Robotics, AI and Machine Vision

Jesus Savage

Bio-Robotics Laboratory
biorobotics.fi-p.unam.mx

School of Engineering
National Autonomous University of Mexico, UNAM

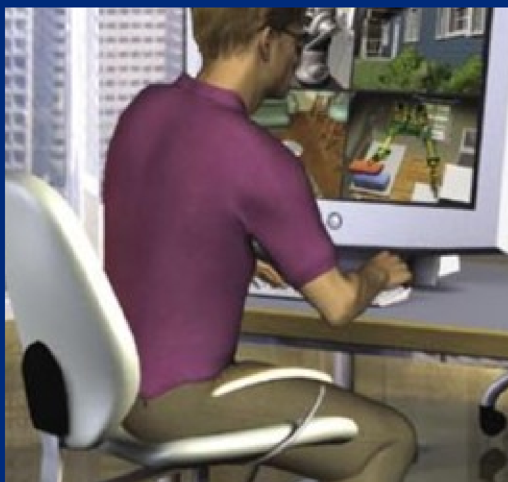
15th International Conference on Electromechanics and Robotics "Zavalishin's Readings - 2020

OUTLINE

- Introduction
- Traditional Robotics Architectures
- Reactive Robotics Architectures
- Probabilistic Robotics Architectures
- Hybrid Robotics Architectures
- A System to Operate Mobile Robots (VIRBOT)
- Video

Introduction

Service Robots in an Intelligent House



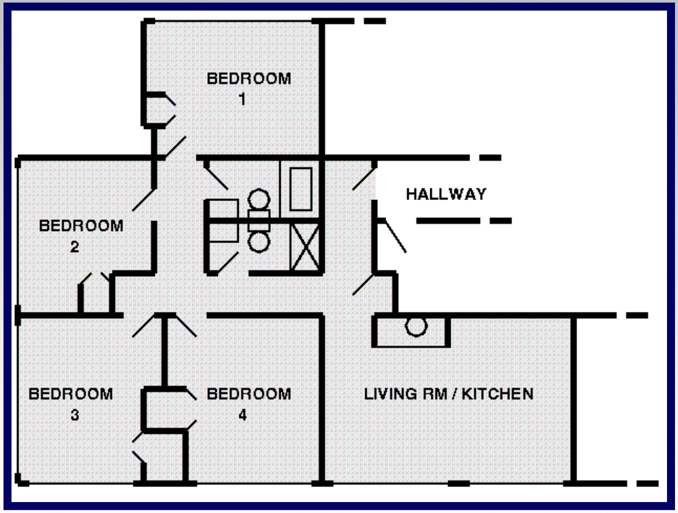
A Robot in Every Home: Overview/The Robotic Future. Bill Gates, Scientific American (2008)

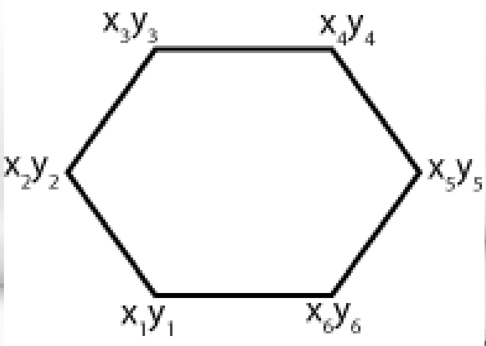
Traditional Robotics Architectures

Features:

There is representation of the environment, with a symbolic representation of the objects in each room.

These are represented by polygons where they have their vertices Xi, Yi, ordered clockwise. These polygons separate the occupied space and the free space where the robot can navigate.

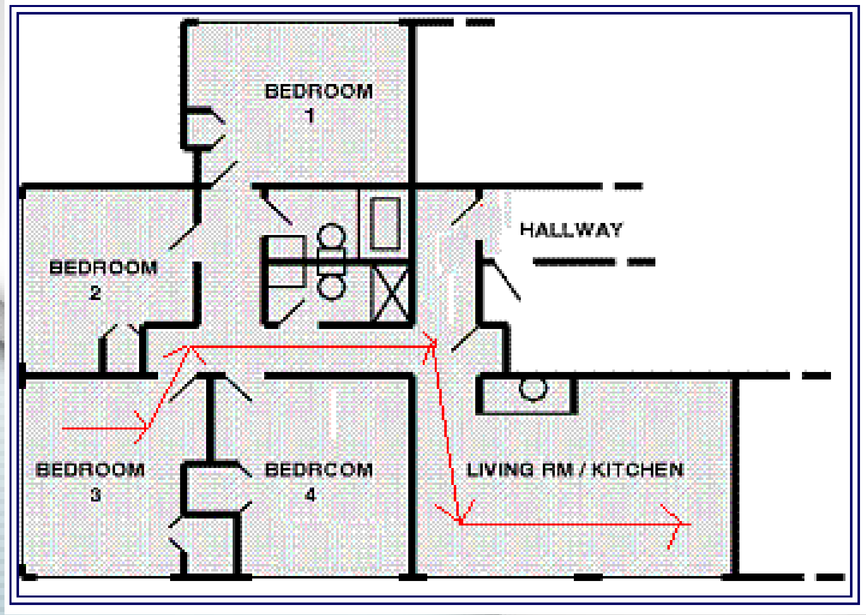




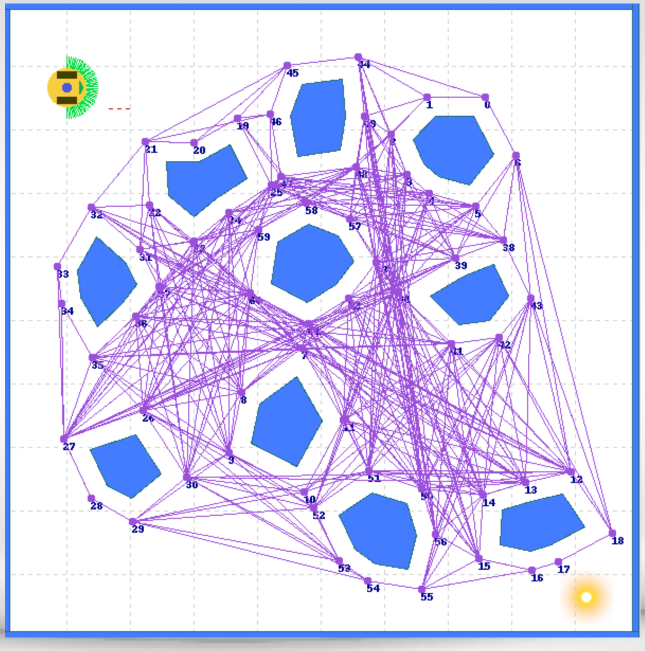
Traditional Robotics Architectures

Features:

Movements and actions are planned using traditional artificial intelligence techniques of searches in topological networks.



Global Path



Local paths for each room

Traditional Robotics Architectures

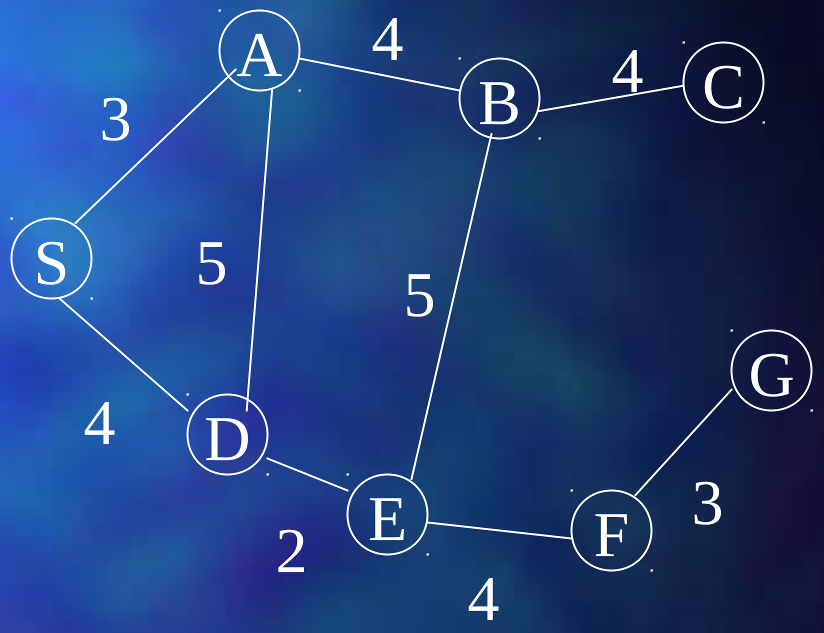
The basic search problem:

Given:

1. Starting point (node)
2. The goal point (node)
3. A map of nodes and connections

Goals:

1. Find some path or find the "best" path (maybe shortest)
2. Traverse the path



Traditional Robotics Architectures

Artificial Intelligence Techniques to Search for:

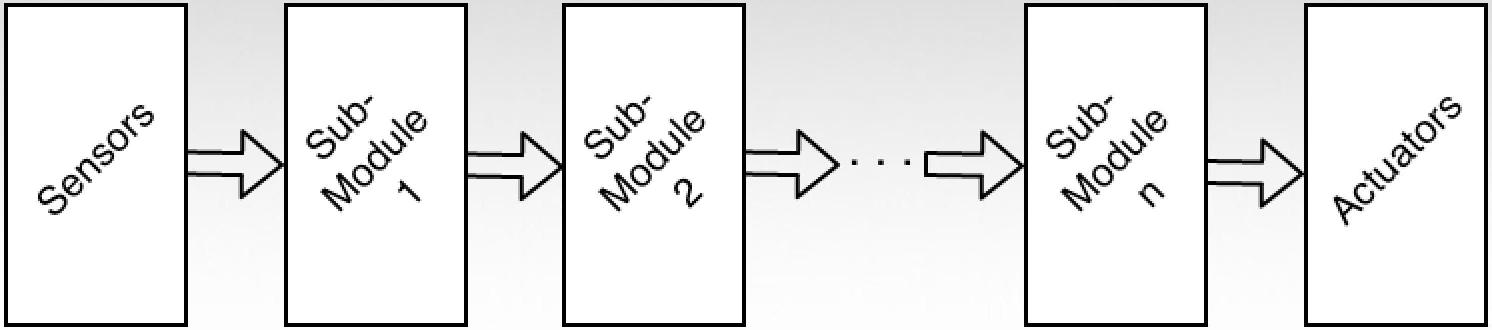
1. Some path
2. Optimal path

- Search
  - Some path
    - Depth-first
    - Hill climbing
    - Breadth-first
    - Beam
    - Best-first
  - Optimal Path
    - British museum
    - Branch and bound
    - **A***
    - Dijkstra

Traditional Robotics Architectures

Features:

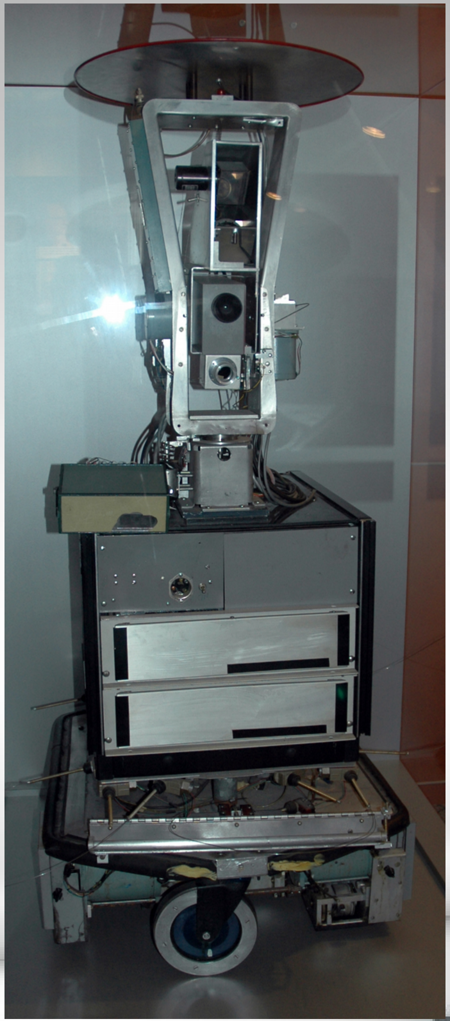
It has a serial organization, if one module fails the entire system fails.



This type of systems are not suitable for dynamic environments or for robots that have errors in movement and sensing.

Traditional Robotics Architectures

The robot Shakey was the first example of this architecture, developed from approximately 1966 through 1972 in Stanford University.



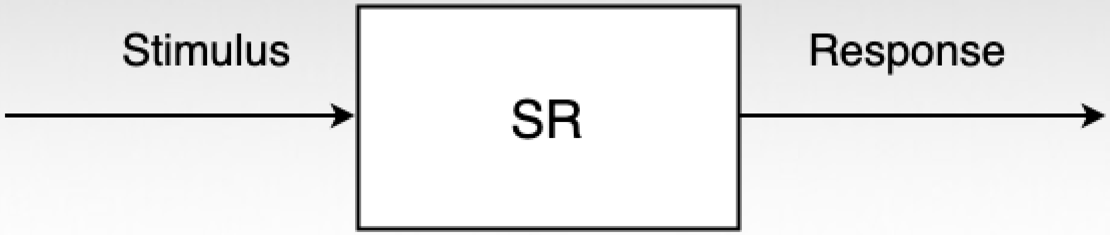
Reactive Robotics Architectures

Features:

- Based on the behavior of insects.
- A representation of the environment is not necessary.
- It does not use action or movement planning.
- It is suitable for dynamic environments and with sensing errors.
- It is based on behaviors running in parallel.

Reactive Robotics Architectures

Behaviors are represented using stimulus-response or SR diagrams.



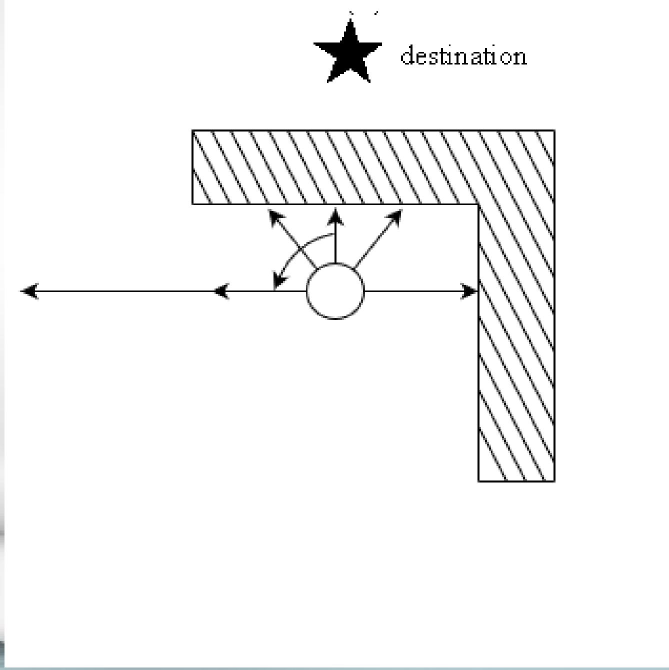
The output of each behavior must be instantaneous from the moment there is an entry. The behaviors are independent of each other. Behaviors can be designed using zero order logic, state machines, potential fields, neural networks, etc.

Reactive Robotics Architectures
Zero Order Logic

Features:

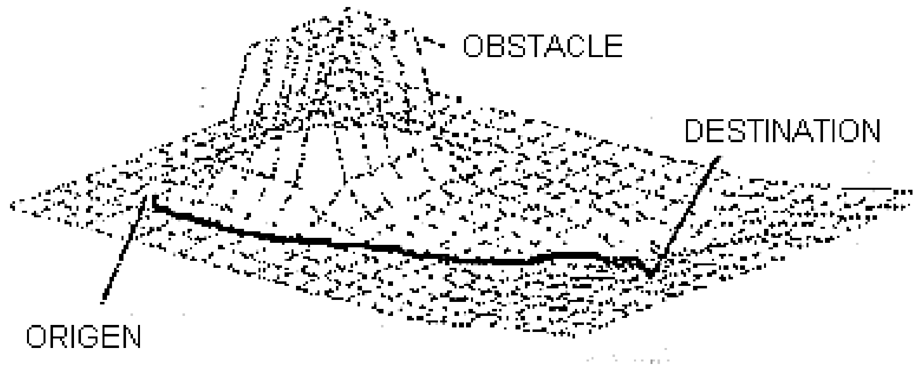
Values of the input sensors are checked and if they comply with a certain property, an output is generated, which lasts a certain time. These behaviors have no memory.

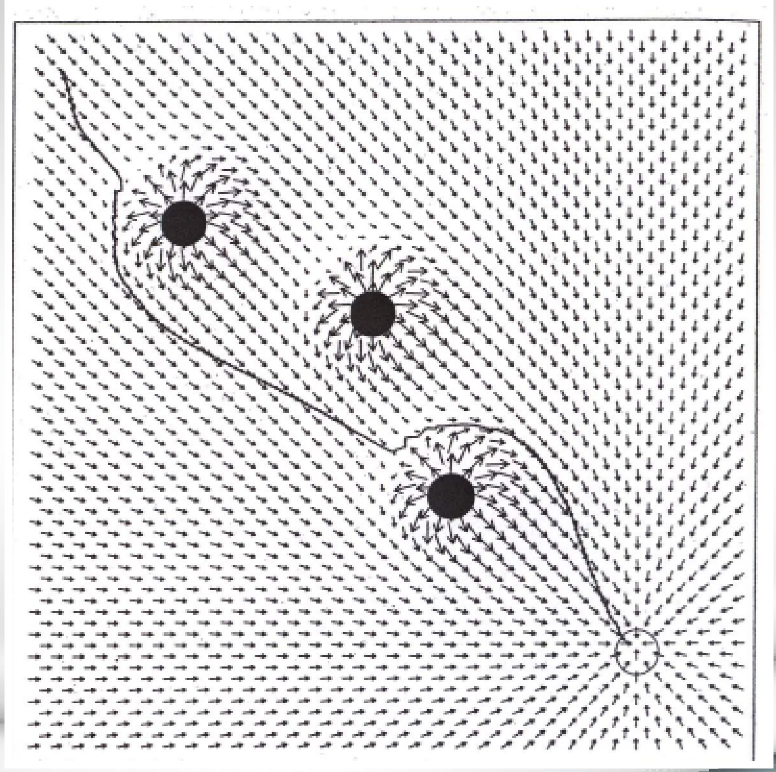
Example: If the light source is above the robot and there is an obstacle in front and on the right side, then the robot turns to the left.



