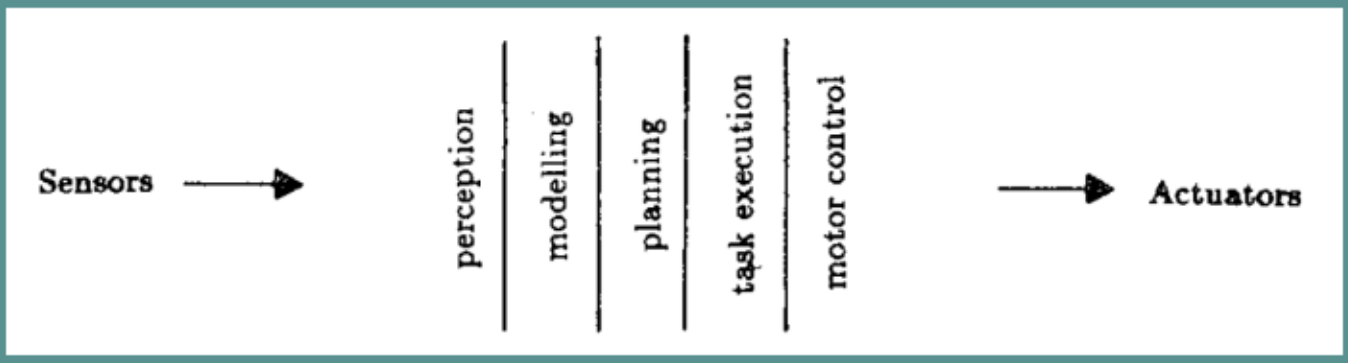
Pico Behavioral Based Robotics

Presenters:
Adam Burdett, Tanner Grimm

Traditional Robot Control System Architecture



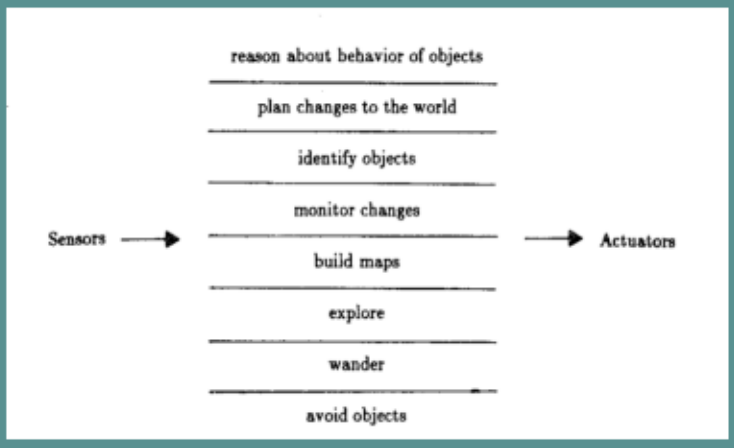
Modules: Perception, Modelling, Planning, Task Execution, Motor Control

Problems: -instance of each piece must be built in order to function at all
-changes to a module:
-must not affect other modules -must be propagated to other modules

This slide is meant to show what a traditional robot control system architecture is modeled like. The vertical modules represent each module being on the same layer of abstraction in a way that ties them all together; that is, each module is connected and must be implemented in order to have a working robot control system. Mentioned on the slide is the problem of making changes to the control system. If a change is made, that change must either not affect the other modules in any way, or the change must be propagated to each other module that it affects which is not a very cohesive design.

-all details in this slide come from Rodney A. Brooks in “A Robust Layered Control System For A Mobile Robot”

Subsumption Architecture



-Rodney A. Brooks
-renowned roboticist
-former Director of CS & AI Lab at MIT
-iRobot (roomba)
-"A Robust Layered Control System For A Mobile Robot" (1986)
-Layers of a control system
-early working control system
-additional layers can be added without changing the original system
-individual layers can concurrently run

Subsumption architecture was first proposed by Rodney A. Brooks, a renowned roboticist and former Director of Computer Science and the Artificial Intelligence lab at MIT. He is also known for his company iRobot which has produced the roomba vacuuming robot which actually has a control system designed with subsumption architecture. Brooks' subsumption architecture was proposed in his paper "A Robust Layered Control System For a Mobile Robot" which was published in 1986. As opposed to the vertical modules seen in a traditional robot control system on the previous slide, a subsumption architecture contains horizontal modules which are layered; each one is called a layer of the control system. The lower layers tend to contain the more basic, necessary functionalities needed for the robot to operate. The benefit of this design is that we have a early working control system as long as the lowest layer in the architecture has been implemented. It is also important to note that each layer can concurrently run and the upper layers can decide whether or not to override lower layer processes in order to accomplish a task through arbitration (which will be touched on later in the presentation).

(TANNER) BackGround(what is Subsumption architecture):

-definition
-souces:
https://www.youtube.com/watch?v=A-fxij3zM7g
-synthesis is always easier than analysis, main theme from vehicles of
.

