Eukalyptus

(Koala Environment Simulator)

Design: Models

Thabo Beeler, Institute for Applied Sciences, Rapperswil (HSR)

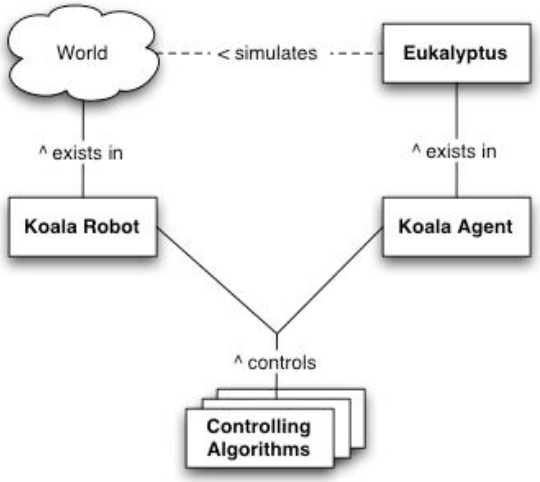
Revision History

Version	When	What	Who
0.1	23.04.2004	Initial version	Thabo Beeler
0.2	24.04.2004	Completed	Thabo Beeler
1.0	05.06.2004	Updated to Iteration 2	Thabo Beeler
1.1	07.06.2004	Continued updating	Thabo Beeler
1.2	12.06.2004	Added path integrator and cortex	Thabo Beeler

Table of contents

Introduction

Eukalyptus is a simulator written for the Koala robot. It should simulate the robots environment (the real world) as a virtual world. Of course not everything will be simulated. The important factors should be present, so that it is possible to develop and test controlling algorithms in a ‚safe' environment.
To the controlling algorithms, controlling the simulator should be as controlling the Koala robot.



This Document is meant to be used in combination with the generated JavaDoc . So we will concentrate on the concepts instead of the concrete implementations.

Terminology

It is sometimes essential to distinguish between the real robot and the virtual agent, and sometimes not. We will use the following terminology:

Koala agent / agent	= refers to the virtual Koala
Koala robot / robot	= refers to the real robot
Koala	= refers to both

Projects: Overview

Eukalyptus is part of the Koala project and can be divided in four subprojects. Each of these subprojects has its own namespace.

Subproject	Package	Short description
Eukalyptus	eukalyptus	The main project containing everything related to the simulator and the display.
Graffity	graffity	This tool visualizes log files.
Path Integrator	pi	The path integrators are used to estimate the position of the Koala.
Cortex	cortex	Everything related to map building and goal directed navigation.

Project: Eukalyptus

Packages: Overview

Package Name	Description
eukalyptus	This package contains the main classes. Only these may be executed.
eukalyptus.pd	The problem domain classes of the simulator.
eukalyptus.pd.koala	These classes describe the koala robot. Mainly interfaces of the sensors and actors.
eukalyptus.pd.koala.local	Implementation of the koala interfaces.
eukalyptus.pd.koala.remote	Stub classes, which allow to remotely display the Koala.
eukalyptus.pd.utils	Utility classes.
eukalyptus.net	Everything related to the network.
eukalyptus.gfx	Graphical representation of the core classes as well as GUI classes.
eukalyptus.test	Test classes.

