Robot Architectures

Henrik I. Christensen

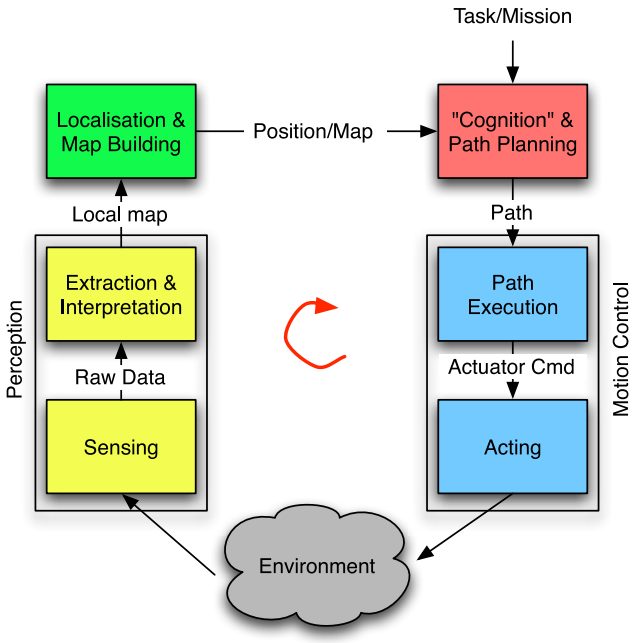
Contextual Robotics
UC San Diego
La Jolla, CA
hichristensen@ucsd.edu

Outline

Outline

- Homework 4?
- Robot Architectures
- Any other business

System Context



Objectives?

- Characterization of architectures
- Discuss some of the key dimensions in design
- Outline a number of common architectures
- Discuss challenges / principles

Question

What is an architecture?

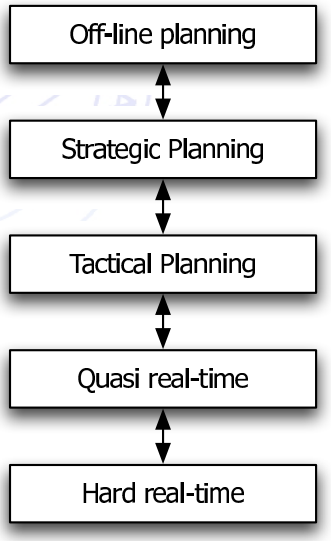
Architecture?

- The structuring principle(s) for design of systems that are highly targeted and efficient from a number of general methods and techniques
- Criteria:
  1. Support for parallelism
  2. Hardware targetability
  3. Support for modularity
  4. Robustness
  5. Run-time flexibility
  6. Performance effectiveness
  7. Documentation

Design Dimensions?

- Temporal decomposition
- Parallelism
- Control decomposition

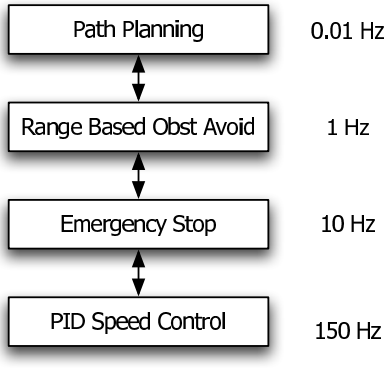
Temporal decomposition of architecture



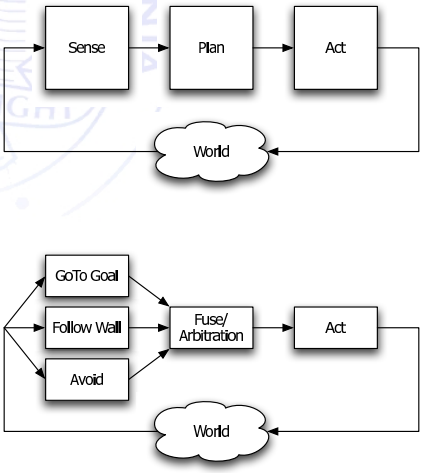
- Division according to temporal requirements
- Provides a coarse division of control
- Layering can be loosely synchronous

Sample Mobile Platform Decomposition

- Division to ensure safety
- Consideration of environmental dynamics
- Hardware might also influence the design

