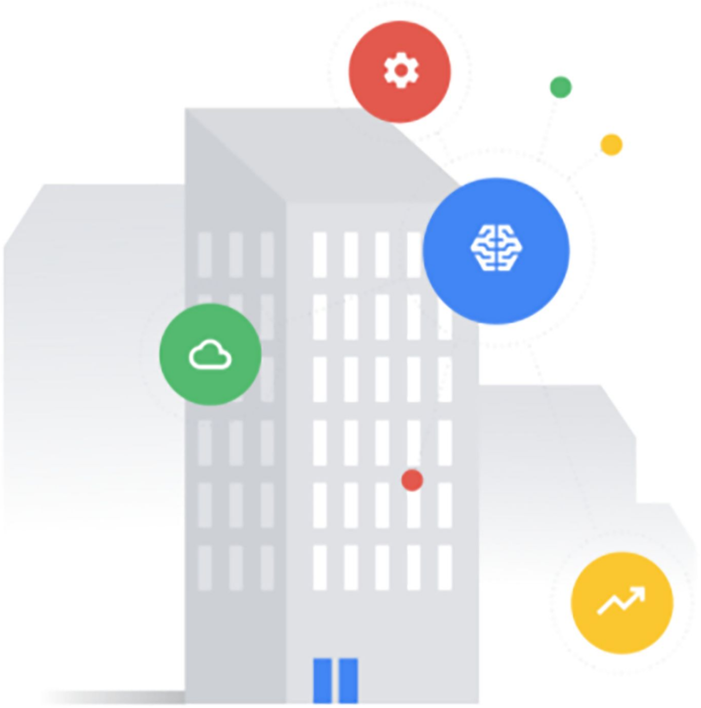
Google

Module 1 | **Lesson 1**

# Digital Buildings Ontology (DBO)



# Before you get started

This learning module has interactive features and activities that enable a self-guided learning experience.
To help you get started, here are two tips for viewing and navigating through the content.

## 1 View this content outside of GitHub.

- For the best learning experience, you're encouraged to download a copy so links and other interactive features will be enabled.
- To download a copy of this lesson, click **Download** in the top-right corner of this content block.
- After downloading, open the file in your preferred PDF reader application.

## 2 Navigate by clicking the buttons and links.

- For the best learning experience, using your keyboard or mouse wheel to navigate is discouraged. However, this is your only option if you're viewing from GitHub.
- If you're viewing this content outside of GitHub:
  - Click the **Back** or **Next** buttons to go backward or forward in the deck. Moving forward, you'll find them in the bottom corners of every slide.
  - Click blue text to go to another slide in this deck or open a new page in your browser.

**Ready to get started?** Let's go!

Google

# Warm-up

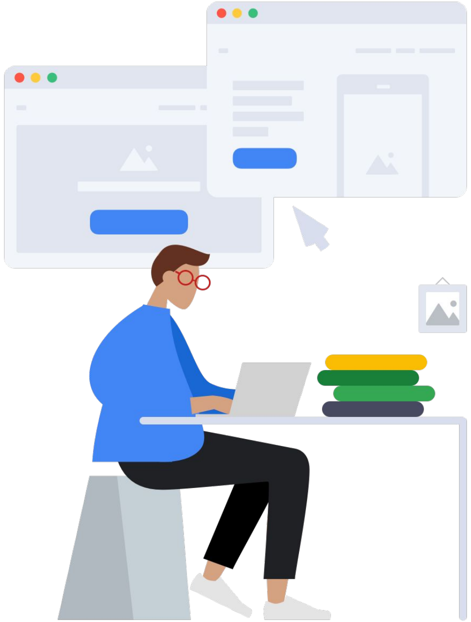
## First, let’s review the basics.

**Do you recall these key terms related to data modeling?**

- Data model
- Ontology

**What do these terms mean?**

Before diving into the first lesson, we’ll review them since they’ll be heavily referenced in this module.



Back

*Click **Next** when you’re ready to review these terms.*

Next

Google

# Warm-up: Key terms

## Data model

A digital representation of a physical system that communicates connections between data points and structures.

You can think of a data model as a relationship graph, like the one on the right. The nodes represent entities and the lines connecting them represent relationships.

