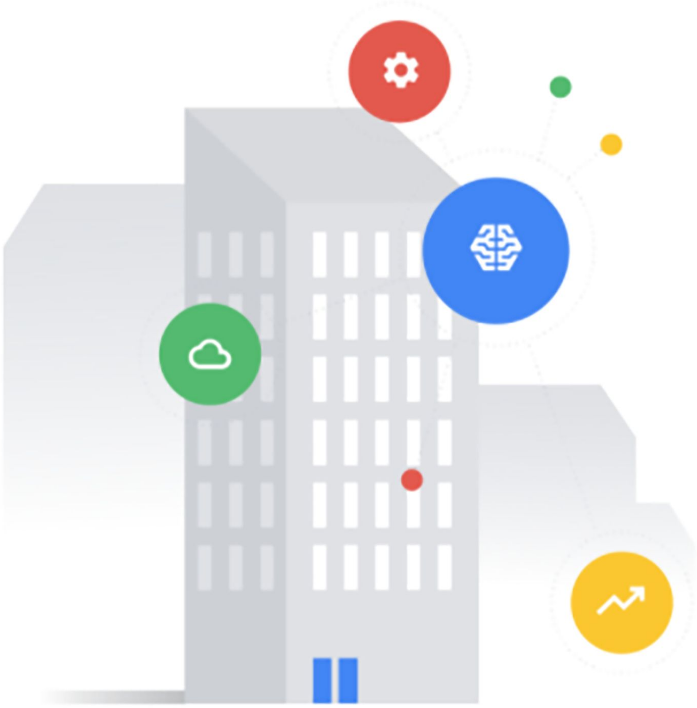
Module 1 | **Lesson 2**

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

Lesson 2

Conceptual model

What you'll learn about:

- Entities in the DBO
- The core concepts of the DBO
- The conceptual model of the DBO

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

- Describe what's considered an entity in the DBO.
- Name the DBO's modeling concepts.
- Name the DBO's organizational concepts.
- Recognize a conceptual model of the DBO.

Back

Next

Google

What’s an entity?

An entity is a concrete concept that represents an actual device or system inside a building.

Some examples of an entity include the representation of a specific thermostat, camera, air handling unit (AHU), electrical meter, or room.

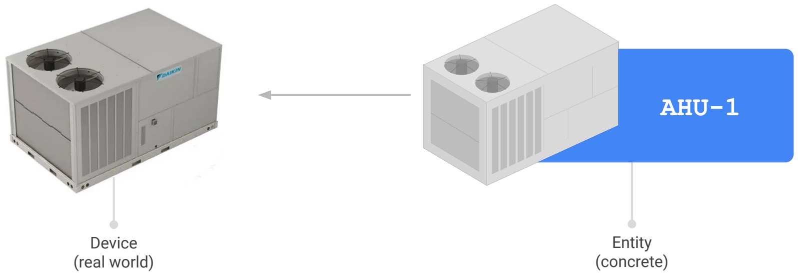
An entity can also represent a specific system or collection of devices such as a chilled water system (CHWS), which is a collection of chillers, pumps, and valves. In this case, the CHWS itself is considered an entity.

An entity is expressed in the building configuration (i.e., concrete model).

Example

Let’s say there’s a single-zone air handling unit (AHU) on the roof of a building.

The concrete instance of the AHU is named `AHU-1`. This is an entity, which represents this specific AHU in the building config.



Back

Next

Google

What’s an entity type?

An entity type is the abstract concept that’s associated with an entity. All entities will have one.

Entity types are composed of smaller abstract concepts that describe the various properties and units of functionality that an entity may have. Entity types are meant to represent all entities of the same type.

An entity type is expressed in the ontology (i.e., abstract model).

Concrete vs. abstract concepts

As discussed in the previous lesson, the DBO is not a “pure graph” ontology. We focus on modeling a device or system as a whole unit.