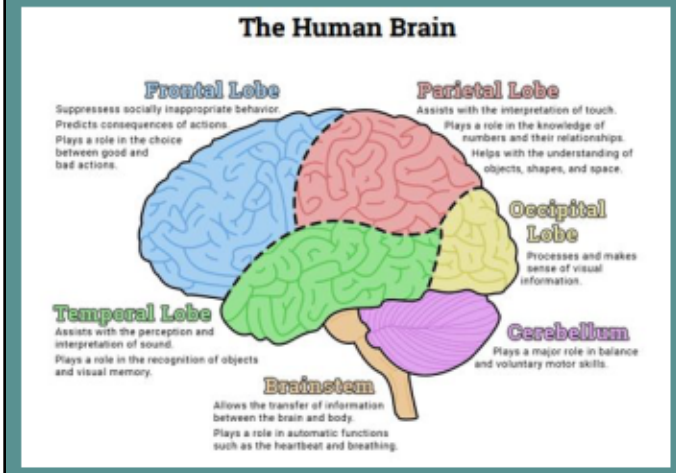
-Vehicles: experiments in synthetic psychology by Valentino Braitenberg

-Robot Programming by Joseph L. Jones (details subsumption as proposed by Rodney Brooks)

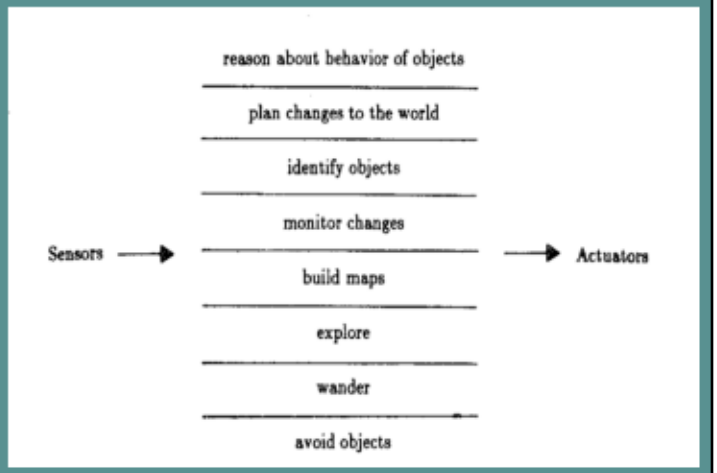
-Rodney Brooks “A Robust Layered Control System for a Mobile Robot”: details subsumption.

referenced from subsumptionBrooks.pdf
Diagram used in subsumptionBrooks.pdf

Subsumption Architecture



-primitive functions never lost
-higher functions appended
-primitive functions always present

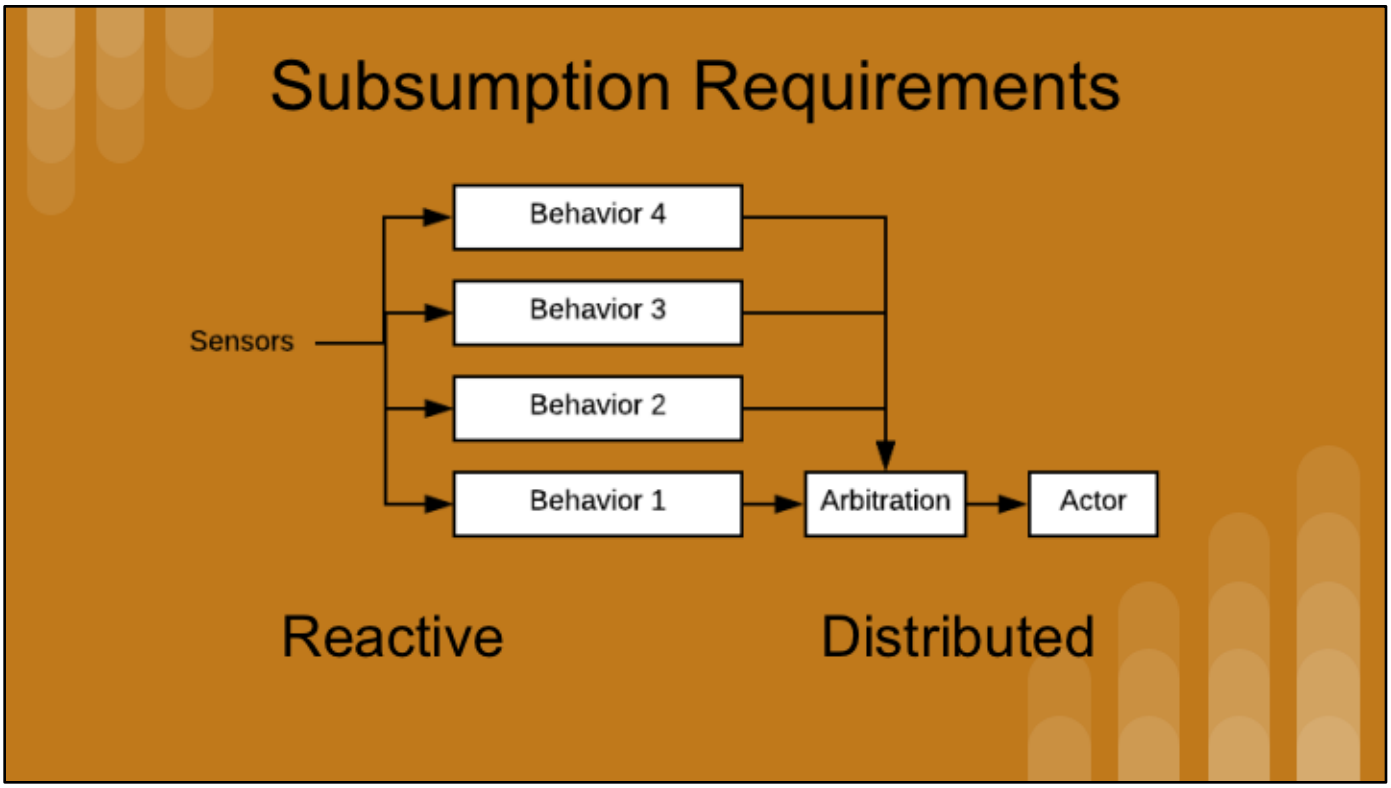


-primitive abilities programmed first
-motor control, collision behavior, etc.
-higher functions added
-primitive abilities always present