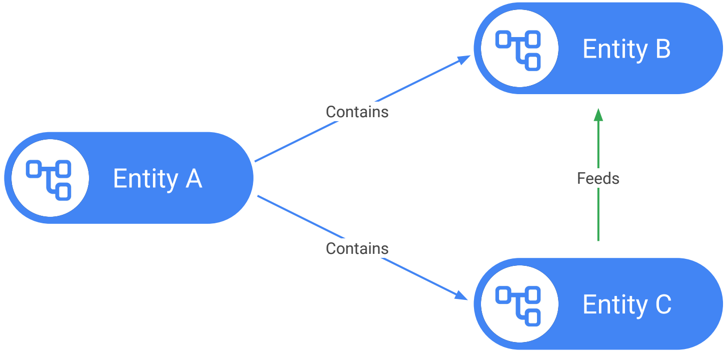
## Data model parts

Every data model includes the following parts:

- **Entities** represent a real-world thing.
- **Entity types** represent the class (or class hierarchy) a specific entity belongs to.
- **Properties** describe the data fields of an entity.
- **Relationships** connect two entities.

### Example

Here's a simple relationship graph. Each entity could represent a variety of different, real-world things. The represented relationship could be between a building (Entity A) that contains a zone (Entity B) and an air handling unit (Entity C).



Back

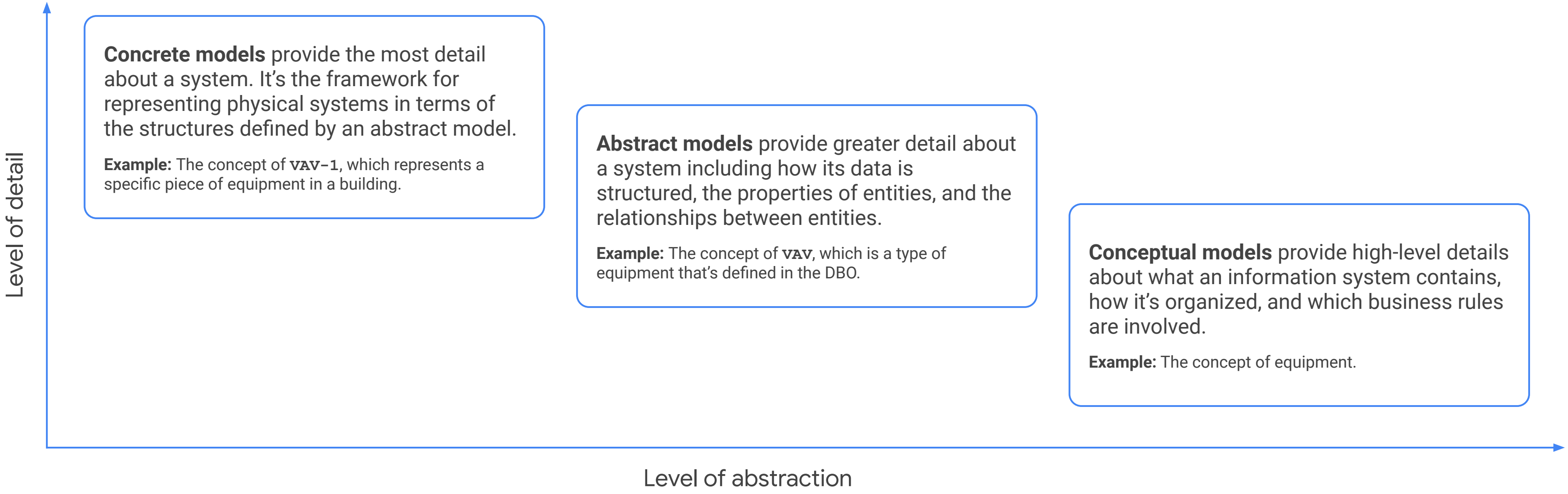
*Click **Next** for more info about data models.*

Next

# Warm-up: Key terms

## **Data model** (continued)

Different types of data models depict varying levels of detail about a system and abstraction of its data.



**Concrete models** provide the most detail about a system. It's the framework for representing physical systems in terms of the structures defined by an abstract model.

**Example:** The concept of `VAV-1`, which represents a specific piece of equipment in a building.

**Abstract models** provide greater detail about a system including how its data is structured, the properties of entities, and the relationships between entities.

**Example:** The concept of `VAV`, which is a type of equipment that's defined in the DBO.

**Conceptual models** provide high-level details about what an information system contains, how it's organized, and which business rules are involved.

**Example:** The concept of equipment.

Back

**Note:** The Digital Buildings Ontology (DBO) captures the conceptual model for the Digital Buildings Project. We usually refer to the DBO as the **abstract model** while a hard deliverable like a building configuration file, which is less abstract, is called a **concrete model**.