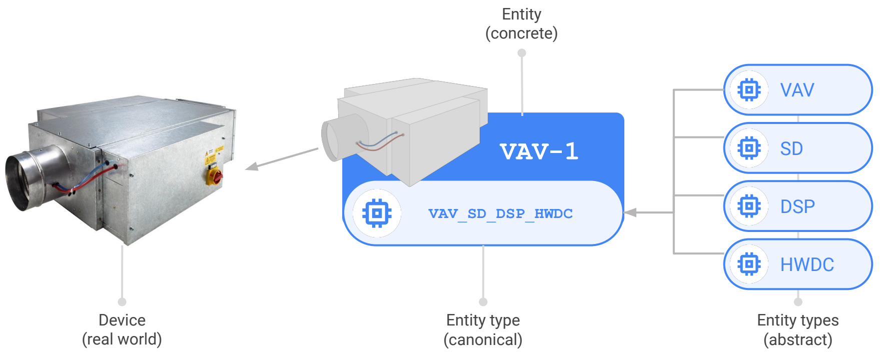
We achieve this by assigning every individual entity (i.e., concrete concept) with a canonical entity type (i.e., abstract concept) composed of smaller abstract concepts.

Think of a concrete concept as the thing you care to model.
Think of an abstract concept as the reusable blocks to build the model.

Example

Let’s say there’s a variable air volume (VAV) unit a building. The concrete instance of the VAV is named `VAV-1`. This is an entity, which represents this specific VAV in the building config.

`VAV-1` is an entity that’s made of several smaller parts and functions. Instead of modeling all of these, `VAV-1` associated with the entity type `VAV_SD_DSP_HWDC`, which includes the abstract concepts `VAV`, `SD`, `DSP`, and `HWDC`. These individual abstract concepts describe all of the smaller parts of `VAV-1`.



Back

Next

Core concepts

The DBO includes two sets of core concepts that you'll use to describe the entities you'll model.

Click on each core concept to reveal more information.

- Modeling concepts
  - Abstract concepts
  - Concrete concepts
- Organizational concepts
  - Global namespace
  - Child namespaces

Back

Next

Google

Core concepts

The DBO includes two sets of core concepts that you'll use to describe the entities you'll model.

Click on each core concept to reveal more information.

- Modeling concepts
  - Abstract concepts
  - Concrete concepts
- Organizational concepts
  - Global namespace
  - Child namespaces

Modeling concepts

The DBO has two types of concepts that are used to describe the properties and relationships of an entity:

- Abstract concepts
- Concrete concepts

Back

Next

Core concepts

The DBO includes two sets of core concepts that you'll use to describe the entities you'll model.

Click on each core concept to reveal more information.

Modeling concepts

- Abstract concepts
- Concrete concepts

Organizational concepts

- Global namespace
- Child namespaces

Abstract concepts

These concepts use the ontology to describe the properties and functionality of an entity.

1. Subfield

A single or compounded word with a specific, human-readable definition. This is the basic unit of meaning in the DBO.

2. Dimensional unit

Applied to subfields to indicate the unit of measurement for numeric values.

3. Field

A grouping of subfields in a structured way to define a concept semantically.

4. Multi-states

A list applied to a field consisting of all the allowed states that describe a device's condition, status, setting, position, or mode (e.g., `ON`, `OFF`, and `AUTO`).

5. Entity type

A composition of fields, other entity types, and attributes that describe the properties and functionality of an entity.

6. Connection type

A standard type of connection between two entities that describes their relationship to each other (e.g., `CONTAINS`, `FEEDS`).

Back

Next

Core concepts

The DBO includes two sets of core concepts that you'll use to describe the entities you'll model.

Click on each core concept to reveal more information.

- Modeling concepts
  - Abstract concepts
  - Concrete concepts
- Organizational concepts
  - Global namespace
  - Child namespaces

Concrete concepts

These concepts use the building config to describe the relationships between devices in the real world and abstract concepts in the DBO.

1. Entity

The concrete instance that represents a device or system in the real world. Entities are always associated with an entity type.

2. Translation

A type of mapping that translates the native payload of a device or system in the real world into its corresponding entity, entity type, and fields.

3. Link

A type of mapping that links the standard fields of two entities to pass data between them.

4. Connection

A named relationship between two entities that defines how they're systematically or spatially connected.

Back

Next

Core concepts

The DBO includes two sets of core concepts that you'll use to describe the entities you'll model.

Click on each core concept to reveal more information.

