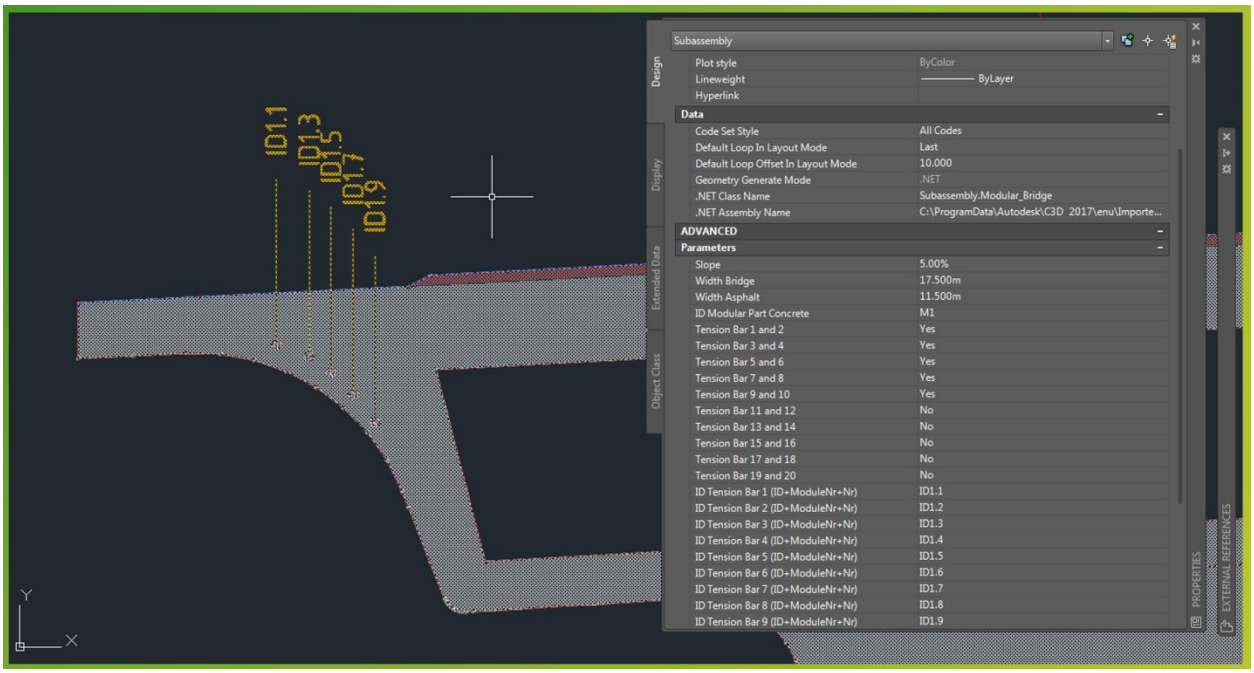


IMAGE 6: ID CODES OF REALTIME SENSORS ARE SET IN THE ASSEMBLIES OF THE DESIGN

After we have made a flexible subassembly we can start to make assemblies for each modular part in the bridge. In our example the bridge parts are prefabricated and thus the modular parts are already classified.

