

Concept: Data model

On this page:

- [Definition](#)
- [Example of a data model: Office building](#)
- [Relation to existing data model breakdown](#)

Related articles

- [Concept: Property](#)
- [Concept: Task](#)
- [Working in a task form](#)
- [Concept: Part](#)
- [Managing the explorer](#)

Definition

A **data model** is a structured representation of a (physical) product. Its level of detail and complexity depend on the product and on requirements from a design and/or manufacturing perspective. In the **KE-chain** application, a data model consists of combinations of two principal components: [part models](#) and [properties](#). Conceptually, the data model is closest to a [Product Breakdown Structure](#).

Still not clear?

Read more regarding **data** modeling on [this](#) Wikipedia article.

Example of a data model: Office building

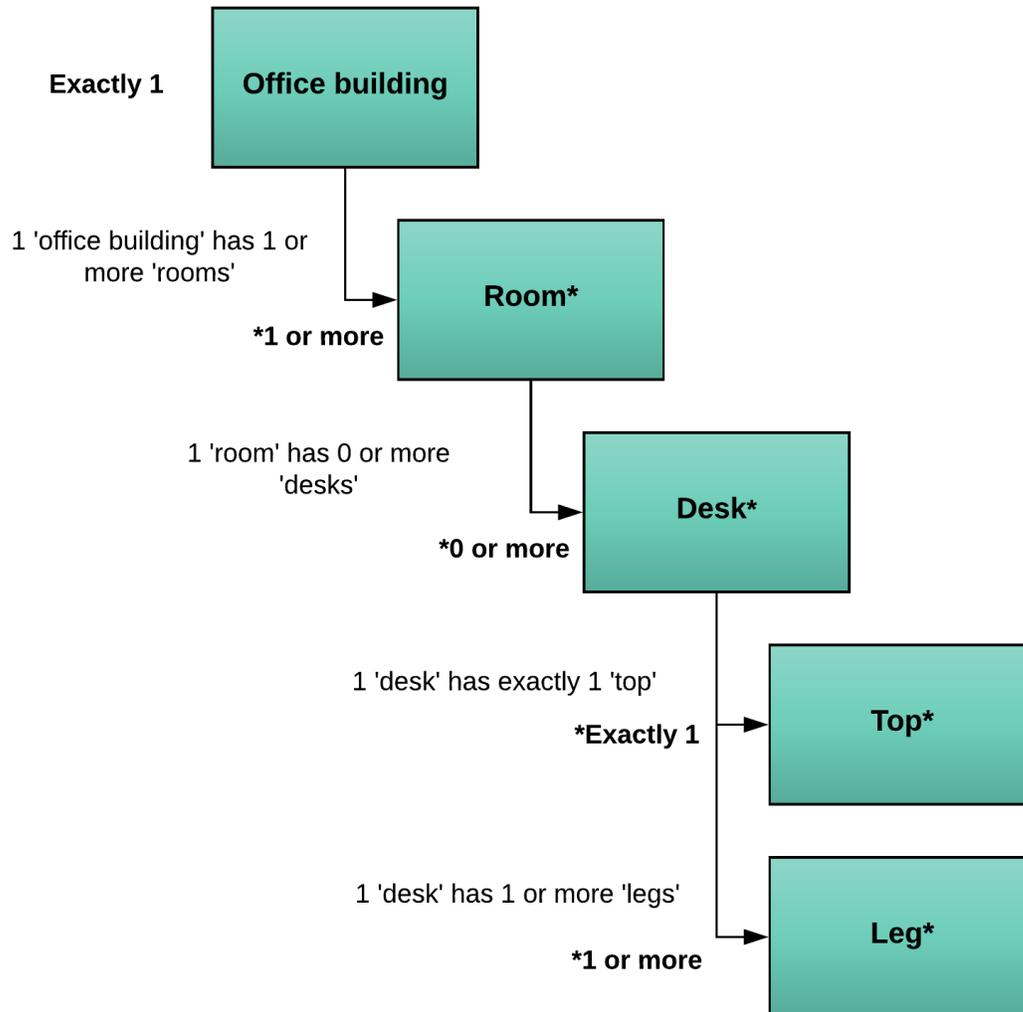
Examples of data models can be found everywhere in our day to day lives. In the figure below, an example of a **data model** called "Office building" is presented. An office consists of multiple **rooms**. These rooms are the *sub-parts* of the office. Similarly, the room can contain multiple **desks**, which in turn consist of a **top** and **legs**. The number of sub-parts is defined by its *quantity*. For example, the building consists of at least 1, but possibly multiple rooms. Therefore, its quantity is defined as **1 or more** in the data model tree. By the same logic, we specify that each room can have **0 or more** desks, with each desk having **exactly 1** top and **1 or more** legs.

How is it related to the product?

This "Office building" data model is linked to an example product called "Yes!Delft incubator". Check out how they are connected by reading the [Concept: Explorer](#) page.

Example product model tree

With 5 part models having various quantities



What are quantities?

Read more about how to define them in the [Concept: Part](#) page.

Relation to existing data model breakdown

The **KE-chain** way of breaking down a data or product model (its *ontology*) is related to common standards such as [STEP](#). While STEP itself is has a broad scope and is extensive, the KE-chain way of data modeling is more down-to-earth and easier to understand. The following picture (taken from [here](#)) shows how the KE-chain data model maps onto a basic product ontology.