Next

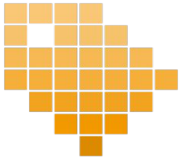
# Warm-up: Key terms

## Ontology

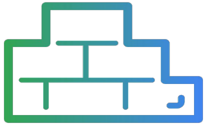
The rules and standardized terms, categories, definitions, and other descriptors that semantically describe an information system.

An ontology is also known as an **abstract model**, a **semantic model**, or a **conceptual model**. Yep! That means an ontology is actually a data model, too.

### Two well-known ontologies for building systems:

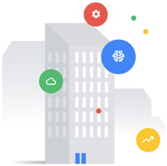


Project Haystack



BrickSchema

### With the Digital Buildings Project, you'll use:



Digital Buildings Ontology (DBO)

Back

*Click **Next** for more info about ontologies.*

Next

Google

# Warm-up: Key terms

## Ontology (continued)

### Why bother with an ontology?

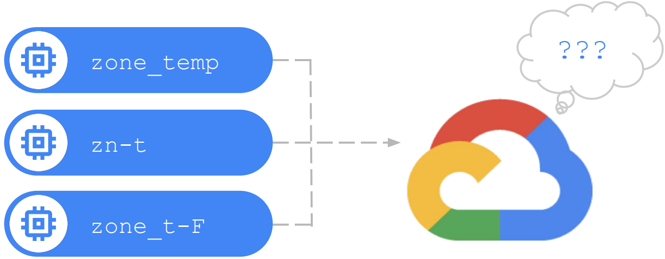
An ontology adds value and efficiency to the process of data modeling. Naming conventions and taxonomies are established by an ontology, which can take a lot of time to develop from scratch.

By adopting and applying an existing ontology, we can skip past that development time and focus on ensuring consistency in our models and interoperability between our systems.

The **Digital Buildings Ontology (DBO)** was developed specifically for the Digital Buildings Project to support our approach to digital buildings. It details entities within domains including HVAC, electrical, and security equipment that are installed in digital buildings. This equipment must be modeled accurately and conform to the DBO so the building model accurately represents the physical building in cloud systems.

### Example

Let's say we've modeled something as simple as a zone temperature sensor. Unfortunately, we did it haphazardly without the guidance of an ontology. We used strings inconsistently to describe the current zone temperature like **`zone_temp`**, **`zn-t`**, **`zone_t-F`**, and so on.



What if a cloud application asked a simple question like, "what's the current zone temperature for all devices with that sensor?" It would need to know ALL the string permutations that were used… Hopefully it can remember all the different variations!

Back

**Note:** A **digital building** is the application of IoT to drive business outcomes and value in the workplace.

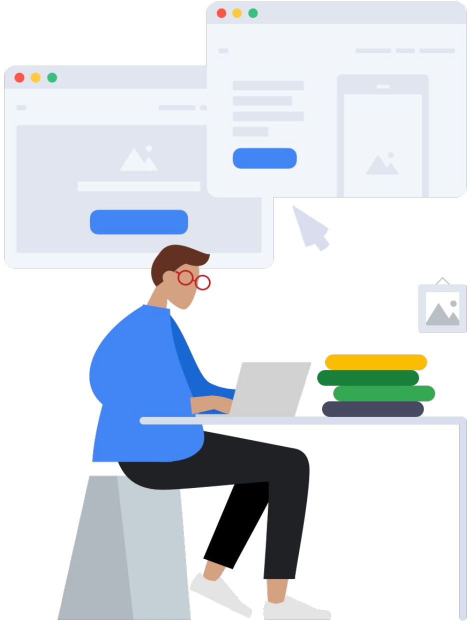
Next

# Warm-up complete

Here are the key terms that were reviewed:

- Data model
- Ontology

Now you’re ready to start Lesson 1.



Back

*Click **Next** when you’re ready to move on to Lesson 1: Introduction to the DBO.*

Next

Google

Lesson 1

# Introduction to the Digital Buildings Ontology (DBO)

## What you'll learn about:

- The Digital Buildings Project
- The Digital Buildings Ontology

## By the end of this lesson, you'll be able to:

- Describe the Digital Buildings Project, including its goals and how they're achieved.
- Explain why we developed the DBO.
- Describe the DBO approach to data modeling.

Back

Next

Google

# What’s the Digital Buildings Project?

The **Digital Buildings Project** is an effort to create a uniform schema and toolset for representing structured information about buildings and the equipment installed in them.

Inspired by similar efforts like Project Haystack and BrickSchema, the Digital Buildings Project emerged from the need to manage Google’s digital building portfolio in a scalable way.

