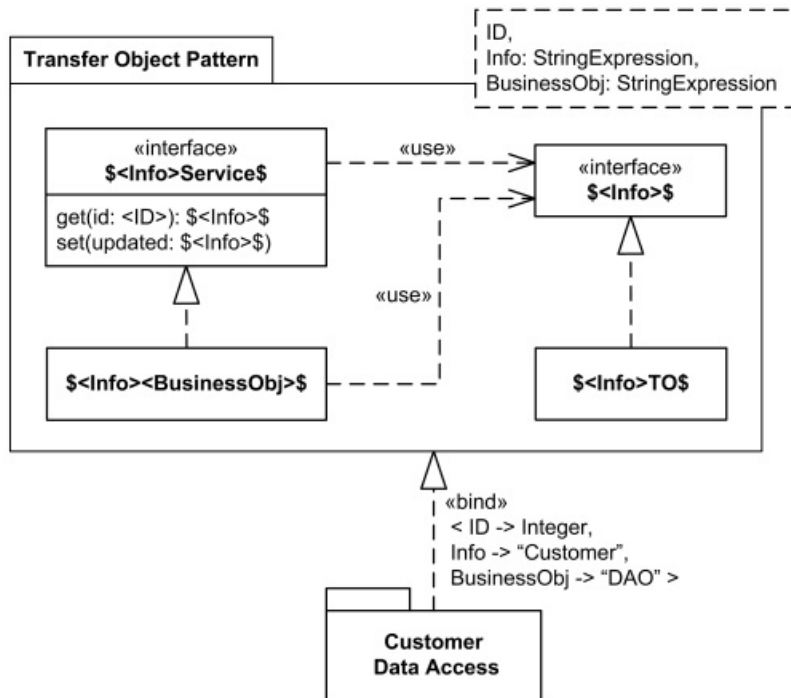


Transfer Object Design Pattern

UML Package Template Example

A UML **package template** example for the design pattern known as **Transfer Object** in **Core J2EE patterns** [CJ2EEP 02] or **Data Transfer Object** (DTO) in **Microsoft .Net Application Architecture Guide** [MAAG 09]. It was also known as **Value Object** (VO) in older version of J2EE design patterns.

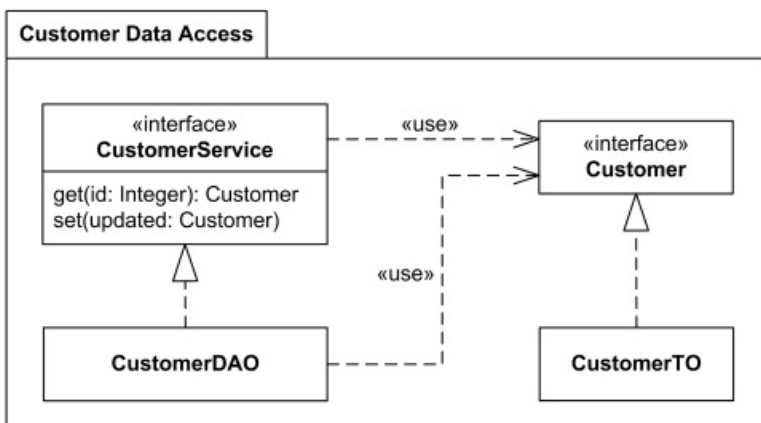
In .Net world, Value Objects are immutable domain objects that do not have a unique identity. An example of the value object could be a Transaction Amount or a Customer Address. It used to be called **structure** in older programming languages. **(Data) Transfer Object** is some "coarse-grained" data structure intended to be moved across different boundary layers or transported over the network, usually serializable, so that a single method call could be used to get or to send the whole Transfer Object to reduce the number of remote calls. Examples of DTOs are Customer, Purchase Order, Account, etc.



UML package template example for the Data Transfer Object design pattern.

The template has unconstrained class parameter ID and two **string expression** parameters. Binding replaces formal class ID parameter with Integer class while string expression parameters are used to build names of actual interfaces and classes. Customer Data Access is **bound** package.

The result of the binding is shown below. Customer Data Access package contains Customer DAO (Data Access Object) and Customer TO (Transfer Object) classes implementing corresponding interfaces.



Customer Data Access is a template bound package.

Noticed a spelling error? Select the text using the mouse and press Ctrl + Enter.

Follow [@uml_diagrams](#)      by [Kirill Fakhroutdinov](#)

This document describes **UML 2.5** and is based on **OMG™ Unified Modeling Language™ (OMG UML®) 2.5** specification **[UML 2.5 FTF - Beta 1]**.

All UML diagrams were created in **Microsoft Visio** 2007-2016 using **UML 2.2 stencils**. You can send your comments and suggestions to [webmaster](#) at webmaster@uml-diagrams.org.

Copyright © 2009-2018 uml-diagrams.org. All rights reserved.