

1.1 Data Transfer Object Pattern

1.1.1 Name and Source

Data Transfer Object pattern & Data Transfer Collection pattern
Page 279-295 in the book "Web Service Patterns: Java Edition" [WSP 03]

Additional and general (i.e. not web service focused) descriptions of the pattern:

Patterns of Enterprise Application Architecture [PEAA 02]

<http://www.martinfowler.com/books.html#ea>

Enterprise Solution Patterns Using Microsoft .NET [ESP 03]

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpatterns/html/DesDTO.asp>

Core J2EE Patterns ("Transfer Object", previously known as "Value object") [CJP 03]

<http://www.corej2eepatterns.com/Patterns2ndEd/TransferObject.htm>

1.1.2 Also Known As

Note that the pattern should **not** anymore be referred to as Value Object.

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnpatterns/html/DesDTO.asp>

A quote from the URL above:

"Some books refer to Data Transfer Object as Value Object. This usage is no longer considered correct. For more information, see 'Patterns of Enterprise Application Architecture' "[PEAA 02].

1.1.3 Type

1.1.4 Intent

To ensure a consistent view of some data, and also improve performance and reduce network traffic by packaging data into one parameter that will be transferred over the net in one single call.

1.1.5 Problem

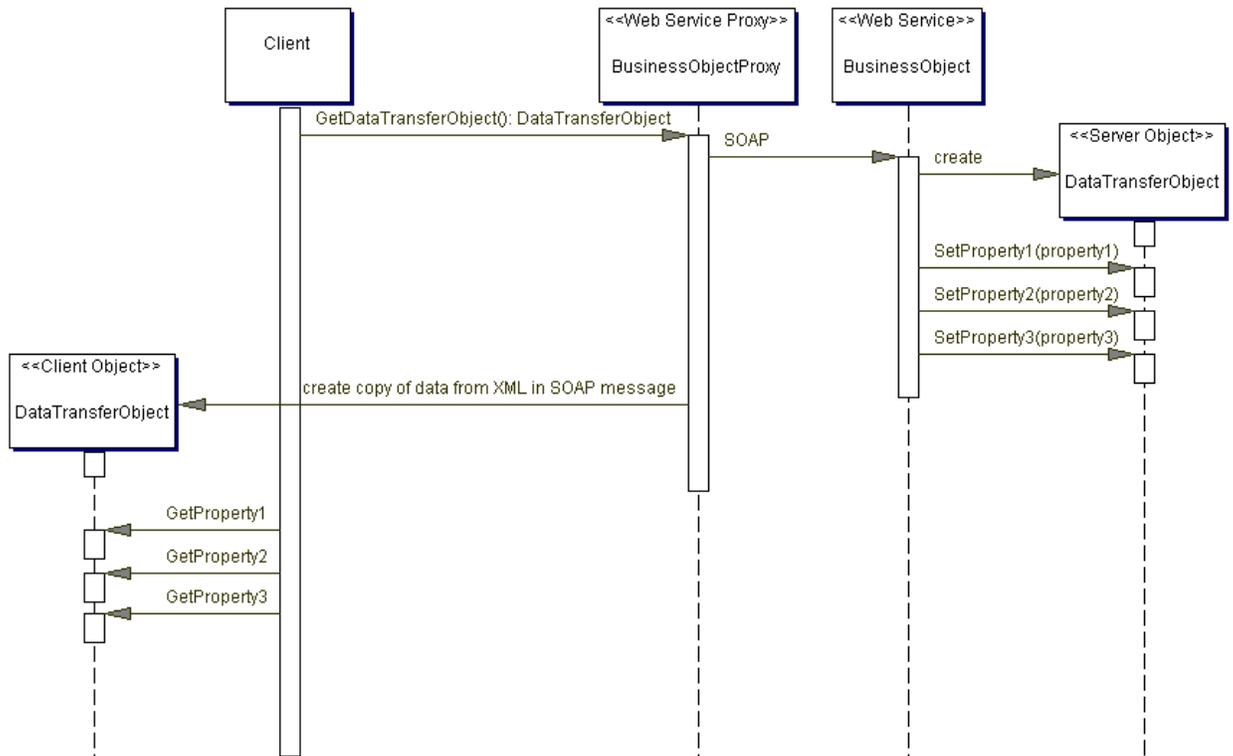
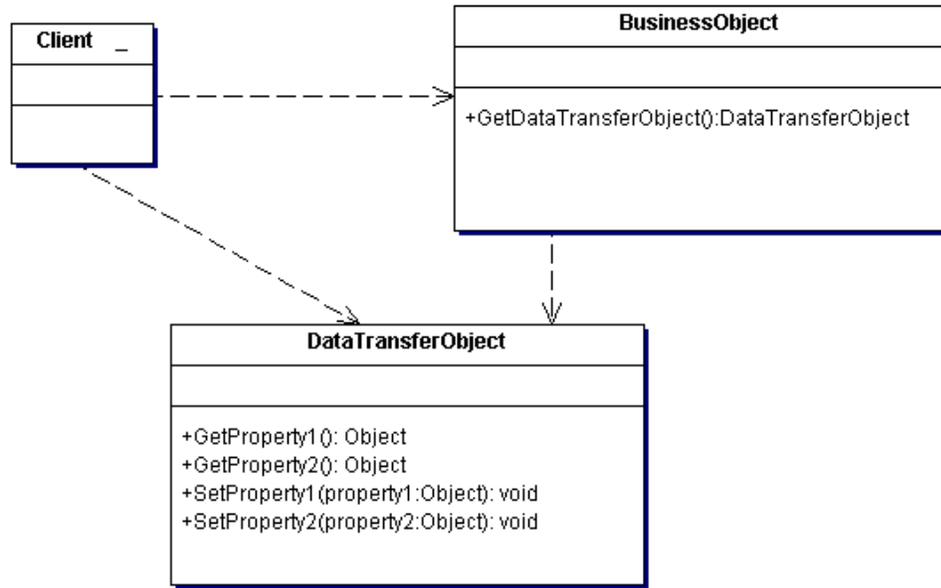
If a business object is a proxy to a remote object, then each invocation of a method that sends or receives a parameter to or from the business object will cause network traffic and have a negative impact of performance. Also, it is possible that some of the data may be inconsistent since data may be updated by another client between the invocations of the methods.

