# MultiRobot Collaboration And High Level Language Decomposition

By

Ted Elhourani & Micah Lesmeister

# Agenda

- Introduction
- Our Goal
- Previous Work – Modeling Robot Behavior
- Previous Work – Robotic Middleware
- Quick Examples
- Our Contribution
- Conclusion and Future Work

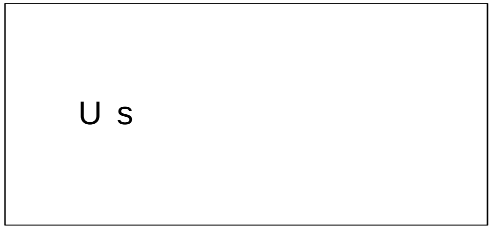
# Introduction

- Robotics Has Not Been Adapted Into Widespread Use For One Reason:
  - User Complexity
  - Limited Capabilities

# Our Goal

- Create A Team Of Robots That Can Perform A Variety Of Tasks In Unknown Environments
- Create An Interface And Language Any One Can Use
- Create A Team Of Robots That Can Interact With Each Other With Little Or No User Intervention

# System Overview



# Overview Multi-robot Systems

- Single Robot Architectures
- Multi-robot Architectures
- Overview of COP/DCOP
- DCOP in the Sensor Network Domain / Real-time Systems
- Application in the Multi-robot domain

