DTU

Before we start:

If you feel ill, go home

Keep your distance to others

Wash or sanitize your hands

Disinfect table and chair

Respect guidelines and restrictions

DTU

Lazaros Nalpantidis & Evangelos Boukas

Software Architectures for Autonomous Systems

Outline

- Introduction to the Course
- Introduction to Software Architectures
  - Operation Paradigms
  - Hardware Abstraction Layer & Middleware
  - Characteristics of Software Architectures
- Summary

Outline

- Introduction to the Course
- Introduction to Software Architectures
  - Operation Paradigms
  - Hardware Abstraction Layer & Middleware
  - Characteristics of Software Architectures
- Summary

Overview

- 5 ECTS
- Technological Specialization course for the MSc. in Autonomous Systems
- Elective for other DTU MSc programmes

- All course lectures and activities:
  - Zoom (lectures) Discord (excercices and support)
  - Mondays 13:00-17:00

Lecturers

- **Lecturers:**
  - Evangelos Boukas, Associate Professor, build. 326, room 020.

  - Lazaros Nalpantidis, Associate Professor, build. 326, room 018.

- **Teaching Assistants:**
  - Rasmus Eckholdt Andersen, build. 326, room 110.

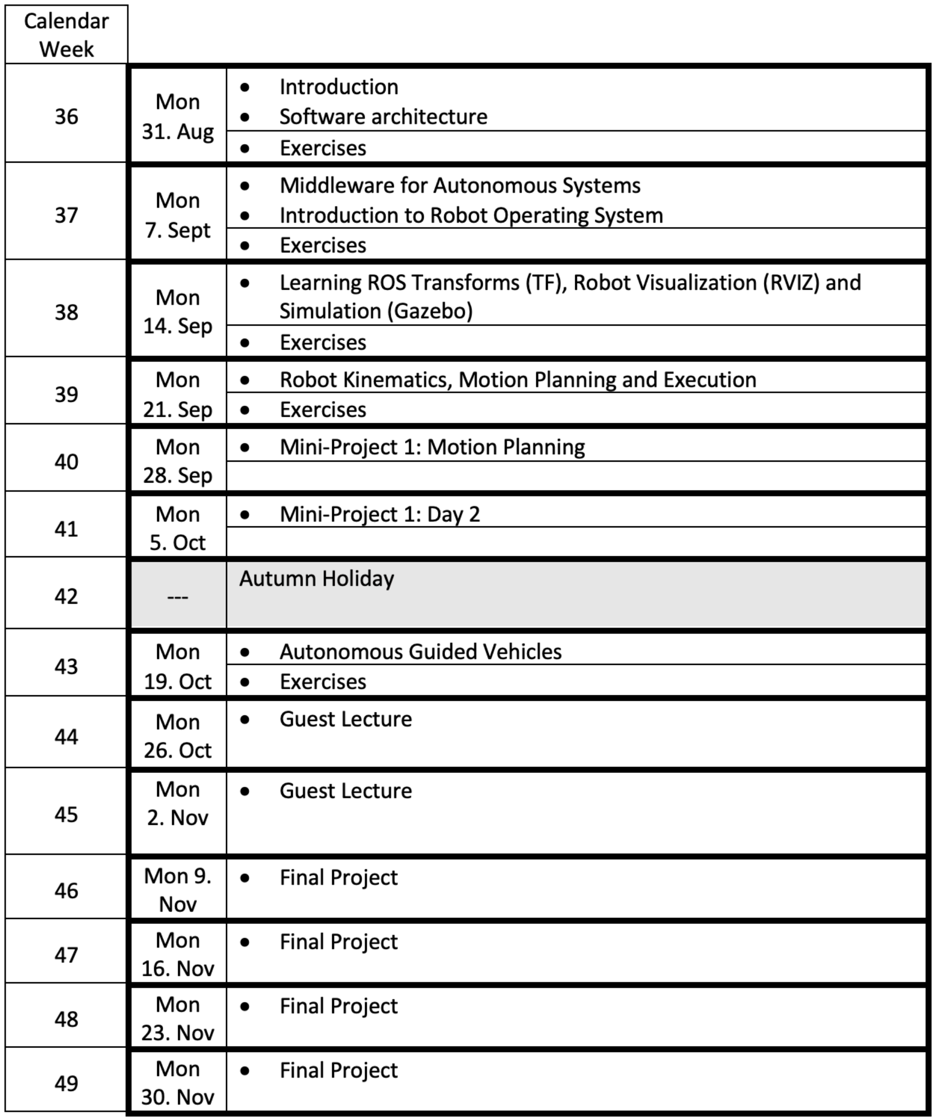
  - Ronja Güldenring, build 326, room 030.

Learning Objectives

A student who has met the objectives of the course will be able to:

- Design components for control of autonomous systems
- Choose components in ROS
- Understand installation and configuration of ROS
- Explain differences in hard and soft real-time
- Explain the control of a robot arm using ROS
- Find errors in process communication
- Verify the timing of software for autonomous systems
- Document software for autonomous systems
- Deploy autonomously navigating robotic agents
- Re-use High-level perception systems
- Simulate autonomous robots

Course Timeline



Calendar Week		
36	Mon 31. Aug	• Introduction • Software architecture • Exercises
37	Mon 7. Sept	• Middleware for Autonomous Systems • Introduction to Robot Operating System • Exercises
38	Mon 14. Sep	• Learning ROS Transforms (TF), Robot Visualization (RVIZ) and Simulation (Gazebo) • Exercises
39	Mon 21. Sep	• Robot Kinematics, Motion Planning and Execution • Exercises
40	Mon 28. Sep	• Mini-Project 1: Motion Planning
41	Mon 5. Oct	• Mini-Project 1: Day 2
42	---	Autumn Holiday
43	Mon 19. Oct	• Autonomous Guided Vehicles • Exercises
44	Mon 26. Oct	• Guest Lecture
45	Mon 2. Nov	• Guest Lecture
46	Mon 9. Nov	• Final Project
47	Mon 16. Nov	• Final Project
48	Mon 23. Nov	• Final Project
49	Mon 30. Nov	• Final Project

Course Format (1/3)

- **Lectures:**
  - All course lectures will be streamed online using **Zoom**. Students can pose questions or use the chat to communicate with the lecturer. All lectures will be broadcasted through the same link.