Subsumption architecture is based off of the development of the brain itself. The brain has different lobes and parts that have different tasks to accomplish or manage. The brainstem is the most basic part of the brain which controls the basic functions which are necessary for life, such as heartbeat and breathing. For subsumption architecture, this module would be the lowest module on the architectures hierarchy which controls the basic functions necessary for the robot to work. Just like how the brain has developed different parts to take care of different functions and suppress lower level brainstem functions, the modules above the lowest module in a robot control system work the same way in subsumption architecture. Each layer is able to add more complex behavior routines to the robot while still maintaining primitive abilities. This model is more robust than the traditional robot control system architecture.

(TANNER) BackGround(Subsumption architecture Example):
-example
idea , we have sub behaviors from evolution ancestors…
See Jones' book page 93 for example.

Comparing brain and subsumption architecture from "Robot Programming" by Joseph L. Jones



Subsumption and Behavior Based Robotics are structured around behaviors.
Behaviors are subroutine. Small pieces of code that encapsulates a behavior.
Behaviors are expressed on certain types of sensor input and compete for expression. When two or more behaviors compete for expression, it is the job of a Arbitrator to pick which behavior is expressed. Often this is done by a priority.

To implement a Subsumption system it needs to support Reactive programming for the behaviors

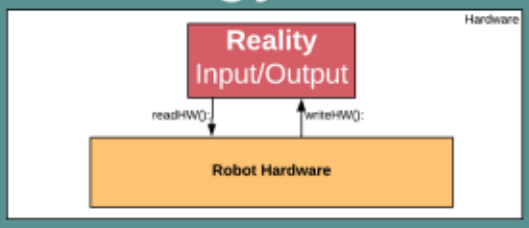
(ADAM) Problem:

What needs does subsumption architecture have?
- behaviors
  - -servo
  - -ballistics
- Arbitrators
  - -collisions
  - -
- distributed and modular
- easily scalable by adding a processors for a behavior..

Picos are a special kind of device shadows. Not only do the keep track of state information, but they also store the logic for the state change.
Picos fit the need to be Reactive and Distributed. Reactive programming entails an event driven and loosely coupled system where different modules can be isolated. Picos are programmed using KRL which is an event driven program, and each Pico is a digital representation of a device (device shadow), and are completely separate from each other pico which satisfies the loose coupling and isolation requirements for reactive programming. This design also bleeds over into satisfying the distributed need which requires independent systems to be linked by a network. Each pico is an independent system and can be linked to each other to form a greater network, so this is also met.

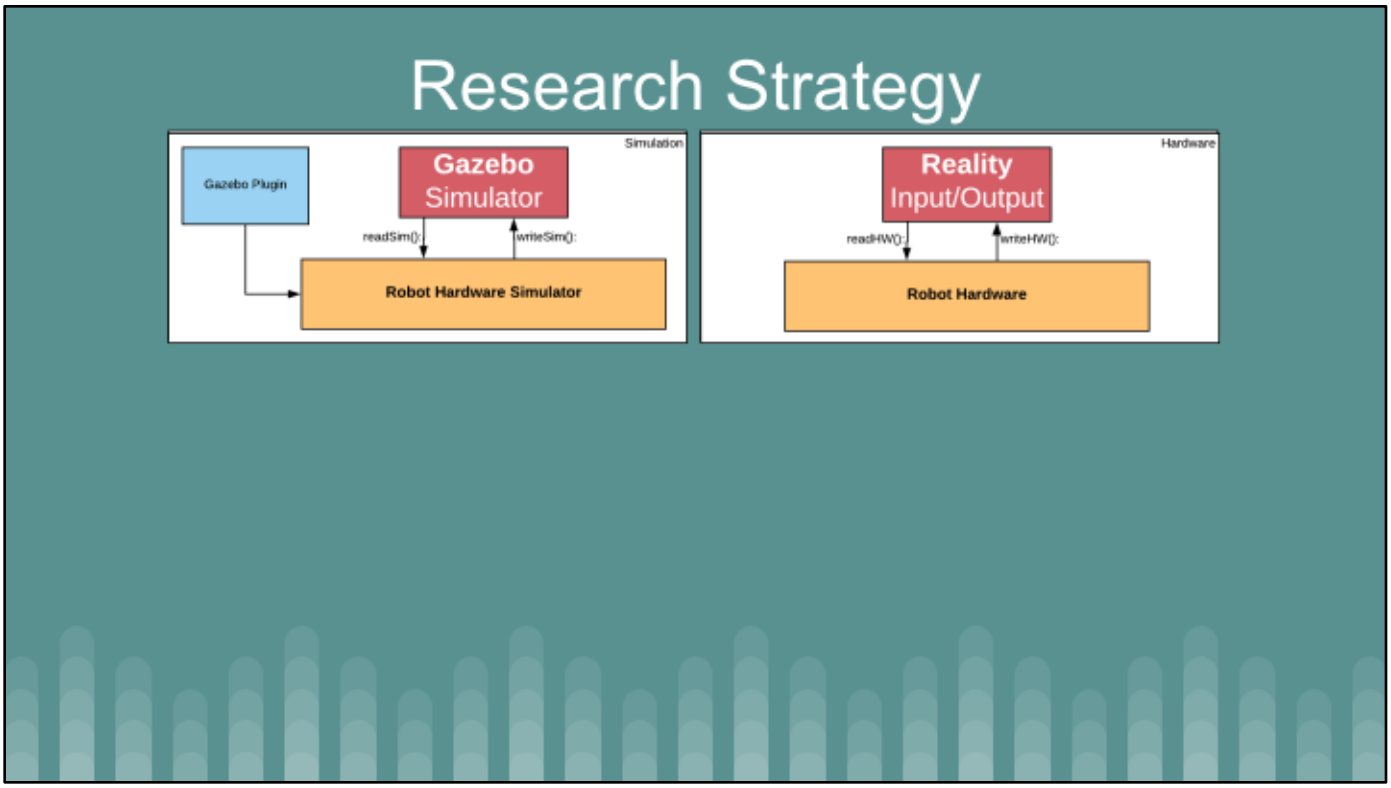
(ADAM)