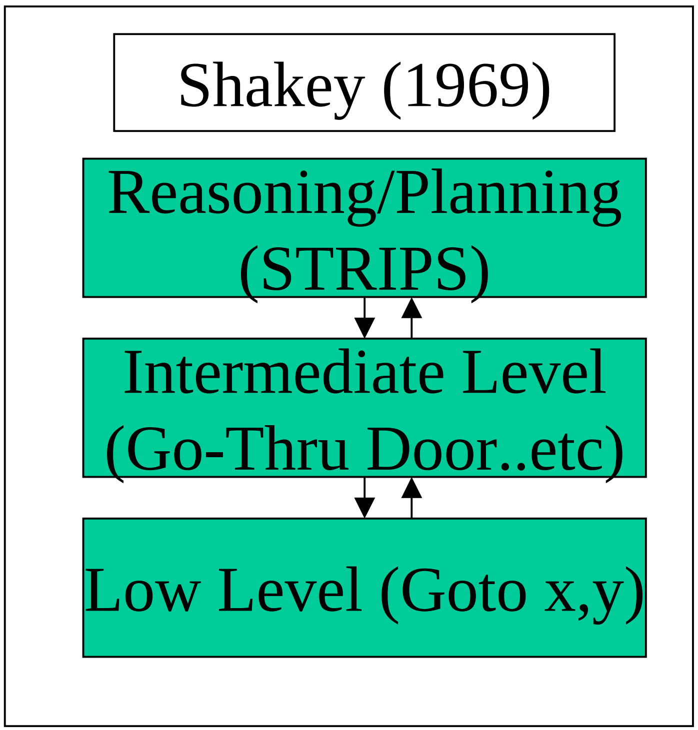
# Single Robot Architectures

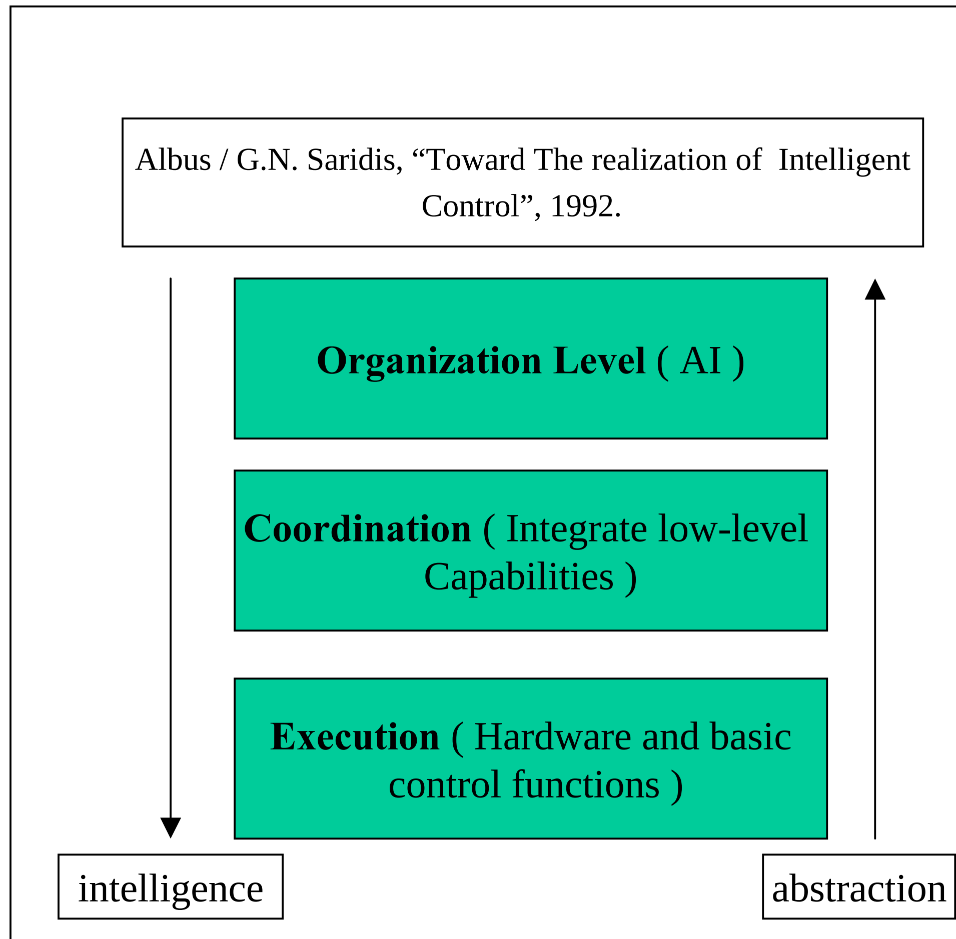
- Deliberative Architectures
- Reactive (Behavior-based Architectures)
- Hybrid Architectures

# Single Robot Architectures

**Deliberative Architectures: Examples**

- **Reliance on accurate models of the world.**



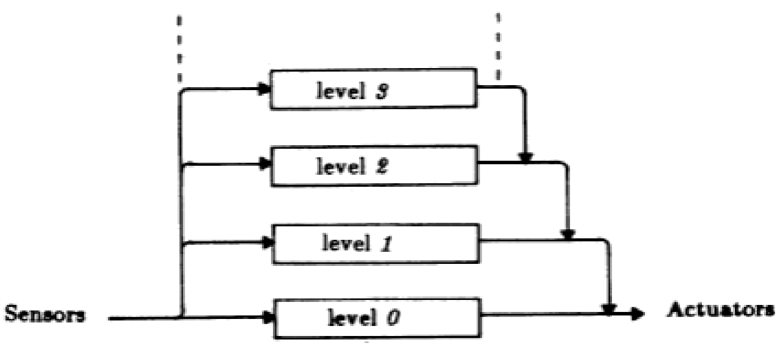
# Single Robot Architectures

**Reactive Architectures : Examples**

- **Limited representations.**

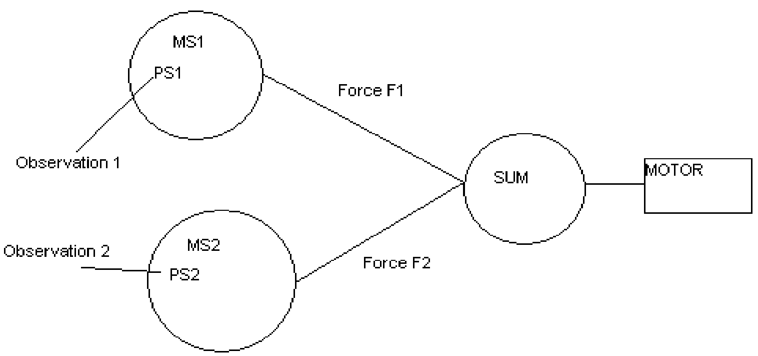
Subsumption: Rodney Brooks, 1986

- Behaviors vs. Functions
- Incremental Build-up
- Higher layers subsume lower ones