Course Format (2/3)

- **Lectures:**
  - All course lectures will be streamed online using **Zoom**. Students can pose questions or use the chat to communicate with the lecturer. All lectures will be broadcasted through the same link.
- **Exercises:**
  - Students will work on all course exercises as groups.
  - The course TAs will be available for questions on **Discord** during the class hours; reach them by writing to the "# general" channel.

Course Format (3/3)

- **Lectures:**
  - All course lectures will be streamed online using **Zoom**. Students can pose questions or use the chat to communicate with the lecturer. All lectures will be broadcasted through the same link.
- **Exercises:**
  - Students will work on all course exercises as groups.
  - The course TAs will be available for questions on **Discord** during the class hours; reach them by writing to the "# general" channel.
- **Group Formation:**
  - students need to form groups of 4 people.
  - The formed groups will work together on all exercises, the mini-Project and the Final Project during the course.
  - Group formation starts from the start of the course and needs to be finalized by the end of the 2nd lecture (7. Sept). Students form groups by themselves and register them in the online excel file.

Important Dates

7. September 2020	Deadline for formation of student groups. Students form groups of 4 people to work together on the exercises and on the course projects. Students register their groups in online excel file.
2. November 2020	Announcement from the teachers of the Final Project description.
30. November 2020	Hand-in of group report of Final Project.
2. December 2020	Information about non-qualification for participating in the exam by email to the members of groups with inadequate reports.

Exams / Evaluation

- In order to successfully complete the course, you need to:
  - Submit a group report about your final Project.
    - Final Project:
      - students are expected to work in their groups of 4 people
      - to solve a bigger assignment that will be given to them.
      - the outcome of the Final Project is a report of 10±2 pages that includes a link to a video demonstrating the group's main achievements.
      - A positive evaluation (pass/fail) of the report is mandatory for the group members to participate in the final exam.
  - Participate in an individual written exam
    - You get a Pass or Fail result.

Outline

- Introduction to the Course
- Introduction to Software Architectures
  - Operation Paradigms
  - Hardware Abstraction Layer & Middleware
  - Characteristics of Software Architectures
- Summary

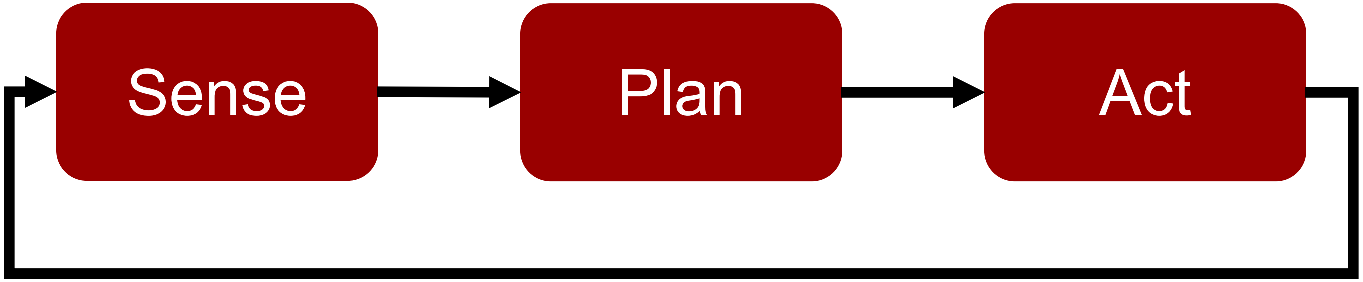
Operation Paradigms / Software Architectures

- A high-level model of how an Autonomous System operates

- Example from Robotics: **Robotic Paradigm**
  - Robot software architectures can be built by combining 3 primitive operations:
    - Sense
    - Plan
    - Act

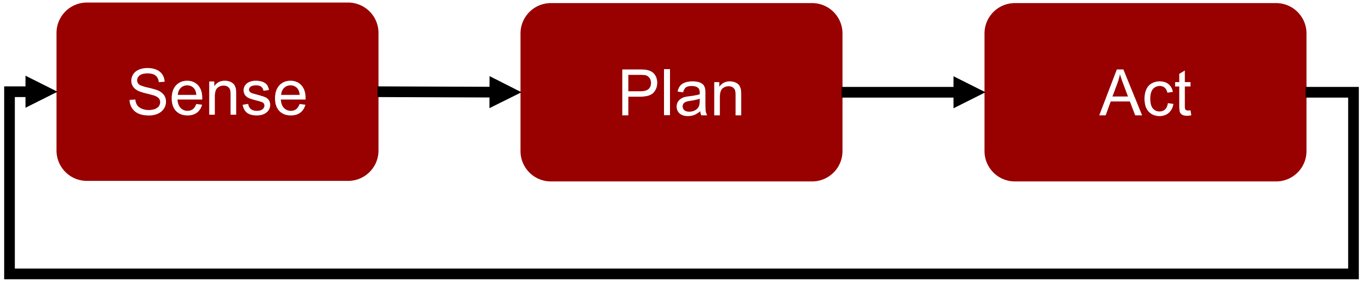
Operation Paradigms / Software Architectures

- Hierarchical (deliberative) paradigm



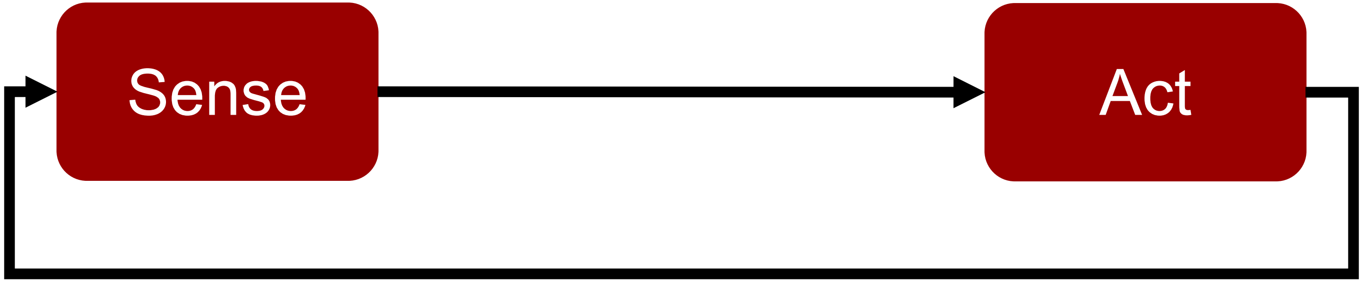
Operation Paradigms / Software Architectures

- Hierarchical (deliberative) paradigm
  - – First introduced with the Shakey Robot in 1966-1972.
  - – A “world model” is needed that fully describes the operation environment.
  - – Planners can be formilized and used to provide next actions.