Research Strategy



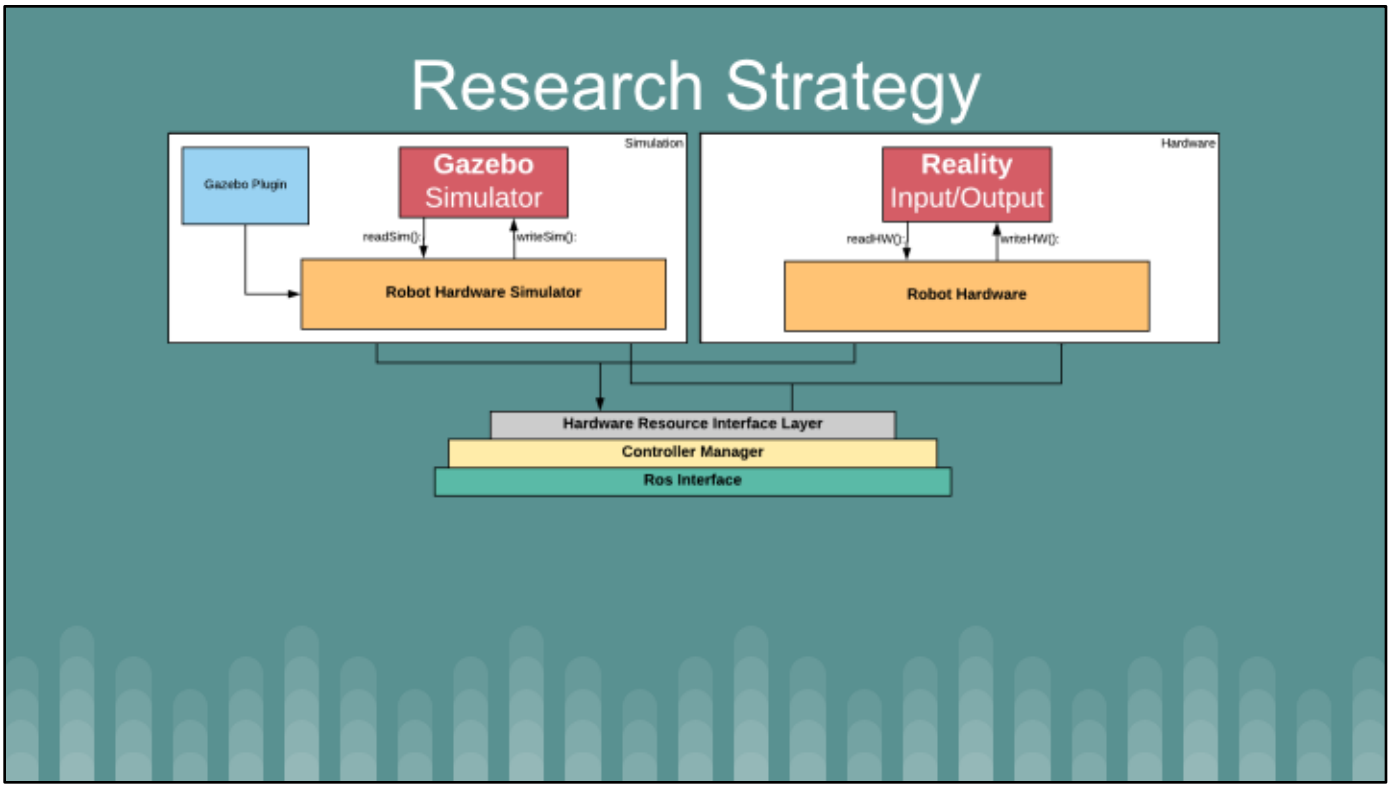
To drive research we have decided to use a robot from a past project, the pico rover.

This represents the physical device in the real world.

