AUTODESK. UNIVERSITY

323997

# Digital Twin: Bringing MEP Models to Life

David A. Fink
Henning Larsen Architects

Alejandro Mata
Ramboll

## Learning Objectives

- Discover the benefits of digital twins for MEP design
- Discover the different types of IoT devices available for MEP systems
- Learn about bringing together MEP models and IoT sensors
- Learn how to visualize live IoT data with BI dashboards

## Description

The architecture, engineering, and construction (AEC) industry is heading toward Industry 4.0 and digital twins are a trend. As MEP design processes become more intelligent, digital twins are now feasible to reach and have many advantages ahead. These are powerful and can enable building owners to create smart buildings and operate with less. MEP models and point-cloud scans are more accessible toward more-affordable sensors. How can we use MEP models after project handover to enhance the operation of buildings along their lifecycles? This class will show the process of converting MEP models to digital twins using IoT devices—specifically, how to connect MEP models and sensors together by using cloud technology, Dynamo, and Revit 2020 software. Also, this class will show how to visualize IoT live data with BI dashboards to increase awareness of how people use buildings and to enhance the maintenance process of buildings. So, learn how to bring MEP models to life as digital twins, comply with client needs, and make the build environment more sustainable

## Speakers

David Fink

David Fink is an American architect living in Copenhagen for the past 18 years where he has worked in design build, architectural design and consulting engineering companies as a BIM Specialist. BIM is one of his passions and he is constantly looking for new ways of expanding the boundaries and use of BIM. He has been involved with BIM implementation and the active use of BIM on projects since 2007 both locally in Denmark and internationally. In his daily work he looks for ways of increasing the quality of projects, finding ways to increase efficiency, consistency and find ways to have some fun. He am part the generation of architects who has taken the ride from the Mayline through CAD to BIM which gives him a holistic view of the AECO industry. At Ramboll he is part of the Integrated Digital Solutions Group where he focuses on the architectural departments and how they work with other disciplines to deliver integrated solutions.

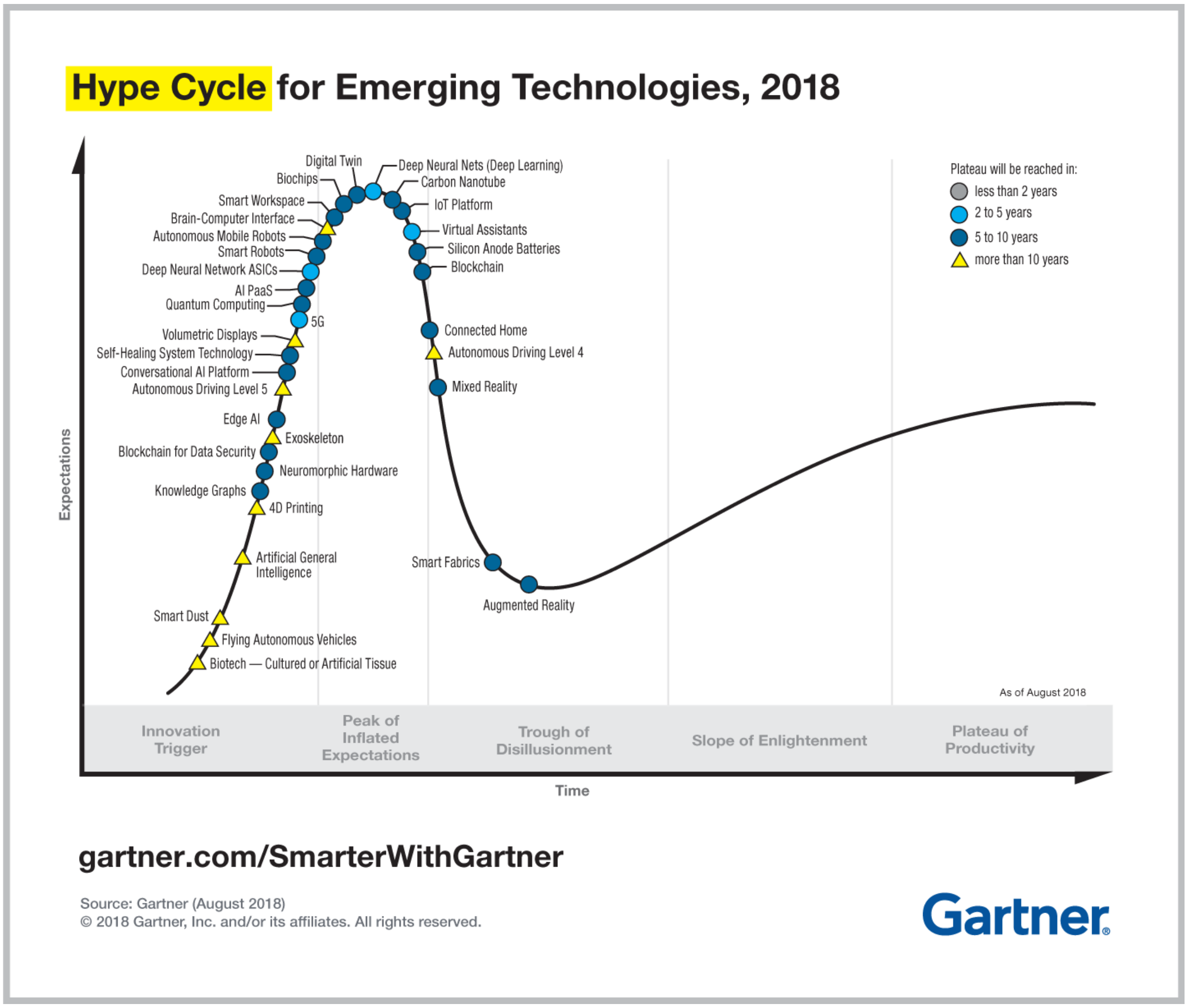
Alejandro Mata

Alejandro Mata is Automation Manager at Integrated Digital Solutions department in Ramboll Denmark. He is MSc. HVAC design engineer with a background in civil engineering and architectural technology from DTU-Technical University of Denmark. Alejandro is passionate about enhancing the performance of AEC industry by promoting a better utilization of building technology, towards automation of digital design processes. His focus is to work smarter and achieve the most effective practices to enhance data utilization and digital collaboration among AEC parties. He has been using has using Autodesk, Inc., products for the last 10 years, with a detailed focus in Revit MEP software, and the Dynamo extension complemented with Business Intelligence cloud solutions. He has gained experience in the last 5 years through well-known Nordic projects. Additionally, he has worked as a teaching assistant at DTU and loves sharing knowledge.

## What is Digital Twin?

Digital Twin is a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level.
-Michael Grievers

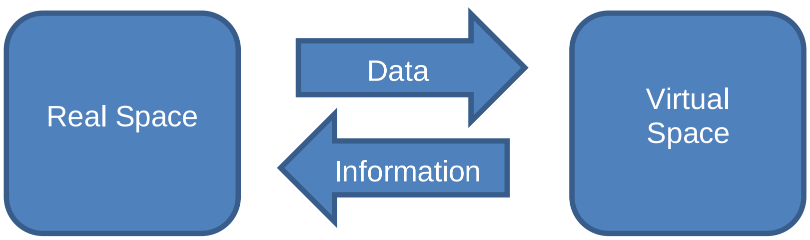
The AECO industry is slowly catching up with the aerospace and manufacturing industries when it comes to Digital Twins. This has been used extensively for prototyping new products, and verifying production line performance, and the performance of physical objects and systems. It has not been used widely within the AECO industry until very recently. With the increase of the available IoT sensors and the desire for more data from assets for verifying performance Digital Twins have become more common. Within the next few years there will be over 20 billion connected sensors and potentially billions of connected objects. The number of organizations using Digital Twins will triple by 2022. Gartner placed Digital twins at the top of their Hype Cycle for Emerging Technologies in 2018 with a forecasted peak in 5 to 10 years.



Historically, the only way to gain knowledge of buildings was to have direct physical contact with the building itself. All the data about the building and its performance was directly contained within the building. The data about the buildings has been stored in static documentation formats such as paper or computer files. Digital Twins build the bridge between the physical and digital worlds to allow for data to flow in real time or near real time, so the data becomes alive.

The Digital Twin concept was first noted in 2002 by Michael Grieves at the University of Michigan as part of Product Lifecycle Management (PLM). His idea was the Real Space and Virtual Space worlds would be linked throughout the lifecycle of the system. Initially it was known as a Mirrored Spaces Model. Later, it was known as a Mirrored Information Model before being known as Digital Twin.