Model

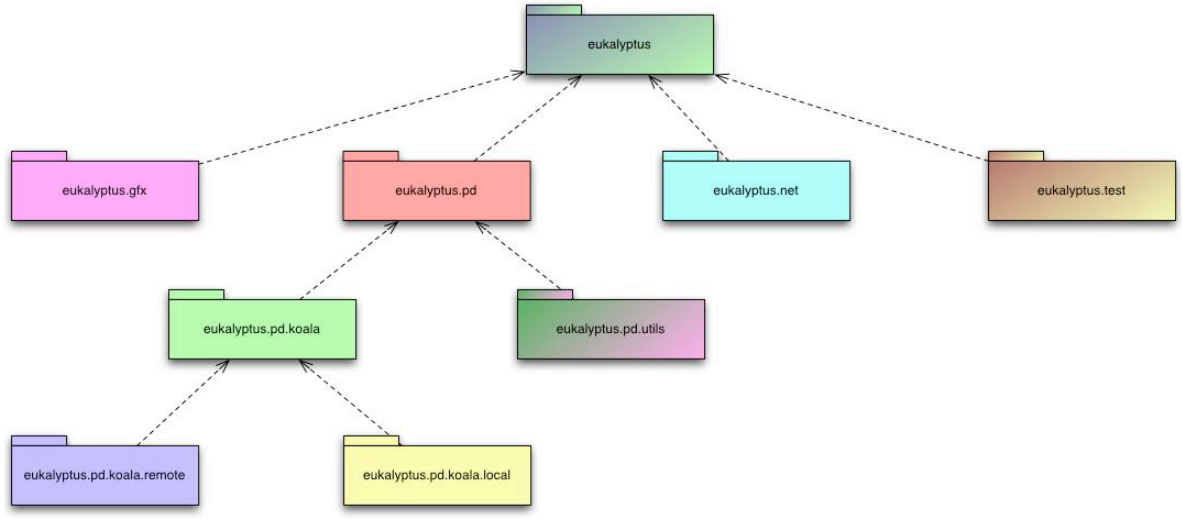


Figure 1: Eukalyptus package structure

In the following we will describe the contents of all the packages except the test package. We will use the colors to show the package of the classes.

Package: eukalyptus

Currently there are five classes in this package. All three may be executed.

Class	Description
Dispatcher	The dispatcher is needed to distribute incoming commands to multiple subscribers. So one can run the simulator and the Koala robot at the same time. Useful for calibration. See KoalaCalibrator.
Eukalyptus	The main application. This starts Eukalyptus.
KoalaCalibrator	This class is used to calibrate the simulator to get a representative simulation of the Koala robot. It sends a command to the Koala, waits for a given time period and then sends the stop command. See Dispatcher.
KoalaCommander	A class that is completely standalone. It communicates with the Eukalyptus Simulator over sockets (TCP/IP). It is used to steer the koala agent with the keyboard. It can also communicate whit the hardware robot.
KoalaCommunicator	A communication centre. It is standalone. It uses sockets to communicate with the koala agent. It can also communicate whit the hardware robot.

Model

Figure 2: Eukalyptus executable classes

Package: eukalyptus.pd

These are the core classes of the simulator.

Class	Description
Door	Represents a door. Doors can be color coded in the input image and are automatically parsed. Doors have two distinct states: open and closed.
InteractiveObject	Interactive objects are all objects on the map except the obstacles.
InteractiveObjectContainer	Contains all interactive objects of a given type.
InteractiveObjectFactory	This factory creates InteractiveObjects. The factory instantiates the objects and performs all the necessary steps. The created object is ready to be used. One should always create objects with the factory!
Landmark	Landmarks are interactive objects that can be recognized by the koala.
MagneticField	A magnetic field interferes with the magnetic compass of the koala agent.
Map	The digital representation of the static world. The representation was chosen to optimize performance.
MapMetaInfo	Contains additional information of a map. Like the used base ratio (size of one pixel in the real world)