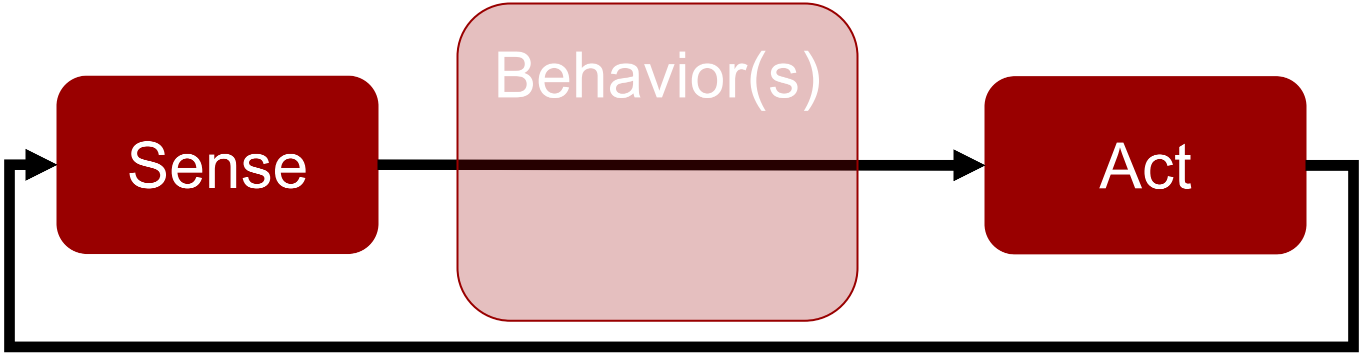
Operation Paradigms / Software Architectures

- Reactive paradigm



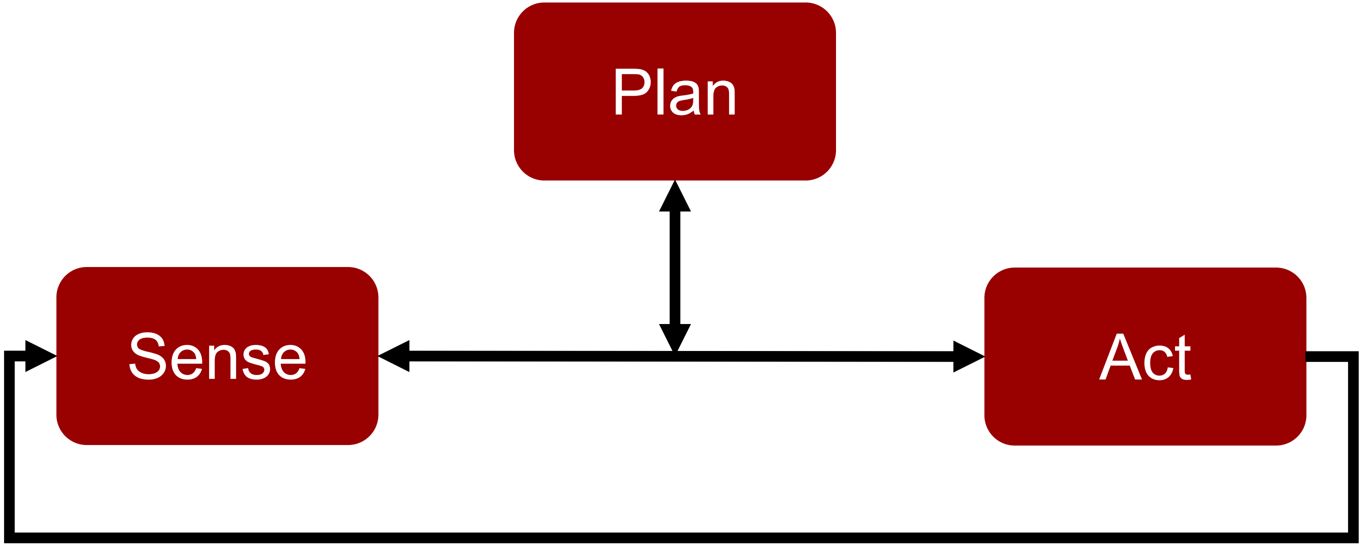
Operation Paradigms / Software Architectures

- Reactive paradigm
  - – Biologically inspired like instincts/reflexes
  - – No need for world models and model updates
  - – Behavior-based robotics (there can be multiple behaviors)