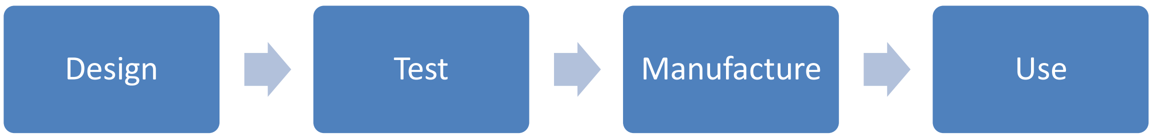
Digital Twin is described as the bi-directional flow of data between "Virtual Space" the digital representation and "Real Space" the physical asset. The data then needs to be accessible in real time or near real time to build a complete a digital picture of the physical asset.




DIGITAL TWIN RELATION BETWEEN PHYSICAL AND VIRTUAL WORLDS

Digital Twins started as basic CAD documentation of the physical world in the late 1900's, but with the growth of BIM working processes the concept of Digital Twins has become a representation much closer to reality. With the ability to assign parametric data to objects it was possible to make the representations move beyond just a physical description, but a functional representation as well. Recently, with the growth of Internet of Things (IoT) technologies it became possible to live stream data to the objects and systems in the physical world to a remote location, analyze the data, and react to modify conditions of the physical object in its actual location. This moves the CAD object from being just a 2D/3D representation randomly positioned in space to be a representation of the physical object that demonstrates not only the form of the physical object, but its behavior as well.

Digital Twins can be used for both prototyping objects as well as verifying and controlling physical objects. An object can be modelled in a pure digital environment with software and subjected to digital simulations to test the limits, and functional qualities of an object before it is produced. This is a huge cost and time savings. Previously, to conduct this simulation a precise physical representation would be created and subjected to physical tests which often resulted in the destruction of the objects, and the need to create a new modified representation of the object to continue testing. Now when prototyping the simulations can be conducted digitally and modifications made in real time without the need to produce any physical object before it is ready to be tested in the environment where it is intended to be used. When individual objects are assembled into a system the complexity increases and the ability to evaluate and understand the system decreases. Digital Twins assist in testing and evaluating the individual parts of complex system.




*DESIGN PROCESS*

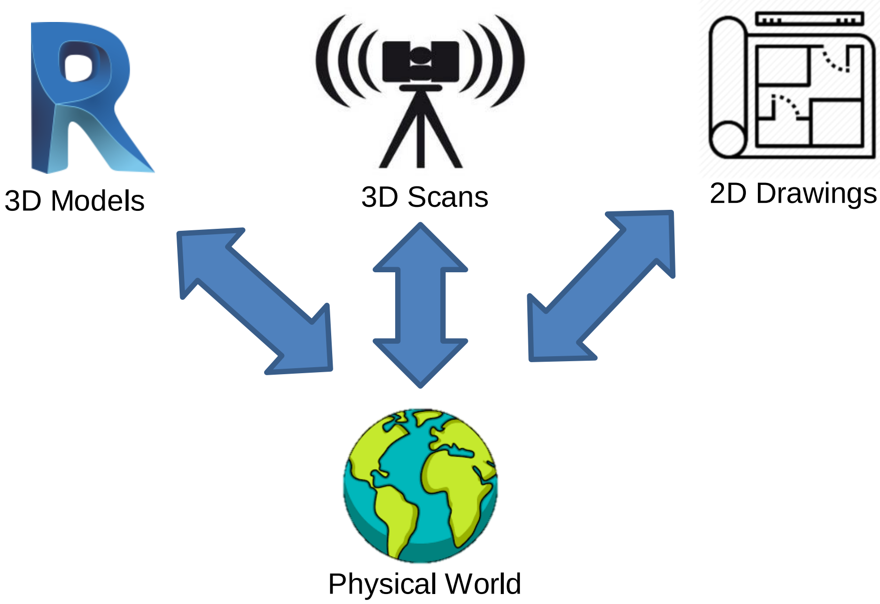
When discussing Digital Twin there are three distinct environments where they are active:

Digital Twin Prototype - This type of Digital Twin describes the prototype of a physical system. It contains the information sets necessary to describe and produce a physical version that duplicates or twins the physical version.

Digital Twin Instance - This type of Digital Twin describes a specific corresponding physical system that the prototypical Digital Twin remains linked to throughout the life of that specific physical system.

Digital Twin Environment - This is an integrated, multi-domain physical application space for operating on Digital Twins. It is used for a variety of purposes. There are two parts to this environment. The predictive, where the Digital Twin is used for predicting future behavior and performance of a system. This will test the system to verify that it will function within an acceptable range. The interrogative, where actual physical systems are for their current and historical performance. Multiple systems can be compared to find patterns where the system does not meet the design criteria so future systems can be modified accordingly.

Digital Twins can be made using 3D models, 3D Scan and even 2D documentation, the requirement to qualify as a digital twin is the link between the physical and virtual worlds where data is transmitted bidirectionally between the two worlds.



## Historic Digital Twins

Looking back to see where the Digital Twin concept came from, one can look to NASA in the 1960’s during the first trips to the moon.  NASA built exact replicas of everything that was launched into space. During production these replicas were prototypes of the actual objects, and after they were on their way to remote destinations, they became a twin of the equipment in use.  All modifications made by the astronauts on their way into space were also made to the twin.  This is probably best documented during the ill-fated Apollo 13 mission to the moon where there was a serious malfunction in the service module two days into their journey.  Before mission control sent instructions to the astronauts, simulations were made to what we could call an “Analog Twin” to simulate all decisions made before implementing them on the physical object thousands of kilometers away.  This is known as a “Mirrored System” since physical modifications were made to a twin physical object even though there were digital calculations made as well.

*NASA Mission Control Dashboard*

Mission control could be seen as a dashboard for the mission where all data was displayed and used to monitor and verify the status of the mission in real time.  With the technology available at the time the dashboard required the space of an auditorium.

Digital Twin is also used in Formula 1 racing.  While the team car is racing at 300km/h on the track there is a team of technicians and engineers sitting remotely in the pit monitoring all the stresses on the car in real time and making small modifications to ensure the car is performing at the highest level possible.  Before getting to the racetrack Digital Twins are used to simulate the performance of the car, so the best possible prototypes can be produced.  If a physical prototype is needed for testing before every race there would not be enough time to produce the best possible cars.  Digital Twins help save time and develop better products by reducing the design time required for each iteration of the cars.

VODAPHONE MCLAREN MERCEDES DASHBOARD

Adding sensors to a building is not difficult. The challenge is gathering the data, structuring and analyzing the data, so it is useful down stream without the need for more investment to make the data usable. One of the biggest challenges is getting the data to be shared across the multitude of systems that could use access to the data. Many data systems are closed, so getting access to the data and ensuring the structure of the data is usable can be difficult.

Unlike industrial asset centric businesses and discrete process manufacturing the data captured in buildings has an extra factor that can be unpredictable with the interaction of people. This makes the assets more dynamic, so it is no longer just predictive maintenance of the asset it becomes controlling a living entity.

Digital Twins are useful at every stage of a project's lifecycle. Currently, their focus is primarily during the operations and maintenance phases. There is lots of data available to be captured and the owner's interest is focused primarily on the ability to control the operations of their assets. The benefit of Digital Twin is the insight that is gained from the data that is harvested to be use proactively to improve performance and gain insight for the next project that could be designed.

## Creating a Digital Twin

When creating a Digital Twin, the system needs to be planned from the beginning rather than imposed on a project at a late stage. The data required, how the data is generated, how the data is received, how the data is stored, who has access to the data, and the types of digital models required must be planned. After the framework is in place, then the technology to be integrated can be selected for the physical asset to enable the capture of the real time flow of data.

The function of the Digital Twin needs to be defined. Will it be for just for monitoring an asset or will it control the systems in the asset as well? Will the data be used for advanced analytics to be used for predictive maintenance? The answers to these questions will drive the decisions needed to define the sensors to be used the way data is captured and the applications required for interpreting the data.

The implementation of the Digital Twin can start small with a single system and be expanded over time. It can start as a series of smaller system specific Digital Twins that are assembled to create the full picture of an asset. It is better to layer the data rather than to continually start new Digital Twins.

IoT plays a role in the creation of Digital Twins. IoT refers to unique identifiable objects and their virtual representations in an internet like structure. The transmission of data from sensors to storage devices is a critical connection. Without this connection the system cannot exist. It is necessary to select the correct network type, the correct protocol, and the correct transmission frequency in order to transmit the data.