Model

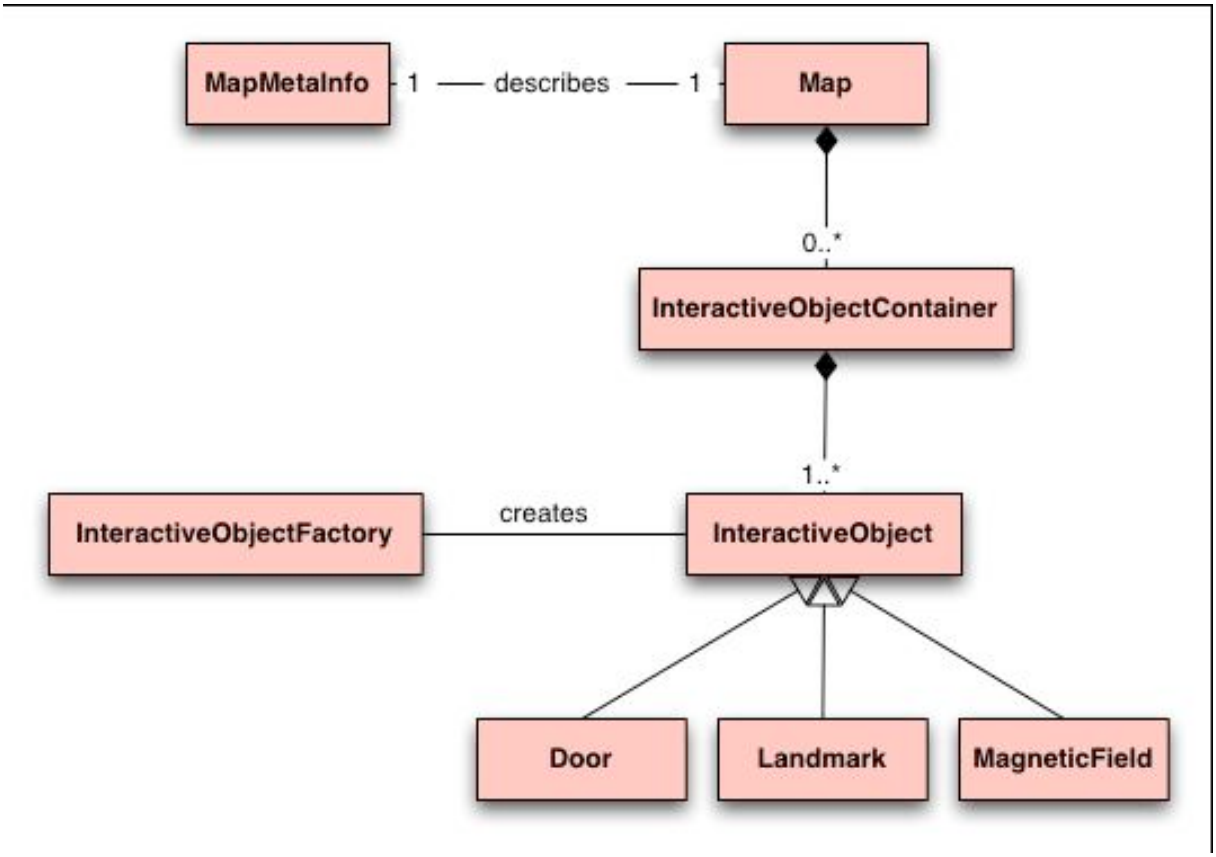


Figure 3: eukalyptus.pd

Package: eukalyptus.pd.koala

These are the core classes of the koala agent. Most classes are interfaces, which will be implemented either by the local implementation or by the remote stubs.

Class	**Description**
Cameras	Cameras is the camera module.
DriveControl	This device controls the movement of the koala. It is an active sensor.
Gyroscope	The gyroscope senses the orientation of the koala. It is a passive sensor.
Koala	This abstract class describes the physical properties of the koala, like the shape. It creates and initializes all the sensors.
KoalaHeartbeat	The heartbeat is a thread that periodically awakes and tells the koala to perform.
KoalaHeartbeatReceiver	A component, which wants to be periodically notified needs to implement this interface. It then can be attached to the KoalaHeartbeat.
MagneticCompass	The magnetic compass senses the north pole (upper screen). It is disturbed by the magnetic fields. It is a passive sensor.
Sensor	This is the base interface for every sensor. It declares base functionality for sending data.
Whiskers	The koala can sense its close environment with the whiskers. It possesses whiskers of different sizes at different positions on the surface.