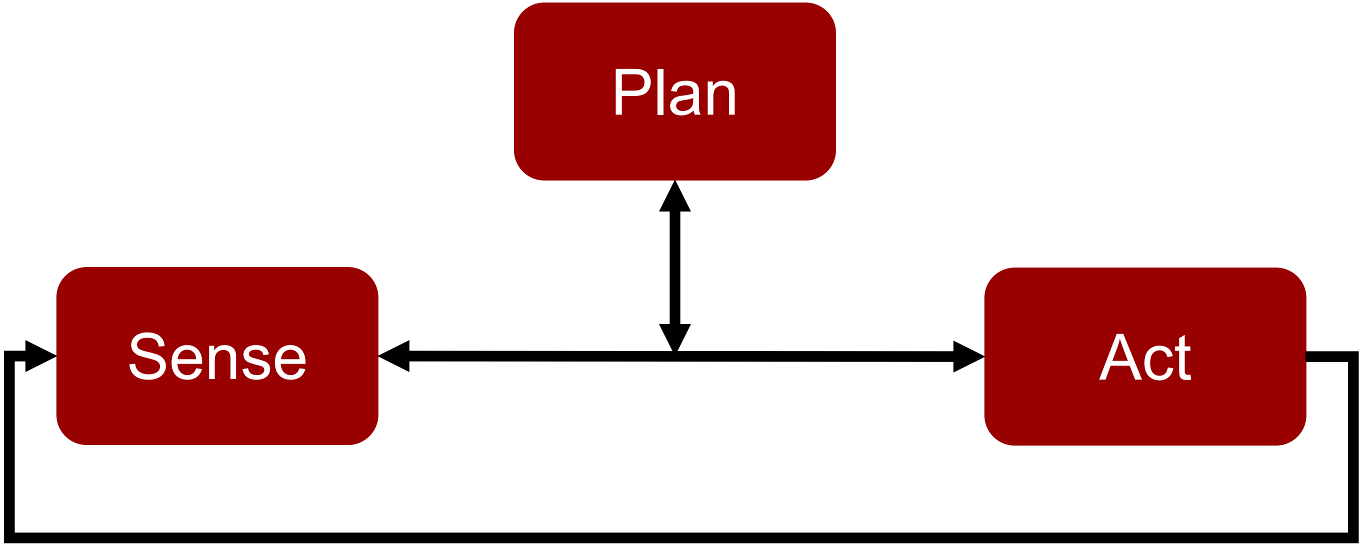
Operation Paradigms / Software Architectures

- Hybrid paradigm / Three Layer Architecture



Operation Paradigms / Software Architectures

- Hybrid paradigm / Three Layer Architecture
  - Best of the 2 first paradigms
    - Planning operates on a long horizon
    - Reactive behaviors act on the short horizon

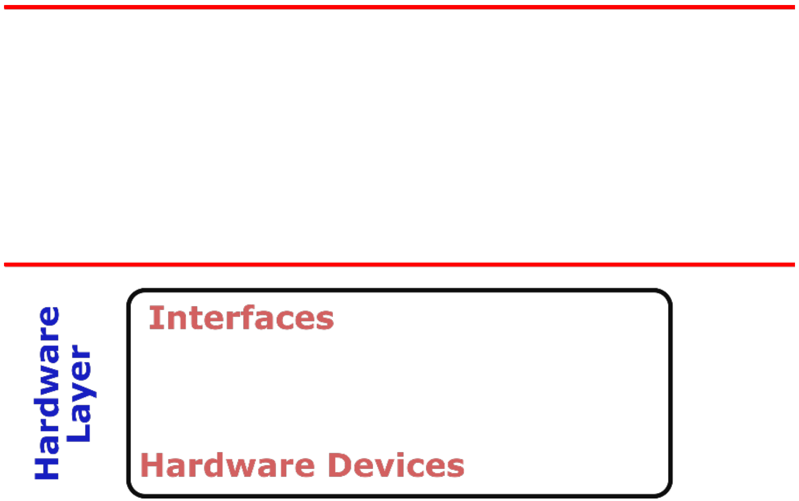


Outline

- Introduction to the Course
- Introduction to Software Architectures
  - Operation Paradigms
  - Hardware Abstraction Layer & Middleware
  - Characteristics of Software Architectures
- Summary