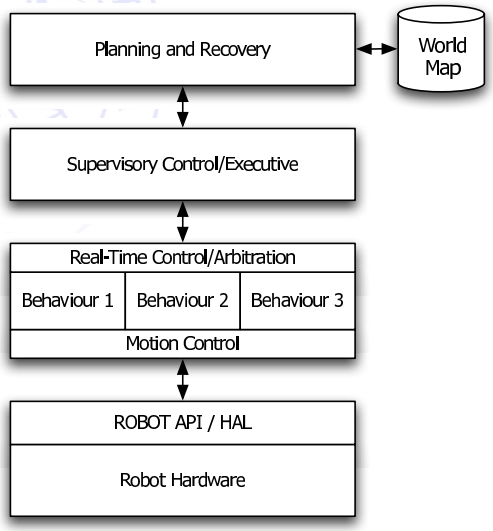


Control Decomposition



- Traditionally the sense-plan-act has been a well known method
- More recent systems have used a parallel decomposition often referred to as a reactive/behavior based system
- Behaviors are situated control modules with well defined context and control specifications

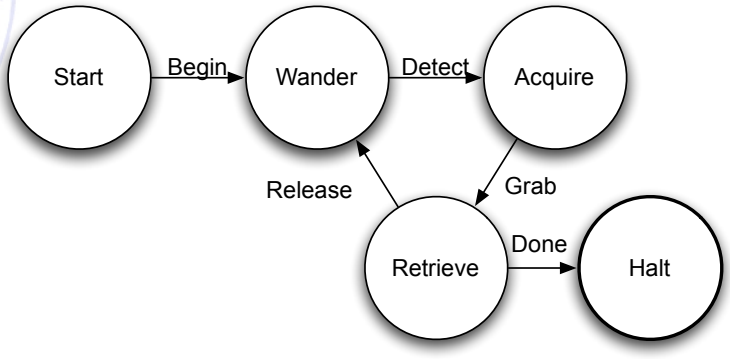
Hybrid Architectures



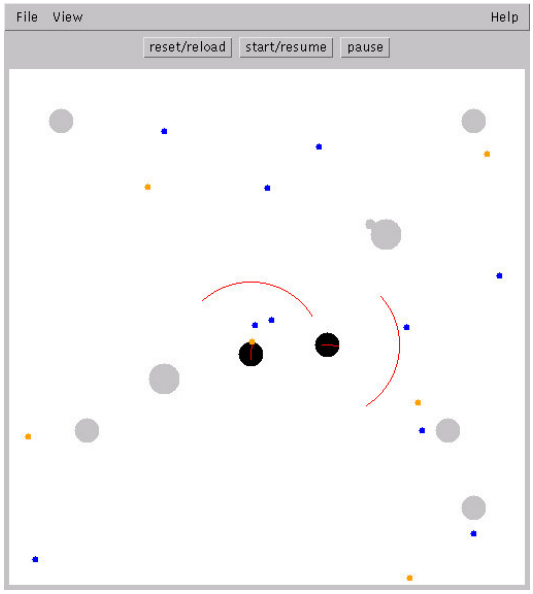
- The Hybrid Architecture Interleaves Deliberation (planning) with reactive control to have flexible handling of mission goals and obstacles. By far the most common architecture today