Model

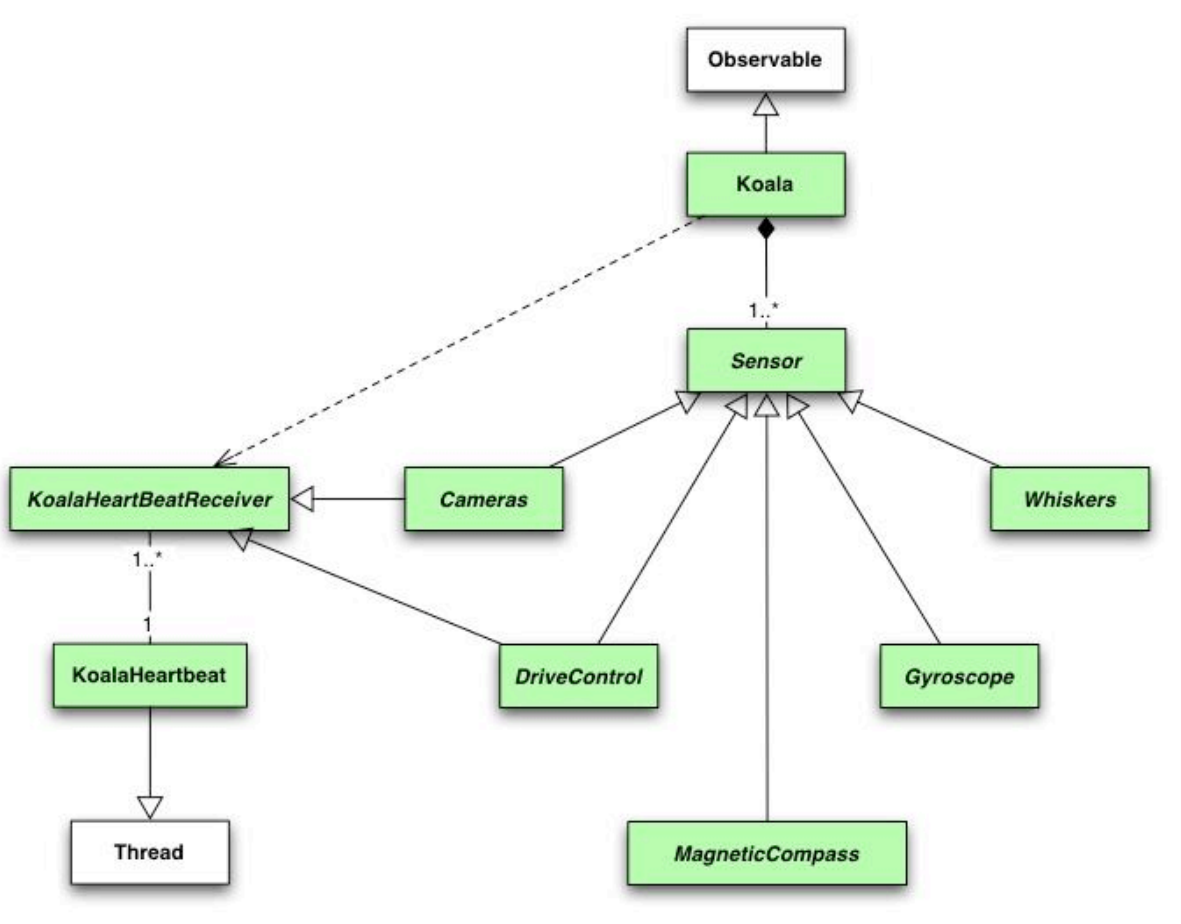


Figure 4: eukalyptus.pd.koala

Package: eukalyptus.pd.koala.local

Class	Description
ActiveSensorImpl	This is the base class for all active sensors. It implements basic tasks like registering subscribers and sending the data to them.
CamerasImpl	Cameras is the camera module. It controls both cameras. It is an active sensor, which performs once per heartbeat. It rotates the cameras and perceives landmarks.
DriveControlImpl	This device controls the movement of the koala. It is an active sensor.
GyroscopeImpl	The gyroscope senses the orientation of the koala. It is a passive sensor.
KoalaImpl	This class describes the physical properties of the koala, like the shape. It creates and initializes all the sensors.
MagneticCompassImpl	The magnetic compass senses the north pole (upper screen). It is disturbed by the magnetic fields. It is a passive sensor.
PassiveSensorImpl	A passive sensor only senses its environment, when requested. Passive sensors accept commands and return the result.
SensorImpl	This is the base class for every sensor. It provides base functionality for sending data.
WhiskersImpl	The koala can sense its close environment with the whiskers. It possesses whiskers of different sizes at different positions on the surface.