## Our goal

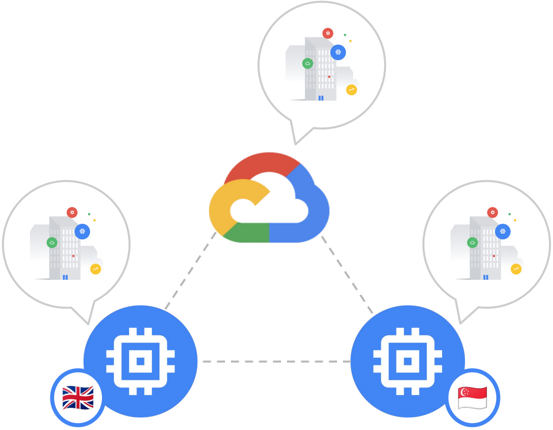
The goal of the Digital Buildings Project is to enable management applications and analyses that are trivially portable between buildings with the following considerations:

- Human readability
- Machine readability and interpretation
- Composable functionality
- Dimensional analysis
- Correctness validation
- Cross compatibility

The end result is analyses that “just work” for entities of the same type in different buildings around the world.

### Example

Let’s say you have a temperature sensor in the UK and a different temperature sensor in Singapore.



Using the Digital Buildings Project’s ontology and tooling, you can establish that these sensors are the same type of entity. Now these sensors can speak the same language with each other and our system, resulting in analyses that “just work.”

Back

Next

Google

# What’s the Digital Buildings Project?

To achieve our goal, the Digital Buildings Project combines abstract modeling, a configuration language, and validation tooling. Here’s how the project is structured:

## Digital Buildings Ontology (DBO)

A custom ontology that was developed for our project. It defines the parameters of the abstract model and the tools for building, validating, and associating real equipment with a concrete model. The ontology is applied to two of your key deliverables: building configuration files and necessary ontology extensions.

You’ll explore the DBO inside and out in Module 1.

## Tooling

Specific tools were developed for this project:

- **Ontology Extension Validator** is used to check if your proposed extension of the DBO is backwards compatible and doesn’t interfere with another entity already defined somewhere else within it. In other words, it maintains internal consistency.
- **Instance Validator** is used to check if your building config is formatted properly and the DBO was applied accurately. It ensures the building telemetry sent to the cloud aligns with what you’ve represented in the concrete model. In other words, it makes sure your building config actually works.
- **RDF/OWL Generator** is used to generate an RDF version of YAML format configs and extensions.

You’ll get hands on with our tooling in Module 2.

## Internal Building Representation (IBR) file format

This format is used to represent where objects are located geospatially.

For the purposes of your onboarding, just be aware that the IBR file format is part of the Digital Buildings Project. However, you won’t be learning about it.

Back

*Click **Next** to reveal the next part of the Digital Buildings Project.*

Next

# What’s the Digital Buildings Project?

To achieve our goal, the Digital Buildings Project combines abstract modeling, a configuration language, and validation tooling. Here’s how the project is structured:

## Digital Buildings Ontology (DBO)

A custom ontology that was developed for our project. It defines the parameters of the abstract model and the tools for building, validating, and associating real equipment with a concrete model. The ontology is applied to two of your key deliverables: building configuration files and necessary ontology extensions.

You’ll explore the DBO inside and out in Module 1.

## Tooling

Specific tools were developed for this project:

- **Ontology Extension Validator** is used to check if your proposed extension of the DBO is backwards compatible and doesn’t interfere with another entity already defined somewhere else within it. In other words, it maintains internal consistency.
- **Instance Validator** is used to check if your building config is formatted properly and the DBO was applied accurately. It ensures the building telemetry sent to the cloud aligns with what you’ve represented in the concrete model. In other words, it makes sure your building config actually works.
- **RDF/OWL Generator** is used to generate an RDF version of YAML format configs and extensions.

You’ll get hands-on practice with our tooling in Module 2.

## Internal Building Representation (IBR) file format

This format is used to represent where objects are located geospatially.

For the purposes of your onboarding, just be aware that the IBR file format is part of the Digital Buildings Project. However, you won’t be learning about it.

Back

*Click **Next** to reveal the next part of the Digital Buildings Project.*

Next

# What’s the Digital Buildings Project?

To achieve our goal, the Digital Buildings Project combines abstract modeling, a configuration language, and validation tooling. Here’s how the project is structured:

## Digital Buildings Ontology (DBO)

A custom ontology that was developed for our project. It defines the parameters of the abstract model and the tools for building, validating, and associating real equipment with a concrete model. The ontology is applied to two of your key deliverables: building configuration files and necessary ontology extensions.

You’ll explore the DBO inside and out in Module 1.

