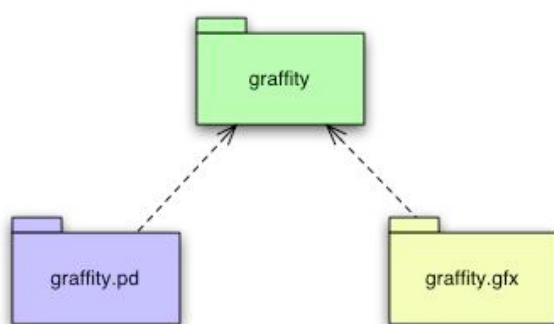


## Project: Graffity

### *Packages: Overview*

Package Name	Description
graffity	This package contains the main class. Only this class may be executed.
graffity.pd	The problem domain classes of Graffity.
graffity.gfx	Graphical representation of the core classes as well as GUI classes.

### Model



*Figure 11: Graffity package structure*

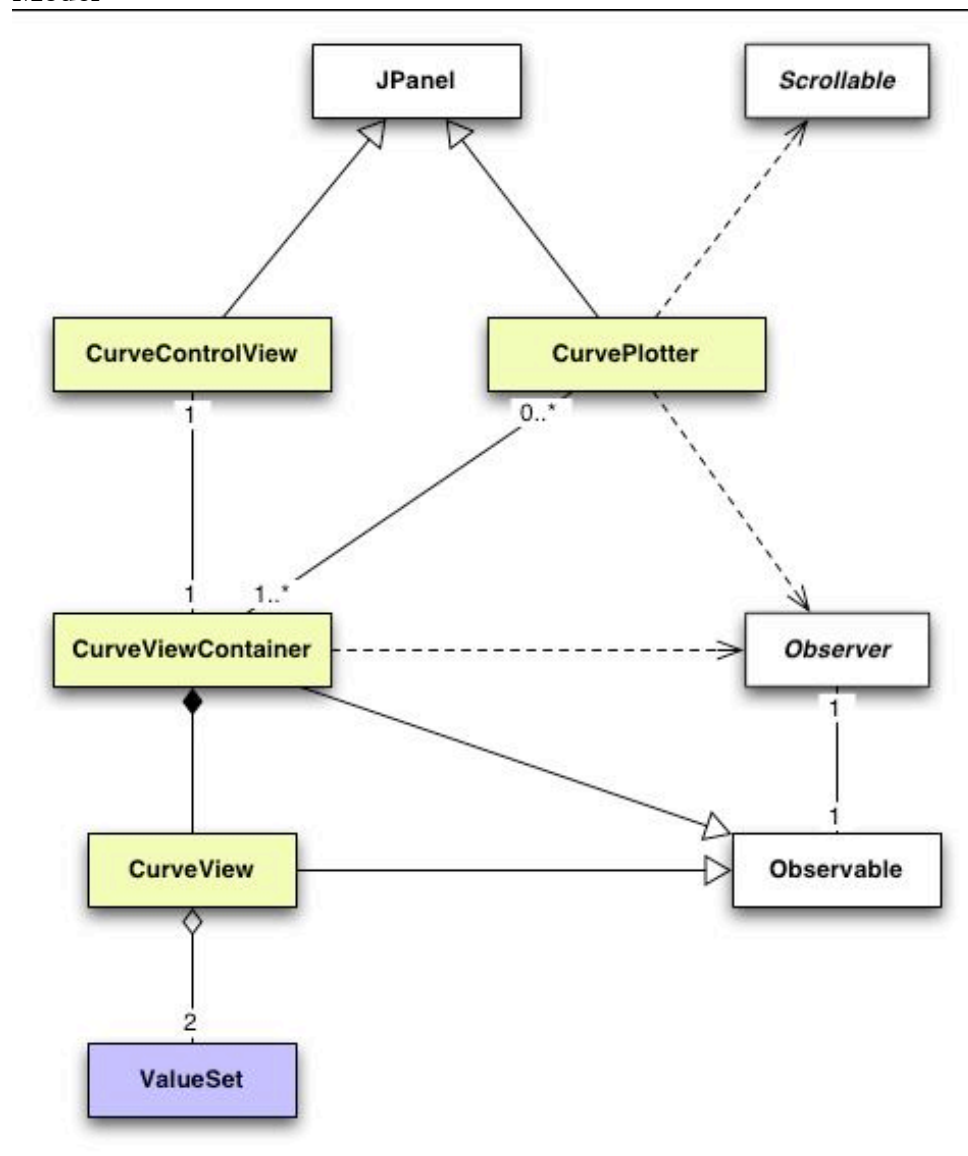
***Package: graffiti***

Class	Description
Graffiti	The main class. Starts the application and displays the GUI.

**Model***Figure 12: graffiti*

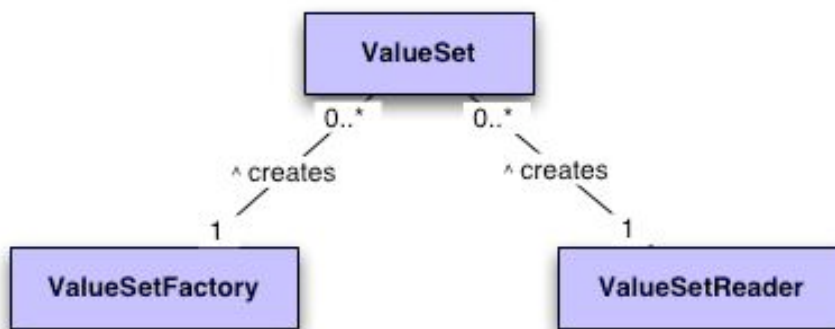
**Package: *graffity.gfx***

Class	Description
CurveControlView	This view serves to control the curves and their representation in the CurveViewContainer.
CurvePlotter	The CurvePlotter is the view that draws all the curves on the screen.
CurveView	A curve is a collection of two-tuples. These can be the result of a function or just data sets.
CurveViewContainer	A CurveViewContainer contains curves, which are somehow related to each other.

**Model**Figure 13: *graffity.gfx*

**Package: graffiti.pd**

Class	Description
ValueSet	A ValueSet contains an amount of values kept in persistent order.
ValueSetFactory	Creates different ValueSets.
ValueSetReader	Parses a log file (CSV) and extracts the ValueSets.

**Model***Figure 14: graffiti.pd*

## Project: Path Integrator

### *Packages: Overview*

Package Name	Description