Model

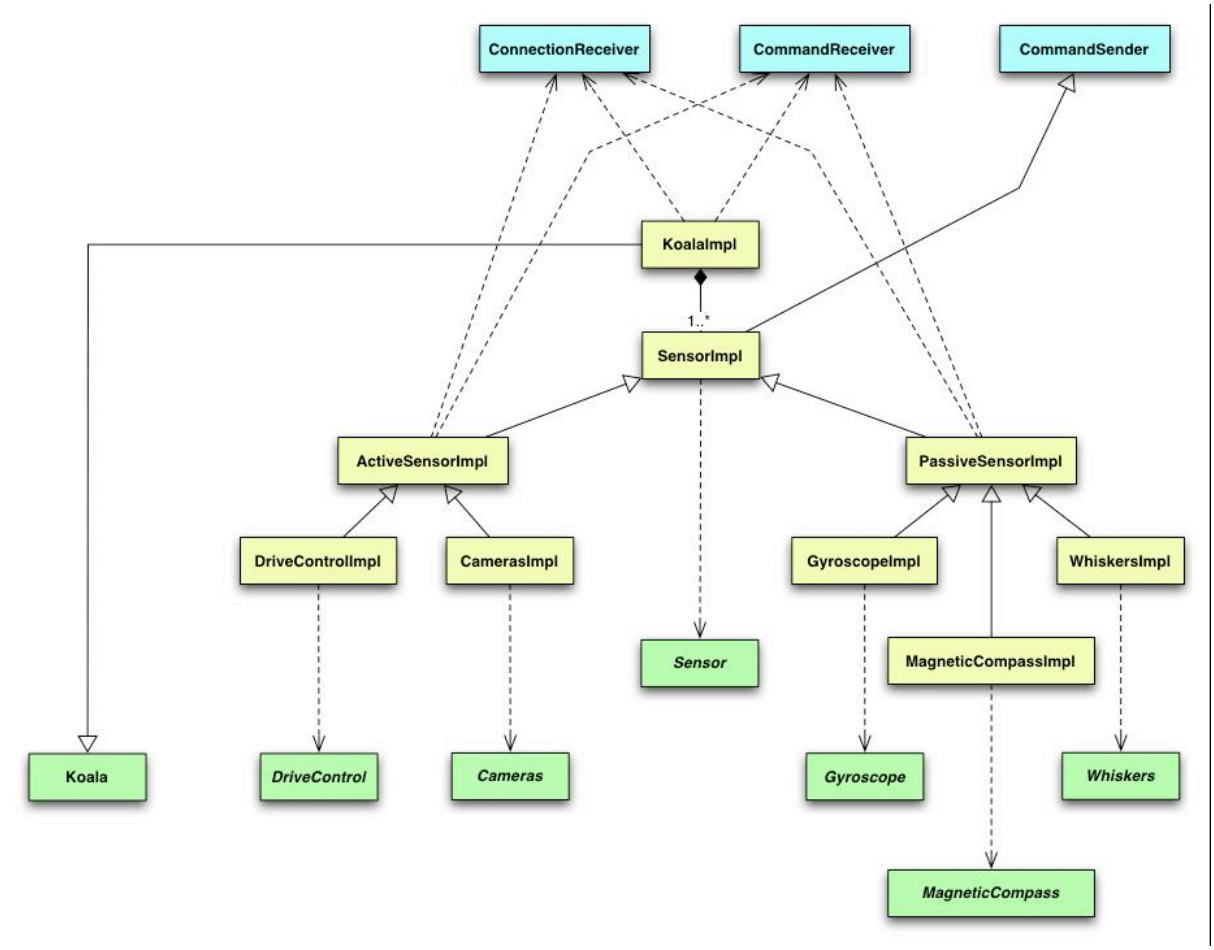


Figure 5: eukalyptus.pd.koala.local

Package: eukalyptus.pd.koala.remote

Class	Description
ActiveSensorStub	This is the base class for all active sensors. In contrast to the PassiveSensorStub this class needs to open two sockets.
CamerasStub	This stub connects to the landmark detection process. It will be notified whenever a new landmark has been detected.
DriveControlStub	This device controls the movement of the koala. This stub connects to a path integrator.
GyroscopeStub	The gyroscope senses the orientation of the koala.
KoalaStub	This class describes the physical properties of the koala, like the shape. It creates and initializes all the sensor stubs.
MagneticCompassStub	The magnetic compass senses the north pole (upper screen). It is disturbed by the magnetic fields.
PassiveSensorStub	A passive sensor stub requests periodically the sensed data from its remote sensor.
SensorStub	This is the base class for every sensor stub. It provides base functionality for sending data.
WhiskersStub	The koala can sense its close environment with the whiskers. It possesses whiskers of different sizes at different positions on the surface.