Motor Schemas: Ronald Arkin, 1987

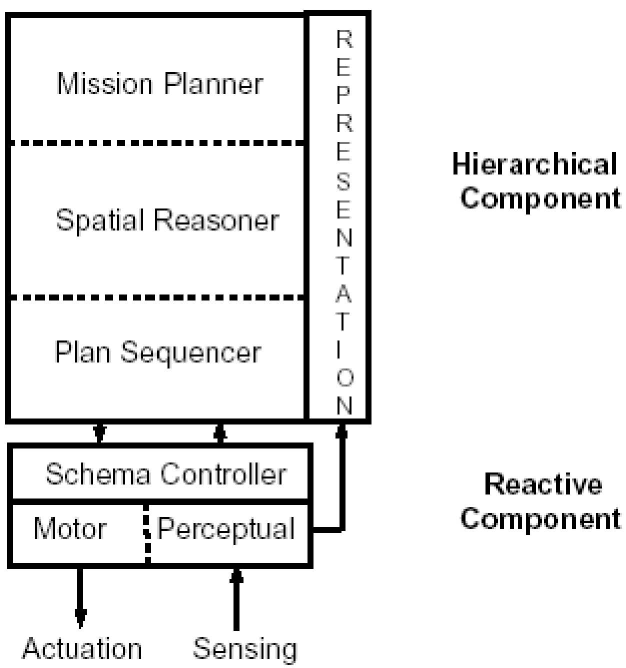
- Motor schemas are behaviors
- Each Motor Schema has its own perceptual schema



# Single Robot Architectures

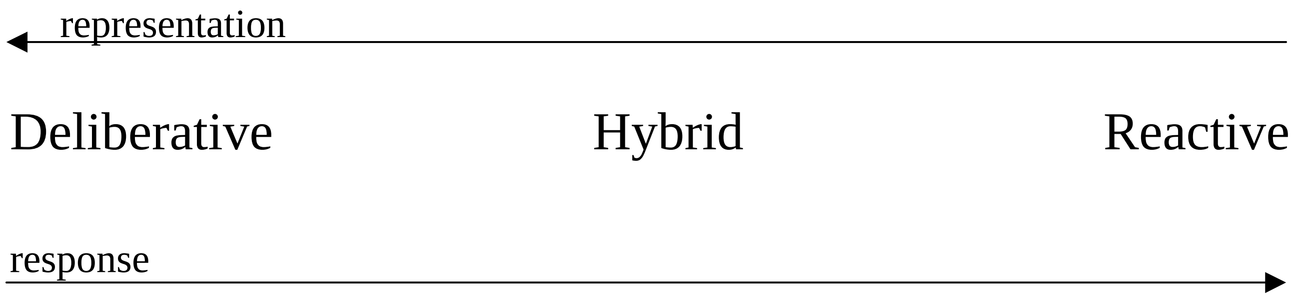
## Hybrid Architectures : Example

R.C. Arkin, “Aura: Autonomous Robot Architecture” 1980’s



# Single Robot Architectures

## Comparison



# Overview Multi-robot Systems

- Single Robot Architectures
- Multi-robot Architectures
- Overview of COP/DCOP
- DCOP in the Sensor Network Domain/ Real-time Systems
- Application in the Multi-robot domain

# Multi-robot Architectures

- Centralized Approaches
- Behavior-based
- Market-based

# Multi-robot Architectures

## **Centralized Approaches**

- One robot plans and synchronizes the actions of all team members.

Advantages:

- The plan is optimal

Disadvantages:

- Slow response in dynamic environments
- Not scalable (in terms of umber of robots)
- Communication congestion
- Failure of the leader robots leads to the failure of the whole system

# Multi-robot Architectures

## Centralized Approaches

### Examples

• B. Brummit and A. Stentz, "Dynamic Mission Planning for Multiple Mobile Robots," Proceedings of the IEEE International Conference on Robotics and Automation, April, 1996.

• Reid Simmons, David Apfelbaum, Dieter Fox, Robert P. Goldman, Karen Zita Haigh, David J. Musliner, Michael Pelican, Sebastian Thrun, "Coordinated Deployment of Multiple, Heterogeneous Robots," In Proceedings of the Conference on Intelligent Robots and Systems (IROS), Takamatsu Japan, October 2000.

# Multi-robot Architectures

## **Behavior-based Approaches**

•Ever robot plans and executes its own actions without prioritizing global benefit.

Advantages:

• Solves most of the problems of the centralized approaches

• Simple to implement

Disadvantages:

• The plans are not optimal.