Reactive Robotics Architectures

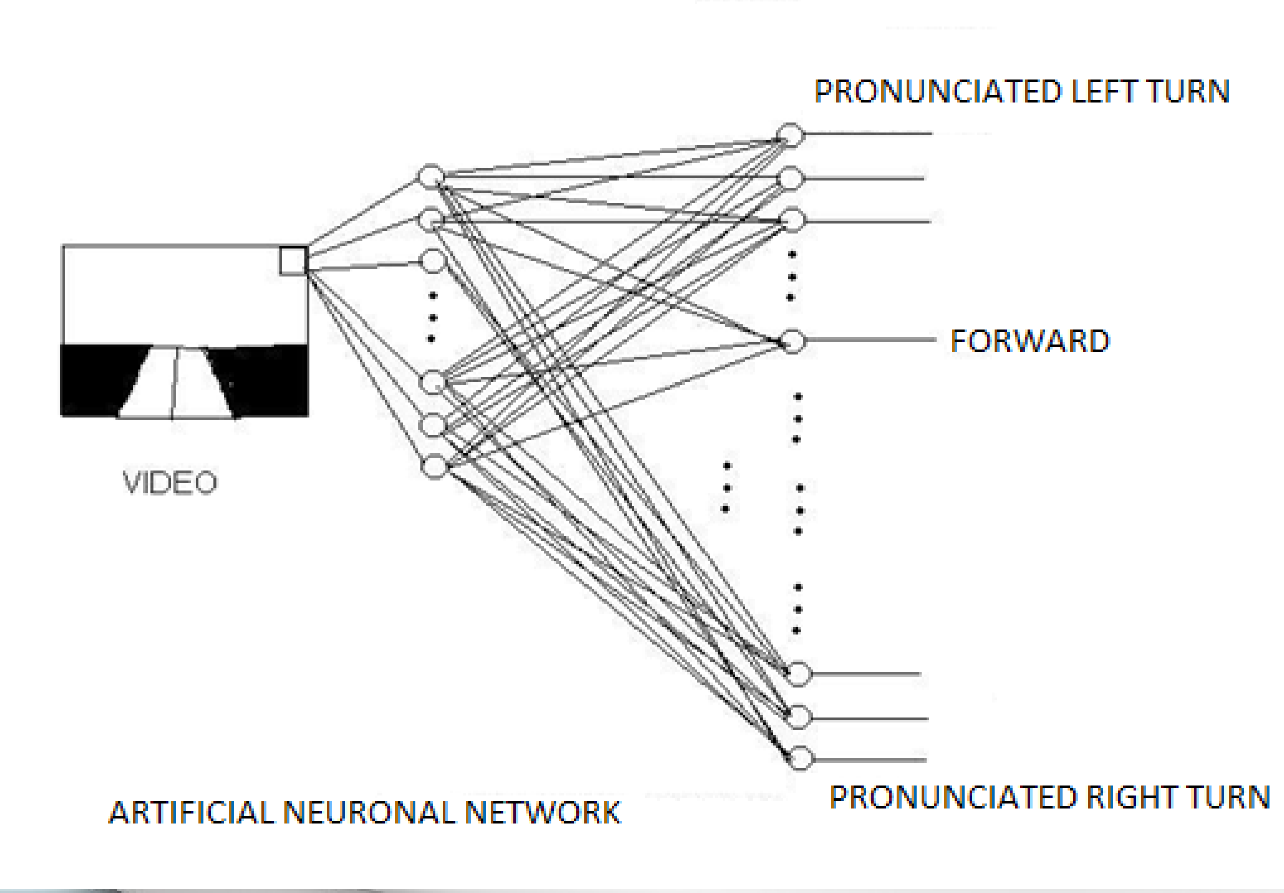
Potential attractive and repulsive fields





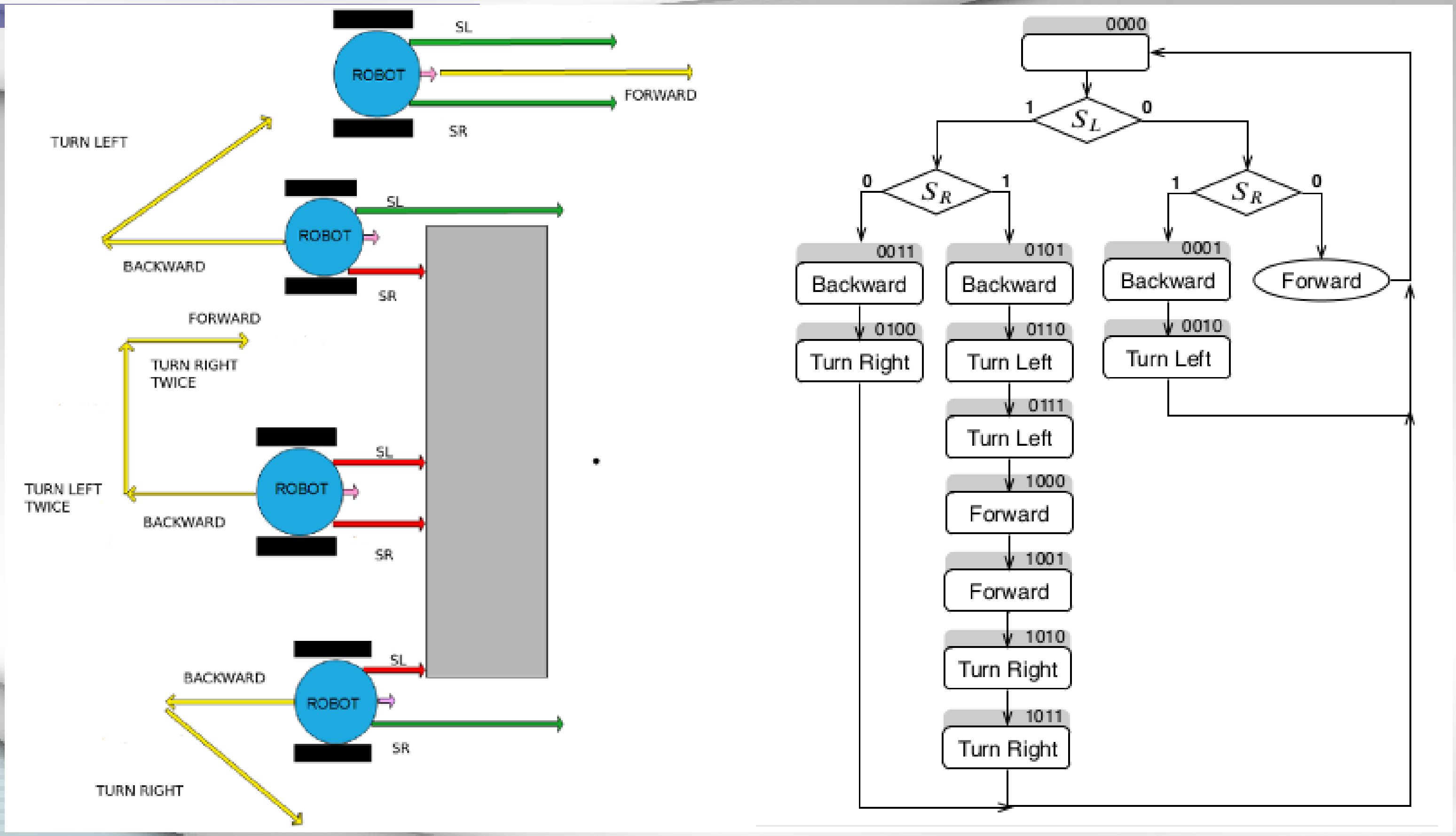
These behaviors have no memory.

Reactive Robotics Architectures Using Artificial Neural Networks



Reactive behaviors without memory using neural networks

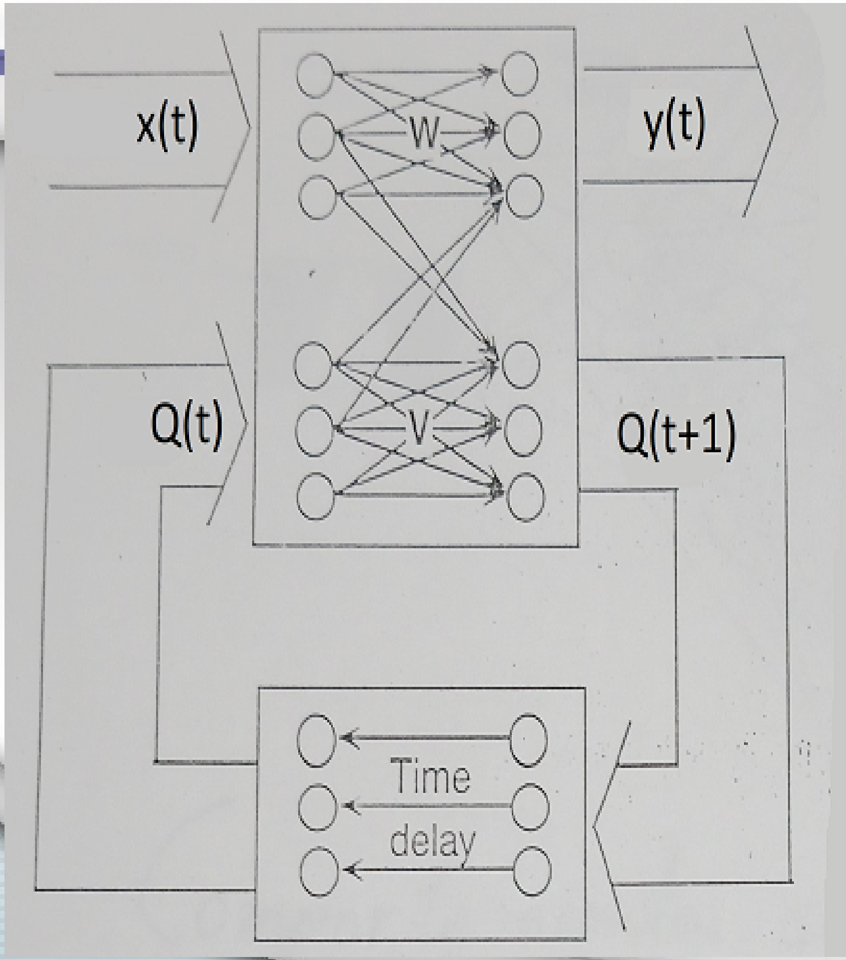
Reactive Robotics Architectures
Using state machine algorithms



These behaviors have memory.

Reactive Robotics Architectures

Using Recurrent Artificial Neural Networks

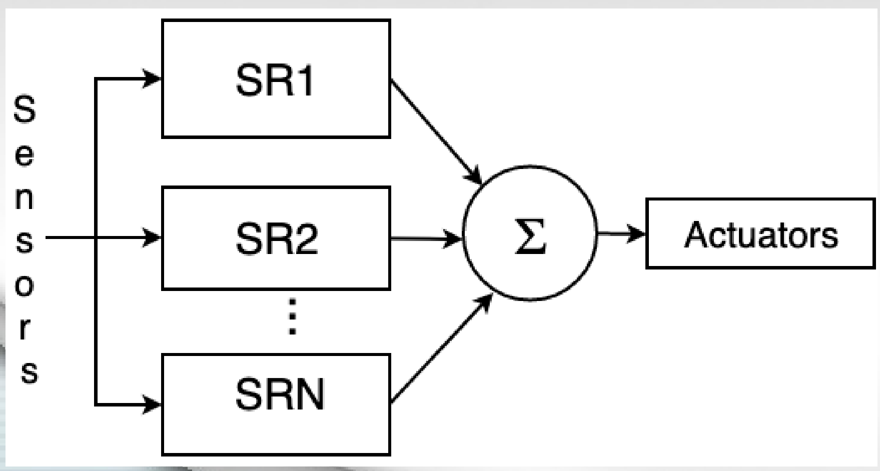


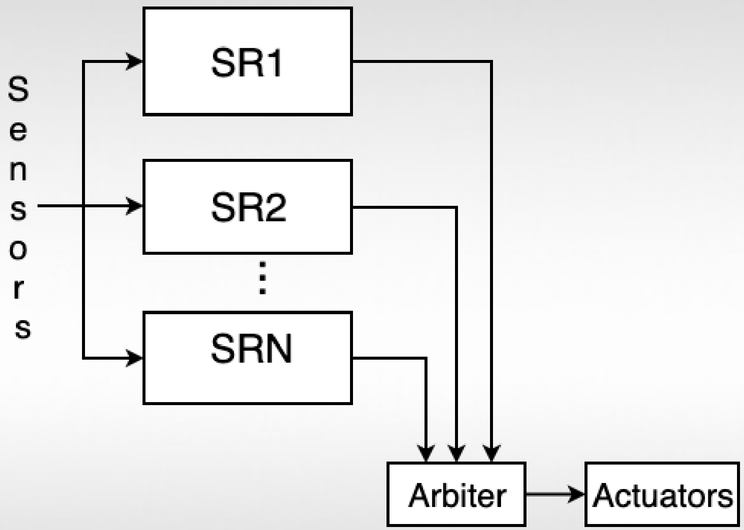
x(t): Input Vector
Q(t): State Vector
y(t): Output Vector

These behaviors have memory.

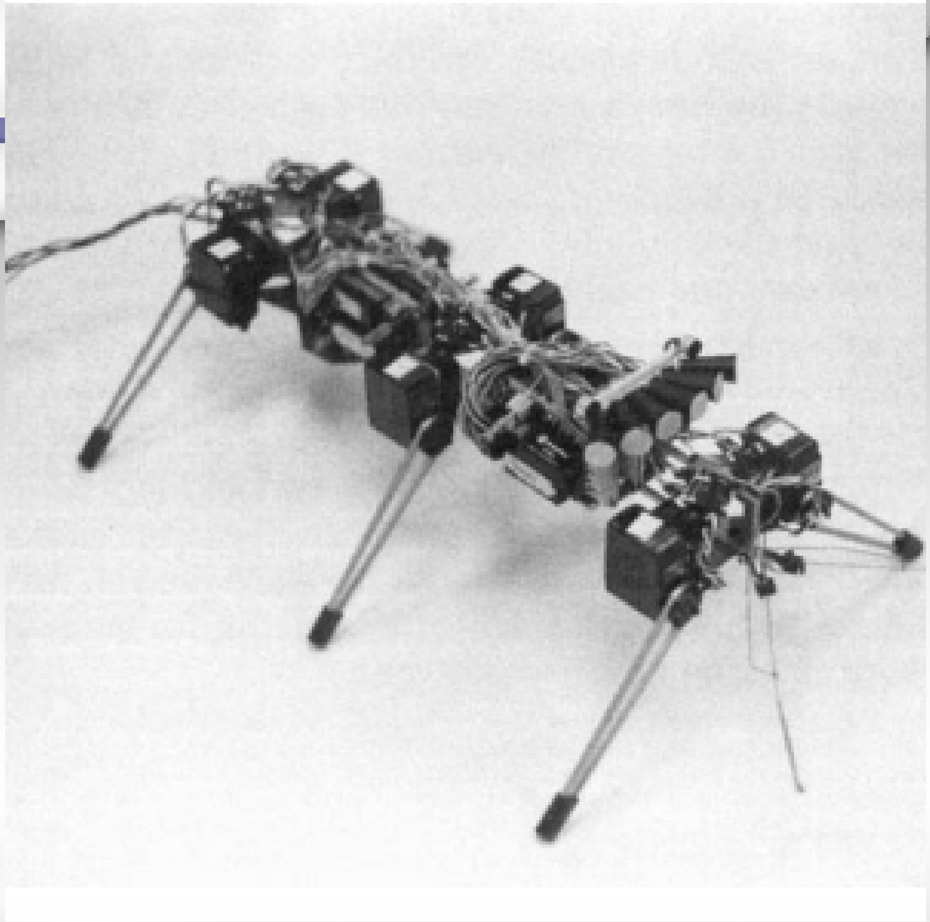
Reactive Robotics Architectures

The SR can be combined in different structures by connecting them in parallel by adding the output of each of them or selecting one of the outputs using an arbiter.





Reactive Robotics Architectures

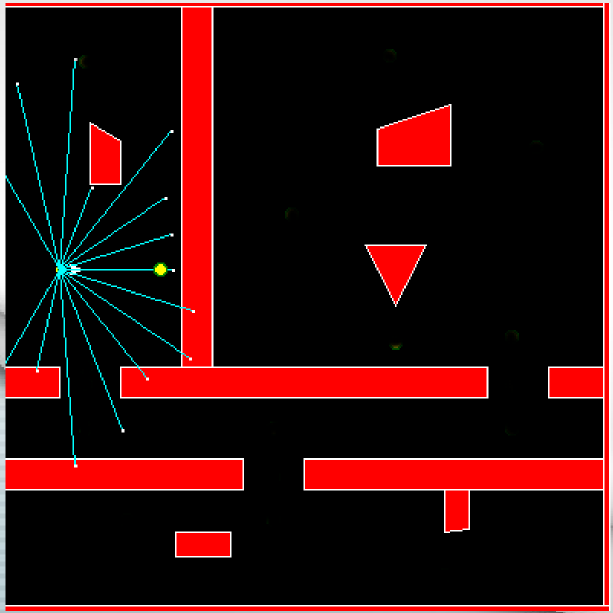


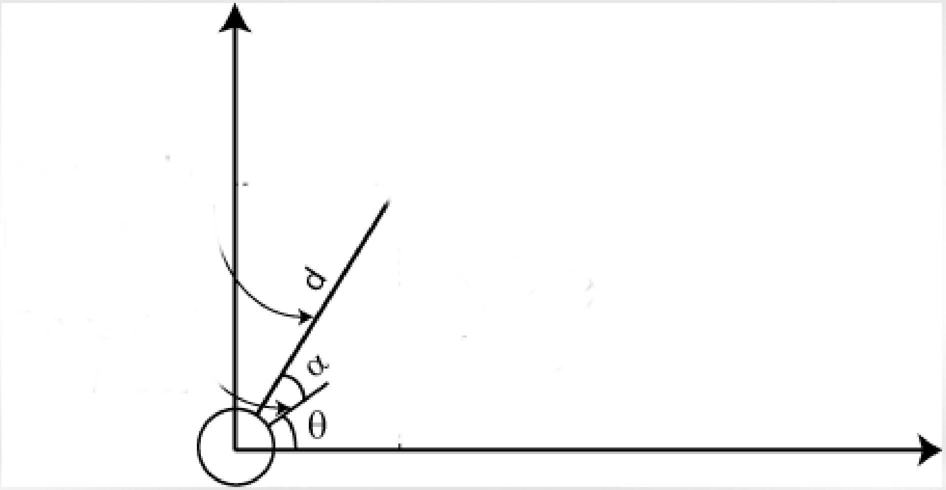
Genghis, a robust hexapodal walker, is an example of this architecture, developed by Rodney Brooks in the MIT in the 80’s.