How can the frequency of network traffic be reduced and inconsistent data be avoided ?

1.1.6 Forces

1.1.7 Solution

Create a data transfer object that will contain a copy of a subset of the data in business objects. This data transfer object will then be sent as one single SOAP message and it will be recreated at the other side, and all the data can then be retrieved with local calls. The business object will provide a web service method that returns such a data transfer object. Since a copy of the data is sent in one call, there will not be any inconsistency that might occur if other clients would cause updates while doing multiple remote calls to the business object proxy.



DataTransferObject <<Client>> – A client sided copy of the server sided data transfer object. There may be multiple data transfer objects that each represents a different subset of the data, and each such data transfer class will be used from the client as well as the server.

Client – A consumer of the web service.

BusinessObjectProxy – A proxy to the web service.

BusinessObject – The web service implementation.

`DataTransferObject<<Server>>` – A data transfer object that will contain a subset of the data in the business object web service. The object will not have any behavioral methods and its only purpose is to be a data object sent over the network. To reduce the footprint of the object, it may therefore provide public attributes instead of accessor methods.

1.1.8 Consequences

If a data transfer object is going to be modified at the client and then be sent back to the server for making updates of some data, then it is necessary that such an instance of a data transfer object must contain at least a unique key that can identify which object to update, i.e. the value of a database primary key could be used.

It may be difficult for the web service provider to predict what groups of related data the clients will want to receive as a data transfer object, and there is a risk that the web service will get a cluttered interface with a lot of methods that return different data transfer objects.

To reduce the footprint of the data transfer objects you may want to use public attributes instead of accessor methods but some tools, i.e. Apache Axis for Java, may not provide an option to generate such code but always generate accessor methods and private attributes.

If the web service in this pattern would be a “Business Object Collection” (BOC) instead of a business object, then that variant of this pattern will be called “Data Transfer Collection pattern” (DTC). The BOC web service will then not only be able to return business objects but also provide methods that can return data transfer objects that represent subsets of business objects.

1.1.9 Related patterns

The “Partial Population pattern” can be used to provide more flexibility to the client about which attributes are wanted in a data transfer object, without cluttering the web service interface with a lot of methods that return different data transfer object classes.