pathintegrator	This package contains the main class. Only this class may be executed.
pathintegrator.pd	The problem domain classes of the path integrator.

### Model

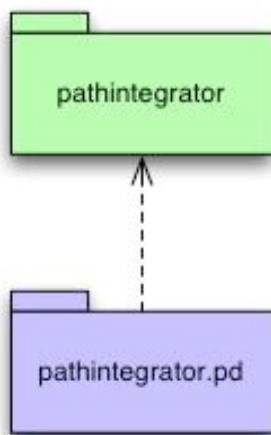


Figure 15: Path Integrator package structure

**Package: *pathintegrator***

Class	Description
PathIntegrator	The main class for all path integrators (pi). It creates the appropriate pi via the PathIntegratorFactory. The type is extracted from the command line.

**Model***Figure 16: pathintegrator*

**Package: pathintegrator.pd**

Class	Description
HistoryPathIntegrator	This pathintegrator uses an internal history to keep track of the speed changes over some time. This is useful when the commands don't arrive in the correct order as happens when using TCP/IP connections on some networks.
<i>PathIntegrator</i>	This is the base interface for all path integrators.
PathIntegratorFactory	Creates different pis. Currently two different types of PIs are available. PathIntegratorSimple PathIntegratorHistory
PathIntegratorImpl	The base implementation for PIs. Handles networking.
PathIntegratorStub	This is not a real PI but only a stub. The stub connects to the remote PI and retrieves the data from it.
SimplePathIntegrator	This pathintegrator is simple and therefore fast. But it has problems when the commands arrive in incorrect order. Because it has no history of the already performed steps, it will be erroneous when this type of problem occurs.