Outline

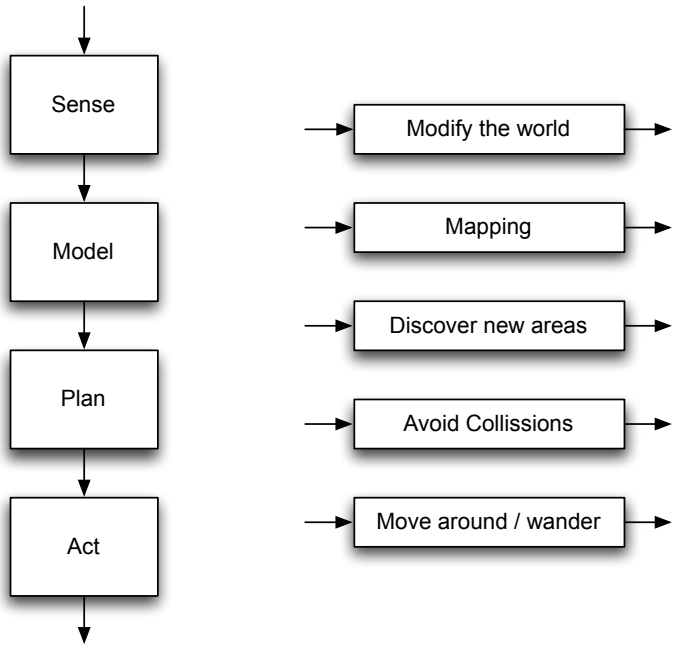
Foraging Example – FSA



Foraging in TeamBots



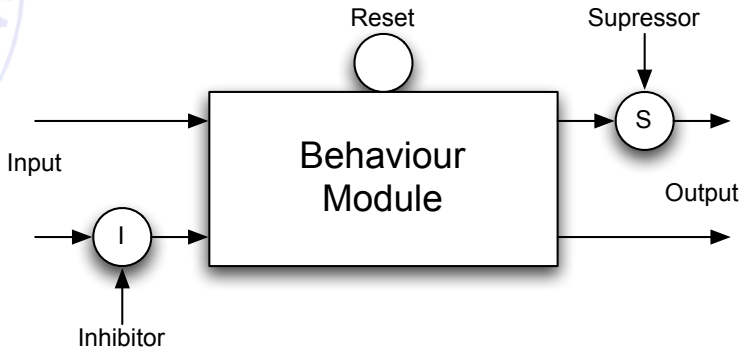
Horizontal vs Vertical Decomposition



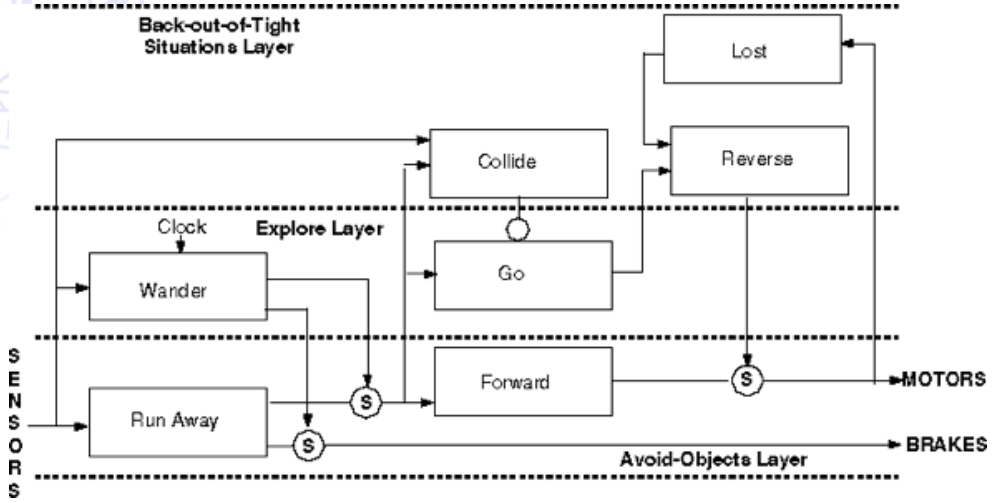
Subsumption Architecture

Name	Subsumption architecture
Background	Reactive architectures
Precursors	[Braitenberg, 83] [Walter, 53]
Design Method	Experimental
Developer	Rodney Brooks (MIT)
Response coding	Discrete / rule based
Coordination	Competitive
Programming	FSA or B-Language
Robots fielded	Allen, Genghis, Squirt, Toto Polly, Seymour,...
References	[Brooks 96] [Horswill 93]

