PartialPipelineFilterProposal

Problem Description

Currently a filter has to support all methods defined by the service to be pipelined. In some situations this is not desirable because

1. The service designer wants to allow only a subset of the methods to be filtered.

2. To optimize the performance of some methods.

Without support from the PipelineFactory, a designer can achieve the same objective by breaking down the interface into smaller interfaces, create a pipeline for one of these sub-interface only, and then build a facade to combine the small interfaces. However, this increases the complexity of the design.

Proposed Solution

A proposal solution is to modify the PipelineFactory so that the first Bridge short-circuits to the terminator if the caller calls a method that is not supported by filters. The factory shall be able to find these methods by comparing the filter interface to the service interface.

Calling sequence for a normal pipeline method:

Caller -> Bridge1 -> Filter1 -> Bridge2 -> Filter2 -> Bridge3 -> Filter3 -> Terminator

Calling sequence for a short-circuit pipeline method:

Caller -> Bridge1 -> Terminator

Discussions