# Multi-robot Architectures

## Behavior-based Approaches

### Examples

• Lynne E. Parker, "ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Coopration," IEEE Transactions on Robotics and Automation, vol. 14, no. 2, April 19998:220-240.

• Barry Brian Werger and Maja J Mataric, "Broadcast of Local Eligibility for Multi-Target Observation," In Lyne E. Parker, George Bekey, and Jacob Barhen editors, Distributed Autonomous Robotic Systems 4, pages 347-356, Springer-Verlag, 2000.

# Multi-robot Architectures

## Market-based Approaches

• Robots act largely independently, but take into account the capabilities of other robots, hence achieving more optimality in task allocation.

Advantages:

• Has all advantages of behavior-based architectures and occasionally allows centralized planning.

Disadvantages:

• Implementation complicated.

# Multi-robot Architectures

## Market-based Approaches

### Examples

- Sylvia Botelho and Rachid Alami. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. In Proc. Of the IEEE Intl. Conf. On Robotics and Automation (ICRA), pages 1234-1239, Detroit, MI, May 1998.

- Brian P. Gerkey and Maja J Mataric. Sold!:Auction methods for multi-robot coordination. IEEE Transactions on Robotics and Automation, 18(5):758-768, October 2002.

# Overview Multi-robot Systems

- Single Robot Architectures
- Multi-robot Architectures
- Overview of COP/DCOP
- DCOP in the Sensor Network Domain/ Real-time Systems
- Application in the Multi-robot domain

# Overview of COP/DCOP

- What is COP?
  - A generalization of CSP.
  - **CSP** can be modeled as a **Constraint Network**.
  - **COP** can be modeled as a **Constraint Network augmented with Cost Functions**.

- Formally:
  - $C = \langle X, D, C_h, C_s \rangle$ Constraint Graph
  - $X = \{x_1, x_2, \ldots\ldots x_n\}$ Set of variables.
  - $Q = \{q_1, q_2, \ldots\ldots q_m\}$ Set of subsets of X.
  - $C_s = \{f_1, f_2, \ldots\ldots, f_m\}$ Set of cost functions $f_i$, with $q_i$ as the scope of $f_i$ (soft constraints).
  - $C_h = \{c_1 \ldots\ldots\ldots\}$ Set of hard constraints
  - $\alpha = \{a_1, \ldots\ldots, a_n\}$ Set of domains.
  - $F(\alpha) = \sum_{j=1}^{l} F_j(\alpha)$ Global Cost Function
  - The goal is to find the optimal instantiation $\alpha^o$ satisfying the hard constraints $C_h$, such that $F(\alpha)$ is optimal.

$F(\alpha^o) = \max( F(\alpha) )$ or $\min( F(\alpha) )$.

# Overview of COP/DCOP

- Example:

Auction Scenario: Every bidder can place 1 bid over multiple items.

S= {$s_1$,…….$s_n$} Set of Items

B={$b_1$,……..$b_n$} Set of Bids

$b_i = (S_i, r_i)$ $S_i \subseteq S$ and $r_i$ is the cost of the bid.

The problem is:

Find a set of bids B’ $\subseteq$ B such that:

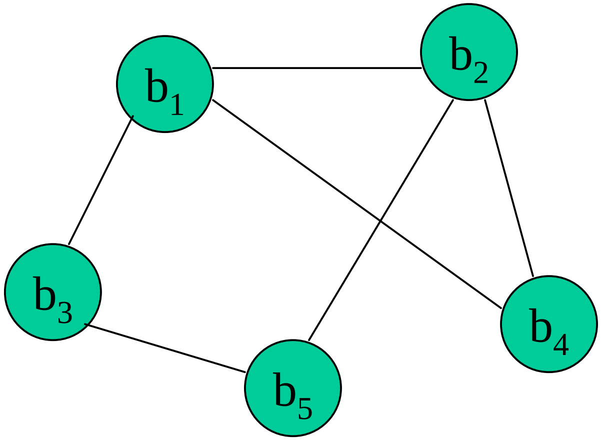
- No two bids share the same item
- Maximize C(B’) = $\sum_{b_i \in B'} r_i$

# Overview of COP/DCOP

- Modeling the Problem as a COP
  - $B = \{b_1, \ldots\ldots b_n\}$ Set of bits (the nodes)
  - $D = \{d_1, \ldots\ldots d_n\}$ Domains Set
  - $C_s = \{f_1 \ldots\ldots f_n\}$ Soft constraints, where $f_i(b_i) = r_i$ if $b_i = 1$, otherwise $f_i(b_i) = 0$
  - $C_h$ set of hard constraints $c_{i,j}$, where $c_{i,j}$ is the set of all bid pairs $(b_i, b_j)$ such that the pair $(b_i = 1, b_j = 1) \notin c_{i,j}$.