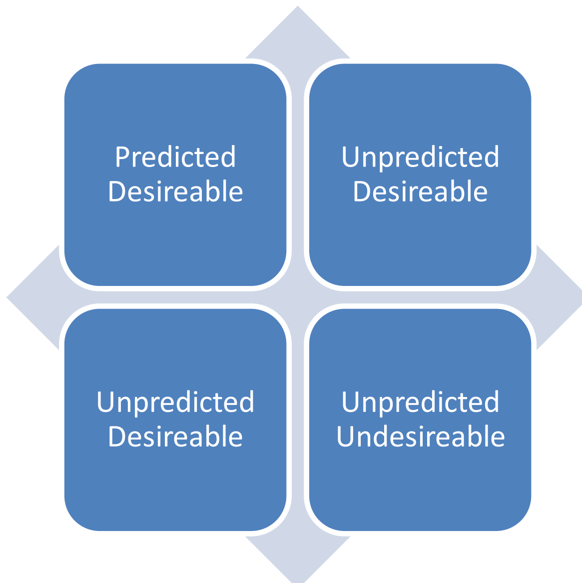
## Evaluating behavior

Digital Twins can help prevent serious accidents by the real time monitoring of the physical asset. By combining 3D scanning and sensors there is the opportunity to monitor existing assets where a digital model has not been created.

The Digital Twins are used for predicting future behavior and performance of physical systems. During the prototyping stage, the behavior prediction could be verifying the behavior of the designed system with associated components to verify that the as-designed system meets the proposed requirements.

The design phase provides the perfect opportunity to use Digital Twins by evaluating the virtual representation of the designed system. The behavior of systems can be verified virtually. The data structure of the model can be established early, so the data generated can be used downstream well into the construction and operation of the project. However, this requires that the information requirements are established before modelling, and the data is structured properly.

There are four possible outcomes when evaluating the behavior of a system. This is not only a measurement of the success or failure of the system, but a means to find faults and correct issues before creating the physical system.




CATEGORIES OF SYSTEM BEHAVIOR

Predicted Desirable – The system performs as predicted.

Unpredicted Desirable – The performance of the system results in unexpected surprises. This result offers new results that were not originally planned for. There are no detrimental effects from using the system as designed.

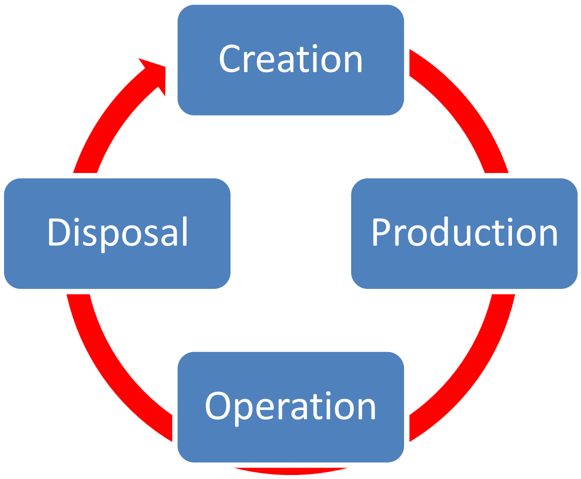
Predicted Undesirable – The system fails as predicted and will require modification. Still the system is performing as planned.

Unpredicted Undesirable – The system fails when not expected to fail. The system requires redesign. If not addressed this can result in possible catastrophic failures. Through simulations this outcome can be minimized. There is always a risk of this outcome in the physical system. If the outcome was never considered; therefore, it was never tested for.

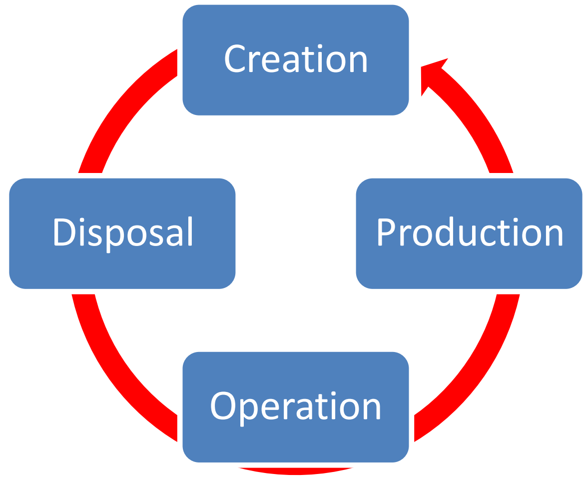
An Interrogative Digital Twin could apply to Digital Twin Instances that could be queried for their current and past histories irrespective of where their physical counterpart resided in the world. Individual instances could be interrogated for their current system state.

## System lifecycle

During the system lifecycle. There are two flows of data the first by the creation of the physical system where data flows forward as traditionally from creation to production to operation to disposal. However, data for a digital twin flows in reverse. Data from the future phase informs the previous stage. This data can be used to improve the performance of the systems by finding the weaknesses and failures that need refinement.




Physical Creation – Design Flow




Digital Twin – Data Flow

*VIRTUAL AND PHYSICAL DATA FLOW*

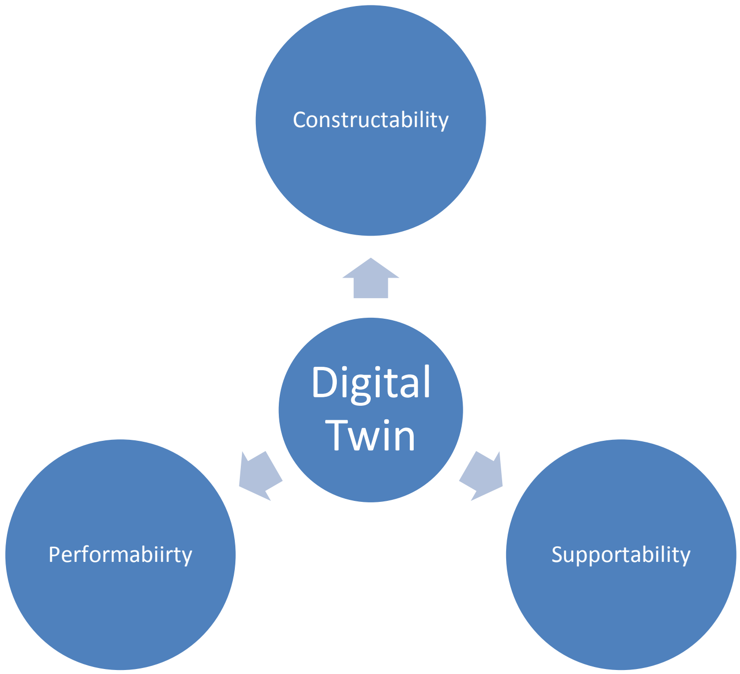
Creation – During this phase the characteristics and behavior of the system are defined. Desirable attributes are defined, and undesirable attributes are identified. Strategies to mitigate the undesirable attributes are developed to prevent them from occurring.

Production – The physical system is created. This is the phase where the manufacturability or constructability of the system is tested. At his point there is the possibility of undesirable behaviors to start appearing.

Operation – The physical system is tested. At this point all undesirable behaviors should be found and resolved. Although, there is still the possibility of unforeseen undesirable behaviors to be found.

Disposal - This is the decommissioning of the system. Decommissioning is typically ignored but, it does require consideration. The knowledge acquired through the previous stages is often lost through the decommissioning. The information generated during the disposal phase can be used towards the design of the next generation of the element.

Digital Twins are used to understand problems that are too complex for human understanding. Models and their associated data can be brought together to inexpensively check for conflicts and clashes so the physical model can be created more efficiently. Previously, the conversion of 2D documentation to a physical object was an inefficient iterative process. Now models can be created and simulated in a virtual system, so when the physical models are created it is primarily for final testing and verification. The destructive testing is conducted on the virtual models which has minimal time and cost implications. More testing can be conducted, and time and waste material is minimized.




*DIGITAL TWIN IMPLEMENTATION MODEL*

## The value of Digital Twin

The value of Digital Twins lies in the data and its connection from the physical system back to the virtual system. Large amounts of data can be generated which is used to inform design and operational decisions. It is important that the data created is reviewed and analyzed to gain greater understanding of the environment that will be affecting the physical system. Data replaces wasted physical materials, time, labor and energy over the lifecycle of the system. Data is never free to acquire. It will require resources such as planning, implementation, sensors, software storage and time. The cost of acquiring data is less than the cost of the physical waste operating an underperforming system. The greatest gains are made during the creation stage this reduces the amount of trial and error during the production phase. For MEP projects Digital Twins can improve the performance of systems, improve indoor environment, reduce energy consumption and reduce operation cost.