Hardware Abstraction Layer & Middleware

Hardware Abstraction Layers #1

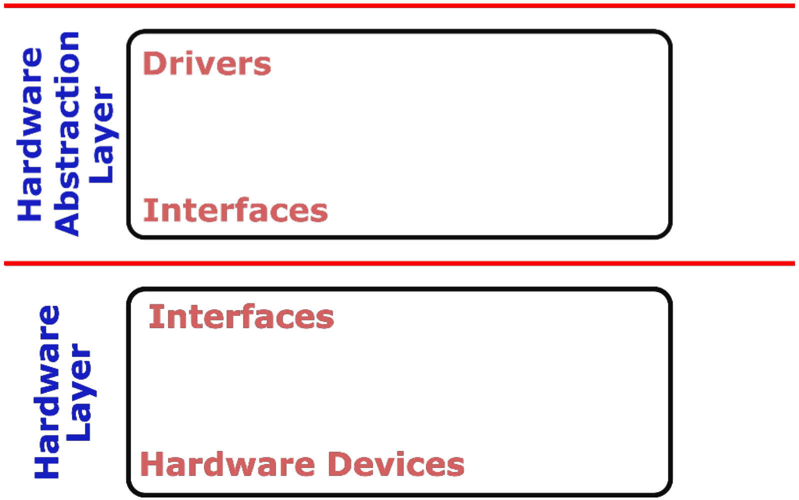


Hardware Layer

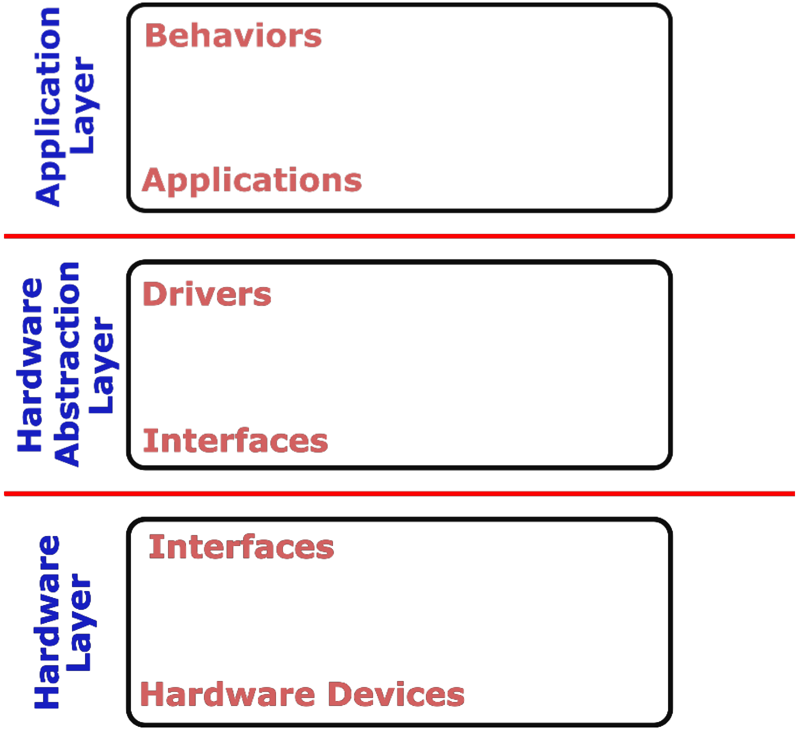
Interfaces

Hardware Devices

Hardware Abstraction Layers #1



Hardware Abstraction Layers #1



Application Layer

Behaviors

Applications

Hardware Abstraction Layer

Drivers

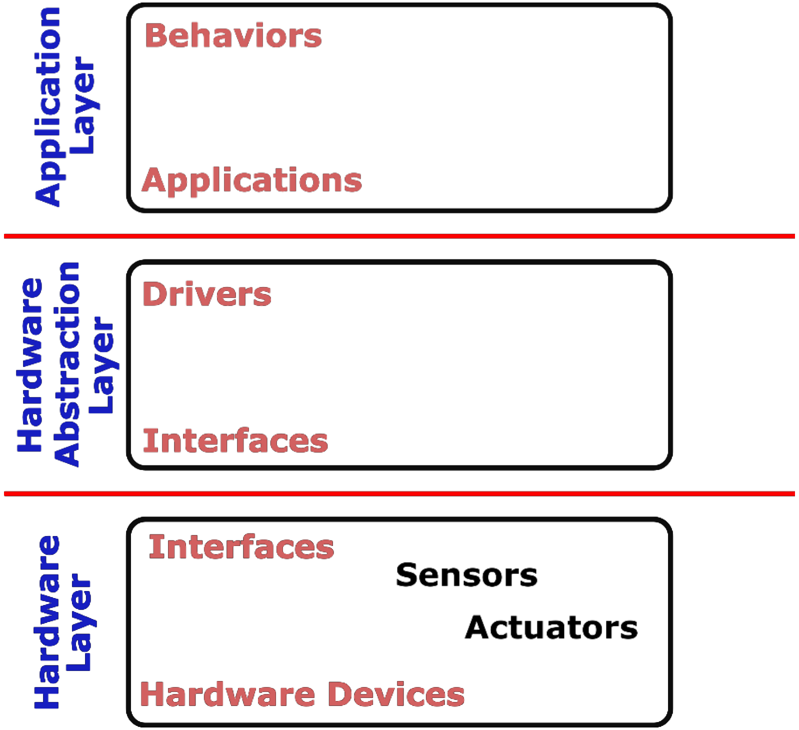
Interfaces

Hardware Layer

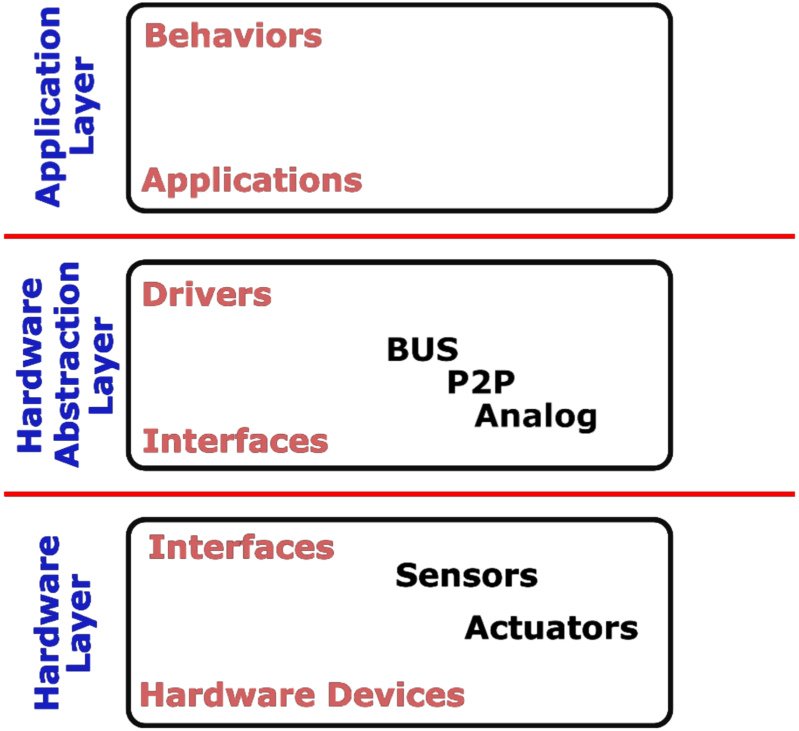
Interfaces

Hardware Devices

Hardware Abstraction Layers #1



Hardware Abstraction Layers #1



Application Layer

Behaviors

Applications

Hardware Abstraction Layer

Drivers

BUS

P2P

Analog

Interfaces

Hardware Layer