Probabilistic Robotics Architectures

Features:

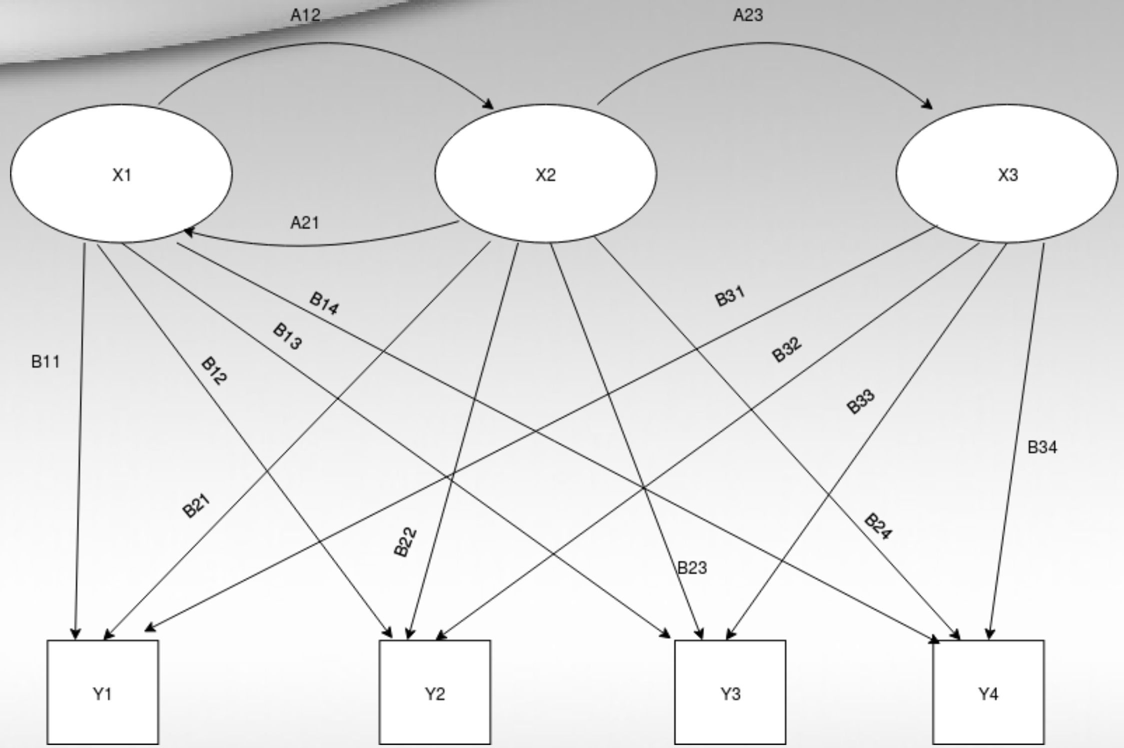
It is based on the concept that both the sensing of the environment that the robot makes, as well as its movements are dependent on random variables, which can be manipulated using probabilistic concepts.





Probabilistic Robotics Architectures

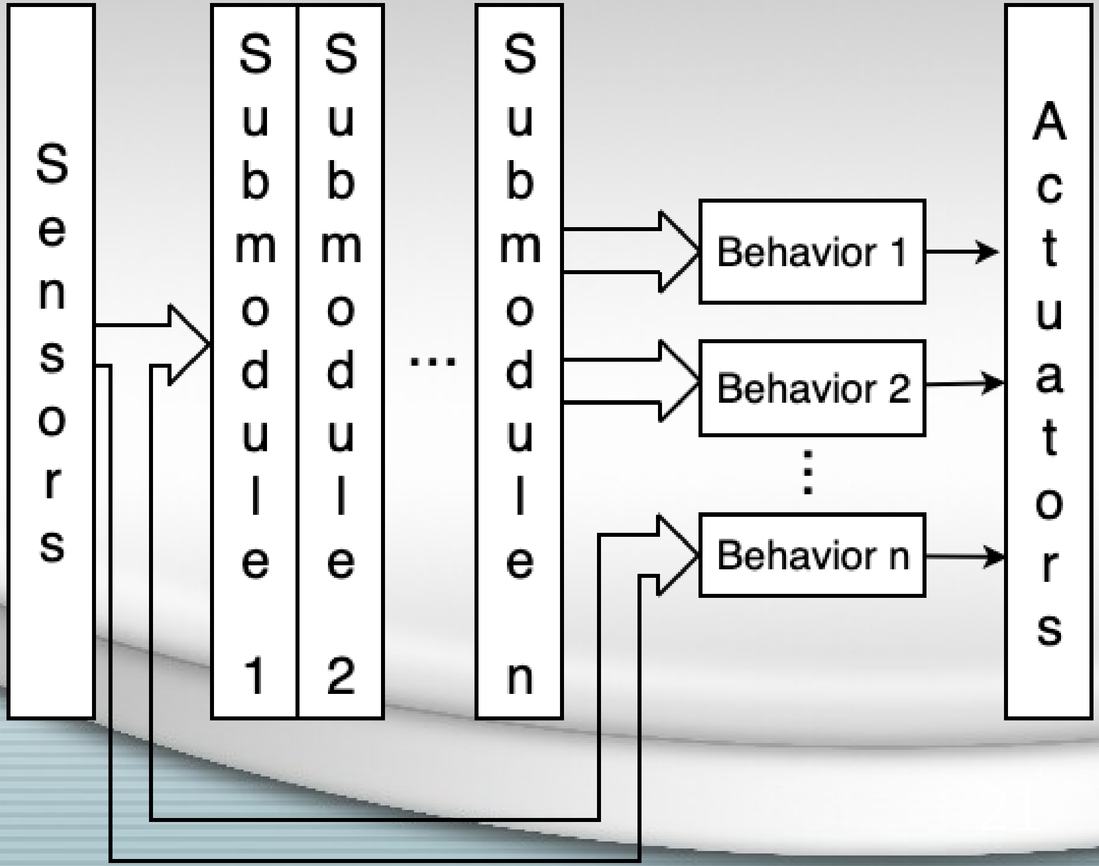
Hidden Markov Models (HMM), Particle Filters, Markov Decision Processes, etc. are used.



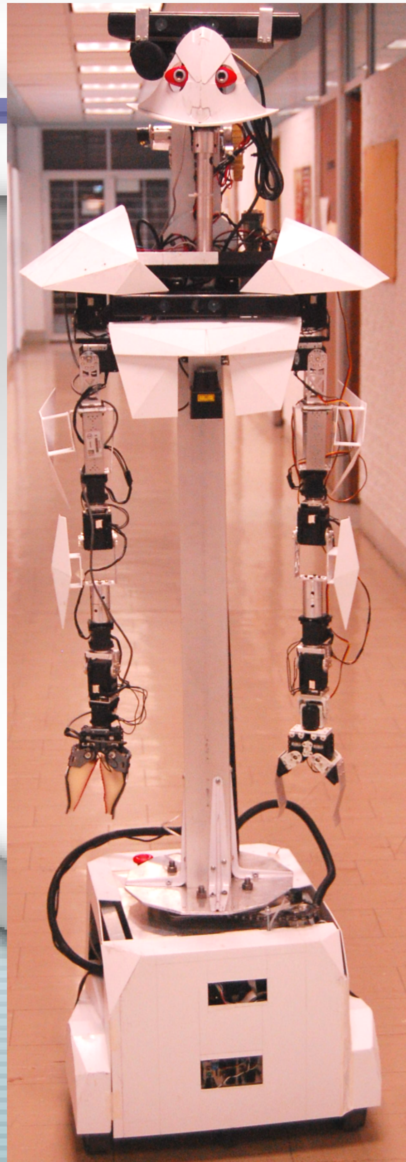
Stanley, an autonomous car created by Stanford University in 2005 under the direction of Sebastian Thrun, is an example of this architecture.

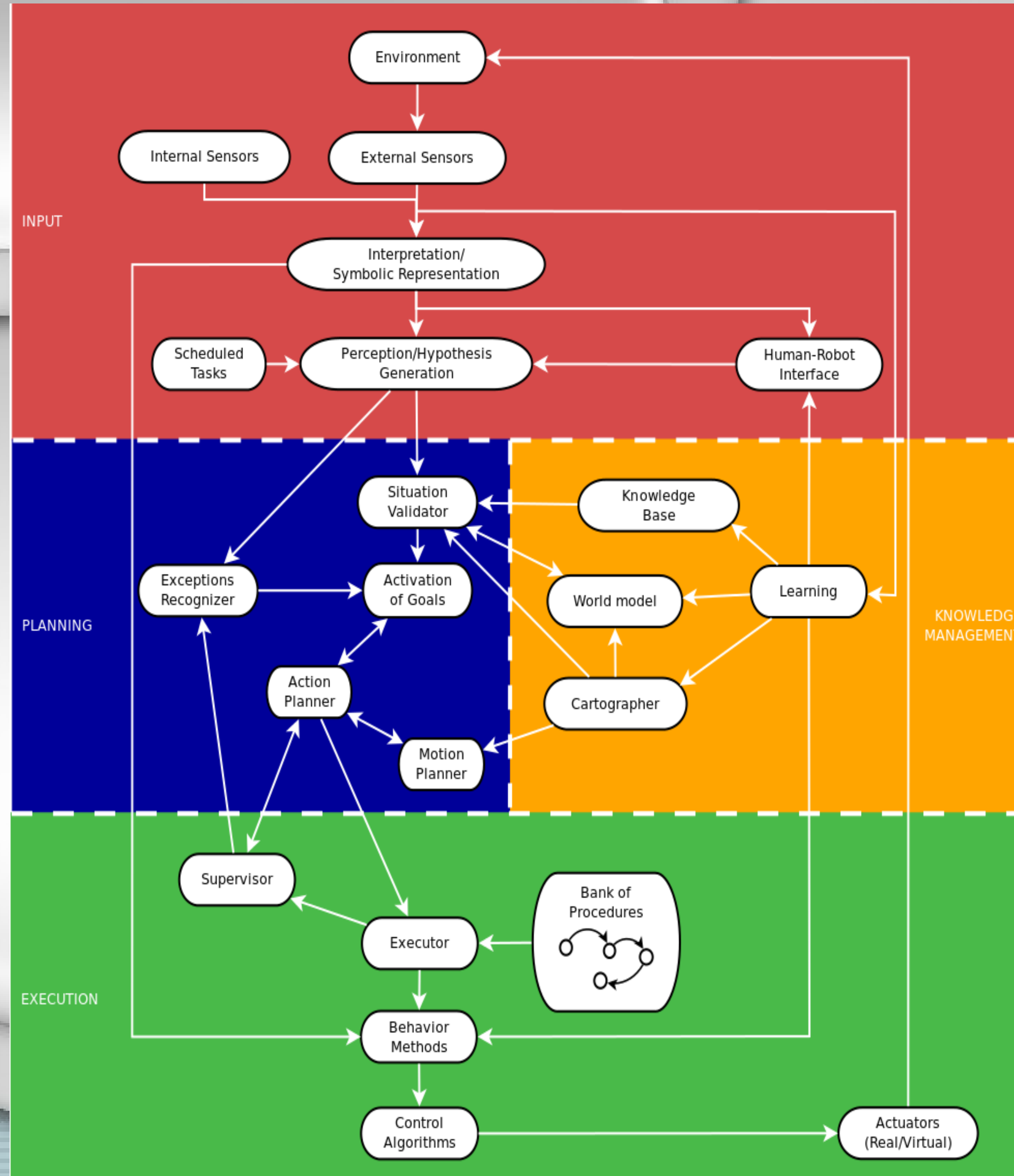
Hybrid Robotics Architectures

Traditional, reactive and probabilistic architectures are combined to replace the deficiencies of each of them.

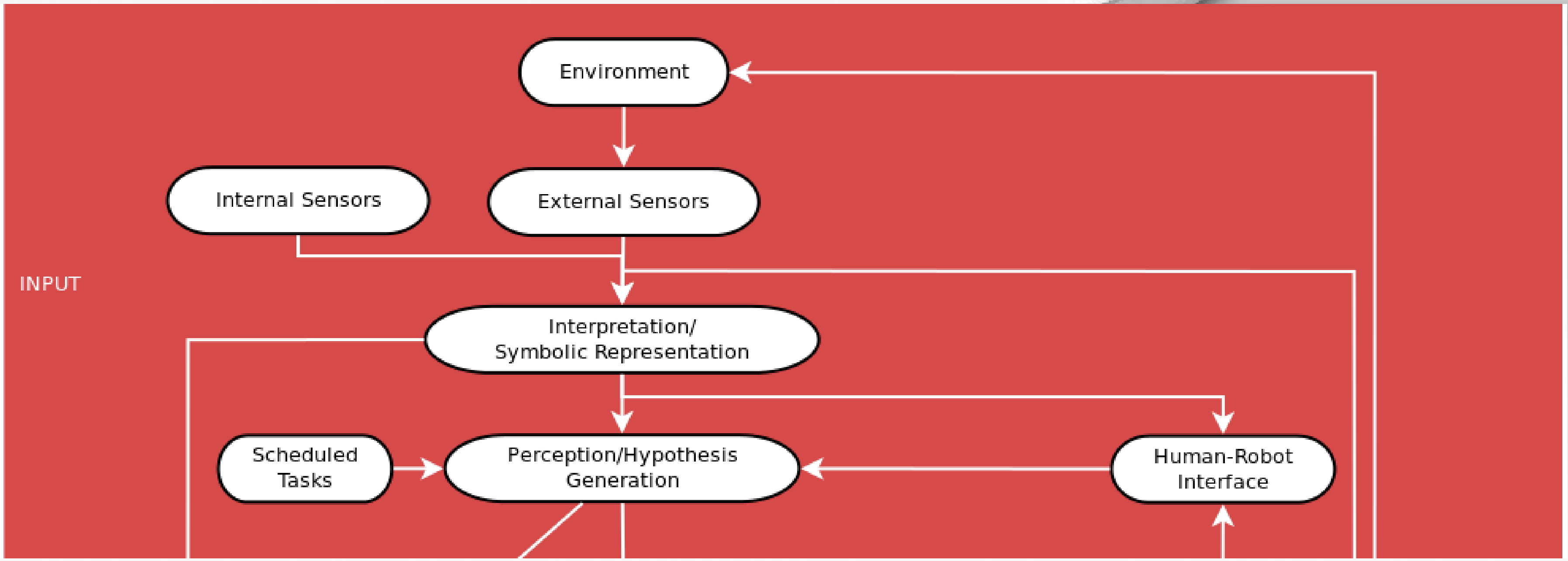


Hybrid model used with the Justina robot: Virtual Real Robot (ViRBot)





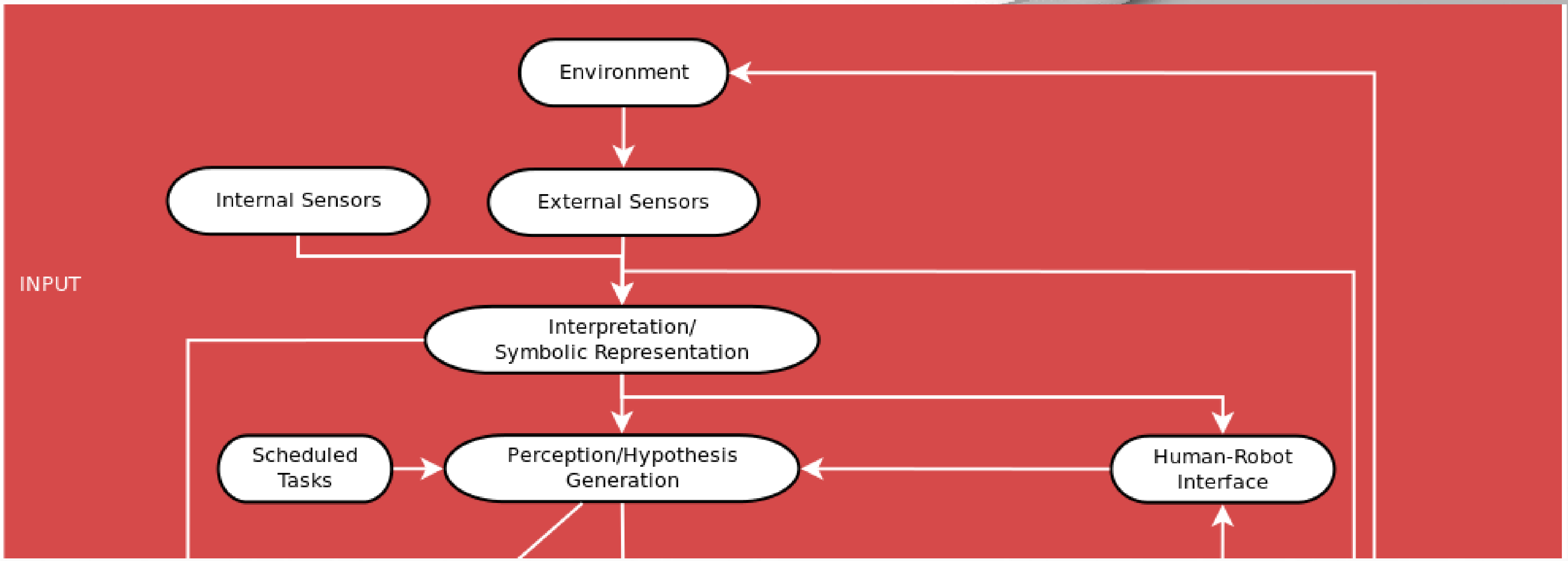
INPUT LAYER



This layer process the data from the robot's internal and external sensors, they provide information of the internal state of the robot, as well as, the external world where the robot interacts.

In some of our robots designs they have lasers, sonars, infrared, microphones and stereo and RGB-D cameras.

INPUT LAYER



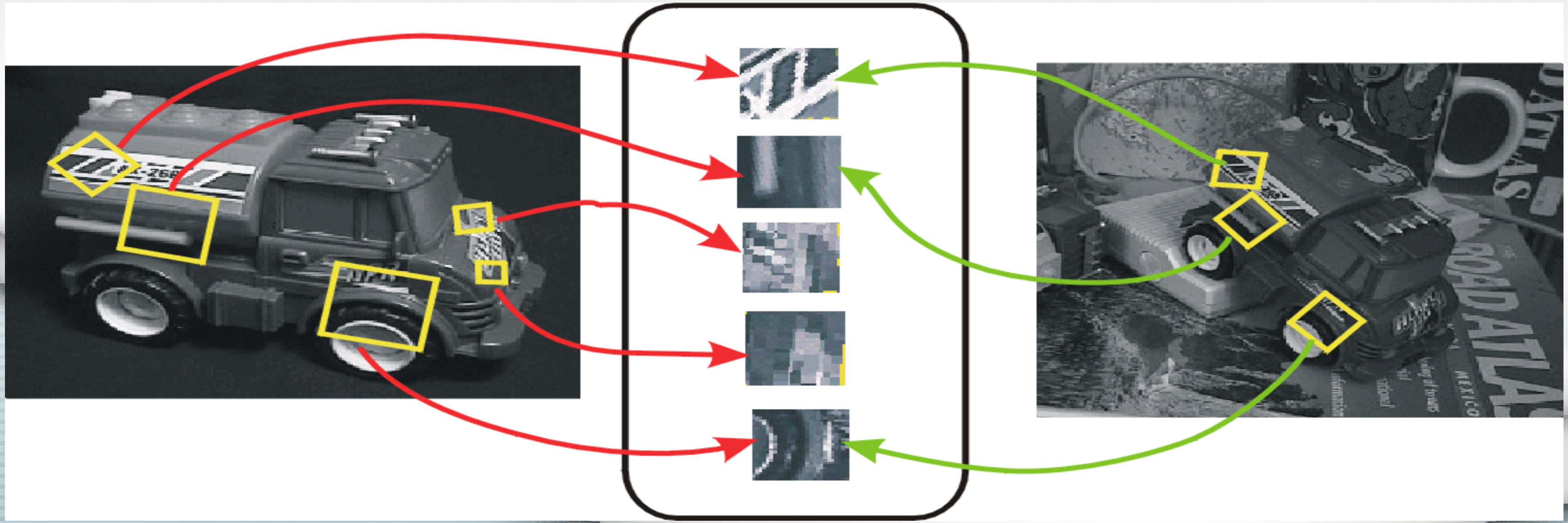
Digital signal processing techniques are applied to the data provided by the internal and external sensors to obtain a symbolic representation of the data, as well as, to recognize and to process voice and visual data.

