**UserInput (Boundary class)**

* Collects user input from the command line, text files (one of the extension components) etc. Parses those inputs and then sends the results to DFT singleton object
* Some of these inputs include DFT parameters like initial position, max search distance and the learning rate

**DFT**

* (Distribution Field Tracker) Handles the business logic and glue code which iterates over the image sequence in order to track the object. Calls methods of the DF class where necessary to get the DF of input images. Contains the model of the object as a member variable and the gradient descent search algorithm (locateObject() ) as a member function in order to locate the object from frame to frame.

**ObjectModel**

* Responsible for containing all information regarding our model of the object in question. Contains methods for updating the model for each iteration of the DFT business logic loop. Stores various parameters of the algorithm, ie. colourBlurSize, spatialBlurSize
* Will contain the extra colour information provided for the “track objects in colour” extension.

**DF**

* Responsible for the creation of distribution fields and operations which make use of/compare/update them
* Contains a number of parameters such as spatialBlurSize which may be changed by the user to adjust the algorithm
* The DF class makes use of several blur functions of the vxl libraries to create its distribution fields eg. vil\_gauss\_filter\_1d, vil\_gauss\_filter\_2d

**EDFT**