- Modeling concepts
  - Abstract concepts
  - Concrete concepts
- Organizational concepts
  - Global namespace
  - Child namespaces

Organizational concepts

The DBO is organized into two levels of namespaces:

- Global namespace
- Child namespaces

We've organized the DBO this way to make it easier for modelers to find and retrieve the modeling concepts needed for data modeling.

Back

Next

Core concepts

The DBO includes two sets of core concepts that you'll use to describe the entities you'll model.

Click on each core concept to reveal more information.

- Modeling concepts
  - Abstract concepts
  - Concrete concepts
- Organizational concepts
  - Global namespace
  - Child namespaces

Global namespace

The global namespace contains curated modeling concepts including:

- Subfields
- Units
- Fields
- States
- Entity types
- Connection types

Curated modeling concepts are broadly defined, making them applicable to reuse by any application domain.

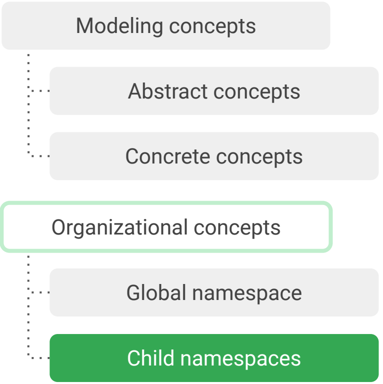
Back

Next

Core concepts

The DBO includes two sets of core concepts that you'll use to describe the entities you'll model.

Click on each core concept to reveal more information.



Modeling concepts

- Abstract concepts
- Concrete concepts

Organizational concepts

- Global namespace
- Child namespaces

Child namespaces

A child namespace is named after a particular application domain. It contains the curated entity types that are specific to that domain. **Fields and other modeling concepts are rarely defined in a child namespace.**

The DBO includes the following child namespaces:

- HVAC
- Lighting
- Metering
- Electrical
- Safety
- IoT

Back

Next

Lesson 2

Knowledge check

Back

Google

Let’s take a moment to reflect on what you’ve learned so far.

- The next slides will have questions about the concepts that were introduced in this lesson.
- Review each question and select the correct response.

If there are more than two answer options, you won’t be able to move forward until the correct answer is selected.

*Click **Next** when you’re ready to begin.*

Next

Knowledge check 1

An entity is the term used to represent the actual “things” that are inside of a building.

Is this statement true or false?

Select the best answer from the options listed below.

True

False

Back

Next

Google

Knowledge check 1

An entity is the term used to represent the actual “things” that are inside of a building.

Is this statement true or false?

Select the best answer from the options listed below.

[✓ True]

False

That’s right! 🎉

Some examples of entities include thermostats, cameras, AHUs, electrical meters, and rooms.

Remember, we use the DBO to describe an individual, concrete entity (e.g., `AHU-1`) with an abstract concept of the entity (e.g., `HVAC/AHU_DFSS_...`).

Back

Next

Knowledge check 1

An entity is the term used to represent the actual “things” that are inside of a building.

Is this statement true or false?

Select the best answer from the options listed below.

True

False

Close... but not quite right! 🤔

Some examples of entities include thermostats, cameras, AHUs, electrical meters, and rooms.

Remember, we use the DBO to describe an individual, concrete entity (e.g., `AHU-1`) with an abstract concept of the entity (e.g., `HVAC/AHU_DFSS_...`).

Back

Next

Knowledge check 2

All entities must have a/an ______. This represents the abstract concept that describes the individual, concrete instance of an entity.

Fill in the blank.

Select the best answer from the options listed below.

thermostat	namespace
entity type	translation

Back

Next

Knowledge check 2

All entities must have a/an ______. This represents the abstract concept that describes the individual, concrete instance of an entity.

Fill in the blank.

Select the best answer from the options listed below.

thermostat

namespace

entity type

translation

Close... but not quite right! 🤔

A **thermostat** is a great example of an entity. However, not all entities have one.

Try again

Back

Next

Knowledge check 2

All entities must have a/an ______. This represents the abstract concept that describes the individual, concrete instance of an entity.

Fill in the blank.

Select the best answer from the options listed below.

thermostat

[✓ namespace]

entity type

translation

Close... but not quite right! 🤔

A **namespace** is used for organizing and structuring the DBO. However, entities aren't directly related to this.