Pattern recognition techniques are used to create models of the objects, places and the persons that interact with the robot.

INPUT LAYER

Vision System

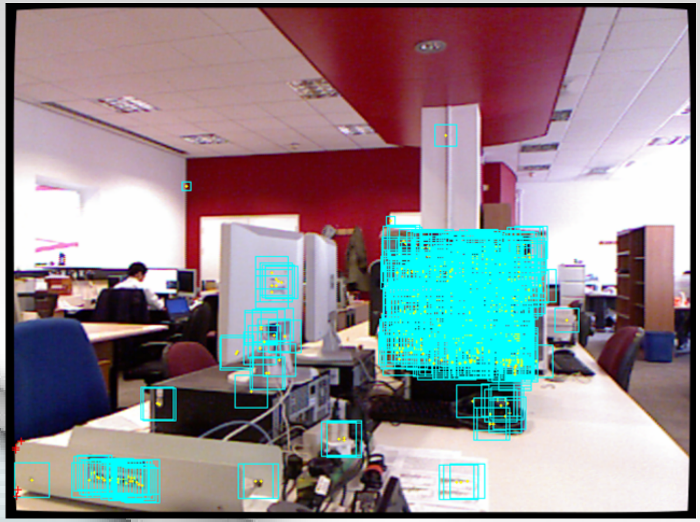
For the recognition of objects, people and places, using RGB-D cameras, vision systems that are robust to partial occlusions, scale and rotation changes are used



Visual Description

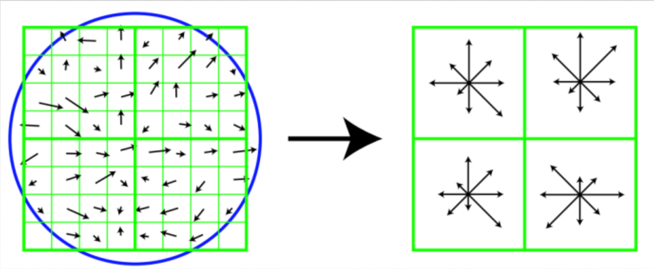
Detection

Detector FAST



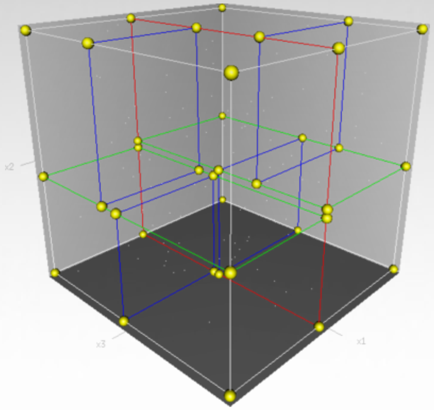
Description

Descriptor SIFT



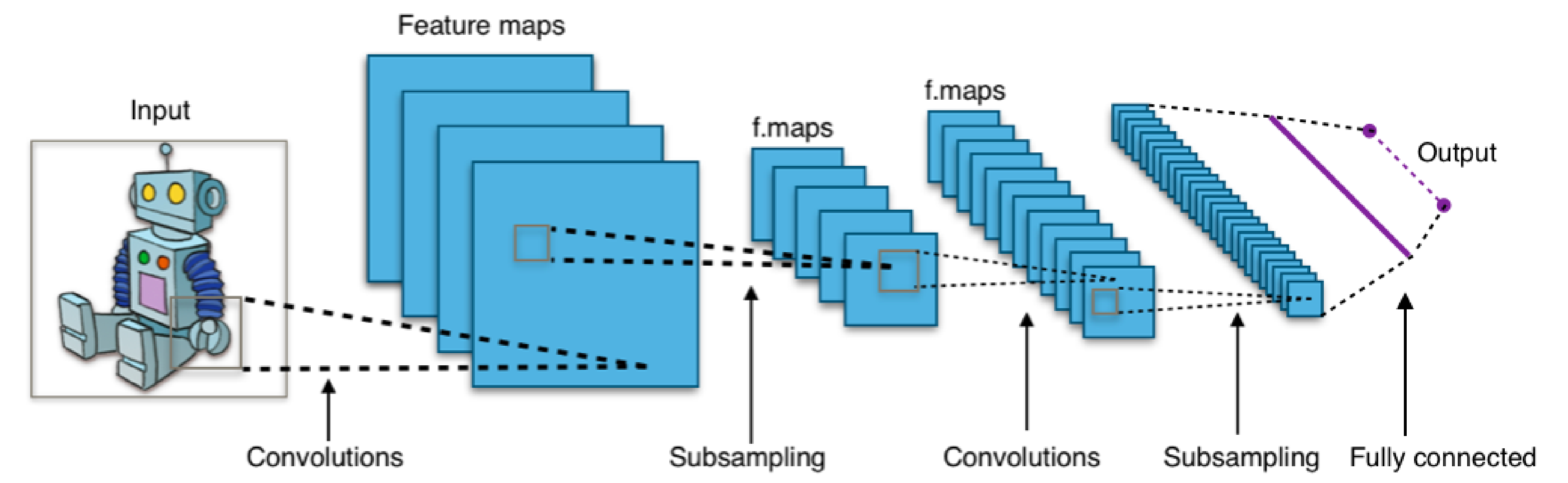
Storage

KD Trees

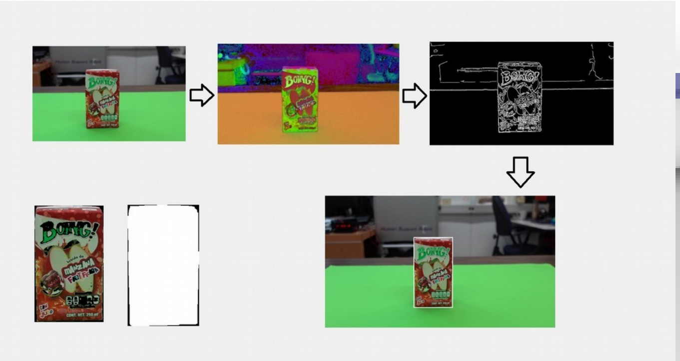


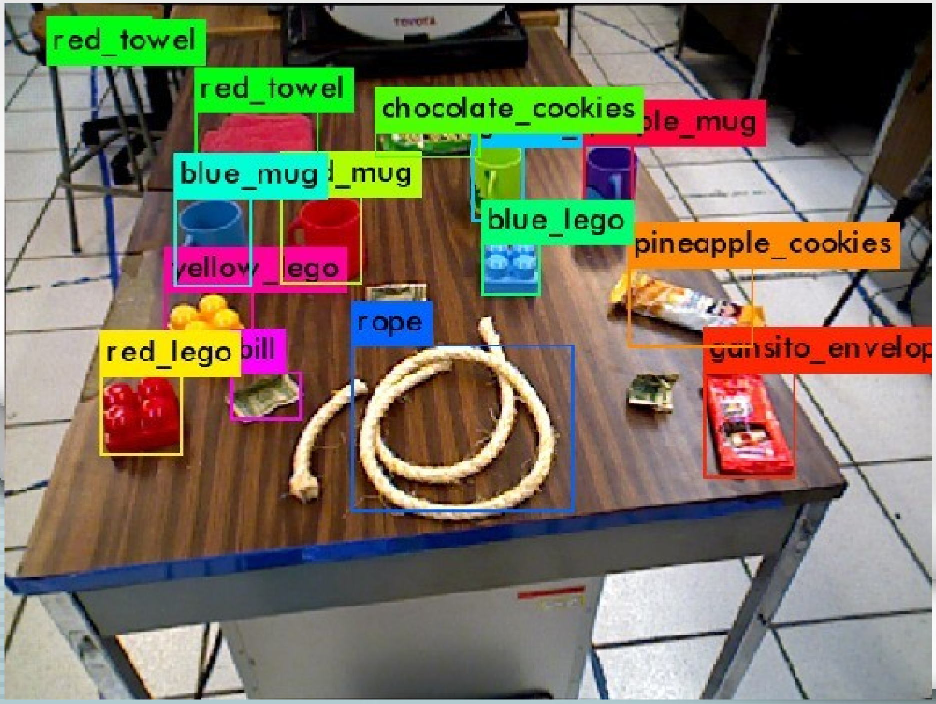
INPUT LAYER

Convolutional neural network



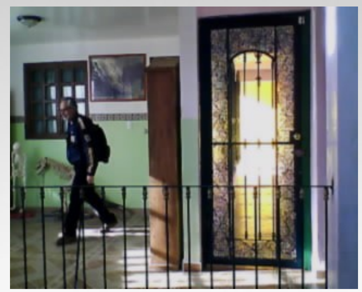
Trainning New Objects in a Deep Neural Network Using YOLO (Edgar Silva)

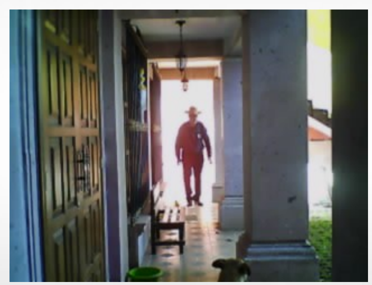


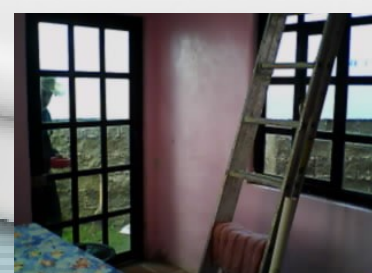


INPUT LAYER

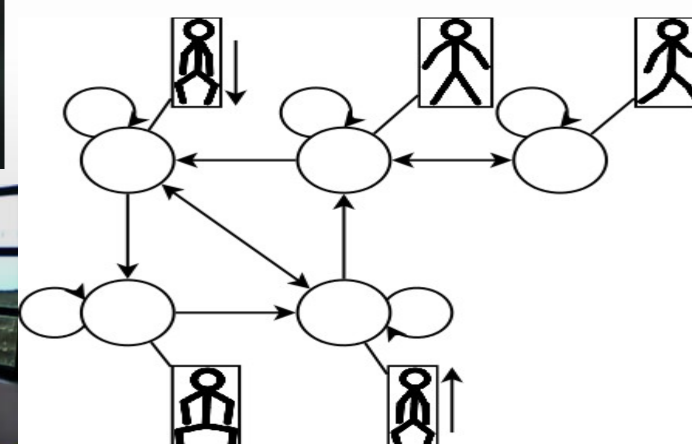
Finding persons and their actions in the environment using deep neural networks (Edgar Vazquez)





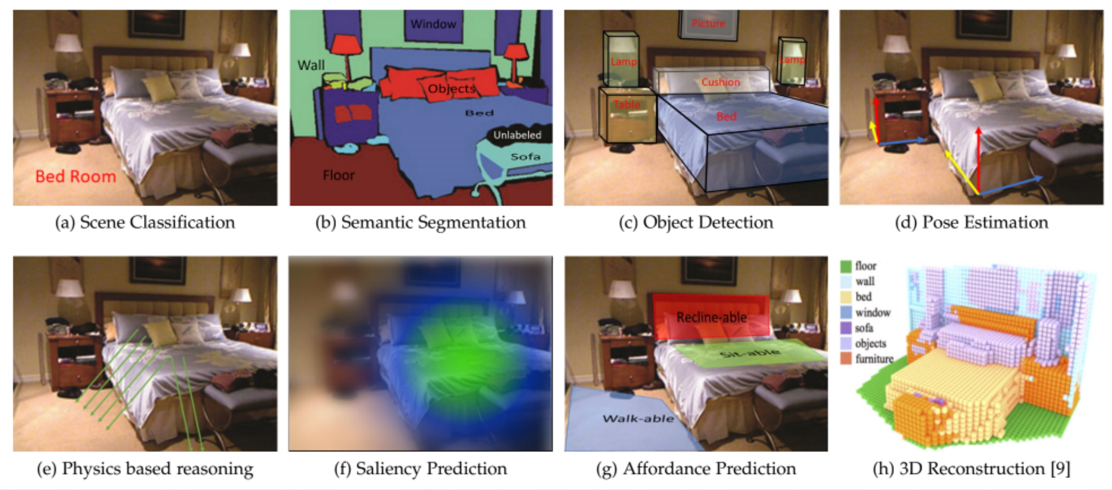


The Use of Hidden Markov Models (HMM) for the Recognition of Human Activities



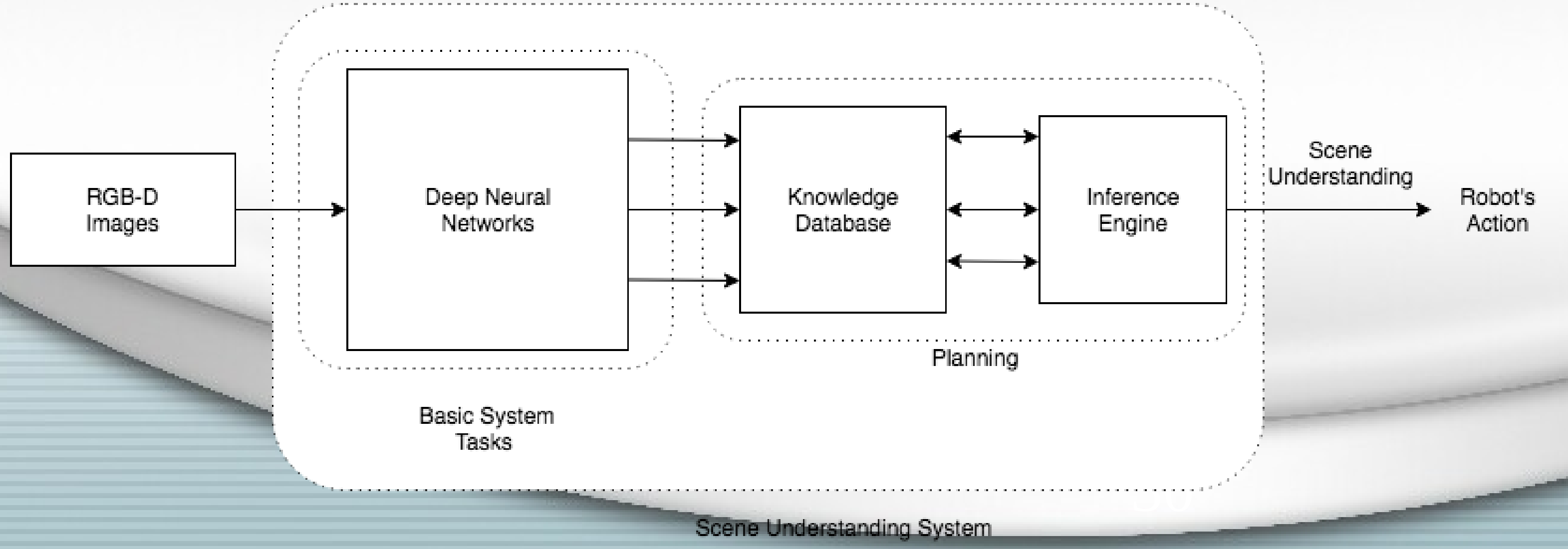
INPUT LAYER

Scene Classification and Understanding (Hugo Leon)

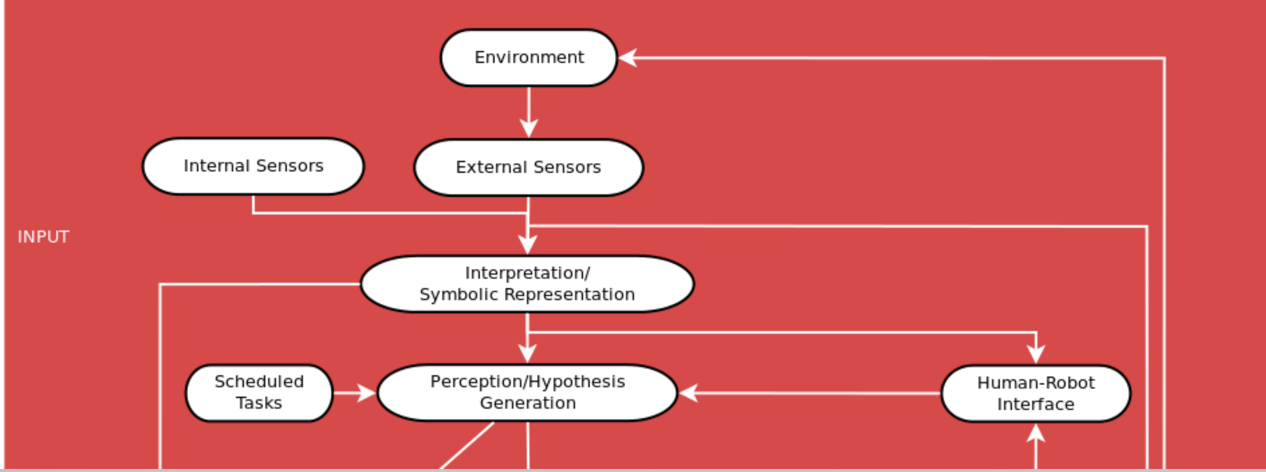


(a) Scene Classification (b) Semantic Segmentation (c) Object Detection (d) Pose Estimation
(e) Physics based reasoning (f) Saliency Prediction (g) Affordance Prediction (h) 3D Reconstruction [9]

N. Silberman, "Indoor segmentation and support inference from rgbd images," Computer Vision–ECCV 2012, pp. 746–760, 2012.