Model

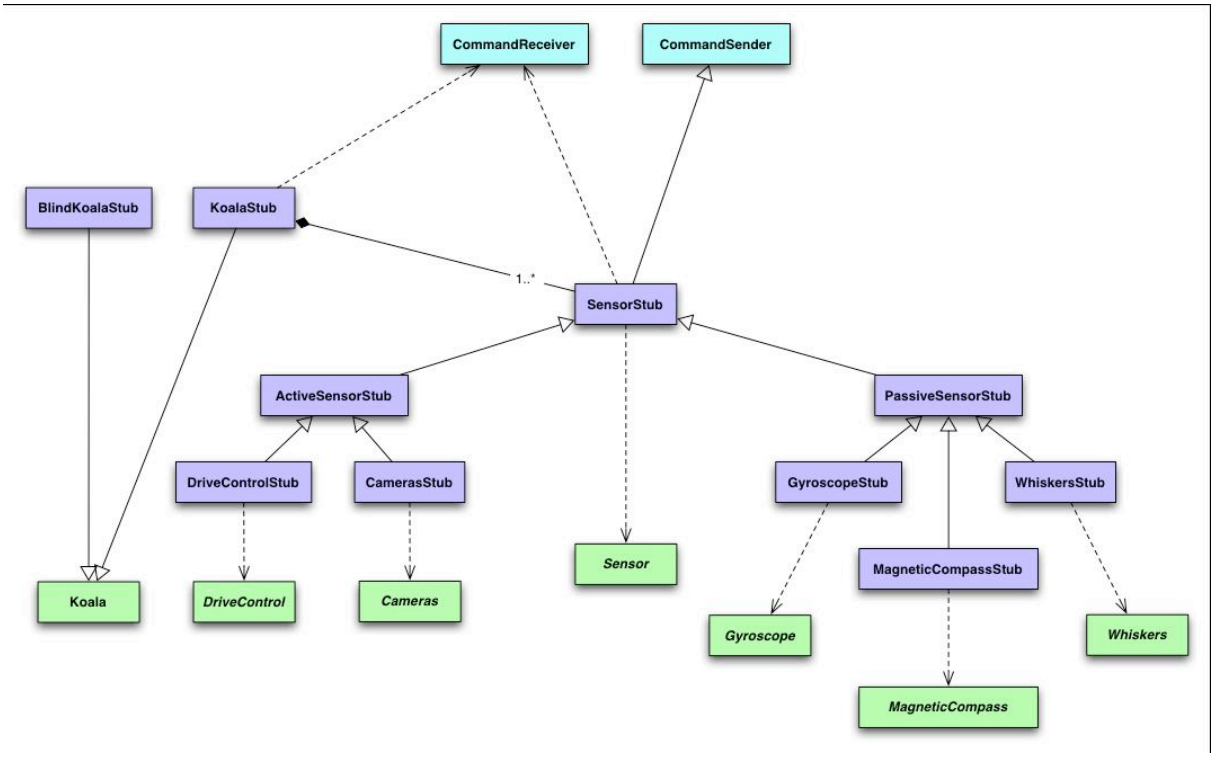


Figure 6: eukalyptus.pd.koala.remote

Model

The following figure shows all classes of the koala packages together.

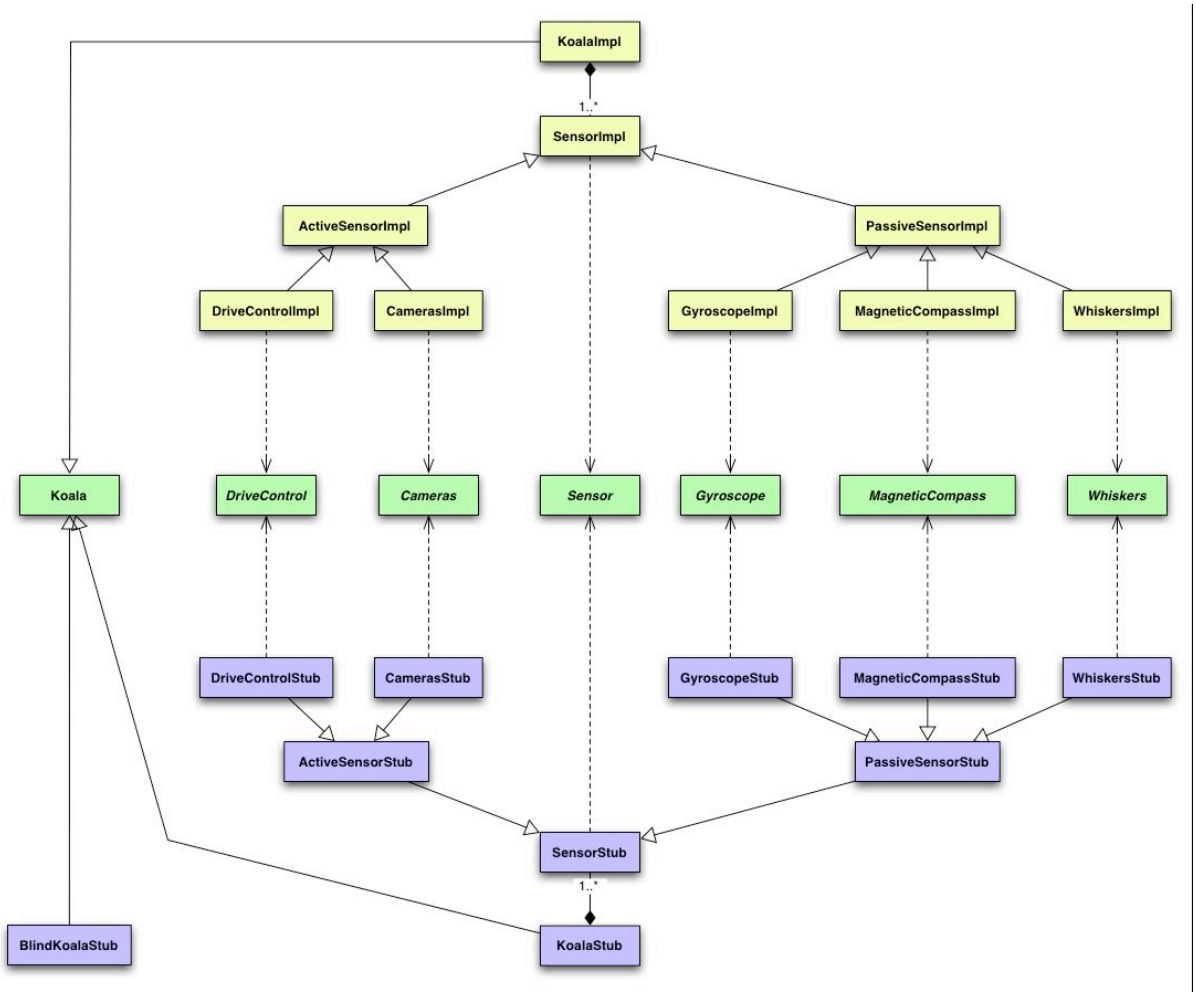


*Figure 7: eukalyptus.pd.koala.**

Package: eukalyptus.pd.utils

This package contains utility classes, which did not fit anywhere else.

