Eukalyptus

(Koala Environment Simulator)

Project summary

Thabo Beeler, University of Applied Sciences, Rapperswil (HSR)

Revision History

Version	When	What	Who
0.1	02.06.2004	Initial version	Thabo Beeler
0.2	26.06.2004	First revision. Added changes suggested by Jörg Conradt	Thabo Beeler

Table of contents

Introduction

This file provides an overview over the Eukalyptus project. It does not explain the design or implementation details. Please refer to the document „Design“ for details.
It also is not a user documentation. Please refer to the document „User Manual“ that for.

Motivation

Goal-directed navigation is a difficult task. Testing it in the real world is time-consuming and, if the robot crashes, money-consuming as well.
To avoid these difficulties, the task can be simulated in a virtual environment instead. These were the main thoughts, which lead to the creation of Eukalyptus, the Koala environment simulator.

The simulator has many more advantages:

- Different map building algorithms can be compared.
- Learning algorithms, such as neuronal networks, can be trained in the simulation and applied to the real robot later on.
- Simulation can be performed on many different machines in parallel.
- The simulation can be run without observation from the operator.
- Long-time simulation is possible.
- The simulation can be in real-time, faster or slower than the “real world”

The name of the developed robot is Koala – the name of the simulator therefore Eukalyptus.

Terminology

Some data, such as live video streaming, cannot possibly be provided by the simulator. So it is sometimes essential to distinguish between the real robot and the virtual agent. 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

Overview

The task was to design the simulator in a way, that the cortex does not notice any differences to the Koala robot. So these two have to remain completely interchangeable.

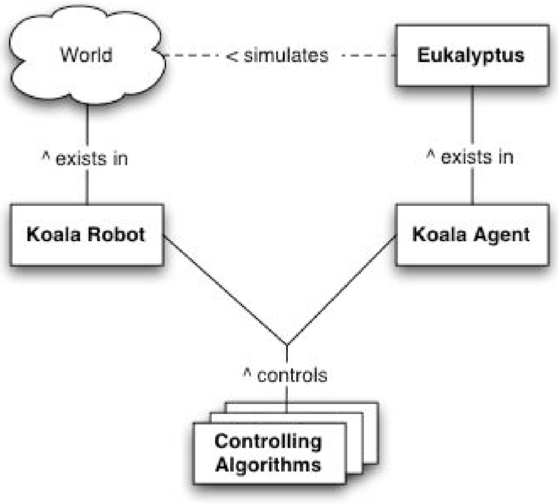

Fig. 1: Base domain model

To achieve this, we use distributed computing. All components communicate over TCP/IP with a special protocol. See the document „Network communication“ for a detailed specification of the communication protocol.
But Eukalyptus is more than just a simulator. It can be used as a remote display of the real robot as well. So one could monitor and control the Koala from any place on the world using the Internet.

To remain unaffected by the underlying hardware, the complete project was realized in Java.

Eukalyptus – The simulator

The simulator needs to simulate two different concepts.

The Koala robot
The Environment

The Koala agent

This software concept tries to simulate the behavior of the real Koala robot. Every component in the real world is erroneous, or better said underlies unforeseeable many physical laws. To simulate this, every software component will have to simulate an error as well.
A component has therefore a lower and an upper variance. By setting them empirically, we can reflect systematic as well as stochastic errors.
Every sensor has different error functions. Please refer to the document „Error specification" for a detailed description.

The Koala has the following sensors

Gyroscope
Magnetic compass
Drive control
Whiskers
Cameras

Gyroscope

The gyroscope is a device, which returns the orientation of the Koala, relative to a preset orientation.

Magnetic compass

The compass is a device, which returns the orientation of the Koala, relative to the North Pole.

Drive control

This device controls the motion of the Koala. It controls the wheels on both sides independently, so the Koala is able to drive almost any curvature. It can even rotate on the spot.

Whiskers

These are devices responsible to sense the closer environment. They work similar to the whiskers of cats or rats. But in contrast to the natural model, these whiskers may only indicate touched or not.
To sense different distances, whiskers are always applied in sets of different sizes.

Cameras

The Koala robot has two cameras to perceive landmarks. The Koala agent simulates just one camera.

The Environment

The Environment consists of a map and so called interactive objects.

The map

The map contains a representation of the obstacles in a room. Eukalyptus is able to import normal gif image files.
The internal representation was chosen to be pixel-based because of various reasons. See the document „Requirements: Map“ for the detailed reasons.

Interactive objects

The environment of the Koala contains several interactive objects.

Landmarks
Magnetic fields
Doors

Landmarks

These are marks the Koala may perceive and recognize. They support the map building process.

Magnetic fields

Fields emitted by electronic devices such as computers. They interfere with the magnetic compass of the Koala.
See the document „Design: Error Specification“ for a description on how they interfere.

Doors

Doors might be open or closed at a time.

Eukalyptus – The display

When running in display mode, the Eukalyptus does not perform any simulation. Instead it connects to a remote Koala, a path integrator and a vision base.
It then retrieves the remote data and displays it.

This was achieved with the stub/skeleton pattern used in the Java RMI as well. There is a stub for every sensor in the Koala, which connects to the real sensor and requests the data.

The drive control stub and the camera stub are somehow special.
The drive control stub does not connect to the drive control but to the path integrator instead.
This is so, because we want to display the position of the Koala, and not its speed.
The camera stub connects to the vision base as well as to the camera sensor.
The connection to the sensor is used to control the pan and tilt angles of the camera.
The connection to the vision base is used to receive the perceived landmarks. This because the evaluation of the camera pictures is a performance consuming process and will be run on a different computer.

Additional tools

Eukalyptus consists of some additional tools.

Dispatcher
Graffiti
KoalaCalibrator
KoalaCommander
KoalaCommunicator

Dispatcher

The purpose of this application is to transparently dispatch incoming commands to multiple clients. It is used to control both, the Koala agent and the Koala robot at the same time.

Graffity

This tool is useful to display and analyze the log files Eukalyptus produces. It interprets the files and displays the contained data as curves on the screen.
The same calculations could easily be done in a spreadsheet calculation application like StarOffice , but it is more convenient to use Graffity.

KoalaCalibrator

To reproduce the movement of the Koala robot in the agent most accurate, one needs to empirically evaluate the systematic and stochastic error rates.
This tool sends a command (speed command) to the Koala for a given time. After that it sends the stop command.

KoalaCommander

Useful when one does not have a joystick at hand. It enables a user to control the Koala with the keyboard. It is not convenient though, so one should use the joystick whenever possible.
The keys are:

i	= move forward
k	= move backward
l	= turn right
j	= turn left
space	= stop

KoalaCommunicator

Visualizes network traffic. The user can manually type in commands and send them to the Koala. Useful to see what happens on the communication layer, and to manually test functions of the Koala without having to start the Eukalyptus display.