Original Augmented FSA



Three layer robot



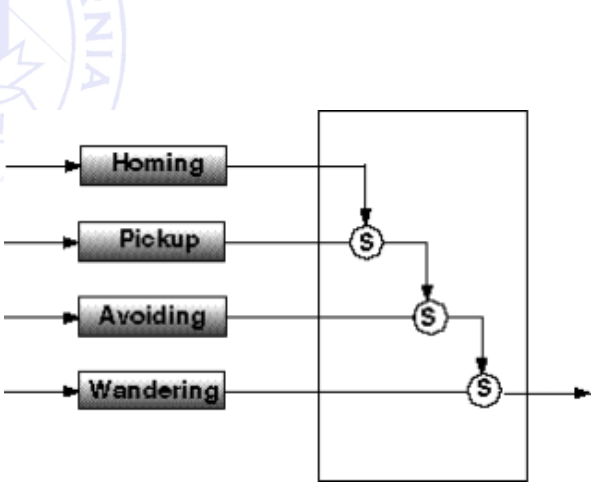
Coordination in subsumption

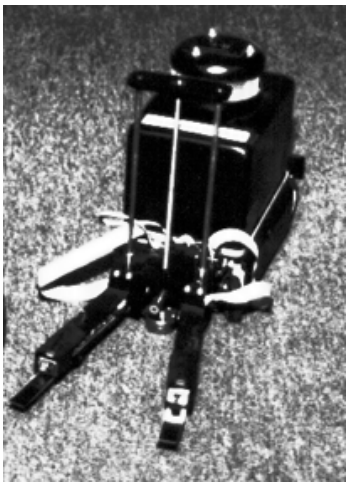
- Inhibition prevents signals transmitted from reaching actuators
- Suppression replaces signal being transmitted by the suppressing signal
- The end result is a priority based arbitration method

Design in Subsumption

- Qualitatively specify the behaviors needed for the task
- Compose and specify the robot's independent behaviors as a set of disjoint actions
- Determine the behavioral granularity
- Ground low-level behaviors onto sensors and actuators

Subsumption foraging robot - Nerd

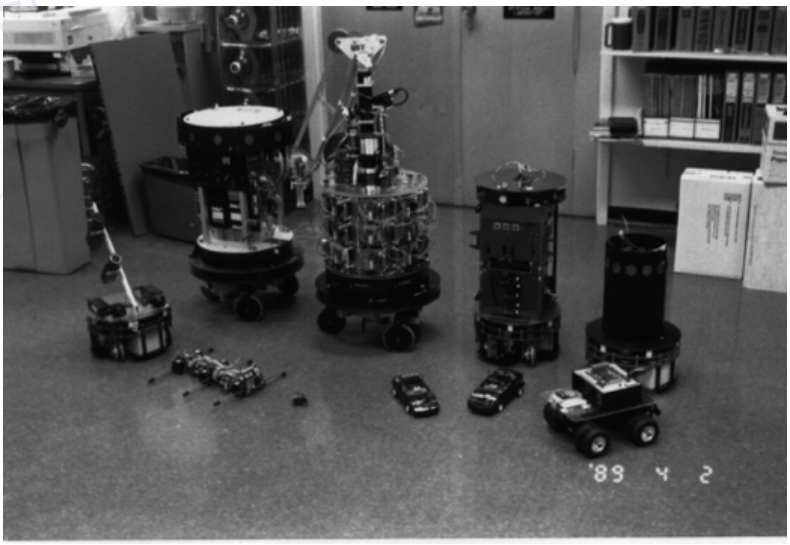




Subsumption evaluation

- Advantages
  - Compiles to hardware
  - Support for parallelism
  - Well adopted to niches
- Weaknesses
  - Poor run-time flexibility
  - Hardwired control
  - Behavioral reuse is often limited

Subsumption robots



Motor Schema

- Based on schema theory (Arbib, 1981)
- Large grain modularity
- Distributed concurrent agents
- Assemblage composition
- Strong cognitive / neuro-scientific evidence