## Issues with Data

There are still security risks associated with Digital Twins. Typically, the data is stored in the cloud, so there is no physical infrastructure associated with the data storage side. However, there is a massive amount of data being collected from endpoints. Each of these endpoints are a potential point of weakness in the system. There is a possibility that data can be compromised between the endpoints and the cloud. Users of the data should have defined roles and it is best if the information transmitted is encrypted. The devices must have rights to send data over existing IT infrastructure.

Bad or misleading data can lead to errors. It must be ensured that the data is validated, and the data obtained can be trusted. It must be ensured that all sensors are sending data that is correct, calibrated, in the correct format and corresponds to the other sensors connected to the same system.

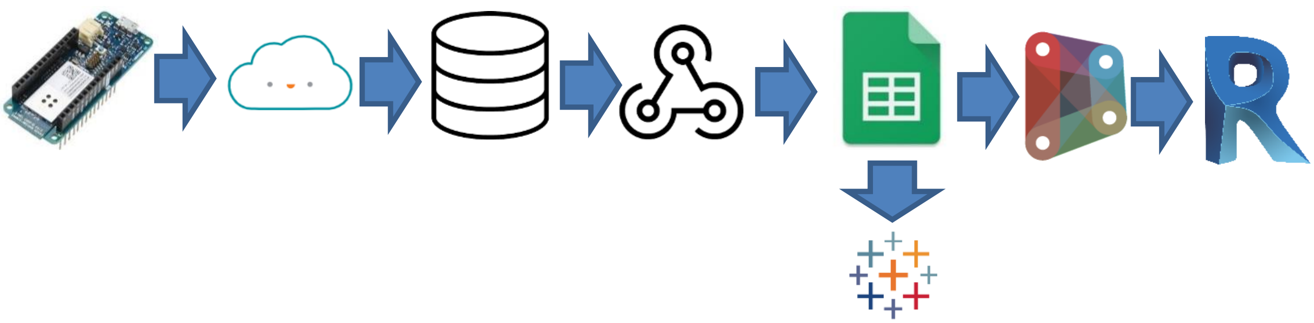
## Our Process

We were inspired by Project Dasher and the Autodesk bridge project at Pier 9. The idea of linking sensors and models became the goal. After a call with Kean Walmsley we quickly found out that we would need to change our strategy. Getting the project into Project Dasher would not be as easy as we originally had hoped. We also saw the cabling challenges that the bridge project exposed, so we wanted to go wireless. From here we embarked on our Digital Twin journey.

Our initial intension was to create a Digital Twin of our office. This is an Interrogative Digital Twin where an existing building is monitored to verify its performance. First, we investigated creating a Digital Twin of the canteen in our building and have a live dashboard on the company intranet which would show how much traffic was in the canteen at any given point in time. This would help find the best time to go to lunch. We thought this data would also be helpful for the staff in the canteen, so they could time the food in accordance with the number of people in the canteen at any time. We had an older model of the canteen and a Matter Port model, so there was our basis. We just needed data.

Initially, we investigated computer vision, but this was not possible due to the new European General Data Protection Regulation (GDPR). We had recently tested computer vision for other applications for cars and street signage, but it could not be used for this situation. We investigated using the Building Management System (BMS), but the system was too old. We could not access the data and the system could not record data. Even if we could get the BMS to work it would not be possible to get historical data. An acoustics engineer in the company recommended that we could use sound levels as an indicator of the number of people in the canteen. It would not be precise, but it could work and help us get around the GDPR problem. This would require a few sensors distributed around the canteen and some manual calibration.

Our project involved three processes, the first was documenting the physical environment. This involved documenting the physical environment with Revit models and laser scans. The second part was capturing live conditions from the physical environment with sensors and converting it into digital data. The final part was making the live connection between the physical and digital worlds. This connection was made by modelling a digital instance of our physical sensor and connecting the two, so data could flow from the physical sensor in the physical world to our digital representation of the sensor in the digital world.



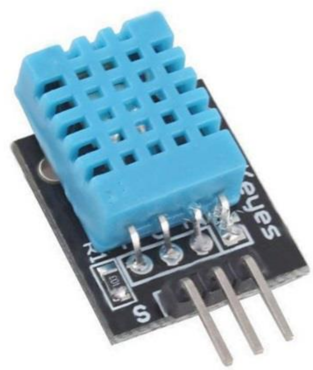
*THE PROCESS – THE BRIDGE FROM DEVICE TO MODEL*

## How to build a Sensor

We started to investigate off the shelf sensors, but quickly found that they will require a larger infrastructure than our budget allowed for or they were closed systems where we could not access the data, so we investigated creating our own. By creating our own sensor, we would have more control over the system. The system included what we were measuring, how the data was transmitted, where it was received, and how it was stored. The first challenge was we never had created a sensor before, but this was a minor detail. We started with a couple of Arduino MRK1000 microcomputers. This was an ideal starting point. The microcomputers have a built in WIFI module plus it is encrypted. There is a wide range of components that can be purchased to turn the microcomputer into a sensor. We embarked on leaning to create the sensor and code the Arduino. We found that the public library across from our office has a Hacklab, so we went to one of their meetings to learn how to start.

*ARDUINO MKR1000 MICROCOMPUTER*

We started with a TMP 36 Temperature sensor.  This sensor runs on voltage output so the results could swing a little.  We then upgraded to the DHT11.  This gave us more constant results plus we could measure relative humidity.



*TMP 36 TEMPERATURE SENSOR*     *DHT11 TEMPERATURE AND HUMIDITY SENSOR*

We then added a photoresistor to measure lighting levels.  This also is dependent on voltage output; the results vary a little but seemed to be within an acceptable range.

*PHOTORESISTOR LIGHT SENSOR*

Finally, we thought we would add a VMA 311 sound sensor to find out if we could measure occupancy levels with sound. This proved to be more challenging. The temperature, humidity and light sensors could read data periodically whereas the sound sensor needed a constant flow of data. This cause the sensor to crash so we decided to do without it.

*VMA 311 Sound Sensor*

The sensor was assembled on a breadboard and then we were ready to code.

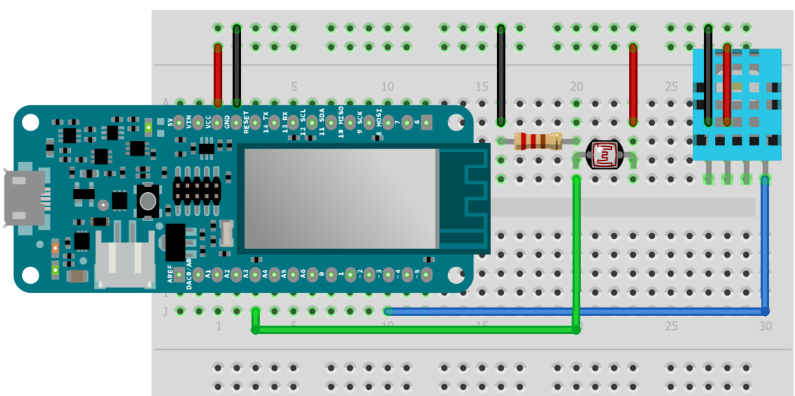
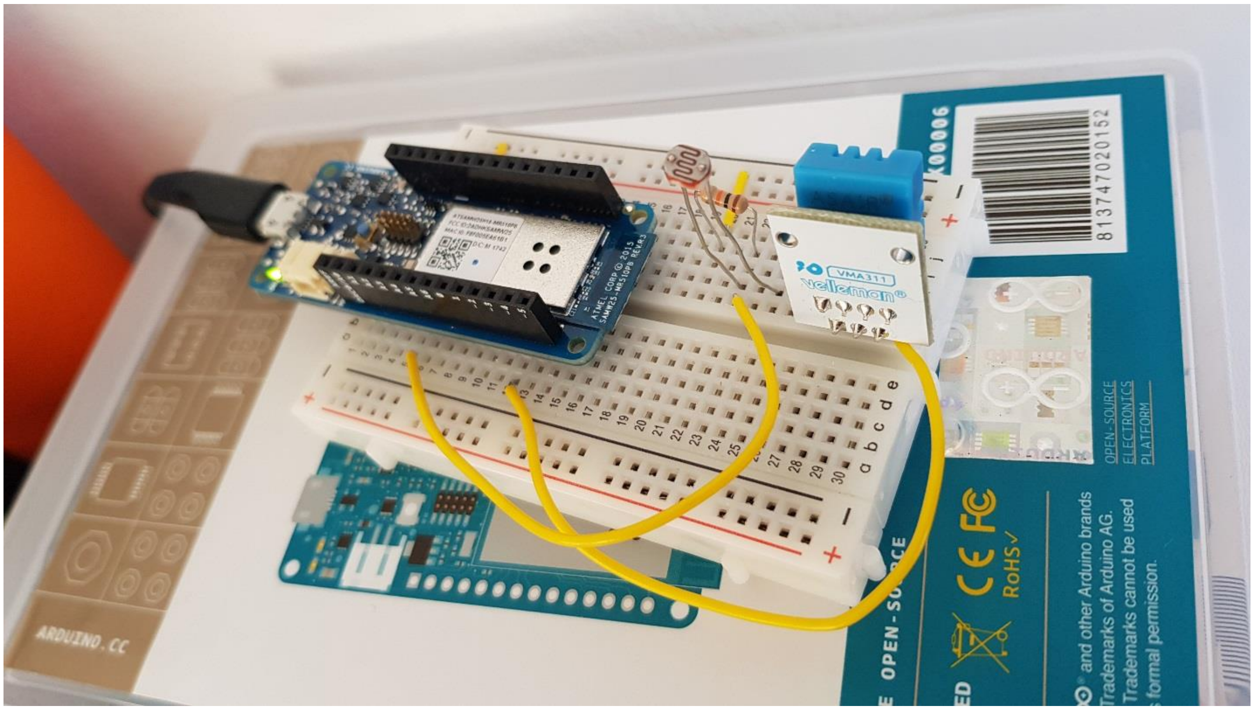