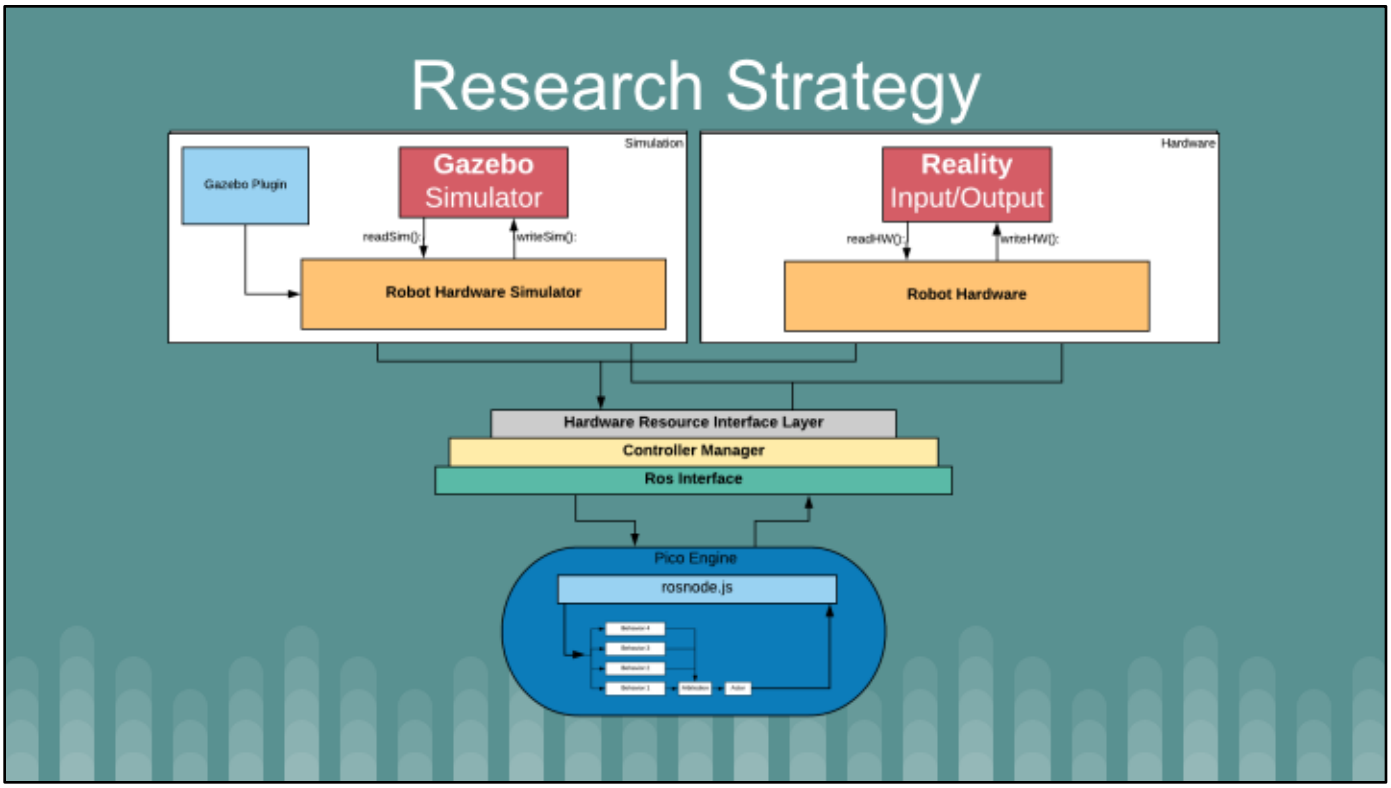


We want to do all our development in software. To accomplish this we have picked Gazebo Simulator.
Gazebo is a robot simulator that allows you to created an environment for a robot. simulate the robots hardware interactions with that environment.

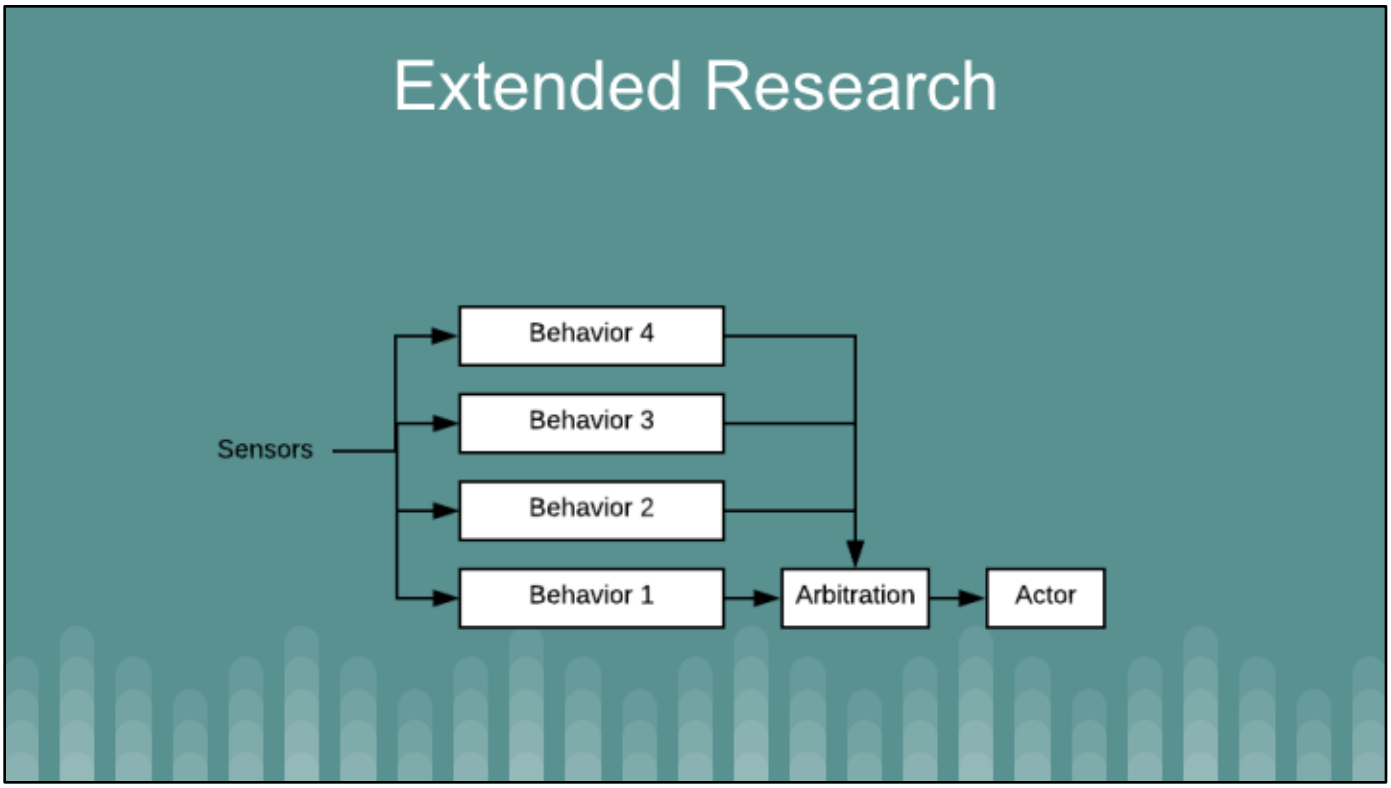
This represents the simulator and the simulated robot.