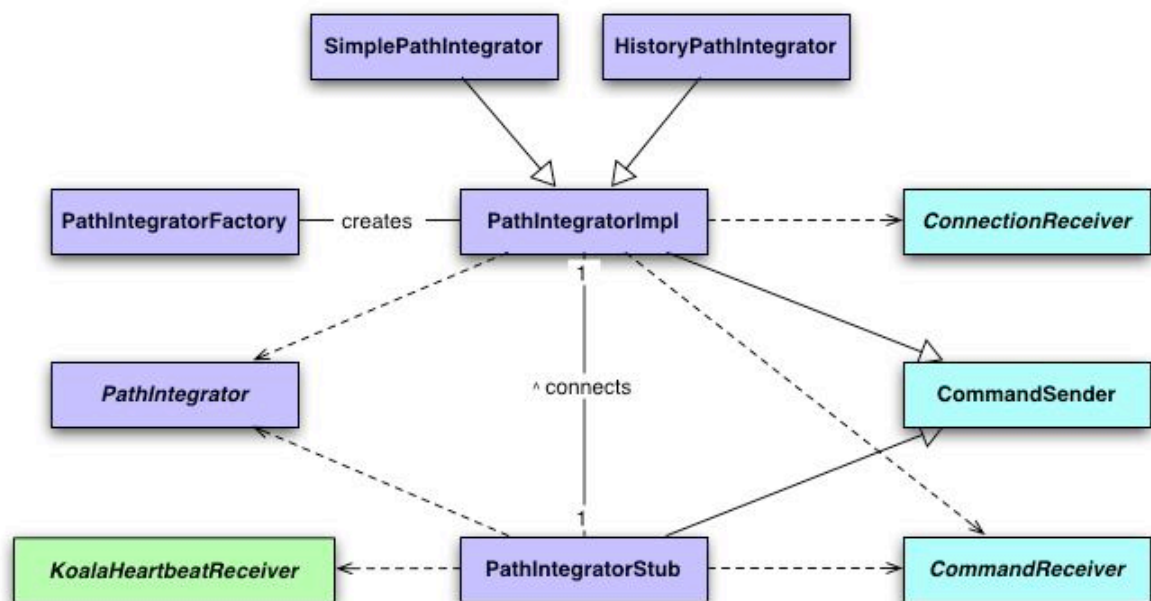
**Model**

Figure 17: pathintegrator.pd

## Project: Cortex

### *Packages: Overview*

Package Name	Description
cortex	This package contains the main classes. Only these may be executed.
cortex.alpha.gfx	Contains the graphical elements of the Alpha cortex.
cortex.alpha.pd	Contains the core components of the Alpha cortex.
cortex.beta.gfx	Contains the graphical elements of the Beta cortex.
cortex.beta.pd	Contains the core components of the Beta cortex.
cortex.gfx	Contains the basic GUI elements.
cortex.pd	Contains the basic core components.

### Model

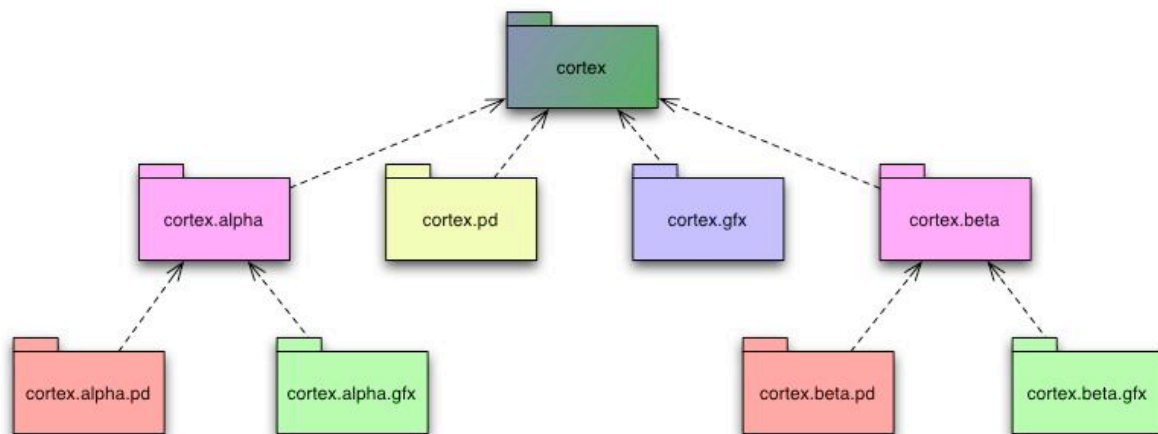
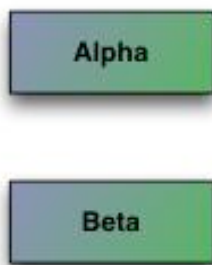