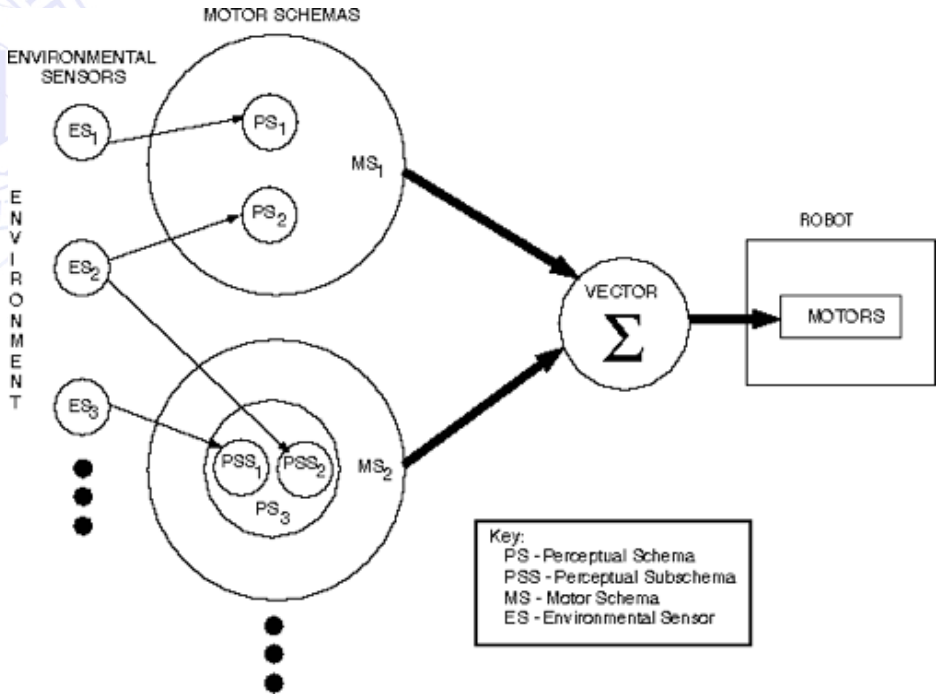
Design considerations

- Response represented as uniform vectors (Forces)
- Cooperative coordination through superposition
- Predefined hierarchy – arbitration / sequencing
- Arbitration not used – gain values control behavioral strength

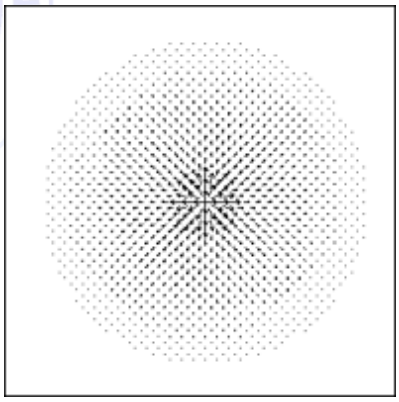
Motor Schemas

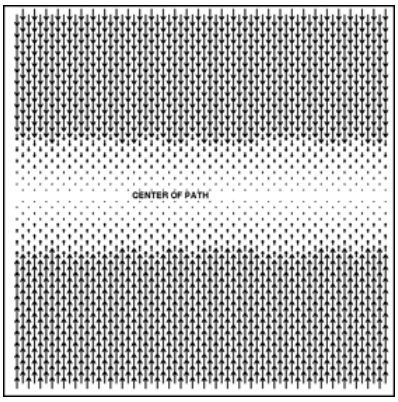
Name	Motor Schemas
Background	Reactive part of AuRA
Precursors	[Arbib 81] [Khatib 85]
Design Method	Ethologically
Developer	Ronald Arkin (GT)
Response coding	Potential field(s)
Coordination	Cooperative superposiion
Programming	parameterized behaviour libraries (e.g., TeamBots)
Robots fielded	HARV, George, Ren, Stimpy, Buzz, Blizzard
References	[Arkin 87] [Arkin 89] [Arkin 92]