ROS stands for Robotics Operating System. A library for managing hardware.
We plan to use ROS to interface with the pico rover.
Gazebo supports the same interface layer.
This diagram shows that the simulator and the pico rover are now interchangeable behind the Ros Interface.

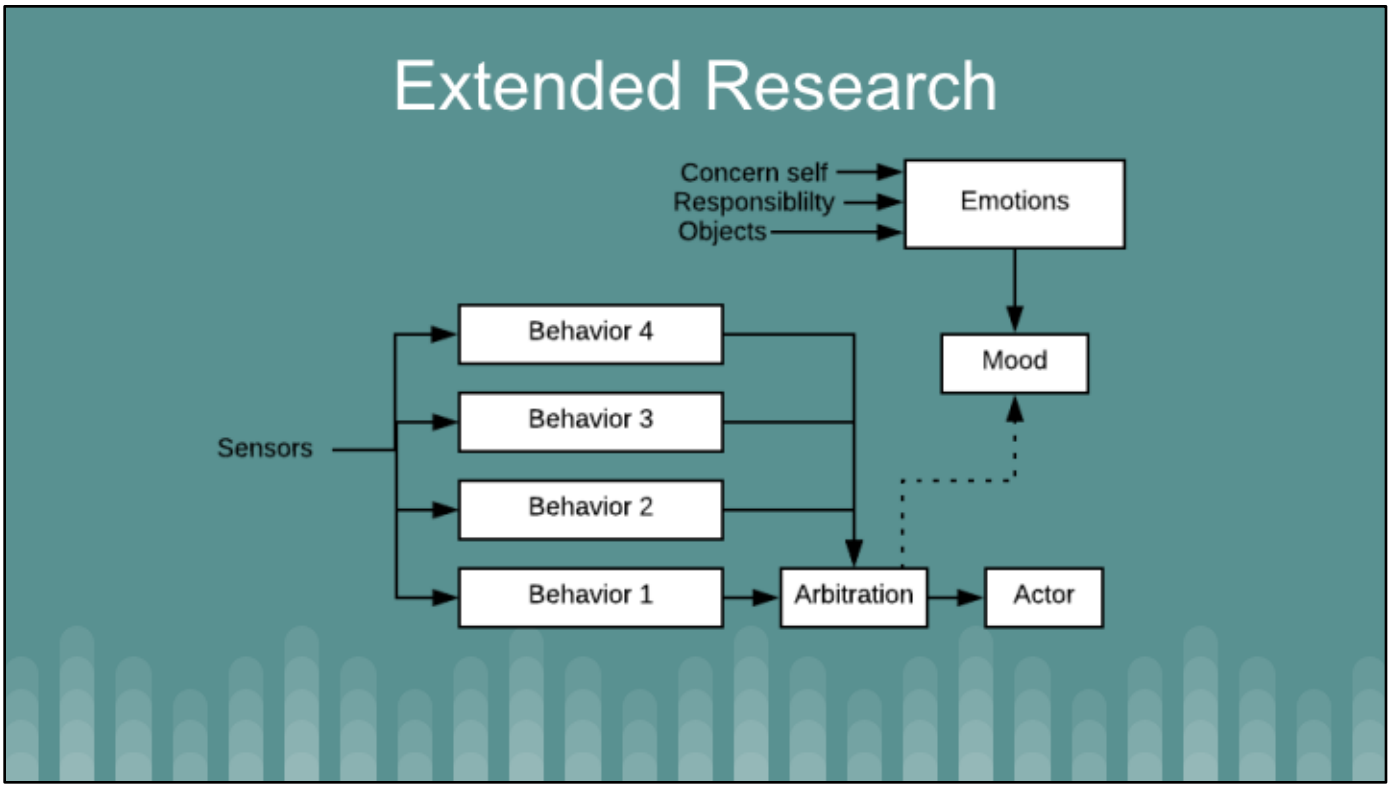


With the Ros Interface implemented, we plan to extend the pico engine to handle Ros communication.
This will allow us to develop subsumption architecture in the pico engine.