INPUT LAYER



SCHEDULED TASKS

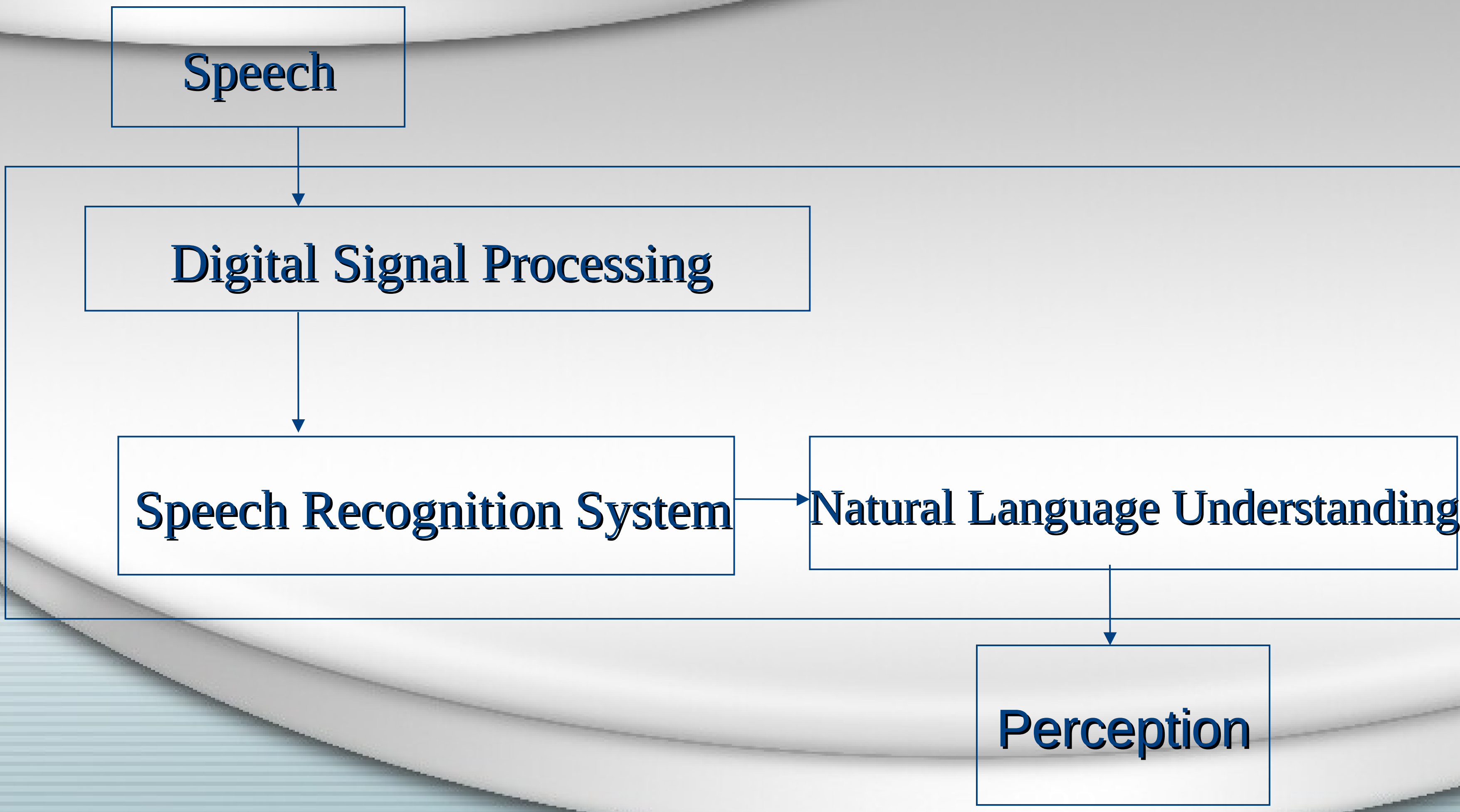
A set of tasks that the robot needs to perform in the time they were scheduled.

HUMAN / ROBOT INTERFACE

Communication between the user and the robot can be through voice and manual gestures.

The robot responds using synthetic voice and simple facial expressions.

INPUT LAYER
HUMAN / ROBOT INTERFACE
Speech Recognition System



INPUT LAYER
HUMAN / ROBOT INTERFACE
Natural Language Understanding

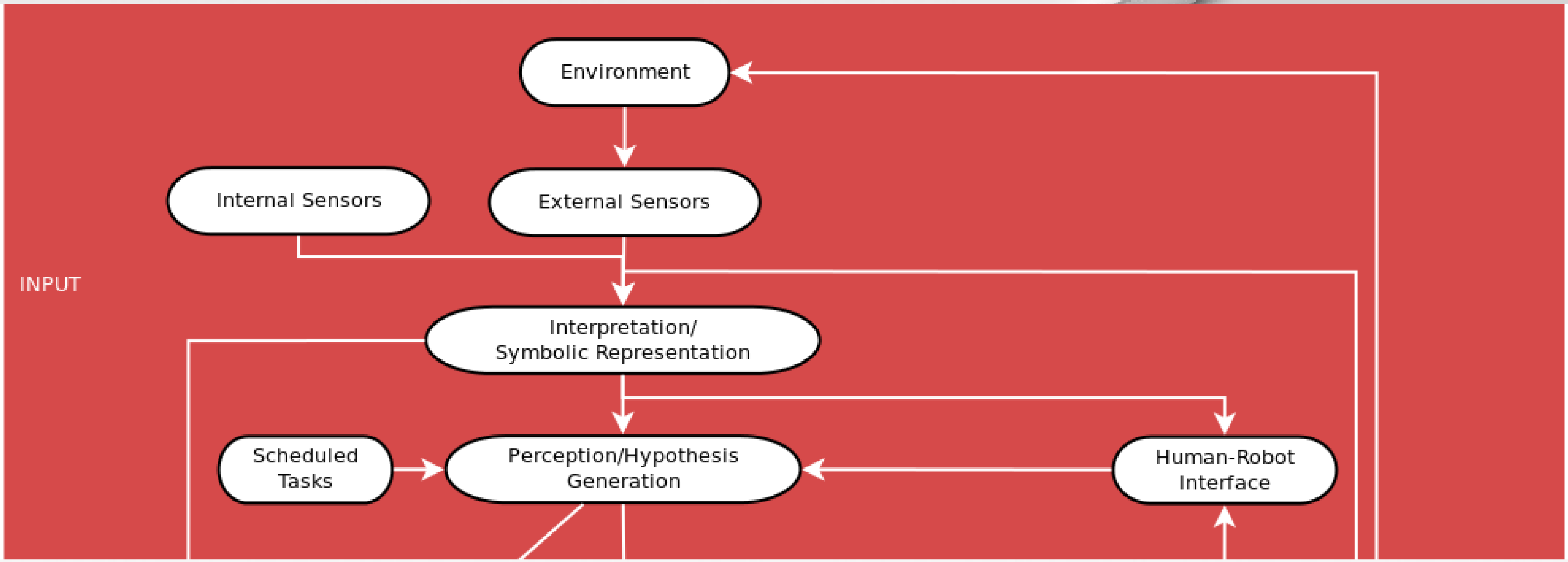
* "Robot, give the newspaper to Dad"
* "Please bring me the newspaper that is there"
  (Dad is giving the order to the robot)

Both phrases are represented using the following Conceptual Dependency primitive:

(ATRANS (ACTOR Robot) (OBJECT newspaper)
(TO Dad) (FROM newspaper-place))

With this representation a planner makes an action plan to achieve what the robot is being asked for.

INPUT LAYER



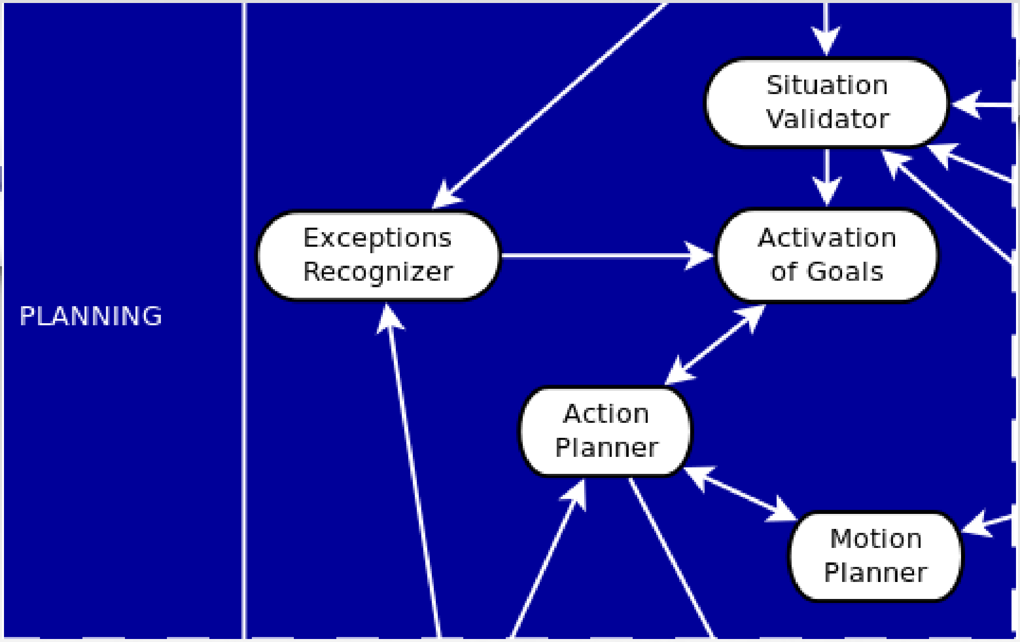
Digital signal processing techniques are applied to the data provided by the internal and external sensors to obtain a symbolic representation of the data.

With the symbolic representation, this module generates a series of beliefs, that represent the state of the environment where the robot interacts.

PERCEPTION

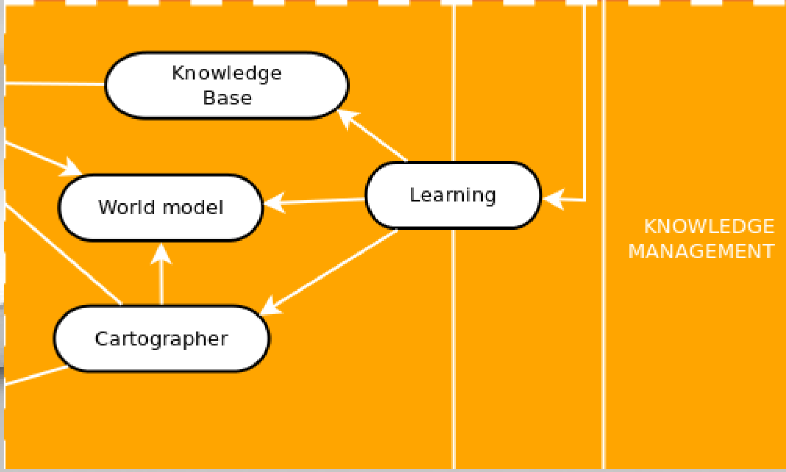
Example: In the following figure, the symbolic representation generates the beliefs: "there is a hole in front of the robot" or "there is a shadow in front of the robot"

PLANNING LAYER



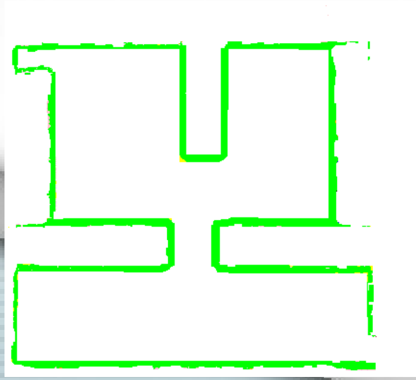
The beliefs generated by the perception module are validated by this layer, it uses the Knowledge Management layer to validate them, thus a situation recognition is created.

KNOWLEDGE MANAGMENT LAYER

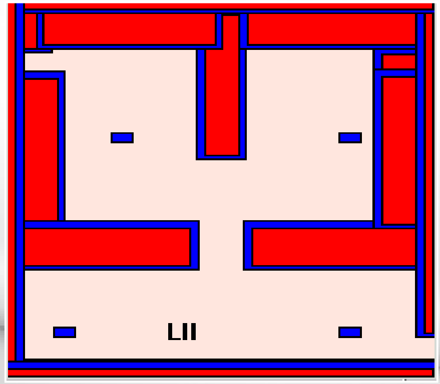


Cartographer

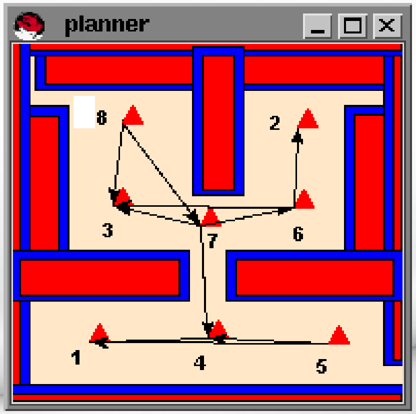
This layer has different types of maps for the representation of the environment, they are created using SLAM techniques.



Raw map



Simbolic map



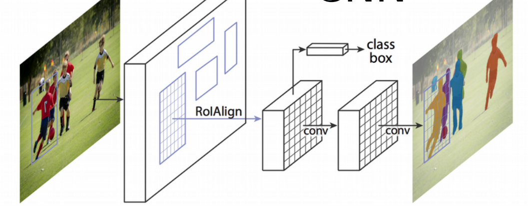
Topological map

KNOWLEDGE MANAGMENT LAYER

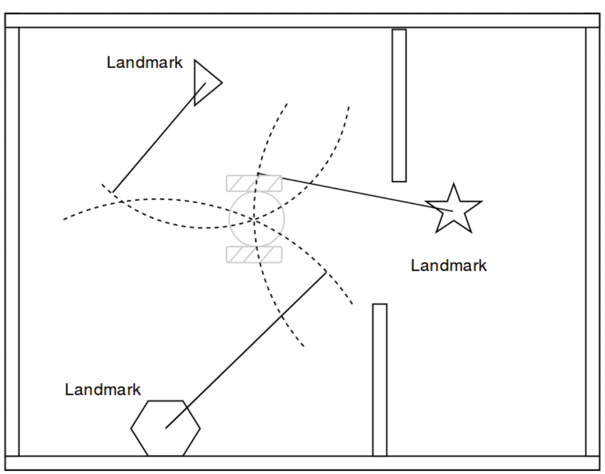
Cartographer

RGB-D Camera

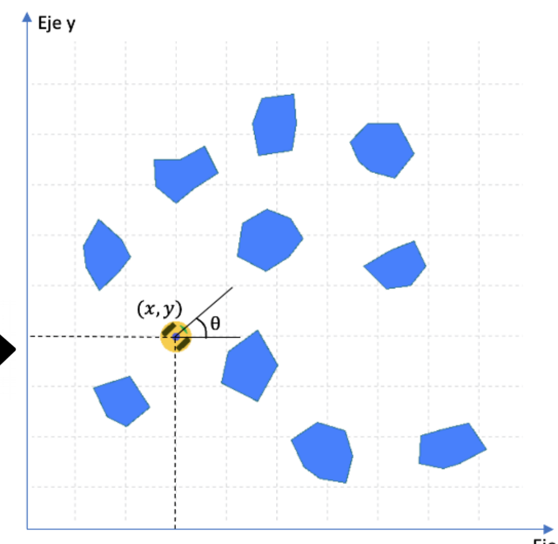
CNN



The Kalman filter and deep neural networks are used to estimate the robot's position and orientation (Diego Cordero)

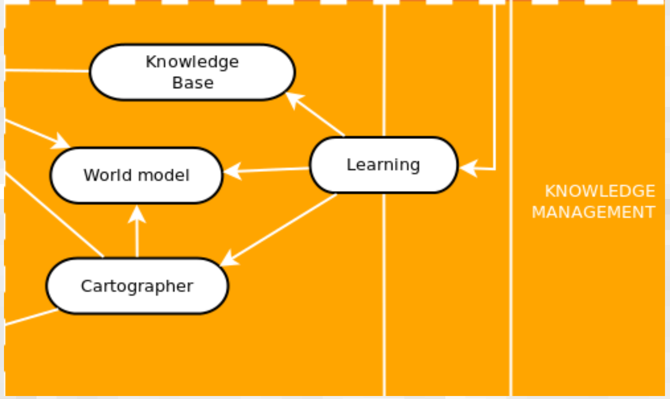


Estimation of the robot’s pose using the Kalman Filter



KNOWLEDGE MANAGMENT LAYER

WORLD MODEL



There is a representation of the objects, people and places where the robot interacts and the relationships that exist between them. There are 5 structures that represent objects, rooms, furniture, humans and the robot

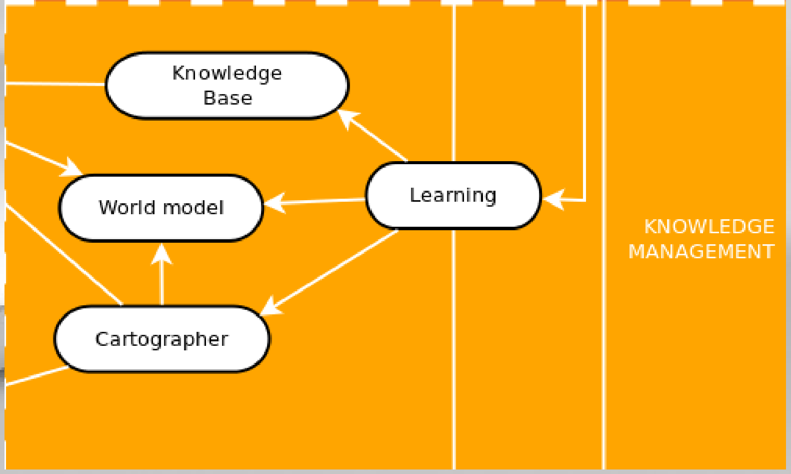
```
(Human
   (name Mother)
   (room studio)
   (zone couch)
   (objects book)
   (pose 1.8 2.0 0.5)
   (locations main-bedroom)
)
```

```
(Furniture
   (name kitchen-table)
   (room kitchen)
   (zone left-side)
   (pose 0.2 2.3 1.0)
   (use eating)
   (attributes fixed)
)
```

```
(Object
   (name newspaper)
   (room outside)
   (zone outside-door)
   (pose 0.5 1.3 0.1)
   (possession nobody)
   (use information)
   (locations living kitchen)
   (attributes movable)
)
```

All this information is updated by the input layer and by the actions of the robot.