**Example:**

- $B = \{b_1=\{1,2,3,4\}, b_2=\{2,3,6\}, b_3=\{1,5,4\}, b_4=\{2,8\}, b_5=\{5,6\}\}$
- $D = \{0,1\}$
- Costs of bids 1 to 5: {8,6,5,2,2}



# Overview of COP/DCOP

## What is DCOP?

A generalization of DCSP.

**DCOP** can be modeled as a **Constraint Network**.

Nodes represent agents.

## Formally:

$C = \langle X, D, C_h, C_s \rangle$ Constraint Graph

$X = \{x_1, x_2, \ldots\ldots x_n\}$ Set of variables each belongs to one agent.

$Q = \{q_1, q_2, \ldots\ldots q_m\}$ Set of subsets of X.

$C_s = \{f_1, f_2, \ldots\ldots, f_m\}$ Set of cost functions $f_i$, with $q_i$ as the scope of $f_i$ (soft constraints).

$C_h = \{c_1 \ldots\ldots\}$ Set of hard constraints

$\alpha = \{a_1, \ldots\ldots, a_n\}$ Set of domains of variables in X.

$F(\alpha) = \sum_{j=1}^{l} F_j(\alpha)$ Global Cost Function

The goal is to find the optimal instantiation $\alpha^o$ satisfying the hard constraints $C_h$, such that $F(\alpha)$ is optimal.

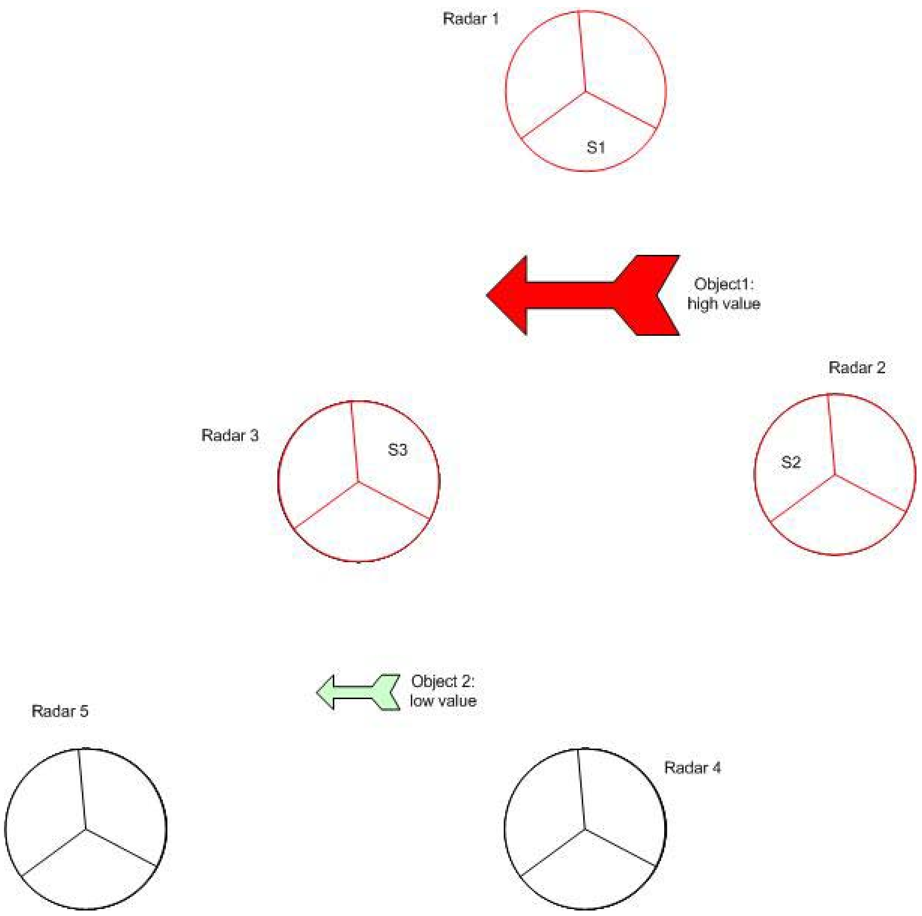
$F(\alpha^o) = \max( F(\alpha) )$ or $\min( F(\alpha) )$.