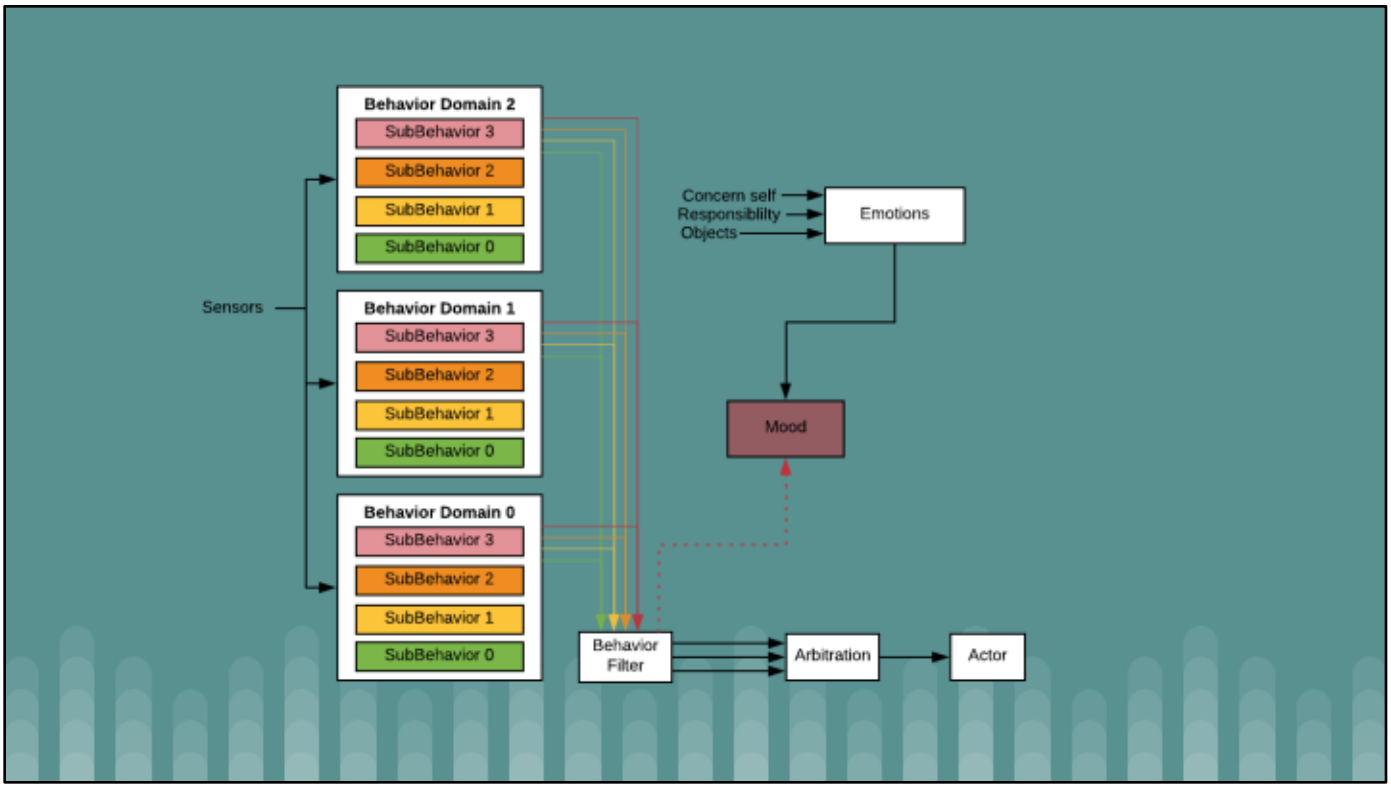


While our research is driven by implementing a behavioral based robot with the use of simulation design testing, we can apply what we are learning to the development of a smart house. I would like my smart house to have a personality, something that makes me laugh. With that thought we tried to imagine what it would take to incorporate emotion into the subsumption architecture.
Tradisional Arbitration is done by priority, what if instead we used emotion based Arbitration?

(ADAM)



We now can dynamically change the expressed behavior based on the programs mood state. The mood will be determined by analysis of sensor information pertinent to its representation of self.
WIth this new model we realize that Behaviors are no longer specific subroutines for a task, but subdomains of a task.