Outline of Schema-based system

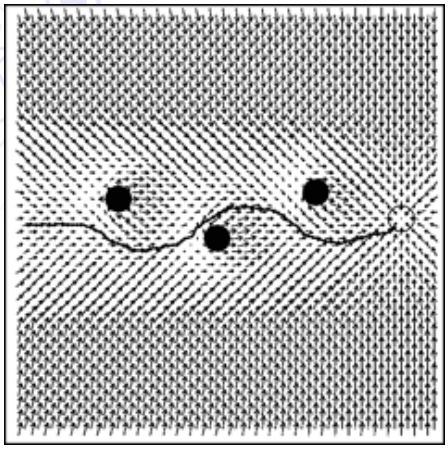


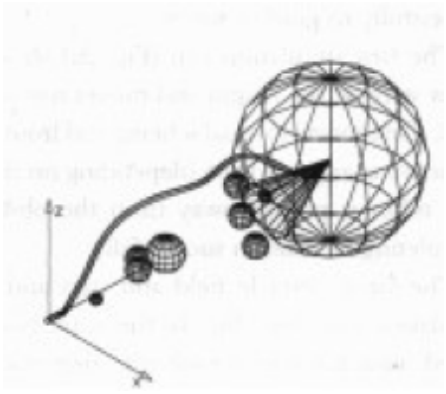
Representative behaviors





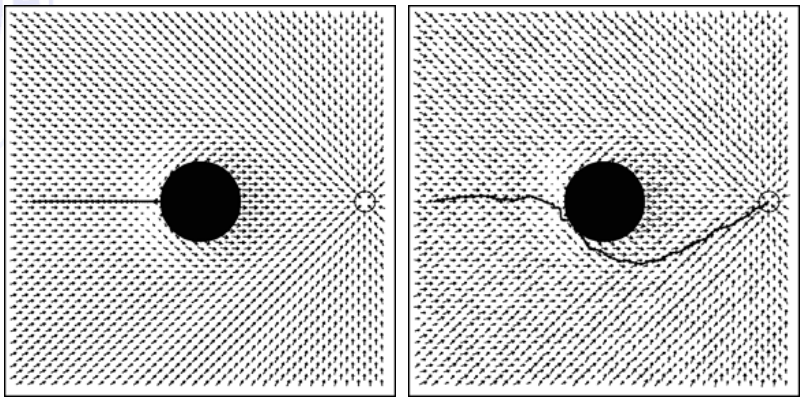
Example trajectories



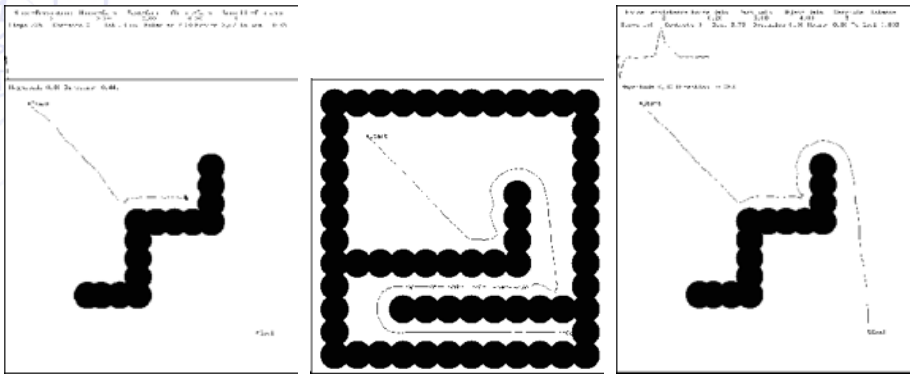


Animal Parallels

The effect of small noise



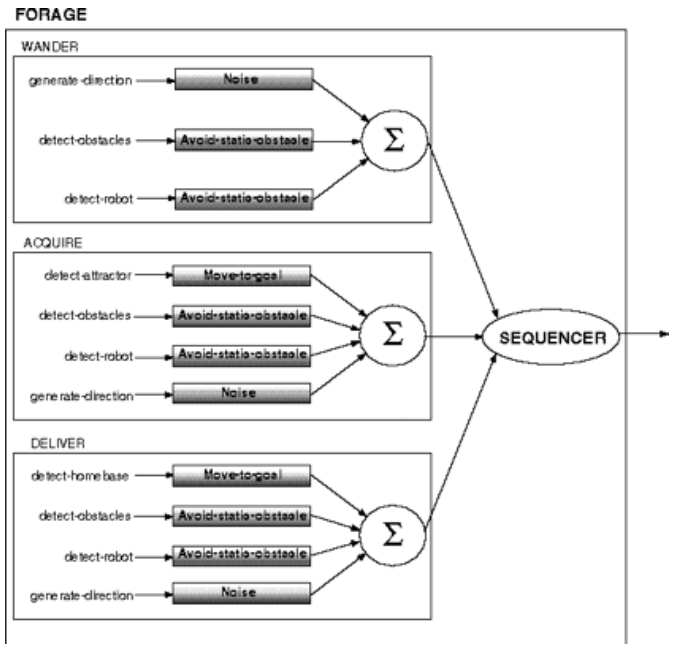
Avoid-past behavior



Design w. Motor Schemas

- Characterize motor behaviors needed
- Compose the most primitive level, use biological guidelines where appropriate
- Develop models to express reactions
- Perform simple simulations
- Determine perceptual needs to satisfy motor schema inputs
- Design perceptual scheme to enable this
- Integrate / test / evaluate / iterate