Class	Description
ConnectionLogModule	Used to log network traffic.
LogModule	Used to log data periodically to files.
VideoStreamer	Reads the YUV 4:2:0 P stream from the video base and converts in to RGB.

Model

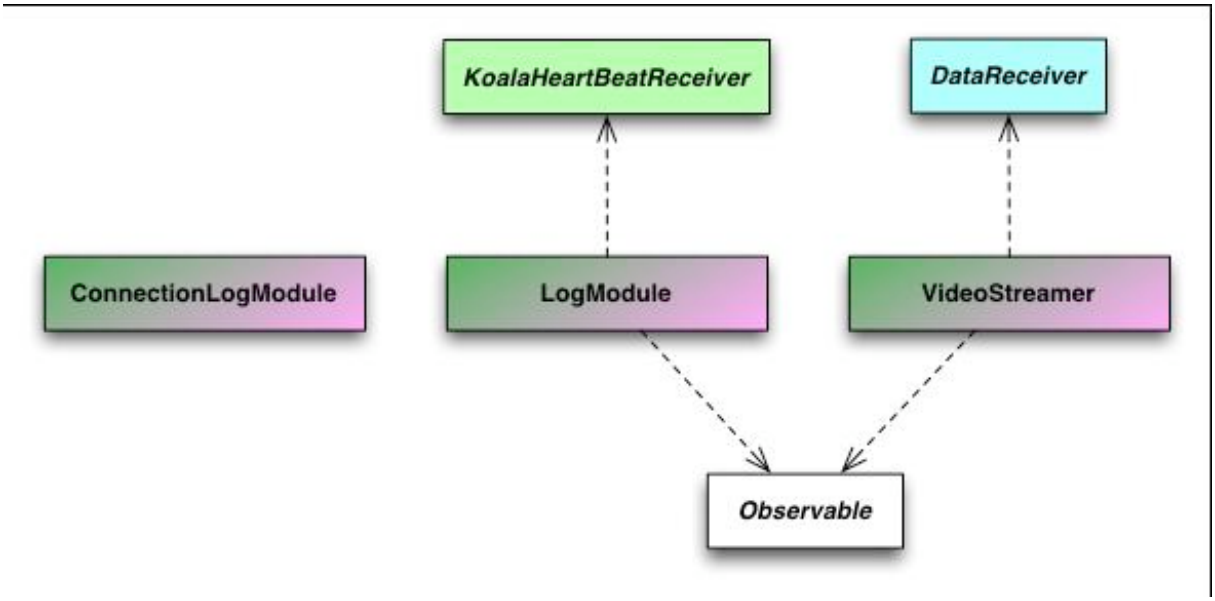


Figure 8: eukalyptus.pd.utils

Package: eukalyptus.net

Everything related to networking.

Class	Description
CommandReceiver	Every component that wants to receive commands needs to implement this interface.
CommandReceiverThread	Every *CommandReceiver* needs a CommandReceiverThread. The thread will continuously read from a socket and pass the extracted commands to the *CommandReceiver*.
CommandSender	This class provides functionality to send commands.
ConnectionReceiver	Every component that wants to accept connections needs to implement this interface.
ConnectionReceiverThread	Every *ConnectionReceiver* needs a ConnectionReceiverThread. This thread waits on a given port for incoming connections.
DataReceiver	Every component that wants to receive data needs to implement this interface.
DataReceiverThread	Every *DataReceiver* needs a DataReceiverThread. The thread will continuously read from a socket and pass the raw data to the *DataReceiver*.
PersistentSocket	This class is a wrapper for the ordinary Socket class of the java.net package. The whole Eukalyptus needs to be robust to downtimes of the distributed components. This socket provides this robustness, as it tries to reconnect when the connection breaks.
Protocol	Implements the communication protocol. See the document „Network communication“ for more details. All methods are static.