Figure 18: Cortex package structure



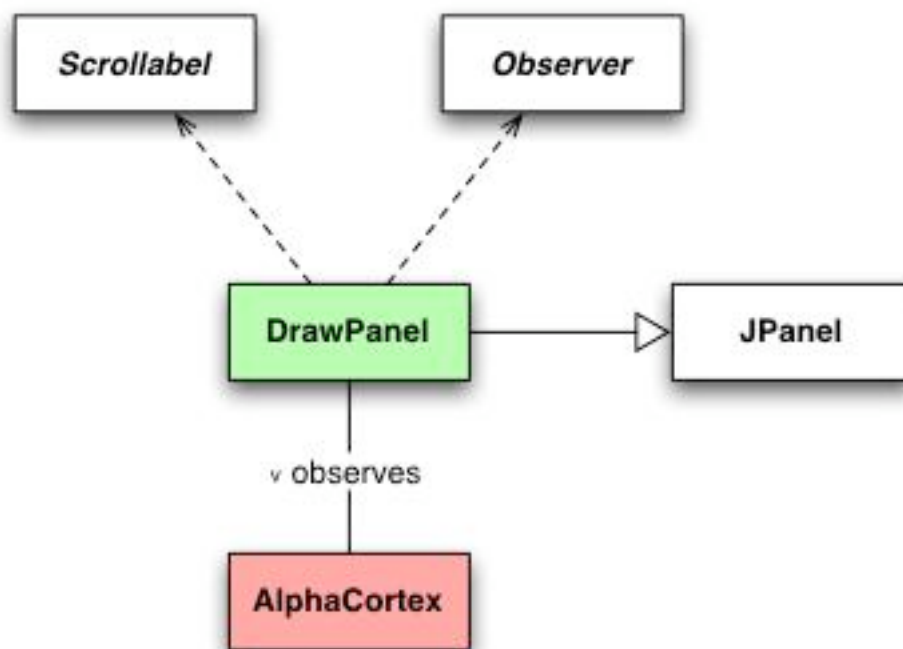
**Package: cortex**

Class	Description
Alpha	The alpha cortex is actually not blessed with any intelligence. Its purpose is to test the functionality of the underlying components (SensoryCortex, Cortex, PathIntegrator, optical system).
Beta	This cortex is a first approach to the idea of the dynamic network construction. Its purpose is to test and demonstrate the idea on a very simple base. It does not correct any errors but only constructs the network.