KNOWLEDGE MANAGMENT LAYER



KNOWLEDGE BASE

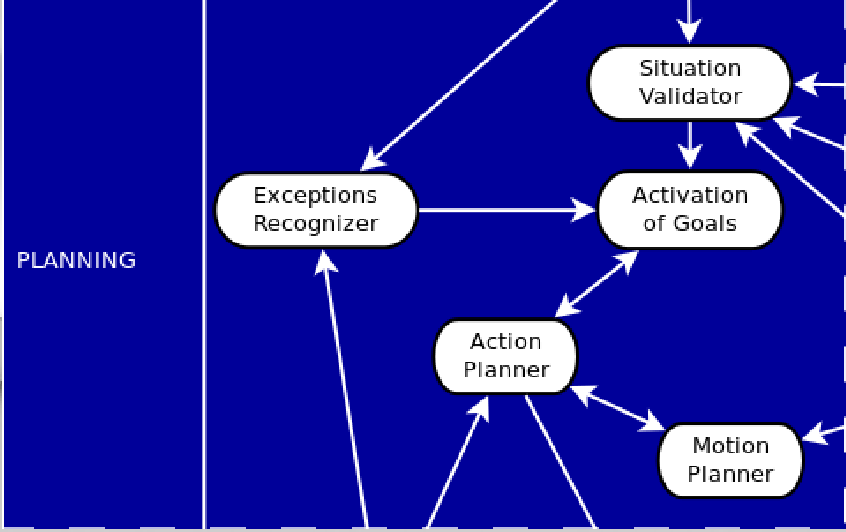
Using a rule-based system, CLIPS (NASA), robot knowledge is represented. This is represented by production rules, which correspond to the actions that the agent would do if certain conditions are met.

For the previous R2R2 example the following rule would be activated:

Shadow Rule {
If there are trees around the robot's path and it is a clear day then there will be a shadow in the path.
}

PLANNING LAYER

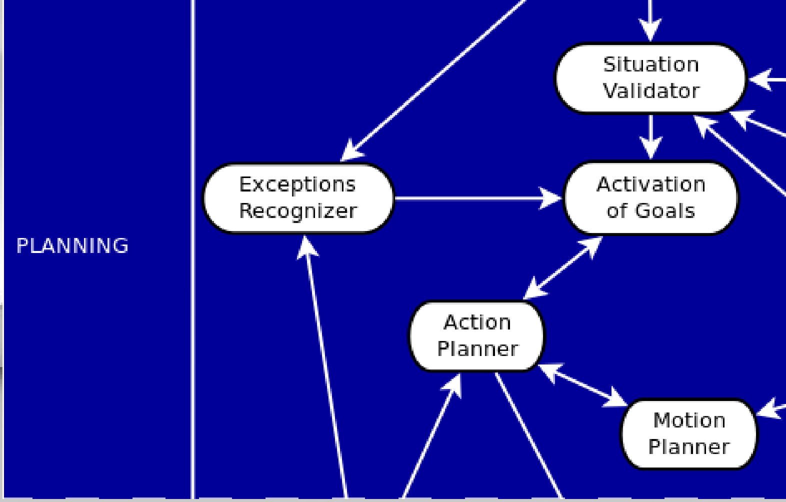
Situation Validator



In the previous example, the cartographer does not report a hole in the area in front of the robot and the representation of knowledge reports a shadow.

Then, with these facts the new situation is that there is no danger and that the robot must continue towards its destination.

PLANNING LAYER

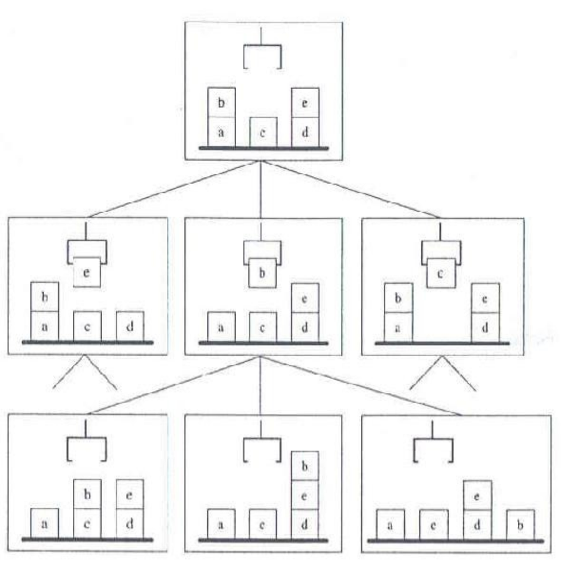


Activation of Goals

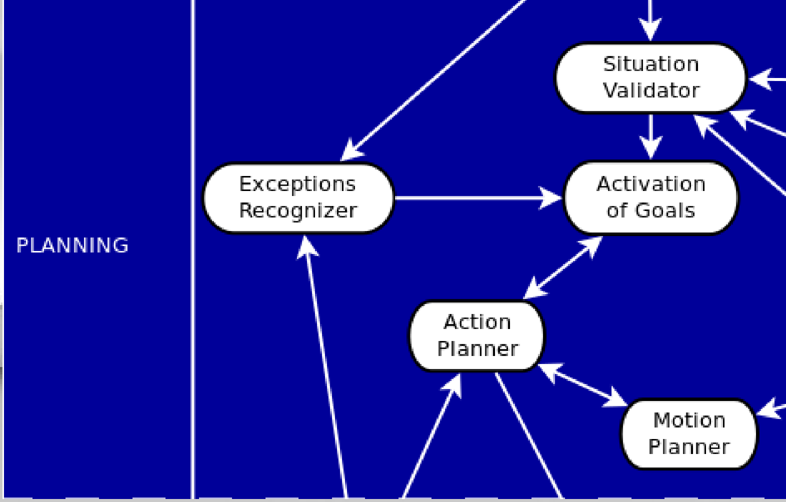
Given a situation recognition, a set of goals are activated to solve it.

Action Planner

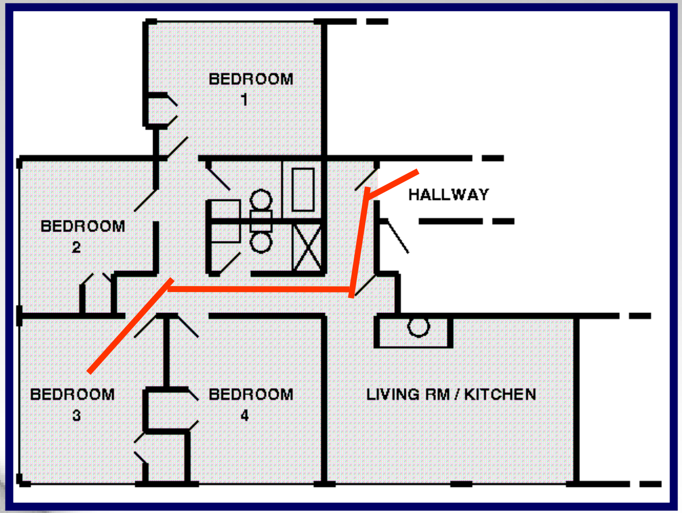
Action planning finds a sequence of physical operations to achieve the activated goals.



PLANNING LAYER

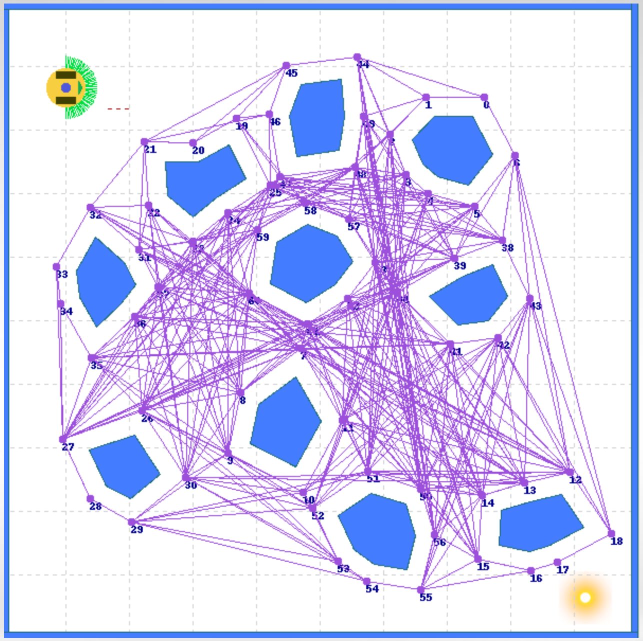


Motion Planner



Global Path:

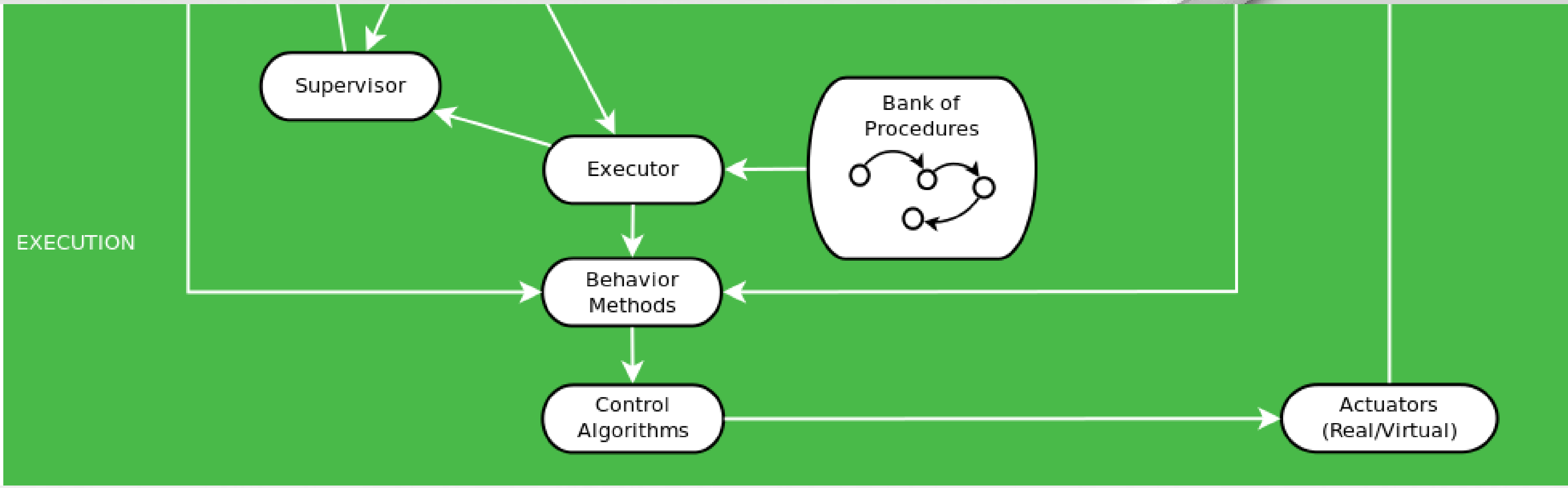
Path between rooms.



Local Path:

Path inside each room.

EXECUTION LAYER



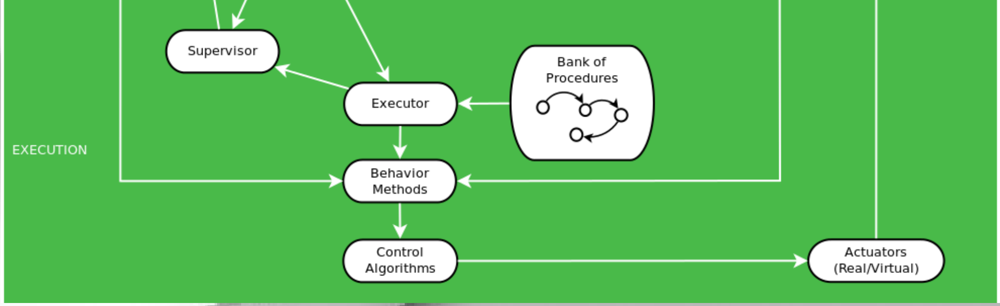
This layer executes the actions and movements plans and it checks that they are executed accordingly.

A set of hard-wired procedures, represented by state machines, are used to partially solve specific problems, finding persons, object manipulation, etc.

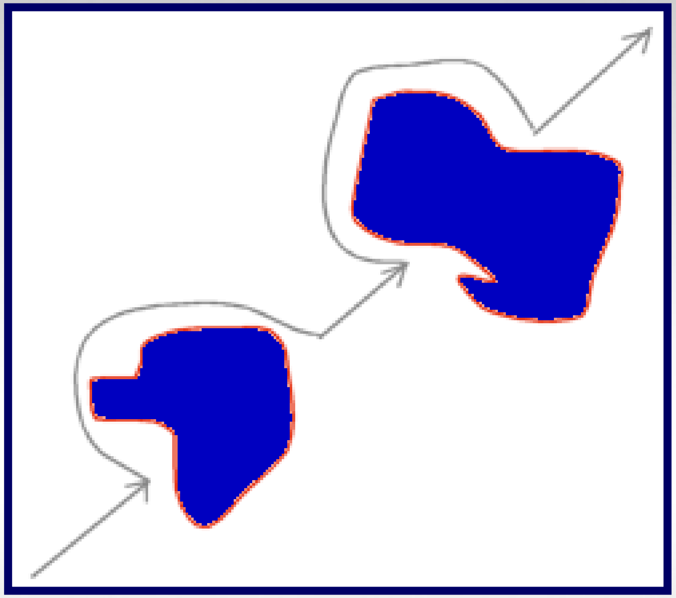
The executor uses these bank of procedures to execute a plan.

EXECUTION LAYER

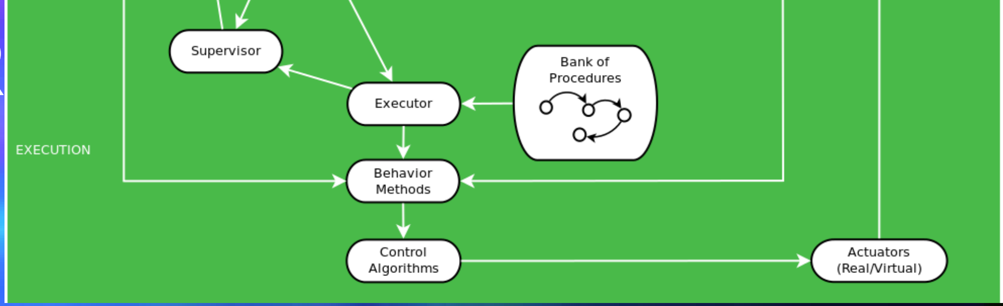
Behavior Methods



Behaviour methods are used to avoid obstacles not contemplated by the movements planner.
The behaviour methods are state machines, potential fields and neural networks.



EXECUTION LAYER

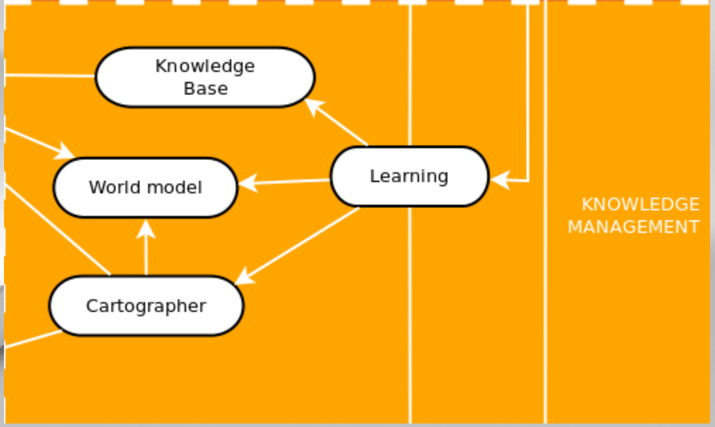


CONTROL ALGORITHMS

Control algorithms, like PID, are used to control the operation of the virtual and real actuators.

KNOWLEDGE MANAGMENT LAYER

Learning



Genetic algorithms and programming

Probabilistic methods:

- Markov chains
- Bayesian classifiers.

Clustering (K-means, Vector Quantization)

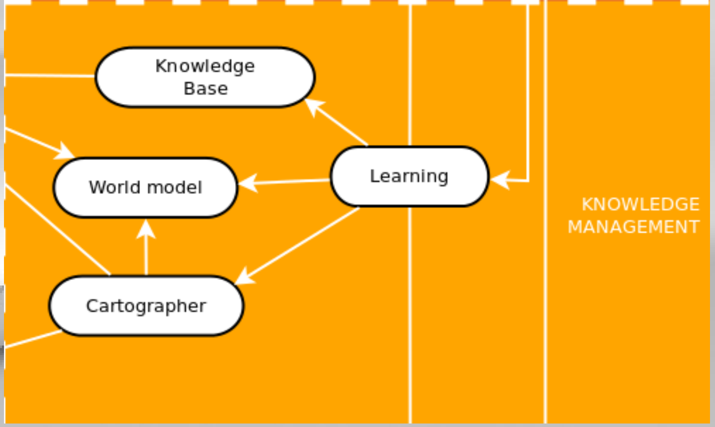
Artificial Neural Networks

Reinforcement Learning

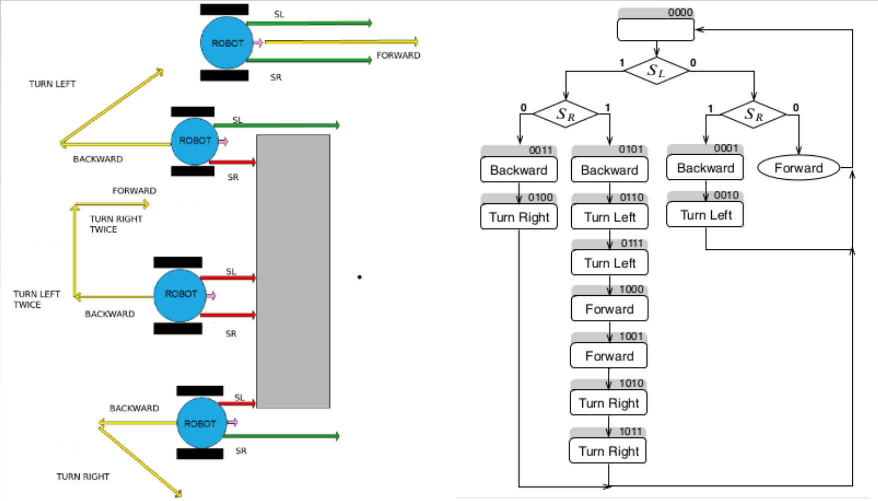
KNOWLEDGE MANAGMENT LAYER

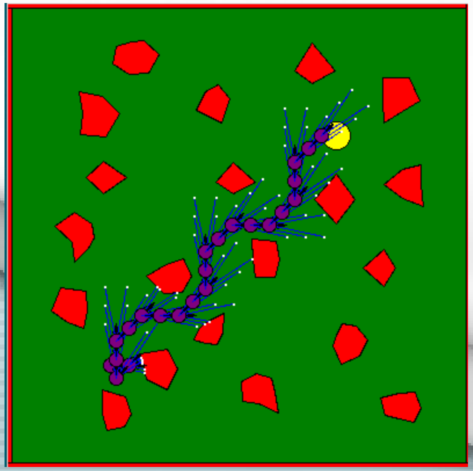
Learning

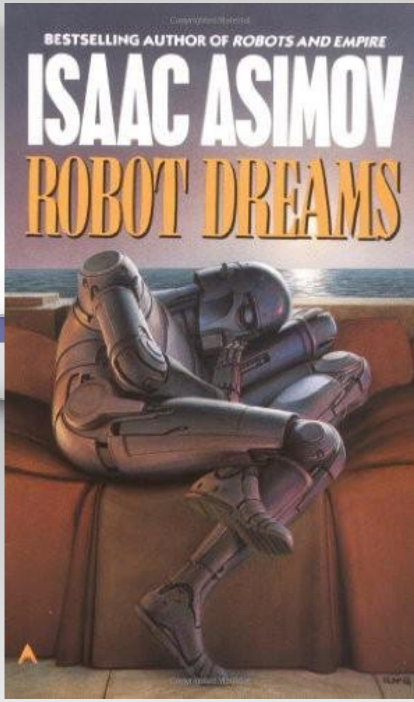
Genetic algorithms and programming



The goal is to use an optimization algorithm, as genetic algorithms (GA), to find the best robot behaviors to avoid obstacles while they tried to reach a destination.



