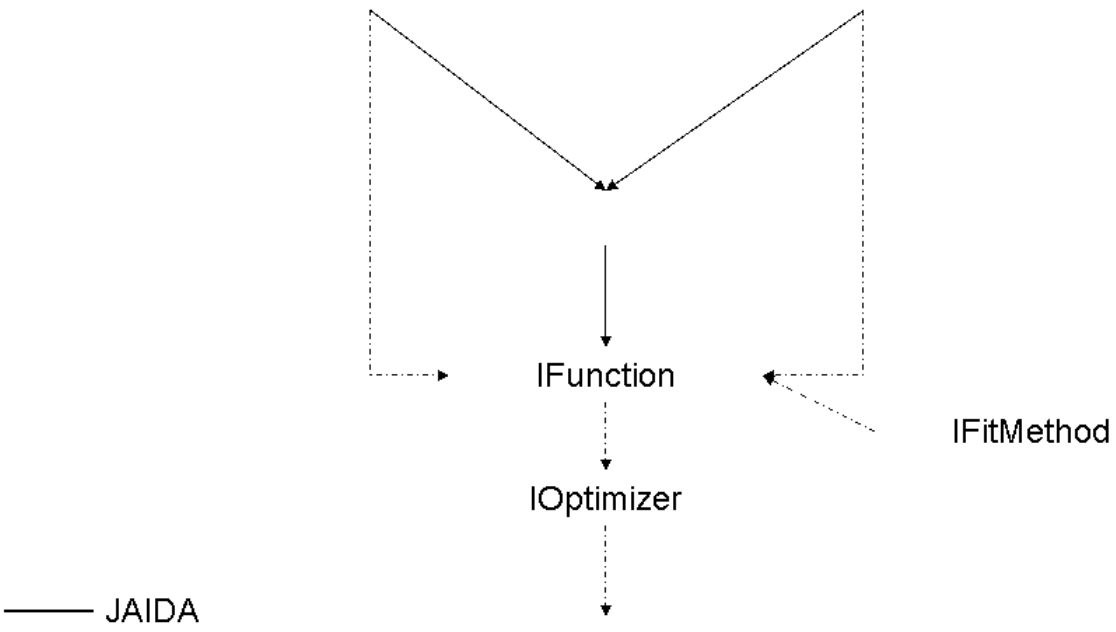
Fitter

The design of the JAIDA implementation of the AIDA IFitter interface is shown in the diagram below.
As shown, the fitter brings the data and the model together, it builds an objective function based on the chosen statistical method (specified by the IFitMethod); the objective function is optimized by the selected IOptimizer. The optmization is perfomed by changing the free parameters of the original function. The result of the fit is return to the user in an IFitResult object.



Both the fit method and the optimizer can be selected when creating the fitter through the IFitFactory. For a list of the available optimizers and fit methods please refer to the JAIDA release notes.

The following code creates a fitter that uses *JMinuit*, the Java implementation of Minuit, as an optimizer and is configured to performe Unbinned Maximum Likelihood fits:

```
// The AIDA Factories
IAnalysisFactory af = IAnalysisFactory.create();
IFitFactory fitf = af.createFitFactory();

// Create the fitter
IFitter fitter = fitf.createFitter("uml","jminuit");
```

Controlling The Parameters

Function's parameters that enter the fit can be controlled via the IFitParameterSettings. These objects can be accessed at the fitter level by using the name of the parameter that they are meant to control. As they belong to the fitter, the fit parameter settings will apply to any fit in which there is a parameter that matches their name.
Throught the IFitParameterSettings it is possible to fix/release parameters, set their step size and their boundaries as shown in the following code:

```
//  Fix parameter "a" and set it to be bounded in [0,1]
fitter.fitParameterSettings("a").setFixed(true);
fitter.fitParameterSettings("a").setBounds(0,1);

// Set the step size for parameter p0 to be 0.01
fitter.fitParameterSettings("p0").setStepSize(0.01);
```

Setting Constraints

At the fitter level it is also possible to set constraints. Even though the idea is to support pretty general constraints on the parameters and on the domain space, currently we only support very simple constraints: it is only possible to force one parameter to be identical to another:

```
//  Set parameter "mean1" to be identical to "mean2"
fitter.setConstraint("mean1=mean2");
```