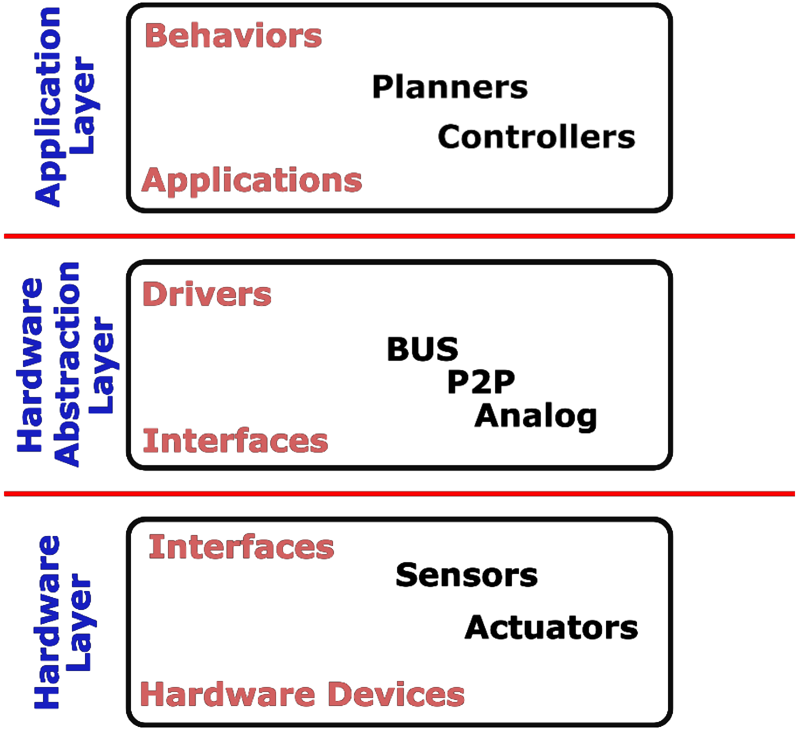
Interfaces

Sensors

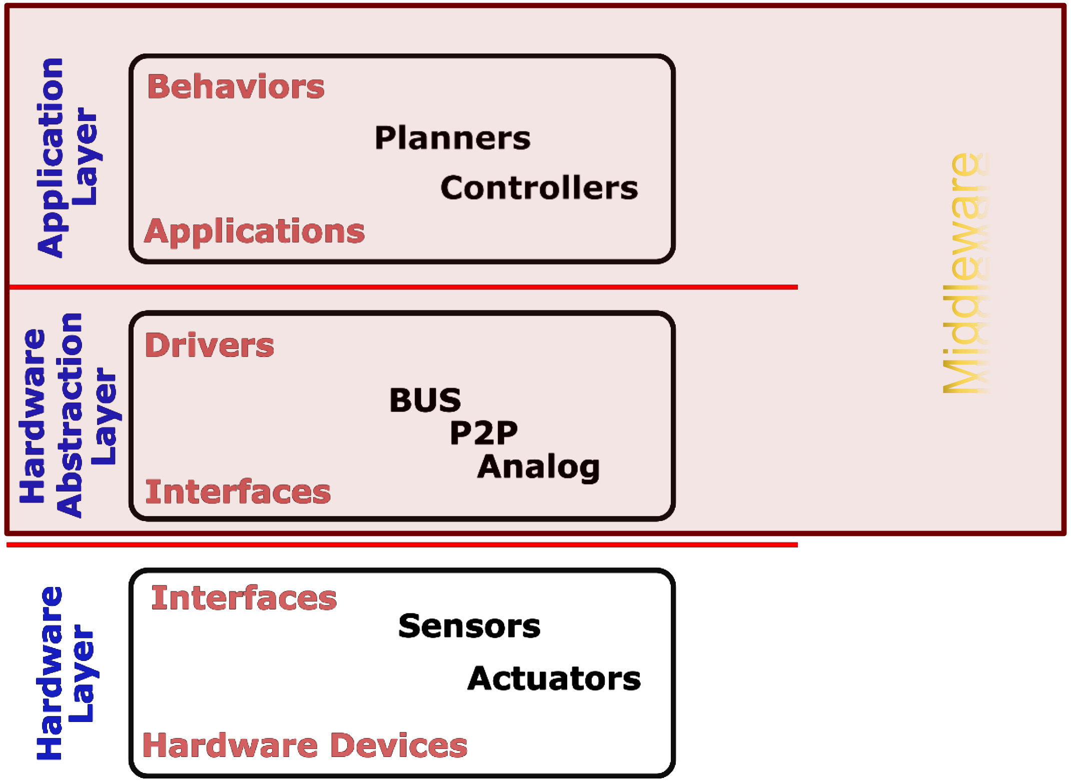
Actuators

Hardware Devices

Hardware Abstraction Layers #1



Hardware Abstraction Layers #1



Hardware Abstraction Layers #2

- Let’s add a little detail..

Hardware Abstraction Layers #2

- Let’s add a little detail..

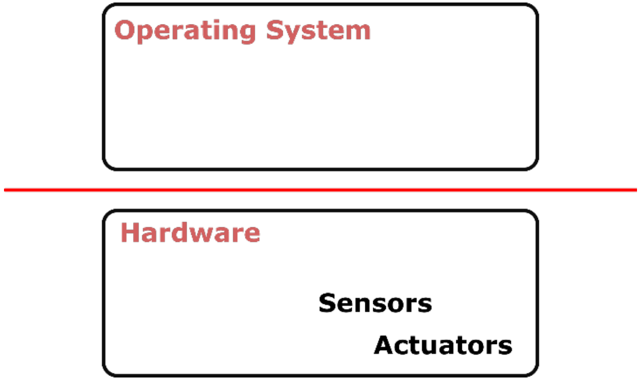
Hardware Abstraction Layers #2

- Let’s add a little detail..

Hardware
Sensors
Actuators

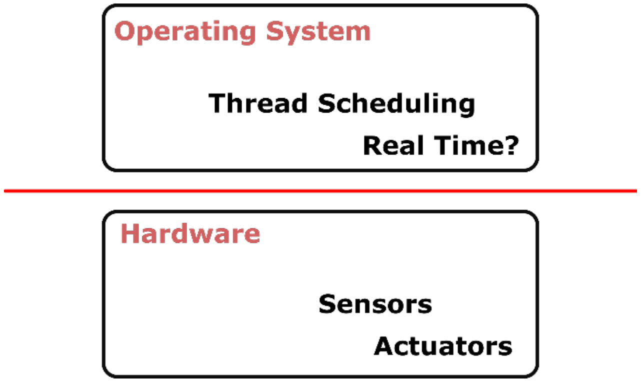
Hardware Abstraction Layers #2

- Let’s add a little detail..



Hardware Abstraction Layers #2

- Let’s add a little detail..



Operating System

Thread Scheduling

Real Time?