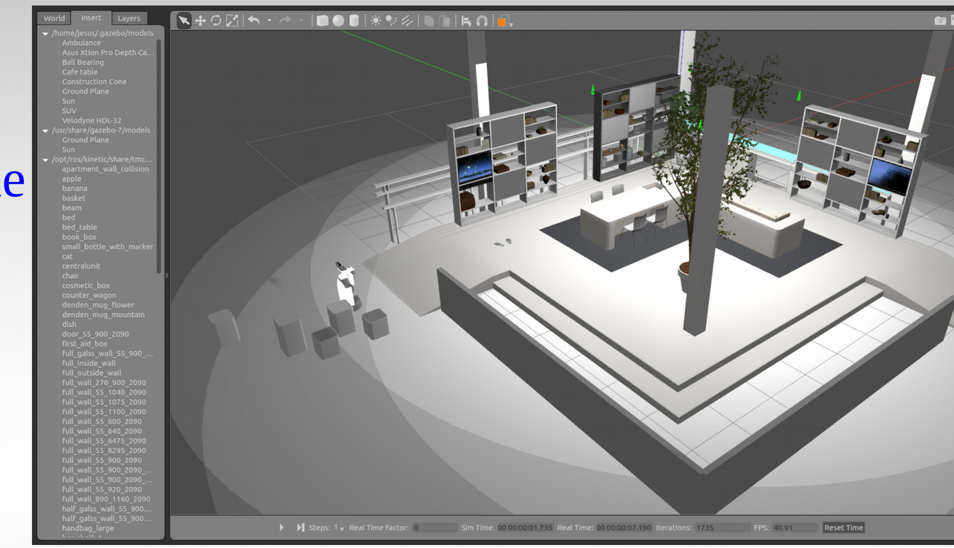




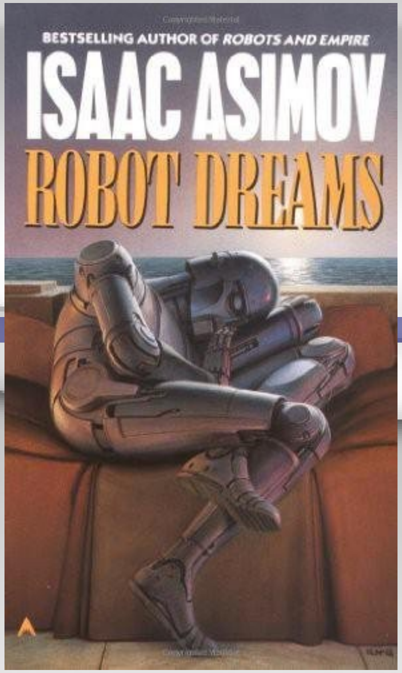
Robot Learning Using Simulators

With simulated images the robot is trained to navigate in new environments while it is in resting mode.

Gazebo 3D robot simulator integrated with a physics engine (Oscar Fuentes)

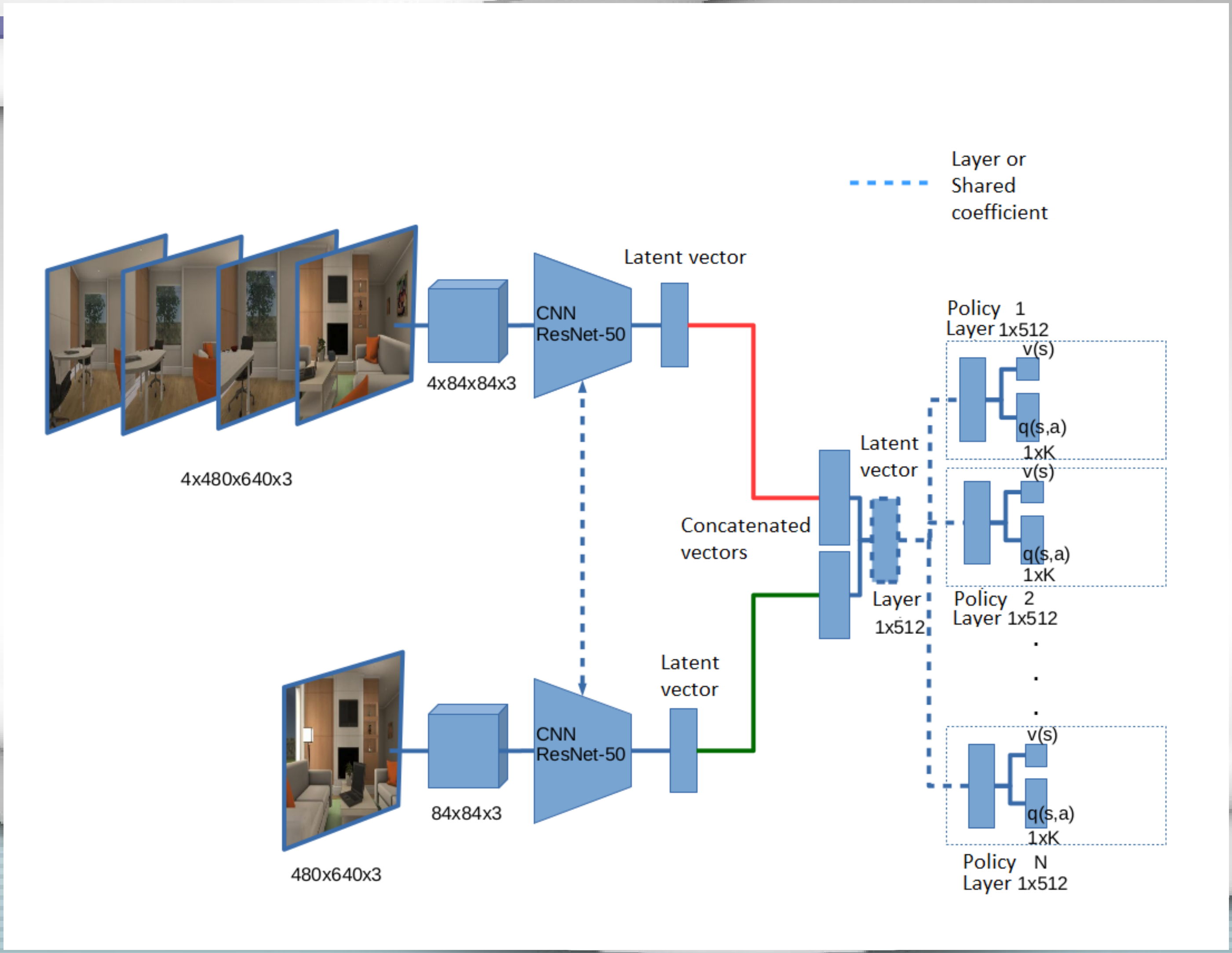


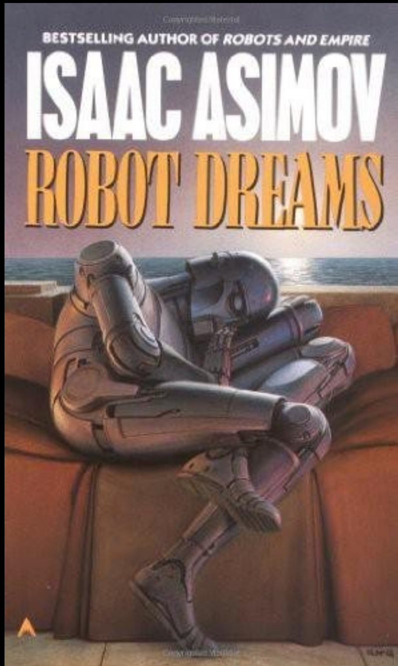
The House Of inteRactions THOR, Allen Institute for Artificial Intelligence (Adrian Sarmiento)



Robot Learning Using Simulators

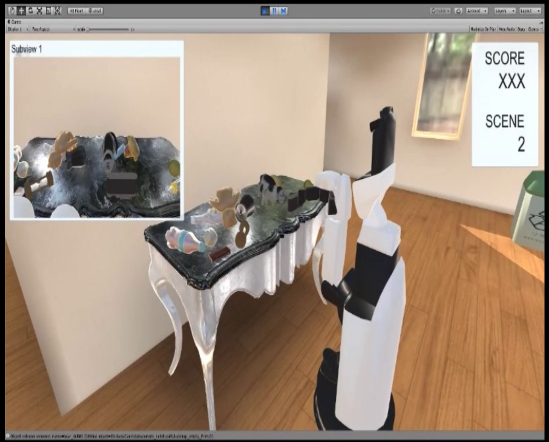
THOR Simulator used to do MDP reinforcement learning for robot's movement. (Adrian Sarmiento)

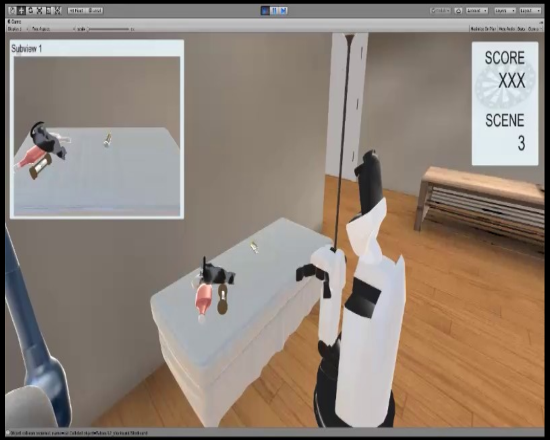


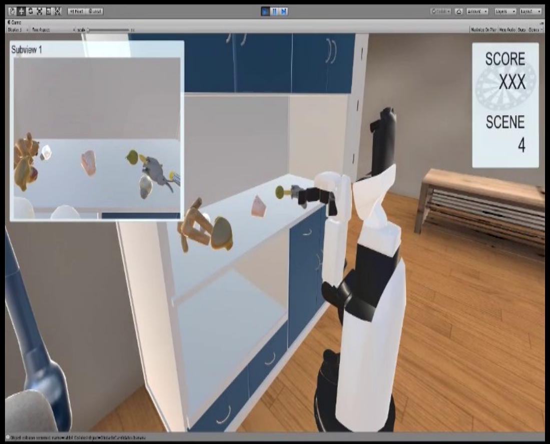


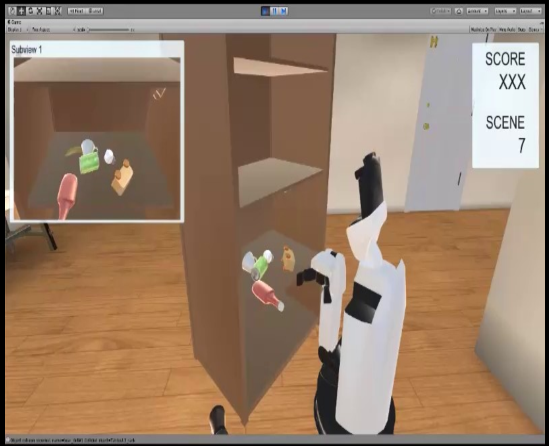
Robot Learning Using Simulators

Unity (Angelica Nakayama)

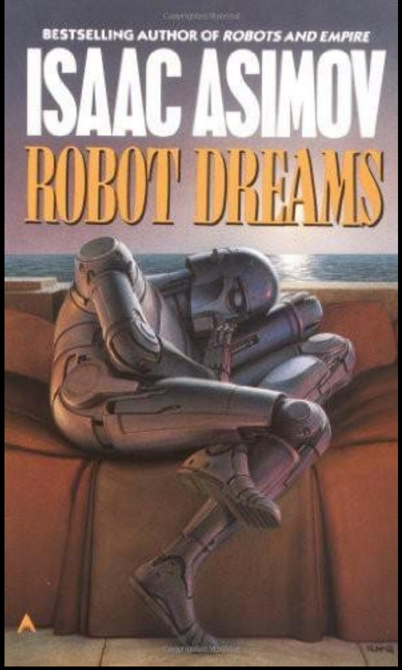








Various obstacle layouts and illumination conditions for the robot to learn how to place objects.



Proposed method: PonNet (A. Nakayama)

- Pre-trained k layers of ResNet-50 as feature extractor
- Additional attention mechanism for balancing RGB and Depth

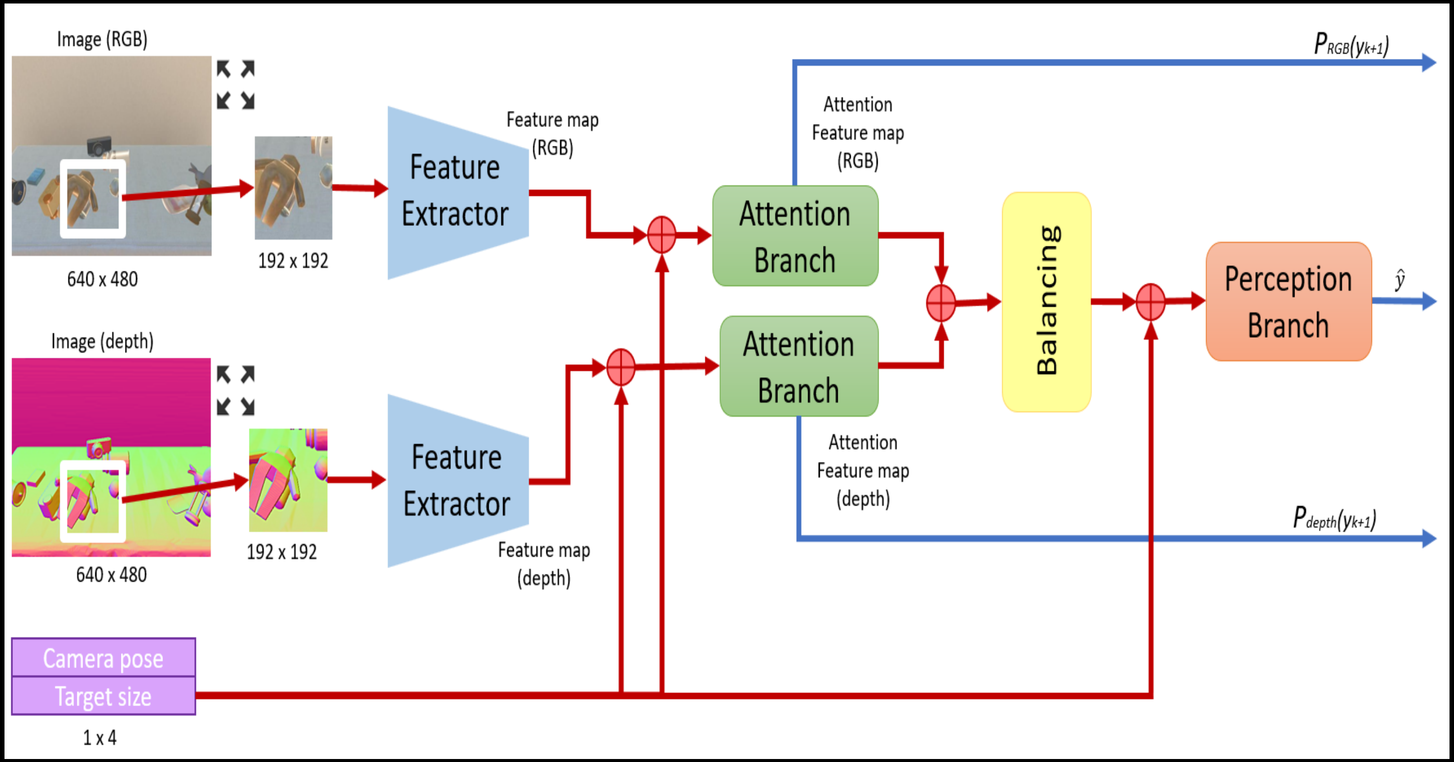
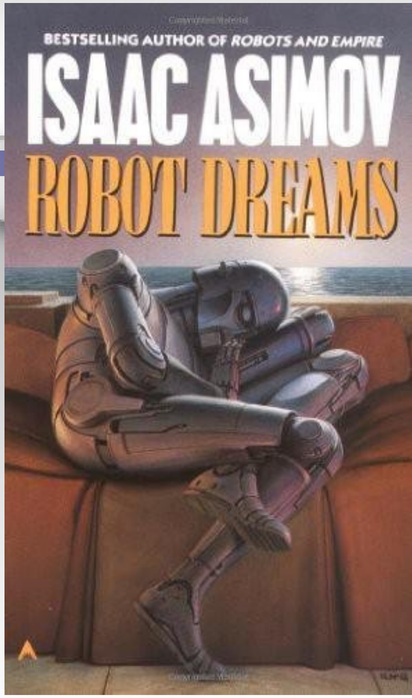
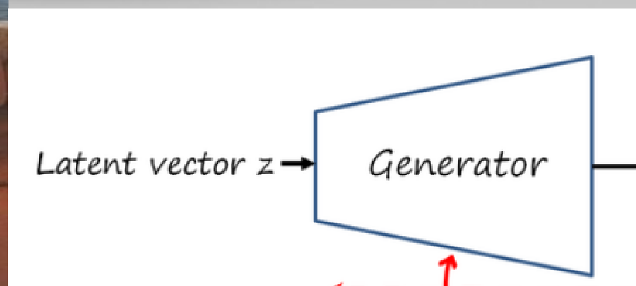


Image Synthesis with Generative Adversarial Nets (GAN)

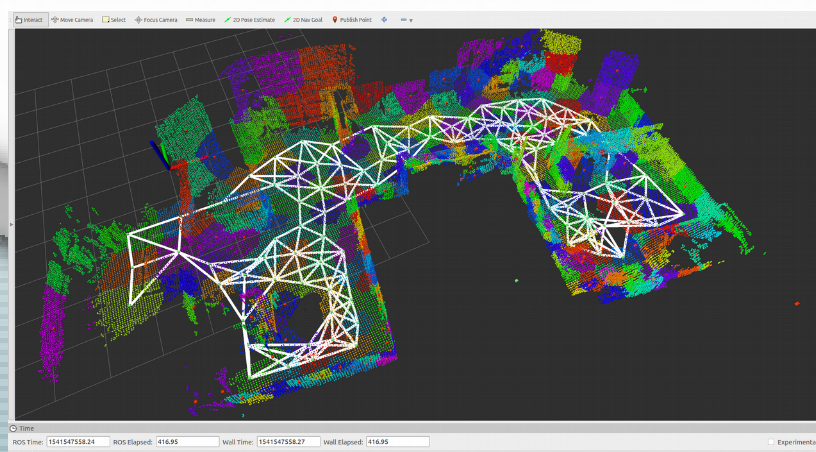
With synthetic images the robot is trained to navigate in new environments while it is in resting mode. (Future Work)





The generator creates synthetic images of the environment.