Foraging example



Evaluation - Motor Schema

- Advantages
  - Support for parallelism
  - Run-time flexibility
  - Timeline for development
  - Support for modularity
- Disadvantages
  - Niche targetability
  - Hardware retargetability

DAMN architecture

Name	DAMN (Dist Arch for Mobile Navigation)
Background	Fine grained subsumption
Precursors	[Brooks 86] [Zadeh 73]
Design Method	Experimental
Developer	Julio Rosenblatt (CMU)
Response coding	Discrete Vote Sets (Fuzzy sets)
Coordination	WTA arbiters
Programming	Mixed C++/Lisp
Robots fielded	DARPA ALV, Navlab ...
References	[Rosenblatt and Payton 89] [Rosenblatt 95]

DAMN architecture

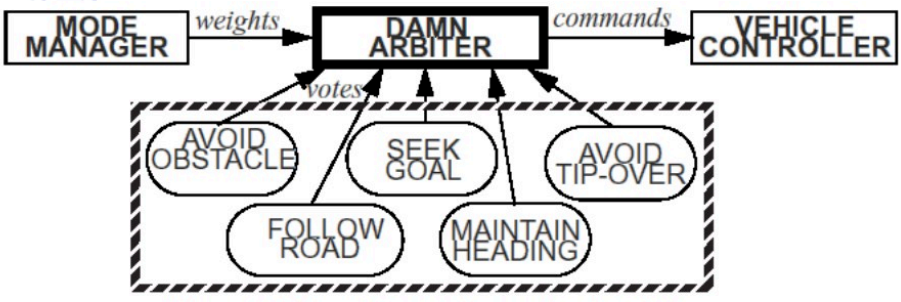


Figure 1: Behaviors sending votes to arbiter

DAMN - NAVLAB

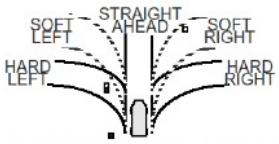


Figure 4: Arc evaluation in the *Obstacle Avoidance* behavior

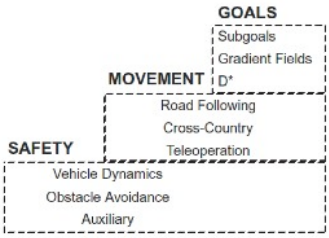


Figure 7: Evolutionary development in DAMN

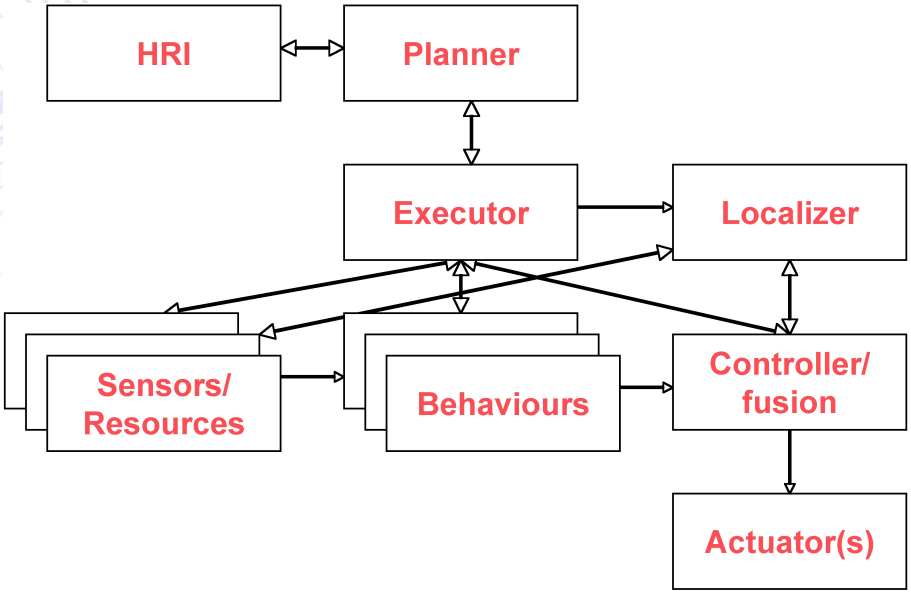
Hybrid Deliberative Architecture

Name	Hybrid Deliberative (AuRA)
Background	Interleaved Control
Precursors	[Brooks 86] [Arkin 86]
Design Method	Experimental
Developer	Ronald C. Arkin et al
Response coding	
Coordination	Context Arbiter
Programming	C/C++/C#/...
Robots fielded	Berra, ...