# Overview Multi-robot Systems

- Single Robot Architectures
- Multi-robot Architectures
- Overview of COP/DCOP
- DCOP in the Sensor Network Domain/ Real-time Systems
- Application in the Multi-robot domain

# DCOP in the Sensor Network Domain/ Real-time Systems

The Sensor Network : A Resource Allocation Problem



# DCOP in the Sensor Network Domain/ Real-time Systems

<u>Formulation of the Resource Allocation Problem as a DCOP</u>

T set of all possible tasks.

$T_p \subseteq T$, set of all tasks that are present.

$X = \{t_1, t_2, \ldots\ldots t_n\}$ Set of all present tasks.

$D = \{allocated, ignored\}$

$C_s = \{f_1, \ldots\ldots\ldots, f_n\}$ Cost of ignoring task $f_i$ (directly proportional to the value of the object).

**Problem?**

Assign values from {allocated,ignored} to all the present tasks such that:

Minimize: $C(T_p) = \sum_{t_i \in Tp} r_i$ , where $r_i = f_i$ if task i is allocated, 0 otherwise.

# DCOP in the Sensor Network Domain/ Real-time Systems

Concerns in real-time systems:

- Uncertainties (e.g. confusion about the presence of a task)
- Real-time system R.T. constraints (e.g. Fast allocation)
- Dynamic environments (e.g. automatic and frequent update)

# DCOP in the Sensor Network Domain/ Real-time Systems

## Tambe & Al.: 2 layers approach

• Adopt-SC: A distributed algorithm that can find the optimal solution in the sensor network domain relatively fast.

• Localized Reasoning: A fast probabilistic reasoning module.

# Overview Multi-robot Systems

- Single Robot Architectures
- Multi-robot Architectures
- Overview of COP/DCOP
- DCOP in the Sensor Network Domain/ Real-time Systems
- Application in the Multi-robot domain

# Application in the Multi-robot domain

## Multi-robot Task Allocation

- Given n robots and t tasks (sub-plans), How can tasks be assigned to each of the robots such that an optimal solution is reached ?
- Approaches in the three main architectures:
  - –Deliberative: Subplans are generated and execution is synchronized by the leader robot.
  - –Behavioral: Each robot plans for itself, when robot executes a sub-plan all the robots are notified.
  - –Market-based: Robots plan for themselves, however they try to trade tasks according to their knowledge of each robot current state and capabilities.

# Application in the Multi-robot domain

**Formulation of the task-allocation problem as a DCOP:**

- (Robot,Task) pairs can be assigned a cost using a cost function.
- The optimal assignment of tasks of robots would minimize the total cost.