*Diagram of the Completed Sensor*

On a little side note. We found this little piece of software called Fritzing. It is like a Revit for circuits. You can model a circuit check that all the connections are made and get a bill of quantities out of it.




*THE FINISHED SENSOR*

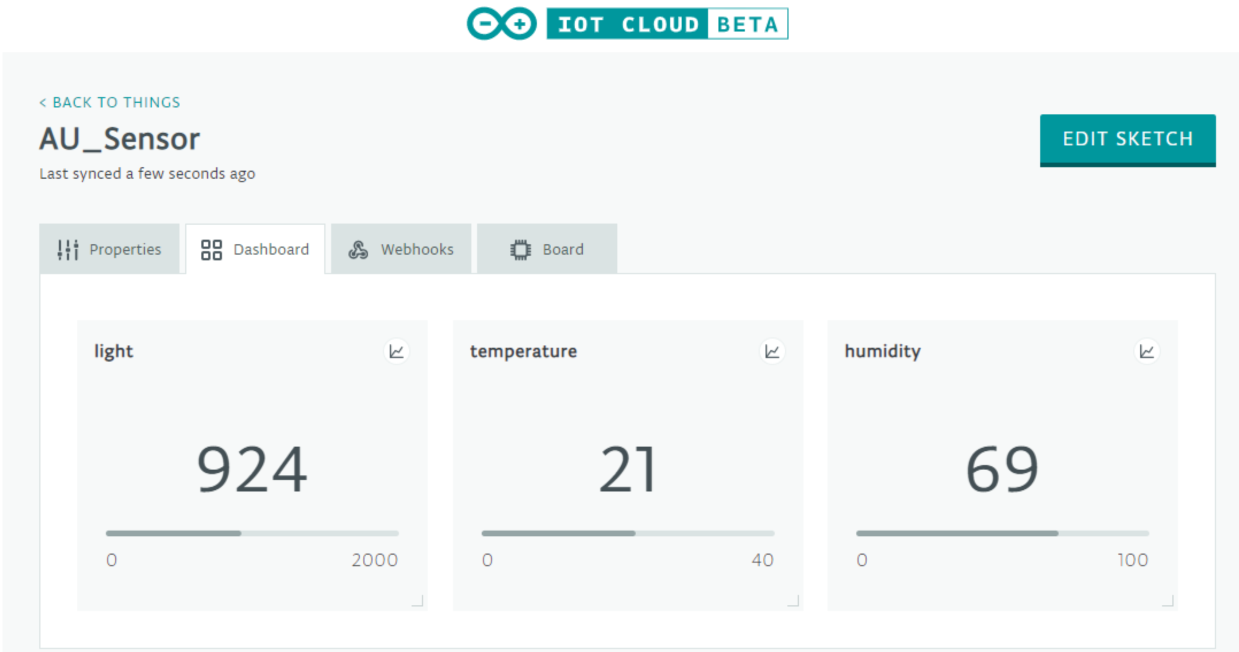
## How to Code the Sensor

The next step was to code the microcomputer to read the data from the sensors. We had never coded an Arduino before, so we were starting at the beginning. Fortunately, Arduino uses C/C++ so there were lots of resources online to figure this out. Finding code online was not too difficult and the Arduino Web Editor was a good place to start. The Web Editor worked directly with the Arduino IoT Cloud, so this became our way to move the data from the sensor into the cloud.

We coded and tested in the Web Editor and viewed the results in the monitor to ensure we were actually reading the data correctly. The variables to be sent to the cloud are set up in the Arduino IoT Cloud. These variables are then sent to the Web editor, so it was easy to get the Web Editor and the IoT cloud to work together. It was just important to set the variables to be sent to the cloud up in the code and make the association.

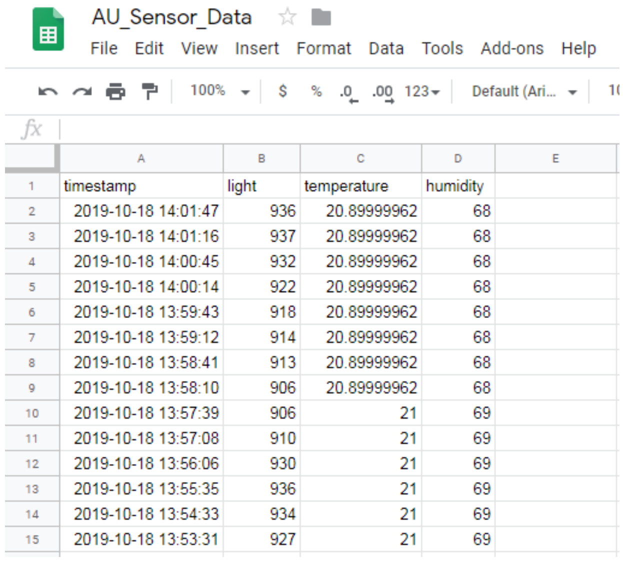

*ARDUINO CLOUD PROCESS*

In the IoT Cloud a dashboard is automatically created. The data is shown either as numerical data or as a graph. It is also possible to download a CSV file of the data, but this data is static. We were looking for live data in order to send it to our model.




*ARDUINO IOT CLOUD DASHBOARD*

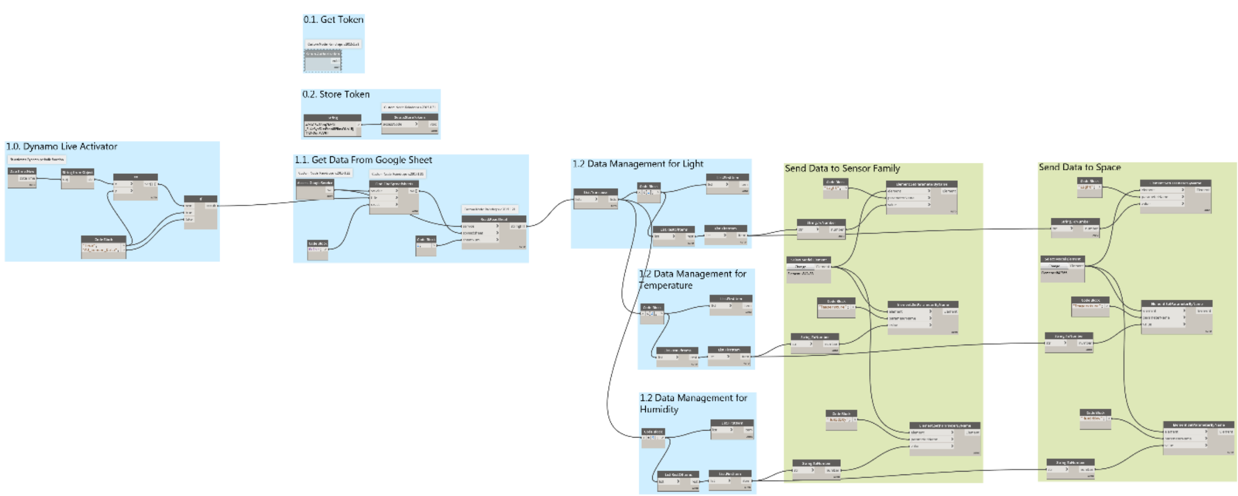
We then created a Google Sheet and connected to the Arduino IoT Cloud with a webhook. This allowed us to send data from the IoT Cloud to a Google Sheet with a few second delay. This also allowed for us to store the complete data set. We tried to store the data in our company database, but the there was too much delay in the data transmission. We used Google Sheets as an alternative to the database.