Try again

Back

Next

Knowledge check 2

All entities must have a/an ______. This represents the abstract concept that describes the individual, concrete instance of an entity.

Fill in the blank.

Select the best answer from the options listed below.

thermostat

namespace

[✓ entity type]

translation

That’s right! 🎉

An **entity type** is an abstract concept of an entity. A composition of fields (and inherently the fields’ states, units, and subfields), other entity types, and attributes that describe the properties and functionality of an entity.

Back

Next

Knowledge check 2

All entities must have a/an _____. This represents the abstract concept that describes the individual, concrete instance of an entity.

Fill in the blank.

Select the best answer from the options listed below.

thermostat

namespace

entity type

translation

Close... but not quite right! 🤔

A **translation** is a type of mapping used between entities and entity types. However, entities don't need to have one.

Try again

Back

Next

Knowledge check 3

Abstract modeling concepts are used to describe the properties of an entity.

Which term is an abstract modeling concept?

Select the best answer from the options listed below.

Subfield	Field
Entity type	All of the above

Back

Next

Knowledge check 3

Abstract modeling concepts are used to describe the properties of an entity.

Which term is an abstract modeling concept?

Select the best answer from the options listed below.

Subfield

Field

Entity type

All of the above

Close... but not quite right! 🤔

Yes, a **subfield** is definitely an abstract modeling concept. However, is it the only one out of all the options?

Try again

Back

Next

Knowledge check 3

Abstract modeling concepts are used to describe the properties of an entity.

Which term is an abstract modeling concept?

Select the best answer from the options listed below.

Subfield	Field
Entity type	All of the above

Close... but not quite right! 🤔

Yes, a **field** is definitely an abstract modeling concept. However, is it the only one out of all the options?

Try again

Back

Next

Knowledge check 3

Abstract modeling concepts are used to describe the properties of an entity.

Which term is an abstract modeling concept?

Select the best answer from the options listed below.

Subfield

Field

[✓ Entity type]

All of the above

Close... but not quite right! 🤔

Yes, an **entity type** is definitely an abstract modeling concept. However, is it the only one out of all the options?

Try again

Back

Next

Knowledge check 3

Abstract modeling concepts are used to describe the properties of an entity.

Which term is an abstract modeling concept?

Select the best answer from the options listed below.

Subfield   Field

Entity type   [✓ All of the above]

That’s right! 🎉

All three options – **subfields**, **fields**, and **entity types** – are abstract modeling concepts. Here’s a reminder of all of them:

- **Subfield:** The basic unit of meaning in the DBO. It’s simply a single or compounded word with a specific, human-readable definition.
- **Dimensional unit:** Applied to subfields to indicate the unit of measurement for numeric values.
- **Field:** A grouping of subfields in a structured way to define a concept semantically.
- **Multi-states:** A list applied to a field consisting of all the allowed states that describe a device’s condition, status, setting, position, or mode (e.g., `ON`, `OFF`, and `AUTO`).
- **Entity type:** A composition of fields, other entity types, and attributes that describe the properties and functionality of an entity.
- **Connection type:** A standard type of connection between two entities that describes their relationship to each other (e.g., `CONTAINS`, `FEEDS`).