**Model***Figure 19: cortex*

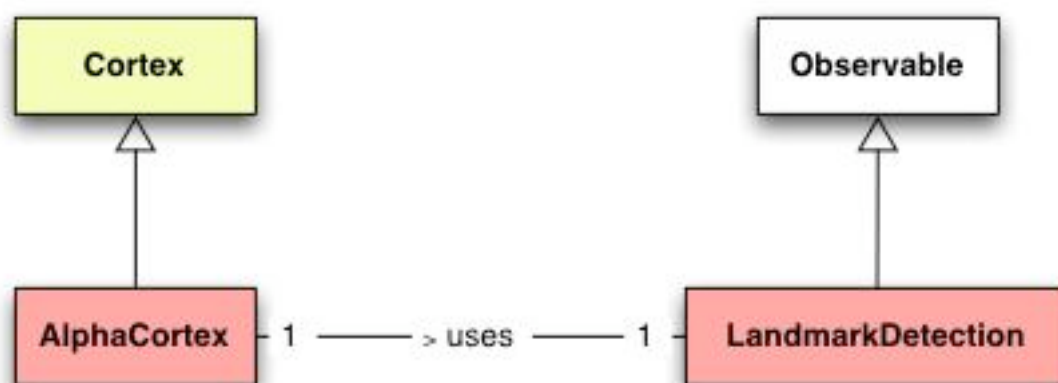
**Package: cortex.alpha.gfx**

Class	Description
DrawPanel	This view simply plots all recorded landmarks to the screen.

**Model***Figure 20: cortex.alpha.gfx*

**Package: cortex.alpha.pd**

Class	Description
AlphaCortex	The AlphaCortex class just contains the LandmarkDetection class.
LandmarkDetection	This is the central element of the Alpha cortex. It receives notifications from the optical system as soon as a landmark was perceived. It then calculates the absolute position and stores it.

**Model***Figure 21: cortex.alpha.pd*

**Package: cortex.beta.gfx**

Class	Description
BetaCortexView	This is the central view of the Beta cortex. It displays the graphical representations of the neurons (ovals) on the screen.
LandmarkOverView	This view shows all the perceived landmarks and their current value (see the document "The cortex - a first approach") for details.
NeuronDetailView	Shows the details for one neuron. These are the weights towards the landmarks and the location of the koala. It shows the calculated energy as well.
NeuronOverView	Displays all the generated neurons and their energies. See the document "The cortex - a first approach" for details.
PropertiesView	Displays some general information like the location of the koala or the lowest energy in the system.

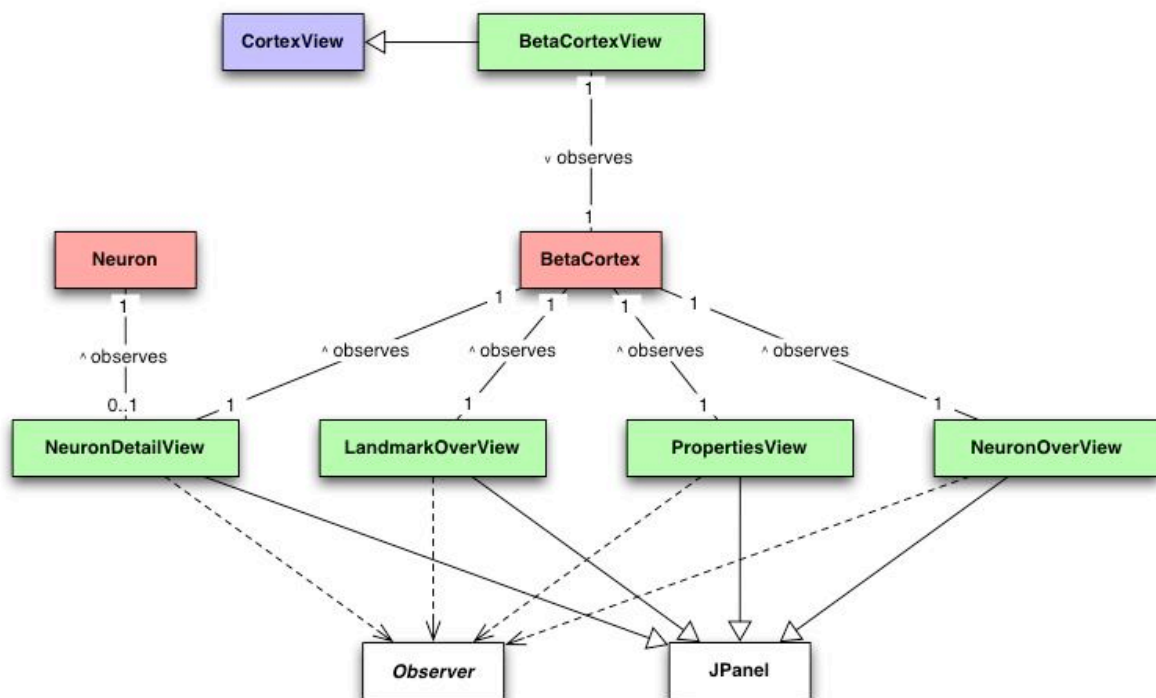
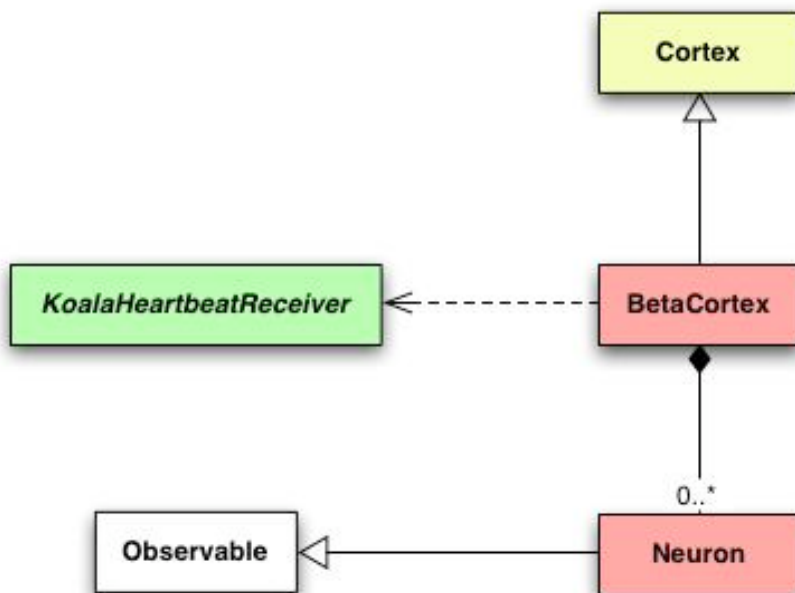
**Model**