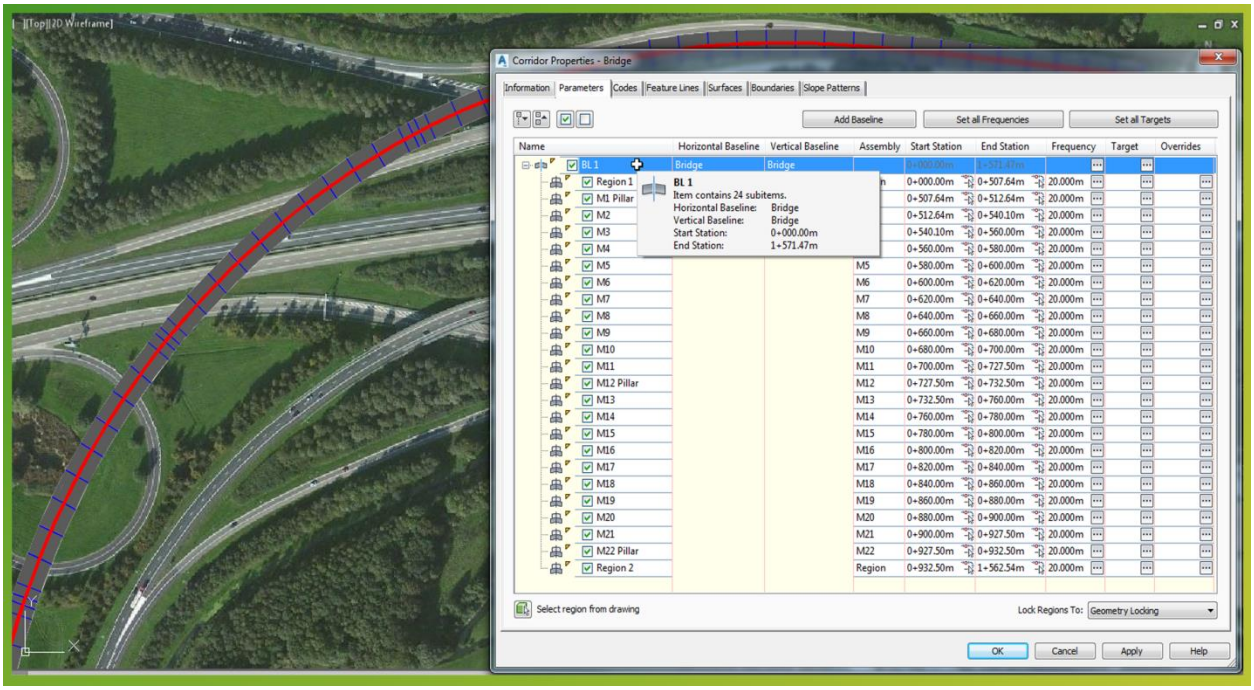


IMAGE 7: DIFFERENT MODULES HAVE DIFFERENT ID'S FOR SENSORS

In each of the modules there are parts that have a similar or different code for ID's in a way that corresponds with the sensors that have to be installed. After building the corridor we have to make corridor solids and link them to the corridor itself for updates.

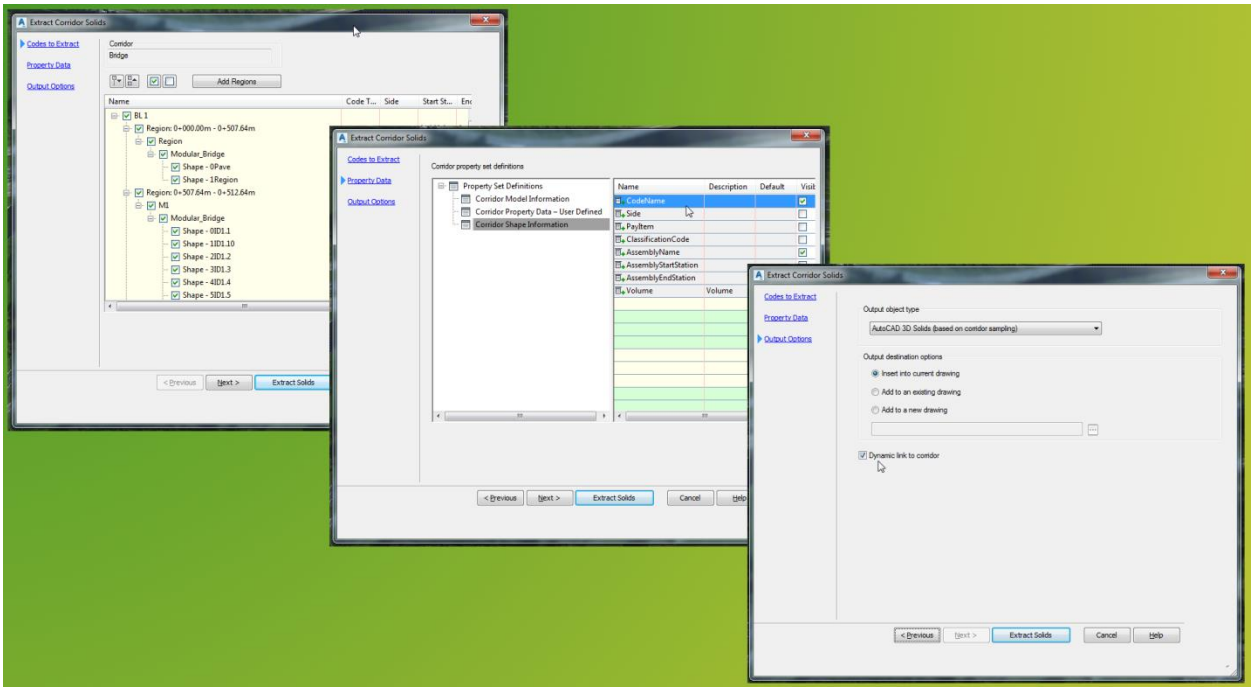


IMAGE 8: EXTRACTING CORRIDOR SOLIDS WITH PROPERTY DATA ATTACHED

After these steps, the result in Civil will be like this:

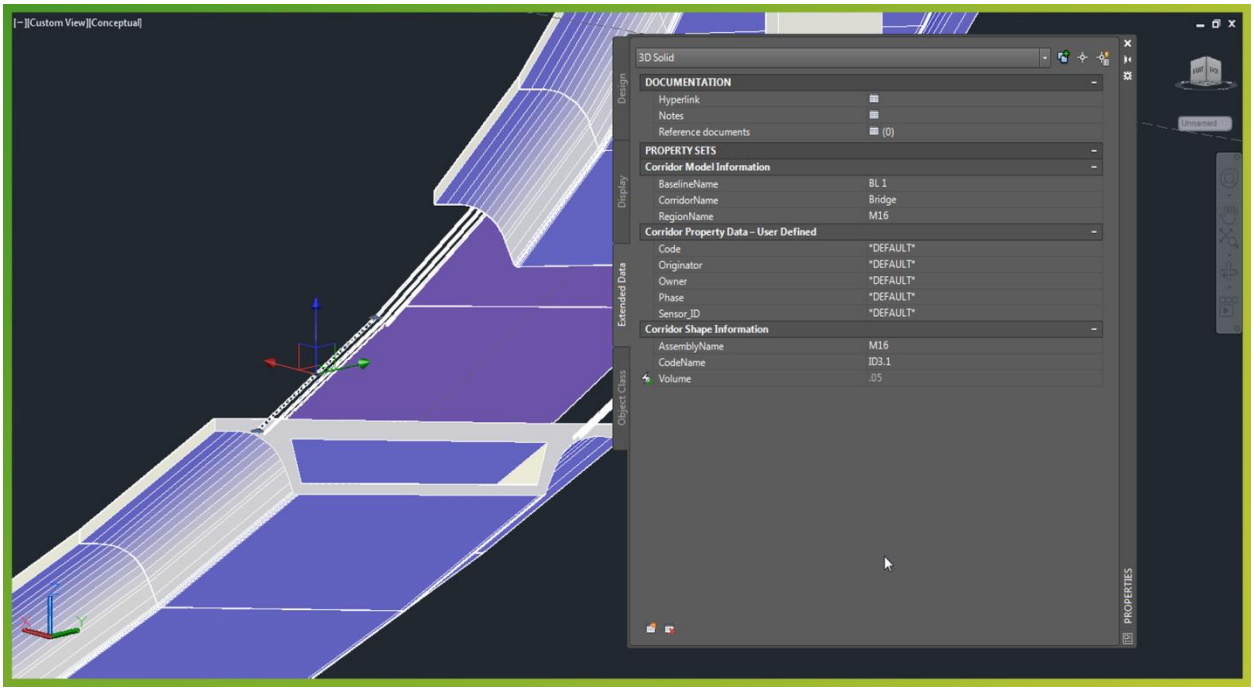


IMAGE 9: PROPERTY DATA

The next step is actually making the connection to Navisworks and external databases.

Learn how to use database tools in Navisworks to link real-time data from various sources (Beginning of Life – Middle of Life)

After the design creation in Civil3D we want to make the actual model in Naviswork. We do this by appending the desired design files into Naviswork through Autodesk Vault. After filling the Navisworks model we get a model that looks like this:

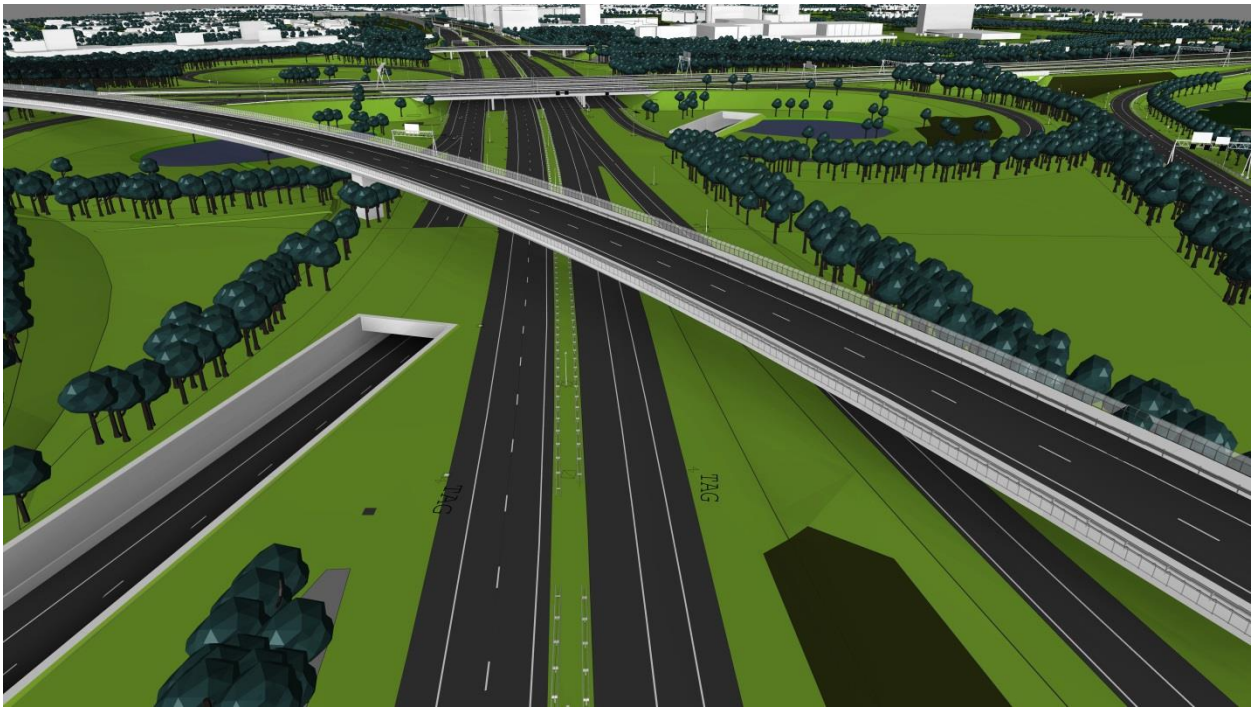


IMAGE 10: NAVISWORKS BIM MODEL

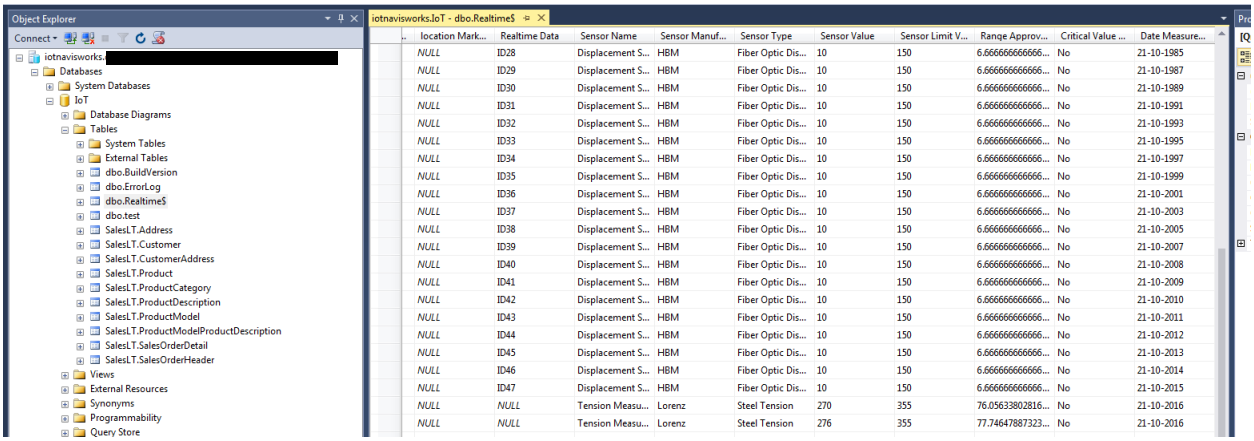
With all parameters and metadata in place from Civil3D we can now look at the connection to external databases. Here we have 2 options again: an Excel database and a SQL database filled from an IoT Hub. The principles are the same and to get to know this workflow, Excel is the preferred method. Once we know what to do with SQL strings we then can go to an IoT Hub, for instance a SQL server running on Linux for your Raspberry PI, or a total solution like Microsoft Azure.

The first step is to build your database and fill it with the data you want to show. In our Excel example the following data is added:

BIM Data Code	Corridor Model Information	Realtime Data	Sensor Name	Sensor Manufacturer	Sensor Type	Sensor Value	Sensor Limit Value	Range Approved Percentage	Critical Value Reached	Date Measurement
A-0000	Tunnel	ID31	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-1991
A-0000	Tunnel	ID32	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-1993
A-0000	Tunnel	ID33	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-1995
A-0000	Tunnel	ID34	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-1997
A-0000	Tunnel	ID35	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-1999
A-0000	Tunnel	ID36	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2001
A-0000	Tunnel	ID37	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2003
A-0000	Tunnel	ID38	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2005
A-0000	Tunnel	ID39	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2007
A-0000	Tunnel	ID40	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2008
A-0000	Tunnel	ID41	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2009
A-0000	Tunnel	ID42	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2010
A-0000	Tunnel	ID43	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2011
A-0000	Tunnel	ID44	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2012
A-0000	Tunnel	ID45	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2013
A-0000	Tunnel	ID46	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2014
A-0000	Tunnel	ID47	Displacement Sensor	HBM	Fiber Optic Displacement	10	150	7	No	21-10-2015
	Bridge		Tension Measurement	Lorenz	Steel Tension	270	355	76	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	276	355	78	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	278	355	78	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	284	355	80	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	289	355	81	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	294	355	83	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	298	355	84	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	286	355	81	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	289	355	81	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	292	355	82	No	21-10-2016
	Bridge		Tension Measurement	Lorenz	Steel Tension	294	355	83	No	21-10-2016

IMAGE 11: EXCEL DATABASE

In a SQL database it will look like this:



location Mark...	Realtime Data	Sensor Name	Sensor Manuf...	Sensor Type	Sensor Value	Sensor Limit V...	Range Approv...	Critical Value ...	Date Measure...
NULL	ID28	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1985
NULL	ID29	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1987
NULL	ID30	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1989
NULL	ID31	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1991
NULL	ID32	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1993
NULL	ID33	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1995
NULL	ID34	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1997
NULL	ID35	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-1999
NULL	ID36	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2001
NULL	ID37	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2003
NULL	ID38	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2005
NULL	ID39	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2007
NULL	ID40	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2008
NULL	ID41	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2009
NULL	ID42	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2010
NULL	ID43	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2011
NULL	ID44	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2012
NULL	ID45	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2013
NULL	ID46	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2014
NULL	ID47	Displacement S...	HBM	Fiber Optic Dis...	10	150	6.666666666666...	No	21-10-2015
NULL	NULL	Tension Measu...	Lorenz	Steel Tension	270	355	76.05633802816...	No	21-10-2016
NULL	NULL	Tension Measu...	Lorenz	Steel Tension	276	355	77.74647887323...	No	21-10-2016

IMAGE 12: SQL DATABASE

We are now going to match certain properties of the database with properties in the Civil3D files in Navisworks.

We build our subassemblies in a way so we can use the codes that are stated and these will be linked with the sensor ID codes that are in the databases For this we need an exact match.