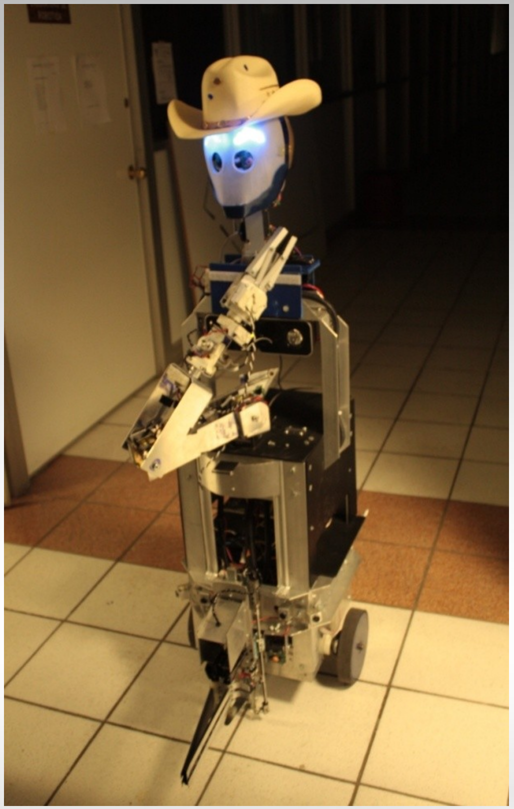
Training Path



Robots

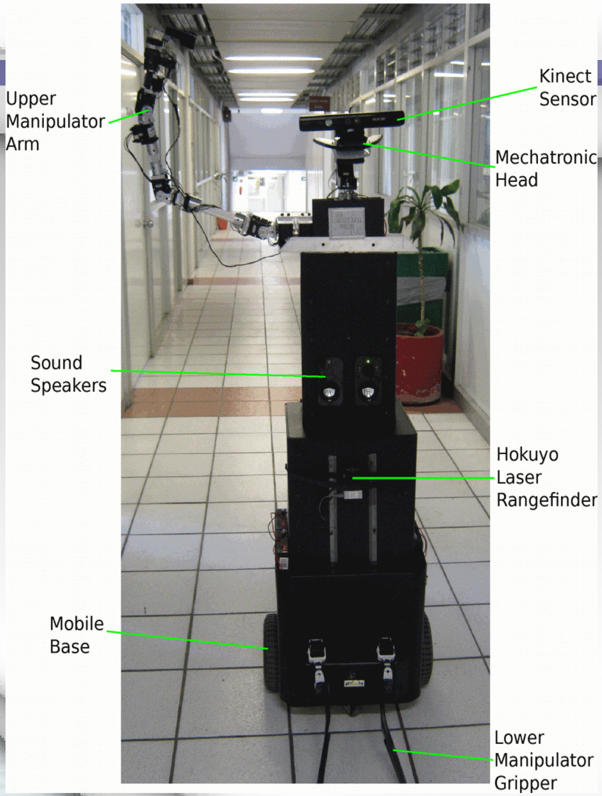
Robot TX8
(1999)

Robot TPR8
(2006)



Robot PACK-ITO
(2009)

Robots



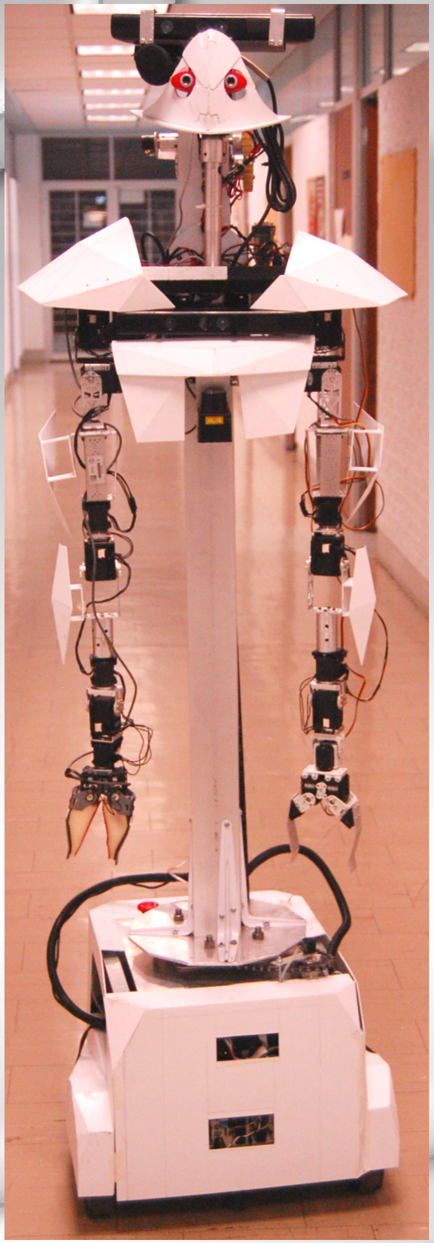
Robot AL-ITA
(2011)

Robot JUST-INA
(2012 - ...)

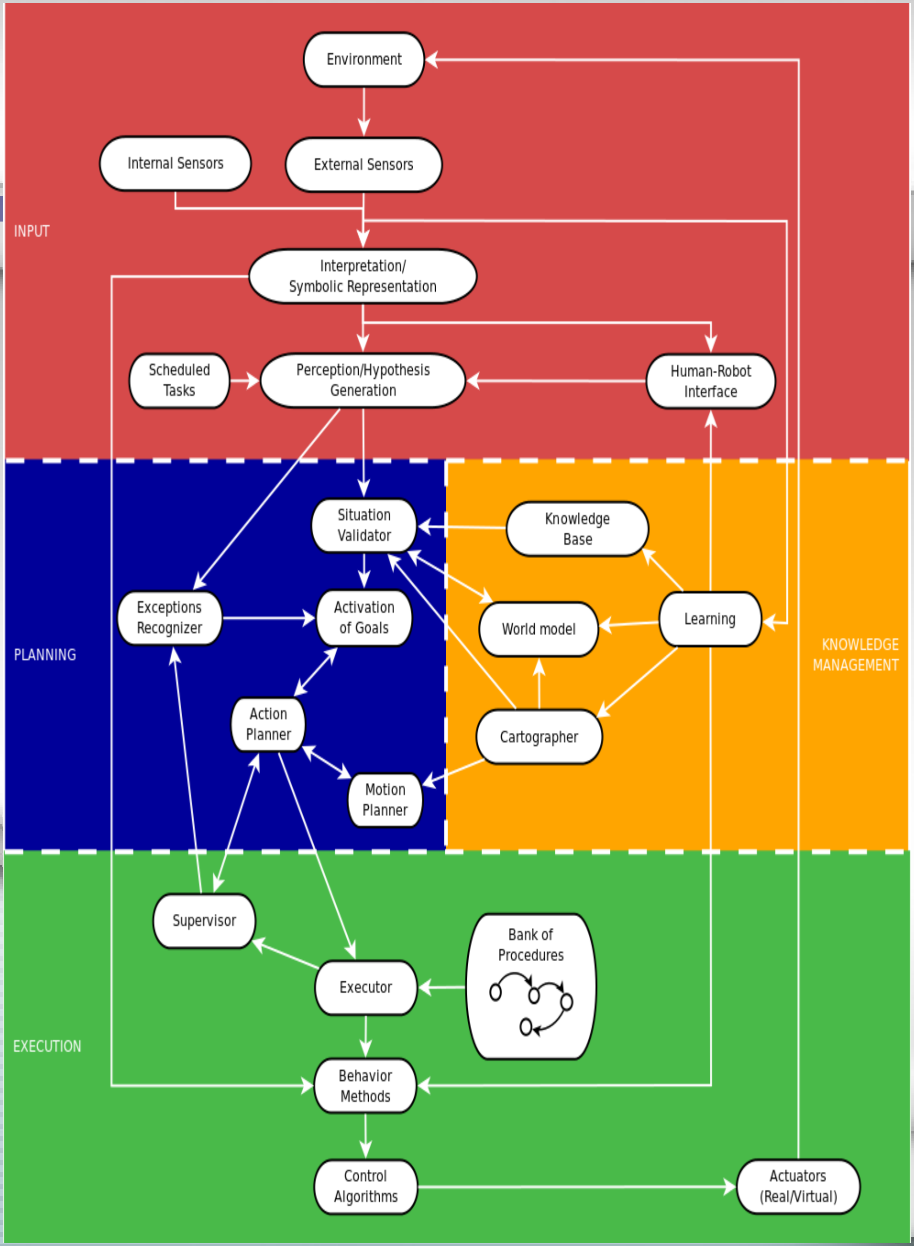
Robot Takeshi Toyota
(2018 - ...)

Robot Justina

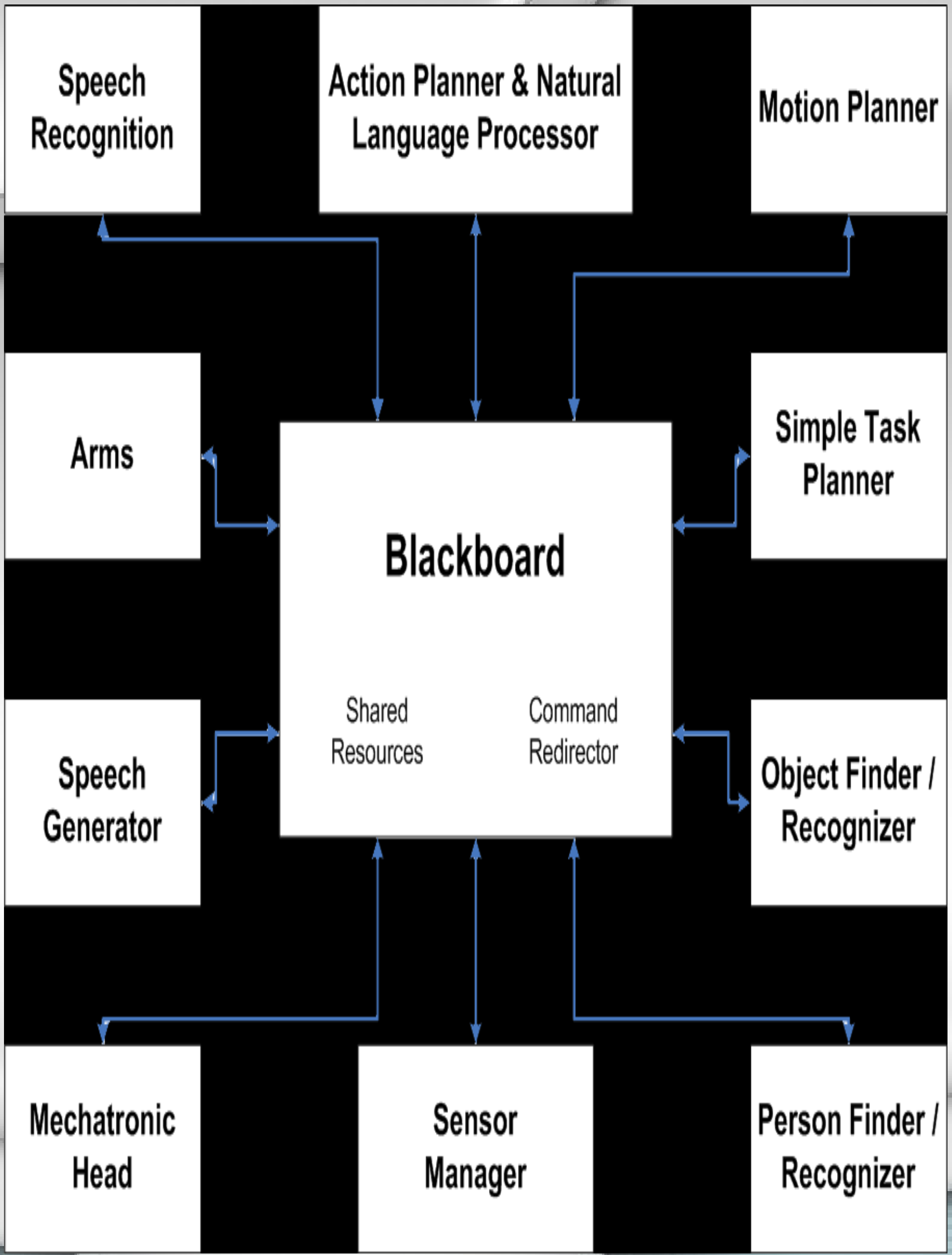
- Mechatronic Head (Pan and Till)
  - Kinect
  - Stereo Cameras
  - Microphone
- Torso
  - Kinect
  - Laser Hokuyo
- Two arms (7 DoF)
- Mobile Base



Blackboard

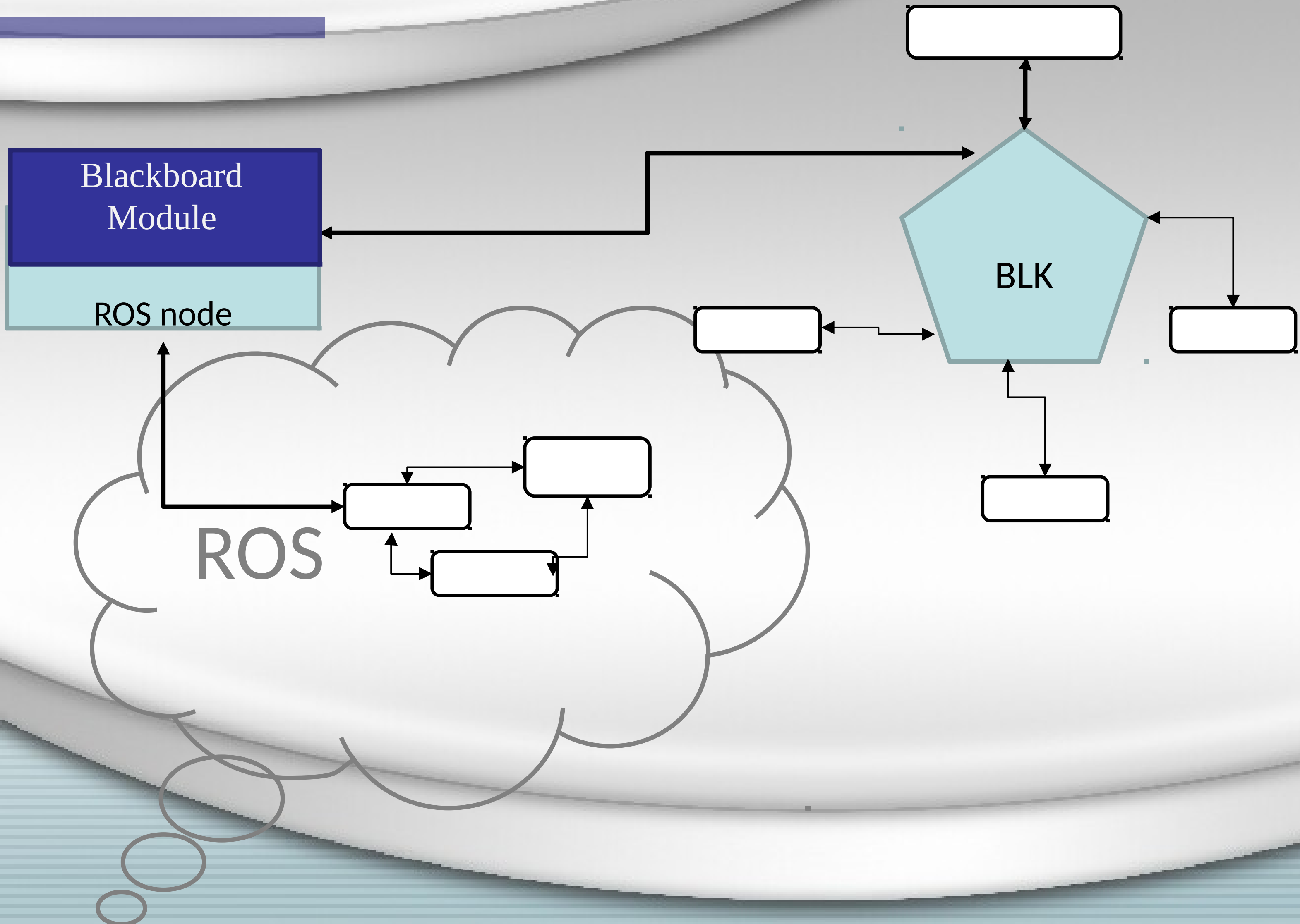


Environment
Internal Sensors
External Sensors
INPUT
Interpretation/
Symbolic Representation
Scheduled
Tasks
Perception/Hypothesis
Generation
Human-Robot
Interface
Situation
Validator
Knowledge
Base
Exceptions
Recognizer
Activation
of Goals
World model
Learning
PLANNING
KNOWLEDGE
MANAGEMENT
Action
Planner
Cartographer
Motion
Planner
Supervisor
Bank of
Procedures
Executor
EXECUTION
Behavior
Methods
Control
Algorithms
Actuators
(Real/Virtual)



Speech
Recognition
Action Planner & Natural
Language Processor
Motion Planner
Arms
Blackboard
Simple Task
Planner
Shared
Resources
Command
Redirector
Speech
Generator
Object Finder /
Recognizer
Mechatronic
Head
Sensor
Manager
Person Finder /
Recognizer

BLACKBOARD ROBOT OPERATING SYSTEM (ROS)



RoboCuP @Home Participation

2007 Atlanta, USA,
Third Place

2018 Montreal, Canada,
Second Place

2019 Sydney, Australia,
Second Place

Justina in China, 2015

Video


Robocup@Home 2014
Qualification Video
Justina
A General Purpose Service Robot
by PUMAS@HOME

Video


pumas@home
Qualification Video 2017
INGENIERIA
ROBOTICS UNAM
Pumas
SINCE 2006

Robotics, AI and Machine Vision

Jesus Savage

Bio-Robotics Laboratory
biorobotics.fi-p.unam.mx

School of Engineering
National Autonomous University of Mexico, UNAM

15th International Conference on Electromechanics and Robotics "Zavalishin's Readings - 2020