Back

Next

Google

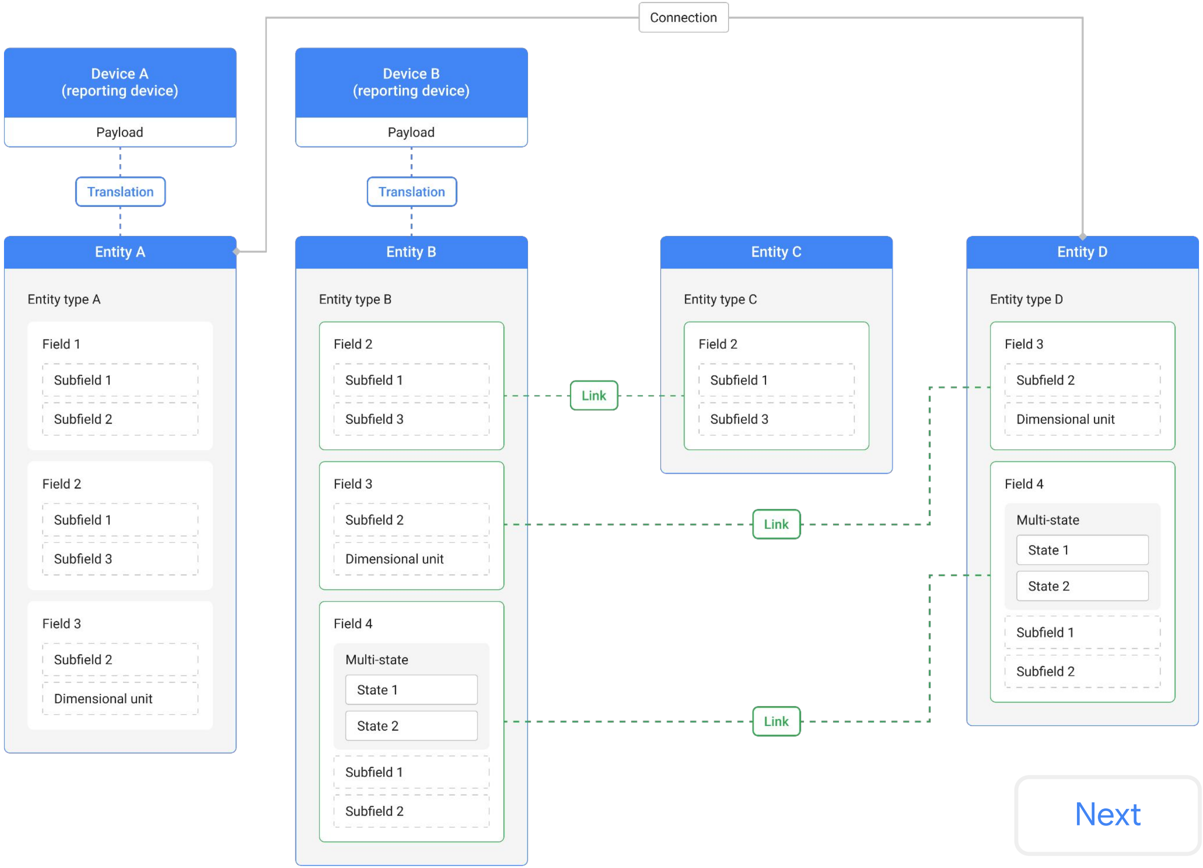
Conceptual model

Great! It looks like you have a good understanding of the DBO's core concepts.

Take a look at the diagram to the right. This is a visualization of the DBO conceptual model.

If you look closely, you should recognize the core concepts you were introduced to including entities, entity types, fields, translations, links, and connections.

Moving forward, you'll explore each of the core concepts individually. You'll start with the DBO's smallest abstract concept and build up to the largest organizational concept.



Back

Next

Google

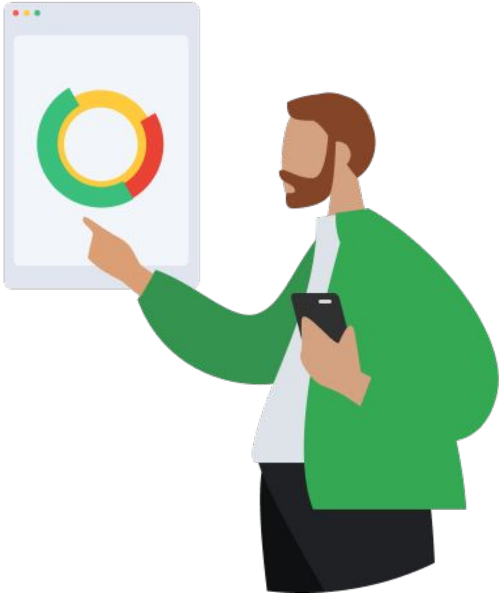
Lesson 2 summary

Let’s review what you learned about:

- Entities in the DBO
- The core concepts of the DBO
- The conceptual model of the DBO

Now you should be able to:

- Describe what is considered an entity in the DBO.
- Name the core concepts of the DBO.
- Recognize a conceptual model of the DBO.



Back

Next

You completed Lesson 2!

Now's a great time to take a quick break before starting Lesson 3.

Ready for Lesson 3?

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.