**Challenges:**

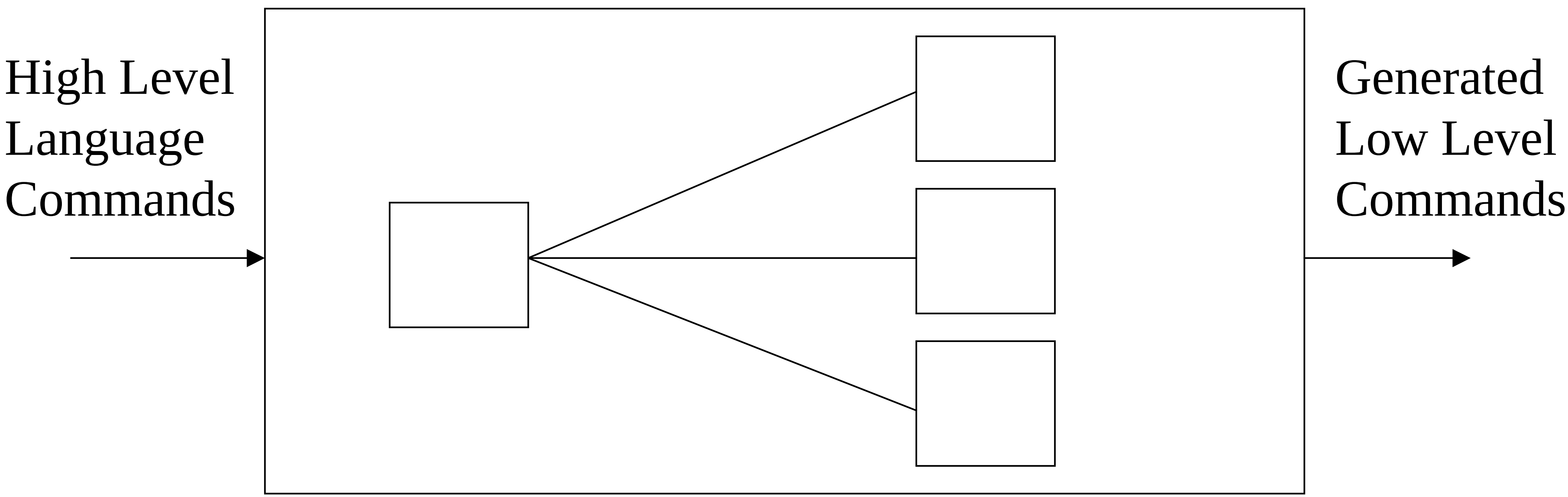
- Applicability of existing DCOP algorithms (r.t. constraints, communication efficiency, optimality requirements…etc)
- Abstracting domain specific components (Uncertainties in robot sensing and actuation …etc)

# Application in the Multi-robot domain

**Evaluation Dimensions:** (Depends on the settings: Robots &environment)

- –Scalability (number of robots)
- –Heterogeneous vs. Homogeneous
- –Predictable environments?
- –Communication channels (B.W. …)
- –Computational complexity (Real-time constraints)
- –Optimality of plans.

# System Overview
# Behavior Compiler



# Georgia Tech – Mission Lab

- Multiagent Robot Control Toolkit:
  - Behavior ‘Assembler’ For Military Applications (Editor) ←
  - Robot Console Program (Robot Execution Environment)
  - Compiler – Converts Behavior Into Applicable C++ Code That The Robots Run
  - 2D Simulator For Behavioral Testing

# Georgia Tech – Mission Lab

1. Finite State Automata Diagram Entered By The User
2. Further Motion Development Has To Be Defined Using Their Configuration Description Language (CDL)

# Georgia Tech – Mission Lab

- Advantages
  - Creating New Missions Is Faster Than Coding Them From Scratch
  - Allows The ‘Building’ Of More Complex Behaviors Using Smaller Ones Already Created
- Disadvantages
  - Use Still Limited To Programmers (Makes Use Of Recursion, Instances, among other C.S. practices)

# Mobile Autonomous Robot Software

- General Research To Provide:

  “Develop *complete*, *effective* and *scalable* software for autonomous robot teams. Demonstrate robot teams with integrated perception, reasoning, learning, communication and cooperative strategies that solve complex multiagent tasks.”

- Carnegie Mellon University, University Of Pennsylvania, Georgia Tech

# High Level Language Decomposition

- Generally Used In Distributed Computing
  - Makes The Low Level Actions Invisible To The User