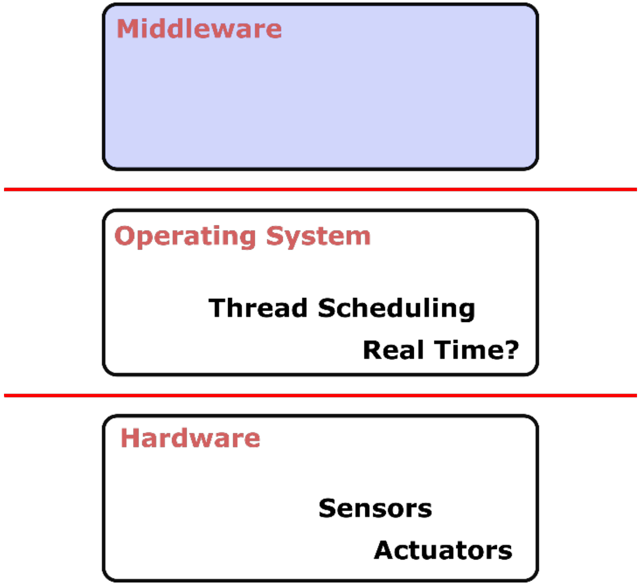
Hardware

Sensors

Actuators

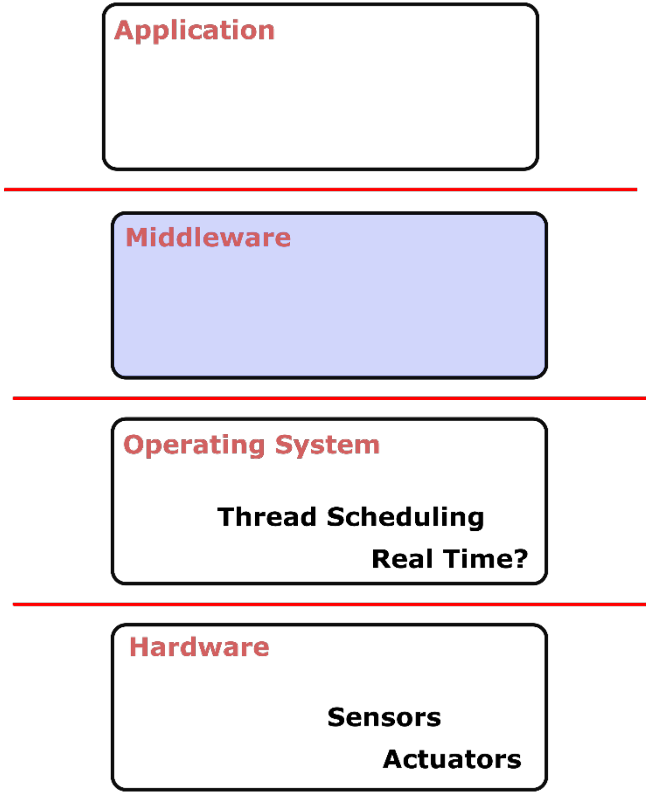
Hardware Abstraction Layers #2

- Let’s add a little detail..



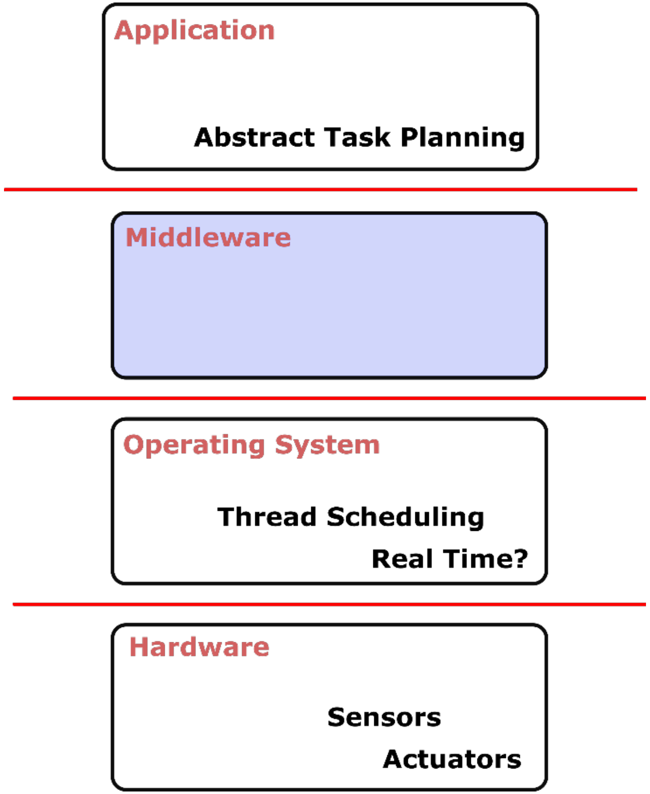
Hardware Abstraction Layers #2

- Let’s add a little detail..



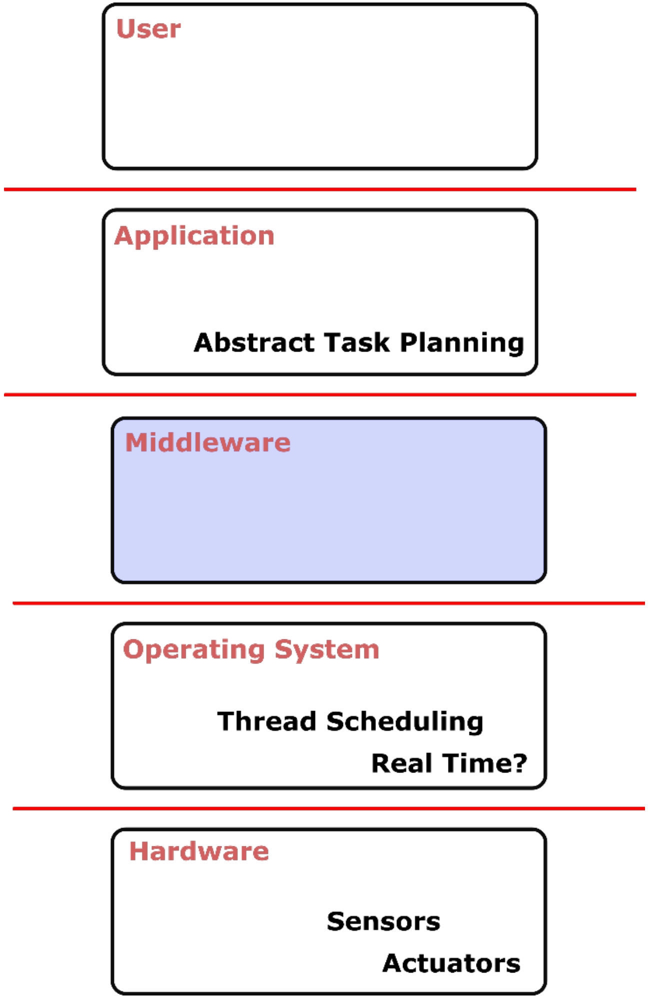
Hardware Abstraction Layers #2

- Let’s add a little detail..