Figure 22: cortex.beta.gfx

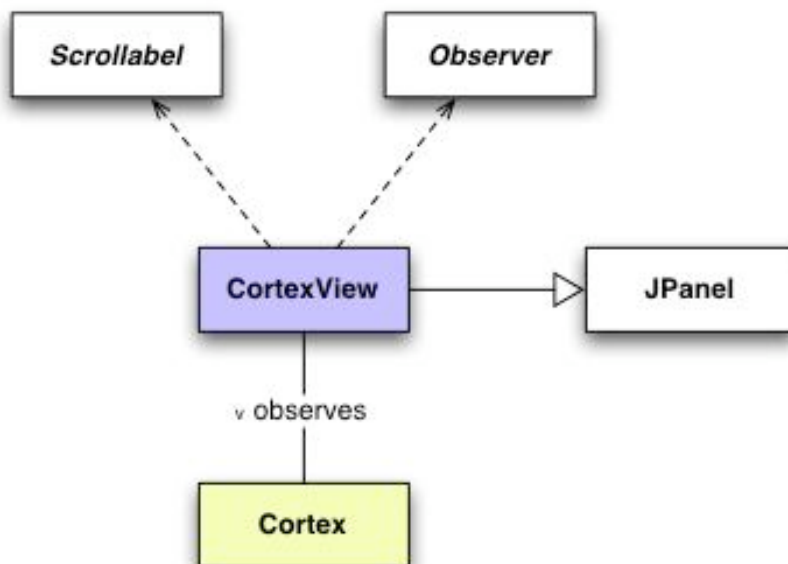
**Package: cortex.beta.pd**

Class	Description
BetaCortex	The main class of the Beta cortex. This class constructs the neurons and links them. It keeps track of the energy and changes the active neuron or dynamically creates new neurons.
Neuron	A single neuron. It stores the weights to the landmarks and to the location of the koala. It also provides functionality to calculate its energy.

**Model***Figure 23: cortex.beta.pd*

**Package: cortex.gfx**

Class	Description
CortexView	The base class for all cortex views.