If we look at for example the Tension Measurement Sensors, we can see we used codes ID1.1 and further. In Navisworks we will have to make sure the SQL selection strings will line up with our desired outcome:

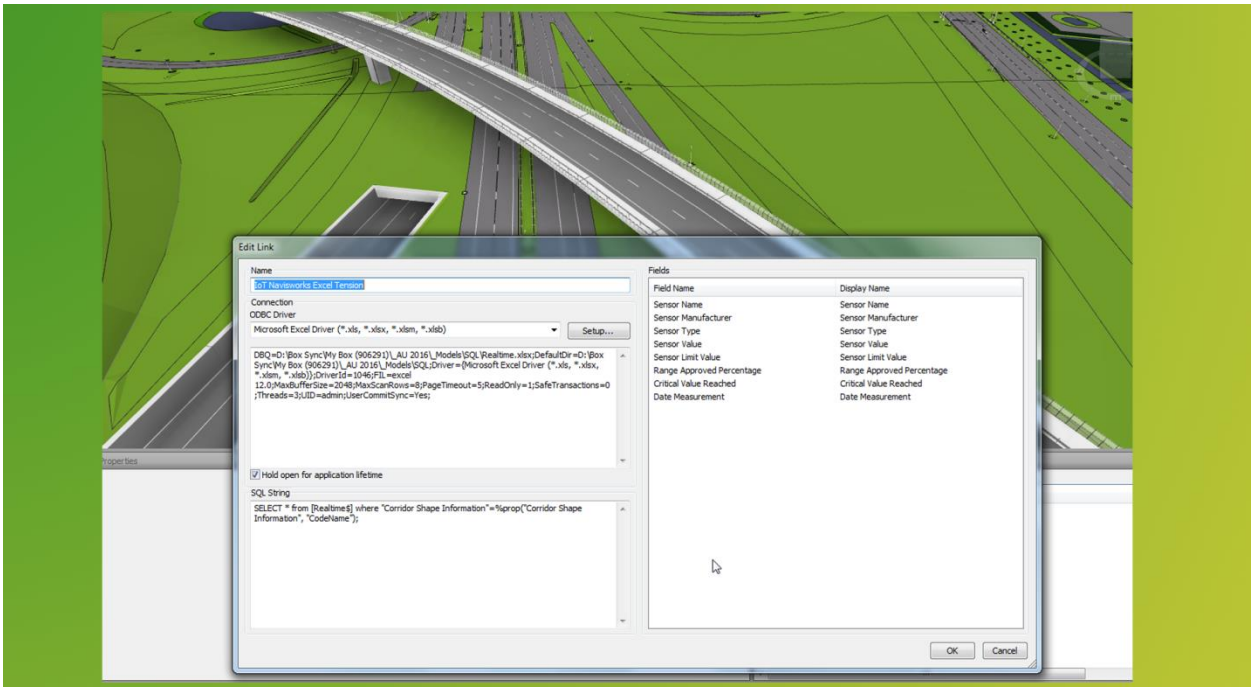


IMAGE 13: SQL EXCEL SELECTION STRINGS

If we want to connect to a SQL Server the connection strings are a little bit different:

Excel Connection String

SELECT * FROM [Realtime$] WHERE "Corridor Shape Information" = %prop("Corridor Shape Information", "CodeName");

SQL Server Connection String

SELECT * FROM Realtime WHERE "Corridor Shape Information" = %prop("Corridor Shape Information", "CodeName");

Obviously there are many ways to edit these connection strings to include/exclude or do other adjustments to your selections. In this case we are selecting all data in the database.

So now we have extra properties added from our Realtime sensor data sources:

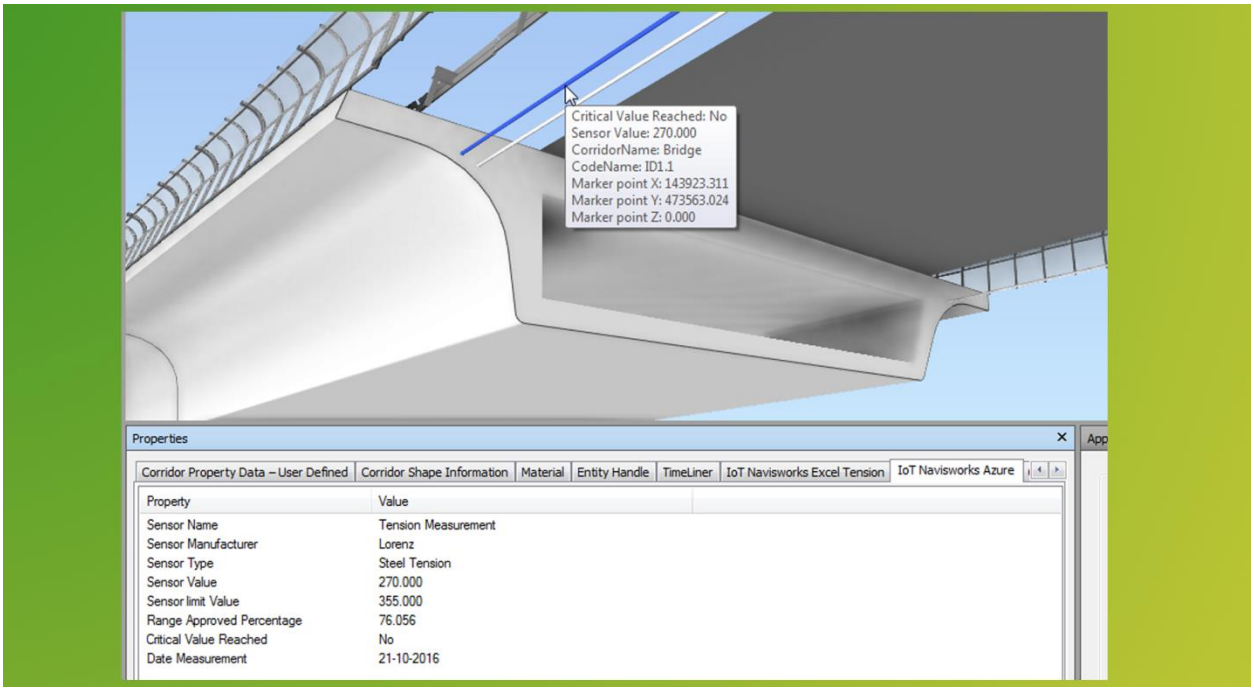


IMAGE 14: EXTRA PROPERTIES ON OBJECT BASIS

So our first goal from understanding why we want and need this data in our BIM models we have achieved. Add layers of extra information to objects to have more control over the built environment. With this in place we can monitor our data and start our forward and backward direction information loops.