Model

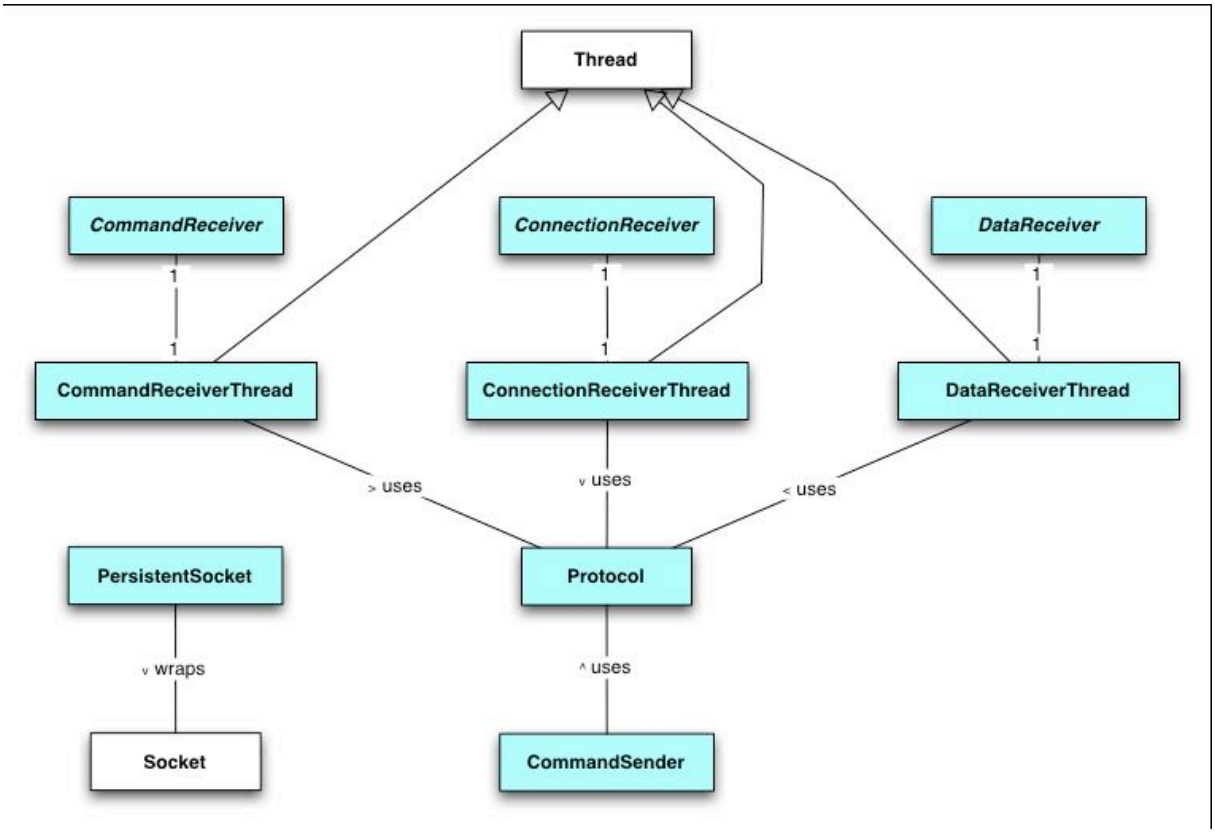


Figure 9: eukalyptus.net

Package: eukalyptus.gfx

Everything related to graphics.

Class	Description
ConnectionLogModuleView	Provides a view and controller for the ConnectionLogModule.
DoorEditorView	The editor panel for the doors.
DoorView	The graphical representation of a door on the map.
EditorPanel	Every editor needs to implement this interface.
InteractiveObjectView	The view on an interactive object. These views are drawn to the map.
InteractiveObjectViewContainer	The view on an interactive object container.
InteractiveObjectViewFactory	InteractiveObjectViews should be created with this factory.
KoalaErrorView	The panel that provides access to the error rates of the koala.
KoalaPropertiesView	This panel shows the properties of the agent. It is an observer that can be turned on or off. (Performance)
KoalaShadowEditorView	This panel allows the user to organize so-called shadows. Shadows are views of BlindKoalaStubs.
KoalaShapeView	This label is a detailed representation of the koala agent. It shows all the sensors. It is an observer that can be turned on or off.
KoalaStatusView	Only in display mode. Shows status information of the Koala.
KoalaView	The graphical representation of the koala agent on the map.
LandmarkEditorView	The editor panel for the landmarks.
LandmarkView	The graphical representation of a landmark on the map.
LogModuleView	The editor panel for the log module.
MagneticFieldEditorView	The editor panel for the magnetic fields.
MagneticFieldView	The graphical representation of a magnetic field on the map.
MapFileChooserFilter	Filter for the load map/save map dialogs. Filters .emp files.
MapImageChooserFilter	Filter for the import map dialog. Filters .gif/.jpg files.
MapView	The graphical representation of the map itself.
Ruler	The rulers are used to visualize the dimensions of the map.
WorldPane	A scroll pane that contains the WorldView. It is used to track the movement of the Koala.
WorldView	This class contains all the map-related views.

	These are the MapView, the KoalaView and all the InteractiveObjectViews.
VideoStreamerView	Only in display mode. Shows the images streamed from the cameras of the Koala.

Model