**Model***Figure 24: cortex.gfx*

**Package: cortex.pd**

Class	Description
Cortex	The Cortex class is the super class for all cortices. It contains the SensoryCortex and a path integrator (stub).
SensoryCortex	The SensoryCortex is responsible for receiving and pre-processing sensory input from the remote components. Such as the location from the pi.

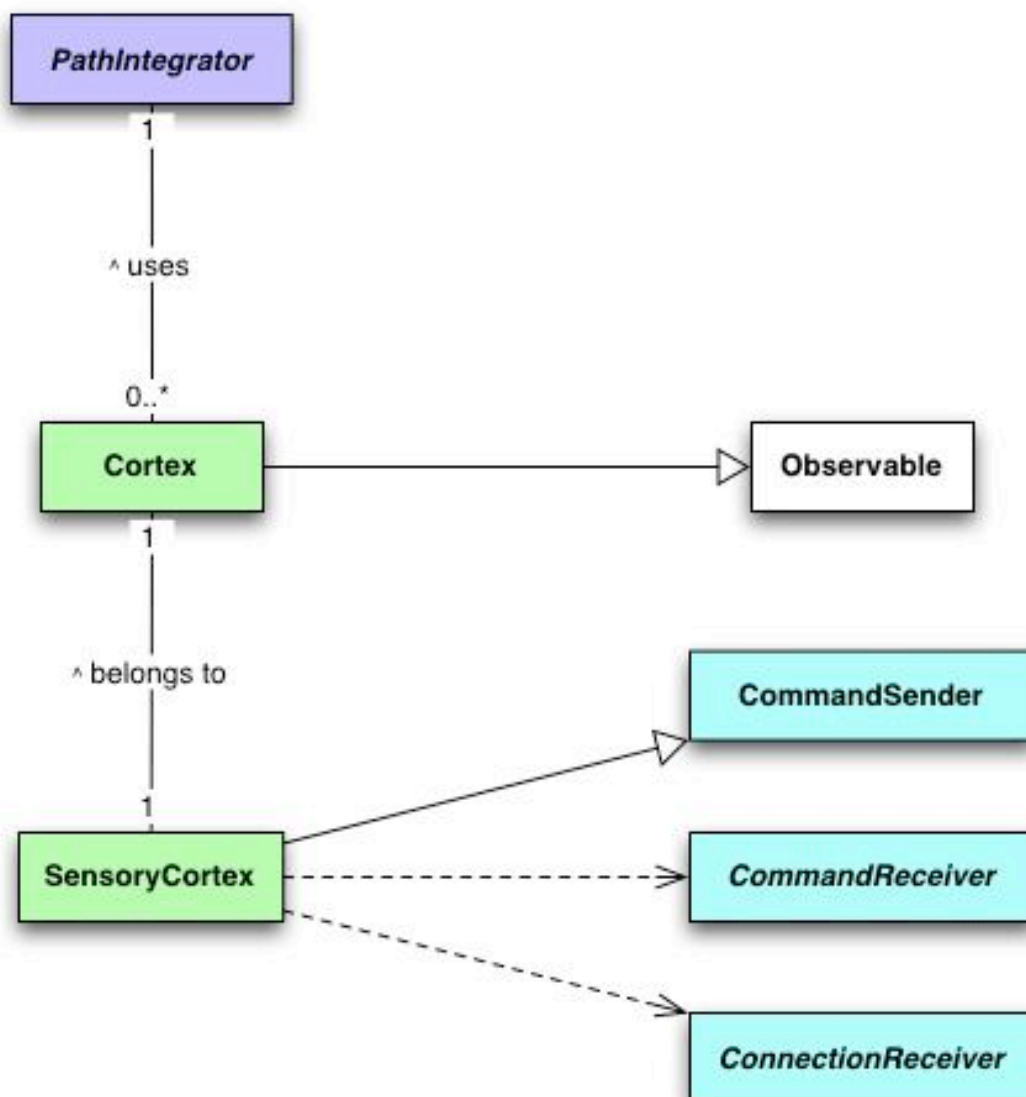
**Model**

Figure 25: cortex.pd