References	[Arkin 88] ...

BERRA – Architecture

Name	BERRA
Background	Scheme models
Precursors	[Arkin 86] [Arkin 88]
Design Method	Experimental / Situated activity
Developer	Oreback, Lindström and Jensfelt (KTH)
Response coding	Potential field
Coordination	Arbitration / Superposition
Programming	Behaviour Libraries
Robots fielded	Asterix, Huey, Dewey and Louie
References	[Lindström, Oreback and Christensen 00]

BERRA



Outline

Software Architectures

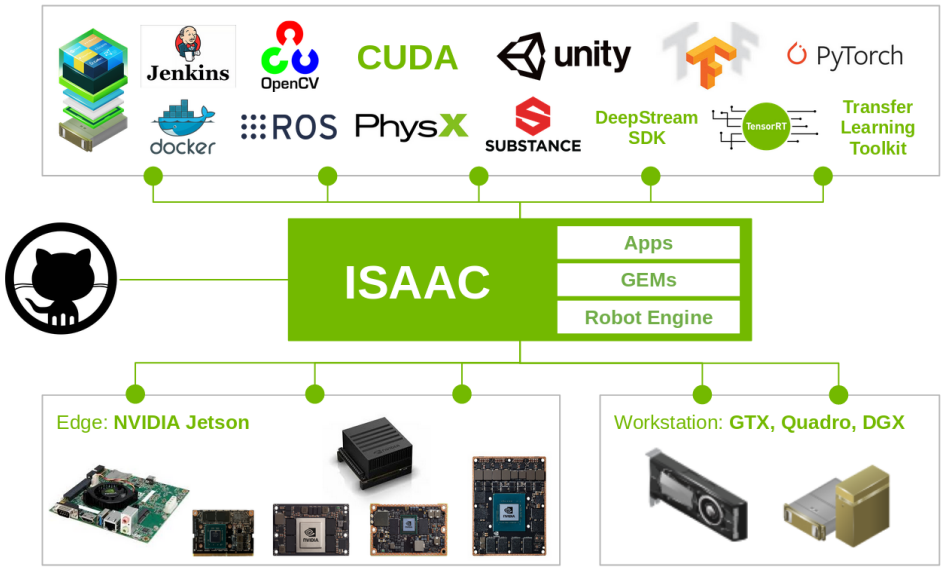
OROCOS Open Robot Control System
- Manipulator Oriented Open Source Control

Player-Stage Open Source Device Library
- Widely used OSS for robotics (predecessor of ROS)

ROS Robot Operating System 1 & 2
- Today the most widely adopted OSS software for robotics
- Comprehensive suite of tools

ISSAC
- NVIDIA entry into robotics architectures
- Strong simulation tools
- Slowly getting more open

ISSAC



Software Architectures - Short Comparison

System	Architecture	Language	Devices	Doc
OROCOS	React	C++	Manip	++
ORCA	HD	C++/Java	Mobile	++
ROS	Any	C++ / Python	Many	+++
ISSAC	Any	C++/Python	Many	+++

Outline

Summary

- Brief summary of major dimensions for architectures
- A few representative architectures
  - Subsumption, Voting Based, Sense Plan Act, Hybrids
- Outline of software systems