Our second goal of having Realtime data in our BIM model is to visualize differences, problems or discrepancies in properties of objects we want to (or already have) build.

We can achieve this by making custom appearance profiles. Here we are using the strong search capabilities of Navisworks to link a visual style to a property that is generated by a Realtime sensor database.

First we have to make different search sets in Navisworks. For each value or sum of values you want to create a visual style of you need a different search set.

For our Tension Sensor data there are 3 search sets; critical value reached “yes”, critical value reached “no”, and to see whats happening with the steel inside the concrete we have a search set to hide the modular concrete parts.

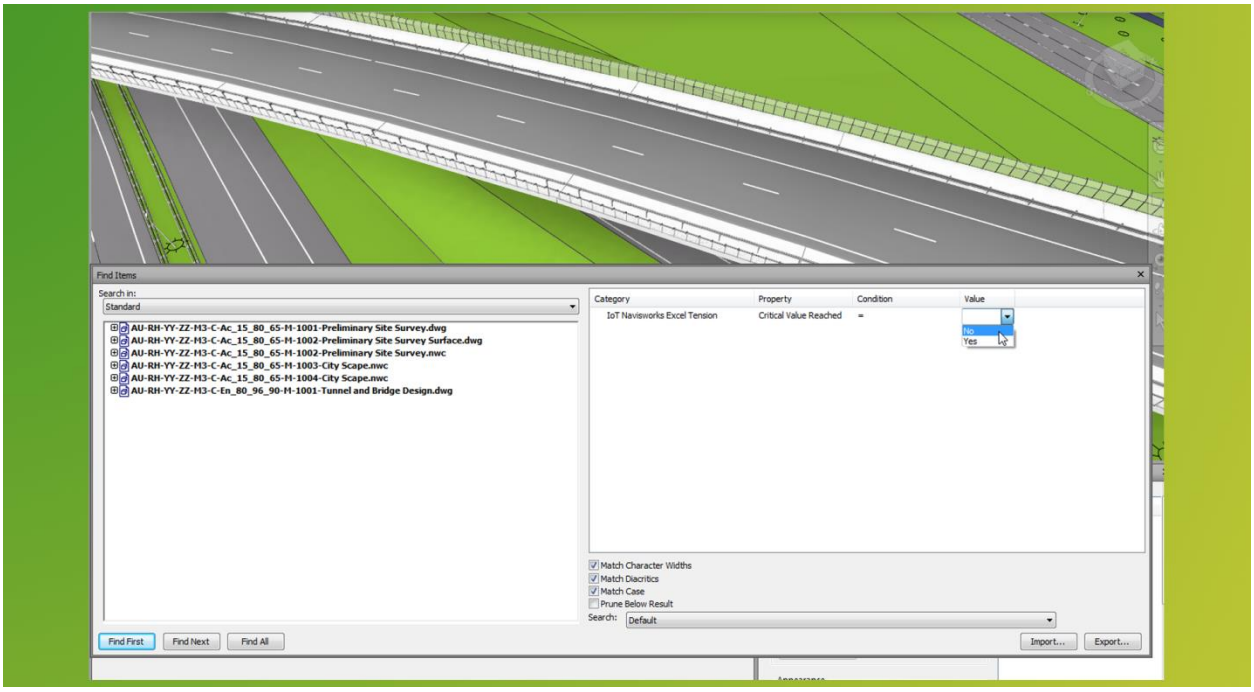


IMAGE 15: DEFINING SEARCH SETS

After the creation of the desired search sets we can start to create appearance profiles:

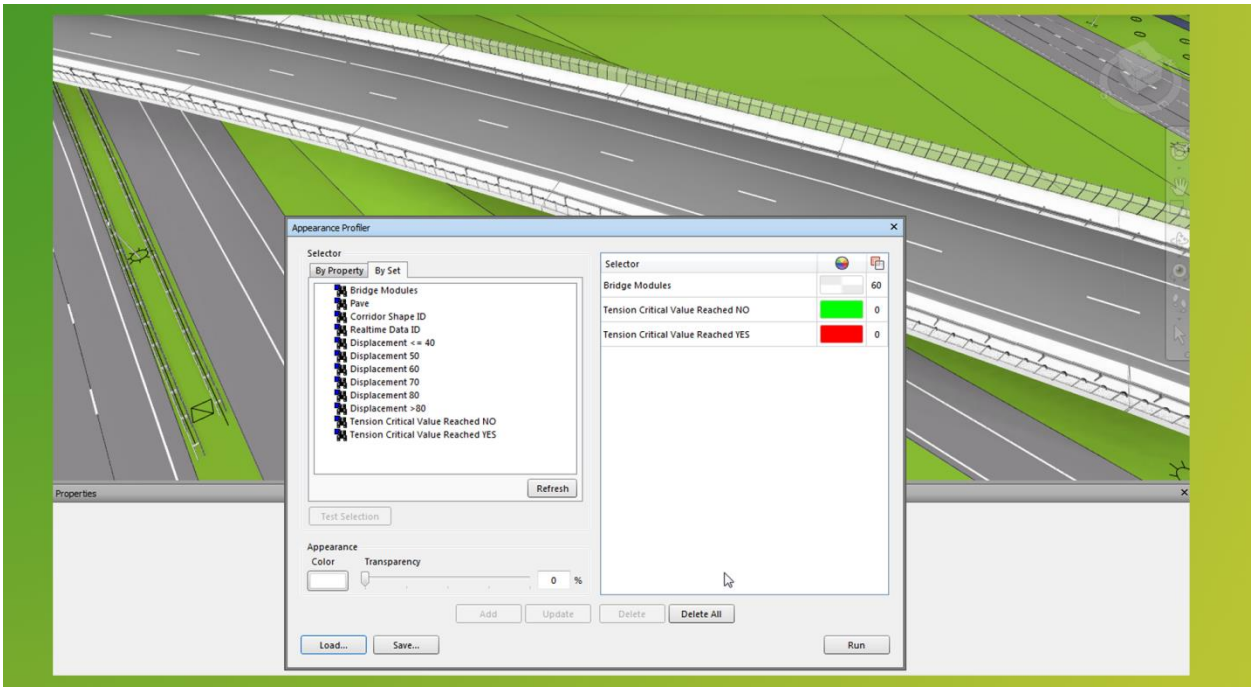


IMAGE 16: DEFINING SEARCH SETS

After we created our desired styles (go wild!) we only have to run these scripts and Navisworks will search for the object properties and will give it the desired visual look. In this project there are 2 examples of styles.