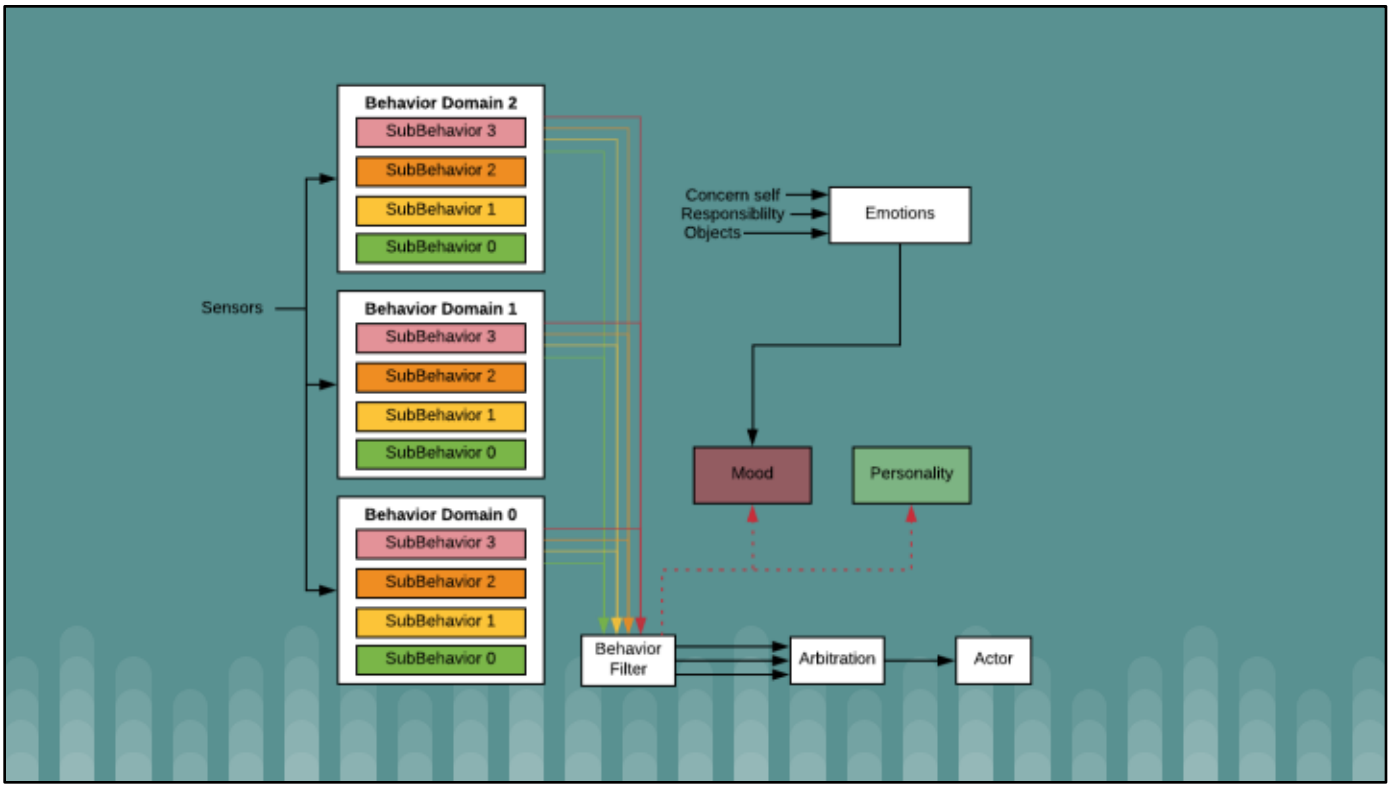


For example we now gain a different sub behavior for each behavior domain. With treating this as a emotion domain we can extract the idea out of arbitration and still provide flexibility in arbitration design.

This proposed mood based subsumption structure is significant because it directly applies to Domain Transfer problem of neural networks. That is, we can currently train a neural network on one domain set efficiently. This Structure allows you to strategically craft each domain and sub Behavior so you can effectively leverage the power of neural networks across multiple domains. You are not limited to the use of neural networks, you could build a hybrid system.

Emotion Arbitration extended

- Diagram of how we could implement jarvus
- Follows the structure described in society of minds
- http://sentic.net/hourglass-of-emotions.pdf

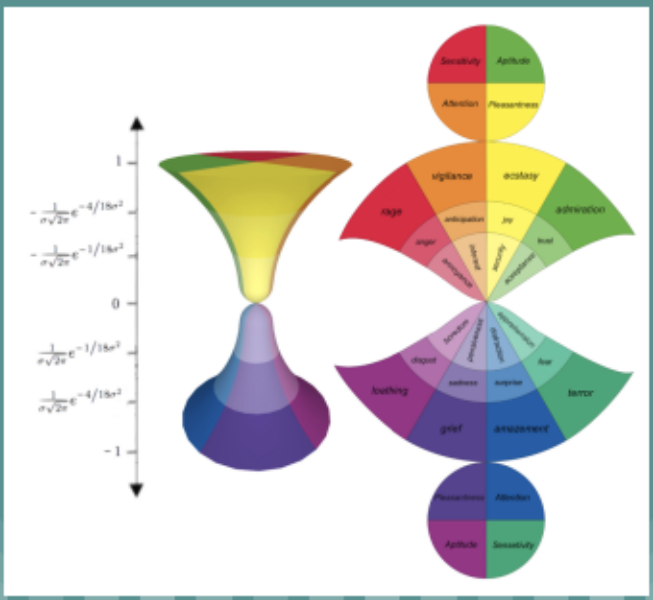


This idea is still evolving, we are now playing with the idea of multiple personal with in this system.

C-3PO Example

C-3PO is an example of where we would like to take the idea of emotion arbitration in a robot. Even though there's a lot more going on in the scene, C-3PO is concerned with R2-D2 repairing his own leg rather than the ship to escape the Empire. C-3PO expresses self awareness as well as selfishness through his actions in this scene and he is humorous at the same time. Again, just an example of wehre we would like to take emotion arbitration in a robot.

Hourglass of Emotion



While looking for what research has been done in emotion based arbitration we came across the Hourglass of emotions.

2011 international conference on Cognitive Behavioural Systems

Questions you should ask.

- Detailed progress
- Why not use neural nets
  - While training is young, use them as behaviors
- What's holding picos back….
- Where have you seen these ideas in publication..
  - Society of minds, hour glass of emotion

Bibliography

Tina. “Every Time C-3PO Whines and Complains in Star Wars.” *YouTube*, YouTube, 28 Feb. 2018, www.youtube.com/watch?v=Z_OjTojCNm0

Cambria, Erik, et al. “The Hourglass of Emotions.” *Sentic.net*, sentic.net/hourglass-of-emotions.pdf.

Brooks, Rodney A. “A Robust Layered Control System For A Mobile Robot .” Ieeeexplore, ieeexplore.ieee.org/stamp/stamp.jsp?tp=&amumber=1087032.

Jones, Joseph L., and Daniel Roth. Robot Programming: a Practical Guide to Behavior-Based Robotics. McGraw-Hill, 2004.

http://gazebosim.org/tutorials/?tut=ros_control

Citations
-