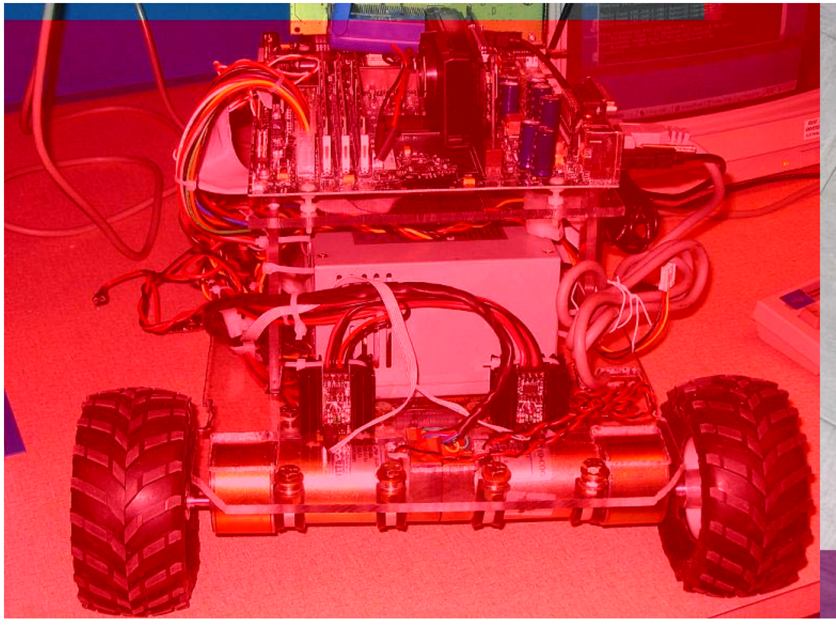
# Decomposition Example

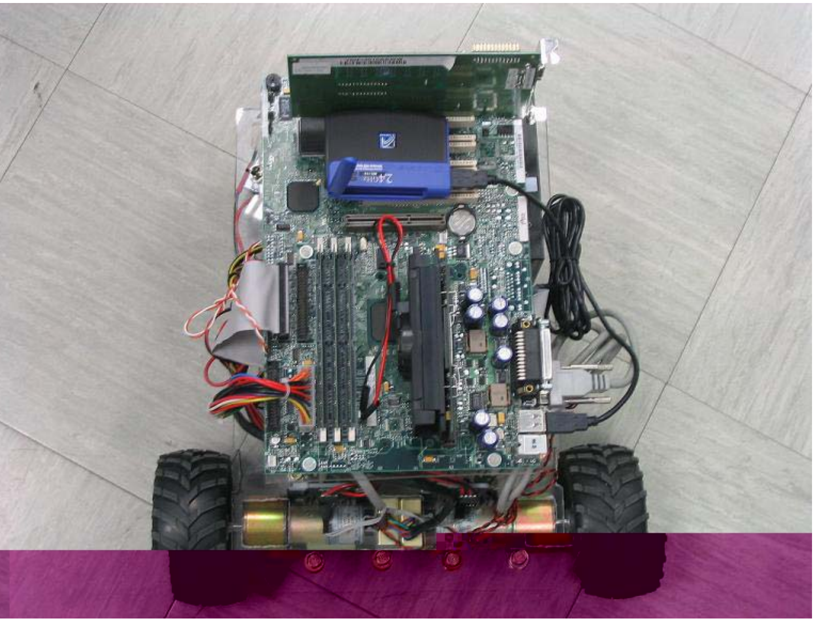
- Consider
  - Want The Team Of Robots To Locate A Black Object In A Desired Area
  - Command: find( 1 Object, Black )
- The Minimum Set Of Low Level Behaviors Needed Would Be:
  - Obstacle Avoidance, Distributed Area Search

# Decomposition Example

- Consider:
  - Want The Team Of Robots To Find And Move All Objects Colored Black To A Central Location
  - Command: find( All Objects, Black )
- The Minimum Set Of Low Level Behaviors Needed Would Be:
  - Obstacle Avoidance, Distributed Area Search, Move Object, Return To Last Searched Position

# Our Test Platform





# Our Test Platform

- Group Of 2 Identical Robots And 1 Host Computer
- Onboard Computing
  - Desktop Motherboard W/ 333MHz Processor
  - 384MB RAM
- Inter-robot Communication
  - 802.11b Wireless Ethernet Adapter
  - Communication Software: Open Source Multiagent Communication Using TCP/IP From Georgia Tech

# Our Test Platform

- Odometry
  - Custom Built Microcontroller Based Control Board To Count Odometer Provided By The Pittman Motors
- Power
  - DC-TO-DC Power Supply
  - Input: 24V Supplied By 2 Sealed Lead Acid Batteries (SLA)

# Contribution

- A new hybrid architecture
- Achieves higher performance in terms of optimality and complexity in a restricted set of real-world applications (Scenarios).
- Make robotics more accessible to the general public
- Formalize processes for translating high-level instructions to low level robot behaviors

# Future Work

- Continue Hardware Implementation Efforts.
- Learn more about DCOP algorithms and design evaluation methodologies (In collaboration with Sumit)
- Enrich knowledge of different multi-robot architectures.
- Improve skills in the robot uncertainty area.
- Formalize the task-allocation problem as a DCOP.

# Future Work

- Create The High-Level Language For User Interface
- Formalize High-Level To Behavior Conversion
- Develop Simulator For Testing Purposes

Questions ?