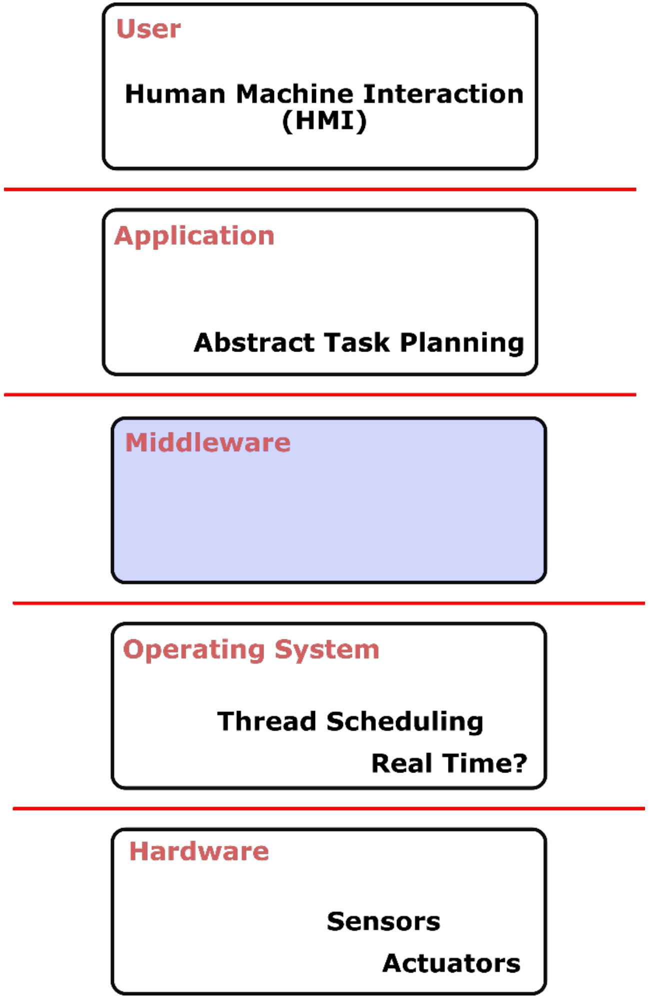
Hardware Abstraction Layers #2

- Let’s add a little detail..



Hardware Abstraction Layers #2

- Let's add a little detail..



User

Human Machine Interaction (HMI)

Application

Abstract Task Planning

Middleware

Operating System

Thread Scheduling

Real Time?

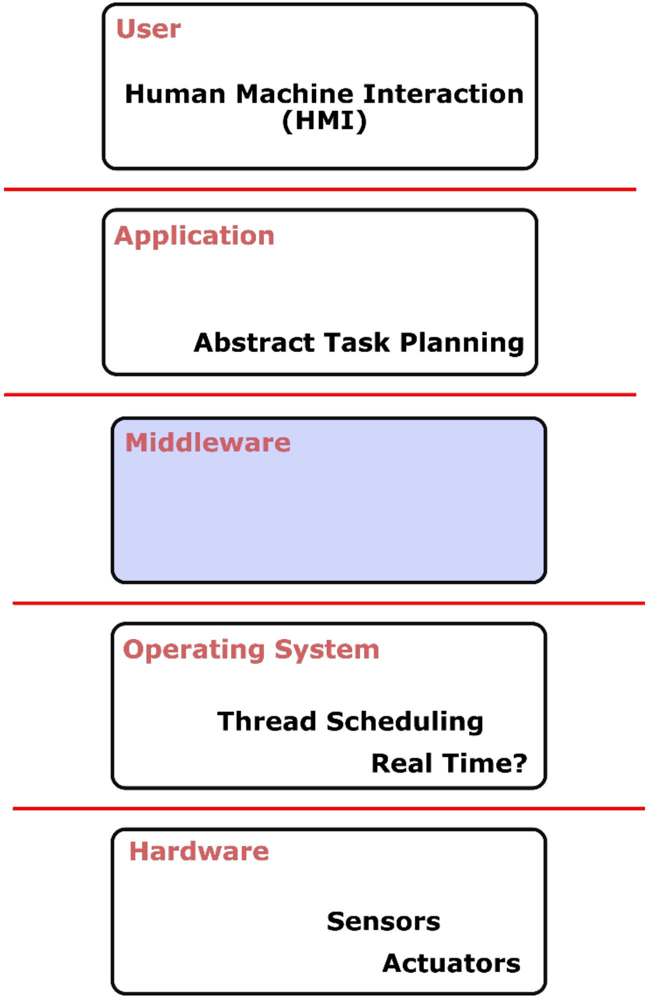
Hardware

Sensors

Actuators

Hardware Abstraction Layers #2

- Let’s add a little detail..
- A class of technologies in order to handle the complexity of distributed systems



Outline

- Introduction to the Course
- Introduction to Software Architectures
  - Operation Paradigms
  - Hardware Abstraction Layer & Middleware
  - Characteristics of Software Architectures
- Summary

Towards Software Architecture

- What characteristics should Software for Autonomous Systems have?

Towards Software Architecture

Must have characteristics:

- Support for multiple components
- Communication between components
- Easy way to write own components
- Possibility to replace individual components
- Easy to extend
- Means for data logging and debugging
- Support for decentralized components

Towards Software Architecture

Nice to have characteristics:

- Robustness
- Hardware abstractions
- Open access (ideal case: open source)
- Hardware/OS independent
- Means for time stamping
- Means for visualization
- …

Outline

- Introduction to the Course
- Introduction to Software Architectures
  - Operation Paradigms
  - Hardware Abstraction Layer & Middleware
  - Characteristics of Software Architectures
- Summary

Outline

Introduction to the Course

- – Get your groups done by next week!

Introduction to Software Architectures

- **– Operation Paradigms**
  - – Hierarchical / Reactive / Three Layer Architecture
- **– Hardware Abstraction Layer & Middleware**
  - – ROS lives here!!
- **– Characteristics of Software Architectures / Frameworks**
  - – Make our life easier and simpler

DTU

Before we leave:

Disinfect table and chair

Maintain your distance to others

Wash or sanitize your hands

Respect guidelines and restrictions outside