## Tooling

Specific tools were developed for this project:

- **Ontology Extension Validator** is used to check if your proposed extension of the DBO is backwards compatible and doesn’t interfere with another entity already defined somewhere else within it. In other words, it maintains internal consistency.
- **Instance Validator** is used to check if your building config is formatted properly and the DBO was applied accurately. It ensures the building telemetry sent to the cloud aligns with what you’ve represented in the concrete model. In other words, it makes sure your building config actually works.
- **RDF/OWL Generator** is used to generate an RDF version of YAML format configs and extensions.

You’ll get hands on with our tooling in Module 2.

## Internal Building Representation (IBR) file format

This format is used to represent where objects are located geospatially.

For the purposes of your onboarding, just be aware that the IBR file format is part of the Digital Buildings Project. However, you won’t be learning about it.

Back

Next

# The DBO difference

## The DBO is concerned with pragmatic data modeling.

In order to attain the Digital Building Project's goal to enable analyses that "just work," we've taken a simpler approach in comparison to a "pure graph" ontology like BrickSchema.

### DBO approach

The DBO describes entities, their characteristics, and telemetry streams at a high level, which allows you to analyze equipment in a building as a whole unit. This is achieved by typing the equipment and its smaller parts into an abstract concept.

### "Pure graph" approach

A "pure graph" ontology is concerned with modeling the equipment and its parts down to its most atomic component.

Depending on your use case, this approach complicates analyses by making data retrieval difficult.

### Example

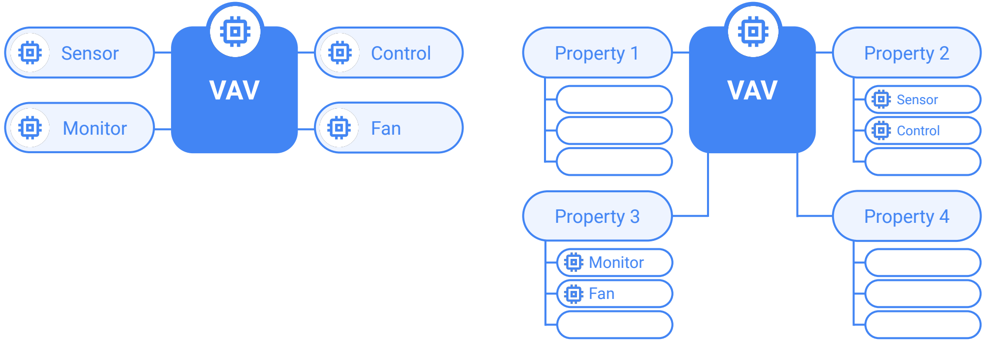
Let's say a VAV needs to be modeled, and it is made of several parts like a sensor, control, and other parts.

#### DBO approach

The abstract concept of the VAV is just a combination of its sensor, control, and other parts.

#### "Pure graph" approach

The abstract concept of the VAV consists of different properties that are further broken down into the VAV's sensor, control, and other parts.



Back

Next

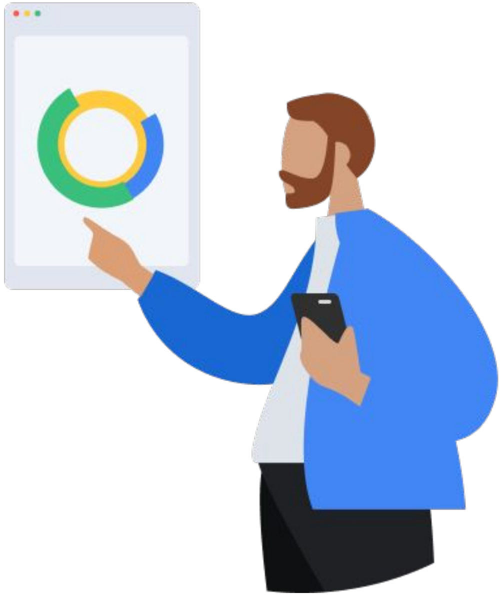
# Lesson 1 summary

## Let's review what you learned about:

- The Digital Buildings Project
- The Digital Buildings Ontology

## Now you should be able to:

- Describe the Digital Buildings Project, including its goals and how they're achieved.
- Explain why we developed the DBO.
- Describe the DBO approach to data modeling.



# You completed Lesson 1!

Now's a great time to take a quick break before starting Lesson 2.

## Ready for Lesson 2?

Let's go!

Back

Google

## Helpful resources

For future reference, keep these resources easily accessible for technical and procedural questions.

- Digital Buildings Project GitHub
  Contains source code, tooling, and documentation for the DBO.