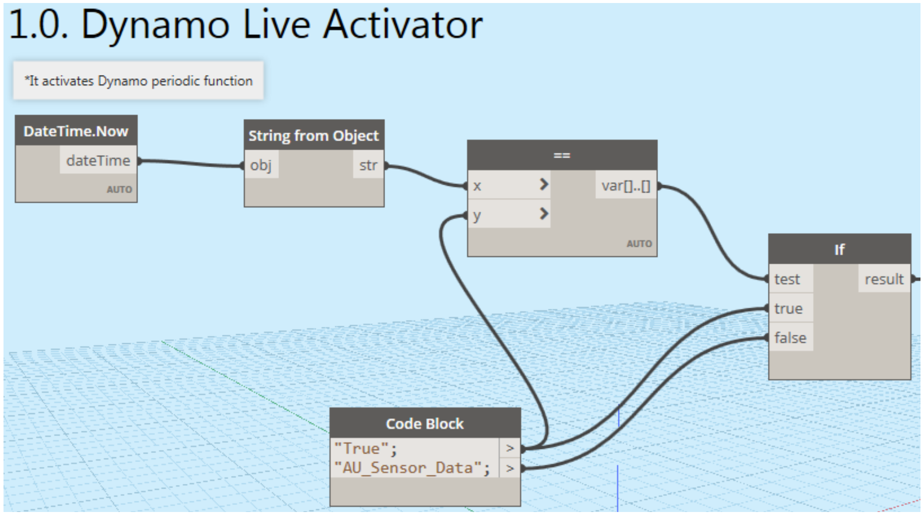

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | timestamp | light | temperature | humidity | |
| 2 | 2019-10-18 14:01:47 | 936 | 20.89999962 | 68 | |
| 3 | 2019-10-18 14:01:16 | 937 | 20.89999962 | 68 | |
| 4 | 2019-10-18 14:00:45 | 932 | 20.89999962 | 68 | |
| 5 | 2019-10-18 14:00:14 | 922 | 20.89999962 | 68 | |
| 6 | 2019-10-18 13:59:43 | 918 | 20.89999962 | 68 | |
| 7 | 2019-10-18 13:59:12 | 914 | 20.89999962 | 68 | |
| 8 | 2019-10-18 13:58:41 | 913 | 20.89999962 | 68 | |
| 9 | 2019-10-18 13:58:10 | 906 | 20.89999962 | 68 | |
| 10 | 2019-10-18 13:57:39 | 906 | 21 | 69 | |
| 11 | 2019-10-18 13:57:08 | 910 | 21 | 69 | |
| 12 | 2019-10-18 13:56:06 | 930 | 21 | 69 | |
| 13 | 2019-10-18 13:55:35 | 936 | 21 | 69 | |
| 14 | 2019-10-18 13:54:33 | 934 | 21 | 69 | |
| 15 | 2019-10-18 13:53:31 | 927 | 21 | 69 | |

*GOOGLE SHEETS DATA*

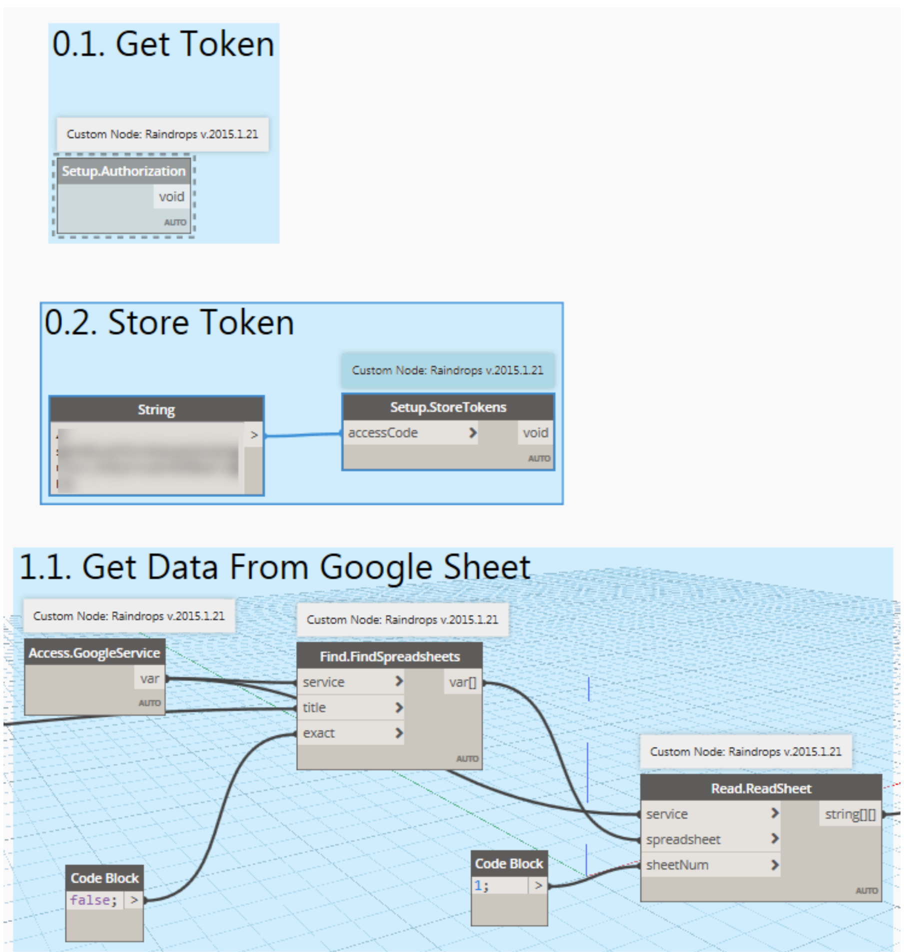
The next step was to connect the data stored online to the model. This was accomplished with a Dynamo graph using the Raindrops package. The next challenge was how to get Dynamo to read the data from the Google Sheet at regular increments. We added an If true loop which would use time and a true/ false trigger to activate the periodic run. This would allow Dynamo to read the new lines of data from the Google Sheet. We use this Dynamo graph to push data into our model with a couple of set value by parameter nodes.




