Modeling Physical Objects

This chapter provides patterns for describing physical things in the world, specifically how complex objects are composed of their parts.

# Pattern: An Assembly and its Components (IOF-based pattern)

## Objective:

This pattern illustrates how to represent a complex object (an assembly) and its relationship to its constituent parts (components), based on the IOF/BFO framework. It provides a foundational structure for a Bill of Materials.

## Scenario:

We have a modern table (ns1:table) which is an assembly. We want to represent its main components: a tempered glass top, a steel frame, steel legs, and nylon wheels.

## Description:

In this pattern, both the main assembly and its components are considered bfo:MaterialEntity. The table itself (ns1:table) is the central entity. Each of its parts, like ns1:tempered-glass-top and ns1:steel-rectangular-frame, is defined as an individual.

The core of this pattern is the relationship bfo:hasContinuantPartAtSomeTime. This property links the whole (the table) to each of its parts (the components). This accurately reflects the structure shown in the "Use Case Pattern Description" diagram.

Information about the material of each component (e.g., "Tempered Glass", "Steel") and its specific qualities (e.g., "Thickness", "Length") is considered descriptive data that can be associated with the component, but the simplest IOF pattern focuses only on establishing the part-whole structure.

A diagram of a computer

AI-generated content may be incorrect.

Diagram: Basic Assembly Structure

This diagram shows only the fundamental part-whole relationships as described in the IOF Core Patterns document.

## Expanding the Pattern with Qualities (CCO-based extension)

Objective:  
While the IOF pattern focuses on structure, the CCO document provides the principles to add measurable attributes. We can create an *extended* version of the pattern that includes qualities like length and thickness, staying true to the CCO design rationale.

Description:  
We can build upon the basic assembly structure by adding qualities. A quality like length is a Specifically Dependent Continuant that inheres\_in its bearer (the component). This is a more advanced but powerful way to model the data from the "Use-Case Example Data" table.

**Use-Case Example Data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Component ID | Product ID | Component Type | Material | Quality Type | Quality Value | Unit |
| TOP001 | TABLE001 | Table Top | Tempered Glass | Thickness | 10 | mm |
| TOP001 | TABLE001 | Table Top | Tempered Glass | Area | 0.8 | m² |
| FRAME001 | TABLE001 | Table Frame | Steel | Length | 100 | cm |
| FRAME002 | TABLE001 | Table Frame | Steel | Length | 80 | cm |
| LEG001 | TABLE001 | Table Leg | Steel | Height | 75 | cm |
| WHEEL001 | TABLE001 | Wheel | Nylon | Diameter | 50 | mm |

A diagram of a company

AI-generated content may be incorrect.

Diagram Data: Assembly with Component Qualities

This diagram shows how to add the "Thickness" quality to the tabletop component.