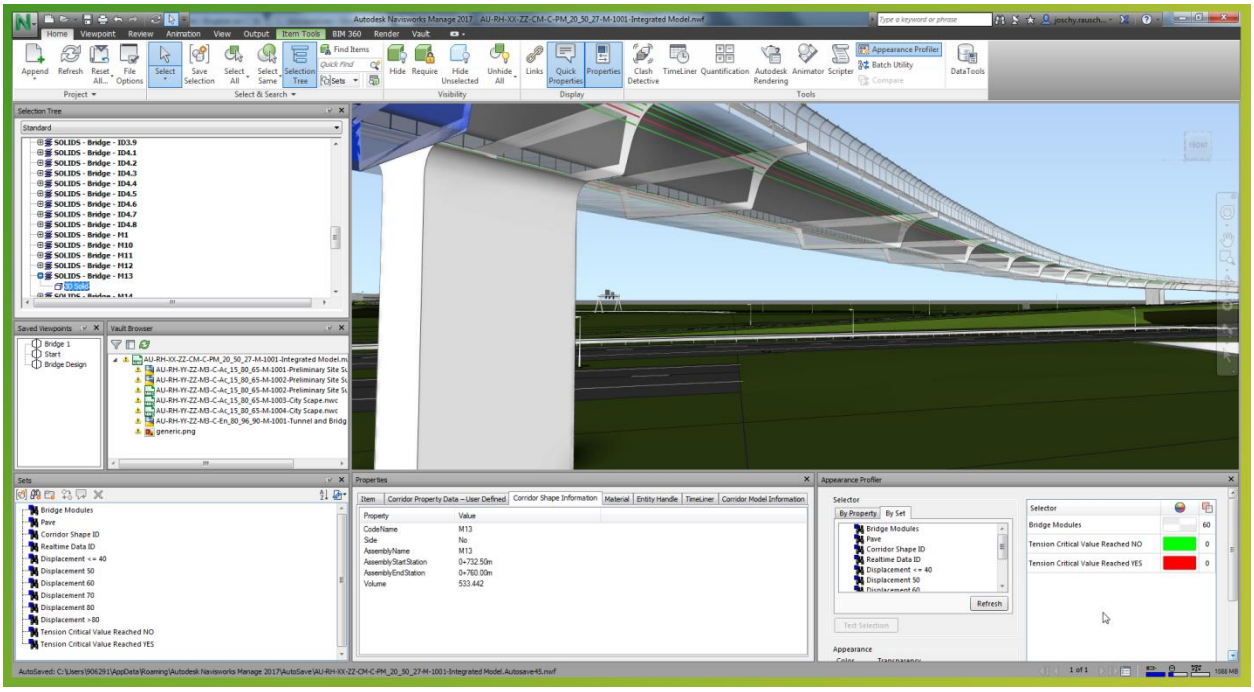


IMAGE 17: CRITICAL VALUES: YES: RED, NO: GREEN

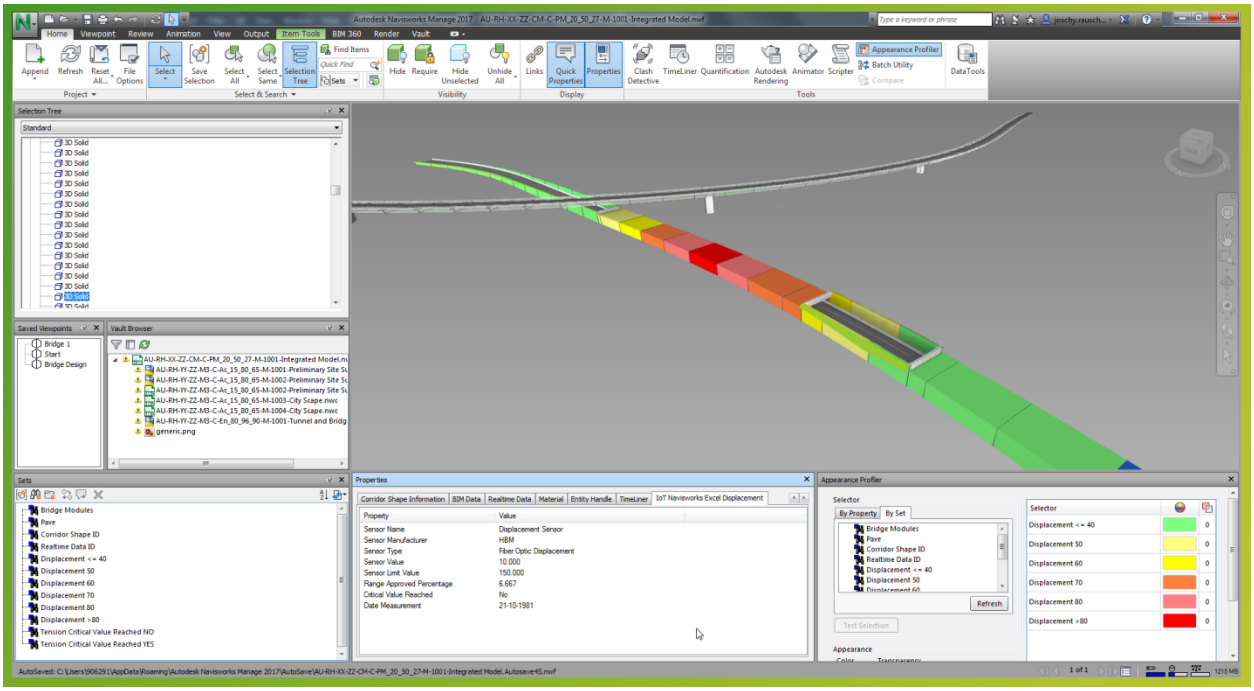


IMAGE 18: DISPLACEMENT VALUES, GREEN: ACCEPTABLE, RED: CRITICAL

Vault to rule all document management structure!

In the whole workflow we have seen, Vault is the backbone of everything. Here we ensure that all our data is stored, backed up, is going through design cycles (validation), communicated with designers, clients and stakeholders and above all is future proof. All of the shown models are built up with data from the Vault:

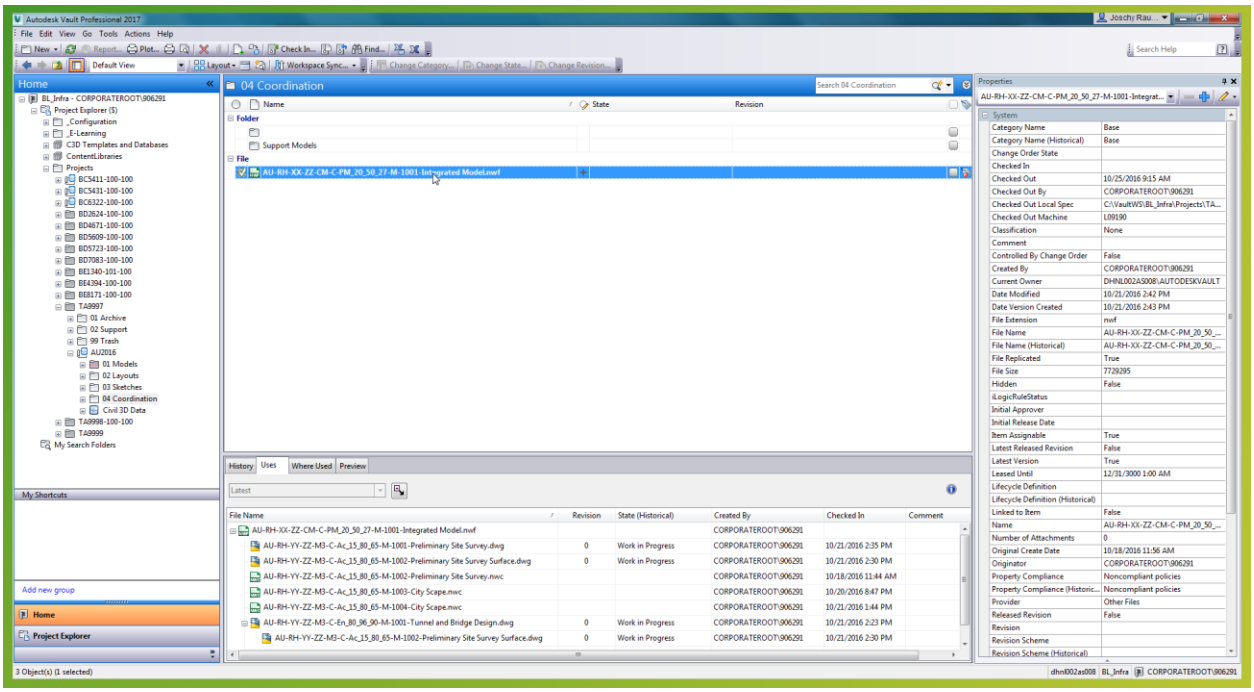


IMAGE 19: VAULT AS DMS