*DYNAMO GRAPH*

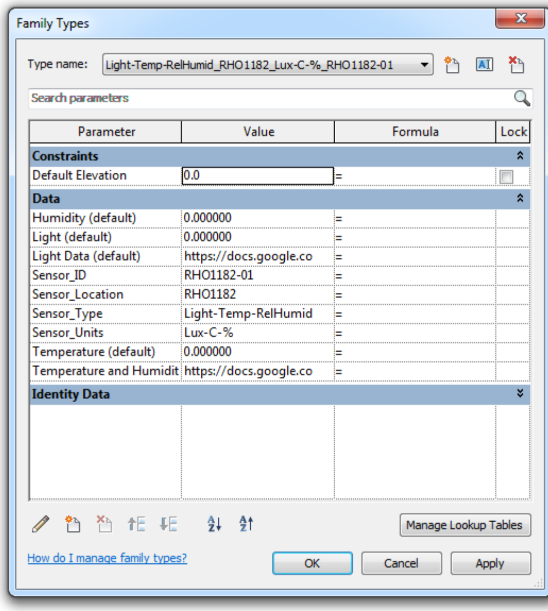


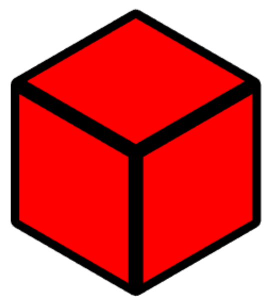

*LIVE ACTIVATOR FOR PERIODIC RUNS*



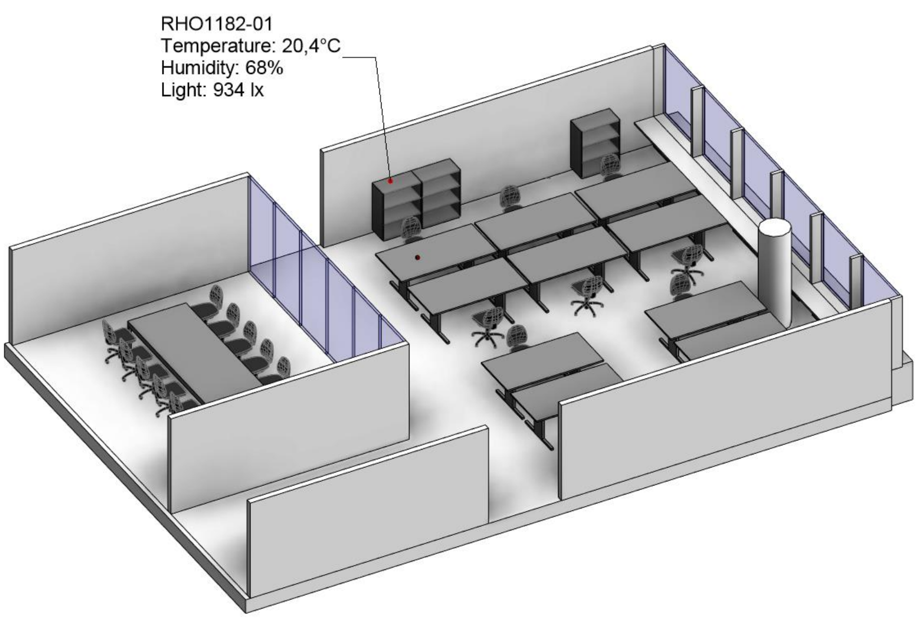

*RAINDROPS NODES FOR READING DATA FROM GOOGLE SHEETS*

In our model we have modeled the virtual sensor as a families and added instance parameters to receive the data from Google Sheets via Dynamo. We also added URL parameters in the family to link the data in the Google Sheets. This allowed for us to get access to the live data dashboards from the physical sensor via the sensor in the model.

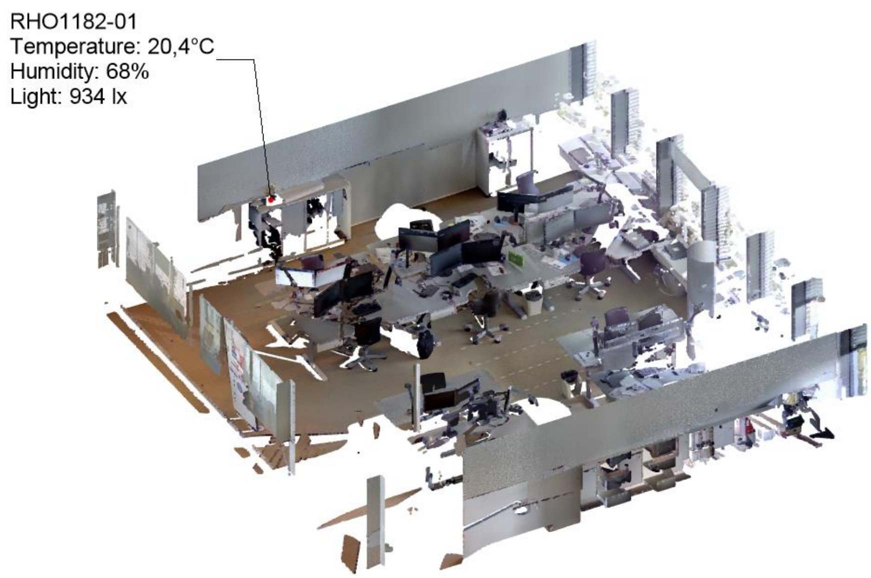


*SENSOR FAMILY*




*SAMPLE MODEL*




*3D Scan*

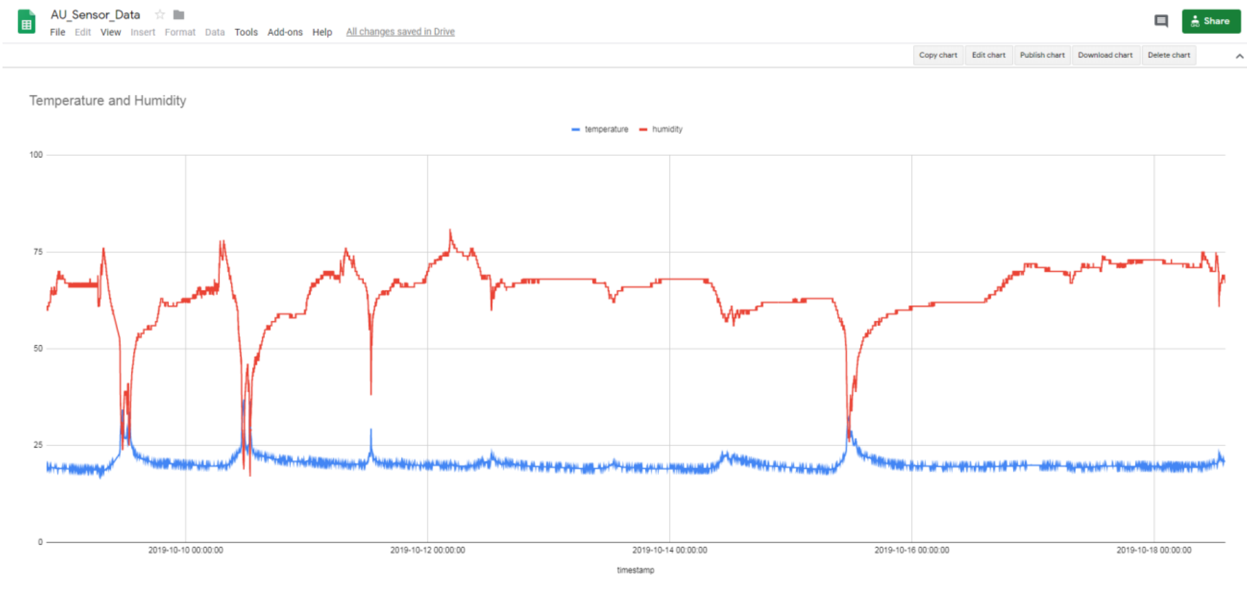
## Dashboarding the Data

To understand the data, dashboards can be used to visualize data. There are several tools that can be used to visualizing data all have advantages and disadvantages. For this example, we investigated Google Sheets, Tableau and Power BI.

Google Sheets and Power BI were advantageous for us since Google Sheets is free and we have licenses for Power BI. Google sheets worked best for reading data. The live data could be visualized in real time like we saw with Project Dasher. However, the graphics were not particularly interesting and there was not the possibility for deeper analysis.

Power BI which we have used extensively, could not read the live data from the Google Sheets. It would require an export to Excel from Google Sheets every time we needed to get the data into Power BI. This was not practical for our purpose. We previously have had good experience with visualizing data in Power BI, but it was not optimal for this purpose.

Tableau was a good alternative to Power BI. Data from Google Sheets could be directly loaded into Tableau. There were many of the visualization possibilities that could be found in Power BI. The data was not live like in Google Sheets, but the data would be loaded into Tableau when the dashboard is opened. It wasn't quite Project Dasher but it would work for our purpose.




*GOOGLE SHEETS TEMPERATURE AND HUMIDITY DATA*

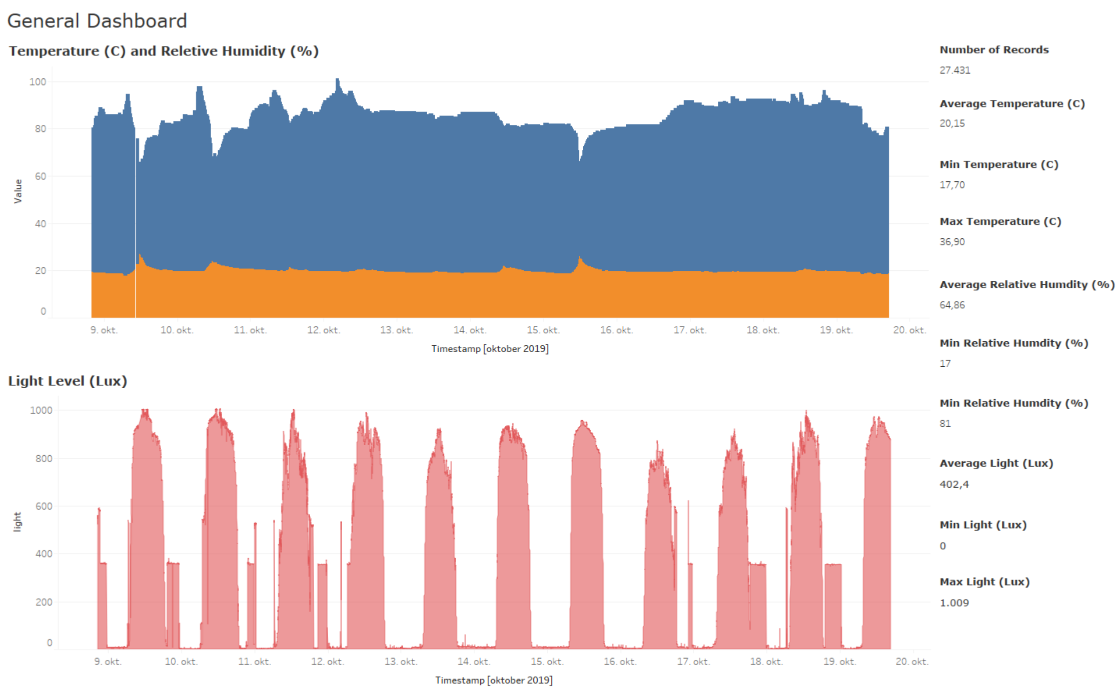



*TABLEAU TEMPERATURE AND HUMIDITY DATA*

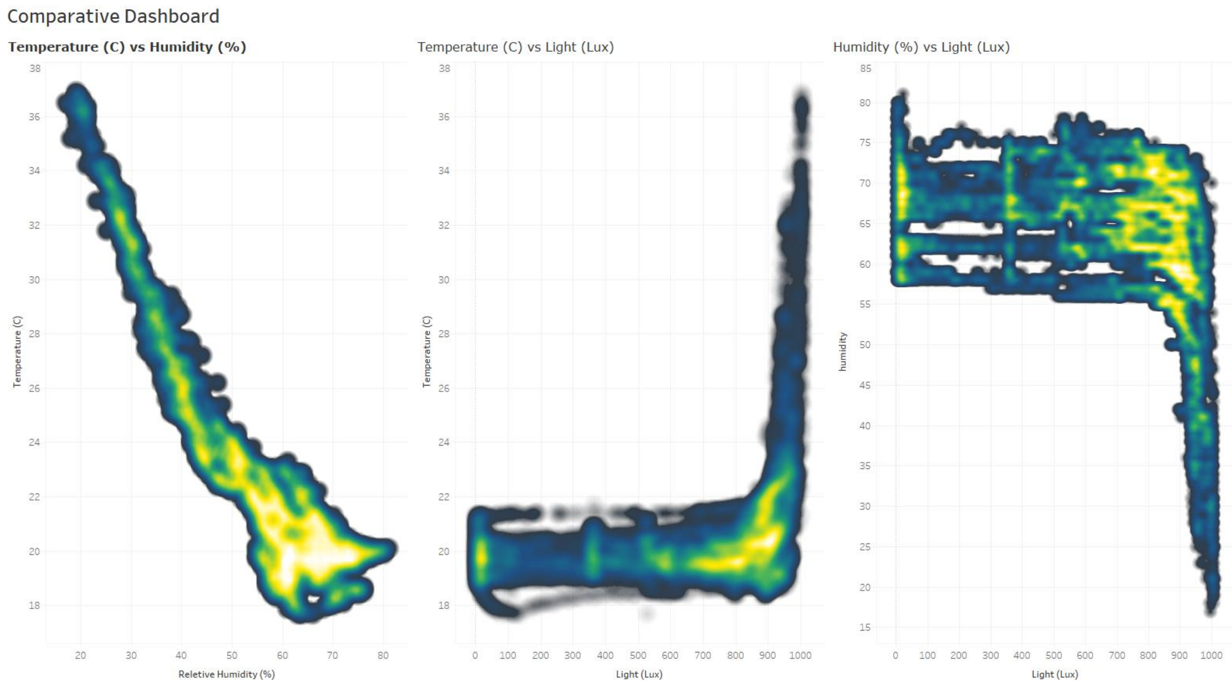
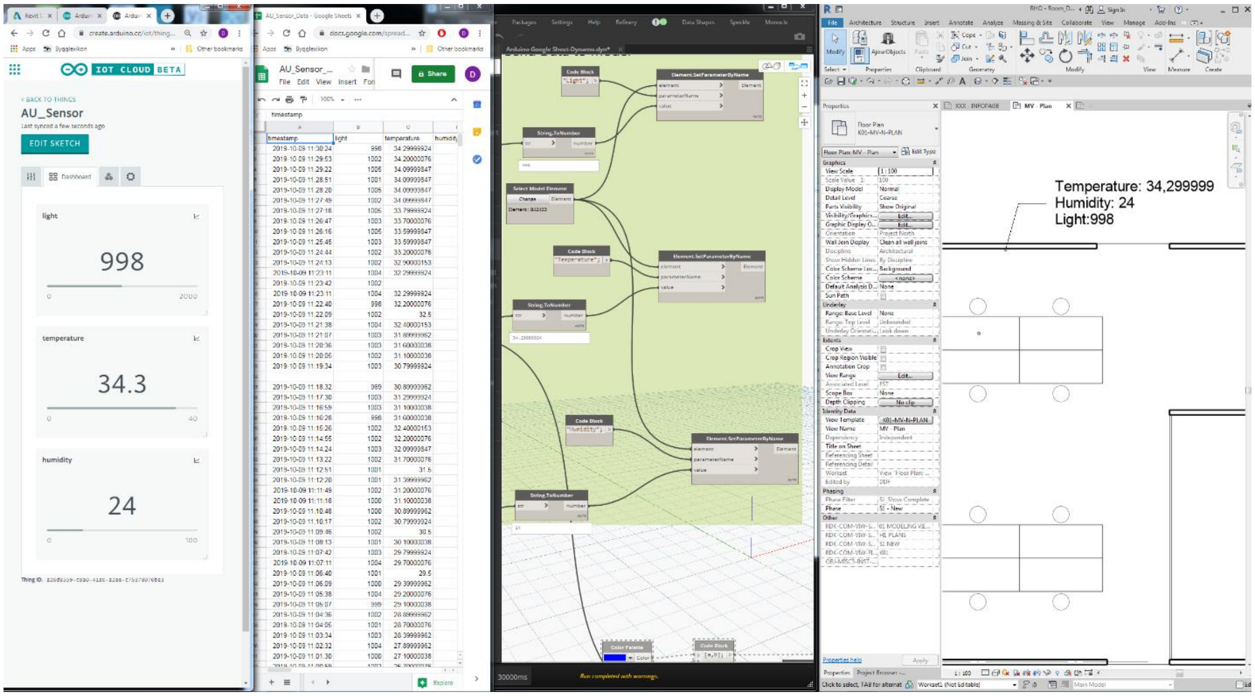



*TABLEAU TEMPERATURE, HUMIDITY AND LIGHT DATA ANALYSIS*

## Connecting the Sensor to the Model

Through this process we could connect the sensor to the model. Here you can see the flow from the sensor to Arduino IoT Cloud to Google Sheets with our webhook through Dynamo and into Revit.




Arduino IoT Google Sheet Dynamo Revit

*DATA FLOW*

## Results

In this case “Digital Twin Instance” has been used to verify the physical environment. The results from our Digital Twin example show the indoor climate would fall into the “Unpredicted Undesirable” behavior as described above. The humidity in the space measured to be too high on a regular basis. This compared to the design intention of the space. This would be noted as an unexpected failure of the system to function as designed.

## Conclusion

Digital Twins can make between 3D models, 3D scans or 2D drawings and the physical world. It is required that there is a relationship made with data between the two worlds. With this process we have demonstrated using Arduino microcomputers and readily available software it is possible for Digital Twins to be created by anyone. To get the virtual and physical environments to communicate requires extra insight into the data transmitted from the IoT devices and the data structure of the model where the data is received.

When we start to link the data from multiple building sensors together there is the possibility to gain insight to how the building is performing.  KPI’s can be established to verify if the building is performing as designed.

The process that we used gave us good insight to whet is required to establish a Digital Twin. The solution we have developed works, but the scalability of the system is limited.  When in a situation where there is no historical data this is a “simple” way to build up your own Digital Twin. It would require that you have the sensors or can build your own and there is an infrastructure ready to transmit and receive data between the sensor and the database where the data is stored.  The other benefit of this approach is the sensors can be customizable to the needs of the data required to create the Digital Twin.